University of Mumbai

Supply chain management for APMC(Agricultural Produce Market Committee) using Blockchain

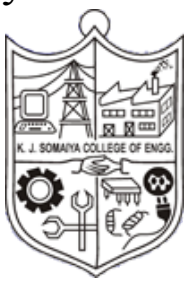
Submitted at the end of semester VII in partial fulfillment of requirements
For the degree of

Bachelors in Technology

by

Ankita Shelke 1811103
Jainam Shah 1811085
Smit Bhatt 1811070

Internal Guide :
Prof. Swapnil Pawar
External Guide :
Dr. Shyam Masakpalli



Department of Computer Engineering

K. J. Somaiya College of Engineering, Mumbai-77

(Autonomous College Affiliated to University of Mumbai)

Batch 2018 -2022

We declare that the proposed work is based on our and/or others' ideas which will be adequately cited and referenced in the reports. We also declare that we will adhere to all principles of intellectual property, academic honesty

Roll No.	**Names of the students**	**Branch**	**Email Id and Mobile no.**	**Signature of the Student**
1811070	Smit Bhatt	Comps	smit.bb@somaiya.edu +91 8108088602	
1811085	Jainam Shah	Comps	jainam02@somaiya.edu +91 9821121884	
1811103	Ankita Shelke	Comps	shelke.aa@somaiya.edu +91 9820361838	

.

Guide

Prof. Swapnil Pawar

Expert

Prof. Gopal Sonune

Contents

Abstract

The lack of openness and trustworthy reporting is the primary cause of traditional supply chain inefficiency. Many businesses suffer from a lack of insight over their whole product supply chain, and as a result, they lose an immediate competitive advantage over their industry's competitors.

The information about an entity is not fully available to others in traditional supply chain models, resulting in erroneous reports and a lack of interoperability. Emails and printed documents provide some information, but because products across the entire supply chain are difficult to trace, they can't provide fully detailed visibility and traceability information. It is nearly impossible for a consumer to determine the genuine worth of a thing they have purchased.

The blockchain is a game-changing solution for traditional supply chain sectors since it is a transparent, immutable, and secure decentralized system. It can assist in the development of a successful supply chain system by addressing the following areas:

Tracking products across the supply chain, verifying and certifying the chain's products

Information about the whole supply chain is shared among supply chain actors. Increasing auditability. The food industry's supply chain is a complex landscape to navigate, as various actors must work together to bring items to their end destination, which is the customers.

Every step of the process presents possible security flaws, integration problems, and other inefficiencies. In today's food supply networks, the biggest emerging issue is counterfeit food and food fraud.

The goal of this project is to create a peer-to-peer decentralized application (DApp)

using the Hyperledger Fabric blockchain, which will allow for full visibility, tracking, and traceability.It would ensure the legitimacy of food by permanently recording the details of a product. We will enable the end user to self-verify a product's legitimacy by exchanging product facts over an immutable infrastructure.

Keywords : *DApp, Blockchain,Hyperledger fabric,Supply-chain*

Nomenclature

Hlf	Hyperledger Fabric
DApp	Decentralized application
P2P	Peer-to-Peer
POC	Proof-of-Concept
API	Application programming Interface
UI	User Interface
IDE	Integrated Development Environment
PoW	Proof of Work
SDK	Software Development Kit

List of Figures

Chapter 1

Introduction

1.1 Background

Huge corporations (technology companies) control the world today thanks to centralized systems Data from billions of users. The information can be viewed and used by businesses. Each transaction is not secure, since it‘s in the hands of a few select companies. The common man who owns the data, is not given credit for/paid for contributing his/her data. This scenario should be changed. Eradication of third-party agents is necessary to allow users to interact and make transactions with one another directly on a peer-to-peer network. The first advantage of doing so, is that users will no longer have to depend on third parties (banks, governments) to proceed with transactions). The second advantage is that users will not have to pay to these third parties for carrying out these transactions. The biggest advantage is that each transaction will be secure and unmodifiable.

This project aims at creating a peer-to-peer decentralized application (DApp) to build a cross platform based mobile application where farmers can enter new produce at the start of the supply chain generating a unique token. In the supply chain, this token can be tracked right up to the consumer who verifies the source of the produce. At each intermediate point, smart contracts would be invoked, allowing the blockchain network to be continuously updated, ensuring the highest level of transparency and traceability.

1.2 Problem Definition

Lack of transparency in the supply chain between the producers and the consumers is the biggest issue faced by the agricultural sector. The wastage of food caused by the delay during the

distribution can be avoided by proper management and traceability.

A blockchain is a continuously growing list of records, called blocks, which are linked and secured using cryptography. Using blockchains, we can easily verify the authenticity of the data in a decentralized manner. It can provide traceability near real-time visibility of farm produce distribution and chain of custody from manufacturing to administration.

This would help in eliminating the blind spots across public and private entities and enhance the trust among the participants.

1.3 Motivation of thesis

To provide visibility into the farm produce distribution chain of custody, from manufacturing to administration, in near real-time. Improve the security, communication, and transparency in the supply chain.

1.4 Scope

The scope of the project is to build an application where farmers can enter new produce at the start of the supply chain generating a unique token.

This token can be tracked in the supply chain till it reaches the consumer who can verify the source of the produce.

At each intermediate point, smart contracts will be invoked and necessary updates would be made to the blockchain network providing transparency and traceability throughout the entire chain.

1.5 Objectives

Objectives for Phase-1 (SEM VII)

1. Understanding the requirements
2. Analyzing the supply chain flow
3. Design a secure unique token system
4. Develop the basic features for the android application
5. Building the Firebase

6. Prototyping and Ground Testing

Objective for Phase-2 (SEM VIII)

1. Choose Blockchain network Type.
2. Setup the network structure and decide the policies.
3. Develop and test the chaincode in Go lang.
4. Create channels and deploy chaincode on the peers in different organizations.
5. Build API on Node and design the functions.
6. Integrate with flutter mobile based applications with the API and blockchain network.

1.6 Hardware and Software Requirements

Hardware Requirements -

- Personal Laptop/Computer having 64-bit processor,1.7 GHz Core
- An android device supporting the latest android version(current version 12.0),supporting camera functions
- Sufficient RAM(8 GB on pc,3Gb on mobile) for application to run

Software Requirements

- Golang
- NPM for Node.JS packages
- Visual studios code
- Firebase(Chrome or Firefox browser)
- Flutter packages for I/O,http requests
- Docker desktop
- Postman for api testing
- Hyperledger Network (fabric binaries)
- Windows 10/11, Linux Operating System (Ubuntu 18.0.4, Ubuntu 20.0.0)

Chapter 2

Literature Survey

2.1 Introduction

For the Literature review, we have gone through several IEEE and white papers published. The topics basically include blockchains and hyperledger networks. In fact, we studied papers that talked about both of them combined. We looked for such specific papers and found ten papers, which in someway address similar issues as the one we are trying to address in our problem definition. We also looked on the internet for the same material and read about the technologies used in papers to solve their problems viz hyperledger and smart contracts(chaincode). In the following section we have written a brief overview of the papers we have studied. These papers proved to be very helpful for our project development and implementation.

2.2 Blockchain

In [1], The block chain is a shared public ledger that records all confirmed transactions in this document.

It is made up of blocks that each include a batch of legitimate transactions. Each block includes the previous block's hash, which connects the two. As a result, the connected blocks create a

chain. The risk of data being maintained centrally is eliminated with blockchain. Decentralization is based on the usage of public and private keys. The user's address is a public key, and the private key, which functions similarly to a password, grants access to their Bitcoin or other digital assets. Data quality is maintained by massive database replication. The blockchain's integrity and chronological sequence are guaranteed by cryptography. Cryptographic technology is used to secure blockchains.

2.3 Decentralized Ledger

In this paper, The block chain is a shared public ledger that records all confirmed transactions in this document.

It is made up of blocks that each include a batch of legitimate transactions. Each block includes the previous block's hash, which connects the two. As a result, the connected blocks create a chain. The risk of data being maintained centrally is eliminated with blockchain. Decentralization is based on the usage of public and private keys. The user's address is a public key, and the private key, which functions similarly to a password, grants access to their Bitcoin or other digital assets. Data quality is maintained by massive database replication. The

blockchain's integrity and chronological sequence are guaranteed by cryptography. Cryptographic technology is used to secure blockchains.

1. Points of Failure / Maintenance: Because there is just one point of failure, centralized systems are simple to maintain. Decentralized systems have more, yet they are still limited. The most challenging to maintain are distributed systems.
2. Fault Tolerance / Stability: Centralized systems can be extremely insecure. In a decentralized system, a single point of failure has no effect on the whole, and there will be many decentralized systems. Distributed systems are extremely stable, and a single failure has little impact.
3. Scalability / Maximum Population: Centralized scalability is limited, Decentralized scalability is moderate, and Distributed scalability is infinite.
4. Ease of development/creation: Centralized systems can be built quickly if you use a framework that you can use anywhere. You must first work out the lower-level aspects such as resource sharing (trade) and communications for Decentralized and Distributed (transport).
5. Due to the fact that centralized systems follow a single framework, they lack diversity and progress slowly. However, once the basic infrastructure is in place for Decentralized and Distributed systems, the potential for evolution is enormous.

2.4 Blockchain platform for COVID-19 vaccine supply management

In [3], we discuss how blockchain technology can be used for assuring the transparent tracing of COVID19 vaccine registration, storage and delivery, and side effects self-reporting. We present

such a system implementation in which blockchain technology is used for assuring data integrity and immutability in case of beneficiary registration for vaccination, eliminating identity thefts and impersonations.

Smart contracts are defined to monitor and track the proper vaccine distribution conditions against the safe handling rules defined by vaccine producers enabling the awareness of all network peers. For vaccine administration, a transparent and tamper-proof side effects self-reporting solution is provided considering person identification and administered vaccine association.

A prototype was implemented using the Ethereum test network, Ropsten, considering the COVID-19 vaccine distribution tracking conditions. The results obtained for each on-chain operation can be checked and validated on the Etherscan, demonstrating various aspects of the proposed system such as immunization actors and safe rules registration, vaccine tracking, and administration. In terms of throughput and scalability, the proposed blockchain system shows promising results.

2.5 Blockchain for organic food supply chain

In [4],The purpose of the research was to implement a Blockchain-based solution to verify the food quality and the origin of the agricultural supply chain. A public Blockchain concept was selected instead of a private Blockchain in this study to ensure transparency by allowing any person to access the network. Instances of the smart contract were created for each physical product and deployed to the Blockchain network.

A Quick Response code which contained the address of the instance, was a reference to the

virtual product. All the actors who are involved in the supply chain must be able to interact with the system to achieve transparency. Each transaction and events related to a product is validated by peers of the Blockchain system.

Product ownership was changed for each relevant transaction. A token-based mechanism was used to indicate the farmers' reputation with their products. Farmers could place a certification request regarding their products and they can gain reputation tokens for each certification done by peers. A unique Quick Response code was used to identify each product within the supply chain. The proposed system has been implemented as a prototype and validated within the study.

2.6 Agricultural Supply Chain Management Using Blockchain Technology

In [5],The author outlines issues with India's current agricultural supply chain, stating that due to a lack of communication between supply chain layers, around 16 percent of produced fruits and vegetables are wasted each year. Smallholders do not benefit from market research conducted by larger firms who sell directly to consumers, resulting in inefficient production and food waste. The producers have a limited understanding of meteorological conditions, markets, and supply and demand. The author proposes using blockchain to solve chain problems by incorporating smart contracts between middlemen for automatic contract execution and to prevent smallholder abuse, as well as incorporating a decentralized database that includes credible information about the chain that is peer-to-peer reviewed and time stamped, allowing producers to have security, traceability for any damages, and production of required amounts of goods, while also increasing efficiency in the supply chain.

2.7 BRUSCHETTA: An IoT Blockchain-Based Framework for Certifying Extra Virgin Olive Oil Supply Chain

In [6],BRUSCHETTA, a blockchain-based application for the traceability and verification of the Extra Virgin Olive Oil (EVOO) supply chain, is presented by Antonio Arena, Alessio Bianchini, and Pericle Perazzo. BRUSCHETTA offers a blockchain-based method for enforcing product certification by tracing the product's full supply chain, from the plantation to the retailer. The purpose is to provide a tamper-proof history of the product, including farming, harvesting, manufacture, packing, conservation, and transportation operations, to the final buyer.

They use the permissioned blockchain network hyperledger fabric (HLF), with four different organizations representing four parties: farmers, oil producers, logistics, and sellers. It includes crucial endorsement policies, such as the fact that a node belonging to one organisation cannot belong to another, and that a transaction between two nodes is permitted only if both parties agree on the quality of the goods transmitted. External dangers such as data deletion, tampering, and theft are mitigated by the ledger's immutability and data encryption.BRUSCHETTA uses Internet of Things (IoT) technologies to connect and run sensors for EVOO quality control on the blockchain. The usage of blockchain allows end consumers to have access to a tamper-proof copy of the product's full history, for example, via their smartphone.

2.8 An Agri-Food Supply Chain Traceability Management System based on Hyperledger Fabric Blockchain

In [7],Angelo Marchese and Orazio Tomarchio provide a detailed concept of a blockchain-based agri-food supply chain traceability system and demonstrate the application of blockchain

technology in this industry by implementing a system prototype. The proposed system, in particular, enables supply chain participants to store and manage product-related traceability data in a distributed and immutable manner.The framework's foundation is a permissioned blockchain, which is built using the Hyperledger Fabric architecture. The core of the system's business logic is executed in the form of a smart contract in this blockchain, which provides many operations that allow users of the system to securely add and edit information in the blockchain. The supply chain members and regulatory departments are the system's users. The former adds and changes information about their products, while the latter deals with supply chain management and regulation. User organizations are the entities involved in system operations, and each user is identified by a certificate issued by a certification authority affiliated with the organization to which the user belongs.Users connect with the blockchain using a client application that runs on an application server, while the latter interacts with users through a frontend application hosted by a web server. Each company has its own web server and application server. The designed framework comprises of a prototype that is deployed within a Kubernetes cluster to simulate the distributed nature of the entire system and to improve portability and interoperability with current organization IT systems.Finally, the system gives a comprehensive picture of the various harvesting, processing, and distribution phases to which product batches are subject, allowing for the reconstruction of each batch's entire life cycle and the retrieval of provenance data.

Chapter 3

Project Design

3.1 System Diagram

APMC(Agricultural Produce Market Committee) is a marketing board established by each state government in India.It's function is to maintain, collect, disseminate and supply information in respect of production,sale storage,processing,pricing and movement of notified agricultural produce. The current APMC system has certain shortcomings like:

- Produce is manipulatively bought at lower price and sold at a much higher price to the market,depriving farmers of their profits.

- There is Data tampering of prices and quality. The lack of traceability of produce often results in wastage of produce due to late and inefficient communication between middlemen.

We have identified the main participants of a general APMC market chain as below:

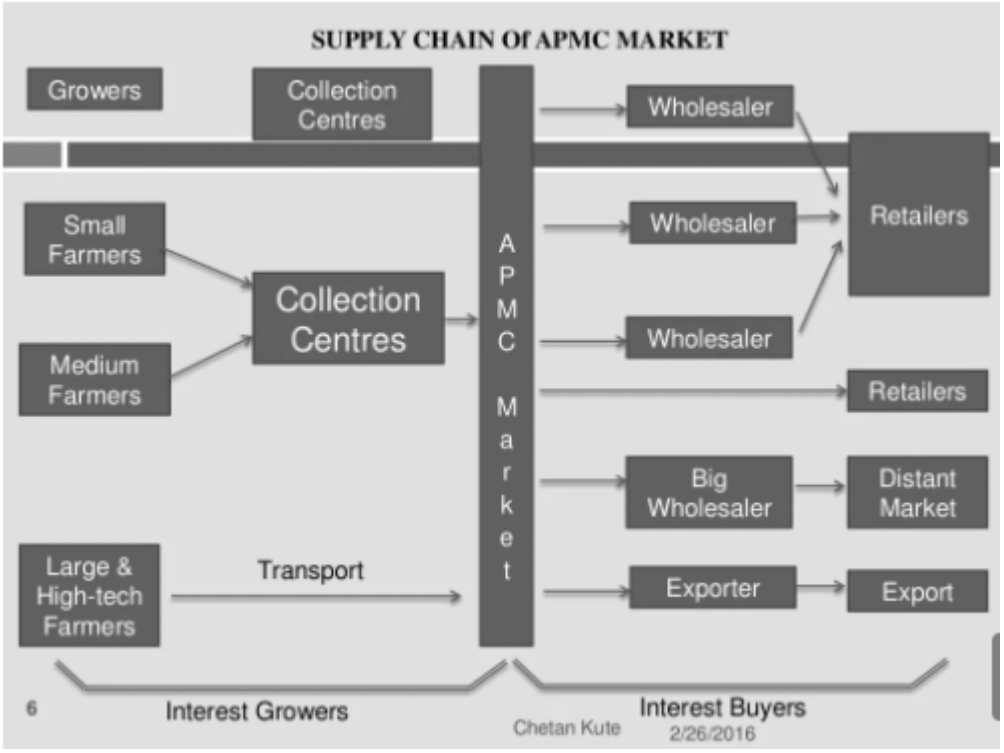


Fig 3.1 Supply Chain structure at Agriculture Produce Market Committee (APMC)

Source: https://www.slideshare.net/ChetanKute/basic-supplychain-apmc-bhaji-market

Here(In fig 3.1) we have identified the main participants as, the farmer or raw producer, collection centers as Manufacturer,Wholesaler,Shipping or transportation, Retailer, and the end consumer. We define the system diagram of our application as:

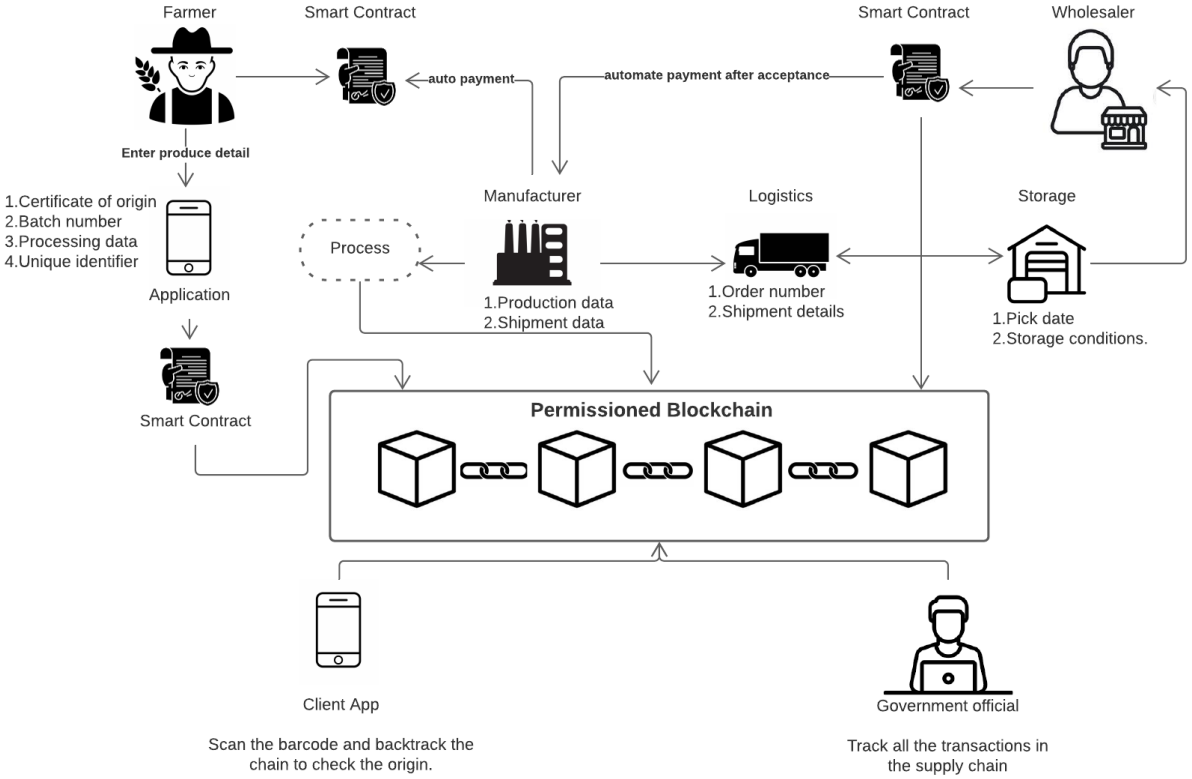


Fig 3.2 Proposed System Diagram

The diagram[3.2] shows the general overview of the flow of products through the supply chain. The starting point is farmers where a new product entry is made and details like product grade, price,quantity are entered and generated into qr code,and the produce is tracked after generating a unique QR code for the same. At each intermediate stage, the chain is updated with necessary smart contract calls updating the entire network.

This application is a distributed application (Dapp) and as the name suggests, it is a distributed peer-to-peer modeled architecture. There is no central server maintaining the blockchain.

The application involves different parties including the farmer, manufacturer, wholesaler, customer, etc, and smart contracts are created between necessary parties to maintain the product

traceability in the chain. The farmer would be able to trace his product till it has been to the end of the chain to the customer.

3.2 Software Project Management Plan

Software Process Model

We will be using the Spiral Model for updating the project and other deliverables. This is done for the correctness and missing areas of the project over time. The two main deliverables apart from module checks and deliveries will be done in two phases.

The system will be divided into 2 phases:

- Phase 1 will focus on Requirement analysis & Application development
- Phase 2 will focus on Building Network & Integration with Chaincode.

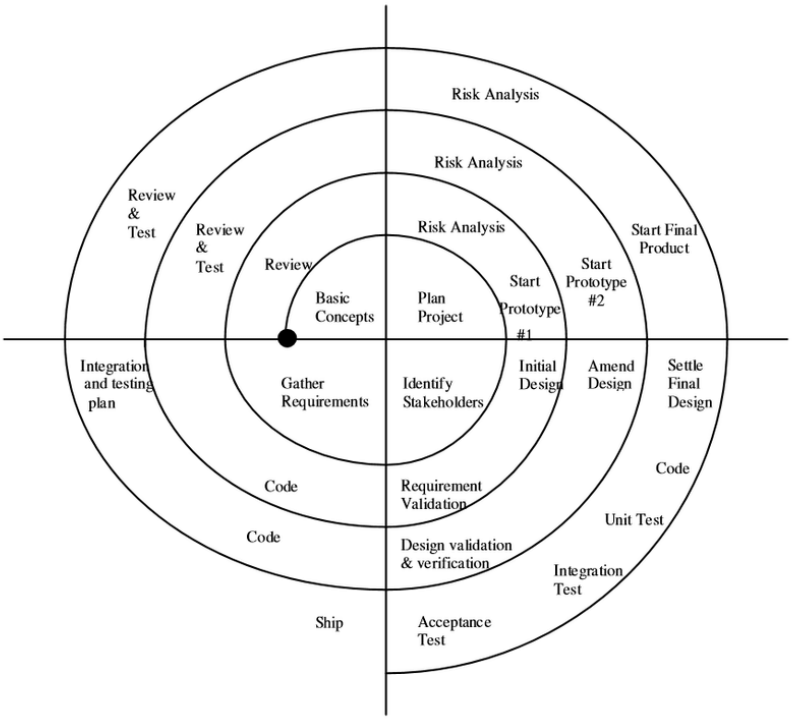


Figure 3.3 Spiral model

Each loop of the spiral model given in figure3.3 consist of requirement gathering,identify and resolving risks,planning project and review and testing the application.

3.3 Roles and Responsibilities

We have adopted a democratic team structure for distributing the work and reporting amongst the team members based on the complexity of the project, scope defined and analyzing the available resources.

Responsibility Matrix

Activity	Ankita	Jainam	Smit	Mentor/Guide
1. Requirement Gathering				
1.1 Interaction with customer	C	C	C	A
1.2 Preparing SRS	C	C	C	A
2. Design				
2.1 Preparing Block diagram	C	C	R	A
2.2 Writing Functional Requirements	C	R	C	A

2.3 Writing Non-Functional Requirements	R	C	C	A
2.4 Developing Use Case	C	C	R	A
2.5 Developing Test Cases	R	C	C	A
3. **Planning**				
3.1 System Design	C	R	R	A
3.2 UML Diagrams	R	C	C	A
4. **Coding**				
4.1 Unit 1	C	R	R	A
4.2 Unit 2	C	R	R	A
4.3 Front end/ UI	R	C	R	A
5. **Testing**				
5.1 Unit 1	E	A	E	
5.2 Unit 2	A	E	E	
5.3 System Testing	E	E	E	A

C: Creator, R: Reviewer, A: Approver E: Executor

3.4 Tools and Technique

This project is being developed using Hyperledger Fabric (A platform for permissioned blockchain prototype).

• Implementation: Programming languages:

1. Flutter, Dart
2. Firebase
3. Golang
4. Javascript

• Tools:

1. Backend: Hyperledger Fabric ,Node js,golang
2. Frontend: Dart, Flutter
3. Collaboration: GitHub, Figma, and Google Drive
4. Text Editor: Android studios, Visual Studios Code

• System requirements:

1. RAM: 4GB (Minimum)
2. Processor: 1.7 GHz Core i3(minimum)
3. Disk Space: Minimum 150 GB (Grows Linearly with the growth of Blockchain)

3.5 System Design UML Diagrams

3.5.1 Use Case Diagram

Actors:1.Consumer

2.Administrator

3.Manufacturer

4.Farmer

5.Wholesaler

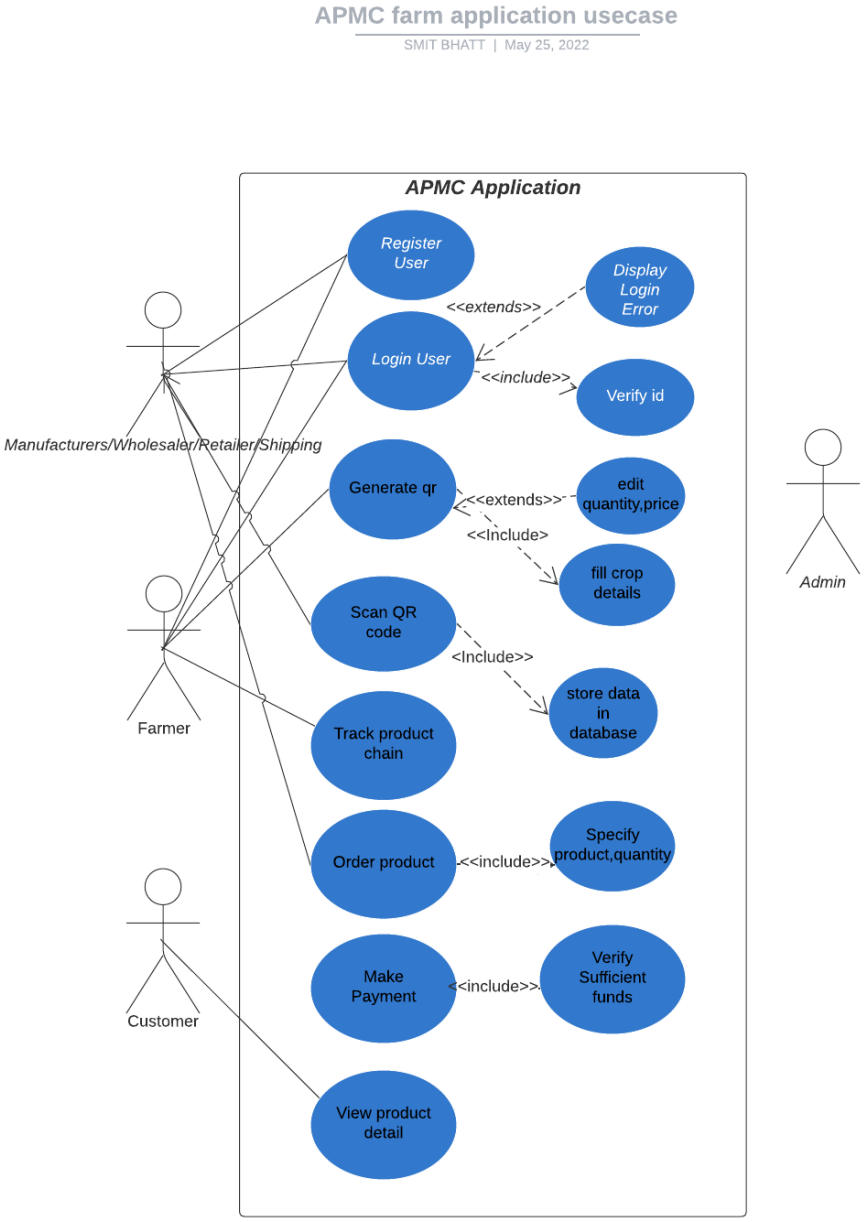


Fig 3.5 Use Case Diagram representing all the Actors and their usecases in the system

3.5.2 Flow Diagram

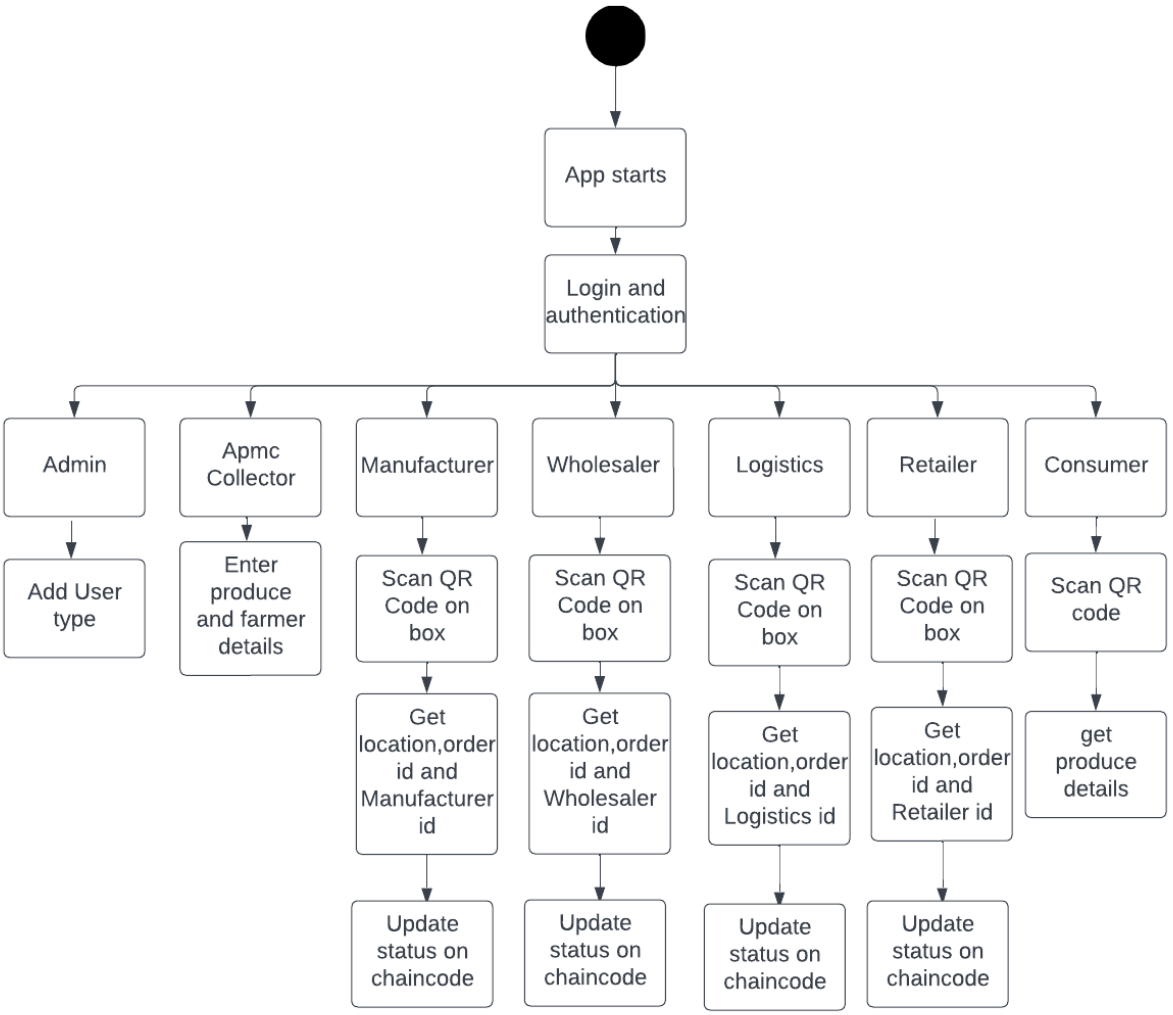


Fig 3.3 Flow Diagram

The Flow diagram represents the flow of functions required according to each type of user in the organization.

The admin can register a user according to their user type(wholesaler,retailer,etc).

The apmc collector/farmer enters produce and farmer details.

The Manufacturer scans QR code on the product box, and location,manufacturer id and date,order id is updated in the ledger.

The process is similar for Wholesaler,Retailer and Shipping, and all status is updated in the chaincode and stored in the Ledger.

finally,The consumer Scans the QR code from the product and gets complete details of the product and information regarding all the stages in the chain.

Chapter 4

Implementation and Experimentation

Risks, Issues and Challenges, and Constraints

Since this technology is quite new and no full-fledged projects or resources relevant to it are in existence, the main challenge was to gather resources and fit them sequentially in our development course.

Commercial Factor: For making transactions through our application, our prototype needs to be deployed in kubernetes cloud for containerizing and also will incur a service charge. This could be a hindrance factor since the technology is new and involves money transactions.

4.1 Functional Requirement

Module 1 - Front End with Flutter

1.1 QR Scanner

This module is focused on generating the QR Code from the given string. Currently based on the input hex string passed as an input corresponding unique code is printed. The version of the QR code is set to auto and the error correction level is chosen as L. Standard color with hex code 0xFF000000 is used to fill the dark areas with a gapless design pattern.

1.2 Add New Farmer

The admin can add a new farmer and enter the personal details like name, phone number and place of the farm. For each new farmer a new ID is generated which can track the produce given by a farmer in the supply chain. After the submit button, the details are stored in the form of String in firebase. When queried, details regarding the farmer can be displayed.

1.3 Generate Unique Token

To make sure each token is unique, we are making use of a hash function from cryptography. Specifically we are using SHA256 hash to generate unique bits in hex that

can be tracked in the chain. When a new box arrives at the source, details such as id + org + time + location + product are taken into account while generating the code.

This gives us two advantages, first as a single bit change in the input can drastically change the hash value, so we ensure that that is tamper proof. Another advantage of using hash functions is collision resistance, preimage resistance, and second preimage resistance.

1.4 Scan and Trace product

This module is used to scan a QR Code captured in the camera and analyze the bits captured. A hex string is encoded in the QR which can be used to find the details on the box and update them in the chain. A flutter Module QR code scanner is used to capture the code from the phone lens.

Module 2 - Back End in Hyperledger

2.1 Design network

The network structure includes a number of organizations, peers and orderers which play a vital role in validating transactions from the client requests. We have proposed a network with 3 organizations each containing two peer nodes. Each organization also has an anchor peer to establish connection amongst the organizations. All the organizations are connected with a single channel with one shared ledger storing all the changes in the network.

2.2 Chaincode Development

Initialize the chanicode and then we can invoke the functions which can alter the world state. Each function takes in some parameters which can alter the structure of food and make necessary changes in the Status on the supply chain. Query function is used to get the current status or stage at which a particular orderId is present at.

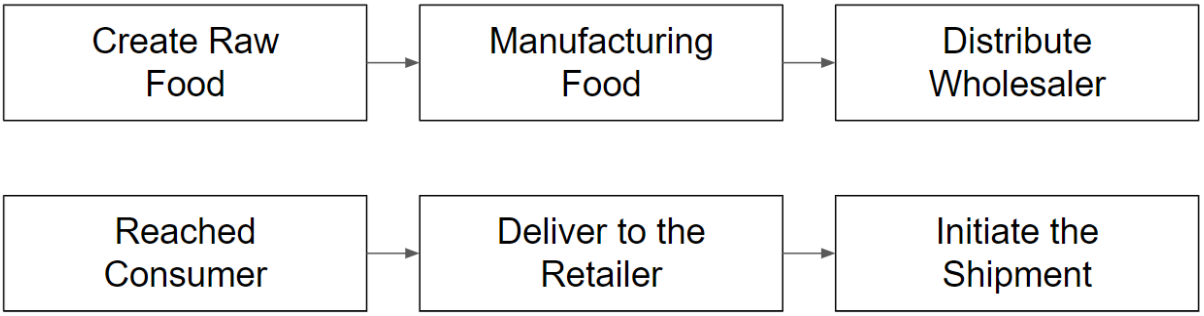


Fig 4.1 Stages in Supply Chain, corresponding status changes in the chaincode.

Module 3 - API to make connection between them

3.1 Node modules

Create function in javascript which can call the chaincode via hyperledger fabric client. Authentication is done and tokens are degenerated which can be used to make POST and GET requests from the flutter application.

4.2 Non-Functional Requirements

- Feedback

Error messages must be read out wherever required. For example, if there is not enough light or the QR code cannot be detected in the window, users must be notified to adjust their device.

- Usability

  The platform must be extremely easy to use and navigate through voice commands. A simple, seamless, interactive, and intuitive interface is a top priority.

- Testability

  A lot of user testing should be conducted to identify pain points and make appropriate changes suitable before rolling out the platform to the public.

- Availability

  Even if the internet connection fails at some point, the platform should store whatever operations have been performed so as to prevent loss of any important information and to avoid creating hassle for the user to re-enter all the information.

- Maintainability and Correctness

  Any changes made by the admin after fine-tuning, improving the model to make it better must be reflected to all the users as soon as possible. The details must be correct and should not be tampered with.

- Flexibility

  The admin should be able to make any changes as permitted by the Business Rules of the platform very flexibly.

4.3 Dependencies

1. Flutter: Flutter is an open-source UI software development kit created by Google. It is used to develop cross-platform applications for Android, iOS, Linux, Mac, Windows, Google Fuchsia, and the web from a single codebase.
2. Crypto: A set of cryptographic hashing functions implemented in pure Dart
3. Path provider: A Flutter plugin for finding commonly used locations on the filesystem. Supports Android, iOS, Linux, macOS, and Windows. Not all methods are supported on all platforms.
4. QR code scanner: A QR code scanner that works on both iOS and Android by natively embedding the platform view within Flutter. The integration with Flutter is seamless, much better than jumping into a native Activity or a ViewController to perform the scan.
5. Firebase: Firebase is a Backend-as-a-Service (BaaS) app development platform that provides hosted backend services such as a real-time database, cloud storage, authentication, crash reporting, machine learning, remote configuration, and hosting for your static files.

4.4 Timeline Chart

Timeline chart for project/thesis work completion

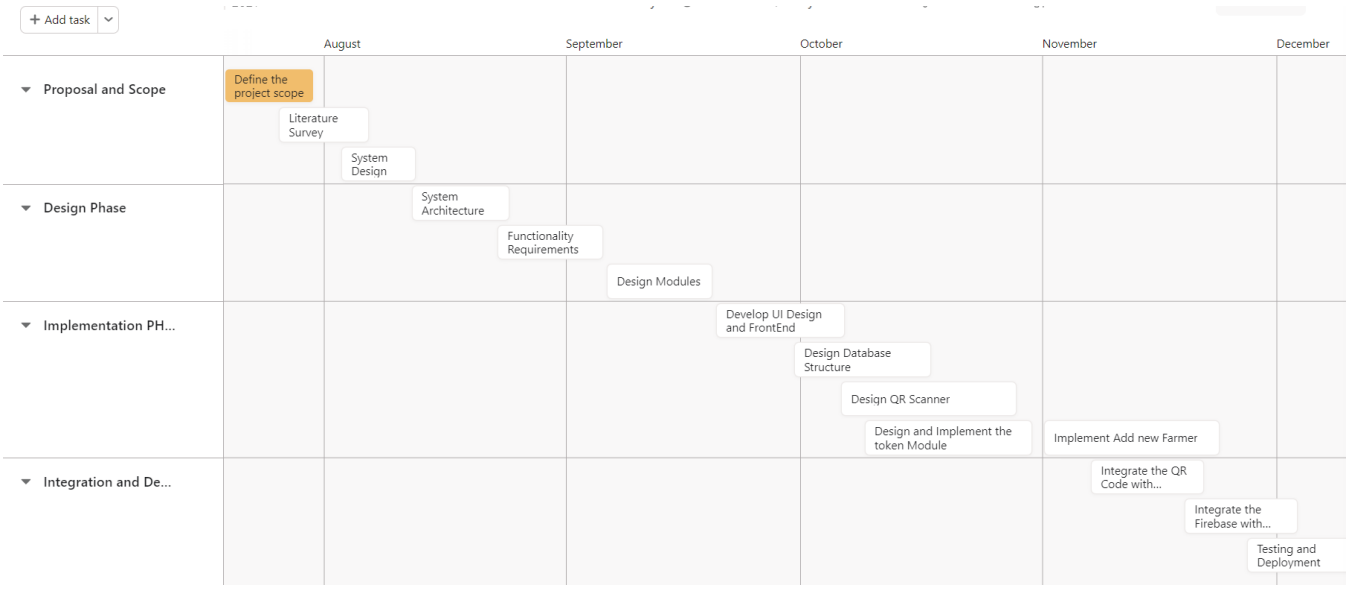
- **Phase I**

Date	**Task**
July Second fortnight	Define the project scope
August First fortnight	Research about supply chain management and product traceability.
August Second fortnight	Prepare architecture diagrams for the entire system
Sept. First fortnight	Understand requirements and features according to traceability requirements
Sept. Second fortnight	Understand and carry out extensive research on available hardware techniques.
Oct. First fortnight	Plan design of the application and prepare design document
Oct. Second fortnight	Begin prototype of phase 1- Implement authentication for the application.
Nov. First fortnight	Integrate QR code scanning in the application.

- **Phase II**

Date	**Task**
Jan Second fortnight	Research on the most suitable blockchain network.
Feb. First fortnight	Define and design the network structure in Hyperledger fabric best suited for supply chain management.
Feb. Second fortnight	Create binaries and certificates for organizations. Setup the environment to run the network in.
March. First fortnight	Develop chaincode and define functions required to update the supply chain changes and stages along the chain.
March. Second fortnight	Debug and test the chaincode by running on the network.
April. First fortnight	Built API required to connect the network with client side application.
April. Second fortnight	Integrate the frontend with Hyperledger fabric network using API developed .
May. First fortnight	Final testing of the entire application,documentation and deployment.

4.5 Gantt Chart



Figure[4.5] Gantt Chart representing the total tasks of each phase according to the timeline mentioned in 4.4.

4.6 Code and Features Implemented

4.6.1 Network Structure

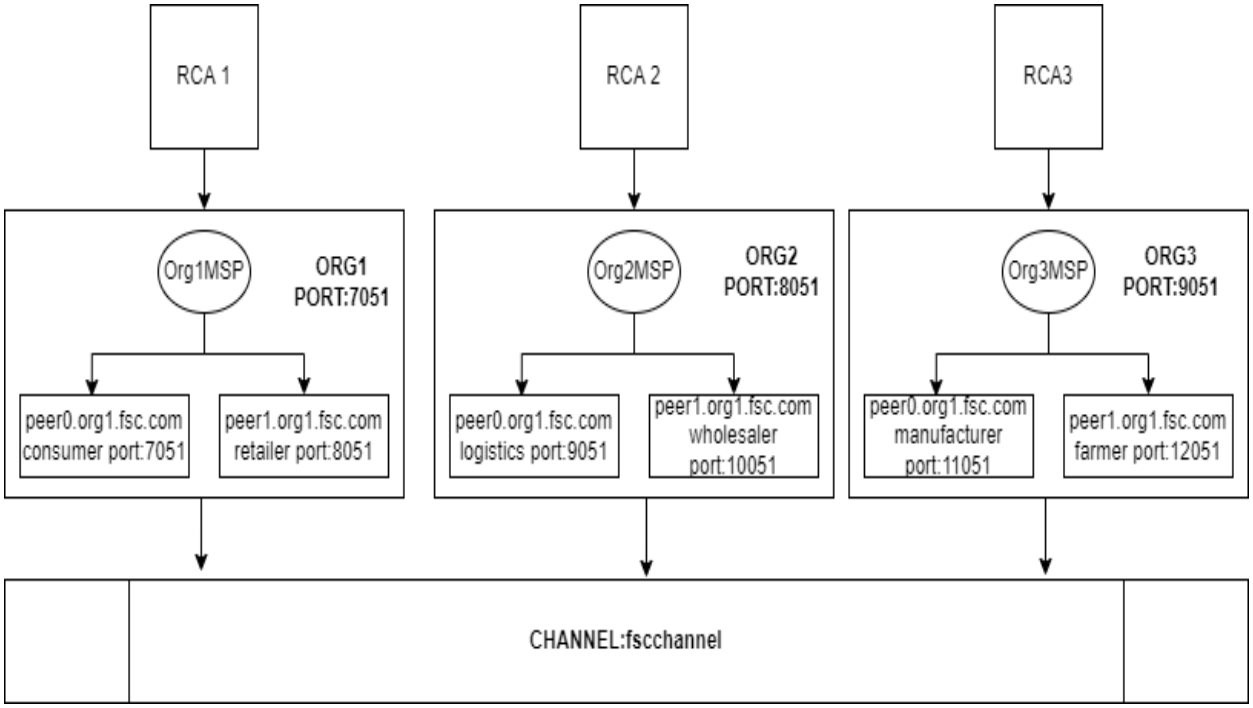


Fig 4.x Hyperledger Network Structure

- In this design fig[4.x], the organization **ORG1** hosts two peer nodes (peer0.org1.fsc.com and peer1.org1.fsc.com) representing the consumer and retailer peers, and handles the interaction involved in the consumer ordering food through the retailer store.
- Elsewhere, **ORG2** hosts two peer nodes (peer0.org2.fsc.com and peer1.org2.fsc.com) representing the logistic and wholesaler peers, and handles the wholesaler's initial shipment request to the logistical entity.
- Meanwhile, **ORG3** hosts two peer nodes (peer0.org3.fsc.com and peer1.org3.fsc.com) representing the manufacturing processor and raw food producer peers, and handles the interaction of sending raw food to manufacturers for processing and packaging.

4.6.2 Chaincode features

```
type food struct {
```

```
  OrderId              string `json:"orderId"`
  FoodId               string `json:"foodId"`
  ConsumerId           string `json:"consumerId"`
  ManufactureId        string `json:"manufactureId"`
  WholesalerId         string `json:"wholesalerId"`
  RetailerId           string `json:"retailerId"`
  LogisticsId          string `json:"logisticsId"`
  Status               string `json:"status"`
  RawFoodProcessDate   string `json:"rawProcessDate"`
  ManufactureProcessDate string `json:"manufactureProcessDate"`
  WholesaleProcessDate  string `json:"wholesaleProcessDate"`
  ShippingProcessDate   string `json:"shippingProcessDate"`
  RetailProcessDate     string `json:"retailProcessDate"`
  ProduceName          string `json:"name"`
  Grade                string `json:"grade"`
  OrderPrice           int    `json:"orderPrice"`
  ShippingPrice        int    `json:"shippingPrice"`
  DeliveryDate         string `json:"deliveryDate"`
}
```

Food structure keeps track of each variable. OrderId is unique to identify an order on the network. Status are of six types indicating at which stage the food has reached on the supply chain.

4.6.3 QR Module

Future<Uint8List> GetQRImage(List dataList) async{

```
String dataString = "";
for (int i = 0; i < dataList.length; i++) {
  dataString += dataList[i];
  if (i == dataList.length - 1) break;
  dataString += ";";
}
```

4.6.4 Unique code generate

```
String generatehash(String id, String org, String product) {
   String time = DateTime.now().toString();
   String location = "Himachal";

   String scancontents = id + org + time + location + product;
   var bytes = utf8.encode(scancontents);
   var digest = sha256.convert(bytes);
   print("Digest as bytes: ${digest.bytes}");
   uniqueidentifiers
      .add([digest.toString(), id, product, org, time, location]);
   print(uniqueidentifiers);
   return digest.toString();
}
```

4.7 Execution

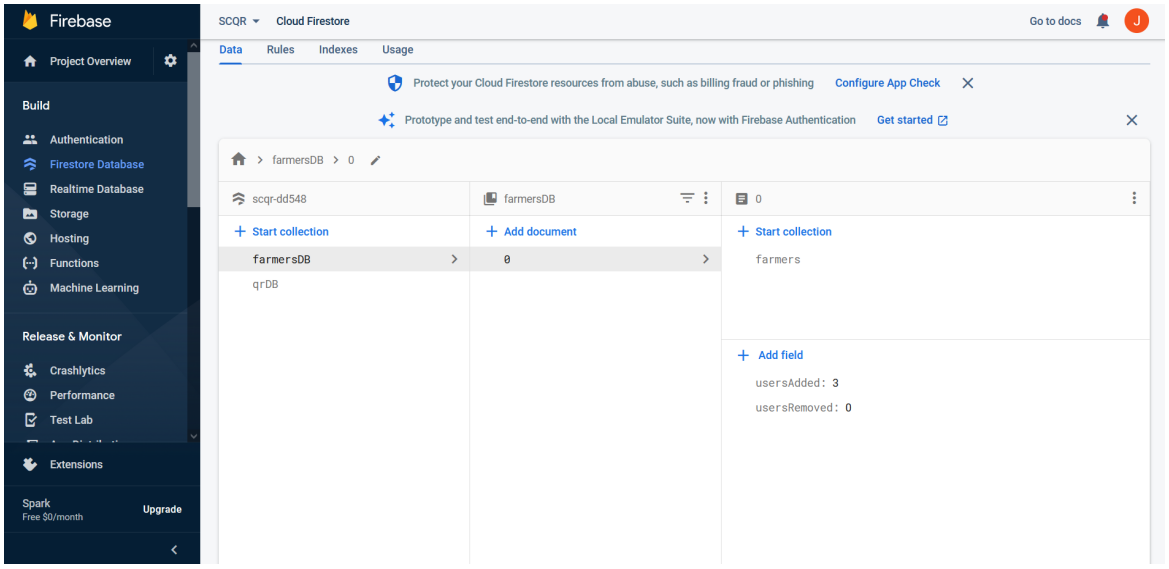


Fig 4.1 Firebase for database

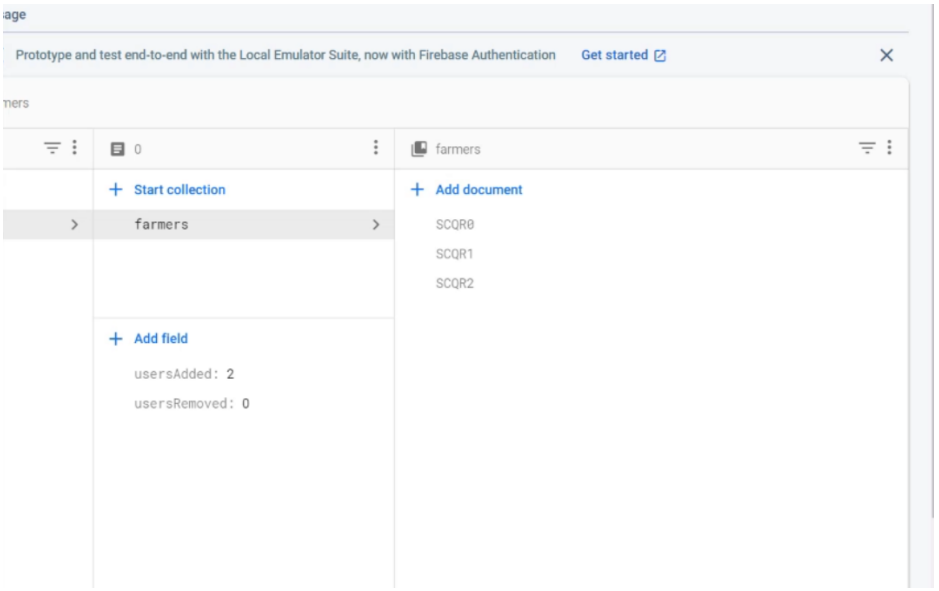


Fig 4.2 Firebase initial data

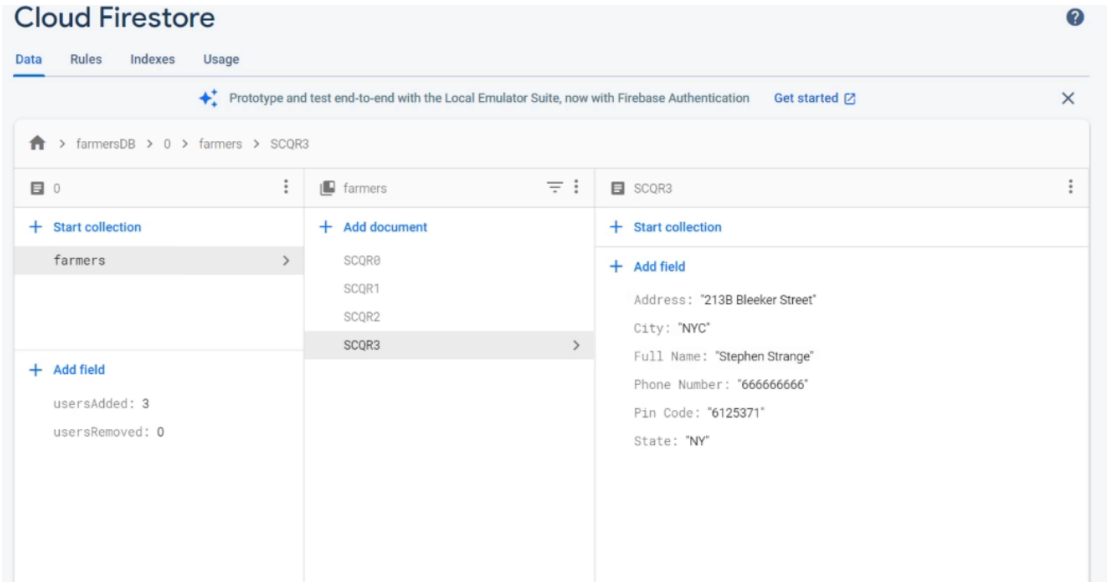


Fig 4.4 Generate QR Code for the item supplied

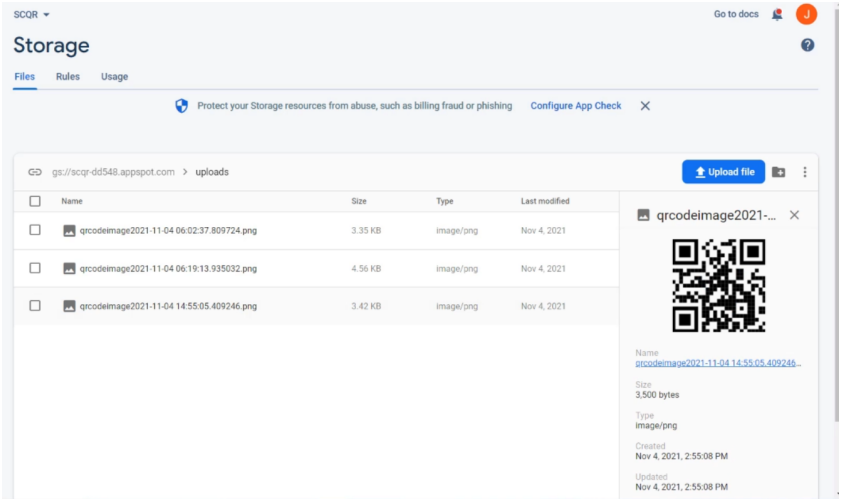


Fig 4.5 QR Code added to the firebase

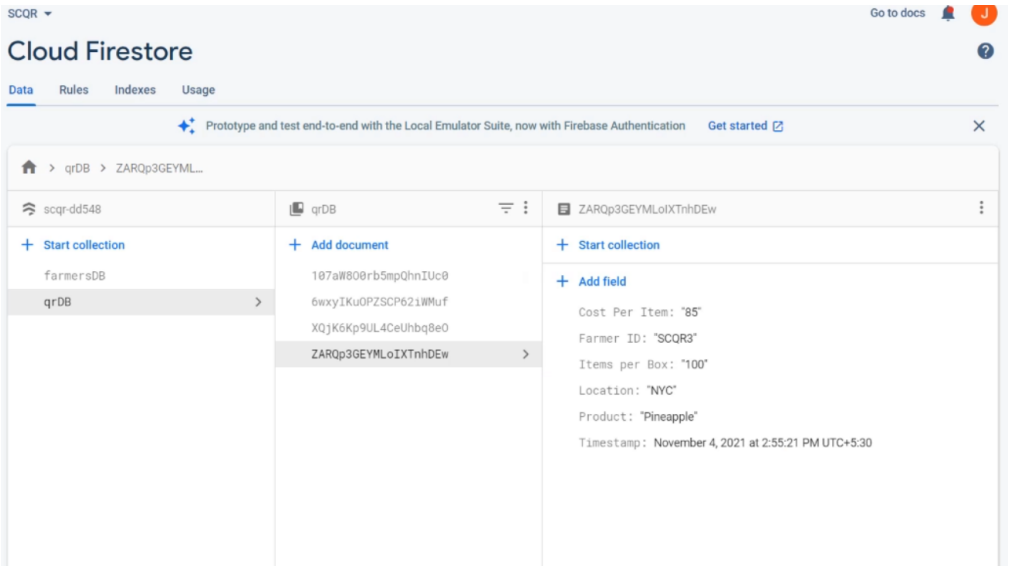


Fig 4.7 Anytime we scan the data with the time-stamp is updated in the database and thus we can keep track of all the items in the supply chain.

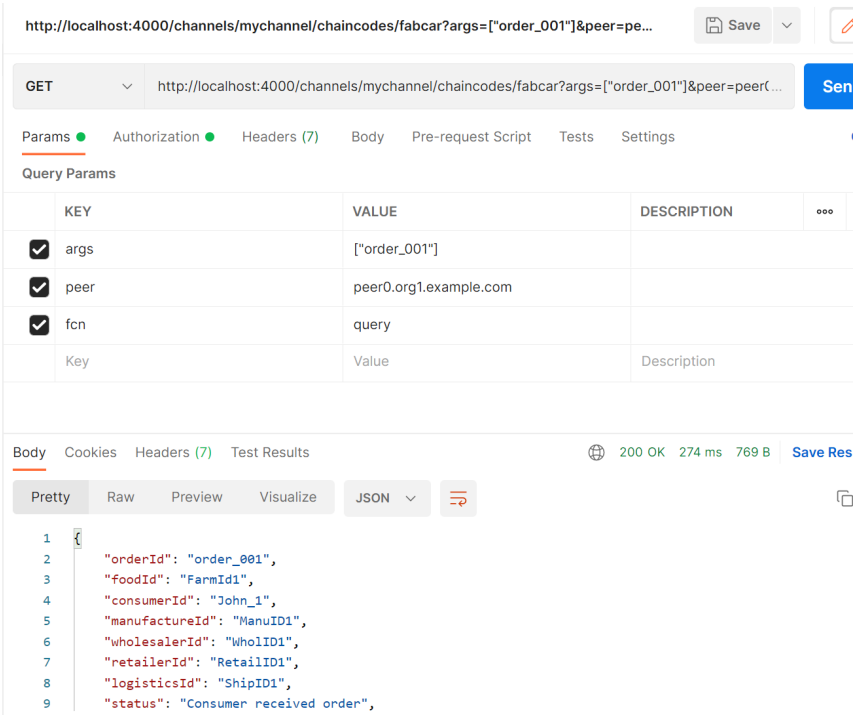


Fig 4.8 Query transaction from the existing World State.

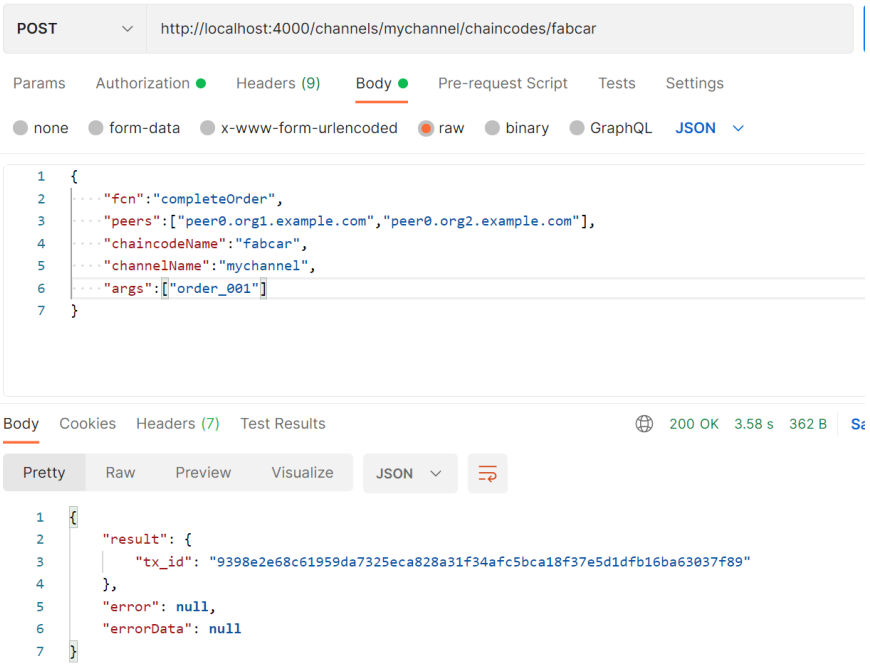


Fig 4.9 Invoking contract to make changes to the world state.

```
[2022-05-12 21:48:46.897] [DEBUG] invoke-chaincode - Logging proposal Responses { version: 1,
  timestamp: null,
  response: { status: 200, message: '', payload: <Buffer > },
  payload:
   <Buffer 0a 20 aa 88 a0 84 7c 1b 82 80 0b 0f a2 fb 35 bd ce f2 8e b2 01 4c ee 26 b7 10 ed cb 26 6e d
e 09 49 3e 12 a1 05 0a 8c 05 12 37 0a 0a 5f 6c 69 66 65 63 ... >,
  endorsement:
   { endorser:
      <Buffer 0a 07 4f 72 67 31 4d 53 50 12 aa 06 2d 2d 2d 2d 2d 42 45 47 49 4e 20 43 45 52 54 49 46 4
9 43 41 54 45 2d 2d 2d 2d 2d 0a 4d 49 49 43 4b 44 43 43 41 63 ... >,
     signature:
      <Buffer 30 44 02 20 2f ef dd b1 32 2b 23 da 5e 83 72 bb 63 d0 06 d4 18 eb 6e de 07 0e 5e 05 b8 7
c 77 eb 80 5f 00 35 02 20 26 89 e9 68 47 78 bd 1d f2 4e e9 08 ... > },
  peer:
   { url: 'grpcs://localhost:7051',
     name: 'peer0.org1.example.com',
     options:
      { 'grpc.max_receive_message_length': -1,
        'grpc.max_send_message_length': -1,
        'grpc.keepalive_time_ms': 120000,
        'grpc.http2.min_time_between_pings_ms': 120000,
        'grpc.keepalive_timeout_ms': 20000,
        'grpc.http2.max_pings_without_data': 0,
        'grpc.keepalive_permit_without_calls': 1,
        name: 'peer0.org1.example.com',
        'grpc.ssl_target_name_override': 'peer0.org1.example.com',
        'grpc.default_authority': 'peer0.org1.example.com' } } }
[2022-05-12 21:48:46.898] [INFO] invoke-chaincode - invoke chaincode proposal was good
```

Fig 4.9 Checking the endorsement policy before sending a proposal.

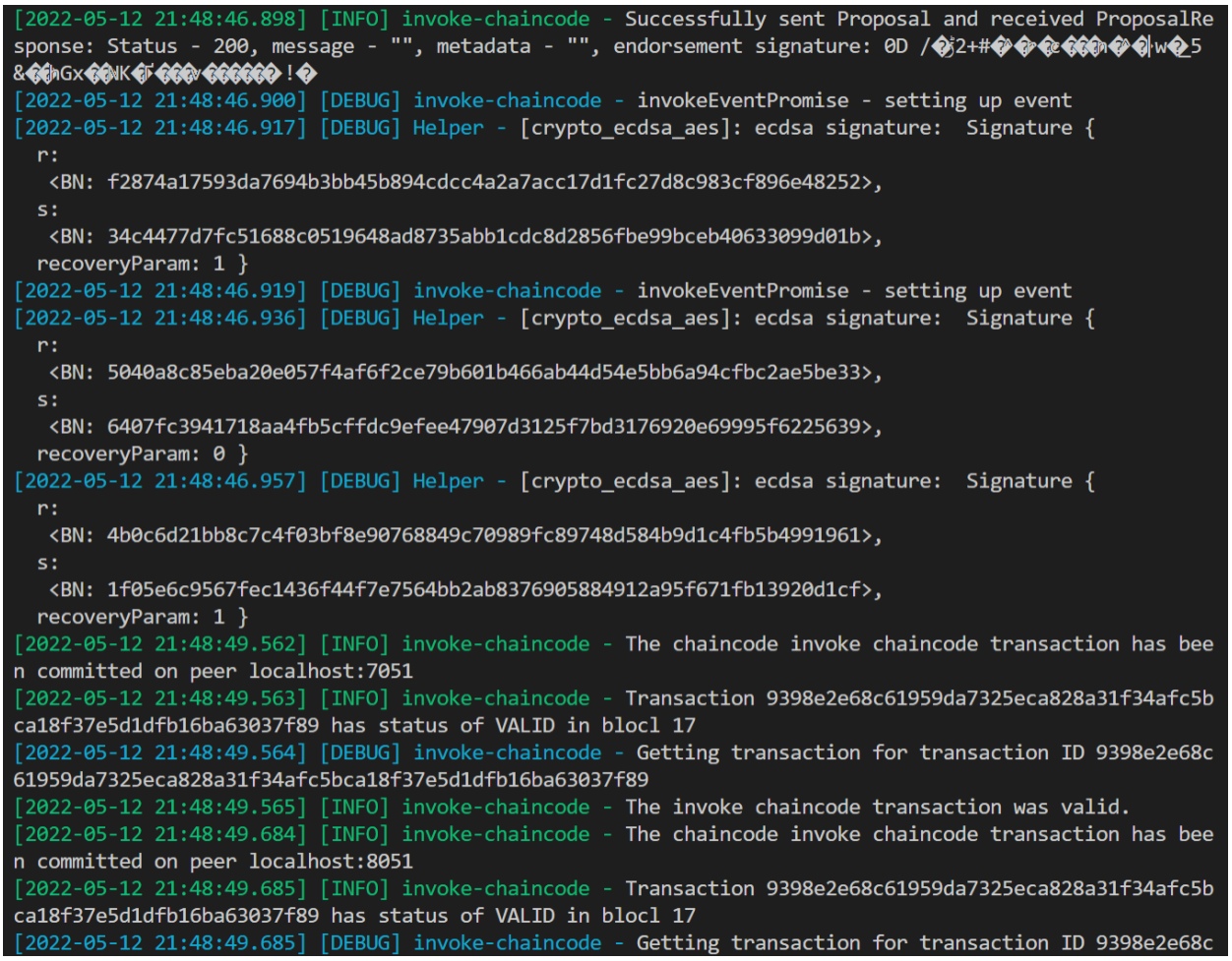


Fig 4.10 Checking for required signatures based on the endorsement policy set.

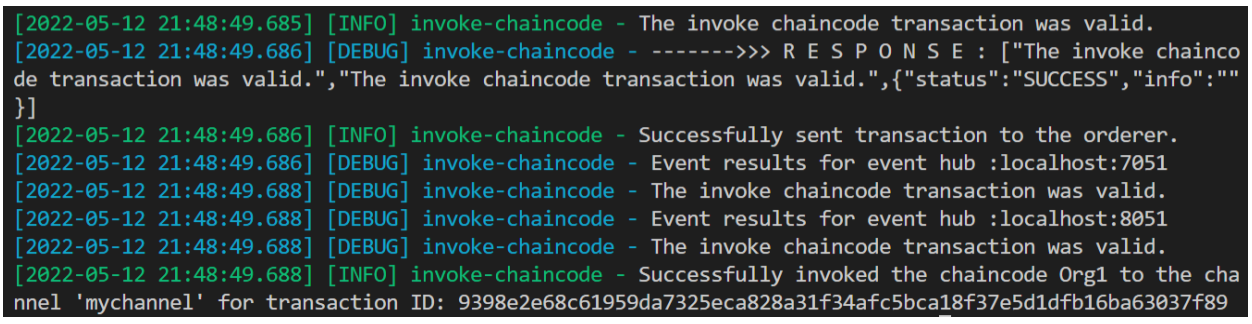
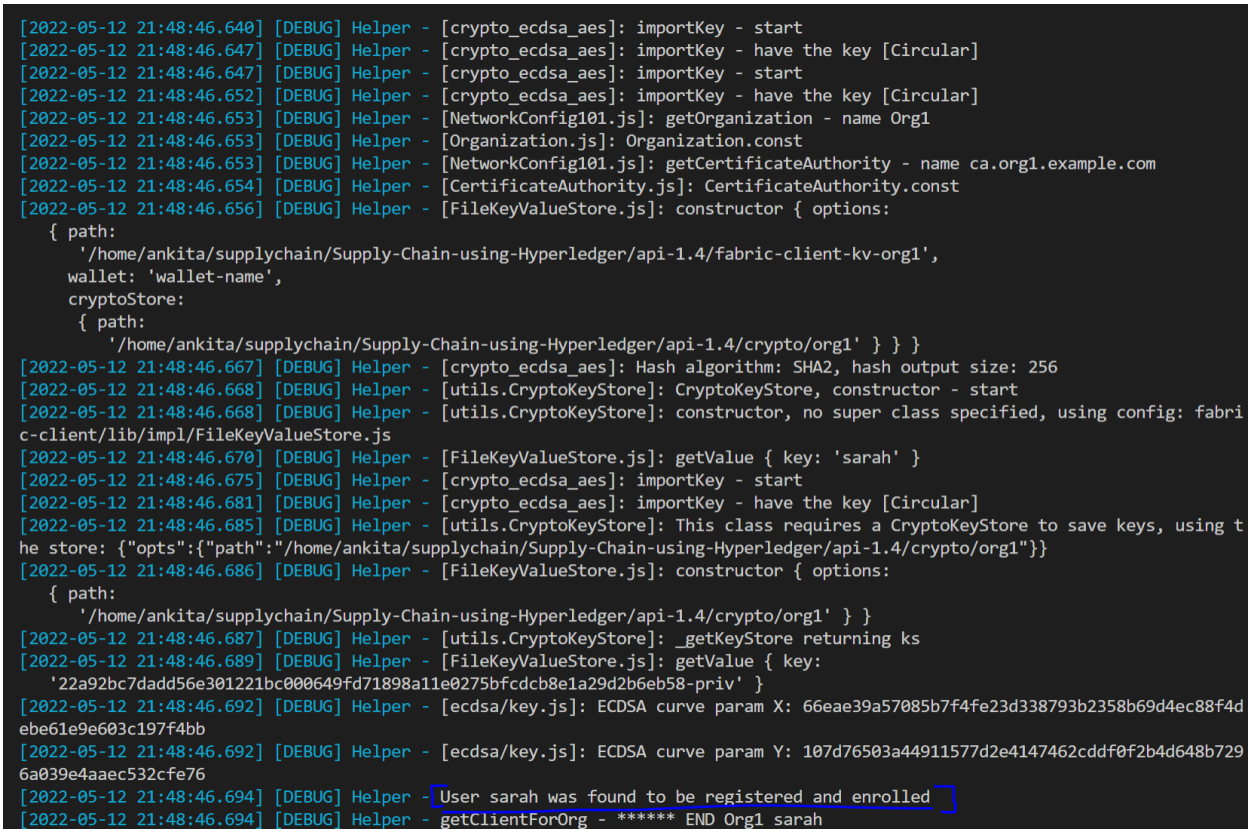
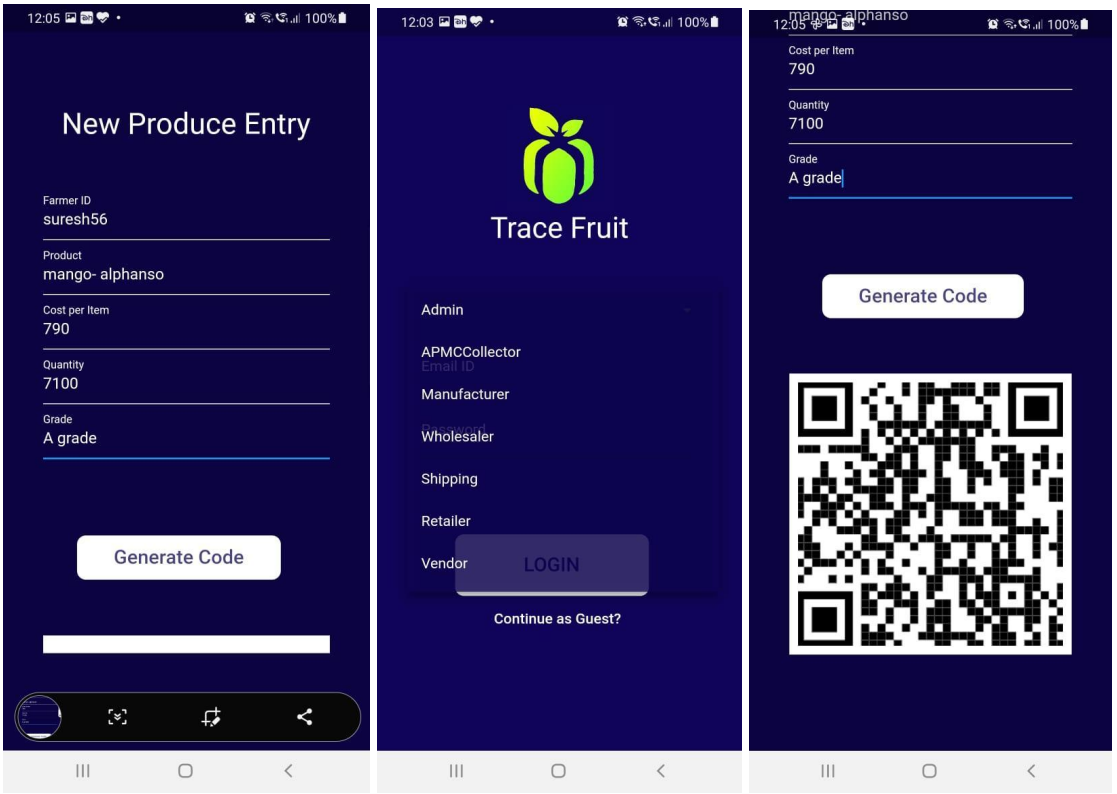


Fig 4.11 Sending the success message to the orderer after making changes to the world state.

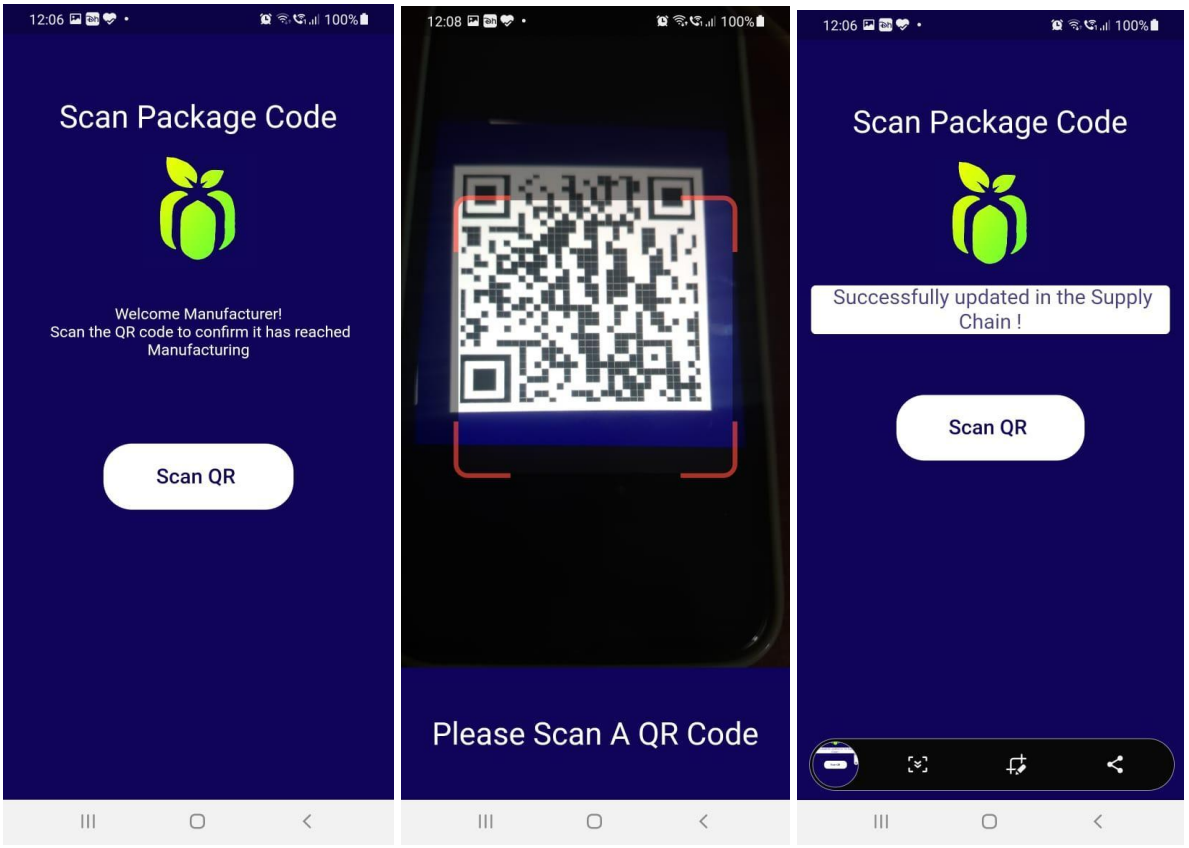


4.12 Register users and generate the corresponding certificates from CA authority.



(a) **(b)** **(c)**

Fig 4.13 (a) Adding raw product information to the chain (b) Login page to choose user type and action corresponding to it later (c) Generated QR code on the given information converted to SHA256 hash.



(a) **(b)** **(c)**

Fig 4.14 (a) Manufactured logged in successfully, will scan QR to update the package stage in the supply chain. (b) Scanning the QR from the package (c) Success message after making necessary updates in the network.

Fig 4.15 Query function invoked after scanning any QR code. This will return all the information regarding dates and users who made requests to update the status on the blockchain.

Chapter 5

Conclusion and Further Work

5.1 Conclusions

The implementation of project phase I completed as of Dec 2021 :

1. The User Interface for the android application was designed and implemented with Flutter development supporting both android and IOS devices.
2. Module 1 to add a new farmer and generate a unique ID was implemented and tested.
3. Various cryptographic functions were analyzed and the SHA-256 hash function was used to combine the product details and make a unique hex code.
4. The QR generate and scan module was implemented and integrated with the system.
5. The Firebase structure was designed and connected with the front-end application and the initial prototype was successfully delivered.

The implementation of project phase II is completed as of May 2022 :

1. The Blockchain network was designed and implemented in Hyperledger fabric. Organizations, policies, endorsement policies, peers were defined and executed.

2. Chaincode was made in Go lang with necessary functions to make Status change in order placed and track the package throughout the chain with Status tracking.
3. Location, date were also added to the fields of structure. This ensured the location was authentic and no tampering occurred.
4. API was built using fabric client and Node JS, to make necessary calls between the client side application and the hyperledger network.
5. Integration of the Phase I and Phase II was successfully completed .

5.2 Further Work

The next phase is focused on creating a Kubernetes cluster with Google Cloud Kubernetes Engine to deploy the containerized app in the cloud. Kubernetes engine eliminates the need to manage, install and operate your own clusters. Although right now our application is a prototype, it can be easily migrated to a scalable production environment. With scalability to accomodate more number of participants in the chain, we plan to further develop IOS based mobile application which will ensure a reach to large masses.

We can also think of integrating an IOT based application having sensors to monitor the temperature and quality of the product at each stage, with the application .

Multiple language support will facilitate participants from all around the world to participate in a single blockchain network. This will help the network grow and correspondingly will increase the number of transactions being validated and number of blocks confirmed placed in the blockchain.

Providing accessibility features such as voice based commands can ensure that people with certain disabilities can also work with the application and encourage them to pursue jobs in the agricultural market supply chain.

5.3 Bibliography

[1]Kaushik, A., Choudhary, A., Ektare, C., Thomas, D., & Akram, S. (2017, May). Blockchain—literature survey. In *2017 2nd IEEE International Conference on Recent Trends in Electronics, Information & Communication Technology (RTEICT)* (pp. 2145-2148). IEEE.

[2]Gohil, D., & Thakker, S. V. (2021). Blockchain-integrated technologies for solving supply chain challenges. *Modern Supply Chain Research and Applications*.

[3]Antal, C., Cioara, T., Antal, M., & Anghel, I. (2021). Blockchain platform for COVID-19 vaccine supply management. *IEEE Open Journal of the Computer Society*, *2*, 164-178.

[4]Basnayake, B. M. A. L., & Rajapakse, C. (2019, March). A Blockchain-based decentralized system to ensure the transparency of organic food supply chain. In *2019 International Research Conference on Smart Computing and Systems Engineering (SCSE)* (pp. 103-107). IEEE.

[5]Hegde, B., Ravishankar, B., & Appaiah, M. (2020, February). Agricultural supply chain management using blockchain technology. In *2020 International Conference on Mainstreaming Block Chain Implementation (ICOMBI)* (pp. 1-4). IEEE.

[6]Arena, A., Bianchini, A., Perazzo, P., Vallati, C., & Dini, G. (2019, June). BRUSCHETTA: An IoT blockchain-based framework for certifying extra virgin olive oil supply chain. In *2019 IEEE International Conference on Smart Computing (SMARTCOMP)* (pp. 173-179). IEEE.

[7]Marchese, A., & Tomarchio, O. (2021). An agri-food supply chain traceability management system based on hyperledger fabric blockchain. In *Proceedings of the 23rd International Conference on Enterprise Information Systems (ICEIS2021)* (Vol. 2, pp. 648-658).

5.4 References

1) https://bitcoin.org/bitcoin.pdf
2) https://ieeexplore.ieee.org/abstract/document/8256979
3) https://ieeexplore.ieee.org/abstract/document/9395097
4) https://www.emerald.com/insight/content/doi/10.1108/MSCRA-10-2020-0028/full/html
5) https://ieeexplore.ieee.org/document/8842690/authors#authors
6) https://www.youtube.com/watch?v=coQ5dg8wM2o
7) https://bitcoin.org/bitcoin.pdf
8) https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=7163223
9) https://www.ibm.com/blockchain/supply-chain
10) https://docs.soliditylang.org/en/v0.8.9/
11) https://flutter.dev/

5.5 Acknowledgment

We hereby appreciate the help provided by our mentor Prof. Swapnil Pawar and our project guide Dr. Ninad Mehendale, Dr. Shyam Masakpalli for their guidance. We are grateful to the Blockchain community for providing us with insightful research.