

1 Introduction

The team is going to develop a technology solution to be able to track the locations that a certain product goes through on its way from where it is produced, to where it is sold on the shelf. To be able to guarantee that the information is real and cannot be tampered with, a blockchain is used to store all the records, changes and information. Blockchain can store any type of information and transaction, and provides the advantage that it is secure, it is always growing and it is a better alternative than using a normal server. A blockchain is especially important and useful for the development of this project as it is immutable; the contents that are stored cannot be changed. This means that the integrity of information stored is preserved.

The system is going to track the logistics records of a desired product. It will be composed of 3 main applications; the producer application, the consumer application, and the web server which stores access credentials to the blockchain platform as well as generating for QR codes. The producer application is going to be used to create the records and store them in the blockchain during each of the stages of the logistic process. The consumer application, will be the one used by the potential final user or customer. The user will scan the QR code in the product and it will retrieve the records of that product and display them to the user. Finally the web server is going to be used for back end operations and procedures, as well as the creation of the QR codes for the products.

1.1 Purpose

The motivation for this project comes from the need to have a better way to check the locations that a certain product has been to before making it to the end user. This project right now is only a proof of concept, to demonstrate that the project is feasible and has practical potential for future applications.

Initially, blockchains were used in bitcoin (Nakamoto, 2008) digital currency to record transactions for the exchange of coins. In recent years, the potential use of blockchain is wider and can be applied to a wider field. This project is specifically looking at the usage of a blockchain in a logistical setting - the tracking of goods and products. It is hoped that this project will contribute and advance the research to the current body of knowledge in logistics tracking.

The project has special focus on the proof of location and unique identification elements of the elements in the logistic chain. The usage of blockchain will ensure that the information uploaded to the servers won't be changed in the future, but there is also a need to secure that information is not tampered with before or during the upload process. Having reliable location and identification information about the product being handled becomes a key part of the system, and therefore a very important part of the research.

1.2 Scope

Producer application

The producer application will be used by the organisation to scan the QR code that will be generated by a web application. After the code is scanned the location of where the scan was made will be verified by a server. Once the location has been verified the transaction will be stored into the

blockchain. This application will not be storing any records or data as it will all be stored on a server.

This will reduce the amount of storage needed by the application.

- Retrieve QR code from web application
- Scan QR code
- Send validate request to location server
- Sends transaction to blockchain
- Does not store any records
- Does not retrieve records
- Only record transaction and sends it to the blockchain

Consumer application

The consumer application will be used by the product consumer to identify the location of where the product originated from. The QR code will be scanned and then the application will retrieve data of the product from the blockchain. The consumer application will not record the items that were already scanned. A QR code provides a fast response therefore the data will be retrieved within seconds.

- Scan QR code
- Retrieve data from blockchain and display
- Will not make any records to the blockchain

Location server

The location server will contain information of the locations. The role of the location server is to validate the location of where the producer application has done the scanning.

- Validates the location of where scan was made by producer application
- Does not generate QR code
- Does not store the blockchain

QR code generator

Generates QR code that would be assigned to the product batch and saved to the blockchain. The QR code is automatically generated which removes the need to manually produce it.

- Generates QR code for product batch
- Does not save the QR code to the blockchain

2. Overall Description

The logistics tracking system is being redeveloped, using a previous team's code as a guide. The previous team's code provided our developers a better understanding of how the system should work, what the requirements are and how to start developing new, cleaner and more efficient code for the new applications. The system will be developed as a proof of concept, so under no circumstances should it be released to the public, at least not in the stage it will be delivered to the supervisor and current client. The current work is an extension of previous work as it includes secure encrypted communication through the exchange of keys.

2.1. Product Features

This section will describe the major features found on the software that a user can perform. For the web application, a QR code is generated based on the Nxt account number and batch ID, ready to be printed and attached to a product for scanning. The web server also manages secret phrases associated with the account numbers.

The producer application makes use of the scanning function to scan the QR code provided and inserts information into the block.

With the consumer application, the end users will be able to scan the QR code found on their product. After the code is scanned, the consumer application will query the blockchain linked by the code. The final result for the user will be the presentation of details related to that specific product, such as time/date and location of when it was packed or manufactured.

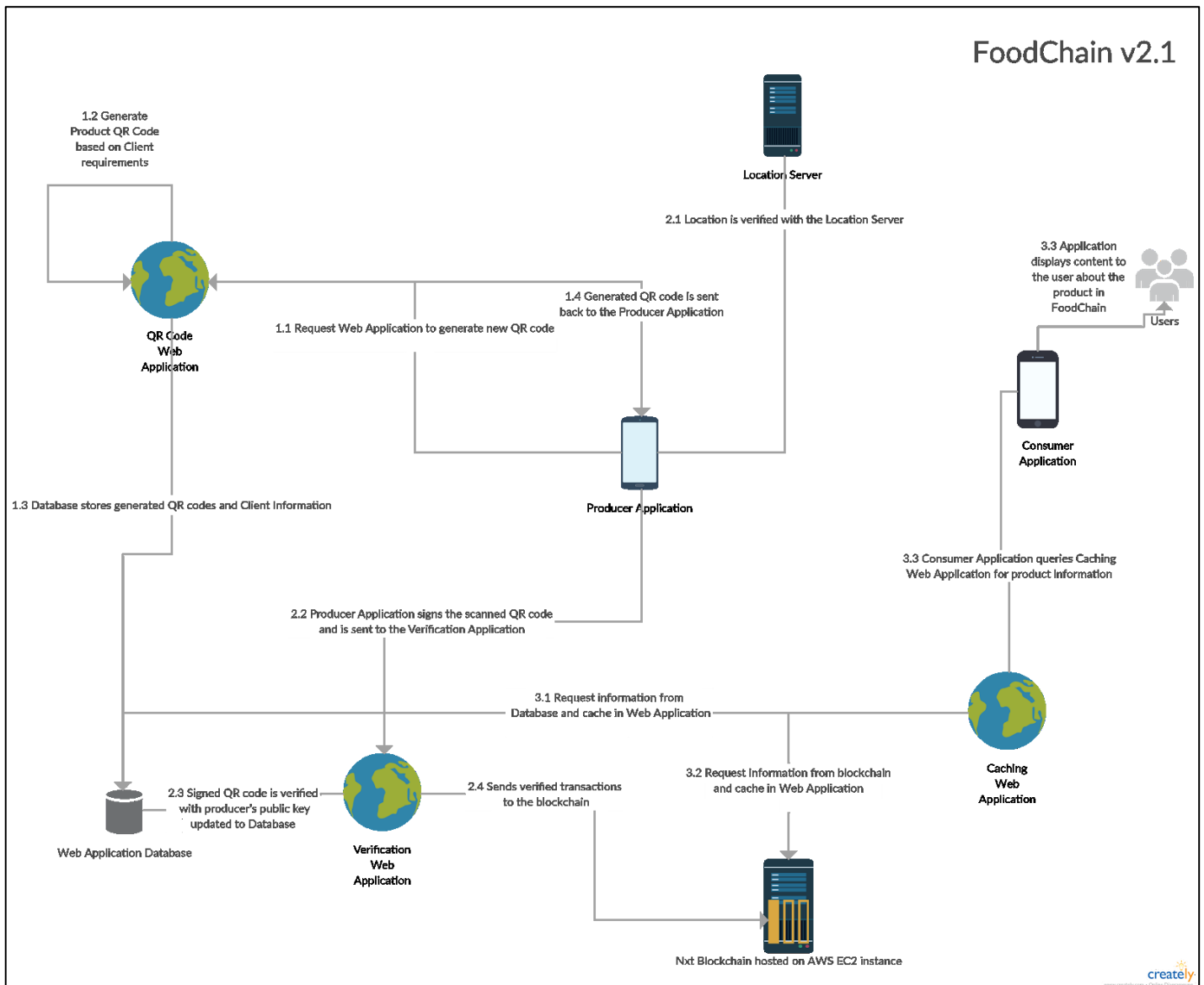
The web server will provide functionality to manage the system and the blockchain information. It will also provide information about the system, such as the status or number of blocks.

2.2. System Requirements

These include the minimum software and hardware requirements to run each component.

- Producer Application
 - Android version 5.0 or higher
 - At least 100 MB free space
 - 3.2 megapixel camera
 - Internet connectivity
- Consumer Application
 - Android version 5.0 or higher
 - At least 100 MB free space
 - 3.2 megapixel camera
 - Internet connectivity
- Web Server
 - Nxt Blockchain node
 - Amazon Web Services EC2 t2.micro instance running Windows Server 2012 R2 (For Nxt node)
 - Amazon Web Services EC2 t2.micro instance running Windows Server 2012 R2 (For QR code generating web application)

3 Architecture Diagram



The system architecture defines the behaviour and structure of the system. It provides a high-level visual depiction of how each component are related to each other, as well as the interaction between the separate individual devices.

This image above will describe the overall architecture on how the each application will communicate with each other.

System Architecture activities:

- Activity 1.X

Activity 1 shows the Producer Application requesting a new QR code for a product. The request is sent to the QR code generating Web Application where parameters are added to the database and a QR code is generated. This QR code is then sent back to the Producer where it can be printed out and stuck on products.

- Activity 2.X

Activity 2 shows the process when the Producer Application scanning a product QR Code. Upon scanning, a request will be made to the location server, which may send back a encrypted string containing information about its longitude and latitude. The Producer Application will forward this with its scanned QR code data to the Verification Web Application. This Web Application will verify the location and parameters with the Database and Blockchain. If it is legitimate they will be added and a success response sent back to the Producer Application.

- Activity 3.X

Activity 3 shows the Consumer Application scanning a product at the end of its journey. It will send a request to the Caching Web Application. This Web Application caches information from the blockchain and database to reduce stress on these key components. The Caching server will respond back with the product's history and locations it has travelled through.

Activity 1 - QR Code Generation

On the Producer's app, they select to request a new QR code, scan their QR code (containing their public/private keypair) and enter in the product details. This information is then sent to the "QR Code Web Application" (activity 1.1).

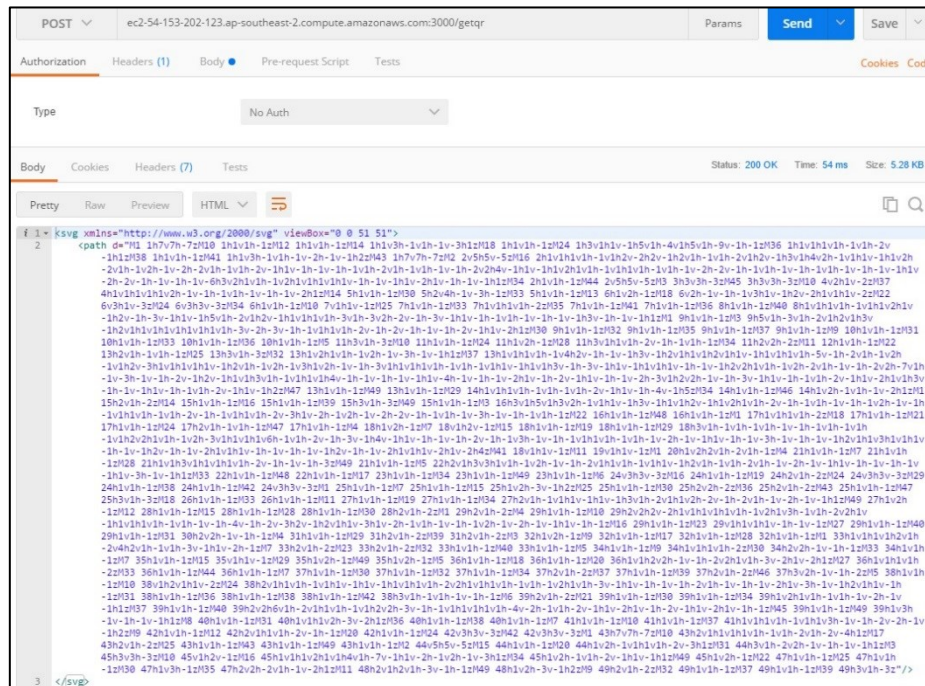
On the QR Code Web Application, a new QR code is generated (activity 1.2). The below image is an example public/private keypair generated by the QR Code Web Application which represents a QR code. The server generates the private key ('Priv Key') and sends it to the Nxt node to get the 'Address' and 'Pub Key' corresponding to that private key

```
Requesting QR Code...
Received QR Code request
MainServer - -----
MainServer - Address: NXT-NFN5-FJK5-KLYZ-EEX4J
MainServer - Pub Key: 44d2c3eb1b2c9aae9c5acfa76416644d805bffaec8e9240f2661160b8e68aa6e
MainServer - Priv Key: 4r5TGikXBjQjzOMK
MainServer - -----
```

The product information and the QR code information is updated into the database (activity 1.3). Below is some sample data of QR code information placed in the database.

```
{
  "_id":0,
  "qrAddress": "NXT-KHJC-NMPY-NS87-ADAJL",
  "qrPubKey": "c31b9ff50f5c478620419bd162acc944eae16cc15c2f1c6f8aab67e24b9bdf14",
  "qrPrivKey": "BdrYq5hShs0FyQmJ",
  "producerAddr": "NXT-QBU9-KSX6-6TH4-H47LR",
  "productName": "a",
  "productId": "b",
  "batchId": "c",
  "producerName": "John Egg Farm",
  "producerLocation": "Croydon Hills, Victoria",
  "timestamp": 1507106442598
}
```

This public/private keypair is then converted to a QR code in SVG format and sent back to the producer (activity 1.4). Below is an example of the SVG response sent back to the Producer.



On the Producer's side, the SVG response is displayed, and the Producer can print it off and stick it on the product.



This QR code request is cached by our caching server (which retrieves QR code updates from the blockchain and provides an API for the consumer app to retrieve information)

You can see a response here from the consumer side.

_id: 0 is the first request made by a producer (producerName, producerAddr)

_id: 1 is the validate transaction sent by the ProductChain server to validate the QR code (crossed out lines were incorrect data, but would be information related to the ProductChain server.

_id: 2 has a MOVE action where the producer who requested the QR code is sending it to another producer

You can see the 'actionAddress' is the Nxt address which generated the transaction. Address ending in AEPHE is our ProductChain server (generating the VALIDATE action). Address ending in H47LR is the first producer address (the one which first requested the QR code and made the first MOVE action). Address ending in AKH4U is the producer that is being sent the product. This is what the databases are holding.

Formatted JSON Data

```
{
  "_id": 0,
  "qrAddress": "NXT-FAJX-9ZCV-RF3R-E69PZ",
  "qrPubKey": "f2e1f57b5842647f57e5feb76abaf2d4e23a65c95f9fd7a623785dc3494dcd3f",
  "qrPrivKey": "lpUyKSpXbNp0KG9V",
  "producerAddr": "NXT-QBU9-KSX6-6TH4-H47LR",
  "productName": "Farm Eggs",
  "productId": "000865",
  "batchId": "0052",
  "producerName": "John Egg Farm",
  "producerLocation": "Croydon Hills, Victoria",
  "timestamp": 1506851313137
},
{
  "_id": 1,
  "action": "VALIDATE",
  "actionAddress": "NXT-HP3G-T95S-6W2D-AEPHE",
  "timestamp": 1506851496,
  "nextProducer": "NXT-QBU9-KSX6-6TH4-H47LR",
  "producerName": "John Egg Farm",
  "producerLocation": "Croydon Hills, Victoria"
},
{
  "_id": 2,
  "action": "MOVE",
  "actionAddress": "NXT-QBU9-KSX6-6TH4-H47LR",
  "timestamp": 1506851643,
  "nextProducer": "NXT-MNDK-R2CB-TX4W-AKH4U",
  "producerName": "Aidan Grocery Store",
  "producerLocation": "Gold Coast Shops, Queensland"
}
```


Account **NXT-FAJX-9ZCV-RF3R-E69PZ** Info

Account has a balance of 0 [Switch Account](#)

Transactions

Ledger

Assets



Trade History

Currencies

Marketplace

Aliases

Actions

Date	Type	Amount	Fee	Account
10/1/2017 9:50:59	 Arbitrary Message	0	1	NXT-QBU9-KSX6-6TH4-H47LR
10/1/2017 9:48:32	 Arbitrary Message	0	2	You

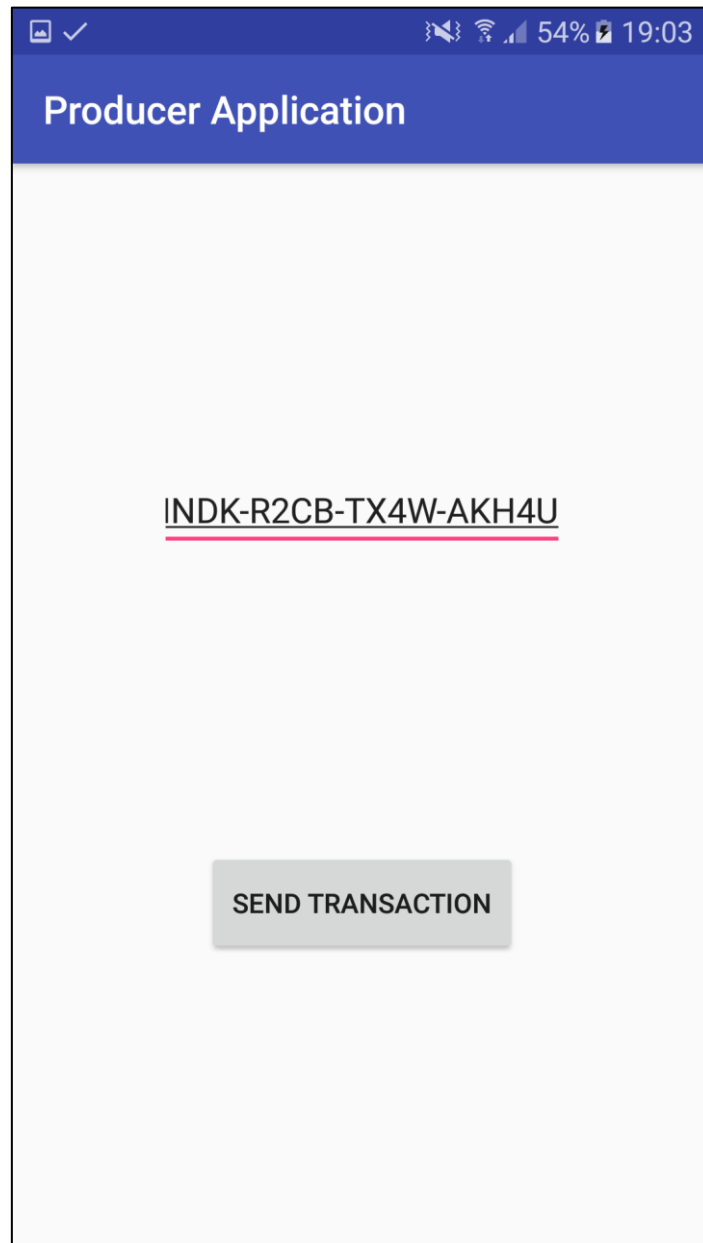
Close

Here you can see what is being stored in the blockchain for the same QR code which was generated. This address (ending in E69PZ) is the QR code generated. You can see the first transaction sent to it was from the ProductChain server "You" at 9:48

Activity 2 - Send Product

When the Producer wants to send a product, they must first scan the Producer's QR code, and then the QR code of the product they are sending to a different location. They must also receive the location verification from the Location Server (Bluetooth Beacon) in activity 2.1.

Before this information is sent to the Verification Web Application (activity 2.2), the Producer must also enter the Nxt address of the Producer who they are sending the product to (see below image).



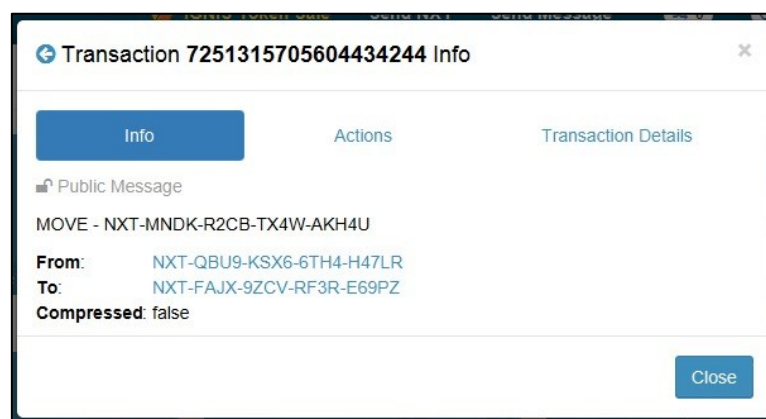
This information about the product being sent is then verified to ensure two things:

1. That the Producer who is attempting to send the product is the same Producer who requested the QR code (for the first send at least. Subsequent send actions would ensure that the Producer sending the product is the same one who received it)
2. That the location proof attached to the send message is consistent with where the Producer is supposed to be located.

Once this has been verified, the information is updated in the database (activity 2.3). Below is some sample data for a product being sent from one location to another.

```
"_id": 2,  
"action": "MOVE",  
"actionAddress": "NXT-QBU9-KSX6-6TH4-H47LR",  
"timestamp": 1507106711000,  
"nextProducer": "NXT-MNDK-R2CB-TX4W-AKH4U",  
"producerName": "Aidan Grocery Store",  
"producerLocation": "Gold Coast Shops, Queensland"
```

This information is then sent to the blockchain (activity 2.4). Below is a screenshot of blockchain transaction information for the same product send action as displayed above. The 'From' field is the Producer sending the product, and the 'To' field is the product's QR code. In the 'Public Message' field, the first item is the action item, in this case 'Move'. The second item is the address of the Producer who will be receiving the product.



Activity 3 – View Information

In activity 3.1 and 3.2, the Caching Web Application retrieves information from the database and blockchain and caches it. Below is an example of the data cached by this web application.

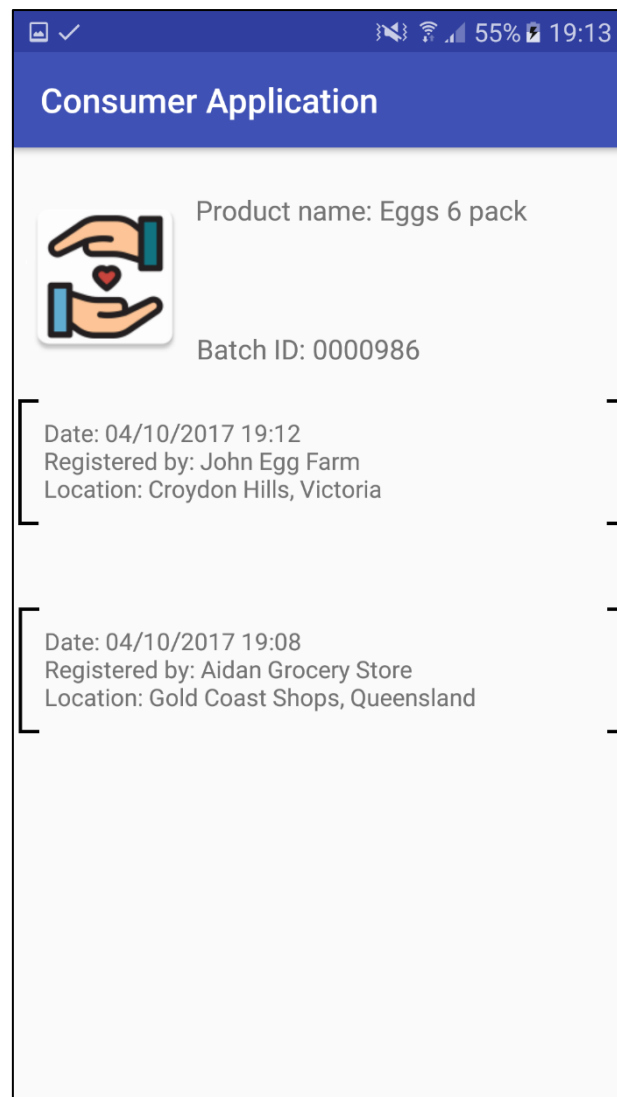
_id: 0 stores the information about the QR code request, including the QR code information that was generated by the QR Code Web Application, and the product information entered by the Producer.

_id: 1 has the information related to the VALIDATE message sent by the ProductChain server to the QR code to validate the QR code. This was not explained here to keep simplicity, but can be explained in more detail later.

_id: 2 contains the information related to the product being sent from one producer to another.

```
{
  "_id": 0,
  "qrAddress": "NXT-ZBB9-5YBY-ZGV8-AGMJK",
  "qrPubKey": "b43dd92ad1f8f35d33ba558fad9f861426bbd6e98029f360efdc0eb1f14a006c",
  "qrPrivKey": "ZoDsiYmVP0l8jii5",
  "producerAddr": "NXT-QBU9-KSX6-6TH4-H47LR",
  "productName": "Eggs 6 pack",
  "productId": "0000854",
  "batchId": "0000986",
  "producerName": "John Egg Farm",
  "producerLocation": "Croydon Hills, Victoria",
  "timestamp": 1507104000523
},
{
  "_id": 1,
  "action": "VALIDATE",
  "actionAddress": "NXT-HP3G-T95S-6W2D-AEPHE",
  "timestamp": 1507104185000,
  "nextProducer": "NXT-QBU9-KSX6-6TH4-H47LR",
  "producerName": "John Egg Farm",
  "producerLocation": "Croydon Hills, Victoria"
},
{
  "_id": 2,
  "action": "MOVE",
  "actionAddress": "NXT-QBU9-KSX6-6TH4-H47LR",
  "timestamp": 1507104515000,
  "nextProducer": "NXT-MNDK-R2CB-TX4W-AKH4U",
  "producerName": "Aidan Grocery Store",
  "producerLocation": "Gold Coast Shops, Queensland"
}
```

On the Consumer application, the consumer will be able to scan a product's QR code, and retrieve information about the product from the Caching Web Application (activity 3.4).



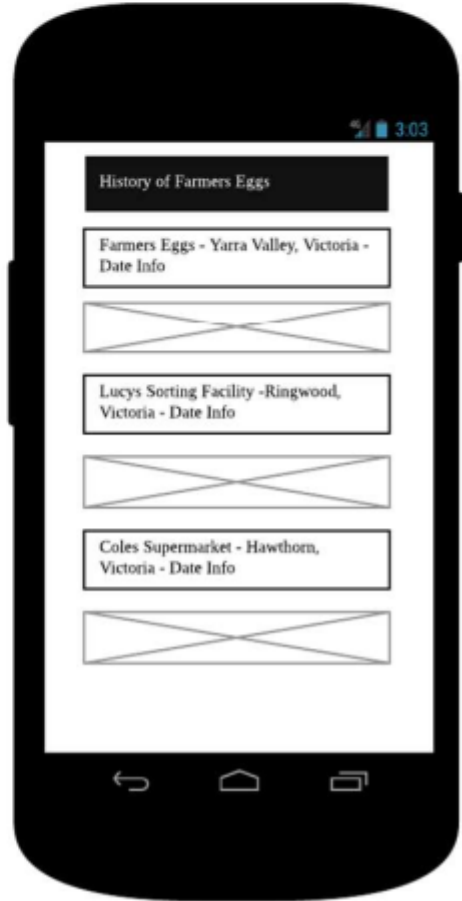
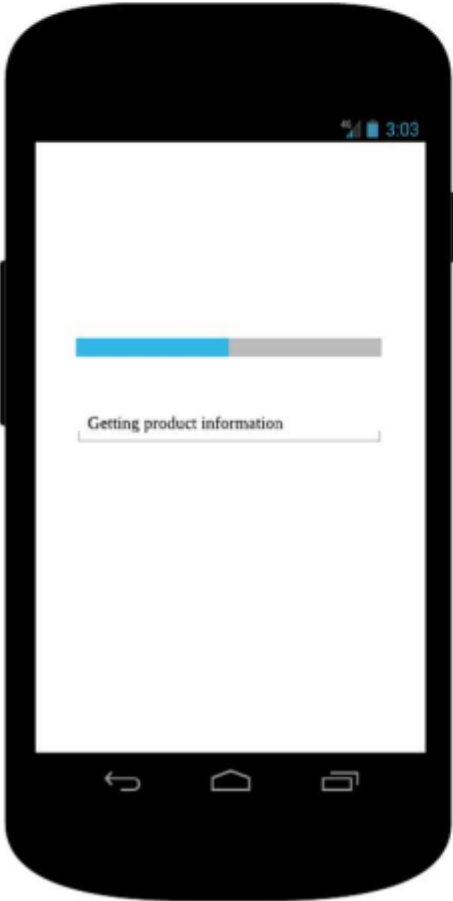
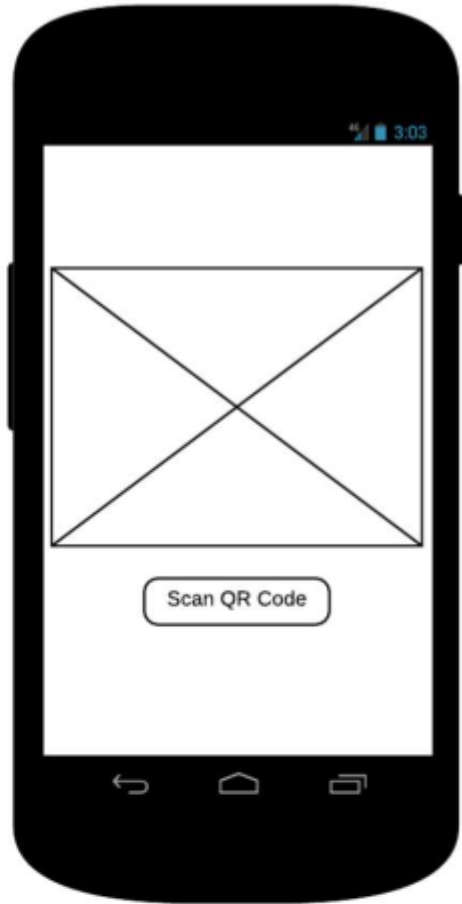
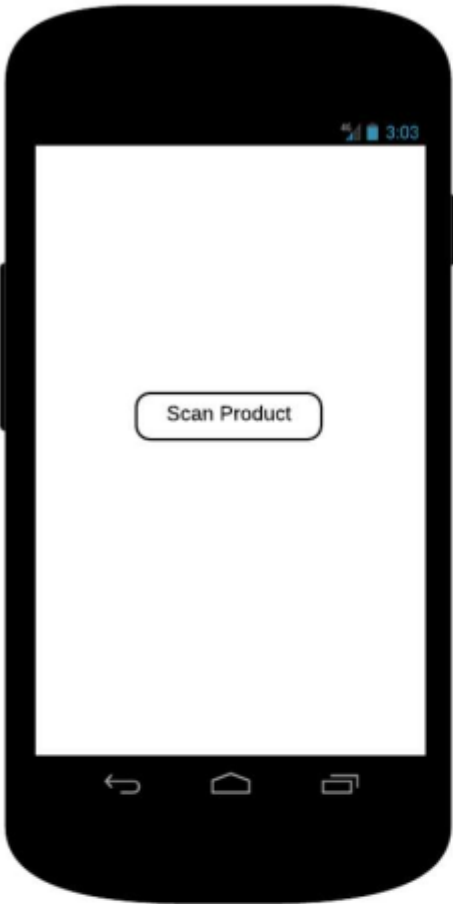
4. Interface Requirements

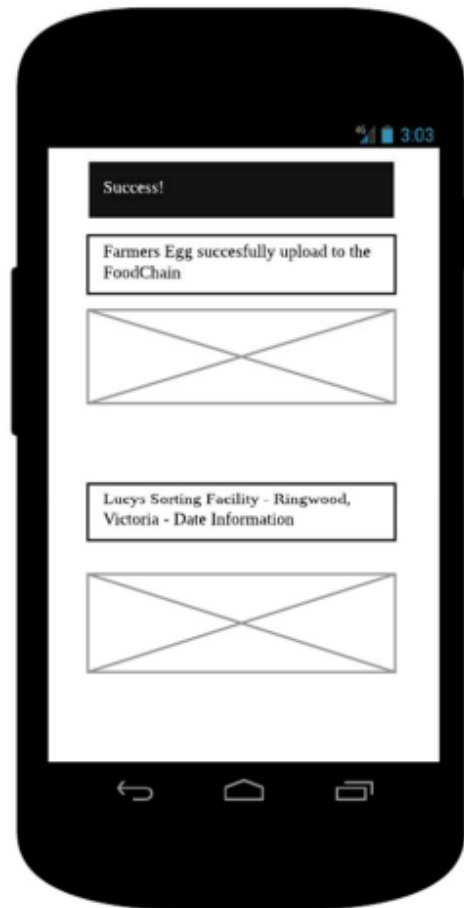
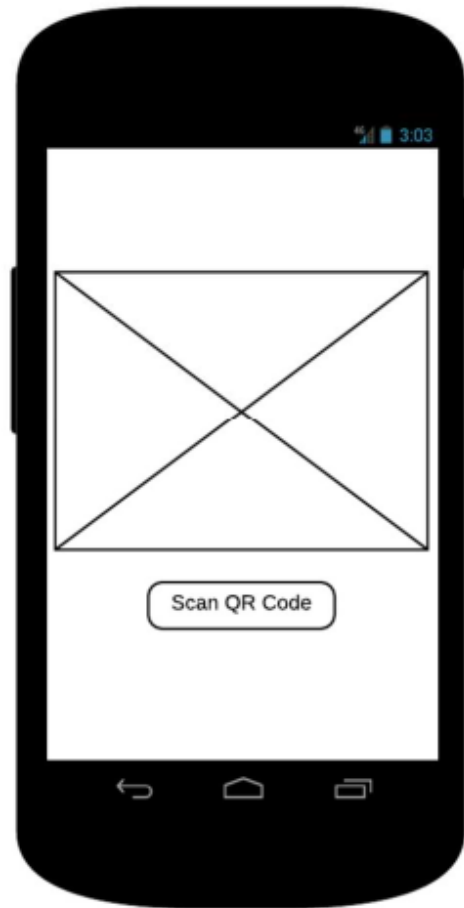
The following section will explain how the different interfaces will communicate with each other. Each category explains different levels of interactions. The categories are broken down into user interaction, hardware, software and communication.

4.1. User Interfaces

Both the consumer and producer mobile applications will function by scanning QR codes to create or retrieve information about a product in hand. The following mockup diagram shows how the consumer application will scan the QR code and will display the information of the logistic path that the product has gone through.

The producer application will work in a similar way. It will also start by scanning the QR code of the product, but after that, instead of getting the information of the product, it will get the current location of the device and will proceed to store the location and information in the blockchain.





4.2. Hardware Interfaces

Both the web application and mobile application do not have any designated hardware. Therefore, it does not have any direct hardware interfaces. The management of physical location is undetermined at present as further research is required for its implementation. There may be additional hardware involved in a potential solution for the proof-of-location solution. The hardware connection from a device to server is managed by the underlying operating systems on each respective device.

A mobile phone running the Android 5.0 mobile operating system or newer is required. As the solution will not be heavily resource dependent, most Android phones will be able to run it without any issues.

In terms of hardware usage, QR codes that are generated by the web application are printed out and will be attached to a product. The QR codes are then scanned by a mobile device, on either producer or consumer application.

4.3. Software Interfaces

This section specifies and describes connection between this system and the use of other required products and interfaces. These include operating systems, tools, libraries or borrowed APIs that our system has to use. Each required software are named (refer to section 1.3 for abbreviations) and its respective specification or version number is listed. The purpose of how the interfacing software is related to the current system is discussed, as well as the definition of the interface in terms of message content and format.

- The Producer and Consumer Applications are designed to run on Android 5.0+
- The Web Server and Nxt blockchain node will be running on separate servers both running Windows Server 2012 R2.
- The Database, containing Nxt account and batch information, will be running database software in the cloud.

The QR code generating web application will retrieve information from the Database to construct the QR code.

The Producer application will then scan the QR code with the camera on the mobile device and retrieve information about the product to be added into the blockchain.

The Producer application will then communicate with the location based server/device to confirm its location is genuine. If it returns true the app will send it to the blockchain node otherwise it will be discarded and an error reported to the user.

In the event that the Producer application's location is verified, it will send the information about the new product to the blockchain node which will create a transaction linking it to the information

on the product's QR code. This transaction will include identifying information for the product, as well as the proof-of-location.

The Consumer application then queries the Database and blockchain to get information about the product and the different stages that it has gone through.

4.4. Communication Interfaces

The communication between each individual parts of the system is important since they depend on each other. The applications described in this document uses several different forms of communication. Communication made through another software application is not in the scope of this section. There will be secure communication between devices using protocols that use encryption.

As most of the communications made are Internet based communications, the mobile producer application will communicate with the web application through HTTPS. When recording and updating the details and location of a product, a HTTP POST request is sent to the blockchain server. Similarly, when an end-user scans a QR code, a HTTP GET request will be sent and the corresponding block or record is retrieved from blockchain server. The user then sees information relating to the block linked by the QR code.

The communications that occur in the system include communications between:

- Web Application Server and the database
The web server (QR code generator) queries the database to get the information to include in and generate the QR code.
- Producer Application and AWS
Producer app writes product information into Nxt blockchain which is hosted on the cloud instance
- Consumer Application and AWS
Consumer app sends request to blockchain node and the product information is returned
- Location Server and Producer Application

The interaction between these two requires further research. The general idea is that the producer application will communicate with the server, potentially in the form of a location check and then the server verifies the sent location