**User Guide for Food-Chain Scanner**

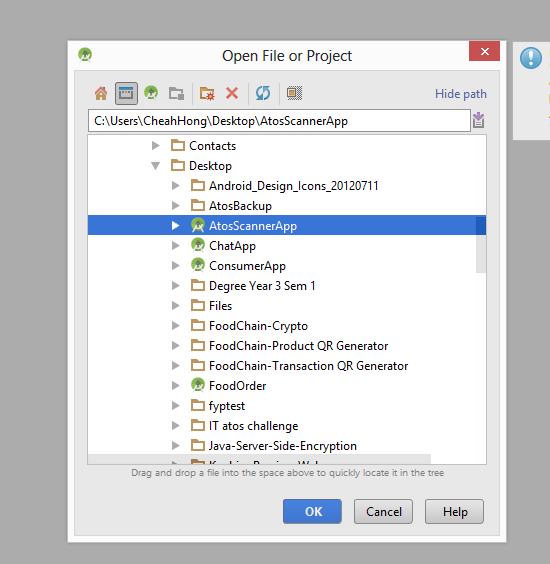
This program is meant to be used locally (WIFI) at every location of the food supply chain (e.g. manufacturer, distributor, market etc) to send location information to Blockchain.

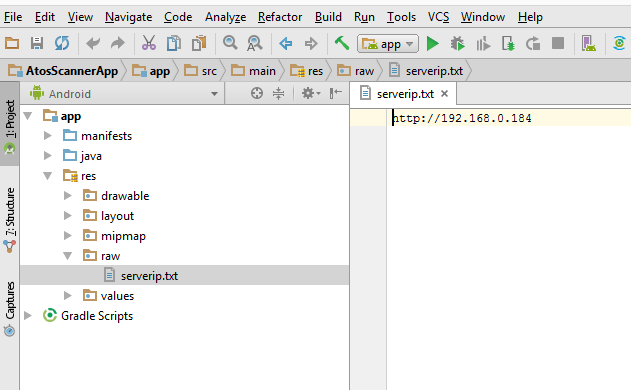
**Requirements:**

* Product QR code and Transaction QR code must present (refer to generate & generate2 user guides)
* Food-Chain Crypto program must be running locally (refer to Food-Chain Crypto user guide)
* XAMPP Apache & SQL must be running (refer to generate2 user guide)
* WIFI connection must be stable(3-4 bar in strength)
* WIFI connection must allow access to <http://174.140.168.136:6876/> (Nxt wallet peer)

**How to build the app:**

1. Install/Update latest Android Studio IDE
2. Open Android Project (AtosScannerApp)



1. Change the **serverip.txt** content in res/raw to your WIFI IP address. 

E.g. **http://192.168.0.184** to **http://192.168.3.144**

1. Run the project

**Devices tested on:**

* Samsung Galaxy S4 (Android Version 5.0.1)
* Samsung Galaxy A5 2016 (Android Version 6.0.1)

**External API:**

* NXT API

**Open source library:**

* Volley(Asynchronous HTTP request)
* Zxing (QR scanner)

**Use Cases:**

1. A worker has sixty apples ready to be packaged as batch
2. The sixty apples are divided into packages of six
3. The worker generates Product QRs using ***generate*** web application (FoodChain-Product QR Generator)
4. The worker generates Transaction QR using ***generate2*** web application (FoodChain-Transaction QR Generator)

* The Transaction QR is available for around 10 transactions because of limited test coins
* In the web application, the Transaction QR is programmed for 100 transactions for 100 batches assuming that the coins are sufficient

1. The worker sticks Product QRs onto the 10 packages of apples
2. For each of the packages, the worker first scans the Product QR, then scans the Transaction QR and lastly send the information to Blockchain using Food-Chain Scanner App
3. The worker put the 10 packages into a box
4. The worker sticks the Product QR onto the box
5. The worker then repeat step 6 for the box QR