

**A PROJECT REPORT ON  
ON  
RATION DISTRIBUTION MANAGEMENT USING  
BLOCKCHAIN.**

**SUBMITTED TO THE SAVITRIBAI PHULE PUNE UNIVERSITY, PUNE IN  
THE PARTIAL FULFILLMENT FOR THE AWARD OF THE DEGREE**

**OF  
BACHELOR OF ENGINEERING  
IN  
INFORMATION TECHNOLOGY**

**BY**

SANGAM PATLE	ROLL NO: 431051
PRATIK BORSE	ROLL NO: 431007
SACHIN GUJAR	ROLL NO: 431017
JINENDRA SANGHVI	ROLL NO: 431061

**UNDER THE GUIDANCE OF  
DR. SUVARNA PAWAR**



**DEPARTMENT OF INFORMATION TECHNOLOGY  
VIIT, PUNE**

**(2019-2020)**



## CERTIFICATE

This is to certify that the project report entitled  
“ **RATION DISTRIBUTION MANAGEMENT USING BLOCKCHAIN.**”

Submitted by

SANGAM PATLE	ROLL NO: 431051
PRATIK BORSE	ROLL NO: 431007
SACHIN GUJAR	ROLL NO: 431017
JINENDRA SANGHVI	ROLL NO: 431061

Is a bonafide work carried out by them under the supervision of **Dr. Suvarna Pawar** and is approved for the partial fulfillment of the requirement of Savitribai Phule Pune University for the award of Bachelor of Engineering (Information Technology)

This project report has not been earlier submitted to any other Institute or University for the award of any degree or diploma.

**Dr.Suvarna Pawar**

Internal Guide  
Dept. of Information Technology

**Prof.Pravin Futane**

Head of Department  
Dept. of Information Technology

External Examiner

**Dr.P.B Kulkarni**

Principal  
Vishwakarma Institute Of Information Technology

Place:Pune

Date:        /        /2020

---

# ***ACKNOWLEDGEMENT***

This project would not have been possible without the constant guidance and support of the faculty members of the IT Department at VIIT, Pune. Especially, the support of our project guide, **Dr. Suvarna Pawar**, without whose esteemed guidance and support, this project would not have been possible. We cannot thank her enough.

Also, we would like to thank **Prof. Pravin Futane**, our Head of Department, for his constant encouragement.

Last but not the least, we would like to thank our parents for constantly motivating us..

SANGAM ASHOK PATLE  
PRATIK GOVINDA BORSE  
SACHIN ANKUSH GUJAR  
JINENDRA MANOJ SANGHVI

---

## **ABSTRACT**

Recently, the Public Ration Distribution System structure is one of the prime government commercial schemes. Low economical group and people below scarcity line use this amenities provided by the government. Due to deception appear in a chain, such amenities do not reach to the needy people. This happens because in the existing system all the work done by physically. To computerize or automate this physical job there is no any specific unreasonable technology or tools involved. Due to this, system facing two problems firstly weight of the material that is given to the people may be inaccurate or imprecise and secondly, at the end of the, illegal wrong entries in the inventory of the shop about the amount of the material given to the consumers. In this work we will describe a blockchain technology-based prototype that can be used in a small website. There are presently many fraud activities and corruptions taking place in the food supply schemes present as it sometimes does not reach the poor or the other sections of the society. This paper focuses on developing blockchain prototype that is used to record all the transactions/records and log all these transactions. A simple end-to-end web based app of this kind of the blockchain prototype can be built that has most of the features and functionalities to carry out all kinds of the transactions between the central government, state government, the district office, ration shop/and the customers, are recorded in the system. The user of the system can view the transactions of any part of the public distribution system. The project have some features that is guaranteed to provide the most important aspect that is, the security using the concept of blockchain

# Contents

<b>1</b>	<b>Introduction</b>	<b>1</b>
1.1	Background . . . . .	1
1.2	Relevance . . . . .	1
1.3	Project Undertaken . . . . .	2
1.3.1	Problem Overview . . . . .	2
1.3.2	Objectives of project . . . . .	2
1.3.3	Motivation . . . . .	2
1.4	Organization Of Project Report . . . . .	3
<b>2</b>	<b>Background</b>	<b>4</b>
2.1	Background . . . . .	4
<b>3</b>	<b>Software Requirements Specification</b>	<b>7</b>
3.1	Introduction . . . . .	7
3.1.1	Project Scope/Feasibility Study . . . . .	7
3.1.2	User Classes and Characteristics . . . . .	8
3.1.3	Assumptions and Dependencies . . . . .	9
3.2	Functional Requirements . . . . .	9
3.2.1	System Feature 1(Functional Requirement) . . . . .	9
3.2.2	System Feature 2 . . . . .	10
3.3	External Interface Requirements . . . . .	11
3.3.1	User Interfaces . . . . .	11
3.3.2	Hardware Interfaces . . . . .	11
3.3.3	Software Interfaces . . . . .	11
3.3.4	Communication Interfaces . . . . .	11
3.3.5	Nonfunctional Requirements . . . . .	11
3.3.6	Performance Requirements . . . . .	12
3.3.7	Safety Requirements . . . . .	12
3.3.8	Security Requirements . . . . .	12
3.3.9	Software Quality Attributes . . . . .	12
3.4	System Requirements . . . . .	12
3.4.1	Database Requirements . . . . .	13
3.4.2	Software Requirements (Platform Choice) . . . . .	13
3.4.3	Hardware Requirements . . . . .	13
3.5	Analysis Models: SDLC Model to be applied . . . . .	13
3.6	Plan of Project Execution . . . . .	14
<b>4</b>	<b>System Design</b>	<b>15</b>
4.1	System Architecture . . . . .	15
4.2	Implementation Constraints . . . . .	16
4.3	Data Flow Diagrams . . . . .	17
4.3.1	DFD Zero . . . . .	17
4.3.2	DFD Multi level . . . . .	18

4.4	UML Diagrams . . . . .	19
4.4.1	Class Diagram: . . . . .	19
4.4.2	Activity diagram . . . . .	20
4.4.3	Use Case . . . . .	21
4.4.4	Sequence Diagram . . . . .	22
<b>5</b>	<b>Implementation</b>	<b>23</b>
5.1	Introduction . . . . .	23
5.2	Tools and Technologies Used . . . . .	23
5.2.1	Java SE 8 . . . . .	23
5.2.2	Eclipse . . . . .	23
5.2.3	Apache Tomcat . . . . .	23
5.2.4	MySQL . . . . .	24
5.2.5	HeidiSQL . . . . .	24
5.3	Methodologies/Algorithm Details . . . . .	24
5.3.1	Hash Generation Algorithm . . . . .	24
5.3.2	Protocol for Peer Verification . . . . .	24
5.3.3	Mining Algorithm For Hash Creation . . . . .	25
5.4	Verification and Validation for Acceptance . . . . .	25
5.4.1	Communication Failure . . . . .	25
5.4.2	Register Phase Validation . . . . .	25
5.4.3	Login Phase Validation . . . . .	27
<b>6</b>	<b>Results and Evaluation</b>	<b>28</b>
6.1	Screen shots . . . . .	28
6.1.1	Admin Login Page . . . . .	28
6.1.2	Admin Add Shop Register Page . . . . .	28
6.1.3	Admin Add Shop Register Page . . . . .	29
6.1.4	Admin Add Distributes Register Page . . . . .	30
6.1.5	Distributes Login Page . . . . .	31
6.1.6	Distributes Add Product Page . . . . .	31
6.1.7	Shop Login Page . . . . .	32
6.1.8	Shop Add Register Page . . . . .	32
6.1.9	Shop Update Product Page . . . . .	33
6.1.10	User Login Page . . . . .	33
6.1.11	User Get Product Page . . . . .	34
6.1.12	Results(Peer to Peer Connection) Page . . . . .	34
6.1.13	Results(Complete Transaction) Page . . . . .	35
<b>7</b>	<b>Conclusions and Future Work</b>	<b>36</b>
7.1	Conclusions . . . . .	36
7.2	Future Scope . . . . .	36
	<b>REFERENCES</b>	<b>37</b>

# List of Figures

4.1	Proposed System Architecture . . . . .	15
4.2	Data Flow Diagrams . . . . .	17
4.3	DFD Multi level . . . . .	18
4.4	Class Diagram . . . . .	19
4.5	Activity diagram . . . . .	20
4.6	Use Case . . . . .	21
4.7	Sequence Diagram . . . . .	22
5.1	Registration Phase . . . . .	26
5.2	Login Phase . . . . .	27
6.1	Admin Login Page . . . . .	28
6.2	Admin Add Shop Register Page . . . . .	29
6.3	Admin Add Shop Register Page . . . . .	30
6.4	Admin Add Distributes Register Page . . . . .	30
6.5	Distributes Login Page . . . . .	31
6.6	Distributes Add Product Page . . . . .	31
6.7	Shop Login Page . . . . .	32
6.8	Shop Add Register Page . . . . .	32
6.9	Shop Update Product Page . . . . .	33
6.10	User Login Page . . . . .	33
6.11	User Get Product Page . . . . .	34
6.12	Results(Peer to Peer Connection) Page . . . . .	34
6.13	Results(Complete Transaction) Page . . . . .	35

# CHAPTER 1

## Introduction

### 1.1 Background

In existing systems, there are security available in public distribution system websites yet there are many number of nodes could not be connected at the same time and also the major point is that several fraud activities occur in these sites. It is mainly due to the reason that the exact transactions done by the lower levels (till the ration shops) cannot be viewed by the government. So, this drawback appears to as a chance for the frauds to manipulate the public distribution system. In the paper “Blockchain prototype for E-governance”, the blockchain prototype built is currently deployed on the local testrpc network, which runs only one system and acts as a single node. This prototype can be modified to incorporates a network of nodes of the blockchain the consists of governments, fair price shops and customers acting as nodes, which can be used by the government. The concept of the blockchain network perspective, that is, the concept of miners that is used to mine all the transactions is not used in their system. The central government has developed a website for the public distribution system .However this system is not free from its limitations. Large number of bogus cards are issues using which the middlemen and the fps owner sells the grains to the open market. This transactions are not recorded and this fraud activities cannot be viewed by the upper levels(central government, state government and district offices).There different cards issued for the people belonging to BPL and the people above the poverty line. Each person must get the allowable quantity as said by the government. However in the current system, people do not get the entitled amount of grains from the ration shops. All of these fraud activities are not transparent. There exists public distribution systems like the “Agent Based Simulation model and unique Identification Based Empirical Model” introduced by N.Hitaswi and K.Chandrasekaran, had an insight into the problems of duplicate and bogus ration cards. However, it did not provide the central government to view all the transactions occurring in the lower levels [9]. S.Kalpanadevi, S.Sukumar, K.Gopinathan and P.Naveen kumar, had developed a public distribution system that used an embedded system technology where they provided fingerprint detection for security. It had limitations due to high cost and problems in data storage (which could be easily managed in our proposed system).

### 1.2 Relevance

A System has represented by a 5-different phases, each phase works with own dependency System  $S = (Q, \Sigma, \delta, q_0, F)$  where

- $Q$  is a finite set of states.
- $\Sigma$  is a finite set of symbols called the alphabet.
- $\delta$  is the transition function where  $\delta : Q * \Sigma \rightarrow Q$



- $q_0$  is the initial state from where any input is processed ( $q_0 \in Q$ ).
- $F$  is a set of final state/states of  $Q$  ( $F \subseteq Q$ ).

All (n) data nodes will return 1 when each have the same blockchain

$Q$  = initial transactional data with genesis block

$\Sigma = \{\text{SHA-256, ConsensusVal, Mining}\}$

$\delta$  = Validate all server ( $S_1 \subseteq S_2 \subseteq S_3 \subseteq S_4$ ).

all server validation process  $q_0$  = Initial transaction  $T[0]$

$F = \{\text{Commit Trans, GetHistoryRecord}\}$

**State =**

1 :if all chains are validate or same

0 :if any  $t(n)$  server consist the invalid chain

**Set dependency**

Sys= {Phash, Tdata, Chash}

$$NodesChain[Nodeid, Chain] \sum_{(i=1)}^n (GetChain) \quad (1.1)$$

Get blockchain from each node and validate with each other.

## 1.3 Project Undertaken

### 1.3.1 Problem Overview

In the proposed research work to design and implement a system for Ration Distribution Management data, where user can store all information in single blockchain without any Trusted Third Party (TTP) in distributed computing environment. The system eliminates data integrity, privacy as well as end user discrepancies.

### 1.3.2 Objectives of project

- To design an approach for public ration distribution where system can store all historical transactional data into block chain manner
- To develop a custom blockchain for proposed public system, that end user can access and view entire data publically
- To develop an own smart contract as well custom mining policy to achieve the efficiency into the system.
- To develop a consensus algorithm for proof of validation between P2P decentralized networks, for data security and eliminate different network attacks.
- To explore and validate how proposed system provides, beneficial influence than classical ration distribution system.

### 1.3.3 Motivation

The world is changing incredibly fast, and we are not all aware of it. Block chain technology and crypto currencies are an irreversible advancement that is disrupting established industries and the ways in which we interact financially. For that reason, I believe understanding and being aware of this block chain wave is incredibly important. The existing systems work as centralized architecture in database system. Large data storage at the required of decentralized data storage

as well as information system. The different attack issues in centralized database architectures, There are no automatic attack recovery in central data architectures, The decentralized architecture provides the automatic data recovery from different attacks. After the analysis of this system we move to develop the decentralized system architecture, and fog computing provide parallel processing in distributed environment.

## 1.4 Organization Of Project Report

The material presented in this project report is organized into 7 chapters.

Chapter 1 After this introductory chapter.

Chapter 2 titled “Background” provides more information about an overview of the topic and also presents the literature survey pertaining to the topic.

Chapter 3, titled “Specification” specifies the software and hardware requirements needed to implement the project.

After this.

Chapter 4, “System Design” provides the architectural design, design and implementation constraints, data design and the various UML diagrams required to understand the project.

Chapter 5, “Implementation” specifies the algorithms used and the till-date implementations.

Chapter 6 :Results and Evaluation.

Finally Chapter 7, “Conclusion And Future Scope” concludes the topic and provides future scope of the project.

## CHAPTER 2

# Background

## 2.1 Background

Smart Contracts [1] Also called crypto-contract, it is a computer program used for transferring / controlling the property or digital currents in specific parties. It does not only determine the terms and conditions but may also implement that policy / agreement. These smart contracts are stored on block-chain and BC is an ideal technology to store these contracts due to the ambiguity and security. Whenever a transaction is considered, the smart-contract determines where the transaction should be transferred / returned or since the transaction actually happened.

Currently CSIRRO team has proposed a new approach to integrate BlockOn IOT with [2]. In its initial endeavor, he uses smart-home technology to understand how IOT can be blocked. Block-wheels are especially used to provide access control system for Smart-Devices Transactions located on Smart-Home. Introducing BC technology in IOT, this search again provides some additional security features, however, every mainstream BC technology must have a concept that does not include the concept of comprehensive algorithms. Moreover, this technology can not provide a general form of block-chain solution in case of IOT usage.

According to IlyaSukhodolski. The AI [3] system presents a prototype of multi-user system for access control over datasets stored in incredible cloud environments. Like other unreliable environments, cloud storage requires the ability to share information securely. Our approach provides access control over data stored in the cloud without the provider's investment. Access Control Mechanism The main tool is the dynamic feature-based feature-based encryption scheme, which has dynamic features. Using Blockchain based decentralized badgers; Our systems provide an irrevocable log for accessibility requests for all meaningful security incidents like large financing, access policy assignment, alteration or cancellation. We offer a set of cryptographic protocols that make the secret or secret key of cryptographic operation confidential. The hash code of the sifter text is only transmitted by the block on laser. Our system has been tested on prototype smart contracts and tested on IteriumBlockchain platforms.

According to Huehuangenet. AI [4] they offer a blockchain and a MedRec-based approach by enabling encryption and attribute based authentication to enable secure sharing of healthcare data. By applying this approach: 1) The fragmented EHR fragment of all patients can be seen as a complete record and can be safely stored against tampering; 2) The authenticity of patients' EHR can be verified; 3) Flexible and finer access control can be provided and 4) it is possible to maintain a cleared audit trail.

According to VipulGoyalet.AI [5] develops new cryptosystems to share encrypted data properly, which we call key-policy attribute-based encryption (KPABE).In our cryptosystem, Cefhettetis labeled with a set of properties and controls that it connects to private key access configurations that a user can decrypt the encryption. We display the utility of our product to share audit log information and broadcast encryption. Our creation supports private key providers, which subscribe to categorized identification-based encryption (HIBE).

Hao Wang et Mate AI [6] They offer a secure electronic health record (EHR) system based on

special-based cryptocooccurs and blockchan technology. In our system, we use attribute-based encryption (ABE) and identity-based encryption (IBE) to encrypt medical data and to use identity-based signature (IBS) to apply digital signatures. . In order to obtain various functions of ABI, IBE and IBS in crypto, we present a new cryptographic primitive, it is called a joint feature-based / identity-based encryption and signature (C-AB / IB-ES). It simplifies system maintenance and does not require the installation of separate cryptographic system for various security requirements. In addition, we use blockconne techniques to ensure the integrity and inspection of medical data. Finally, we offer a demonstration application for medical insurance business.

According to Yan Michalevskyet. Al [7] system introduces the first practical decentralized ABE scheme with proof of policy-hiding. Our creation is based on the basic encryption of decentralized internal product, which is an encryption strategy launched in this paper. This ABB scheme supports results, disputes, and threshold policies, which protect the access policies of those parties that are not authorized to decrypt content. In addition, we handle the receiver's privacy issue.

Using our plan with Vector Commitment, we hide a complete set of attributes presented by the individual with the recipient; Just disclose the feature that regulates the authority. Finally, we propose random-polynomial encoding that immerses this scheme in the presence of corrupt officials. Al [8] they successfully address these issues by offering a clearepolicy feature-based data sharing plan with direct cancellation and keyword search. In the proposed scheme, the non-terminated users' private key is not required to be updated during the cancellation of direct revocation of features. In addition, a keyword search has been realized in our plan, and the search is stable with the increase in time features. Specifically, the policy is hidden in our plan, and therefore, the privacy of users is preserved. Our security and performance analysis show that the proposed plan can deal with security and efficiency concerns in cloud computing.

According to SarmadullahKhanet. Al [9] embedded power transactions in blockchain are based on their defined characteristics through the signature of many manufacturers. These signatures have been verified and customers are satisfied with the features that do not open any information that meet those features. The public and private key manufacturers have been created for these customers and using this key ensures that the support process is authorized by customers. There is no central authority required in this perspective. To protest against collision attacks, the makers are given secret pseudo-functional work seeds. Comparative analysis shows the efficiency of the proposed approach to existing people.

According to Ruuguet. Al [10] To guarantee the validity of the EHR surrounding the block channel, he has submitted a special-based signature scheme with multiple officials, in which the patient supports the message according to the specifications, but there is no evidence that he does not have any other information. In addition, there are many officers without generating a reliable individual or a central person in order to generate and deliver a public / private key, which avoids the escrow problem and adapt to the mode of data storage distributed in the Block Block. By sharing the secrecy of the secret pseudo-festive festivals in the authorities, this protocol opposed the attack of N-1 affiliated with officials. Under the computational BillineDiffie-Hellman concept, we also formally demonstrate that, in relation to the specialty-signatory's enforceability and complete privacy, this specialty-based signature scheme is safe in random decorative models. Comparison shows the efficiency and qualities among the proposed methods and methods in other studies.

According to Damiano Di Francesco Maesaet. al [11] Access control systems in computer security are used to regulate access to important or valuable resources. To use such resources, the rights of subjects are usually expressed through access control policies, which are evaluated on request from time to time regarding access to current reference. They proposes a new approach based on blockchain technology so that the policies expressing the right to access a resource can be published and allow distributed transfer of such rights among users. In their proposed protocol, policies and rights appear in public on the block exchange; as a result, any user can at any time connect the policy with any resource and the subjects who currently have the right to access the resource. This solution allows distributing an audit policy, preventing a party from fraud by the

rights granted by an enforceable policy. They also show a possible working implementation on the basis of XACML policies posted on Bitcoinblockchain. The main advantage of this approach is that the policy is published on blockchain, thus being visible to the topics of the scenario and access rights can be transferred from one user to another through the blockchain transaction. The approach has been validated through context implementation based on bitcoin.

According to MikkoKiviharju et.al [12] Present, tightly connected, highly dynamic and distributed computing environments presents big challenges for information security. Entities with the most sensitive assets and the most challenging safety requirements, daily struggle to protect their information systems against both threats both inside and outside. Specific examples include government, financial institutions and healthcare. The initial model developed for the most valuable asset in the high-value (computing) environment was actually originated within the military from the United States, the United Kingdom and the other Cold War era by the associated forces. As a result of the theoretical work there became a concept called multi-level security (MLS). The level of assurance required to implement the separation of different security domains for MLS system was very high: only complete physical and electrical or cryptographic separation is sufficient. In this work, we researched the viability of cryptographically applied role-based access control using a state-of-the-art functional cryptography scheme in a distributed, multi-level security setting. The concept of cryptographic access control implementation is somewhat understandable.

According to Qingsu He et.al [13] as a new technology, Blockchain has attracted the attention of industry and academics to share data in organizations. Many block-based data sharing applications, such as Internet of Things Device Management, require encryption-protected data on dual capabilities-privacy-protection access services. On one hand, they need to keep sensitive data private, so that others cannot trace sensitive data stored in the block. On the other hand, they need to support the access control properly both from the time and the characteristics of the users.

According to YogachandranRahulamathavan et.al [14] The Internet of Things (IoT) has penetrated deep into our lives and the number of IoT devices per person is expected to grow considerably in the next few years. Due to the characteristics of IoT devices (i.e., low power and low battery), sophisticated safeguards are required for the use of these devices in important applications. Academicians and industry researchers have now taken advantage of the Blockchain concept to gain security in IoT applications. To preserve the privacy of transaction data using blockchain architecture attribute-based encryption technology. This is the first approach that combines state-of-the-art encryption technology with Blockchain technology. The simplicity and finer nature of attribute-based encryption controls, which can be viewed and used Transaction data.

According to Axin Wu et.al [15] Attribution-based encryption, especially ciphertext-policy attribute-based encryption, plays an important role in data sharing. In the data sharing process, the secret key does not contain specific information of users, who can share their secret key with other users without looking for a profit. In addition, the Specialty Authority can generate a secret key from a feature set. If the secret key is misused, it is difficult to determine whether the private key used in abuse comes from users or the specialty authority. In addition, the Access Control Structure usually leaks sensitive information in a distributed network, and the efficiency of attribute-based encryption is a hindrance to its applications. Given the efficiency of ABE, specialty protection based on privacy protection and secret key misuse, blockchain-enabled and confidential-protection characteristic-based encryption is proposed.

## CHAPTER 3

# Software Requirements Specification

### 3.1 Introduction

Recently, the Public Ration Distribution System structure is one of the prime government commercial schemes. Low economical group and people below scarcity line use this amenities provided by the government. Due to deception appear in a chain, such amenities do not reach to the needy people. This happens because in the existing system all the work done by physically. To computerize or automate this physical job there is no any specific unreasonable technology or tools involved. Due to this, system facing two problems firstly weight of the material that is given to the people may be inaccurate or imprecise and secondly, at the end of the, illegal wrong entries in the inventory of the shop about the amount of the material given to the consumers. In this work we will describe a blockchain technology-based prototype that can be used in a small website. There are presently many fraud activities and corruptions taking place in the food supply schemes present as it sometimes does not reach the poor or the other sections of the society. This paper focuses on developing blockchain prototype that is used to record all the transactions/records and log all these transactions. A simple end-to-end web based app of this kind of the blockchain prototype can be built that has most of the features and functionalities to carry out all kinds of the transactions between the central government, state government, the district office, ration shop/and the customers, are recorded in the system. The user of the system can view the transactions of any part of the public distribution system. The project have some features that is guaranteed to provide the most important aspect that is, the security using the concept of blockchain.

A Software requirements specification document describes the intended purpose, requirements and nature of software to be developed. It also includes the yield and cost of the software. A Software Requirements Specification (SRS) is describes the nature of a project, software or application. In simple words, SRS is a manual of a project provided it is prepared before you kick-start a project/application. A software document is primarily prepared for a project, software or any kind of application.

#### 3.1.1 Project Scope/Feasibility Study

##### Project Scope

Blockchain, the digital ledger technology that can securely maintain continuously growing lists of data records and transactions, has the power to potentially transform e ration, according to industry experts. By simplifying and expediting the way the e ration industry processes data in such areas as revenue cycle management, Ration Distribution Management data interoperability and supply chain validation, blockchain has the power to dramatically .reduce back-office data input and maintenance costs and improve data accuracy and security. This scope of proposed work in below data-driven areas:

- Can blockchain enable multiple e ration transactional records to securely link and be accessible across non-affiliated provider organizations to improve care coordination?
- If any block chain invalid during the validation of data servers, then system will automatically recover whole blockchain using majority of servers
- Will blockchain technology enable system to more easily, effectively and securely gain access to their own e ration transactions records?

### **Feasibility Study**

- The feasibility study is major factor which contributes to analysis of system. In earlier stages of
- S/W development, it is necessary to check weather system is feasible or not. Detail study was carried out to check work ability of proposed system, so the feasibility study is system proposal regarding to its workability, impact on organization, ability to meet user requirements and effective use of resources.

### **Technical Feasibility**

- Technical study is the study of hardware requirements and software requirements
- Considering all specified requirements, the project is technically feasible.

### **Economic Feasibility**

- The economic feasibility will review the expected cost to see it they are in-line with the projected budget or if the project has an acceptable return on investment.

### **Operational Feasibility**

- Operational feasibility is a measure of how well a proposed system solves the problems, and takes advantages of the opportunities identified during scope definition .

### **Time Feasibility**

- Similar to economic feasibility, a rough estimate of the project schedule is required to determine if it would be feasible to complete the systems project within a required timeframe

## **3.1.2 User Classes and Characteristics**

- To design approach for Ration Distribution Management transaction where system store all historical data into block chain manner
- To create a fog computing environment hierarchy for parallel data processing for end users applications.
- To design and implement own SHA family block for whole block chain.
- To design and implement a new mining technique for generate new block for each transaction.
- To implement verification algorithm which can validate each peer on every access request.
- To implement a verification algorithm which can validate each peer on every access request to eliminate different network attacks.

### 3.1.3 Assumptions and Dependencies

1. The new nodes agree with the transaction of block which is sending by the old nodes
2. The new nodes don't agree with the transaction of block which is sending by the old nodes.
3. The old nodes agree with the transaction of block which is sending by the new nodes.
4. The old nodes don't agree with the transaction of block which is sending by the new nodes

## 3.2 Functional Requirements

- System must be fast and efficient
- User friendly GUI
- Reusability
- Performance
- System Validation input
- Proper output

### 3.2.1 System Feature 1(Functional Requirement)

1. System must validate the previous block before commit block.
2. User can access the data over the internet 24\*7.
3. If any block has changed by third party attacker or unauthorized user, it must show during transaction current blockchain is invalid.
4. It can recover the invalid blockchain using other data nodes, with the help of majority of trustiness.
5. The node or user who wants to initiate a transaction would record and broadcasts the data to the network.
6. The node or user who receives the data verifies the authenticity of the data received in the network. Then the verified data is stored to a block.
7. All nodes or users in the network validate the transaction by executing either the proof of work algorithm or the proof of stake algorithm to the block that needs validation.
8. Consensus algorithm used by the network will store the data to the block that is added to blockchain. And all nodes in the network admit the respective block and extend the chain base on the block.



### 3.2.2 System Feature 2

**Decentralization:** Decentralization is the dispersion of functions and controls from a central authority to all the units involved. In blockchain a centralized authority is not available. Instead, every blockchain user (miner) is provided with a copy of the transaction ledger and a new block is added by validating transaction by the miners involved. In a decentralized environment the network operates on a peer-to-peer (user-to-user) basis. The researchers in use this element of blockchain as one of the major aspects in developing Ethereum digital currency.

**Consensus model(s) :** The consensus model(s) help preserve the sanctity of data recorded on blockchain. In , it is reported that various consensus mechanisms and issues could result when the consensus mechanism fails including blockchain forks, consensus failures, dominance issues, validating nodes and deficient performance of the blockchain network. A consensus protocol has three properties based on applicability and efficiency:

- a). **Safety:** A consensus protocol must be safe and consistent, meaning that all nodes should produce the same output which is valid in accordance with the protocol rules.
- b). **Liveness:** A consensus protocol promises liveness of all non-faulty nodes to yield a value.
- c). **Fault Tolerance:** A consensus protocol provides tolerance while providing recovery to a failure node participating in consensus.

**Transparent:** The blockchain network routinely checks in with itself every ten minutes in order to self-audit the ecosystem of a digital value, which reconciles transactions that happen in ten-minute intervals. A collection of these transactions is referred to as a “block”. Two resulting properties, transparency and inability of corruption, are generated.

**Open source :** A decentralized and closed-source application needs user to trust that the application is decentralized and the data cannot be accessed from a central source. Closed-source applications act as a barrier to adoption by users. The repugnance to a closed source network is noticeable when the application is intended to receive, hold, or transfer user funds. Even though it is possible to launch a closed source decentralized application, the level of the difficulty to achieve the desired result would be catastrophic, making it obvious for the users to favor open source participants. Open sourcing a decentralized application modifies the structure of business practices who used to favor the Internet as the common denominator.

**Identity and access :** The identity and accessibility of a blockchain are related to three main criteria including public or permission less, private or permission, and consortium. These criteria of blockchain was discussed in detail in. A private blockchain restricts the users from having the authority to validate block transactions and create smart contracts. This is appropriate for the traditional businesses and governance models. Public block chains are designed to cut the middleman out in transactions while keeping the security intact. In public blockchains any user with access to internet can join the network by participating in block verifications and creating smart contracts. Consortium blockchains are partly private and allows a few predetermined selective nodes to have full control. It is a substitute for allowing any random user with an internet connection to verify transactions. The platform like private blockchain provides efficiency and privacy of transactions.

**Autonomy :** The main objective of blockchain technology is to switch the trust from one centralized authority to the whole network without interference. Every node in the blockchain system can transfer and update information securely. A decentralized autonomous organization (DAO), which is frequently categorized as decentralized autonomous corporation (DAC), is explained as an organization that follows a set of rules prearranged as computer programs and termed as smart contracts. The transaction record and smart contract details are maintained as blocks in blockchain.

**Immutability :** Immutability is something that is unchanging over a period. In the context of blockchain, immutability is relevant to data or information stored in the blocks. Once the data or information is written in a block of blockchain nobody can alter it. This is highly essentially beneficial for auditing data. On one hand, the provider of data can verify that the data is se-

cure, efficient, and has not been tampered with or altered. On the other hand, the recipient of data is confident that data is authentic and unaltered. The immutability element of blockchain is extremely beneficial for databases used in financial transactions since the records are reserved forever and cannot be changed unless somebody takes control of more than 51% of the nodes in the network simultaneously.

**Anonymity :** explains the anonymity element of blockchain technology. The blockchain address of a miner is necessary for this element and no other detail is required, resulting in anonymity resolving trust issues. Anonymity of an entity inside a set of entities is not distinguishable. In a communication system, the anonymity set can be divided into two sets: the sender sets and the recipient sets.

## 3.3 External Interface Requirements

### 3.3.1 User Interfaces

The Interface will be in the form of an application. It is designed to be functional and minimal in its styling. All options will be displayed in a menu based format. Web application will be used to setup the page layout and add minimal styling to make the interface user friendly.

### 3.3.2 Hardware Interfaces

A webserver will be required so that the students and the mess admin can connect to it to exchange information. The servers have a database to store all the data entries. The Server will have to have a high speed 1 Gigabit Ethernet connection to the college's local network.

### 3.3.3 Software Interfaces

The server will be hosted using Apache Tomcat Web server for deploy the logistic sine map service. It will also have a MySQL relational database. The main backend processing will be done using Java Server Pages (JSP) including connecting to and accessing the database and processing requests.

### 3.3.4 Communication Interfaces

The main communication protocol will be Hyper Text Transfer Protocol (HTTP). This will be used to transfer information back and forth from the client to the server. HTTP GET and POST will be used to send the information.

### 3.3.5 Nonfunctional Requirements

- Accessibility
- Capacity, current and forecast
- Compliance
- Response time
- Robustness
- Scalability
- Security

- Stability
- Supportability
- Testability

### **3.3.6 Performance Requirements**

The only way in which systems will meet their performance targets is for them to be specified clearly and unambiguously. It is a simple fact that if performance is not a stated criterion of the system requirements then the system designers will generally not consider performance issues. While loose or incorrectly defined performance specifications can lead to disputes between clients and suppliers. In many cases performance requirements are never ridged as system that does not fully meet its defined performance requirements may still be released as other consideration such as time to market. In order to assess the performance of a system the following must be clearly specified:

- Response Time
- Workload
- Scalability
- Platform

### **3.3.7 Safety Requirements**

This Specification shall be sufficient detailed to allow the design and implement to achieve the required safety integrity and allow an assessment of functional safety.

### **3.3.8 Security Requirements**

System security during the data transmission, when user upload image and download image from cloud server, then eliminate the third party un-authenticate access using proposed authentication mechanism.

### **3.3.9 Software Quality Attributes**

Product is portable; it can run between only two connected systems or a large Network of computers. Product is maintainable; i.e. in future the properties of the product can be changed to meet the requirements.

## **3.4 System Requirements**

- Processor:- Intel Pentium 4 or above
- Memory:- 2 GB or above
- Other peripheral:- Printer
- Hard Disk:- 500gb

### 3.4.1 Database Requirements

- It should be SQLite database on platform.
- Database must be integrated with key constraints
- It should be maintain the relational base on RDMS and normalization
- System will create database backup on periodic basis.
- It will execute all commands like DML, DDL and DCL as well as we required some security measurements for sql injection.

### 3.4.2 Software Requirements (Platform Choice)

Technologies and tools used in Policy system project are as follows Technology used:

#### Front End

- JDK 1.7 and onwards
- Internet Explorer 6.0/above
- Tool : Net beans 7.4 onwards
- Java

#### Back-End

- MYSQL 5.1
- Heidi SQL

### 3.4.3 Hardware Requirements

- System : i5 with 2.7 GHz.
- Hard Disk : 300 GB
- Monitor : 15 VGA Color.
- Mouse : Logitech.
- Ram : 4 GB

## 3.5 Analysis Models: SDLC Model to be applied

Agile SDLC model is a combination of iterative and incremental process models with focus on process adaptability and customer satisfaction by rapid delivery of working software product. Agile Methods break the product into small incremental builds. These builds are provided in iterations. Each iteration typically lasts from about one to three weeks. Every iteration involves cross functional teams working simultaneously on various areas like

- Planning
- Requirements Analysis
- Design
- Coding

- Unit Testing and
- Acceptance Testing.

At the end of the iteration, a working product is displayed to the customer and important stakeholders.

### 3.6 Plan of Project Execution

Milestones Table

Table 3.1: System Implementation Plan

Sr. No	Task Name	Begin date	End date	Remarks
1	Selecting project domain	15 July 2019	20 July 2018	Done
2	Understanding project need	21 July 2019	25 July 2019	Done
3	Understanding pre requisites	26 July 2019	30 July 2019	Done
4	Information Gathering	1 Aug 2019	30 Aug 2019	Done
5	Literature Survey	1 Sept 2019	15 Sept 2019	Done
6	Refine Project Scope	16 Sept 2019	18 Sept 2019	Done
7	Concept Development	19 Sept 2019	20 Sept 2019	Done
8	Planning and Scheduling	21 Sept 2019	23 Sept 2019	Done
9	Requirements analysis	24 Sept 2019	25 Sept 2019	Done
10	Risk identification and monitoring	26 Sept 2019	27 Sept 2019	Done
11	Design and modeling	28 Sept 2019	15 Oct 2019	Done
12	Design review and refinement	16 Oct 2019	20 Oct 2019	Done
13	GUI design	21 Oct 2019	20 Nov 2019	Done
14	Implementation	21 Nov 2019	15 Feb 2019	Done
15	Review and suggestions for Implementation	15 Mar 2020	20 Mar 2020	Done
16	Outcome assessment	21 Mar 2020	30 Mar 2020	Done
17	Testing and Quality Assurance	1 Apr 2020	10 Apr 2020	Done
18	Review and suggestions for Testing and QA	11 Apr 2020	15 Apr 2020	
19	Refined QA activities	16 Apr 2020	30 May 2020	

# CHAPTER 4

## System Design

### 4.1 System Architecture

The proposed figure 1 shows the overall execution of proposed system.

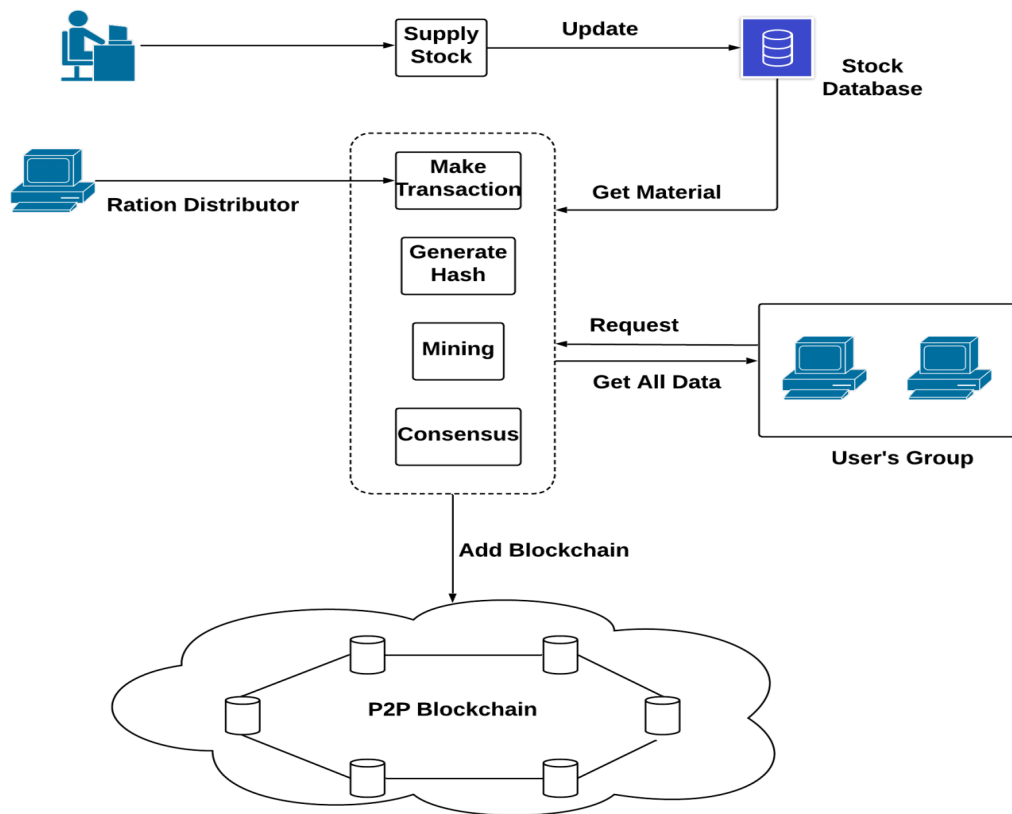


Figure 4.1: Proposed System Architecture

## 4.2 Implementation Constraints

1. We create a multiple e ration transnational data and stored all transnational data into multiple data nodes.
2. Each node will holds the specific block for each transaction.
3. Same block has replace for all nodes, and generates a valid block chain.
4. System will retrieve data from all data nodes and commit the transaction, it should be any kind of DDL, DML as well as DCL transnational query.
5. If any block chain invalid during the validation of data servers, then system will automatically recover whole blockchain using majority of servers.
6. We will address and eliminate the run time server attacks and recover it using own blockchain.

## 4.3 Data Flow Diagrams

### 4.3.1 DFD Zero

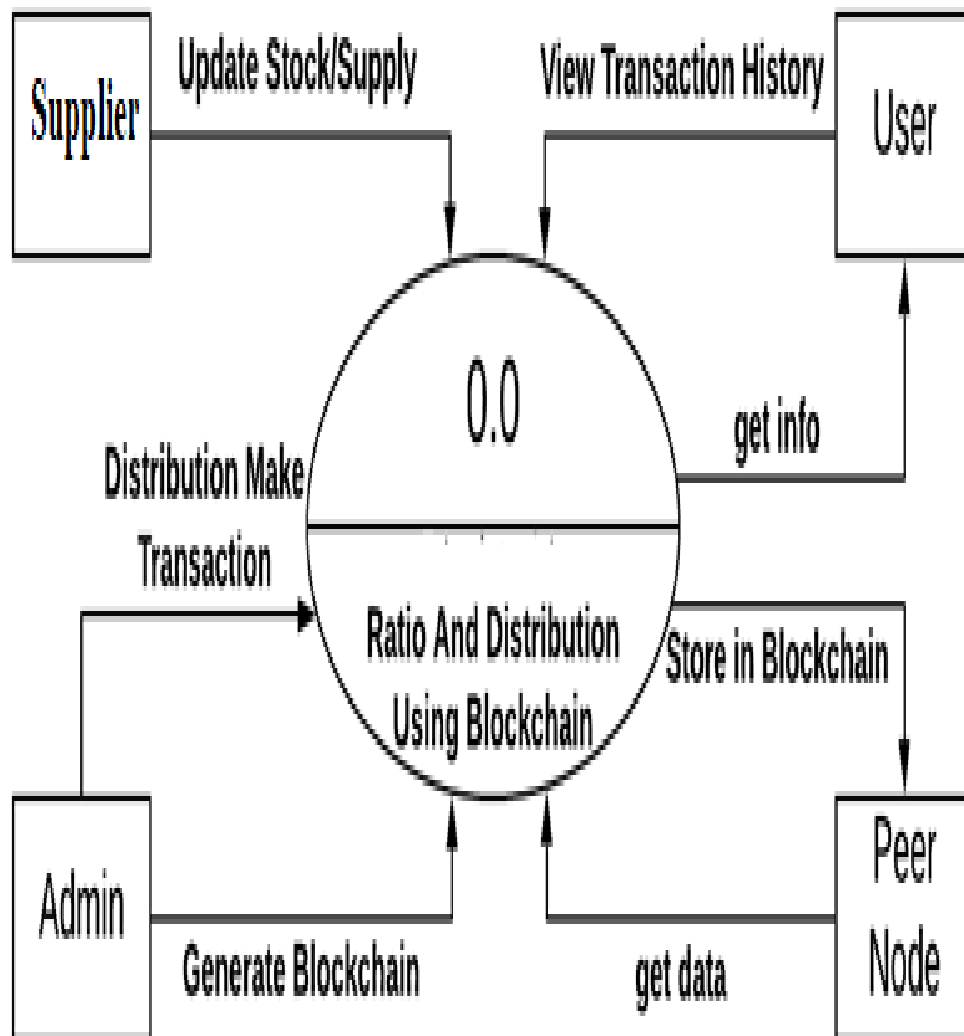


Figure 4.2: Data Flow Diagrams



### 4.3.2 DFD Multi level

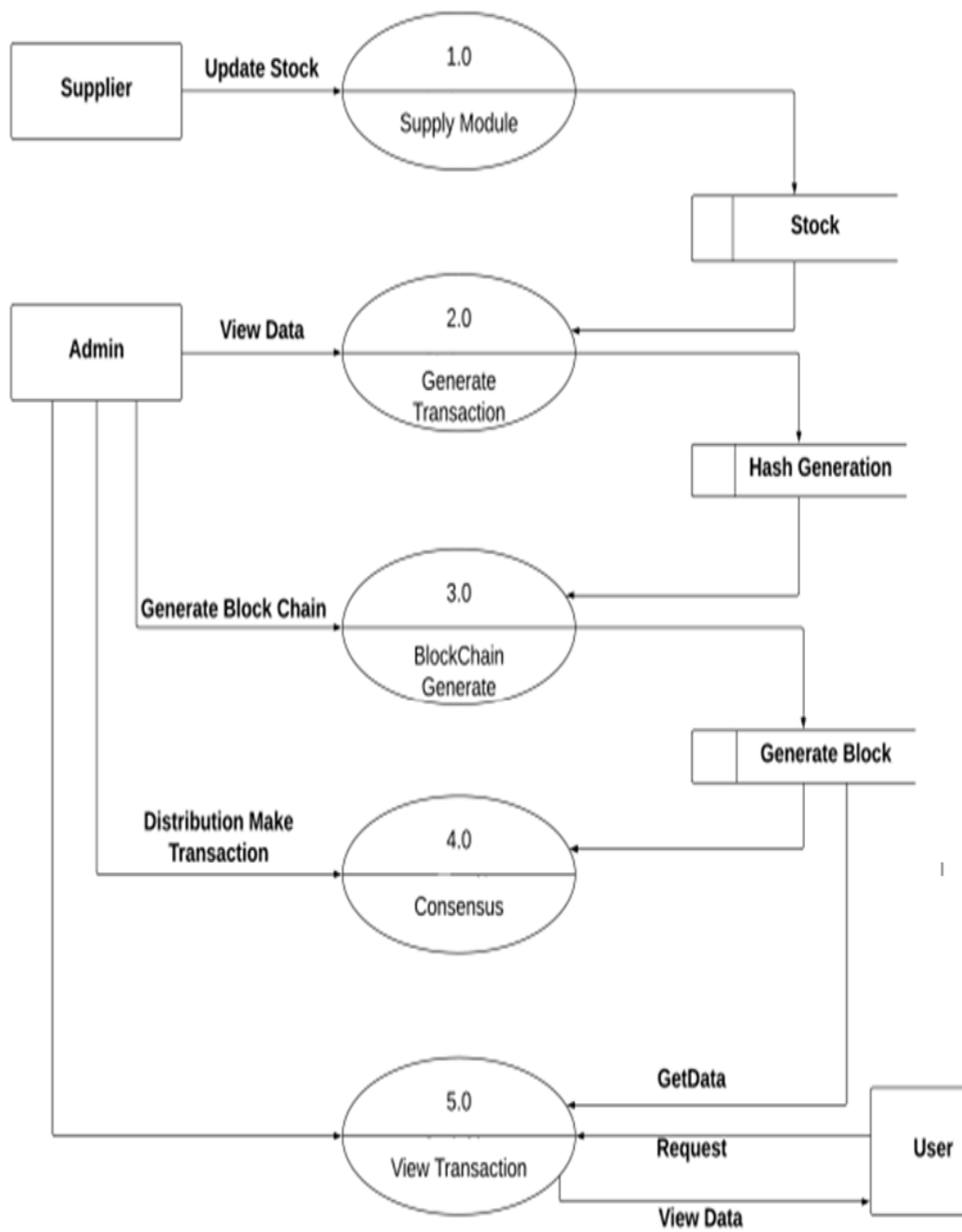


Figure 4.3: DFD Multi level

## 4.4 UML Diagrams

### 4.4.1 Class Diagram:

Class diagram is a static diagram. It represents the static view of an application. Class diagram is not only used for visualizing, describing, and documenting different aspects of a system but also for constructing executable code of the software application. Class diagram describes the attributes and operations of a class and also the constraints imposed on the system. The class diagrams are widely used in the modeling of object-oriented systems because they are the only UML diagrams, which can be mapped directly with object-oriented languages. Class diagram shows a collection of classes, interfaces, associations, collaborations, and constraints. It is also known as a structural diagram.

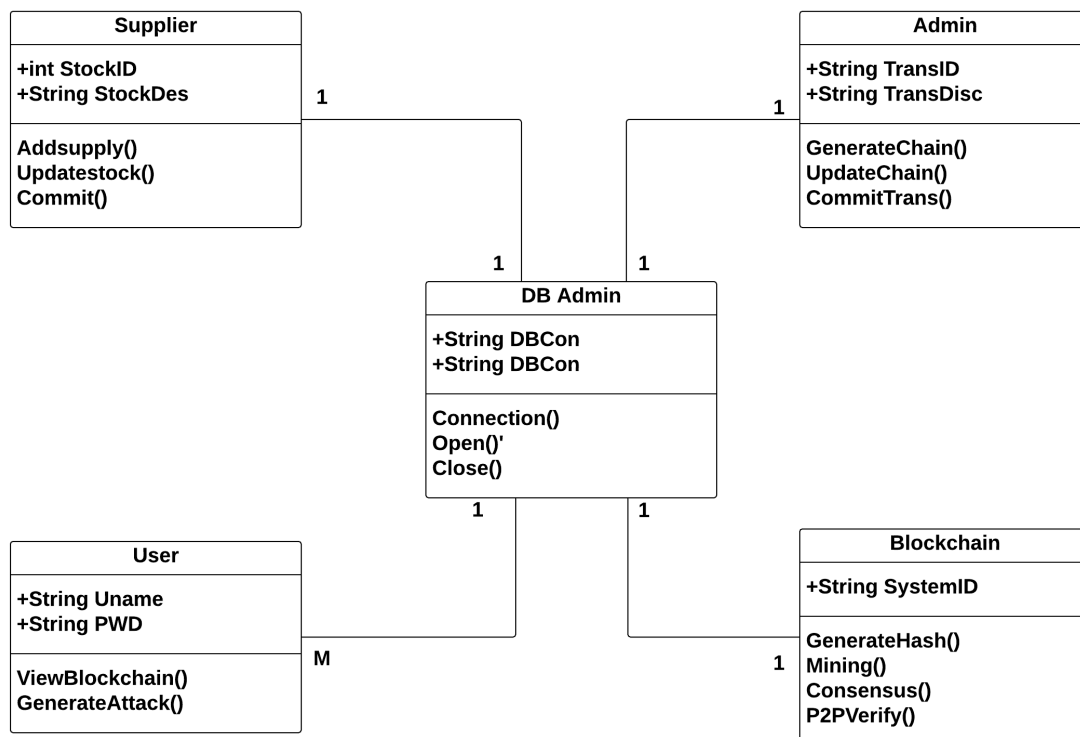


Figure 4.4: Class Diagram

#### 4.4.2 Activity diagram

Activity diagram is another important diagram in UML to describe the dynamic aspects of the system. Activity diagram is basically a flowchart to represent the flow from one activity to another activity. The activity can be described as an operation of the system. The control flow is drawn from one operation to another. This flow can be sequential, branched, or concurrent. Activity diagrams deal with all type of flow control by using different elements such as fork, join, etc.

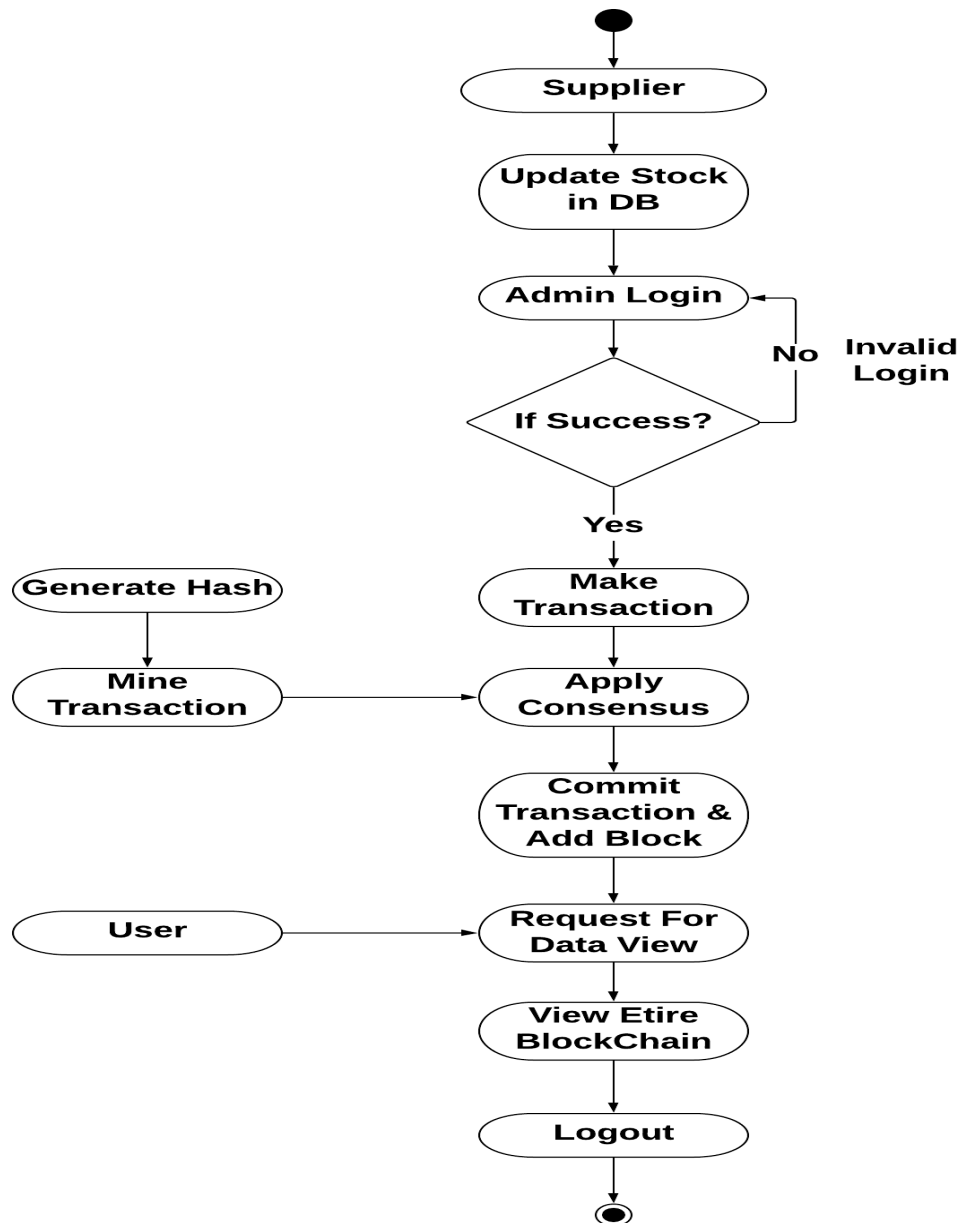


Figure 4.5: Activity diagram

### 4.4.3 Use Case

A use case diagram is a dynamic or behavior diagram in UML. Use case diagrams model the functionality of a system using actors and use cases. Use cases are a set of actions, services, and functions that the system needs to perform.

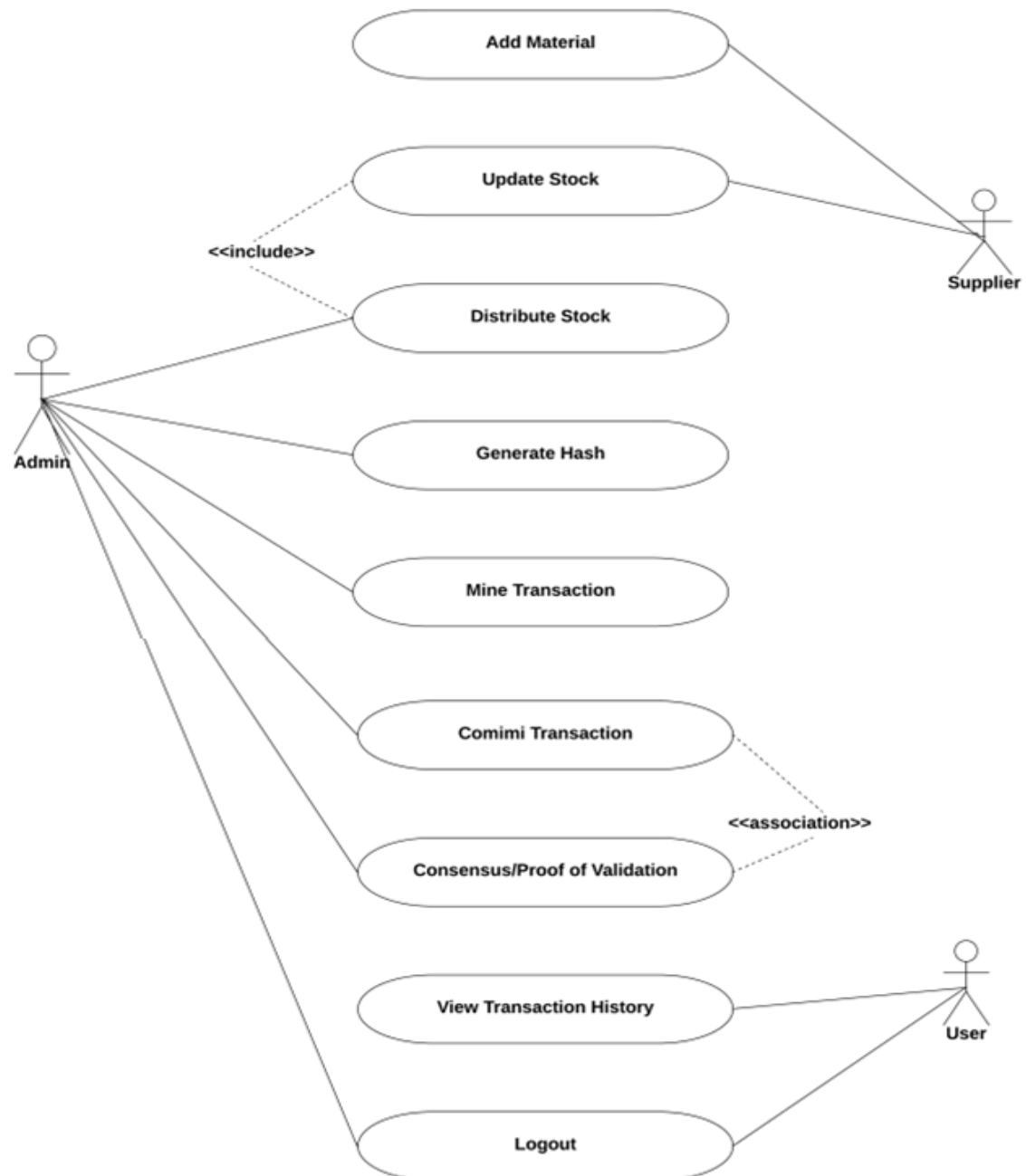


Figure 4.6: Use Case

#### 4.4.4 Sequence Diagram

A sequence diagram shows object interactions arranged in time sequence. It depicts the objects and classes involved in the scenario and the sequence of messages exchanged between the objects needed to carry out the functionality of the scenario. A sequence diagram shows, as parallel vertical lines (lifelines), different processes or objects that live simultaneously, and, as horizontal arrows, the messages exchanged between them, in the order in which they occur.

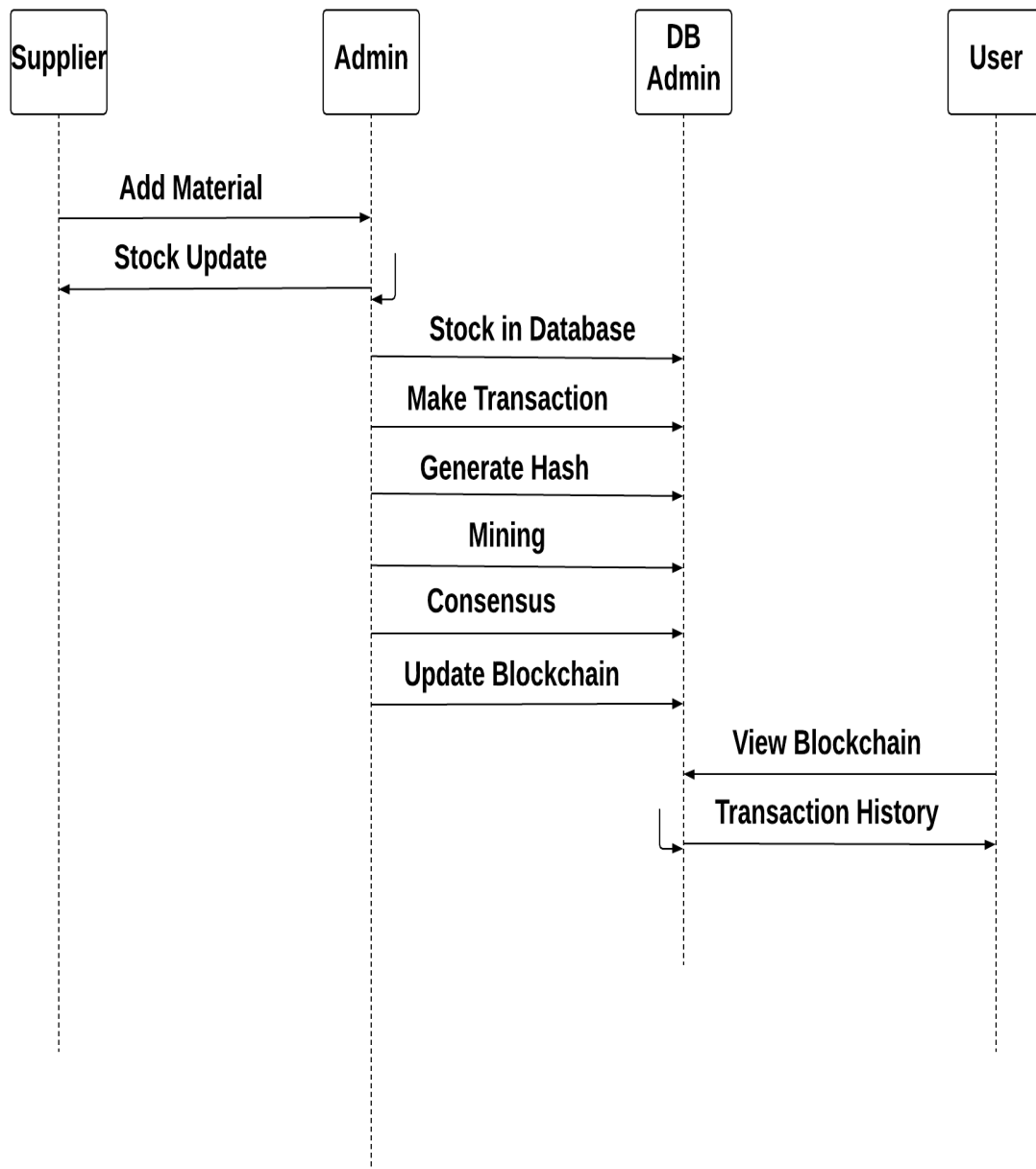


Figure 4.7: Sequence Diagram

## CHAPTER 5

# Implementation

### 5.1 Introduction

- **Admin:** The administrators generate the supply and add the information into the master stock, admin also can view the whole stock and transnational information.
- **User:** each user is the part of transaction where he can see total supply, distribution related information of our portal.
- **Distributor:** This is the important module in system, where any distribution has done by distributor, system generates the transaction and entire blockchain has validate at same time. Basically blockchain execute four different phases these are below

### 5.2 Tools and Technologies Used

#### 5.2.1 Java SE 8

Java Platform, Standard Edition (Java SE) lets you develop and deploy Java applications on desktops and servers. Java users the rich user interface, performance, versatility, portability, and security that today's applications require.

#### 5.2.2 Eclipse

Eclipse is an integrated development environment (IDE) used in computer programming, and in 2014 was the most widely used Java IDE in one website's poll. It contains a base workspace and an extensible plug-in system for customizing the environment. Eclipse is written mostly in Java and its primary use is for developing Java applications, but it may also be used to develop applications in other programming languages via plug-ins. The initial codebase originated from IBM VisualAge. The Eclipse software development kit (SDK), which includes the Java development tools, is meant for Java developers. Eclipse software development kit (SDK) is free and open-source software, released under the terms of the Eclipse Public License, although it is incompatible with the GNU General Public License.

#### 5.2.3 Apache Tomcat

The Apache Tomcat software is an open source implementation of the Java Servlet, JavaServer Pages, Java Expression Language and JavaWebSocket technologies. The Java Servlet, JavaServer Pages, Java Expression Language and Java WebSocket specifications are developed under the Java Community Process. The Apache Tomcat software is developed in an open and participatory environment and released under the Apache License version 2. The Apache Tomcat project is intended to be a collaboration of the best-of-breed developers from around the world. Apache

Tomcat software powers numerous large-scale, mission-critical web applications across a diverse range of industries and organizations.

### 5.2.4 MySQL

MySQL is an open-source relational database management system (RDBMS). The MySQL™ software delivers a very fast, multithreaded, multi-user, and robust SQL (Structured Query Language) database server. MySQL Server is intended for mission-critical, heavy-load production systems as well as for embedding into mass deployed software. MySQL is offered under two different editions: the open source MySQL Community Server and the proprietary Enterprise Server. MySQL Enterprise Server is differentiated by a series of proprietary extensions which install as server plugins, but otherwise shares the version numbering system and is built from the same code base.

### 5.2.5 HeidiSQL

HeidiSQL is free software, and has the aim to be easy to learn. "Heidi" lets you see and edit data and structures from computers running one of the database systems MariaDB, MySQL, Microsoft SQL or PostgreSQL. Invented in 2002 by Ansgar, with a development peak between 2009 and 2013, HeidiSQL belongs to the most popular tools for MariaDB and MySQL worldwide.

## 5.3 Methodologies/Algorithm Details

### 5.3.1 Hash Generation Algorithm

A hash is a function that meets the encrypted demands needed to solve for a blockchain computation. A hash, like a nonce or a solution, is the backbone of the blockchain network. Hashes are of a fixed length since it makes it nearly impossible to guess the length of the hash if someone was trying to crack the blockchain. A hash is developed based on the information present in the block header

**Input :** Genesis block, Previous hash, data d,  
**Output :** Generated hash H according to given data  
 Step 1 : Input data as d  
 Step 2 : Apply SHA 256 from SHA family  
 Step 3 : CurrentHash= SHA256(d)  
 Step 4 : Return CurrentHash

### 5.3.2 Protocol for Peer Verification

**Input :** User Transaction query, Current Node Chain CNode[chain], Other Remaining Nodes blockchain NodesChain[Nodeid] [chain],  
**Output :** Recover if any chain is invalid else execute current query  
 Step 1 : User generate the any transaction DDL, DML or DCL query  
 Step 2 : Get current server blockchain  
 Cchain  $\leftarrow$  Cnode[Chain]  
 Step 3 : For each  

$$NodesChain[Nodeid, Chain] \sum_{i=1}^n (GetChain)$$
 End for  
 Step 4 : Foreach (read I into NodeChain)  
 If (!equals NodeChain[i] with (Cchain))

```
Flag 1
Else Continue Commit query
Step 5 : if (Flag == 1)
Count = SimilaryNodesBlockchian()
Step 6 : Caculate the majority of server
Recover invalid blockchin from specific node
Step 7: End if
End for
End for
```

### 5.3.3 Mining Algorithm For Hash Creation

**Input :** Hash Validation Policy  $P[]$ , Current Hash Values  $hash\_Val$

**Output :** Valid hash

Step 1 : System generate the  $hash\_Val$  for  $i$ th transaction using Algorithm 1

Step 2 : if ( $hash\_Val.valid$  with  $P[]$ )

Valid hash

Flag =1

Else

Flag=0

Mine again randomly

Step 3 : Return valid hash when  $flag=1$

## 5.4 Verification and Validation for Acceptance

### 5.4.1 Communication Failure

Failure to communicate is the norm, rather than the exception. Data node example It can have three or more server machines, each of which is storing part file system data. Therefore, search valid and invalid and quick, automatic recovery from them is one of the main goals Block Chain

### 5.4.2 Register Phase Validation



Test#	Description	Test Inputs	Expected Results	Pass/ Fail	Test Priority (High/ Medium/ Low)	Defect Severity (High/ Medium/ Low)
1	Name shouldn't too long (up to 30 characters)	Name: aabbbbbbbbbbbbbbb bbbbbbbbbbbbbbbbbb bbbbbbbbbbbbbbbbbb bbbbbbbbbbbbbbbbbb bbbbbbbbbbbbbbbbbb bbbbbbbbbbbbbbbbbb bbbbbbbbbbbbbbbbbb bbbbbbbbbb	Inputs should not be accepted. It should display message "Enter valid Name"	Fail	High	Medium
2	Name shouldn't too short (up to 2 characters)	Name: aa	Inputs should not be accepted. It should display message "Enter valid Name"	Fail	High	Medium
3	Name shouldn't contain special symbols	Name: abc%	Inputs should not be accepted. It should display message "Enter valid Name"	Fail	High	Medium
4	Name shouldn't contain numbers	Name: abc45	Inputs should not be accepted. It should display message "Enter valid Name"	Fail	High	Medium
5	Email field should contain '.com' string	Email=abcdef@redif fmail Username=ABCDE F	Inputs should not be accepted. It should display message "Enter valid Email"	Fail	High	Medium
6	Email field should not contain '#' symbol	Email=john26#redif fmail.com Username=John	Inputs should not be accepted. It should display message "Enter valid Email"	Fail	High	Medium
7	Mobile No. Field should contain 10 digits only	Mobile No.:848294583356	Inputs should not be accepted. It should display message "Enter valid Mobile No."	Fail	High	Medium
8	Mobile No. Field should contain 10 digits only	Mobile No.:848294	Inputs should not be accepted. It should display message "Enter valid Mobile No."	Fail	High	Medium
8	Mobile No. Field should not contain special symbol	Mobile No.: 8482\$56782	Inputs should not be accepted. It should display message "Enter	Fail	High	Medium

Figure 5.1: Registration Phase

### 5.4.3 Login Phase Validation

Test #	Description	Test Inputs	Expected Results	Pass/Fail	Test Priority (High/Medium/Low)	Defect Severity (High/Medium/Low)
1	Check for inputting values in Email field	Email=abcdef@rediffmail Username=ABCDEF	Inputs should not be accepted. It should display message "Enter valid Email"	Fail	High	Medium
2	Check for inputting values in Email field	Email=john26#rediffmail.com Username=John	Inputs should not be accepted. It should display message "Enter valid Email"	Fail	High	Medium
3	Check for inputting values in Username field (username should not contain numbers)	Email=sk@vahoo.com Username=Mark24	Inputs should not be accepted. It should display message "Enter valid Username"	Fail	High	Medium
6	Check for inputting values in Email field	Email=dvan@vahoo.com Username=dave	Inputs should be accepted.	Pass	High	Medium
7	Check for inputting values in Email field	Email=knki@rediffmail.com Username=john	Inputs should be accepted.	Pass	High	Medium

Figure 5.2: Login Phase

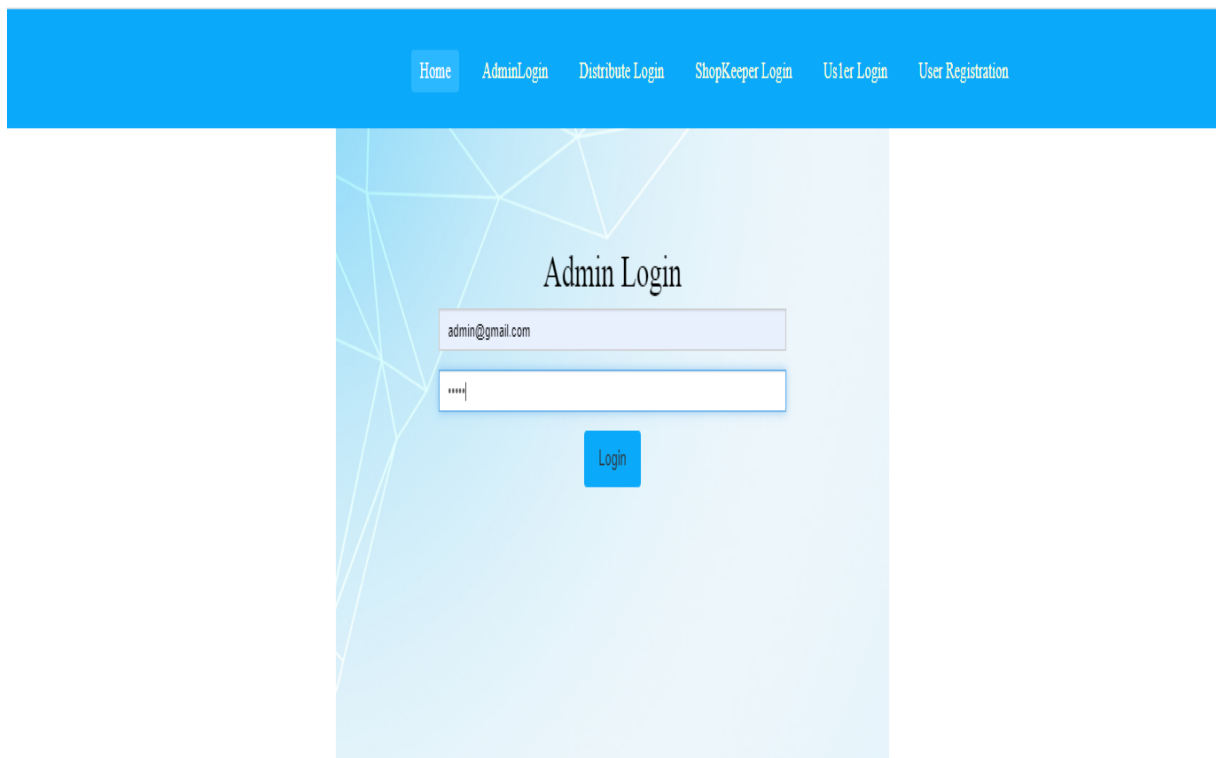
## CHAPTER 6

# Results and Evaluation

### 6.1 Screen shots

#### 6.1.1 Admin Login Page

Admin Login in static email id and password.



The screenshot displays the Admin Login page. At the top, there is a blue navigation bar with the following links: Home, AdminLogin, Distribute Login, ShopKeeper Login, Usler Login, and User Registration. The main content area has a light blue background with a geometric pattern. The title 'Admin Login' is centered. Below the title, there are two input fields: the first contains the email address 'admin@gmail.com', and the second contains a masked password '\*\*\*\*'. A blue 'Login' button is positioned below the password field.

Figure 6.1: Admin Login Page

#### 6.1.2 Admin Add Shop Register Page

Home Shop Registration Distribute Registration View Shop Registration View Distribute Registration Logout

### Registration

1

abc

pune

0000000000

abc@gmail.com

aa

\*\*\*\*\*

Save

Figure 6.2: Admin Add Shop Register Page

### 6.1.3 Admin Add Shop Register Page

Home Shop Registration Distribute Registration View Shop Registration View Distribute Registration Logout

### Registration

1

abc

pune

0000000000

abc@gmail.com

aa

\*\*\*\*\*

Save

Figure 6.3: Admin Add Shop Register Page

#### 6.1.4 Admin Add Distributes Register Page

Home Shop Registration Distribute Registration View Shop Registration View Distribute Registration Logout

### Registration

1

xyz

pune

7777777777

xyz@gmail.com

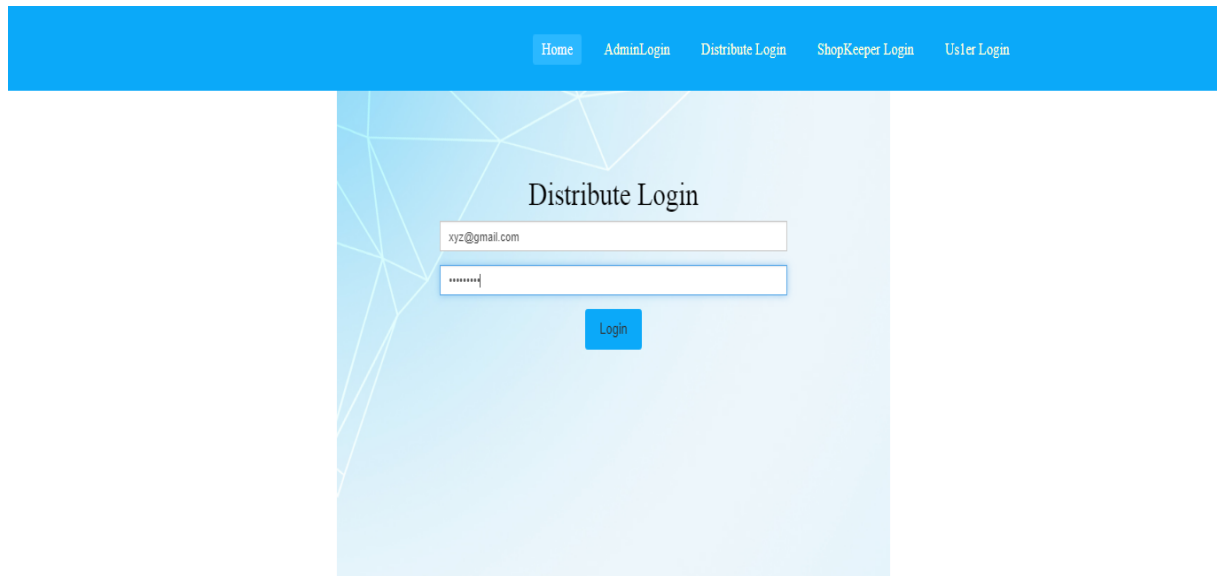
xyz

\*\*\*\*\*

Save

Figure 6.4: Admin Add Distributes Register Page

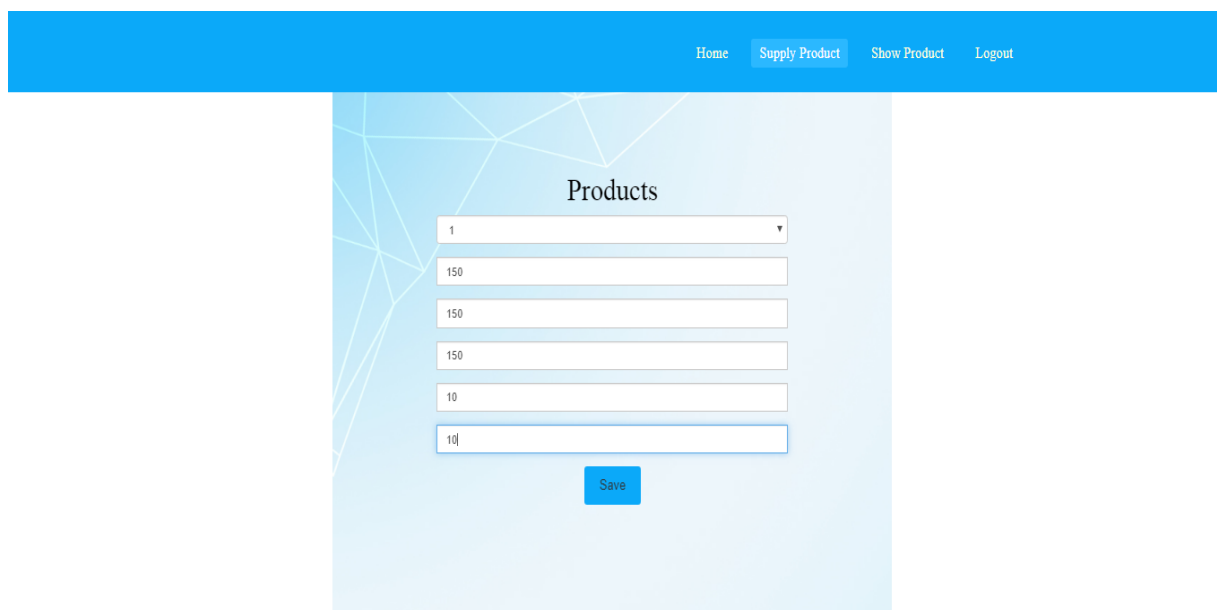
### 6.1.5 Distributes Login Page



The screenshot shows a web application interface for the 'Distribute Login' page. At the top, there is a blue navigation bar with five links: 'Home', 'AdminLogin', 'Distribute Login' (which is highlighted), 'ShopKeeper Login', and 'User Login'. Below the navigation bar, the main content area has a light blue background with a geometric pattern. The title 'Distribute Login' is centered at the top of this section. Below the title, there are two input fields: the first contains the email 'xyz@gmail.com' and the second contains a masked password '\*\*\*\*\*'. A blue 'Login' button is positioned below the password field.

Figure 6.5: Distributes Login Page

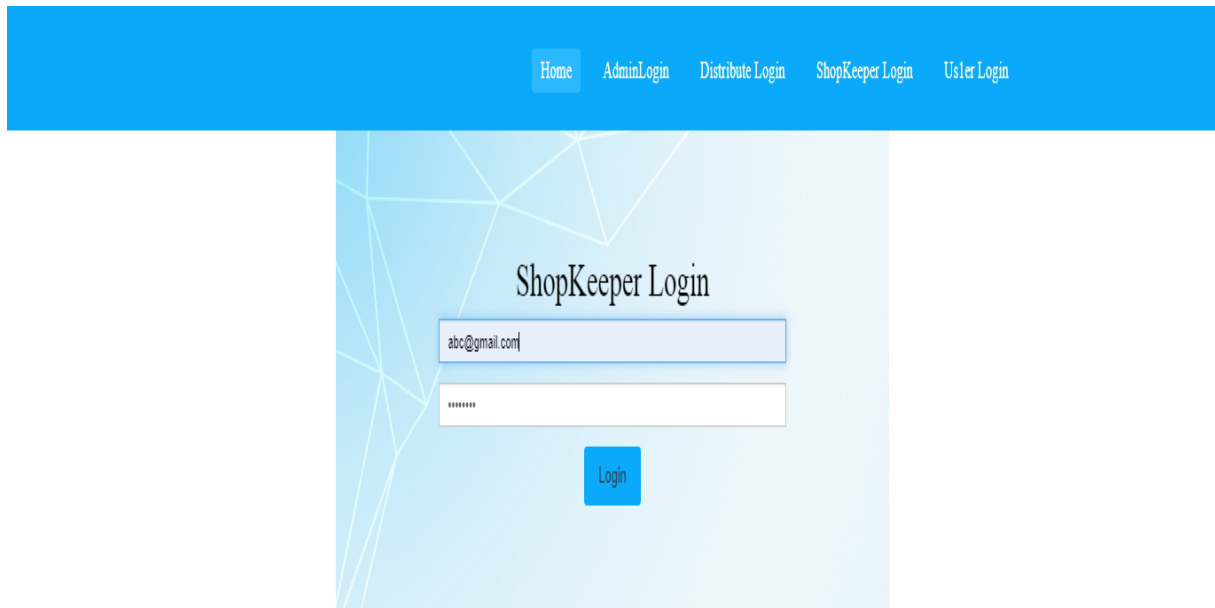
### 6.1.6 Distributes Add Product Page



The screenshot shows a web application interface for the 'Products' page. At the top, there is a blue navigation bar with four links: 'Home', 'Supply Product' (which is highlighted), 'Show Product', and 'Logout'. Below the navigation bar, the main content area has a light blue background with a geometric pattern. The title 'Products' is centered at the top of this section. Below the title, there are six input fields: the first is a dropdown menu showing '1', the next two contain '150', the next contains '150', the next contains '10', and the last contains '10'. A blue 'Save' button is positioned below the last input field.

Figure 6.6: Distributes Add Product Page

### 6.1.7 Shop Login Page



Home AdminLogin Distribute Login ShopKeeper Login User Login

### ShopKeeper Login

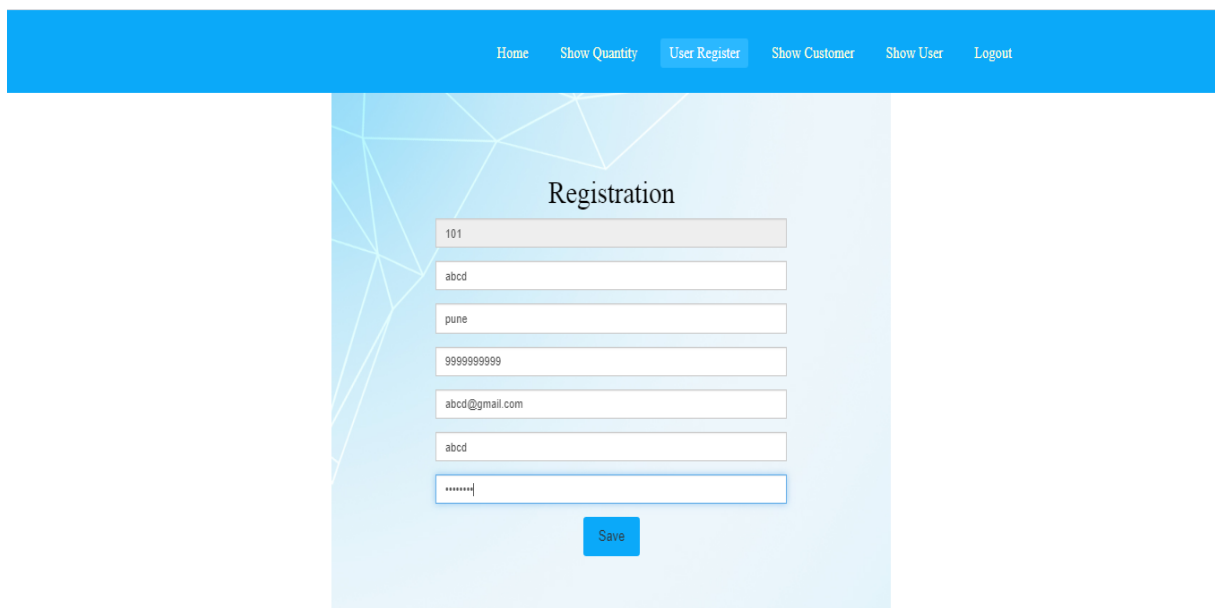
abc@gmail.com

\*\*\*\*\*

Login

Figure 6.7: Shop Login Page

### 6.1.8 Shop Add Register Page



Home Show Quantity User Register Show Customer Show User Logout

### Registration

101

abcd

pune

999999999

abcd@gmail.com

abcd

\*\*\*\*\*

Save

Figure 6.8: Shop Add Register Page

### 6.1.9 Shop Update Product Page

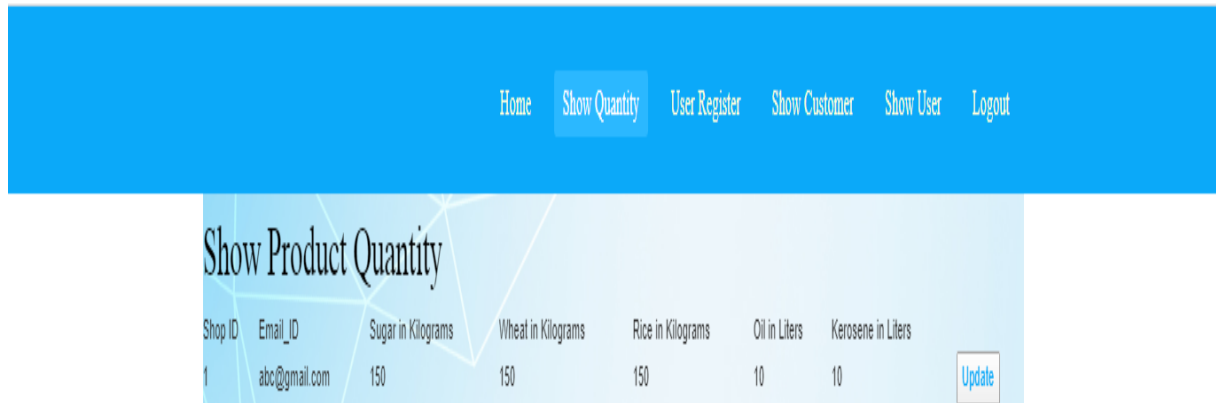


Figure 6.9: Shop Update Product Page

### 6.1.10 User Login Page

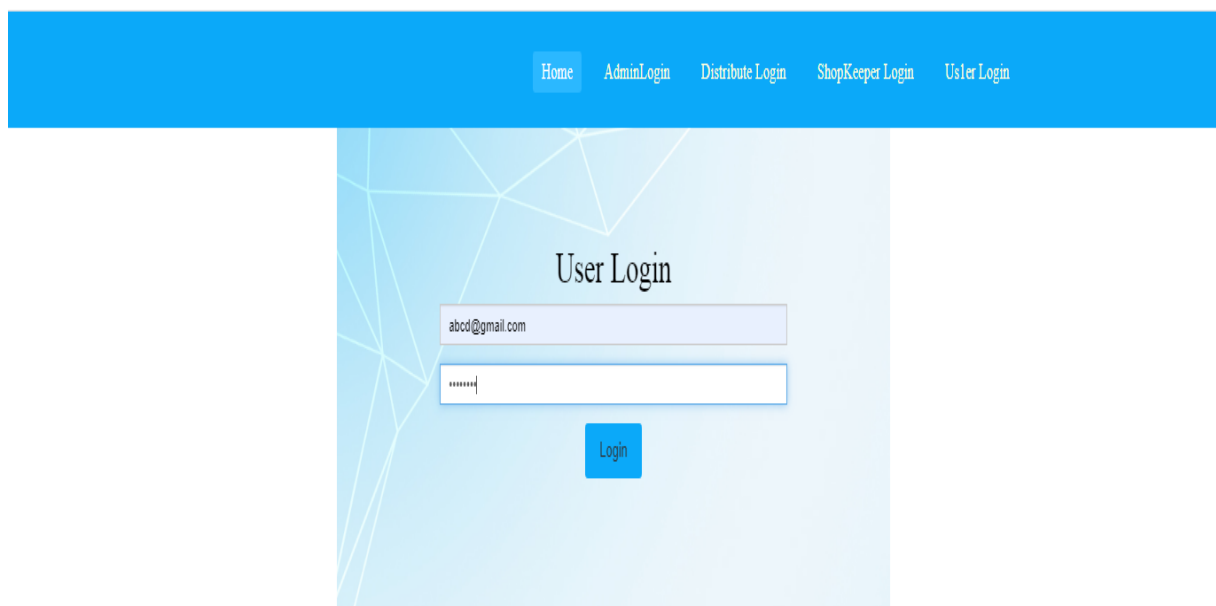


Figure 6.10: User Login Page



### 6.1.11 User Get Product Page

Product	Quantity
Sugar	150
Wheat	150
Rice	150
Oil	10
Kerosene	10

1

abc@gmail.com

50

10

15

13

15

Update

Figure 6.11: User Get Product Page

### 6.1.12 Results(Peer to Peer Connection) Page

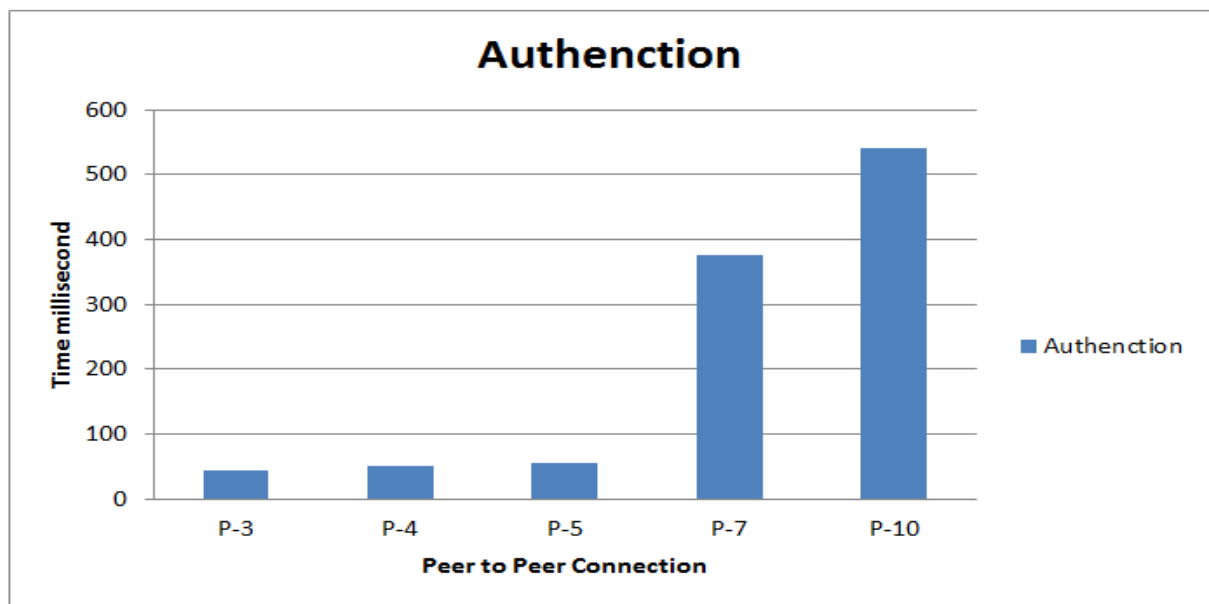


Figure 6.12: Results(Peer to Peer Connection) Page

### 6.1.13 Results(Complete Transaction) Page

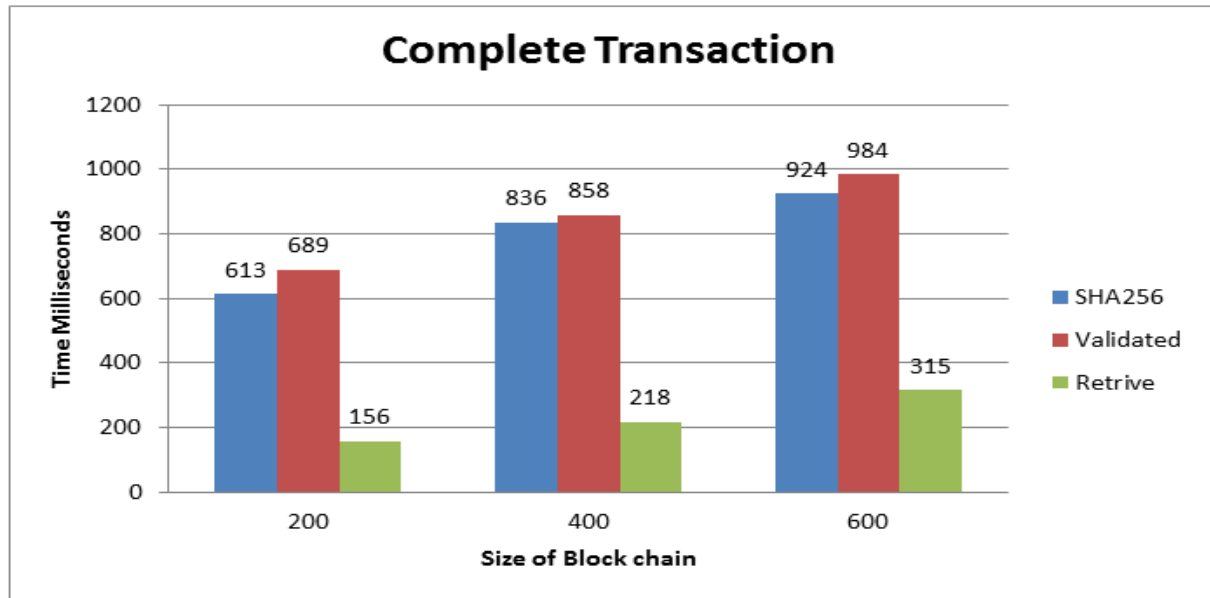


Figure 6.13: Results(Complete Transaction) Page

## CHAPTER 7

# Conclusions and Future Work

### 7.1 Conclusions

There are many research directions in applying Blockchain technology to the e ration transaction due to the complexity of this domain and the need for more robust and effective information technology systems. An inter operable architecture would undoubtedly play a significant role throughout many e ration transaction use cases that face similar data sharing and communication challenges. From the more technical aspect, much research is needed to pinpoint the most practical design process in creating an inter operable ecosystem using the Blockchain technology while balancing critical security and confidentiality concerns in e ration transaction. Whether to create a decentralized application leveraging an existing Blockchain, additional research on secure and efficient software practice for applying the Blockchain technology in e ration transaction is also needed to educate software engineers and domain experts on the potential and also limitations of this new technology. Likewise, validation and testing approaches to gauge the efficacy of Blockchain-based health care architectures compared to existing systems are also important (e.g., via performance metrics related to time and cost of computations or assessment metrics related to its feasibility). In some cases, a new Blockchain network may be more suitable than the existing Blockchains; therefore, another direction may be investigating extensions of an existing Blockchain or creating a e ration transaction Blockchain that exclusively provides e ration transaction services. Blockchain technology should prevent the insecurity and injustice that are part of these e ration registries. The shared ledger technology should bring trust. Will this truly be the case? And will it be possible to replace well-functioning e ration Registration systems that are not corrupt and are kept and managed the proper way, Will a blockchain-based system be less complicated and less expensive than the current well-functioning e ration Registration systems its big question.

### 7.2 Future Scope

To implement the proposed system on multiple peer to peer network, with fog computing which reduce the transactional data processing time.

# REFERENCES

- [1] "Smart Contracts," <http://searchcompliance.techtarget.com/definition/smart-contract>, 2017, [Online; accessed 4-Dec- 2017]
- [2] A. Dorri, S. S. Kanhere, and R. Jurdak, "Blockchain in internet of things: Challenges and Solutions," arXiv:1608.05187 [cs], 2016. [Online]. Available: <http://arxiv.org/abs/1608.05187>
- [3] Sukhodolskiy, Ilya, and Sergey Zapechnikov. "A blockchain-based access control system for cloud storage." Young Researchers in Electrical and Electronic Engineering (EIConRus), 2018 IEEE Conference of Russian. IEEE, 2018.
- [4] Yang, Huihui, and Bian Yang. "A Blockchain-based Approach to the Secure Sharing of Healthcare Data." Proceedings of the Norwegian Information Security Conference. 2017.
- [5] Goyal, Vipul, et al. "Attribute-based encryption for fine-grained access control of encrypted data." Proceedings of the 13th ACM conference on Computer and communications security. Acm, 2006.
- [6] Wang, Hao, and Yujiao Song. "Secure cloud-based EHR system using attribute-based cryptosystem and blockchain." Journal of medical systems 42.8 (2018): 152.
- [7] Michalevsky Y, Joye M. Decentralized Policy-Hiding Attribute-Based Encryption with Receiver Privacy.
- [8] Wu, Axin, et al. "Hidden policy attribute-based data sharing with direct revocation and keyword search in cloud computing." Sensors 18.7 (2018): 2158.
- [9] Khan S, Khan R. Multiple authorities attribute-based verification mechanism for Blockchain mircogrid transactions. Energies. 2018 May;11(5):1154.
- [10] Guo, Rui, et al. "Secure attribute-based signature scheme with multiple authorities for Blockchain in electronic health records systems." IEEE Access 776.99 (2018): 1-12.
- [11] Ouaddah, Aafaf, Anas Abou Elkalam, and Abdellah Ait Ouahman. "FairAccess: a new Blockchain-based access control framework for the Internet of Things." Security and Communication Networks 9.18 (2016): 5943-5964.
- [12] Kiviharju, Mikko. "Enforcing Role-Based Access Control with Attribute-Based Cryptography in MLS Environments."
- [13] He, Qingsu, et al. "A privacy-preserving Internet of Things device management scheme based on blockchain." International Journal of Distributed Sensor Networks 14.11 (2018): 1550147718808750.
- [14] Rahulamathavan, Yogachandran, et al. "Privacy-preserving blockchain based IoT ecosystem using attribute-based encryption." 2017 IEEE International Conference on Advanced Networks and Telecommunications Systems (ANTS). IEEE, 2017.

- [15] Wu, Axin, et al. "Efficient and privacy-preserving traceable attribute-based encryption in blockchain." *Annals of Telecommunications* (2019): 1-11.