A Project Report

on

Fertilizer Company Authentication on Ethereum Platform

Submitted in partial fulfillment of the requirements

for the award of the degree of

BACHELOR OF TECHNOLOGY

in

Computer Science & Engineering

by

K. Venkata Nikhila	(174G1A05B0)
N. Tejaswini	(174G1A0599)
S. Vamsi Kasyap	(174G1A05A5)
K. Sai Narasimha Reddy	(174G1A0569)

Under the Guidance of

Dr. G. K. Venkata Narasimha Reddy M.Tech, Ph.D

Professor

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

SRINIVASA RAMANUJAN INSTITUTE OF TECHNOLOGY: ANANTAPURAMU
(Affiliated to JNTUA, Accredited by NAAC with 'A' Grade, Approved by AICTE, New Delhi & Accredited by NBA (EEE, ECE & CSE))

2020-2021

SRINIVASA RAMANUJAN INSTITUTE OF TECHNOLOGY: ANANTAPURAMU
(Affiliated to JNTUA, Accredited by NAAC with 'A' Grade, Approved by AICTE, New Delhi & Accredited by NBA (EEE, ECE & CSE))

Certificate

This is to certify that the project report entitled Fertilizer Company Authentication on Ethereum Platform is the bonafide work carried out by **K. Venkata Nikhila** bearing Roll Number 174G1A05B0**, N. Tejaswini** bearing Roll Number 174G1A0599**, S. Vamsi Kasyap** bearing Roll Number 174G1A05A5 and **K. Sai Narasimha Reddy** bearing Roll Number 174G1A0569 in partial fulfillment of the requirements for the award of the degree of **Bachelor of Technology** in **Computer Science & Engineering** during the academic year 2020-2021.

Guide	**Head of the Department**
Dr. G. K. V. Narasimha Reddy M.Tech, Ph.D Professor	Dr. G. K. V. Narasimha Reddy M.Tech, Ph.D Professor & HOD

Date:
Place: Ananthapuramu

EXTERNAL EXAMINER

ACKNOWLEDGEMENT

The satisfaction and euphoria that accompany the successful completion of any task would be incomplete without the mention of people who made it possible, whose constant guidance and encouragement crowned our efforts with success. It is a pleasant aspect that we have now the opportunity to express my gratitude for all of them.

It is with immense pleasure that we would like to express my indebted gratitude to my Guide **Dr. G. K. V. Narasimha Reddy M.Tech, Ph.D, Computer Science & Engineering**, who has guided me a lot and encouraged me in every step of the project work. We thank him for the stimulating guidance, constant encouragement and constructive criticism which have made possible to bring out this project work.

We express our deep-felt gratitude to **Dr. P. Chitralingappa M.Tech, Ph.D Associate Professor,** and **Mrs. M. Soumya M.Tech, Assistant Professor** project coordinator valuable guidance and unstinting encouragement enableus to accomplish our project successfully in time.

We are very much thankful to **Dr. G. K. V. Narasimha Reddy M.Tech, Ph.D, Professor & Head of the Department, Computer Science & Engineering,** for his kind supportand for providing necessary facilities to carry out the work.

We wish to convey my special thanks to **Dr. G. Bala Krishna Ph.D, Principal**of **Srinivasa Ramanujan Institute of Technology** for giving the required informationin doing our project work. Not to forget, we thank all other faculty and non-teaching staff, and my friends who had directly or indirectly helped and supported us in completing our project in time.

We also express our sincere thanks to the Management for providing excellent facilities**.**

Finally, we wish to convey our gratitude to our family who fostered all the requirements and facilities that we need.

Project Associates

Declaration

We, Ms. K. Venkata Nikhila with reg no: 174G1A05B0, Ms. N. Tejaswini with reg no: 174G1A0599, Mr. S. Vamsi Kasyap with reg no: 174G1A05A5, Mr. K. Sai Narasimha Reddy with reg no: 174G1A0569 students of SRINIVASA RAMANUJAN INSTITUTE OF TECHNOLOGY, Rotarypuram, hereby declare that the dissertation entitled "FERTILIZER COMPANY AUTHENTICATION ON ETHEREUM PLATFORM" embodies the report of our project work carried out by us during IV year Bachelor of Technology under the guidance of Dr. G. K. Venkata Narasimha Reddy M.Tech, Ph.D, Department of CSE, SRINIVASA RAMANUJAN INSTITUTE OF TECHNOLOGY, and this work has been submitted for the partial fulfilment of the requirements for the award of the Bachelor of Technology degree.

The results embodied in this project have not been submitted to any other University of Institute for the award of any Degree or Diploma

K. VENKATA NIKHILA	Reg no: 174G1A05B0
N. TEJASWINI	Reg no: 174G1A0599
S. VAMSI KASYAP	Reg no: 174G1A05A5
K. SAI NARASIMHA REDDY	Reg no: 174G1A0569

Contents

List of Figures

List of Screeens

List of Abbreviations

Dapp	Decentralized Application
EVM	Ethereum Virtual Machine
EOA	Externally Owned Accounts
PoW	Proof of Work
PoA	Proof of Authority
SRS	Software Requirement Specification
ERC-20	Ethereum Request for Comment
IDE	Integrated Development Environment
UML	Unified modelling language

ABSTRACT

Decentralized applications are the type of applications that are operated in the decentralized environment in the absence of the central server. Authentication of fertilizer companies is very much necessary in order to verify the companies which are selling genuine fertilizers.

We are using a decentralized application deployed on Ethereum public blockchain. A smart contract that was written in solidity language is deployed onto the internal node present in Rinkeby test network.

The node which deployed the smart contract had got access to enter and delete the fertilizer companies on to the blockchain, this functionality is enabled with the help of the user-defined modifier which was defined in smart contract rest of the nodes can just check whether the company from which they brought fertilizers is genuine or not.

The Application Binary Interface (ABI) can be used for interaction between smart contract and JavaScript.

CHAPTER 1

INTRODUCTION

The main purpose of Fertilizer Authentication is to make sure that the Authentic fertilizers are reached to the consumers. In order to achieve our objective, we are making use of Ethereum platform. In which each and every action performed is considered as a programmed transaction. We are using a decentralized application deployed on Ethereum public blockchain.

In order to verify the fertilizer companies which are selling genuine fertilizers, Authentication of fertilizer companies is implemented.

A smart contract that was written in solidity language is deployed onto the internal node present in Rinkeby test network. The node which deployed the smart contract had got access to enter and delete the fertilizer companies on to the blockchain. This functionality is enabled with the help of the user-defined modifier which was defined in smart contract rest of the nodes can just check whether the company from which they brought fertilizers is genuine or not.

1.1. Objective:

Now a days the whole world is becoming digitalized. Due to this the technologies are being increased drastically. Also greater enhancement in the technologies at the same time fraud also increased very much. To ensure the soil fertility and crop production fertilizer is the key concern for the agriculture system. Different types of fertilizers are used by the farmers. The uses of fertilizer have increased due to the expansion of irrigation facilities and make the cultivation process productive with varieties of crop plantation.

Most commonly used fertilizers are MoP, TSP, and Urea, and these fertilizers are holding leading percentage of 70-75% while comparing with total fertilizer used by farmers. To fulfill the farmers demands of these fertilizers some of them are produced by the factories and also imported from other countries.

1.2 Blockchain Technology:

Blockchain is a shared, immutable ledger that facilitates the process of recording transactions and tracking assets in a business network. A Blockchain network can track orders, payments, accounts, production and much more.

1.2.1 Types of Blockchain:

1. **Public blockchain network:**

A public blockchain is one that anyone can join and participate in, such as Bitcoin. Drawbacks might include substantial computational power required, little or no privacy for transactions, and weak security. These are important considerations for enterprise use cases of blockchain.

2. **Private blockchain network:**

A private blockchain network, similar to a public blockchain network, is a decentralized peer-to-peer network. However, one organization governs the network, controlling who is allowed to participate, execute a consensus protocol and maintain the shared ledger. A private blockchain can be run behind a corporate firewall and even be hosted on premises.

3. **Permissioned blockchain network:**

Business who set up a private blockchain will generally set up a permissioned blockchain network. It is important to note that public blockchain networks can also be permissioned. Participants need to obtain an invitation on permission to join.

4. **Consortium blockchains:**

Multiple organizations can share the responsibilities of maintaining a blockchain. A consortium blockchain is ideal for business when all participants need to be permissioned and have a shared responsibility for the blockchain.

1.3 What is Ethereum?

When we hear the word “Ethereum” we typically associate it with a cryptocurrency

– like Bitcoin. It’s important to understand that Ethereum is much more than just a simple cryptocurrency, it’s actually an open software platform built on blockchain technology that enables developers to build and deploy decentralized applications. Within the ethereum platform, is a cryptocurrency called ether that is used to power applications built on the Ethereum blockchain.

Ethereum is software running on a network of computers that ensures that data and small computer programs called smart contracts are replicated and processed on all the computers on the network, without a central coordinator. The vision is to create an unstoppable censorship resistant self-sustaining decentralized world computer. It extends the blockchain concepts from Bitcoin which validates, stores, and replicates transaction data on many computers around the world (hence the term ‘distributed ledger’). Ethereum takes this one step further, and also runs computer code equivalently on many computers around the world. What Bitcoin does for distributed data storage, Ethereum does for distributed data storage plus computations. The small computer programs being run are called smart contracts, and the contracts are run by participants on their machines using a sort of operating system called a “Ethereum Virtual Machine”.

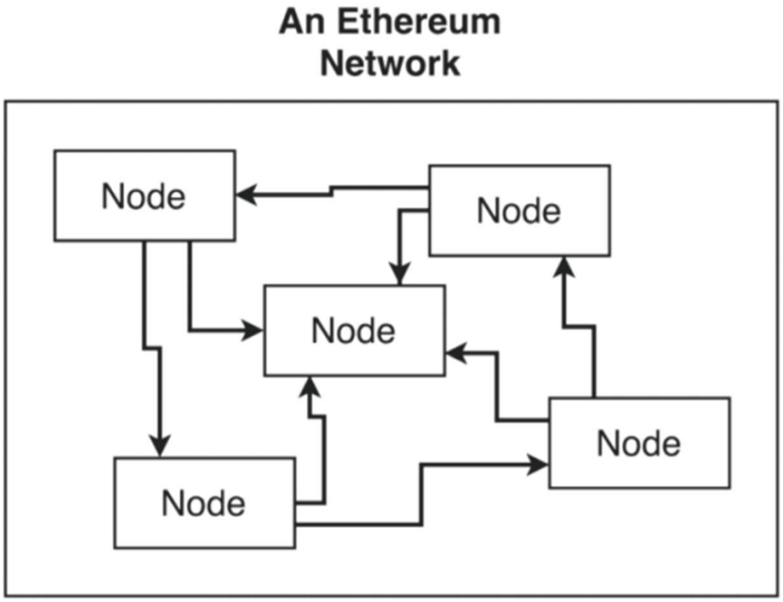


FIGURE-1: ETHEREUM NETWORK

A Ethereum network consists of more number of nodes. Here each node specifies a client system.

- Ethereum networks are used to transfer the money and store data.
- There are different types of Ethereum networks.
- Each Ethereum network is formed by one or more number of nodes.
- Each node is a machine running an Ethereum client.
- Anyone can run a node.
- Each node can contain a full copy of the blockchain.
- The ‘blockchain’ is a database that stores a record of every transaction that has ever Taken place.

1.3.1 DIFFERENT ETHEREUM NETWORKS:

1. ROPSTEN

- PoW testnet
- History:
  - Started in November 2016. Named after a subway station in Stockholm.
  - Was DoS attacked in February 2017 which made synching slow and made clients consume a lot of disk space.
  - Was revived in March 2017 and became usable again.
- Pros:
  - Best reproduces the current production environment, i.e. system and network conditions on the live Ethereum mainnet, because it's PoW net.
  - Can be used with both geth and parity.
  - Ether can be mined.

KOVAN

- PoA testnet started by the Parity team
- History:
  - Started in March 2017. Named after a subway station in Singapore.

- Pros:
    - Immune to spam attacks (as Ether supply is controlled by trusted parties)
- Cons:
    - Not supported in geth.
    - Doesn't fully reproduce the current production environment as it uses PoA.
    - Ether can't be mined.

RINKEBY

- PoA testnet started by the Ethereum team. Uses Clique PoA consensus protocol.
- History:
    - Started in April 2017. Named after a metro station in Stockholm.
- Pros:
    - Immune to spam attacks (as Ether supply is controlled by trusted parties)
- Cons:
    - Supported by geth only
    - Doesn't fully reproduce the current production environment as it uses PoA.
    - Ether can't be mined.

SOKOL

- PoA testnet started by the POA.network team
- History:
    - Started in December 2017. Named after a subway station in Moscow.
- Pros:
    - Immune to spam attacks (as Ether supply is controlled by trusted parties)
- Cons:

- Not supported in geth.
- Doesn't fully reproduce the current production environment as it uses PoA.
- Ether can't be mined

1.4 WHAT IS SMART-CONTRACT?

The term smart contract has been used over the years to describe a wide variety of different things. In the 1990s, cryptographer Nick Szabo coined the term and defined it as "a set of promises, specified in digital form, including protocols within which the parties perform on the other promises." Since then, the concept of smart contracts has evolved, especially after the introduction of decentralized blockchain platforms with the invention of Bitcoin in 2009.

Smart contracts are typically written in a high-level language, such as Solidity. But in order to run, they must be compiled to the low-level bytecode that runs in the EVM. Once compiled, they are deployed on the Ethereum platform using a special contract creation transaction, which is identified as such by being sent to the special contract creation address. Each contract is identified by an Ethereum address, which is derived from the contract creation transaction as a function of the originating account and nonce. The Ethereum address of a contract can be used in a transaction as the recipient, sending funds to the contract or calling one of the contract's functions.

All smart contracts in Ethereum are executed, ultimately, because of a transaction initiated from an EOA. A contract can call another contract that can call another contract, and so on, but the first contract in such a chain of execution will always have been called by a transaction from an EOA. Contracts never run "on their own" or "in the background." Contracts effectively lie dormant until a transaction triggers execution, either directly or indirectly as part of a chain of contract calls.

1.5 ETHEREUM TECHNOLOGY:

DECENTRALIZATION:

- Fault tolerance— decentralized systems are less likely to fail accidentally because they rely on many separate components that are not likely.

• Attack resistance— decentralized systems are more expensive to attack and destroy or manipulate because they lack sensitive central points that can be attacked at much lower cost than the economic size of the surrounding system.

PEER TO PEER: In peer-to-peer systems, each peer or client can directly send and receive data directly from any other client. Each of the peers in the network behaves both as a server and as a client. Each node can request data to be saved in some nodes database, or it can request to read from a node's database. In Ethereum, all nodes can request from another node some information, about Ethereum's current state (smart contract, account balance, last block etc.). The node knows that some information is correct and valid because it can be verified it with Ethereum's consensus protocol.

PROOF OF WORK: Proof of work is a protocol that has the main goal of deterring cyber-attacks such as a distributed denial-of-service attack (DDoS) which has the purpose of exhausting the resources of a computer system by sending multiple fake requests.

CHAPTER 2

LITERATURE SURVEY

2.1 Existing System:

In traditional Fertilizer Company Authentication system there is a chance of manipulating the database which contain Authentic Fertilizer Companies.

The manipulation of the database which contains Authentic Fertilizer Companies without tracking the activity happening during manipulation will in turn deteriorate the trust on the system.

The proof of existence of certain value at certain point time can not be proved in the traditional system.

2.2 Proposed System:

The Big difference that our Fertilizer Authentication Dapp can bring is it will take the transparency to the next level. The each and every operation performed in Dapp will be called as "TRANSACTION" and this transaction will get recorded. These transactions will gets stored on Ethereum Blockchain and if we store data on blockchain it cannot be manipulated at any cost. If we want to remove the data the data on Blockchain it can be done by "COMPENSATING TRANSACTION**".**

CHAPTER 3

ANALYSIS

3.1 Introduction

The Analysis Phase is where the project life cycle begins. This is the phase where you break down the deliverables in the high-level Project Charter into the more detailed business requirements. Gathering requirements is the main attraction of the Analysis Phase. The process of gathering requirements is usually more than simply asking the users what they need and writing their answers down. Depending on the complexity of the application, the process for gathering requirements has a clearly defined process of itsown. This process consists of a group of repeatable processes that utilize certain techniques to capture, document, communicate, and manage requirements. This formal process, which will be developed in more detail, consists of four basic steps.

1. **Elicitation** – I ask questions, you talk, I listen
2. **Validation** – I analyze, I ask follow-up questions
3. **Specification** – I document, I ask follow-up questions
4. **Verification** – We all agree

Most of the work in the Analysis Phase is performed by the role of analyst.

3.2 Software Requirement Specification

SRS is a document created by system analyst after the requirements are collected. SRS defines how the intended software will interact with hardware, external interfaces, speed of operation, response time of system, portability of software across various platforms, maintainability, speed of recovery after crashing, Security, Quality, Limitations etc.

The requirements received from client are written in natural language. It is the responsibility of system analyst to document the requirements in technical language so that they can be comprehended and useful by the software development team.

3.3 Hardware Requirements

Any Contemporary PC.

3.4 Software Requirements

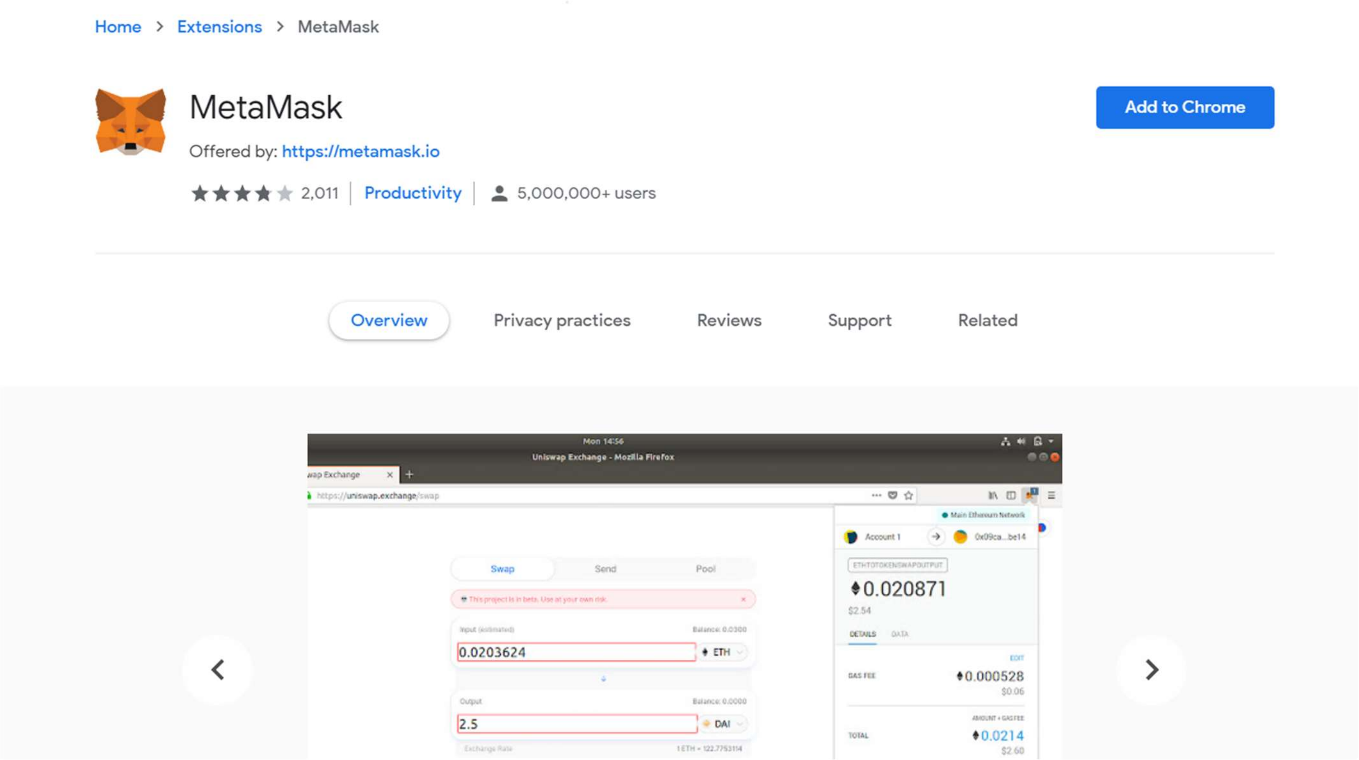
➢ Operating System	**:**	Windows 10
➢ Tools	**:**	MetaMask extension
➢ IDE	:	Remix-Ethereum IDE
➢ Languages Used	**:**	Solidity

3.4.1 MetaMask extension:

MetaMask is a browser plugin that serves as an Ethereum wallet. Users can store Ether and other ERC-20 tokens in the metamask wallet. The wallet can also be used to interact with decentralized applications, or dapps.

Steps:

1. Open Google Chrome and type MetaMask extension. Click on MetaMask for Chrome webstore. A new window will open that contains MetaMask and click on add to chrome option on the top-right of the window as shown below.



Home > Extensions > MetaMask

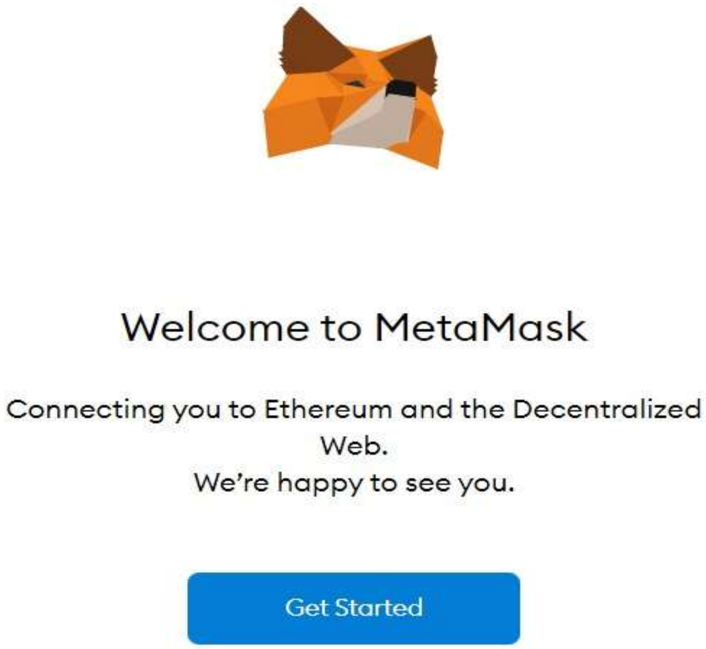
MetaMask

Offered by: https://metamask.io

★★★★☆ 2,011 | Productivity | 5,000,000+ users

Add to Chrome

Overview Privacy practices Reviews Support Related

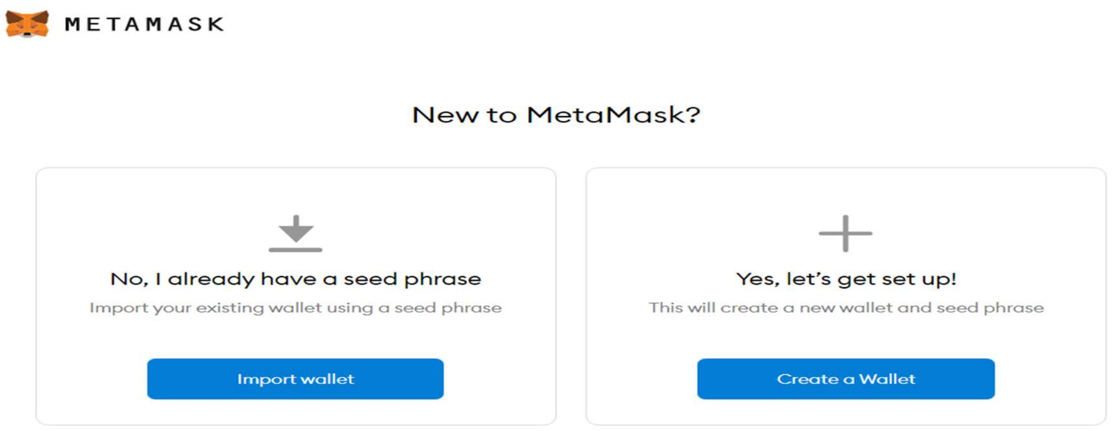


Welcome to MetaMask

Connecting you to Ethereum and the Decentralized Web.
We’re happy to see you.

Get Started

2. Now click on “Get Started”. Then select create a new wallet or import wallet.



New to MetaMask?

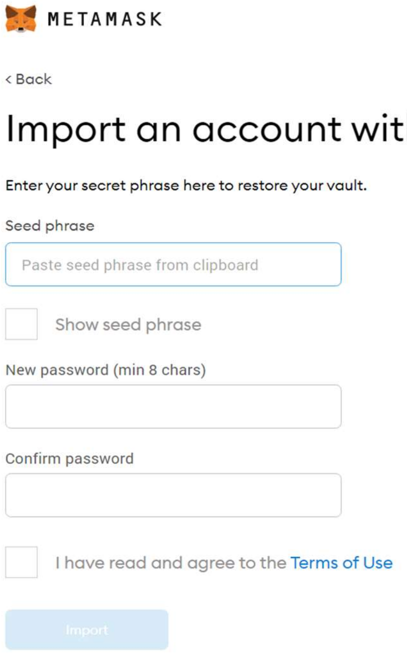
No, I already have a seed phrase

Import your existing wallet using a seed phrase

Yes, let’s get set up!

This will create a new wallet and seed phrase

3. Now add the details given and click on import.



Import an account with seed phrase

Enter your secret phrase here to restore your vault.

Seed phrase

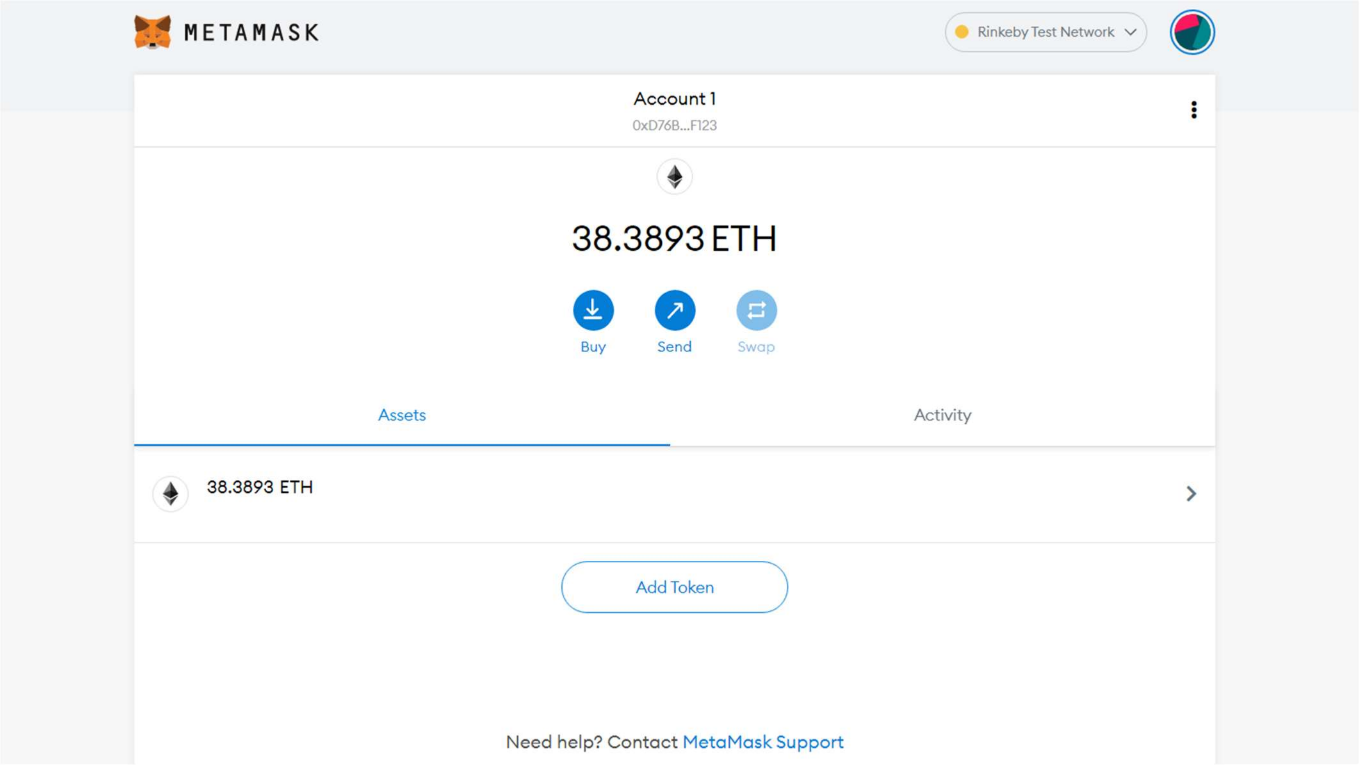
Paste seed phrase from clipboard

□ Show seed phrase

New password (min 8 chars)

Confirm password

□ I have read and agree to the Terms of Use



38.3893 ETH

38.3893 ETH

4. Hence the process is completed.

CHAPTER 4

DESIGN

4.1 UML Introduction:

The unified modeling language allows the software engineer to express an analysis model using the modeling notation that is governed by a set of syntactic, semantic and pragmatic rules. A UML system is represented using five different views that describe the system from distinctly different perspective.

UML is specifically constructed through two different domains, they are:

- UML Analysis modeling, this focuses on the user model and structural model views of the systems.
- UML Design modeling, which focuses on the behavioral modeling, implementation modeling and environmental model views.

4.1.1 Usage of UML in Project

As the strategic value of software increases for many companies, the industry looks for techniques to automate the production of software and to improve quality and reduce cost and time to the market. These techniques include component technology, visual programming, patterns and frameworks. Additionally, the development for the World Wide Web, while making some things simpler, has exacerbated these architectural problems. The UML was designed to respond to these needs. Simply, systems design refers to the process of defining the architecture, components, modules, interfaces and data for a system to satisfy specified requirements which can be done easily through UML diagrams.

4.2 Data Flow Diagram

A data-flow diagram is a way of representing a flow of a data of a process or a system (usually an information system). This also provides information about the outputs and inputs of each entity and the process itself. A data-flow diagram has no control flow, there are no decision rules and no loops. Specific operations based on the data can be represented by a flowchart.

The data-flow diagram is part of the structured-analysis modeling tools. When using UML, the activity diagram typically takes over the role of the data-flow diagram.

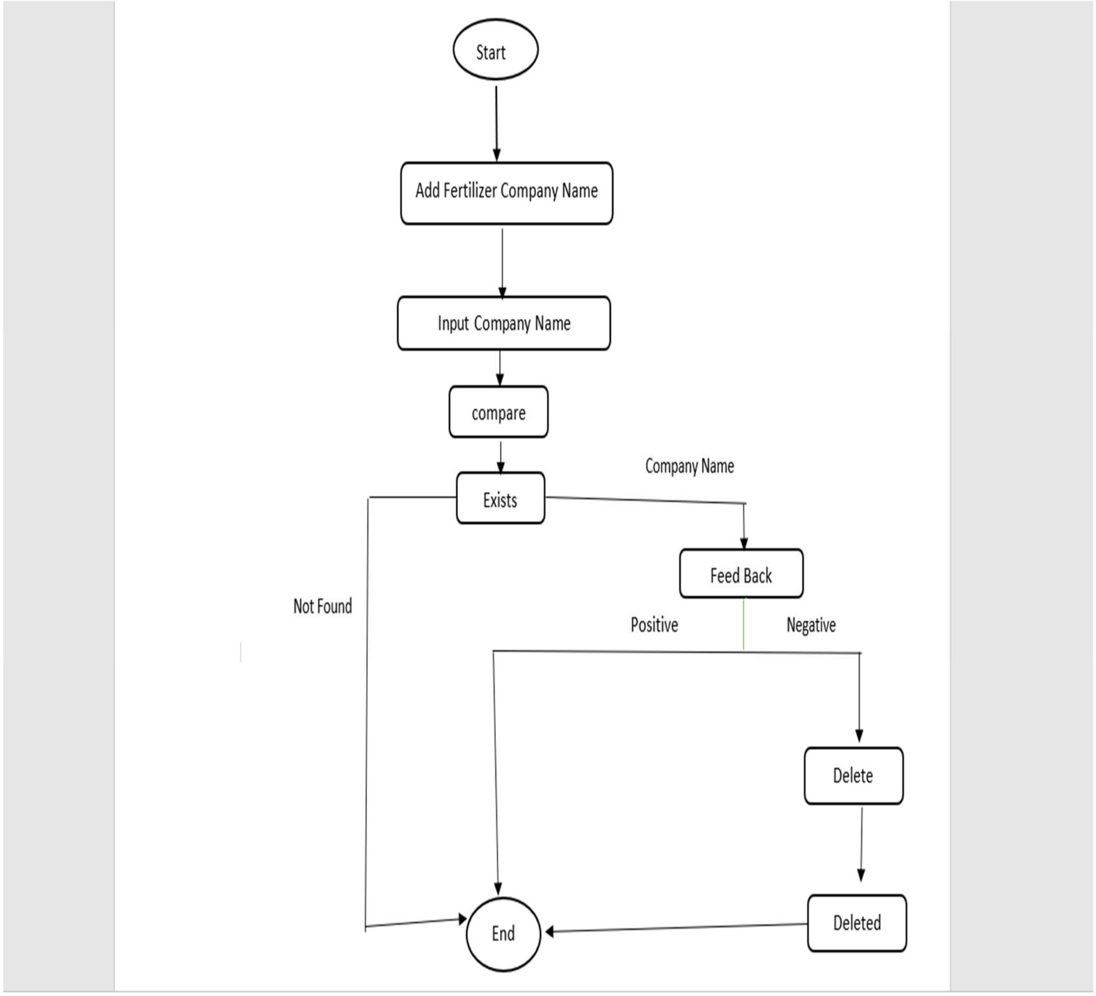


Fig 4.1: Data flow diagram of fertilizer company authentication

The above data flow diagram describes how the fertilizer company is being authenticated. So first we enter the fertilizer company names. Then we give a company name as input. Then we compare the given input with the fertilizer company names. If the name exists, then customer is able to enter the feedback of that company.

After submitting the feedback by the customer, depending on the feedback the owner is going to delete the fertilizer company name from the blockchain based on the limit of the number of negative feedbacks. And the status of the company is shown as deleted. If the fertilizer company name is not found then the process is stopped.

4.3 Steps involved in Design:

- Adding fertilizer company
- Input fertilizer company
- Compare
- Feedback
- Delete

Each step has its own specific reason and plays prominent role in building up a model of the project. Each step has been explained in detail in implementation part.

CHAPTER 5

IMPLEMENTATION

Here in our project Authentication of fertilizer companies happens by storing the names of the companies that can be trusted on the blockchain. The data that is stored on the blockchain can not be changed each and every change that is made will be recorded. So this will provides trust while authentication.

5.1 Libraries Used

Solidity is a high-level programming language designed for implementing smart contracts. It is statically-typed object-oriented(contract-oriented) language. Solidity is highly influenced by Python, C++, and JavaScript which runs on the Ethereum Virtual Machine (EVM). Solidity supports complex user-defined programming, libraries and inheritance. Solidity is primary language for blockchains running platforms.

Pragma:

Pragmas are instructions to the compiler on how to treat the code. All solidity source code should start with a “version pragma” which is a declaration of the version of the solidity compiler this code should use. This helps the code from being incompatible with the future versions of the compiler which may bring changes. The above-mentioned code states that it is compatible with compilers of version greater than and equal to 0.4.16 but less than version 0.7.0.

```
pragma solidity >=0.4.16 <0.7.0;
```

Contract:

The contract keyword declares a contract under which is the code encapsulated.

```
contract Test{
//Functions and Data
}
```

State variables:

State variables are permanently stored in contract storage that they are written in Ethereum Blockchain. The line *uint public var1* declares a state variable called var1 of type uint (unsigned integer of 256 bits). Think of it as adding a slot in a database. Similarly, goes with the declaration *uint public var2* and *uint public sum*.

```
uint public var1;
uint public var2;
uint public sum;
```

A function declaration:

This is a function named *set* of access modifier type *public* which takes a variable *x* and variable y of datatype *uint* as a parameter. Anyone can call the function set and overwrite the value of var1 and var2 which is stored in Ethereum blockchain. Function get will retrieve and print the value of the state variable sum. This was an example of a simple smart contract which updates the value of var1 and var2.

```
function set(uint x, uint y) public

function get() public view returns (uint)
```

5.2 Implementation:

Here we are using Remix-Ethereum IDE to compile and deploy the solidity code.

5.2.1 Implementing Libraries:

```
pragma solidity ^0.4.23;
```

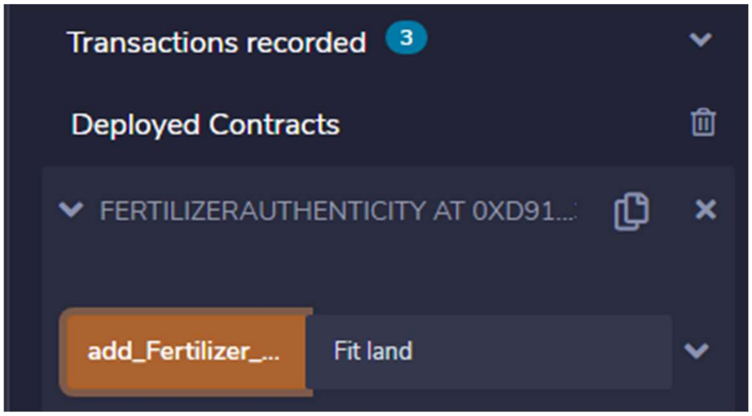
Input:

5.2.2 Register company:

Here we are entering or adding the list of fertilizer company names in to the blockchain. And the entering of company names can be done only by the owner of the block.

Code:

```
function add_Fertilizer_Company(string Company_Name) public OnlyOwner{

      Fertilizer_Companies.push(string(Company_Name));
}
```



Fit land

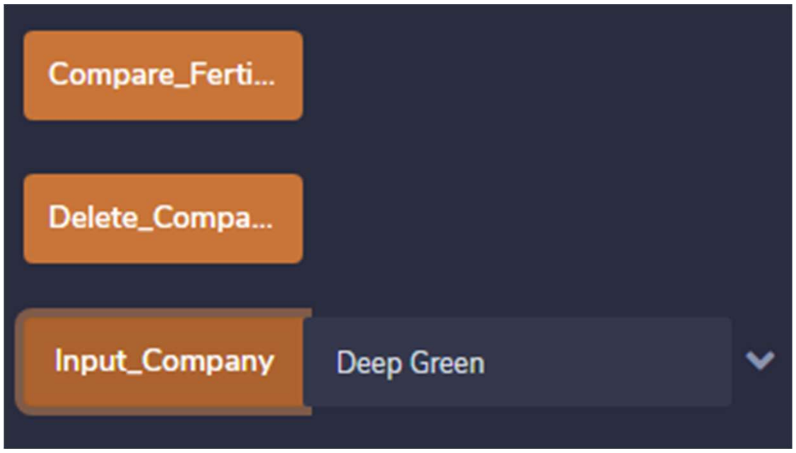
Fig.5.2.1. Register fertilizer company

5.2.3 Entering input company:

In this the fertilizer company name is entered as an input string, where any fertilizer company name can be entered.

Code:

```
function Input_Company(string Company) public{

Fertilizer_Name = Company;
}
```



Deep Green

Fig5.2.2. Entering input_company

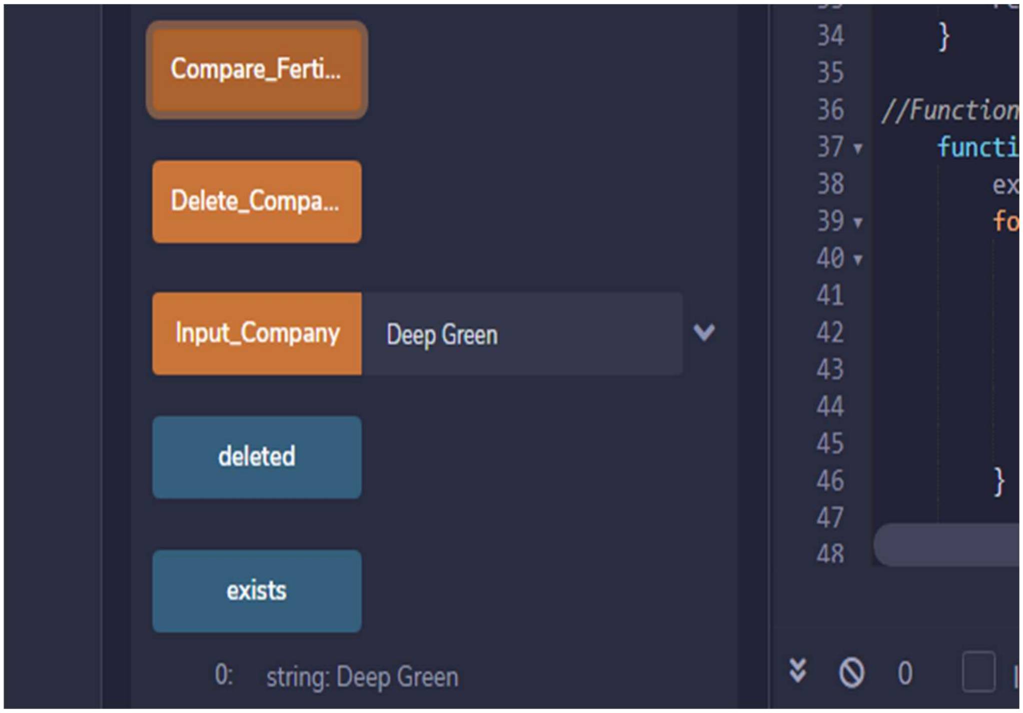
5.2.4 Comparing the input:

In this the comparison is done between the given input with the list of existed fertilizer companies. If the input company matches with the existed one then it shows the company name. If the fertilizer company name is not matched then it shows that the company is “does not exists”.

Code:

```
function Compare_Fertilizer_Companies() public{
      exists = ":does not exists";
      for(uint i=0;i<Fertilizer_Companies.length;i++){
         if(keccak256(abi.encode(Fertilizer_Companies[i])) ==
keccak256(abi.encode(Fertilizer_Name))){

            exists = Fertilizer_Companies[i];
            break;
         }
      }

   }
```

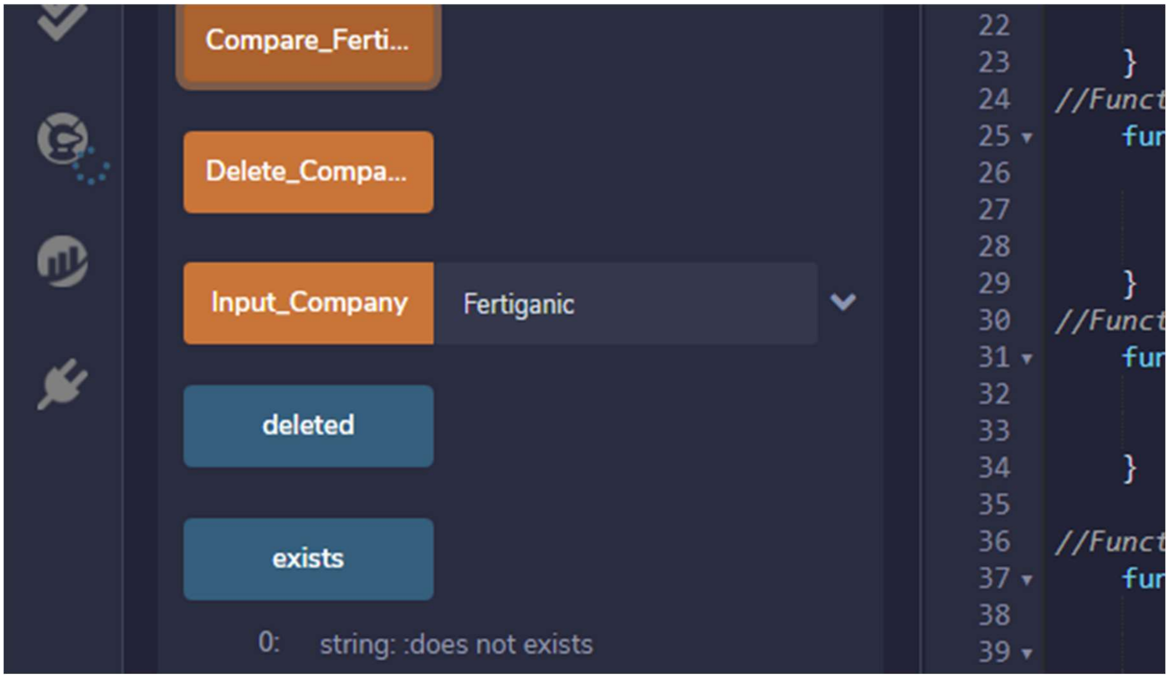
If the fertilizer company name exists:



```
0:   string: Deep Green
```

Fig.5.2.3. If the fertilizer company name exists

If the fertilizer company does not exists**:**



Input_Company Fertiganic

```
0:   string: :does not exists
```

Fig.5.2.4. If the fertilizer company does not exists

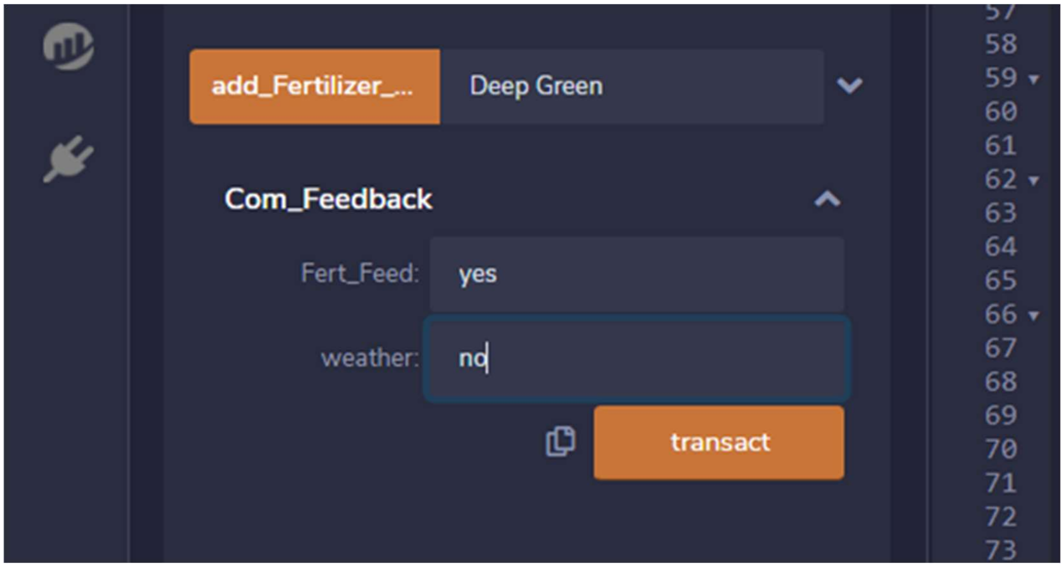
5.2.5 Feedback:

If the given input company name exists then the feedback is taken from the customer based on two factors i.e., weather and effect of fertilizer. So the conditions that are to be considered are as follows:

- If the weather condition is not good and the fertilizer effect is also not good then the feedback is written as “no”.
- If the weather condition is good and the fertilizer effect is not good then the feedback is written as “no”.
- If the weather condition is not good and the fertilizer effect is good then the feedback is written as “yes”.
- If the weather condition is good and the fertilizer effect is good then the feedback is written as “yes”.

- **Code:**

```
function Com_Feedback( string Fert_Feed,string weather) public{
        if(keccak256(abi.encode(Fert_Feed)) == keccak256(abi.encode("no")) &&
keccak256(abi.encode(weather)) == keccak256(abi.encode("no")) ){
            feed = "no";
        }
        if(keccak256(abi.encode(Fert_Feed)) == keccak256(abi.encode("no")) &&
keccak256(abi.encode(weather)) == keccak256(abi.encode("yes")) ){
            feed = "no";
        }
        if(keccak256(abi.encode(Fert_Feed)) == keccak256(abi.encode("yes")) &&
keccak256(abi.encode(weather)) == keccak256(abi.encode("yes")) ){
            feed = "yes";
        }
        if(keccak256(abi.encode(Fert_Feed)) == keccak256(abi.encode("yes")) &&
keccak256(abi.encode(weather)) == keccak256(abi.encode("no")) ){
            feed = "yes";
        }

        if(keccak256(abi.encode(feed)) == keccak256(abi.encode("no")) &&
keccak256(abi.encode(exists)) == keccak256(abi.encode( Fertilizer_Name))){
            feedback.push(Fertilizer_Name);

        }
    }

    mapping(string => uint) mapper;
```



add_Fertilizer_... Deep Green

Com_Feedback

Fert_Feed: yes

weather: no

Fig.5.2.5. Entering feedback

5.2.6 Deleting the company:

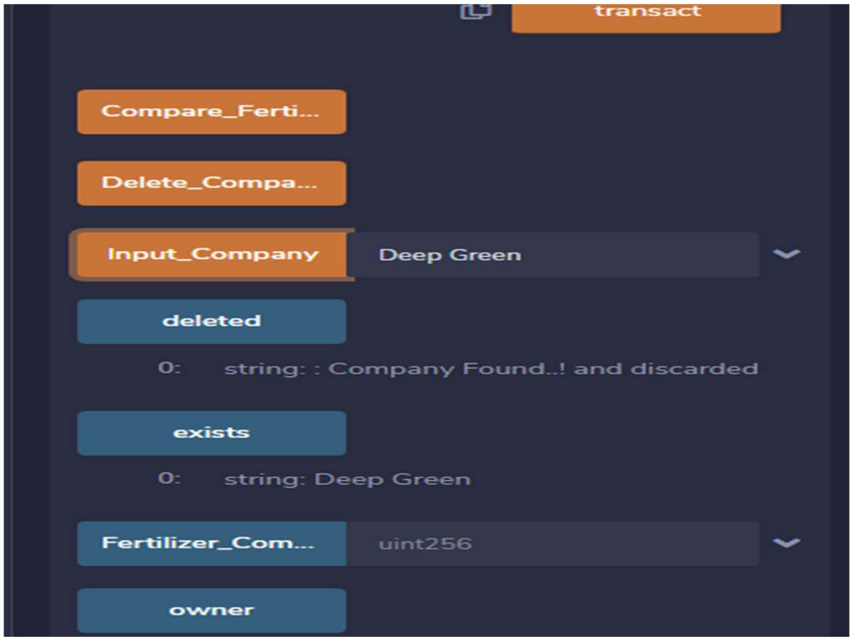
Deleting the fertilizer company name can be possible by the address which belongs to the owner who deployed smart contract on to the blockchain. The deletion operation is performed on the basis of the frequency of companies that were present in the feedback array. Here we have mapper mapping which contains the key as string and value as the frequency of each company present in the feedback array. The company and its respective frequency are present as key and value pairs in the mapper of type mapping.

The condition for the deletion operation is the frequency from the mapper mapping is greater than 2 then the company will be deleted from the array. Otherwise, company will still remain in the array.

Code:

```
function Delete_Company()public OnlyOwner  {

    deleted = 'sorry cannot be discarded';

    for(uint i=0; i<Fertilizer_Companies.length; i++) {
      uint count = 0;

      for(uint j=0; j<feedback.length; j++) {

        if( keccak256(abi.encode(Fertilizer_Companies[i])) == keccak256(abi.encode(feedback[j])) ) {
          count += 1;
        }
      }
      mapper[Fertilizer_Companies[i]] = count;
    }
    for(uint x=0;x<Fertilizer_Companies.length;x++){
      if(mapper[Fertilizer_Companies[x]] > 1){
        Fertilizer_Companies[x] = Fertilizer_Companies[Fertilizer_Companies.length-1];
        Fertilizer_Companies.length--;
        deleted = ": Company Found..! and discarded";
        break;
      }
    }
  }
}
```

If the limit count is exceeded :



Input_Company: Deep Green

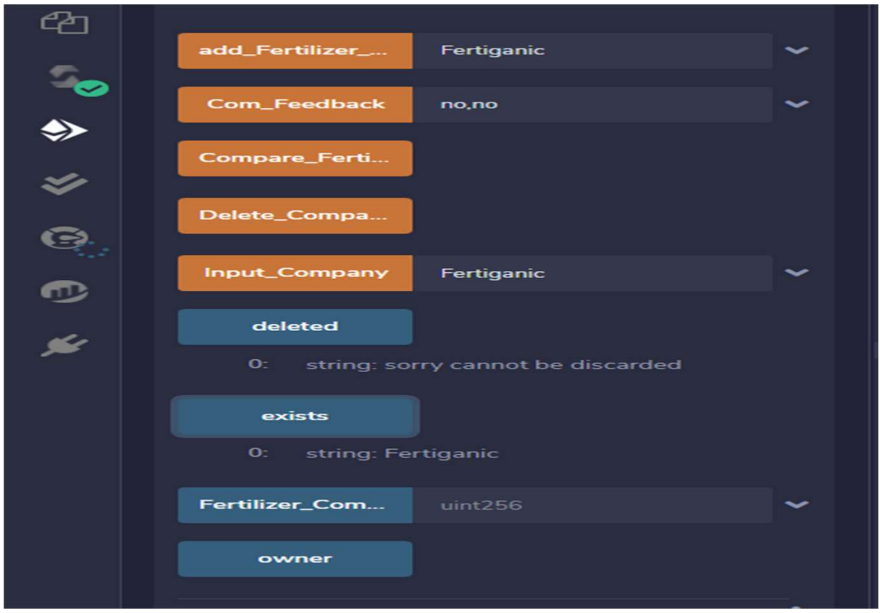
```
0:    string: : Company Found..! and discarded
```

```
0:    string: Deep Green
```

Fertilizer_Com...: uint256

Fig5.2.6. Deleting the fertilizer company

If the limit count is not exceeded:



add_Fertilizer_...: Fertiganic

Com_Feedback: no,no

Input_Company: Fertiganic

```
0:    string: sorry cannot be discarded
```

```
0:    string: Fertiganic
```

Fertilizer_Com...: uint256

Fig.5.2.7. If the limit count is not exceeded

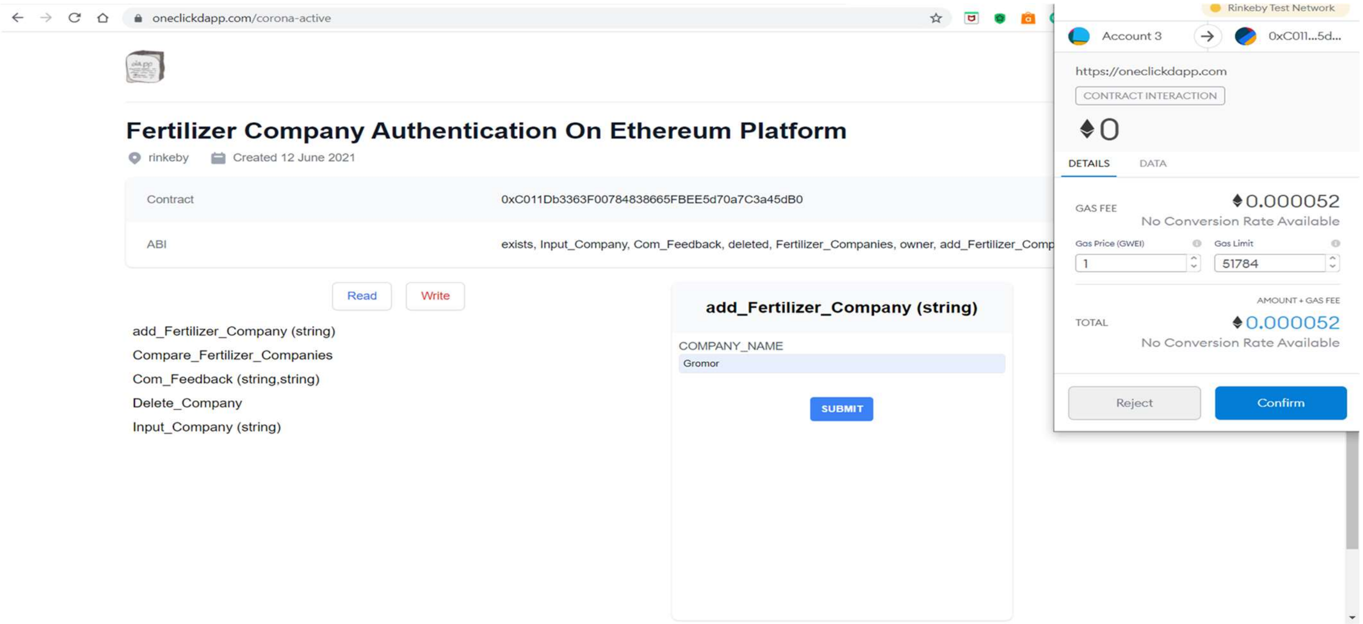
5.3 One Click Dapp Plugin:

Frontend:

Now we are connecting the above backend with the frontend by using “One Click Dapp Plugin”. The One Click Dapp is then linked with the MetaMask.

5.3.1 Registering Fertilizer Companies:

Now click on “Write” and select “add_fertilizer_company”. Enter the fertilizer company name and then click on “submit” button. We cannot add more than one at a time that is, after clicking on submit button only we can enter another fertilizer company name. And click on confirm in the MetaMask extension.



oneclickdapp.com/corona-active

Fertilizer Company Authentication On Ethereum Platform

rinkeby Created 12 June 2021

Contract 0xC011Db3363F00784838665FBEE5d70a7C3a45dB0

ABI exists, Input_Company, Com_Feedback, deleted, Fertilizer_Companies, owner, add_Fertilizer_Comp

add_Fertilizer_Company (string)
Compare_Fertilizer_Companies
Com_Feedback (string,string)
Delete_Company
Input_Company (string)

add_Fertilizer_Company (string)

COMPANY_NAME: Gromor

Rinkeby Test Network

Account 3 → 0xC011...5d...

https://oneclickdapp.com

CONTRACT INTERACTION

0

GAS FEE 0.000052 No Conversion Rate Available

Gas Price (GWEI): 1

Gas Limit: 51784

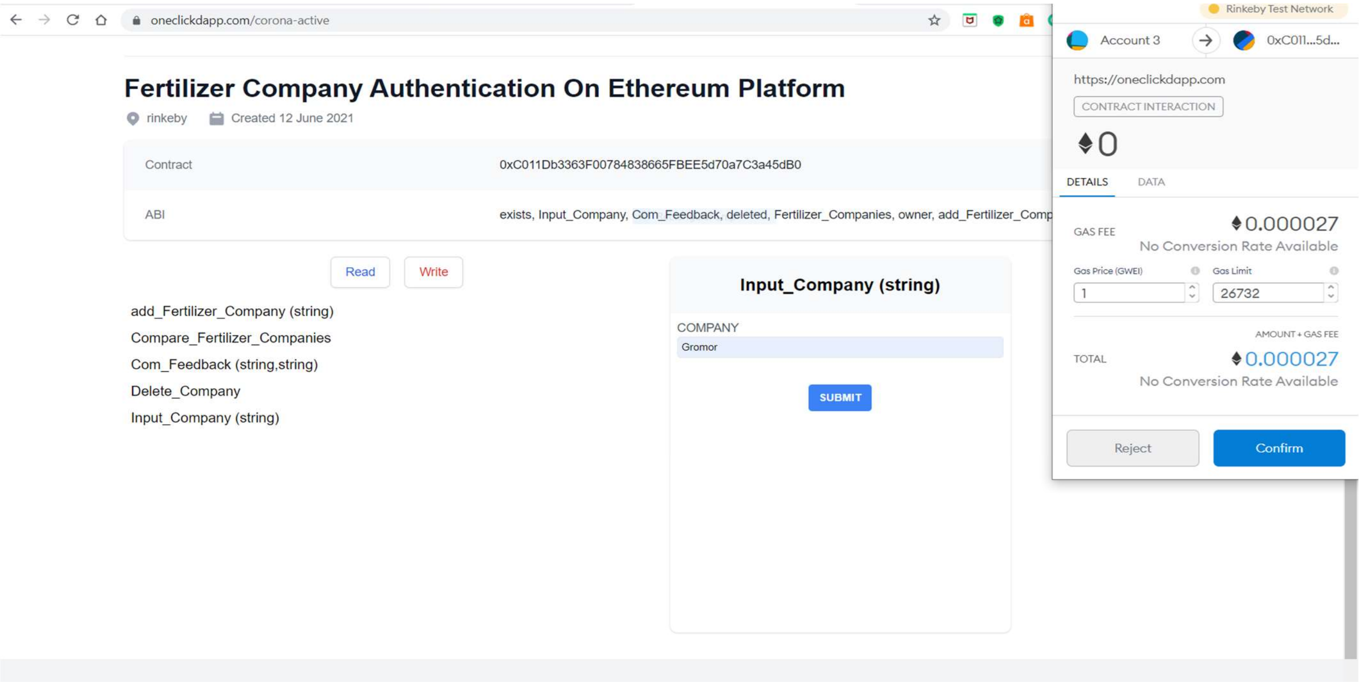
AMOUNT + GAS FEE

TOTAL 0.000052 No Conversion Rate Available

Fig.5.3.1. Registering fertilizer company

5.3.2 Giving Input for Authentication:

Now click on “Input_Company” and enter the input string in the text box and click on “submit” button and click on confirm in the MetaMask.



Fertilizer Company Authentication On Ethereum Platform

rinkeby Created 12 June 2021

Contract	0xC011Db3363F00784838665FBEE5d70a7C3a45dB0
ABI	exists, Input_Company, Com_Feedback, deleted, Fertilizer_Companies, owner, add_Fertilizer_Comp

add_Fertilizer_Company (string)
Compare_Fertilizer_Companies
Com_Feedback (string,string)
Delete_Company
Input_Company (string)

Input_Company (string)

COMPANY: Gromor

Rinkeby Test Network

Account 3 → 0xC011...5d...

https://oneclickdapp.com

CONTRACT INTERACTION

♦0

GAS FEE ♦0.000027 No Conversion Rate Available

Gas Price (GWEI): 1 Gas Limit: 26732

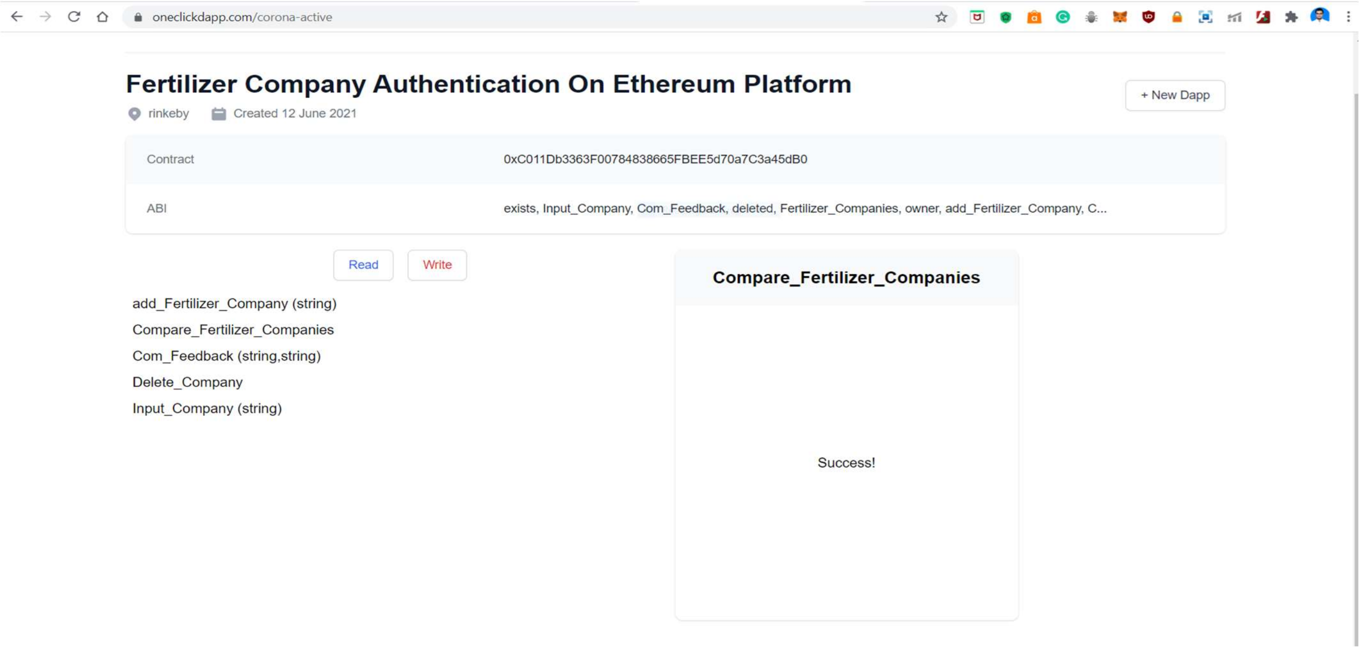
AMOUNT + GAS FEE

TOTAL ♦0.000027 No Conversion Rate Available

Fig.5.3.2. Giving input for authentication

5.3.3 Authenticating Fertilizer Companies:

The entered input_company name is then compared with the existing fertilizer companies.



Fertilizer Company Authentication On Ethereum Platform

rinkeby Created 12 June 2021

Contract	0xC011Db3363F00784838665FBEE5d70a7C3a45dB0
ABI	exists, Input_Company, Com_Feedback, deleted, Fertilizer_Companies, owner, add_Fertilizer_Company, C...

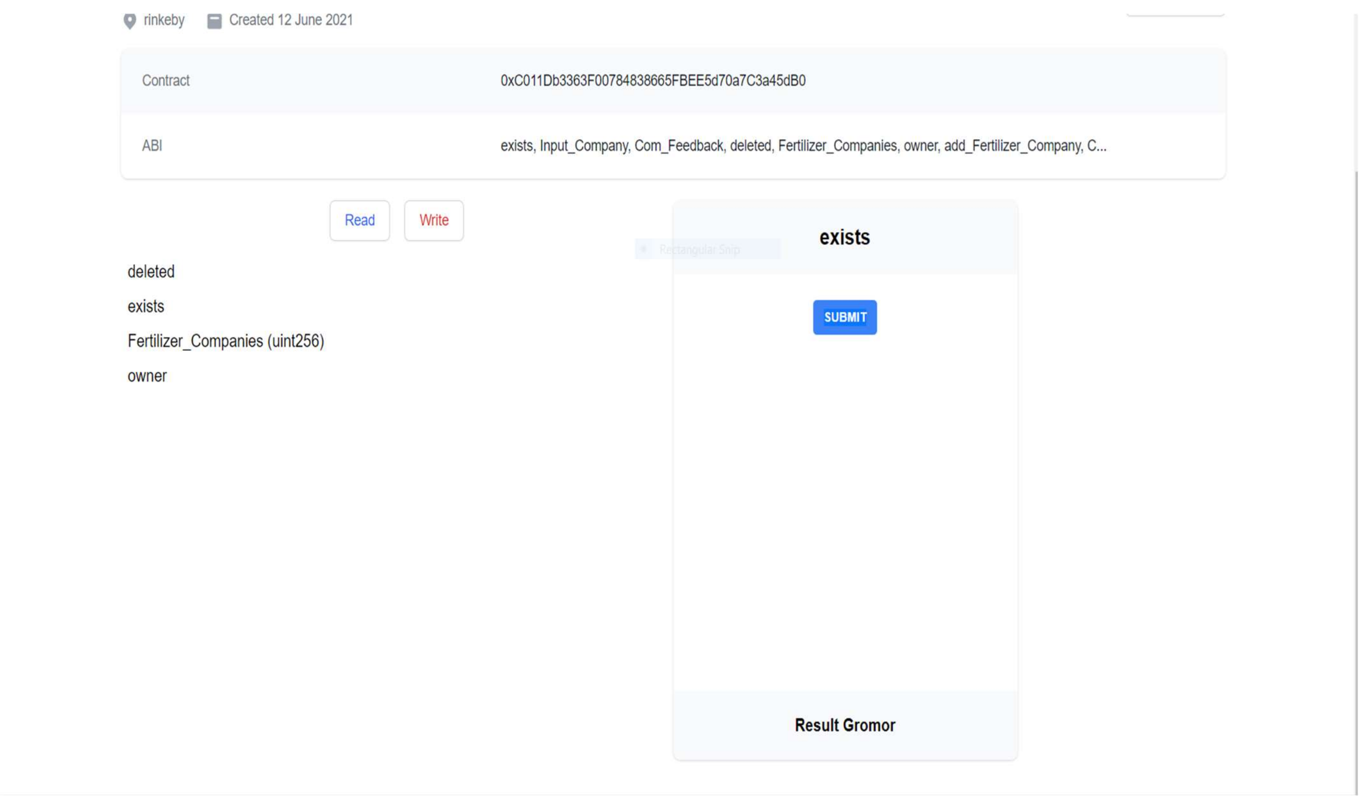
add_Fertilizer_Company (string)
Compare_Fertilizer_Companies
Com_Feedback (string,string)
Delete_Company
Input_Company (string)

Compare_Fertilizer_Companies

Success!

Fig.5.3.3.1. Authenticating fertilizer company

So first open One Click Dapp and login to that and connect to the MetaMask. If the company name is exists then by clicking on the exists “submit” button it displays the “Success!” and also the company name is shown at the bottom in the “Result”. If it does not exists then the “Not found” command is displayed.



rinkeby Created 12 June 2021

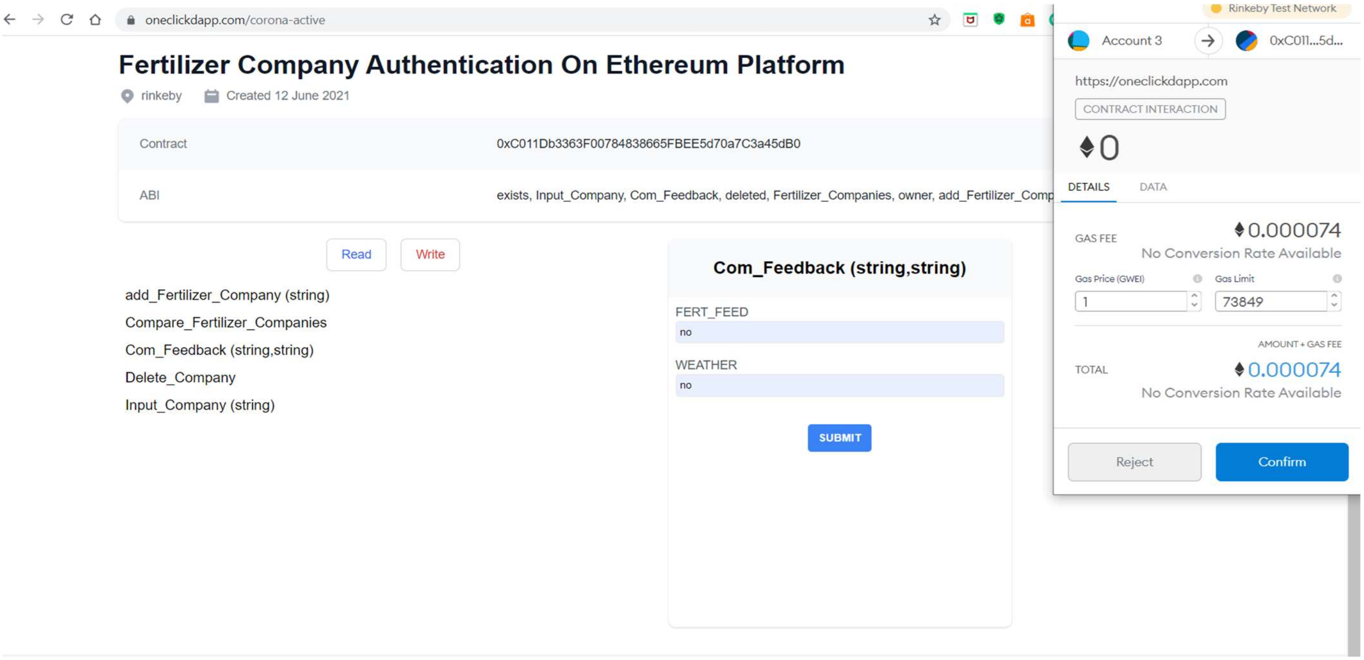
Contract	0xC011Db3363F00784838665FBEE5d70a7C3a45dB0
ABI	exists, Input_Company, Com_Feedback, deleted, Fertilizer_Companies, owner, add_Fertilizer_Company, C...

Result Gromor

Fig.5.3.3.2. Authenticating fertilizer company

5.3.4 Giving feedback:

Feedback is given by selecting the “Com_Feedback” option. Now the feedback is given based in the conditions of the fertilizer and weather and then submit the feedback by clicking on “submit button.



oneclickdapp.com/corona-active

Fertilizer Company Authentication On Ethereum Platform

rinkeby Created 12 June 2021

Contract 0xC011Db3363F00784838665FBEE5d70a7C3a45dB0

ABI exists, Input_Company, Com_Feedback, deleted, Fertilizer_Companies, owner, add_Fertilizer_Comp

add_Fertilizer_Company (string)
Compare_Fertilizer_Companies
Com_Feedback (string,string)
Delete_Company
Input_Company (string)

Com_Feedback (string,string)

FERT_FEED: no

WEATHER: no

https://oneclickdapp.com

CONTRACT INTERACTION

0

GAS FEE 0.000074
No Conversion Rate Available

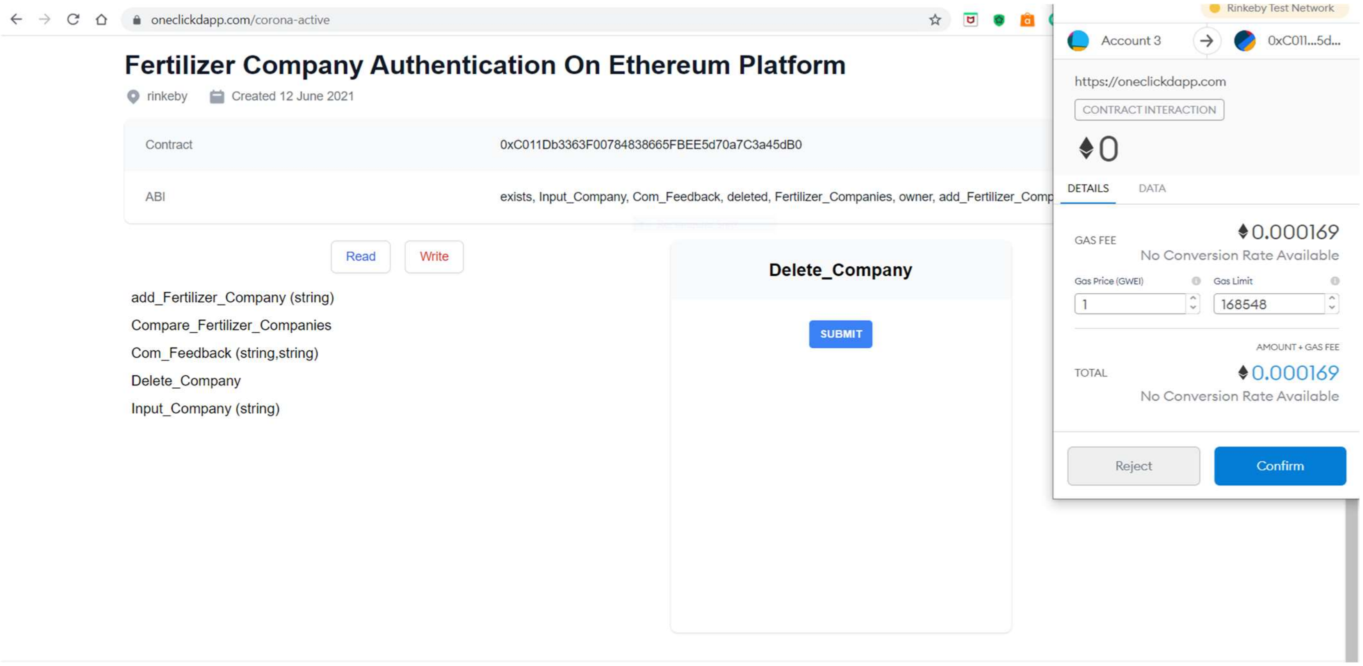
Gas Price (GWEI): 1 Gas Limit: 73849

AMOUNT + GAS FEE
TOTAL 0.000074
No Conversion Rate Available

Fig.5.3.4. Giving feedback

5.3.5 Deleting the Fertilizer company based on Feedback:

The fertilizer company is deleted based on the feedback given by the customers. If the feedback is negative and more than the limit count, then it is deleted from the block. The deletion of the company can be accessed only by the owner of the block.



oneclickdapp.com/corona-active

Fertilizer Company Authentication On Ethereum Platform

rinkeby Created 12 June 2021

Contract 0xC011Db3363F00784838665FBEE5d70a7C3a45dB0

ABI exists, Input_Company, Com_Feedback, deleted, Fertilizer_Companies, owner, add_Fertilizer_Comp

add_Fertilizer_Company (string)
Compare_Fertilizer_Companies
Com_Feedback (string,string)
Delete_Company
Input_Company (string)

Delete_Company

https://oneclickdapp.com

CONTRACT INTERACTION

0

GAS FEE 0.000169
No Conversion Rate Available

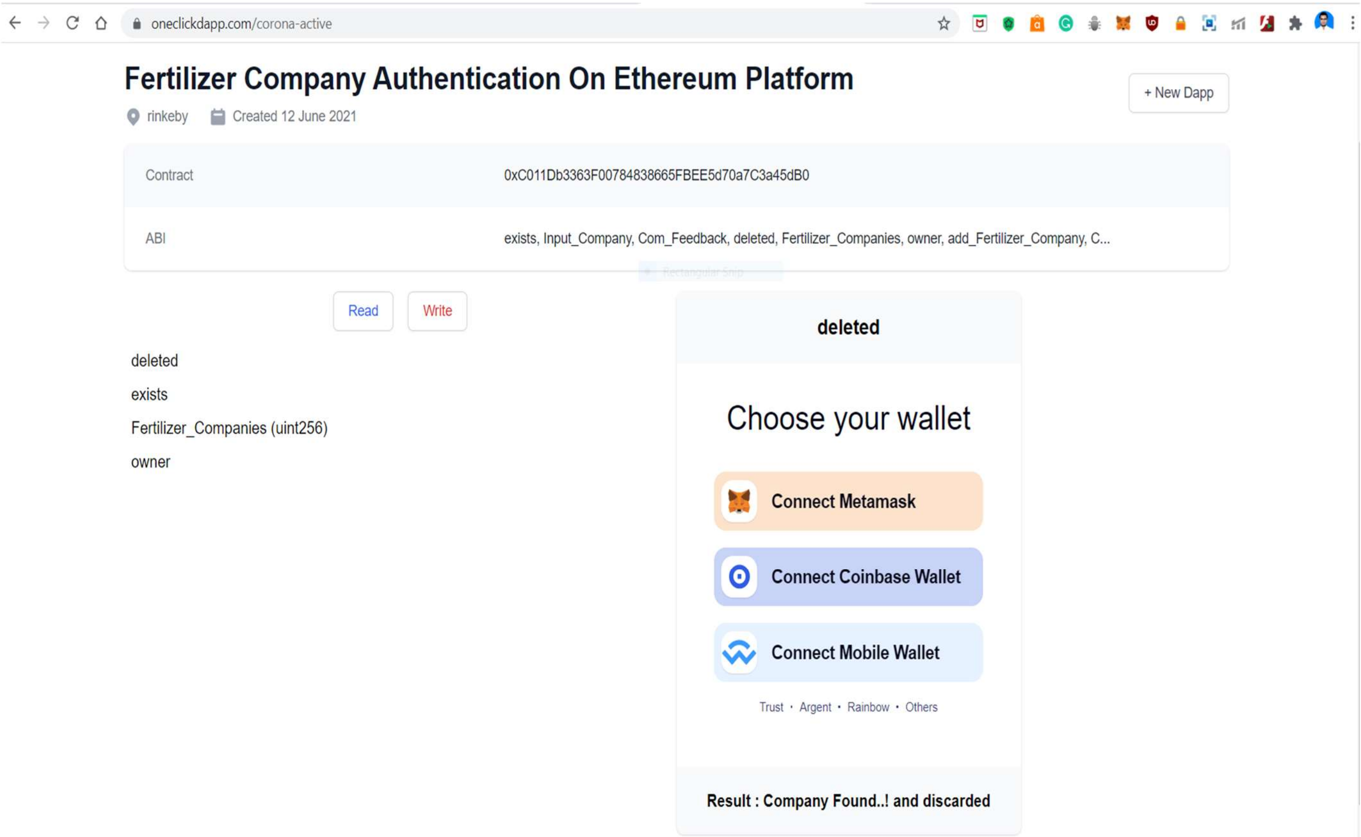
Gas Price (GWEI): 1 Gas Limit: 168548

AMOUNT + GAS FEE
TOTAL 0.000169
No Conversion Rate Available

Fig.5.3.5. Deleting the Fertilizer company based on Feedback

5.3.6. Status Of Deletion Operation:

To know the status of the fertilizer company name i.e., deleted or not by clicking on the deleted option we can know the status where the "Result" is shown as "Company Found..! and discarded". If the limit count is not reached then the company name is not deleted and the status is shown as "sorry cannot be discarded".



oneclickdapp.com/corona-active

Fertilizer Company Authentication On Ethereum Platform

rinkeby   Created 12 June 2021

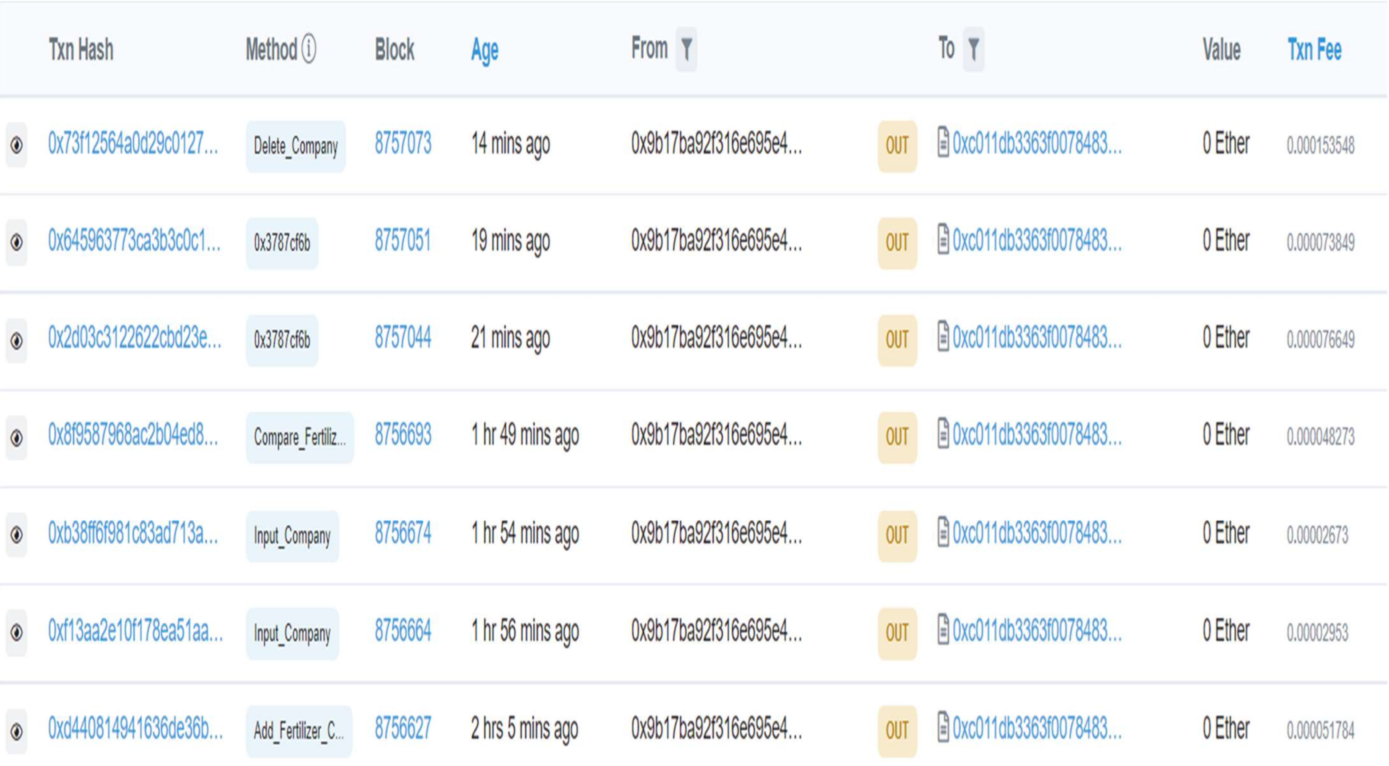
Contract: 0xC011Db3363F00784838665FBEE5d70a7C3a45dB0

ABI: exists, Input_Company, Com_Feedback, deleted, Fertilizer_Companies, owner, add_Fertilizer_Company, C...

Result : Company Found..! and discarded

Fig: 5.3.6. Status Of Deletion Operation

5.3.7. Transactions Recorded on Ethereum Rinkeby network:



Txn Hash	Method	Block	Age	From		To	Value	Txn Fee
0x73f12564a0d29c0127...	Delete_Company	8757073	14 mins ago	0x9b17ba92f316e695e4...	OUT	0xc011db3363f0078483...	0 Ether	0.000153548
0x645963773ca3b3c0c1...	0x3787cf6b	8757051	19 mins ago	0x9b17ba92f316e695e4...	OUT	0xc011db3363f0078483...	0 Ether	0.000073849
0x2d03c3122622cbd23e...	0x3787cf6b	8757044	21 mins ago	0x9b17ba92f316e695e4...	OUT	0xc011db3363f0078483...	0 Ether	0.000076649
0x8f9587968ac2b04ed8...	Compare_Fertiliz...	8756693	1 hr 49 mins ago	0x9b17ba92f316e695e4...	OUT	0xc011db3363f0078483...	0 Ether	0.000048273
0xb38ff6f981c83ad713a...	Input_Company	8756674	1 hr 54 mins ago	0x9b17ba92f316e695e4...	OUT	0xc011db3363f0078483...	0 Ether	0.00002673
0xf13aa2e10f178ea51aa...	Input_Company	8756664	1 hr 56 mins ago	0x9b17ba92f316e695e4...	OUT	0xc011db3363f0078483...	0 Ether	0.00002953
0xd440814941636de36b...	Add_Fertilizer_C...	8756627	2 hrs 5 mins ago	0x9b17ba92f316e695e4...	OUT	0xc011db3363f0078483...	0 Ether	0.000051784

Fig.5.3.7. Transactions Recorded on Ethereum Rinkeby network

CHAPTER 6

TESTING

The main objective of testing is to uncover a host of errors, systematically and with minimum effort and time. Stating formally, we can say,

Testing is a process of executing a program with the intent of finding an error.

- A successful test is one that uncovers an as yet undiscovered error.
- A good test case is one that has a high probability of finding error, if it exists.

The first approach is what known as Black box testing and the second approach is White box testing. We apply white box testing techniques to ascertain the functionalities top-down and then we use black box testing techniques to demonstrate that everything runs as expected.

Black-Box Testing:

This technique of testing is done without any knowledge of the interior workings of the application The tester is oblivious to the system architecture and does not have access to the source code. Typically, while performing a black-box test, a tester will interact with the system's user interface by providing inputs and examining the outputs without knowing how and where the inputs are worked upon.

- Well suited and efficient for large code segments
- Code access is not required
- Clearly separates user's perspectives from the developer's perspective through visibly defined roles

White-Box Testing:

White-box testing is the detailed investigation of internal logic and structure of the code. It is also called “glass testing” or “open-box testing”. In order to perform white-box testing on an application, a tester needs to know the internal workings of the code.

The tester needs to look inside the source code and find out which part of the code is working inappropriately.

In this, the test cases are generated on the logic of each module. It has been uses to generate the test cases in the following cases:

- Guarantee that all independent modules have been executed.
- Execute all logical decisions and loops.
- Execute through proper plots and curves.

Performance Evaluation

This project has been successfully executed its source code. Initially there were some errors in the code. By resolving them, the code is fully free from errors and bugs.

The performance of our project will purely depend on the consensus algorithm used as we are doing our project on the Ethereum test network which uses a proof of work consensus algorithm so due to usage of proof of work on an average it takes 15 seconds for executing a transaction. But however, ethereum had recently shifted to proof of stake which will execute each transaction at a much faster pace.

CONCLUSION

Finally the demand for security in the technologies is very need now a days . As the technology is growing rapidly. We should also able to protect that usage of technology in an efficient manner. Ethereum and Smart-Contracts are one of the efficient and secured technologies to communicate with the peer participants. Ethereum uses highly secured techniques and algorithms to provide utmost security to the clients in the network. Inventing the technology without the proper security is a kind of useless task in the present day world.

The system will create an opportunity where the farmer can find a reasonable and trustable system which will make the availability of the fertilizers from trusted companies. We describe our abstract system with a proposed architecture and workflow to describe the secured access control and distribution management. Moreover, the user will have trust in the system as it uses smart contract features which confirm data privacy and security.

REFERENCES

BOOKS

[1] Dr. V. K. Saraswat, Dr. Bimal Kumar Roy and C P Gurnani, Blockchain Technology Concepts and Applications, 9 September 2020.

[2] Robert Infante and Thomas Bertani, Building Ethereum Dapps, 9 March 2019.

WEBSITES

[1] https://www.dappuniversity.com/. January 2018.

CONFERENCE PAPERS

[1] Sabbir Ahmed, Md.Tarif Hosen, Md. Ezazul Islam,Md. Hasibul Haan, Blockchain based Fertilizer Distribution System : Bangladesh Perspective, January 10, 2020.