T.C. Maltepe University

Faculty of Engineering and Natural Sciences

Software Engineering Department

A Blockchain-Based Fake Product Detection Application

SE 401 Project Report

170706008

Kutay Öndersev

Project Advisor: Mehmet Ali Aksoy Tüysüz

CONTENTS

1. Introduction

The production and trade of counterfeit products is growing day by day. Many parties can be adversely affected as a result of counterfeit product trade. While the consumer suffers financial loss, it can also be negatively affected in matters such as health. In addition, trust in companies whose products are sold counterfeit decreases and may adversely affect the sales of original products. In addition, each manufacturer may not be able to open their own store and meet the product with the user before purchasing, so they may have problems in proving their originality. And companies from time to time have to engage third parties to prove the authenticity of their product. With this project, these intermediaries will disappear and only the manufacturer will prove the authenticity of the product to the user. There are several methods used to check product authenticity. The worst of these is to physically examine the original product with the counterfeit product. This is a tricky and misleading method. Because manufacturers of counterfeit products may be competent to replicate the original product. In addition, time and resource investment will be high for this method. Another method is to direct with a qr code placed on the product. Usually, when the qr code is scanned, the product's website is redirected. This method cannot be a proof of authenticity. A safer solution is required.

1.1 Purpose and Importance of the Project

The aim of this project is to provide the end user with a system that can easily check whether the product is original or not. This system will enable manufacturers, vendors and end-user actors to establish a trust network with each other. The end user will see all the information recorded in our system by the manufacturer regarding the production and distribution stages of the product they control. And he/she will make sure that this information is unchangeable, correct information. When he/she realizes that the product is original, he will shop with confidence. Thus, the manufacturer will not only protect its brand reputation without being harmed by counterfeit products, but will also be appreciated for the transparency it offers to its customers. While the manufacturer enters the production stages of the product into our system, he will make sure that this data is kept in a secure infrastructure. All the data entered in the system are guaranteed by him, that they are entered by him and cannot be changed. Thus, it is transparent and secure from end to end.

1.2 The Innovation (originality) of the Project

Presenting all production stages of a product transparently to the user is a new innovation, especially for small businesses and manufacturers. Large companies and manufacturers may try to convey authenticity to their customers, as long as they allocate sufficient financial resources, although it is not known how safe it is. But for small producers this is costly. Small or large manufacturers will be able to prove the originality of their products thanks to this project. Another innovation is that the data in our system is stored on the blockchain base. Blockchain is the building block of your system that ensures data consistency and trust. Every production stage written on the blockchain is presented to the end user for review. And this correct information proves the authenticity of the product.

1.3 Technology Areas to which the Project is Related

This project is a web project and will run in client-server architecture. The manufacturer will have an interface where they can save the production stages. And there will be another interface where the consumer can only query. The consumer will be faced with a screen where he/she will be able to make the inquiry using a QR code or a code produced exclusively for the product, and have information about the product. The technologies I will use while doing this project are Java Springboot (for web backend development), PostgreSQL (for holding manufacturer information), Solidity (for developing blockchain transactions), Web3js, JavaScript, and a blockchain platform (Ethereum).

1.4 Detailed Description of The Project

This project has the most accurate application possible today due to the philosophy of the technology it uses and the solution method required by the problem. The basic premise of blockchain technology and the surrounding technologies is that the system is immutable and decentralized. Keeping the data in the blocks in the chain with the data in the previous block with the cryptography method has become popular because the system data is both unique and unchangeable, so it does not need a central control or middle man to provide controls.

While blockchain is most famous for its role in facilitating the rise of digital currencies over the past several years, there are also many other non-cryptocurrency uses for this technology. Indeed, some blockchain proponents believe that the technology could far outpace cryptocurrencies themselves in terms of its overall impact, and that the real potential of blockchain is only just now being discovered. As such, it's likely that financial advisors and many others in the investing world will encounter blockchain technology much more in the years to come, whether it is linked with a specific cryptocurrency or if it's being utilized in any number of other applications. [1]

Believing that the purchased product is original occurs with confidence. If you trust the person, institution or document that says the product is original, then you will not hesitate to buy it. Ensuring this reliability is a multi-parameter issue. Because trusting the person who provides the trust is also in question. For example, if there is no trust in the institutions that give this assurance, what is said is meaningless. Blockchain in this case is a system, not an organization managed by a group of people. And the system itself provides trust by the way it exists. Therefore, it can be accepted that it is the number one choice in industries where immutability and decentralized are needed.

On the other hand, a topic that increases trust is the details. The more detailed a subject is, the more open and transparent it is. Therefore, it is reliable. Thanks to this project, companies will be able to enter detailed information about their products and this information will be displayed to the end user on a reliable system.

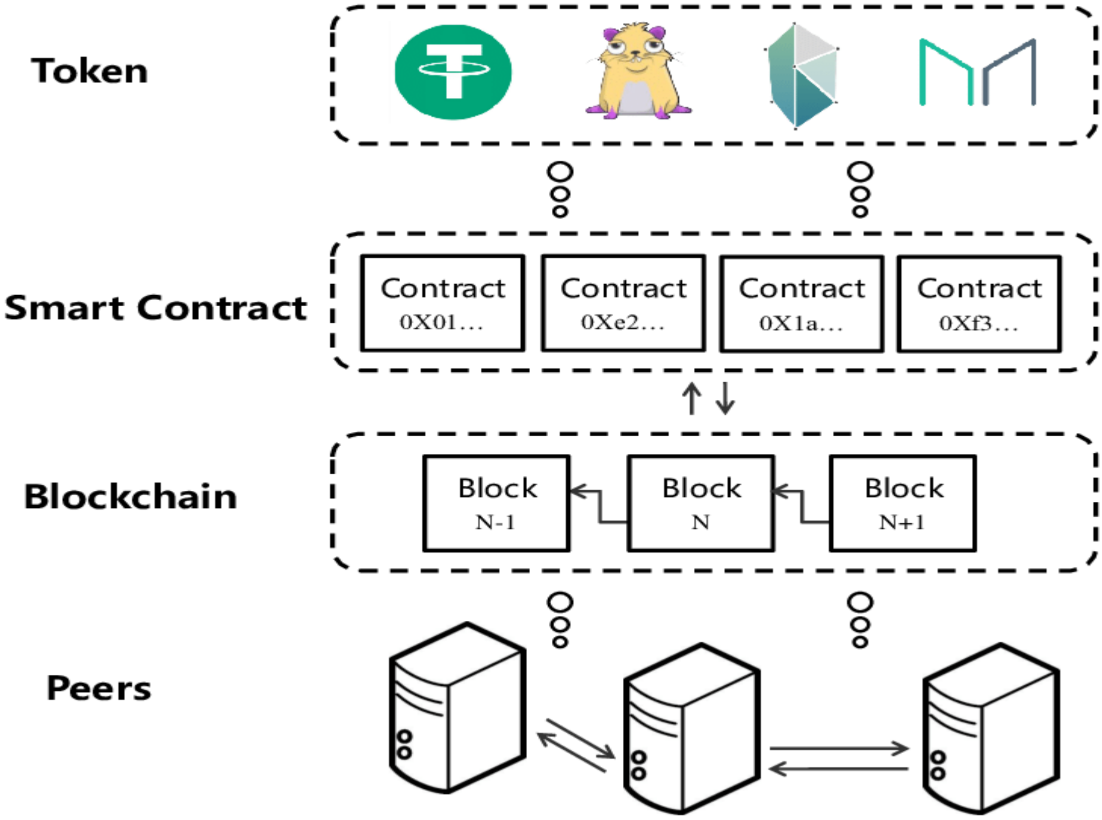
1.5 Detailed Problem Requirements

What is a Blockchain

A blockchain is a distributed database that is shared among the nodes of a computer network. As a database, a blockchain stores information electronically in digital format. Blockchains are best known for their crucial role in cryptocurrency systems, such as Bitcoin, for maintaining a secure and decentralized record of transactions. The innovation with a blockchain is that it guarantees the fidelity and security of a record of data and generates trust without the need for a trusted third party. [2]

One key difference between a typical database and a blockchain is how the data is structured. A blockchain collects information together in groups, known as blocks, that hold sets of information. Blocks have certain storage capacities and, when filled, are closed and linked to the previously filled block, forming a chain of data known as the blockchain. All new information that follows that freshly added block is compiled into a newly formed block that will then also be added to the chain once filled.

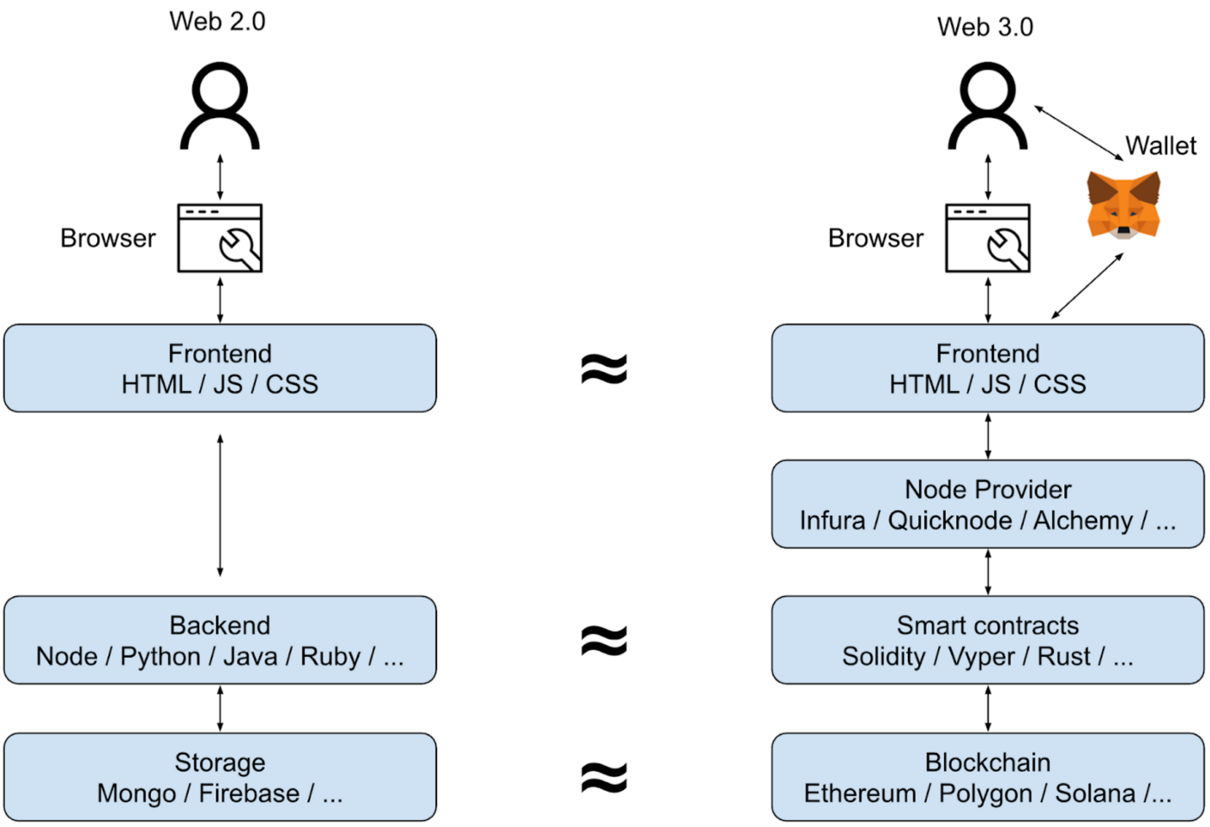
A database usually structures its data into tables, whereas a blockchain, like its name implies, structures its data into chunks (blocks) that are strung together. This data structure inherently makes an irreversible time line of data when implemented in a decentralized nature. When a block is filled, it is set in stone and becomes a part of this time line. Each block in the chain is given an exact time stamp when it is added to the chain.



Web3.js is a collection of libraries that allow you to interact with a local or remote ethereum node using HTTP, IPC or WebSocket. [3]

The ethers. js library aims to be a complete and compact library for interacting with the Ethereum Blockchain and its ecosystem. It was originally designed for use with ethers.io and has since expanded into a more general-purpose library.

Theese libaryies provide us that interact the smart contract where deployed Ethereum blockchain and this way we can use our smart contract functions.



What is a React.js

React. js is an open-source JavaScript library that is used for building user interfaces specifically for single-page applications. It's used for handling the view layer for web and mobile apps. React also allows us to create reusable UI components.

What is a SpringBoot

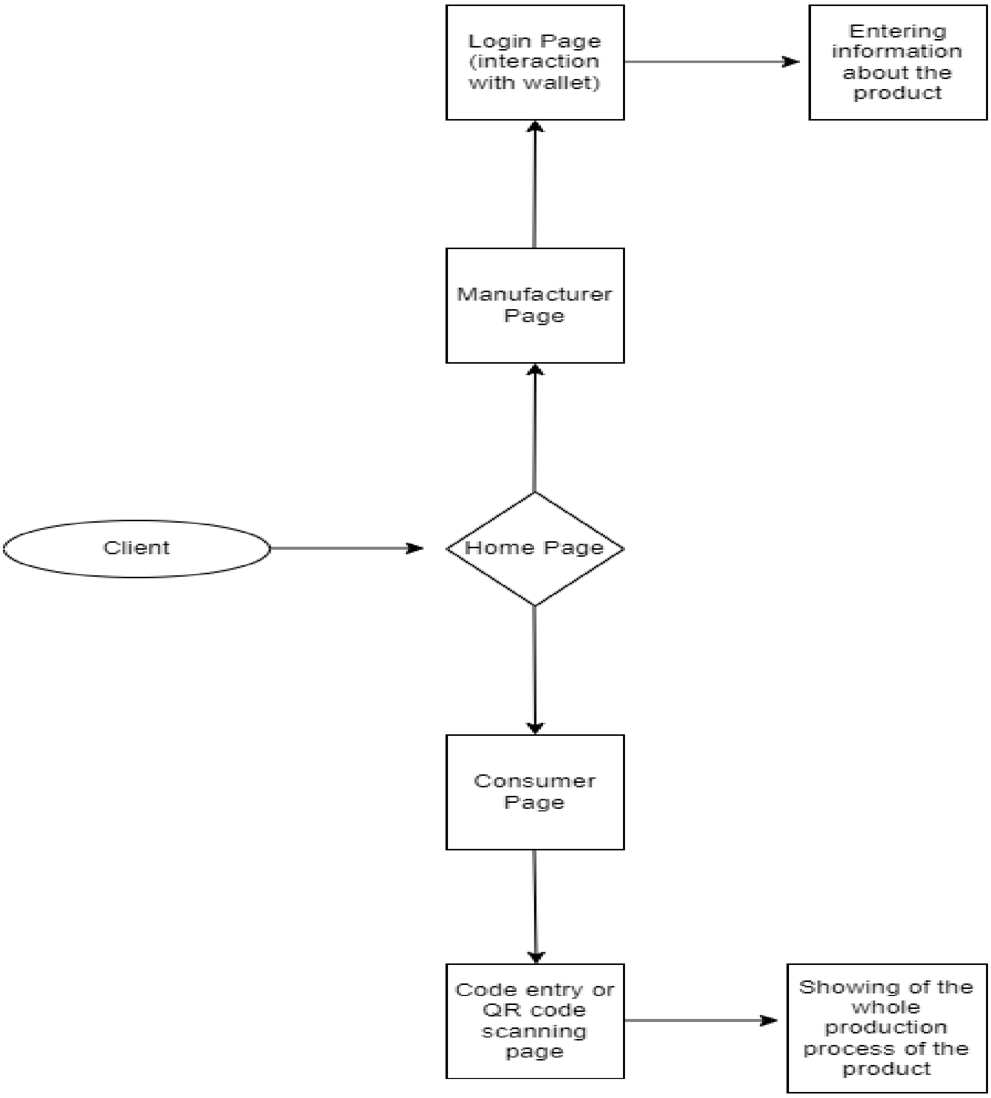
Spring Boot helps developers create applications that just run. Specifically, it lets you create standalone applications that run on their own, without relying on an external web server, by embedding a web server such as Tomcat or Netty into your app during the initialization process. We use as a backend server technology.

In this project, the database which we need to store main data is blockchain actually but any relational database is used too for store unnecessary user information data.

1.6 High Level Design

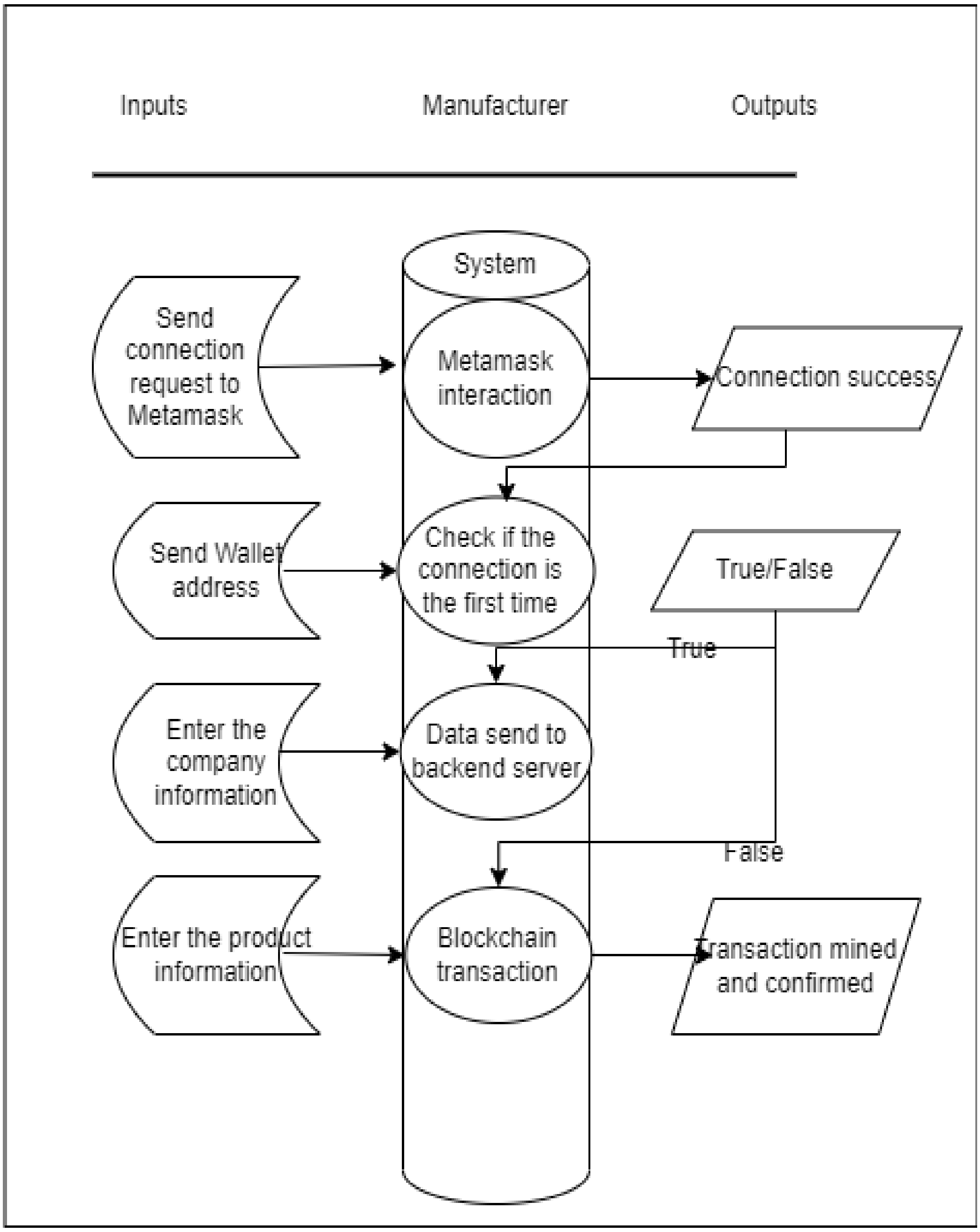
The website was created to be used by two actors. Both the consumer and the manufacturer (manufacturer) will be able to make their own transactions. Depending on which actor is on the homepage of the site, the user is asked to choose in order to be able to redirect. If the user chooses the role of manufacturer, we understand that he wants to see the production stages of a product and check its originality. At this stage, the user will have a code information about the product he will control. This code acts as a key to show the production stages of the product. After entering that code on this screen, information about the product will be listed for the user. With this page, the consumer will have extensive information about the product.

If the user chooses the role of manufacturer (manufacturer), it means that the manufacturer wants to register a product. At this stage, the producer goes through a login process. Once the manufacturer is included in the system, he can now enter and save information about his product. When information is recorded, it becomes information written on the blockchain.

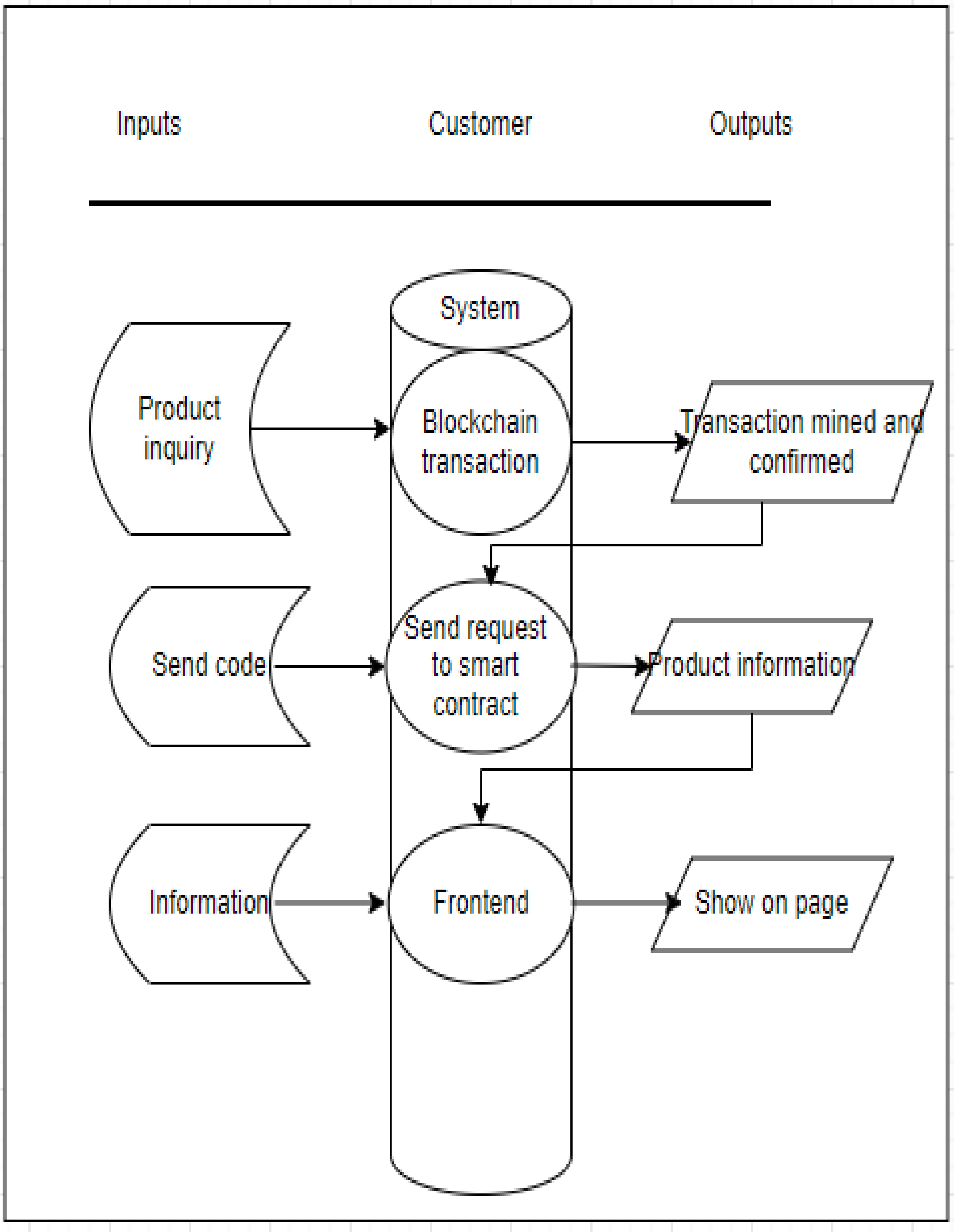

Login Page (interaction with wallet)
Entering information about the product
Manufacturer Page
Client
Home Page
Consumer Page
Code entry or QR code scanning page
Showing of the whole production process of the product

2. Detailed System Architecture

2.1 Input/Output Diagrams


Inputs
Manufacturer
Outputs
System
Send connection request to Metamask
Metamask interaction
Connection success
Send Wallet address
Check if the connection is the first time
True/False
True
Enter the company information
Data send to backend server
False
Enter the product information
Blockchain transaction
Transaction mined and confirmed

Metamask interaction is required in order to trigger our smart contract with the Manufacturer digital wallet. When the metamask connection is successful, this result is captured and the address of the person is sent to the backend server. Here, it is checked whether the person is connecting to our system for the first time. If it is the first time to connect, some information about itself is requested. This information is information that will not cause any security problems and is only required to create the profile of the person. If the person has logged in before, they can now create a product record. After filling and sending the product information, this information will be written to the blockchain via the smart contract. This transaction will be confirmed by miners on the blockchain. After the transaction is confirmed, a result in the form of transaction mined will be obtained. This means that the transaction was successful and the data was written to the blockchain.



After the customer enters our system, he/she will only have an inquiry code for that product. This code actually helps us to establish a key-value relationship on the blockchain base. Thanks to this code, we have the opportunity to see the information of the product we have. The customer enters the code into the system and sends the inquiry request. This action triggers a function on our smart contract and reads over the blockchain. The read data is returned in a certain format. This returned information is processed on the frontend and presented as a page that the user can easily examine. In this way, the person questions and obtains unique information about the product.

2.2 User Classes

The main code density in this project is on the smart contract. A smart contract that will run on Ethereum is written in Solidity language.

What is Solidity ?

Solidity is an object-oriented, high-level language for implementing smart contracts. Smart contracts are programs which govern the behaviour of accounts within the Ethereum state. Solidity is a curly-bracket language. It is influenced by C++, Python and JavaScript, and is designed to target the Ethereum Virtual Machine (EVM). Solidity is statically typed, supports inheritance, libraries and complex user-defined types among other features. With Solidity you can create contracts for uses such as voting, crowdfunding, blind auctions, and multi-signature wallets. When deploying contracts, you should use the latest released version of Solidity. Apart from exceptional cases, only the latest version receives security fixes. Furthermore, breaking changes as well as new features are introduced regularly. Currently used a 0.y.z version number to indicate this fast pace of change. [6]

It contains functions and values in the field that we define as a solidity contract (it can be thought of as a class, but it is not exactly the same). However, since every transaction made in the Ethereum network has a cost, the deployment of the written smart contracts to the blockchain network is also a cost. For this reason, the written contracts should be as simple and optimized as possible. There are many points to consider for both the security and cost-effectiveness of our code.

On the backend side, the classes to be used will be in a layered architecture according to the MVC model.

What is MVC ?

Model–view–controller (MVC) is a software design pattern commonly used for developing user interfaces that divide the related program logic into three interconnected elements. This is done to separate internal representations of information from the ways information is presented to and accepted from the user.

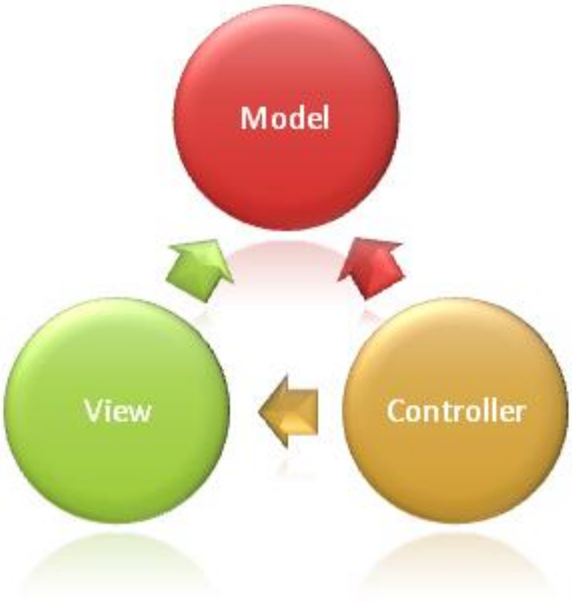
Traditionally used for desktop graphical user interfaces (GUIs), this pattern became popular for designing web applications. Popular programming languages have MVC frameworks that facilitate implementation of the pattern.

Model : The central component of the pattern. It is the application's dynamic data structure, independent of the user interface. It directly manages the data, logic and rules of the application.

View : Any representation of information such as a chart, diagram or table. Multiple views of the same information are possible, such as a bar chart for management and a tabular view for accountants.

Controller : Accepts input and converts it to commands for the model or view. [7]

We will have an Entity class, that is, a model class, where Company information will be kept. The methods that will perform the necessary operations on this data will be written in the CompanyService class in the Service layer. This class will keep its connection with the database table where the company information information is kept in the Company information repository interface with dependency injection.



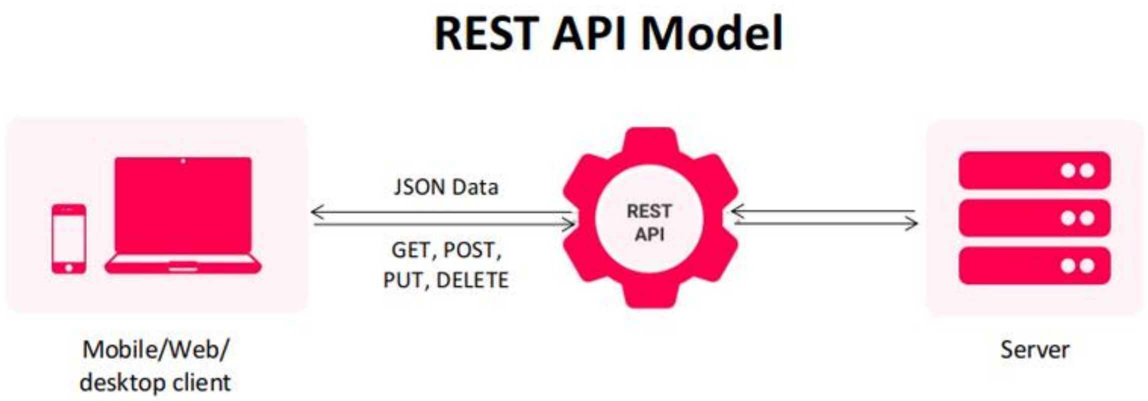
What is Dependency Injection ?

In software engineering, dependency injection is a technique in which an object receives other objects that it depends on, called dependencies. Typically, the receiving object is called a client and the passed-in ('injected') object is called a service. The code that passes the service to the client is called the injector. Instead of the client specifying which service it will use, the injector tells the client what service to use. The 'injection' refers to the passing of a dependency (a service) into the client that uses it. The service is made part of the client's state. Passing the service to the client, rather than allowing the client to build or find the service, is the fundamental requirement of the pattern. The intent behind dependency injection is to achieve separation of concerns of construction and use of objects. This can increase readability and code reuse.

Dependency injection is one form of the broader technique of inversion of control. A client who wants to call some services should not have to know how to construct those services. Instead, the client delegates to external code (the injector). The client is not aware of the injector. The injector passes the services, which might exist or be constructed by the injector itself, to the client. The client then uses the services.

This means the client does not need to know about the injector, how to construct the services, or even which services it is actually using. The client only needs to know the interfaces of the services, because these define how the client may use the services. This separates the responsibility of 'use' from the responsibility of 'construction'. [8]

By using these methods, our class that will work as a RESTFUL API that will send information in JSON format to the outside world will be our CompanyController class.



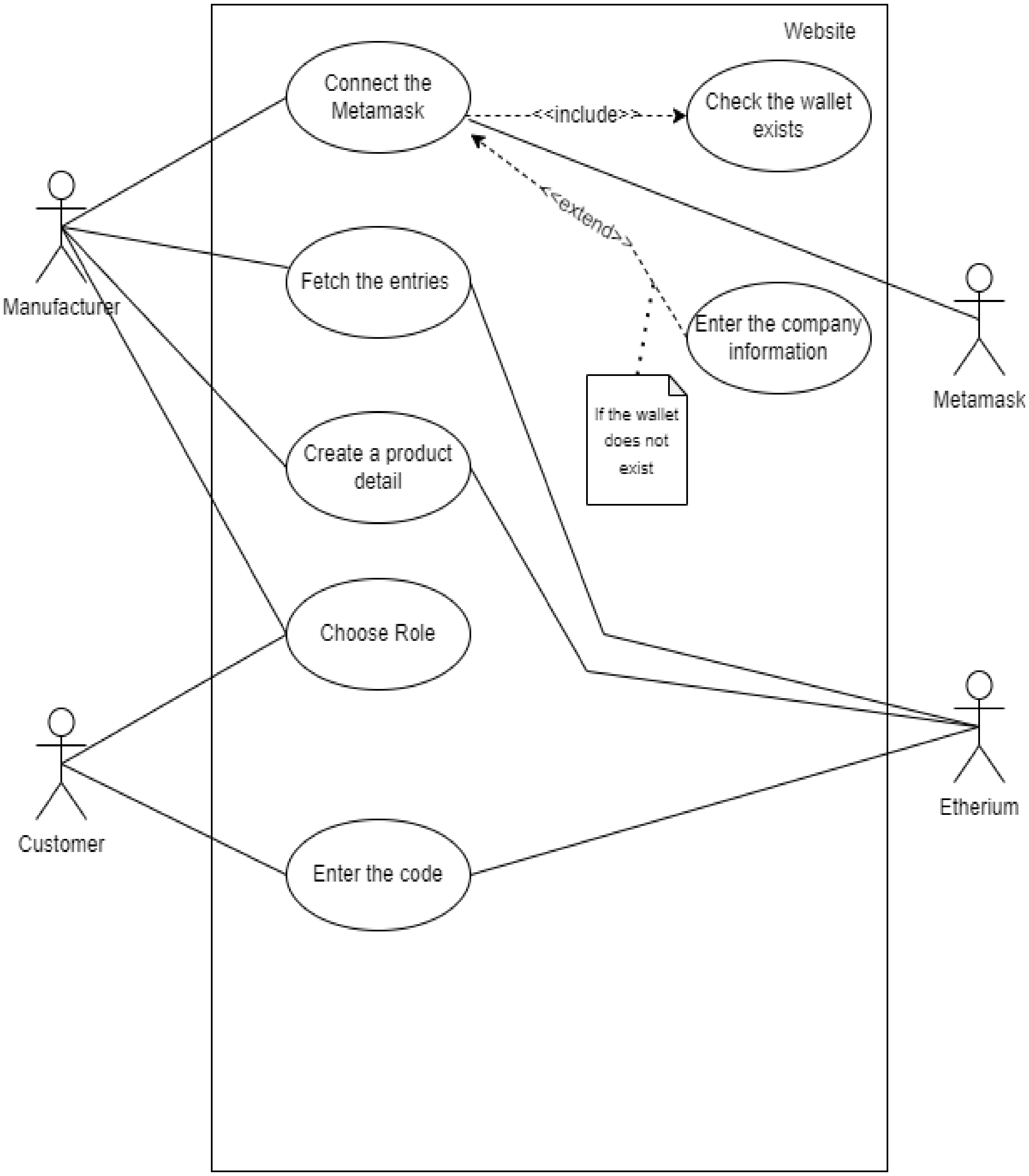
2.3 Use-Case Diagram

There are 2 main actors and 2 actors to interact with. Manufacturer is an actor who wants to enter product details and wants become more trusted company about product originality.

Customer is an actor who wants to check product is original or counterfeit.

MetaMask provides the simplest yet most secure way to connect to blockchain-based applications. You are always in control when interacting on the new decentralized web.MetaMask generates passwords and keys on your device, so only you have access to your accounts and data. You always choose what to share and what to keep private. Available as a browser extension and as a mobile app, MetaMask equips you with a key vault, secure login, token wallet, and token exchange—everything you need to manage your digital assets. [4]

Ethereum is a platform powered by blockchain technology that is best known for its native cryptocurrency, called Ether, or ETH, or simply Ethereum. The distributed nature of blockchain technology is what makes the Ethereum platform secure, and that security enables ETH to accrue value.The Ethereum platform supports Ether in addition to a network of decentralized apps, otherwise known as dApps. Smart contracts, which originated on the Ethereum platform, are a central component of how the platform operates. Many decentralized finance (DeFi) and other applications use smart contracts in conjunction with blockchain technology. [5]



2.4 Use-Case Definitions and Scenarios

The scenarios and use case definitions will be faced by the user are as follows :

Use Case Name: Choose Role
Actor: Manufacturer and Customer
The user sees 2 options to choose a role. Customer or Manufacturer. After choosing a role, it is directed to the relevant page.

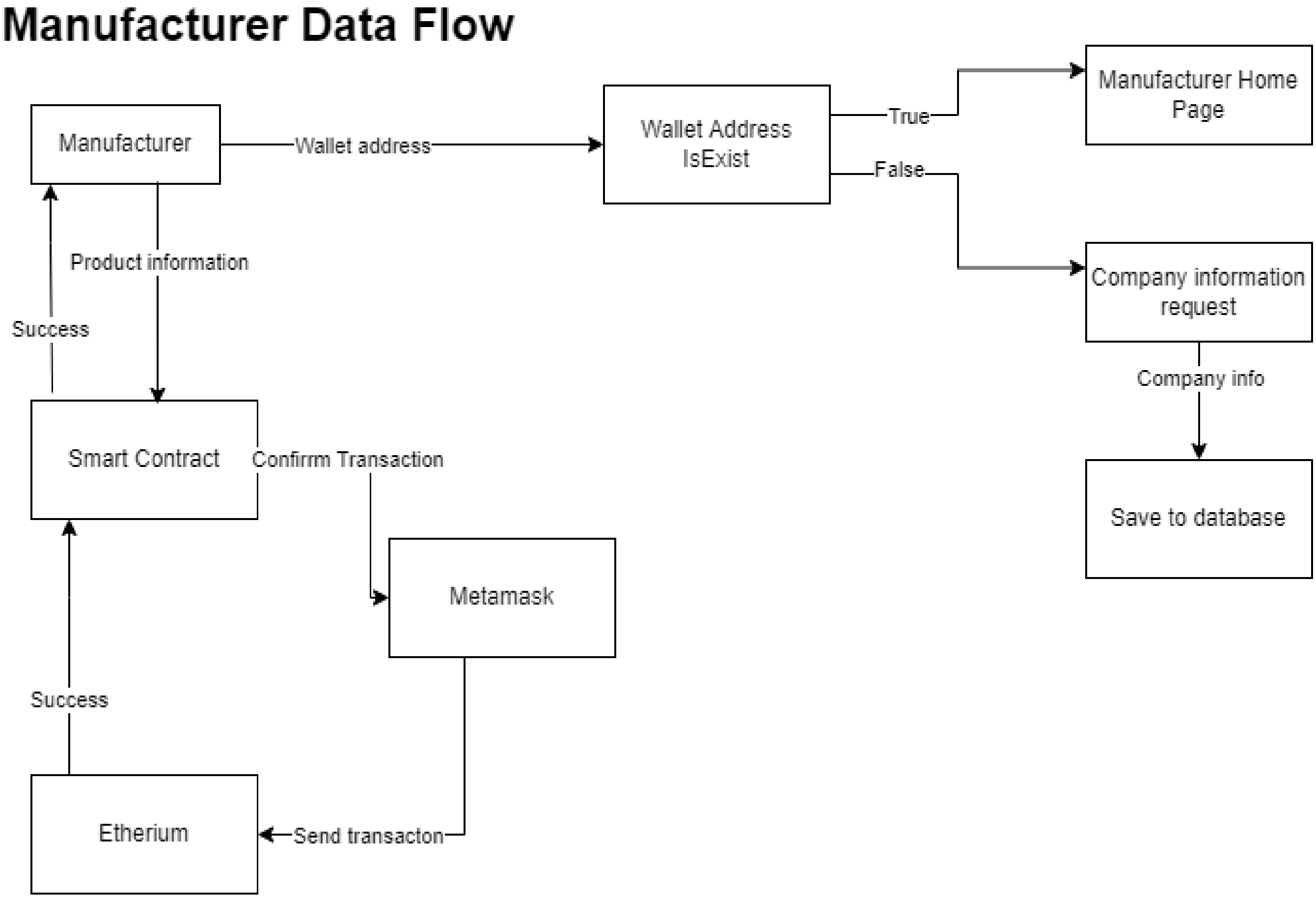
Use Case Name: Enter the Code
Actor: Customer
After select the role customer enters the code and displays the product information he/she checked.

Use Case Name: Connect the Metamask
Actor: Manufacturer
Manufacturer needs connect website and digital wallet with MetaMask plugin. If the manufacturer is doing this for the first time, company information is requested.

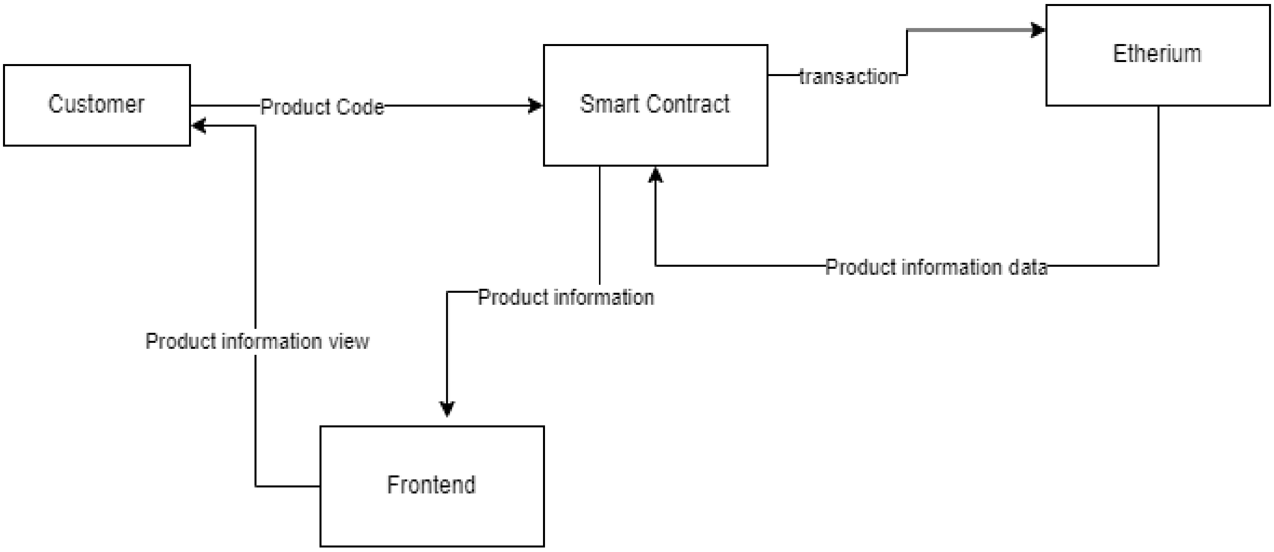
Use Case Name: Fetch the entries
Actor: Manufacturer
After the manufacturer is connected/login. He/She can list the records that entered before.

Use Case Name: Create a product detail.
Actor: Manufacturer
The manufacturer enters the details of the product on the web page and saves this data on the blockchain. During detailing, the person approves this transaction via Metamask. After the transaction is confirmed by the blockchain, the data is written to the blockchain.

2.5 Data Flow

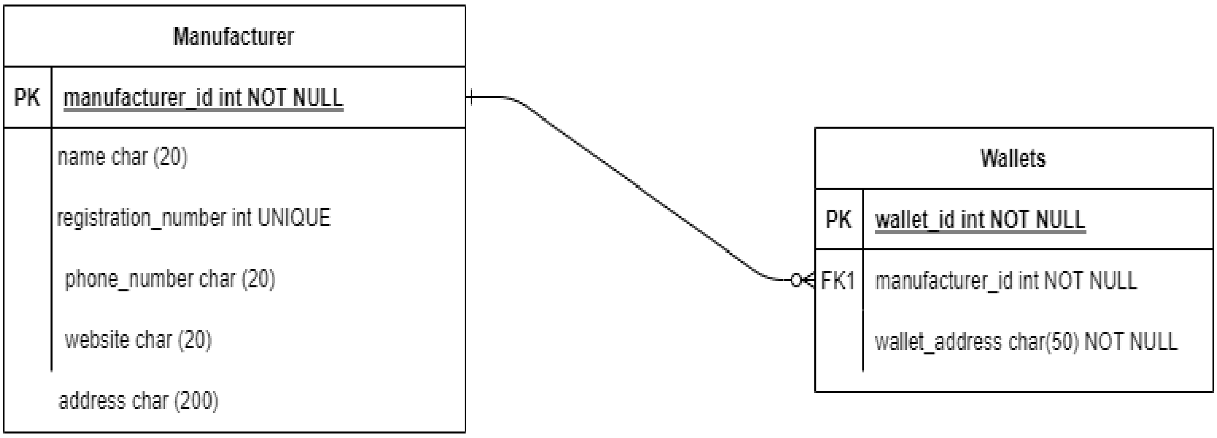


Customer Data Flow



3. Database Design

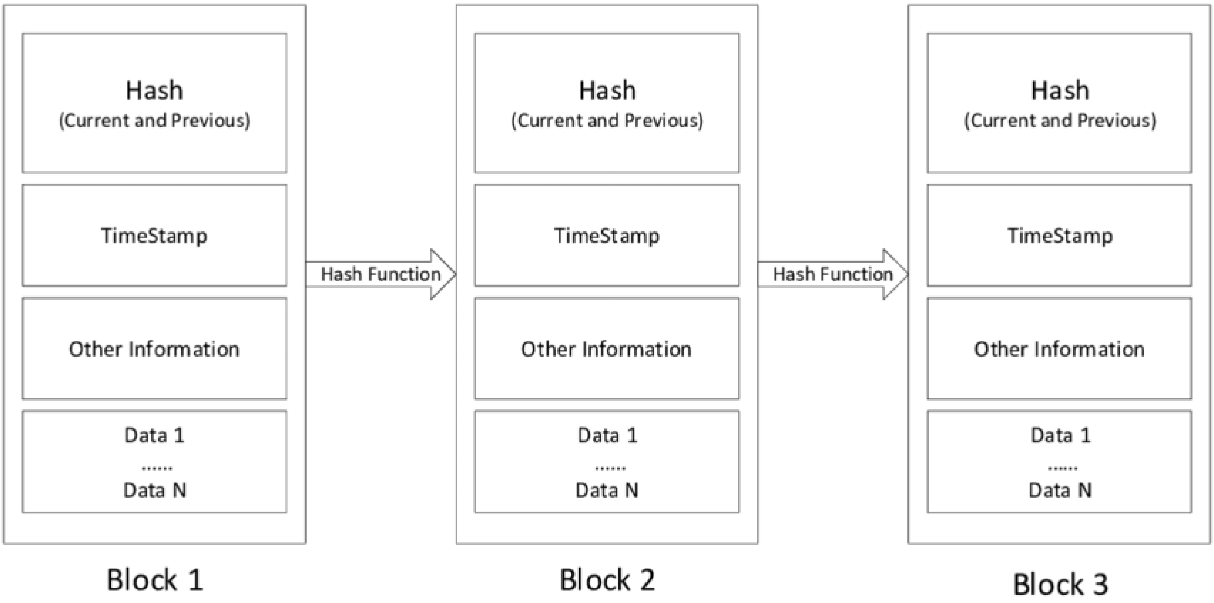
In this project, the main information is the detailed information of the products. However, due to the nature of the project, this information should not be kept in a central database. Blockchain is also used for this reason. We record other information that should not be kept centrally and should not be interfered with on the blockchain base. Blockchain actually serves as both an environment where our program runs and a database where our data is kept.



The characteristics of each record to be kept on the blockchain are as follows.

Registrant's name
Registrant's address
Registration number
Product name
Batch number of the product
Explanation
Registration date

In this way, which product this record belongs to and who made this record will be visible. Keeping this data in a central database (database running on a server) carries the possibility that this data can be manipulated by one person (admin) or more than one person. This undermines the reliability of the data. With Blockchain, every record creation transaction is transparent and traceable. It can be seen by anyone in the world. Therefore, the data becomes both transparent and reliable.



So what kind of structure does the blockchain have in itself? As the name suggests, a blockchain is a chain formed by the merging of blocks. Here, the structure called a block is a structure that has its own specific features and contains the data recorded in it. It contains timestamp, informations and data. What makes the block unique is the hash value in it. This hash value is created from the information in the block and the hash information of the previous block. This is the hash formation in the part that connects the blocks. If the information in the block is changed, the hash of the block will change. The hash information of other blocks connected with this block becomes incompatible with the previous one and the chain is broken. In short, if the information of the blocks were changeable, there would be no blockchain. Therefore, each block is an unalterable whole.

References

[1] Forget Bitcoin: Blockchain is the Future By NATHAN REIFF Updated July 26, 2021
Reviewed by ERIKA RASURE
https://www.investopedia.com/tech/forget-bitcoin-blockchain-future

[2] Blockchain Explaned By ADAM HAYES Updated January 07, 2022
Reviewed by JULIUS MANSA Fact checked by SUZANNE KVILHAUG
https://www.investopedia.com/terms/b/blockchain.asp

[3] web3.js - Ethereum JavaScript API Documentation
https://web3js.readthedocs.io/en/v1.5.2/

[4] A crypto wallet & gateway to blockchain apps
https://metamask.io

[5] What is Ethereum By JAKE FRANKENFIELD Updated December 22, 2021
Reviewed by SOMER ANDERSON
https://www.investopedia.com/terms/e/ethereum.asp

https://ethereum.org/en/what-is-ethereum/

[6] What is Solidity
https://docs.soliditylang.org/en/v0.8.11/

[7] Model, View, Controller MVC
https://en.wikipedia.org/wiki/Model%E2%80%93view%E2%80%93controller

[8] Dependency Injection
https://en.wikipedia.org/wiki/Dependency_injection