**QR SDK MANUAL**

***SET UP GUIDE:***

**SDK Microservice:**

Install node.js: <https://nodejs.org/en/> **(4.2.x)**

Install dependencies: npm install express node-qr-image

Start the server: npm ./app.js [win: node app.js]

Testing the server: navigate to <http://localhost:3000/> in browser

**SDK Web Client:**

Open the Web Client: With server started, open the index.html file in the root of directory in Firefox.

**SDK Sample Mobile Client:**

Install Cocoapods: <https://cocoapods.org/>

Install project Pods: pod cache clean –all

Update project Pods: pod update

Open project: open ./QRSDK.xcworkspace **(Requires Xcode 7.2)**

Deploy to device: Select device and press ‘Run’

***USAGE GUIDE:***

**Step 1:** Start the node microservice.

**Step 2:** Open the app on mobile device.

**Step 3:** Openthe web client in Firefox.

**Step 4:** Scan the QR code on screen with the mobile client.

**Step 5:** Allow access to web cam in web client.

**Step 6:** Use the web client to scan the new QR code on mobile device screen.

***COMPONENT BREAKDOWN:***

**SDK Microservice:**

Node microservice with a HTTP server and a RESTful GET endpoint. When the GET endpoint is called the service generates a random QR Code and returns it to the client in SVG format.

**SDK Web Client:**

The web client is split up into two components **1** – A page which displays the QR image from the node microservice and **2** – A page which uses the webcam to scan QR codes and convert them to a string.

**1**: The display page makes a GET request to the microservice over HTTP and retrieves the randomly generated SVG QR code image and renders it. It has a timeout (set to 10 seconds) after which it redirects to the reader page.

**2**: The reader page which accesses a live feed from the web cam using native browser API and renders it to a canvas, it scans for QR codes in the webcam’s live feed and converts them to a string. If this string is verified it then re-directs to authenticated URL ( in this case [http://google.com](http://google.com/) ).

**SDK Sample Mobile Client:**

The sample mobile client uses the devices camera to scan for QR codes using the Smerk library (<https://github.com/MattFoley/Smerk>) and converts them to a string, it then pushes to new view which renders this QR string with some arbitrary ‘biometric’ data as a new QR code.

***NEXT STEPS:***

* Create an set up and run script for the node microservice with a ‘package.json’ file and .bat / .sh executable.
* Convert the web client into a single page application bringing the two components together.
* Build a sample client for Android using existing open source QR reader and render libraries.
* Integrate SensiPass SDK into mobile client and implement between render and reader views.
* Append actual SensiPass biometric data to new QR code on render.
* Implement basic authentication of biometric data on web client.

***NOTES:***

* Web client requires Firefox.
* Web client & Microservice can be run on Window, Linux & Mac
* Mobile client requires OS X 10.10+ to be built and deployed to iOS device.
* **Important** Code was written in a hurry and is hacked together (i.e not clean code) however, with the exception of the changes specified above, the basic architecture is solid.