

Creating a Blockchain Network for Supply Chain

A PROJECT REPORT

Submitted by

BL.EN.U4CSE19011

B Jagapraavin

BL.EN.U4CSE19013

Benaka V Rao

BL.EN.U4CSE19043

Ganti Surya Sai Ashwin

in partial fulfillment for the award of the degree of

BACHELOR OF TECHNOLOGY

IN

COMPUTER SCIENCE AND ENGINEERING



AMRITA SCHOOL OF COMPUTING, BANGALORE

AMRITA VISHWA VIDYAPEETHAM

BANGALORE 560 035

December – 2022

AMRITA VISHWA VIDYAPEETHAM

AMRITA SCHOOL OF COMPUTING, BANGALORE, 560035



BONAFIDE CERTIFICATE

This is to certify that the project report entitled “**Creating a Blockchain Network for Supply Chain**” submitted by

BL.EN.U4CSE19011

B Jagapraavin

BL.EN.U4CSE19013

Benaka V Rao

BL.EN.U4CSE19043

Ganti Surya Sai Ashwin

in partial fulfillment of the requirements as part of **Bachelor of Technology** in “**COMPUTER SCIENCE AND ENGINEERING**” is a bonafide record of the work carried out under my guidance and supervision at Amrita School of Computing, Bangalore.

A handwritten signature in blue ink, appearing to read "Beena B M".

Dr. Beena B M

Associate. Professor

Dept. of CSE

Dr. Sriram Devanathan

Professor and Chairperson,

Dept. of CSE

This project report was evaluated by us on 29.12.22

A handwritten signature in blue ink, appearing to read "Suja P".

EXAMINER1

Dr. Suja P.

A handwritten signature in blue ink, appearing to read "S. Guruvammal".

EXAMINER2

Ms. Guruvammal S.

ACKNOWLEDGEMENTS

The satisfaction that accompanies successful completion of any task would be incomplete without mention of people who made it possible, and whose constant encouragement and guidance have been source of inspiration throughout the course of this project work.

We offer our sincere pranams at the lotus feet of “AMMA”, **MATA AMRITANANDAMAYI DEVI** who showered her blessing upon us throughout the course of this project work.

We owe our gratitude to **Prof. Manoj P.**, Director, Amrita School of Engineering, Bangalore.

We thank **Dr. Sriram Devanathan**, Principal and Chairperson-CSE, Amrita School of Engineering, Bangalore for his support and inspiration.

It is a great pleasure to express our gratitude and indebtedness to our project guide **Dr. Beena B M**, Associate Professor, Department of Computer Science and Engineering, Amrita School of Computing, Bangalore for her/his valuable guidance, encouragement, moral support, and affection throughout the project work.

We would like to thank express our gratitude to project panel members for their suggestions, encouragement, and moral support during the process of project work and all faculty members for their academic support. Finally, we are forever grateful to our parents, who have loved, supported and encouraged us in all our endeavors.

ABSTRACT

The Blockchain era offers critical opportunities for the supply chain management control. The aim of our project is the employment of blockchain generation in the field of deliver/supply chain management. Though BCT (Blockchain chain technology) has been widely involved in the field of cryptocurrencies, non-financial applications which include supply chain, electricity, and food industry are also planning. Blockchain can offer an eternal, sharable, auditable document of merchandise through their supply chain, which improves product traceability, authenticity, and legality in a greater value-effective way. Blockchain is a distributed ledger technology that captured the attention of all practitioners and educators last year. A large number of conceptual studies and small number of empirical studies have been published that focus on addressing contemporary issues and recommending future research directions on Supply Chain governance.

TABLE OF CONTENTS

Ch No.	Description	Page No.
	ACKNOWLEDGEMENTS	I
	ABSTRACT	II
	TABLE OF CONTENTS	III
	LIST OF FIGURES	VII
1	INTRODUCTION	1
	1.1 Introduction To Block Chain	1
	1.2 Why is Blockchain Important	2
	1.3 Introduction to Smart Contracts	2
	1.4 Ethereum	3
	1.5 Motivation	4
	1.6 Complications	4
	1.7 Problem Statement	5

	1.8 Definition of the problem statement	5
2	LITERATURE SURVEYS	6
	2.1 An overview of Blockchain Technology: Architecture, Consensus and Future Trends.	6
	2.2 The Integrated Supply Chain Management System	7
3	SYSTEM SPECIFICATIONS	8
	3.1 Software Requirements	8
	3.1.1 React.js, Ether.js and Etherscan	8
	3.1.2 Node.js	9
	3.1.3 Goerli test network and Goerli Faucet	9
	3.1.4 Meta Mask and Alchemy	10
	3.1.5 Solidity	10
	3.1.6 Remix IDE	11
	3.1.7 Hardhat IDE	11

	3.1.8 Windows 11	11
	3.1.9 Visual Studio Code (VS Code)	12
4	SYSTEM DESIGN	13
	4.1 Overview of the Architecture 1	13
	4.2 Overview of the Architecture 2	15
5	SYSTEM IMPLEMENTATION	17
	5.1 Introduction	17
	5.2 Configuration	17
	5.3 Imports	18
	5.4 Building and deployment of Contract	18
	5.5 Contracts, functions and its uses	19
6	RESULTS	24
	6.1 User Interface	24

	6.2 Test Results on Goerli Network	25
7	CONCLUSION AND FUTURE SCOPE	31
	7.1 Conclusion	31
	7.2 Future Scope	32
8	REFERENCES	33

LIST OF FIGURES

Fig No.	Description	Page No.
4.1	Overview of part arrival	13
4.2	Overview of Part going through different Phases	15
5.1	Import Functions	18
5.2	Security Contract	19
5.3	Security Contract	20
5.4	Investigator Contract	21
5.5	Investigator Contract	21
5.6	Final Deploy Function	22
6.1	The User Interface	25
6.2	Goerli Network Status	26
6.3	The Output on Goerli Network	27

6.4	Remix IDE Output (Deployed Investigator Contract)	29
6.5	Remix IDE Output (CLI Investigator Contract)	30

CHAPTER - 1

INTRODUCTION

1.1 INTRODUCTION TO BLOCK CHAIN

What is Blockchain?

Blockchain maybe be shared, changeless ledger that facilitates the tactic of recording transactions and tracking assets across corporate networks. Assets are often tangible (house, cars, cash, land) and intangible (intellectual property, patents, copyrights, trademarks). Almost anything of value can be tracked and traded on a blockchain network, reducing risk and lowering costs for everyone involved.

Blockchain could be a style of shared database that differs from common databases in the way information is stored. Blockchain stores data/information in blocks and links them together using cryptographic algorithms.

As new data/information comes in, it is entered into a new block. When a block is filled with data, it is concatenated with the previous block and the data is concentrated in chronological order.

Blockchains can store many alternative types of information's, but the most common use so far is a transaction ledger.

In the case of Bitcoins, the blockchain is employed in a decentralized manner, so that all users collectively hold control, rather than being controlled by one individual or group/clusters.

A decentralized blockchain is changeless or immutable. This suggests that the information entered can't be undone. For Bitcoin, this suggests that transactions are always recorded and it can be viewed by anyone.

1.2 WHY IS BLOCKCHAIN IMPORTANT?

Business in the information. The quicker and lot of correct the reception, the higher the results. Blockchain is right for providing this information because it provides instant, shared associates utterly clear information that holds on in an immutable ledger that may solely be accessed by licensed network members. Blockchain networks will track orders, payments, accounts, production and many more. And since members share a unified read view of truth, they can see every transaction detail from the start to the finish, resulting in bigger or greater confidence, new efficiencies, and Opportunities.

1.3 INTRODUCTION TO SMART CONTRACTS

A Smart Contract is a program that is stored on the blockchain and is executed when certain conditions are met, they are often used to automate the execution of contacts, so that all parties can see the results immediately, without intermediaries and without wasting time. You can also automate workflows and trigger further actions when conditions are met.

A Smart Contract is a self-executing contract where the terms and conditions between the buyer and the seller are written directly in lines of code. The code and the contracts it contain live on a decentralized blockchain network. The code that controls the execution and the transaction is traceable and irreversible.

Smart Contracts works by following simple “if/when..then..” statements written in code on the blockchain. A network of computers takes action when certain conditions are met and verified.

1.4 ETHEREUM

Ethereum is basically a decentralized platform that establishes a peer-to-peer network that securely executes and verifies application code, known as smart contract. Smart Contract allow participants to transact with each other without a trusted central authority.

1.5 MOTIVATION

Why use blockchain over traditional Software SDLC?

A Key attribute of blockchain technology that sets it apart from traditional database technology is its public verifiability, which is made possible through integrity and transparency.

Integrity: Each user is positive that the information or data they are retrieving is uncorrupted and unaltered since the moment it was recorded.

Transparency: Each user can verify how the blockchain has been appended over time.

1.6 COMPLICATIONS

- Since Blockchain technology is still evolving there are limited resources on solidity and smart contracts.
- Connection between front end and back end was complicated.
- Learning about test network was a huge network as the mainnet requires real ethers to perform transactions. (1ether = Rs.98,996.76)

1.7 PROBLEM STATMENT

To get ready for deployment of the first track/trace/alert covering stores inward process from physical delivery of material at security checkpoint to binning. The target is that binning entails that a material is ready for issue in conformity with the process, with 1 hour as the design goal.

To refine the above target and make it more engineering inspection complaint, with engineering inspection of incoming material and avoid work stoppage arising from missing items, especially items in kitted form.

1.8 DEFINITION OF PROBLEM STATEMENT

The main objective of our project is to create a system that transfers the material box from security to the bins in an efficient time (within 1-hour maximum).

It has been noticed that the transfer of the material box from security to the bins takes an estimated 4 hours which is an inefficient process, also to make sure that each material box goes to its corresponding base and to ensure the condition of the material box and to document all the process using blockchain technology as modification of data will be recorded and can't be manipulated as an anonymous user.

CHAPTER – 2

LITERATURE REVIEW

2.1 AN OVERVIEW OF BLOCKCHAIN TECHNOLOGY: ARCHITECTURE, CONSENSUS AND FUTURE TRENDS.

- Blockchain technology has basically shown the potential to transform the traditional industry with these key features:
Decentralization, sustainability, anonymity and verifiability.
- In this paper, we will present a comprehensive overview of blockchain. We then discuss the Blockchain architecture and the key features.
- Finally there are some consensus algorithms that are basically used in blockchain which are analyzed, and which compare these protocols in different aspects.
- Today Blockchain-based applications are emerging, and it plans to lead in-depth investigation of blockchain-based applications in the future.

2.2 THE INTEGRATED SUPPLY CHAIN MANAGEMENT SYSTEM.

- Supply chain management system identifies an appropriate decomposition of supply chain function and encapsulates into agents.
- Developing the protocol and the strategies for the communication of the information, coordination of the decisions, and management of change.
- It basically develops or uses the State-of-the-art algorithms for the agent decision-making.
- Developing an incremental, “anytime” model of problem solving for each functional agent so that it can provide rapid response to unplanned for events.
- Extending each function-oriented agent so that it is able to answer a greater number of questions within its functional domain.

CHAPTER – 3

SYSTEM SPECIFICATIONS

3.1 Software requirements

3.1.1 React.js, Ether.js and Etherscan.

The main goal of React.js is to develop a user interface (UI) that improves the speed of the application. It uses a virtual DOM (JavaScript object), which improves the performance of the application. The JavaScript virtual DOM is faster than the normal DOM. We can use React.js client-side and server-side as well as with other Frameworks. It uses component and data models to improve readability and help maintain larger applications.

The ether.js library is basically a complete and compact library whose primary purpose is to interact with the Ethereum Blockchain and its ecosystem. It was initially designed for its use with ethers.io and has since expanded into a more of general library. Etherscan is basically a well-known blockchain explorer that allows the users to view and allows users to investigate the data on the Ethereum network. This platform allows the users to track all the Ethereum blockchain transactions, user's wallet activity and other smart contracts.

3.1.2 Node.js

Node.js is intended for server-side programming and is primarily implemented for non-blocking event servers, such as traditional websites and backend API services, but was originally designed with a push-based architecture in mind. Also, Node.js is an open-source server environment which is free to use and which uses JavaScript on the server.

3.1.3 Goerli test network and Goerli Faucet

Goerli was initially launched in 2019 and is one of only two public Ethereum PoS test networks currently which is maintained by the community with Sepolia. The TestNet supports cross-clients, which means that developers can use different programming languages in their Ethereum development efforts.

Ethereum Faucet is a development tool to take an Ether (ETH) test network to test and troubleshoot your decentralized application or protocol before working on the Ethereum mainnet, where real Ether must be used. Most faucets either require social authentication (e.g. a Twitter post or a login that confirms you're a real person) or place you in a queue to wait for a testnet token pass through the faucet. Alchemy Goerli Faucet is free, fast and requires no authentication, although you can optionally log into Alchemy to increase the amount of drip.

3.1.4 Meta Mask and Alchemy

MetaMask is a cryptocurrency software wallet which is used to interact with the Ethereum Blockchain. It allows the users to access their Ethereum wallets through a browser extension or mobile app, which can then be used to interact with the decentralized applications.

Alchemy is basically a blockchain scaling platform that allows developers to securely build, test and monitor the DApps (Decentralized Applications). The platform provides reliable network connectivity and node management endpoints.

3.1.5 Solidity

Solidity is a statically typed programming language designed for developing Smart Contracts that runs on Ethereum. It is an object-oriented programming language created specifically by the Ethereum network team to build and design Smart Contract on the Blockchain platform. It is used to create Smart Contract that implements business logic and create a chain of transaction records in blockchain system.

3.1.6 Remix IDE

Remix IDE is a support tool with a graphical interface for developing smart contracts, which is Used by professionals as well as beginners. Remix will get you started in record time. Remix works well with other tools and allows for a simple deployment process to the channel of your choice. Remix is famous for its visual debugger.

3.1.7 Hardhat IDE

Hardhat is a development environment that helps developers test, compile, deploy and debug dApps (Decentralized Application) on the Ethereum blockchain. It plays an important role in assisting programmers and developers in managing tasks, which is important for the development of Smart Contracts and dApps. Hardhat is a solidity development environment built using Node.js

3.1.8 Windows 11

Microsoft's Windows OS (operating system) family of software runs on a range of gadgets, including desktops, laptops, tablets and smartphones. The latest version is Windows 11, which was made available in July 2021. A graphical user interface, a web browser, a collection of apps, and support for hardware like keyboards, mouse, and gaming controllers are all included in the Windows OS. Additionally, the OS comes

with several essential functions including networking, security, and system administration.

3.1.9 Visual Studio Code (VS Code)

Visual Studio Code is a simple code editor that supports development activities such as debugging, task execution, and version control. Its purpose is to provide only the tools developers need for a quick code build, debugging cycle, and leave more complex workflows for more comprehensive IDEs, such as the Visual Studio IDE.

CHAPTER - 4

SYSTEM DESIGN

4.1 OVERVIEW OF THE ARCHITECTURE 1

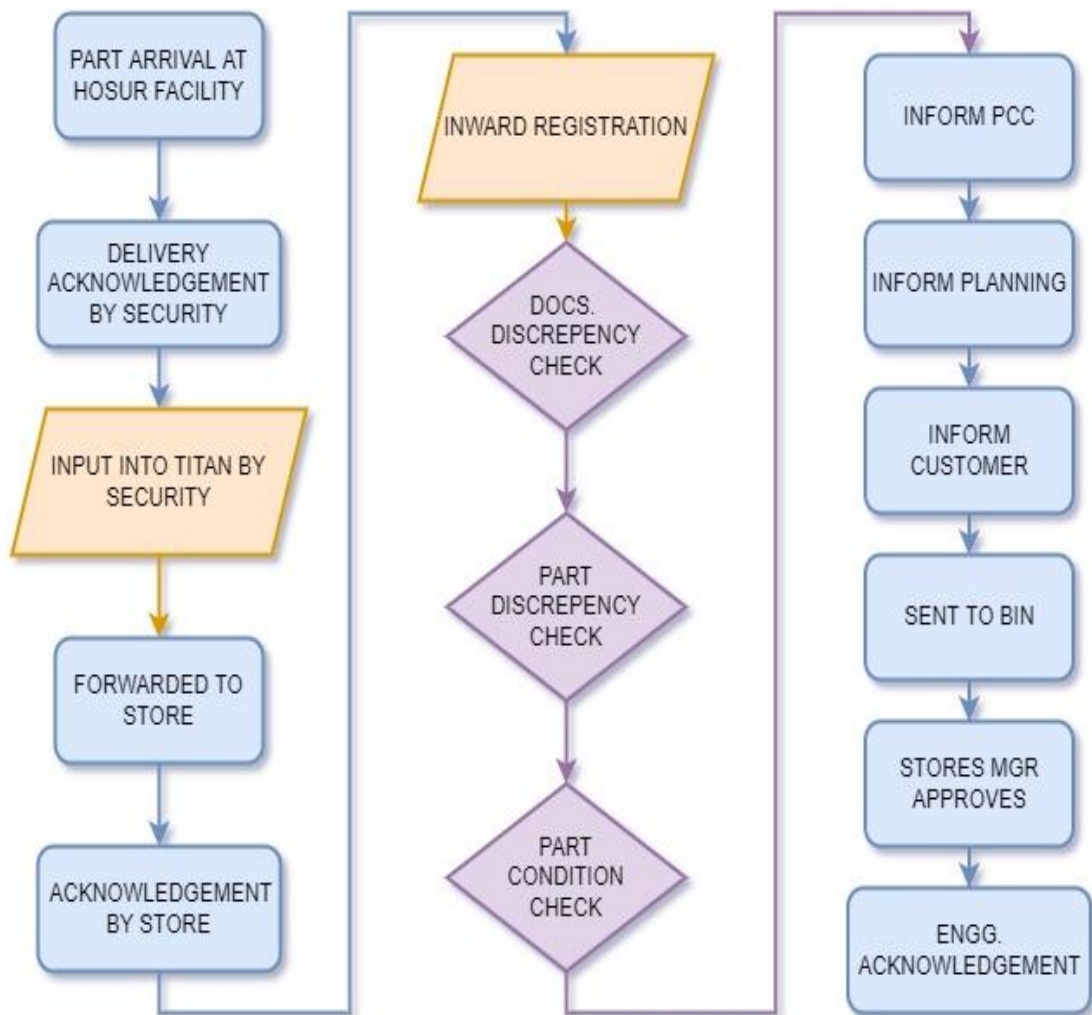


Fig 4.1 Overview of part arrival

Initially the part reaches the Hosur Facility and the Security acknowledges whether the part has come from the verified user. After the verification of the part, the security inputs about the part into his system and the part is then sent to store. Then in the store, they will investigate each part, whether the part has any damage on it or as such.

The condition of part is then checked and later it is sent to its base. Like for example if the part is an Air India's part, it must go to Air India's Base only. Then the manager of that base approves the part and finally work with the part begins.

4.2 OVERVIEW OF ARCHITECTURE 2

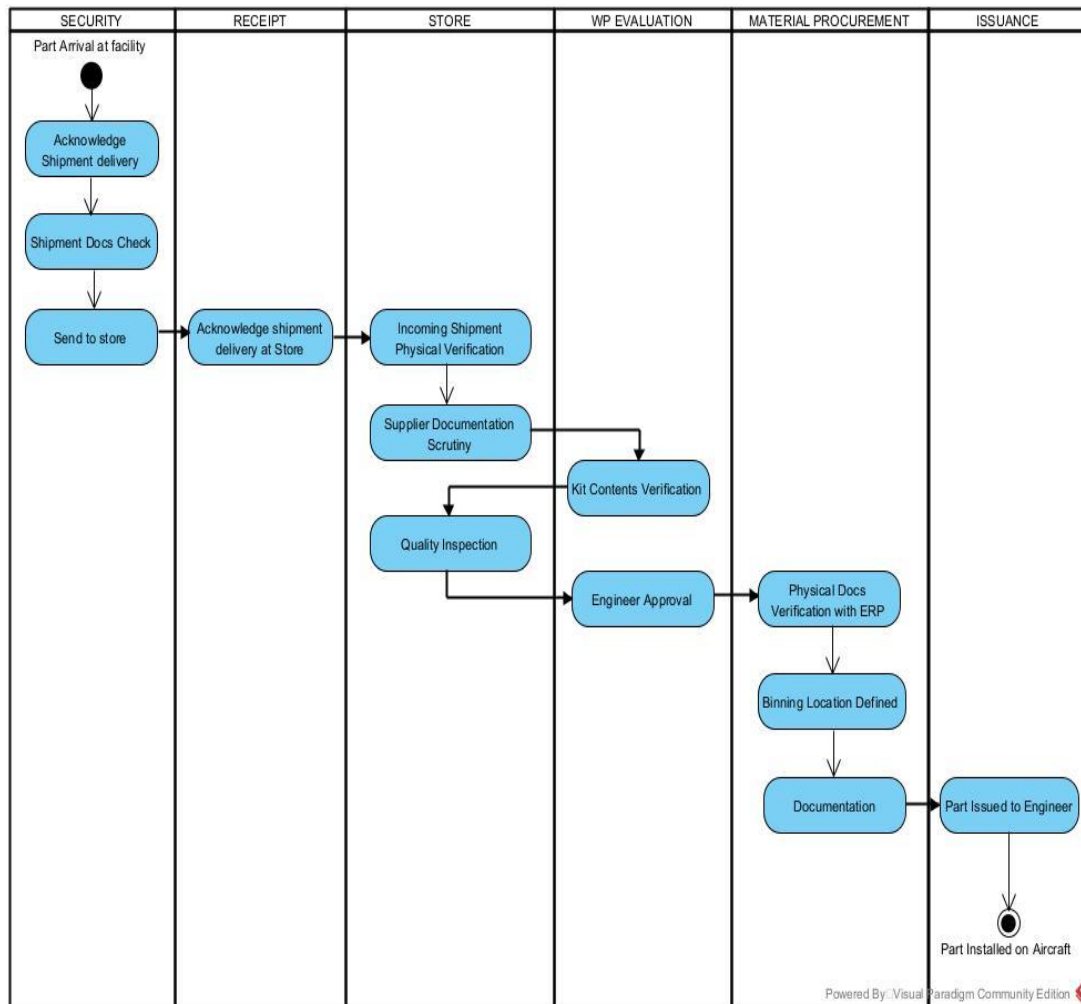


Fig 4.2 Overview of Part going Through different Phases

Here, there are different phases a part must go through, so in the first phase (i.e Security phase) initially the part is arrived at the Hosur facility, and it is checked by

the security, then the security acknowledges the part and the security checks for the shipment documents and after verifying the part has come from authorized user, security then send the part to the store. Then in the second phase (i.e Store phase) The physical verification of the part is done, like checking whether there is any damage on the part or did they send a wrong part, so basically a complete quality inspection is done. After the quality inspection of the part, it is then sent to the bin or call it a base. Base is where the part actually belongs to. Like for example if the part is an Air India's part, it must go to Air India's Base only. Then finally the documentation is done and the part is issued to the engineer. Finally, the engineer uses the part and begins his work. With the help of our project we are making this process seamless and more efficient so that a mechanic who is waiting for a part at expected time receives the same and the workflow is not stopped with respect to the same.

CHAPTER – 5

SYSTEM IMPLEMENTATION

5.1 INTRODUCTION

Our process:

- Create and deploy Smart Contracts on to Goerli test network.
- Using Ether.js, Goerli Etherscan and Alchemy Configuration to communicate with smart contracts and check their health and maintenance as per requirement and Metamask to process the transactions.
- Using React.js for user interface for better and easy communication for the transactions to be authorized and processed.

5.2 CONFIGURATION

- `npm install --save-dev hardhat`
- `npm install --save-dev "hardhat@2.12.4" "@nomicfoundations/hardhat-toolbox@^2.0.0"`
- `npx hardhat`
- `npm install create-react-app`

5.3 IMPORTS

```
import abi from "./contract/Security.json";  
import {useState,useEffect} from 'react';  
import {ethers} from "ethers";  
import Buy from "./components/Buy";  
import Memos from "./components/Memos";
```

Fig 5.1 Import Functions

5.4 BUILDING AND DEPLOYMENT OF CONTRACT

- npm install dotenv --save
- npx hardhat run --network goerli scripts/finalDeploy.js
- npm start

5.5 CONTRACTS, FUNCTIONS AND ITS USES

```
contracts > Security.sol
1  //SPDX-License-Identifier: UNLICENSED
2  pragma solidity 0.8.17;
3
4  contract Security{
5      mapping(address=>uint) public contributors;
6      address public security;
7      uint public sessionTime;
8      bool public target1;
9      bool public target2;
10     uint public barcode;
11     mapping(int=>string) public packageStatus;
12
13
14     constructor(uint _deadline){
15         //target1=_target1;
16         //target2=_target2;
17         sessionTime=block.timestamp+_deadline;
18         security=msg.sender;
19         //packageStatus[0] = "Damaged";
20         //packageStatus[2] = "Partial Accepted";
21         //packageStatus[1] = "Accepted";
22
23
24     }
25     struct Memo{
26         bool target1;
27         bool target2;
28         uint256 timestamp;
29         address from;
30     }
31
32     Memo[] memos;
33
34     int public packst = -1;
35     function processOrder(bool target1,bool target2) public returns(int) {
36         require(block.timestamp < sessionTime,"SessionTimeout");
37     }
```

Fig 5.2 Security Contract

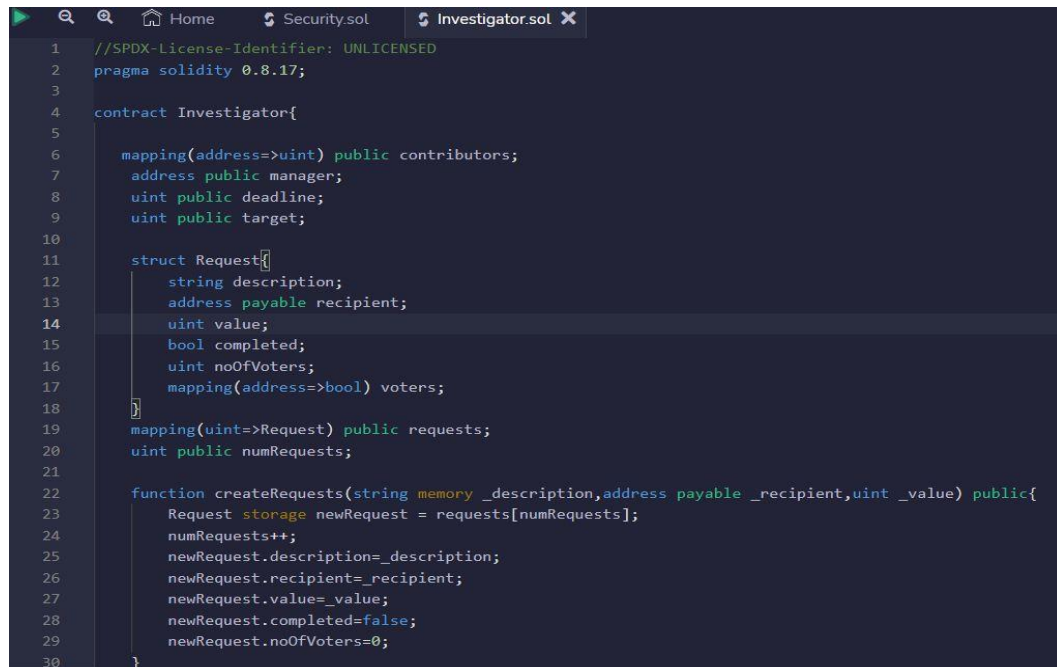
```
contracts > Security.sol
37
38     if(target1==true && target2==true){
39         packst = 1;
40     }else if((target1==false && target2==true) || (target1==true && target2==false)){
41         packst = 2;
42     }else{
43         packst = 0;
44     }
45     memos.push(Memo(target1,target2,block.timestamp,msg.sender));
46     return packst;
47
48
49 }
50 function getMemos() public view returns (Memo[] memory){
51     return memos;
52 }
53
54 //int public status = packageStatus[packst];
55
56 }
57
```

Fig 5.3 Security Contract

Security is a smart contract designed to process the order at the security gate once a package has arrived at the facility. It is designed in a way a security has to answer a few questions to process the order securely and complete the transaction.

From Fig 5.3, the processOrder function takes 2 inputs i.e the boolean answers as the input and processes the function as per the parameters.

The getMemos function helps store the boolean values, the timestamp and the user address who is calling the process order function. This helps us in tracking which user had processed the order without losing integrity.

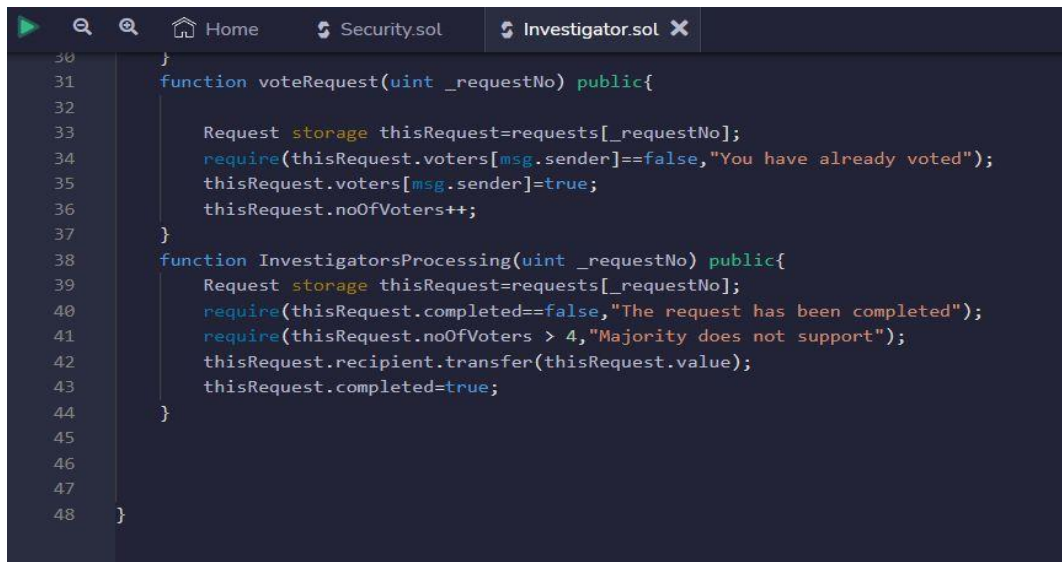


```

1 //SPDX-License-Identifier: UNLICENSED
2 pragma solidity 0.8.17;
3
4 contract Investigator{
5
6     mapping(address=>uint) public contributors;
7     address public manager;
8     uint public deadline;
9     uint public target;
10
11     struct Request{
12         string description;
13         address payable recipient;
14         uint value;
15         bool completed;
16         uint noOfVoters;
17         mapping(address=>bool) voters;
18     }
19     mapping(uint=>Request) public requests;
20     uint public numRequests;
21
22     function createRequests(string memory _description,address payable _recipient,uint _value) public{
23         Request storage newRequest = requests[numRequests];
24         numRequests++;
25         newRequest.description=_description;
26         newRequest.recipient=_recipient;
27         newRequest.value=_value;
28         newRequest.completed=false;
29         newRequest.noOfVoters=0;
30     }

```

Fig 5.4 Investigator Contract



```

30 }
31 function voteRequest(uint _requestNo) public{
32
33     Request storage thisRequest=requests[_requestNo];
34     require(thisRequest.voters[msg.sender]==false,"You have already voted");
35     thisRequest.voters[msg.sender]=true;
36     thisRequest.noOfVoters++;
37 }
38 function InvestigatorsProcessing(uint _requestNo) public{
39     Request storage thisRequest=requests[_requestNo];
40     require(thisRequest.completed==false,"The request has been completed");
41     require(thisRequest.noOfVoters > 4,"Majority does not support");
42     thisRequest.recipient.transfer(thisRequest.value);
43     thisRequest.completed=true;
44 }
45
46
47
48 }

```

Fig 5.5 Investigator Contract

The investigator contract is the second phase of the project wherein the status and condition of the product is verified by one or more investigators to ensure the quality of the product.

In the above contract, we have 3 main functions namely:

- createRequests: This function helps create a request by the manager so that the investigators can process the order for further usage and processing of the product.
- voteRequest: This function is used by the voters or the investigators to process the state and quality of the product.
- InvestigatorProcessing: This function is again used by the manager and can be called when the number of voters/investigators is greater than 50%. This helps secure the consensus mechanism and follow the main property of BlockChain technology which helps maintain integrity and authenticity.

```
scripts > JS finalDeploy.js > main > contract
1  const hre = require("hardhat");
2
3  async function main() {
4    const Security = await hre.ethers.getContractFactory("Security");
5    const contract = await Security.deploy([200000]); //instance of contract
6
7    await contract.deployed();
8    console.log("Address of contract:", contract.address);
9  }
10 main().catch((error) => {
11   console.error(error);
12   process.exitCode = 1;
13 });
```

Fig 5.6 Final Deploy Function

Under the deployment file, we use hardhat and Ether.js to communicate with goerli test network wherein we deploy our contract using the get contract factory which stores the metadata of the contract on the goerli network and then the contract is deployed with a time limit so that it expires once its use case is completed and once the contract is deployed its address is fetched from goerli test network and printed on to the terminal. In case of any error during the deployment it is handled by hardhat.

CHAPTER - 6

RESULTS

6.1 USER INTERFACE

It is designed using React.js, HTML and CSS. React.js is JavaScript framework created by Facebook for easily making user interfaces and connecting them with backend.

Homepage is the user interface seen by the security guard to process the order received at the facility. It is designed in such a way that a naïve user can also operate it with some basic skills.

It also displays which user initiated and processed the transaction along with the timestamp. This helps admin dashboard view which transaction was processed when. This also helps the payroll on how to pay the personale, managers view on procurement of the item and many more use cases.



Process Form

Please Answer the below questions :

The Package received in sealed Condition?

☐ true
☐ false

Document and Saftey Check of the Package Verified?

☐ true
☐ false

Messages

	12/29/2022, 6:53:00 PM		0x9b49f59A07e69DBcB212c432330BE6a492Dfe686
	12/29/2022, 6:57:36 PM		0x9b49f59A07e69DBcB212c432330BE6a492Dfe686
	12/29/2022, 7:09:12 PM		0x9b49f59A07e69DBcB212c432330BE6a492Dfe686
	12/29/2022, 11:39:24 PM		0x9b49f59A07e69DBcB212c432330BE6a492Dfe686

Fig 6.1 The User Interface

6.2 TEST RESULTS ON GOERLI NETWORK.

When a contract is being deployed onto the network it uses gas, gas is a amount of Kwei that is used to process the transaction. This gas value may vary from second to second, and the deployment time also may vary based on how busy the network is, all these results can be viewed on the goerli network status page and also a sample figure can be viewed below.



Fig 6.2 Goerli Network Status

As you can see from the above diagram it gives us detailed information on how busy the network is. Gas price mentioned above is for a contract/kilo pricing but the actual gas value depends on only how much your contract may require. This is called a smart pricing so that the user pays for only what they are using and don't have to pay the full amount and keep the system underutilized.

For each transaction processed, a unique transaction hash is created as a sub chain to the main contract address. These hashes are a proof of transaction as one hash is encrypted using hash of the previous block. As you can see from the Fig 6.3 upon completion of each transaction a sub chain of the contract address is updated along with the unique block number. It also gives the age of the transaction when it was processed and the transaction fee which is basically the gas to process the transaction. In this way each part is interdependent on the other so if any change in either one of

the part the whole chain breaks from that point which tells us there is a loss in integrity of the information or that the information is changed and is not the same as before. So, in case of modifying a intermediate transaction breaks the chain and the integrity of the contract is lost.

The screenshot displays the Etherscan interface for the Goerli Testnet Network. At the top, the Etherscan logo and network name are visible. A search bar and navigation links (Home, Blockchain, Tokens, Misc) are present. The main section shows the contract address: 0xF6a57d6b23b945134a54Ad930810C4d04B6c81dF. Below this, there are tabs for Transactions, Erc20 Token Txns, Contract, and Events. The Transactions tab is active, showing a list of the latest 5 transactions. Each transaction entry includes the Txn Hash, Method, Block number, Age, From address, To address, Value, and Txn Fee. The transactions show a series of calls to the contract, all with a value of 0 Ether and a small transaction fee.

Txn Hash	Method	Block	Age	From	To	Value	Txn Fee
0x24b586f32bec2086a6...	0x6839f52	8222390	1 hr 34 mins ago	0x9b49f59a07e69dcb2...	IN 0xf6a57d6b23b945134a...	0 Ether	0.00014806
0x01c6d5c4065468e55e...	0x6839f52	8221317	6 hrs 4 mins ago	0x9b49f59a07e69dcb2...	IN 0xf6a57d6b23b945134a...	0 Ether	0.00014844
0xd614262696f0ea198...	0x6839f52	8221268	6 hrs 15 mins ago	0x9b49f59a07e69dcb2...	IN 0xf6a57d6b23b945134a...	0 Ether	0.00014833
0x40d8127392f337518...	0x6839f52	8221247	6 hrs 20 mins ago	0x9b49f59a07e69dcb2...	IN 0xf6a57d6b23b945134a...	0 Ether	0.00017831
0x636b90a6b9973d1f7f...	0x60806040	8221238	6 hrs 22 mins ago	0x9b49f59a07e69dcb2...	IN Contract Creation	0 Ether	0.00000271

Fig 6.3 The Output on Goerli Network

Under the investigator contract a user initiates a vote to process the order, then the other users have to vote for approval and the investigator processing function executes when the number of votes are greater than 50%, else it throws an error “Majority does not support”. This means all the required investigators have not voted yet, therefore the block cannot be created unless there is 51% stake is attained. This mechanism is used because a single individual cannot be trusted on the condition of the product irrespective of their specialization, sometimes there might be a error an individual can miss on the naked eye so if a 51% users approve that the part is secure then only it has to be processed so that the TAT(Turn around Time) is not effected at the installation of the part. This helps save time almost instantly and if there is a damage in the part it can be rectified at the Investigator phase so that a engineer does not have to be waiting for that task and can be reassigned onto a different task

DEPLOY & RUN TRANSACTIONS ✓ ➤

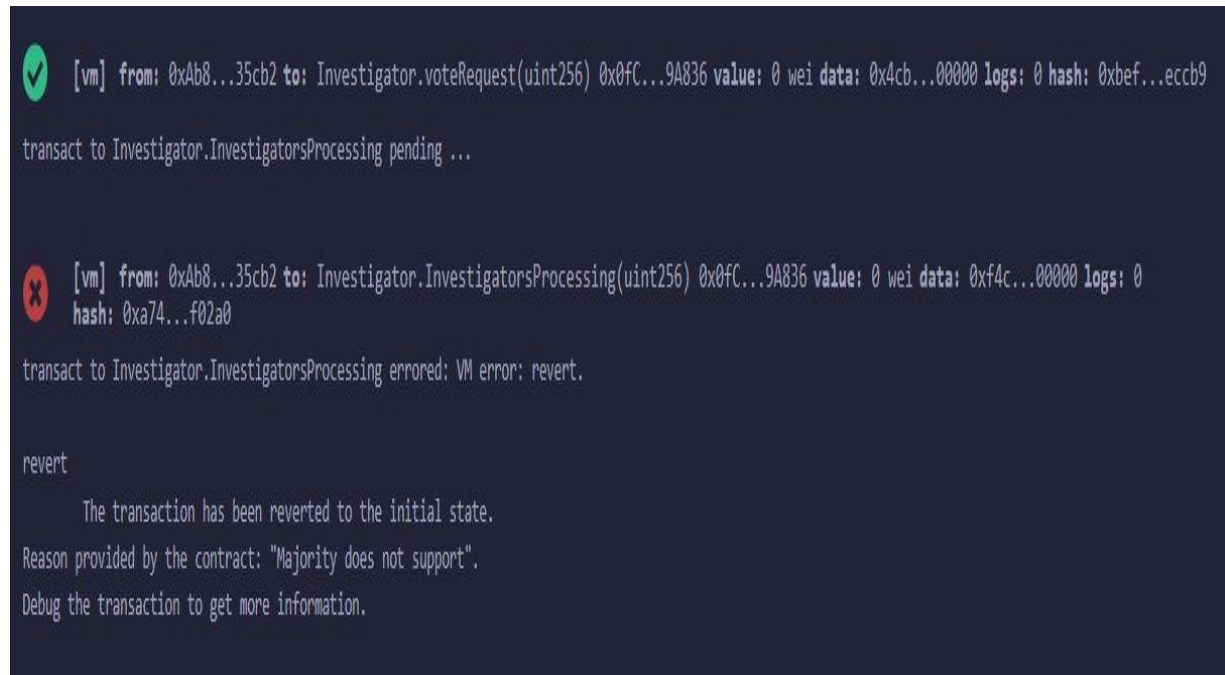
Balance: 0 ETH

createReq...	"Order 7011 Approval",0x5B3E	▼
Investigat...	0	▼
voteRequest	0	▼
contributors	address	▼
deadline		
manager		
numReque...		
requests	0	▼

0: string: description Order 7011 Approval
1: address: recipient 0x5B38Da6a701c568545dCfcB03FcB875f56beddC4
2: uint256: value 10
3: bool: completed false
4: uint256: noOfVoters 0

target

Fig 6.4



```
✓ [vm] from: 0xAb8...35cb2 to: Investigator.voteRequest(uint256) 0x0fc...9A836 value: 0 wei data: 0x4cb...00000 logs: 0 hash: 0xbef...ecb9
transact to Investigator.InvestigatorsProcessing pending ...

✗ [vm] from: 0xAb8...35cb2 to: Investigator.InvestigatorsProcessing(uint256) 0x0fc...9A836 value: 0 wei data: 0xf4c...00000 logs: 0
hash: 0xa74...f02a0
transact to Investigator.InvestigatorsProcessing errored: VM error: revert.

revert
    The transaction has been reverted to the initial state.
Reason provided by the contract: "Majority does not support".
Debug the transaction to get more information.
```

Fig 6.5

CHAPTER – 7

CONCLUSION AND FUTURE SCOPE

7.1 CONCUSION

- Blockchain follows consensus mechanism which helps preserve state and incase of any intermediate changes the whole chain is disconnected resulting in change of integrity. This technology is a wonderful use case in supply chain as it helps track product status without compromising on unauthorized anonymous changes.
- Once the application is developed and tested, it can be used to track, plan and alert the concerned users as per their privileges. As in a security will be able to process the order at the gate, an investigator will be able to verify how secure the product is, stores manager to put the product in respective bin, Engineer can pick it up from the same and many more.
- Initially, the transfer of material box from security to the bin takes 4 hours but after implementation of Blockchain technology using smart contracts, it would optimize the process to 1 hour for transfer of box from security, to the investigator, the bins and finally to the engineer.

7.2 FUTURE SCOPE

- Implementing this technology for other parts of the supply chain, post Security and Investigator this technology can be implemented onto the part fetching and ordering from the client side.
- Maintaining the integrity, improving the security measures and making ease of username password by mapping it to the Metamask account ID and deploying the same onto a cloud environment for better access control.
- Investigation of changes, adding backtracking algorithms to track where the chain was modified and then using that particular has to identify which ID modified data so that further action can be taken.
- Improve Speed of the supply chain management not only in terms of technology but also in terms of transportation of and ordering of product form facility.
- We can also add a dashboard for the client to create a request for the service of the aircraft
- We can create a admin dashboard with perfect user experience for centralized authorized access of data.

CHAPTER – 8

REFERENCES

- [1] Z. Zheng, S. Xie, H. Dai, X. Chen and H. Wang, "An Overview of Blockchain Technology: Architecture, Consensus, and Future Trends," 2017 IEEE International Congress on Big Data (BigData Congress), 2017, pp. 557-564, doi: 10.1109/BigDataCongress.2017.85.
- [2] Mark S. Fox, John F. Chionglo, Mihai Barbuceanu Department of Industrial Engineering, University of Toronto "The Integrated Supply Chain Management System"
- [3] S. Bhalerao, S. Agarwal, S. Borkar, S. Anekar, N. Kulkarni and S. Bhagwat, "Supply Chain Management using Blockchain," 2019 International Conference on Intelligent Sustainable Systems (ICISS), 2019, pp. 456-459, doi: 10.1109/ISS1.2019.8908031.
- [4] S. Tavassoli, M. Sardashti and N. K. N. Toussi, "Supply chain management and information technology support," 2009 2nd IEEE International Conference on Computer Science and Information Technology, 2009, pp. 289-293, doi: 10.1109/ICCSIT.2009.5234400.
- [5] R. A. Canessane, N. Srinivasan, A. Beuria, A. Singh and B. M. Kumar, "Decentralised Applications Using Ethereum Blockchain," 2019 Fifth International Conference on Science Technology Engineering and Mathematics (ICONSTEM), 2019, pp. 75-79, doi: 10.1109/ICONSTEM.2019.8918887.