Supply Chain Management in Agriculture using Blockchain

Godse Shubham, Gite Rahul, Ekhande Nilesh, Akolkar Aditya

Guide: Prof. Mundhe Bhalchandra

DEPARTMENT OF COMPUTER ENGINEERING

BACHELOR OF ENGINEERING

Sahyadri Valley College of Engineering,Rajuri 413204

Abstract

Block chains are now firmly established as a digital technology that combines cryptographic, data management, networking, and incentive mechanisms to support the verification, execution, and recording of transactions between parties. While block chain technologies were originally intended to support new forms of digital currency for easier and secure payments, they now hold great promise as a new foundation for all forms of transactions. Agribusiness stands to become a key beneficiary of this technology as a platform to execute 'smart contracts' for transactions, particularly for high-value produce. First it is important to distinguish between private digital currencies and the distributed ledger and block chain technologies that underlie them. The distributed and cross-border nature of digital currencies like Bit coin means that regulation of the core protocols of these systems by central banks is unlikely to be effective. Monetary authorities are focused more on understanding 'on-ramps' and 'off-ramps' that constitute the links to the traditional payments system rather than being able to monitor and regulate the currency itself. In contrast to the digital currency feature of block chain, the distributed ledger feature has the potential for widespread use in agribusiness and trade financing, especially where workflows involve many different parties with no trusted central entity.

CHAPTER1

INTRODUCTION

1.1 OVERVIEW

An increasing demand in society for greater information about food reflects the need for more transparency and the lack of trust. At the same time, more and more food products and beverages are branded and accompanied by a variety of certification schemes, with an increasing risk of fraud (selling unqualified product with highquality labels or claims) and adulteration. In the current situation, much of the compliance data and information is audited by trusted third parties and stored either on paper or in a centralised database and these approaches are known to suffer from many informational problems such as

the high cost and inefficiency of paper-based processes and fraud, corruption and error both on paper and in IT systems. These information problems, indicating that current transparency and trust systems have not been able to solve or at times even have exacerbated the problems of low transparency and trust in agrifood chains, pose a severe threat to food safety, food quality, and sustainability. In particular, food integrity has become a major concern. Food integrity refers to the fairness and authenticity of food in food value chains both at the physical layer and the digital layer, where the digital layer should provide reliable and trustworthy information on the origin and provenance of food products in the physical layer. Blockchain technology provides a means to ensure permanence of records and potentially to facilitate the sharing of data between disparate actors in a food value chain. This potential may lead to an exciting paradigm shift facilitating transparency and trust in food chains that ensures food integrity.

"Supply Chain Management in Agriculture using Blockchain"

1.2 MOTIVATION

The last three years have seen an explosion of interest in Blockchain Technology (BCT) with a great many companies and research institutions focusing on potential applications of this technology across aange of financial, industrial and social sectors. However, the technology has also been surrounded by a great deal of exaggeration and hype resulting in misplaced expectations and misunderstandings. BCT is still in an early stage of development, with considerable potential for real-life commercial applications. Innovation in blockchain architectures, applications and business concepts is happening at a fast pace; it is often characterised by decentralised, open source development, and it is perceived as being disruptive to traditional players in many industries.

1.3 PROBLEM DEFINITION AND OBJECTIVES

To develop an agricultural supply chain management system with BCT using java as a programming language.

1.3.1 Objectives:

1. To implement a java based web application.
2. To implement AES.
3. To implement visual cryptography.
4. To implement block chain.
5. To implement distributed database system using WLAN.

1.4 PROJECT SCOPE AND LIMITATIONS

Project will be developed as a prototype model using JSP and servlet technology. It will run as a local host. System will be communicate through wireless local area network. System communication will be

limited in the wireless local area network, but in future if we will host the project using WAN , it can communicate world wide.

1.5 METHODOLOGIES OF PROBLEM SOLVING

BCT Agricultural products are the foundation of the people's survival, and the quality of agricultural products has always been the focus of attention of society and the government; the original agricultural product traceability system is too difficult to tamper with data due to the excessive concentration of data storage, it faces the challenge of fraudulent data tracing, and it is difficult for consumers to trust such traceability results. Moreover, the centralized storage method is not conducive to the centralized management of traceable data from many enterprises, and there will be problems of low traceability and difficulty in government supervision. The emergence of blockchain technology provides a new solution for data security problems of food traceability, its decentralization, anti-tampering and other characteristics and data encryption technology improve the difficulty of data fraud and ensure data security. If the blockchain is combined with the traceability of agricultural products, the safety of traceable data and the tampering of data can be guaranteed to the greatest extent, the producer's production behavior can be regulated, and consumers' confidence in food quality can be improved. This project mainly proposes a framework of agricultural product traceability system based on blockchain technology, it uses blockchain to store the traceability data of agricultural products safely, and proposes a traceability model of agricultural products, which can cover the entire industrial chain of agricultural products, and consumers can query the authentic source of traceability of agricultural products.

CHAPTER2

LITERATURESURVEY

- A model in Agri-food Supply Chain Costing using ABC Costing: A empirical research for Peruvian coffee supply chain

  Andrea Villalva-Catano,˜ Edgar Ramos-Palomino, Kelsey Provost, Eduardo Casal DOI 10.1109/IESTEC46403.2019.00009 2019 7th International Engineering, Sciences and Technology Conference (IESTEC) This article examines the fundamental causes of Peruvian coffee's high logistical costs in the supply chain. A cost analysis technique will aid in the exploration, analysis, and development of high supply chain costs in order to stabilise the current coffee crisis. Indeed, the findings were studied in order to improve, assist, and aid small-business growth over time.
- A Theoretical Implementation: Agriculture- Food Supply Chain Management using Blockchain Technology S. Madumidha1, P. Siva Ranjani2, U.Vandhana3, B.Venmuhilan4 978-1-7281-1034-

9/19/2019 IEEE This paper describes a fully decentralised blockchain-based traceability system that can be used to create agricultural building blocks that are continuously integrated with IoT devices from provider to consumer. To do so, we created the ”Provider-Consumer Network,” a fictional end-to-end food traceability system. The goal is to establish a distributed ledger that is available to all network users and so provides transparency.

- Blockchain in Agriculture by using Decentralized Peer to Peer Networks Mrs S.Thejaswini, Ranjitha K R, Department of CSE, Siddaganga Institute of Tech-

  “Supply Chain Management in Agriculture using Blockchain”

  nology, Tumkur, Karnataka, India. The distributed ledger, centralised servers, P2P (Peer to Peer) networks, As in [1] [10]RFID (Radio-Frequency Identification) tag, consensus verification, and other features of blockchain technology play a major role in the agriculture industry by improving transparency and food provenance in the supply chain, which is characterised by the distributed ledger, centralised servers, P2P (Peer to Peer) networks, and consensus verification. As a result, the proposed work investigates the various issues that arise in agricultural production and proposes solutions to those issues utilising blockchain technology.
- Blockchain technology in current agricultural systems: from techniques to ap-

  plications WANG1, HAINING YIN4, DEWEI YI5, AND LAIHUNG YAU6 DOI 10.1109/ACCESS.2020.3014522, IEEE Access We conduct a survey in this research to examine both the methodology and applications of blockchain technology in the agriculture sector. The technical features, such as data structure, cryptographic algorithms, and consensus procedures, are first thoroughly explained. Second, to demonstrate the usage of blockchain techniques, existing agricultural blockchain applications are categorised and assessed. In addition, examples of how practitioners leverage popular platforms and smart contracts to construct agricultural applications are offered. Finally, we highlight the fundamental challenges that many future agricultural systems face, as well as the attempts and potential solutions that have been made to address these issues.
- Blockchain-based Data Traceability Platform Architecture for Supply Chain Management

  Yihang Wei The IEEE 6th International Conference on Big Data Security on Cloud (BigDataSecurity), the IEEE International Conference on High Performance and Smart Computing (HPSC), and the IEEE International Conference on Intelligent Data and Security will all be held in 2020. (IDS) Based on the multidisciplinary knowledge and technology of the Fabric Alliance chain architecture, perceptual identification technology, and cryptographic knowledge,

CHAPTER3
SOFTWAREREQUIREMENT
SPECIFICATION

3.1 PROJECT SCOPE

- To develop prototype model for agricultural supply chain management.
- This model will be run at local host using Glassfish server.
- BCT features such as decentralization, cryptography and hash codes will be implemented.

3.2 ASSUMPTIONS AND DEPENDENCIES

This document will provide a general description of project, including user requirements, product perspective, and overview of requirements, general constraints. In addition, it will also provide the specific requirements and functionality needed for this project such as interface, functional requirements and performance requirements.

3.2.1 User Classes and Characteristics

Find the different user classes that you anticipate will use this product. User classes can be differentiated based on use frequency or product functions subset used or technical expertise or privilege levels or educational level and experience. It also describe the pertinent behavior or characteristics of each user class. Few requirements may limited only to specific user classes. Differentiate the very most important or useful user classes for this item or product from those who are less
significant to satisfy.

3.3 FUNCTIONAL REQUIREMENTS

Functional user requirements is nothing but very high-level statements about what the system should and also it should describe clearly an overview of system
services in detail.

3.4 EXTERNAL INTERFACE REQUIREMENTS

3.4.1 User Interfaces

The user interface or UI for the software should be compatible to be used by any standard browser such as IE, Mozilla or Google chrome. Using this UI user can have access to the system. The UI or user interface can be developed by using many tool or software package like JFrame.

3.4.2 Hardware Interfaces

A hardware interface is needed to run the software. Java (JDK) and NetBeans compatible hardware is required which is minimal requirement.

3.4.3 Software Interfaces

It uses Java as the front end programming tool. MySQL has been used as back end application tool. Latest version of java anything higher than 7.0 can be used.

3.5 NON FUNCTIONAL REQUIREMENTS

3.5.1 Performance Requirements

- System can work optimal or faster on 4 GB or more of RAM.
- The system is targeted to be available all time. Once there is a fatal error or system down, the system will provide understandable feedback to the user.

3.5.2 Safety Requirements

- The system is designed in modules where errors can be detected.

3.5.3 Security Requirements

- The system is designed in modules where errors can be detected and fixed easily.

3.5.4 Software Quality Attributes

- Usability:

  This relates to how easily people can use app/website. A measure of usability could be the time it takes for end users to become familiar with my app/website functions, without training or help.

- Reliability:

  This can be defined as the available time or UP time of software.

- Performance:

  This is essentially how fast app/website works. A performance requirement for the app/website could be start in less than 20 seconds.

- Security :

  Say that app/website saves all the previous code and lets you reuse a saved code.

3.6 SYSTEM REQUIREMENTS

3.6.1 Database Requirements

MySQL Database

MySQL is on open source database which is mainly a RDBMS i.e. relational database management system. As a database server, primary function of this software is to storing and retrieving data as requested by other from end software applications like java which may Or may not run either on the same computer or on different computer. This can be across the network either in internet or intranet.

3.6.2 Software Requirements

1. Operating System: Microsoft Windows 7 and Above

2. Programming Language: Java
3. IDE: Netbeans

3.6.3 Hardware Requirements

1. Processor: Intel Core I3 or Higher
2. RAM: 4 GB or Higher
3. Hard Disk: 100 GB (min)

3.7 ANALYSIS MODELS: SDLC MODEL TO BE APPLIED

SDLC model to be applied

Waterfall Model:

The Waterfall Model is among very first and old model of software development life cycle. It is also called as a linear-sequential life cycle model. This is very simple in nature and easy to understand or use. This is step by step method so next step can only be begin once earlier has been completed. This is mainly used for small scale project. Constant or fixed requirement should be there for this type of model.

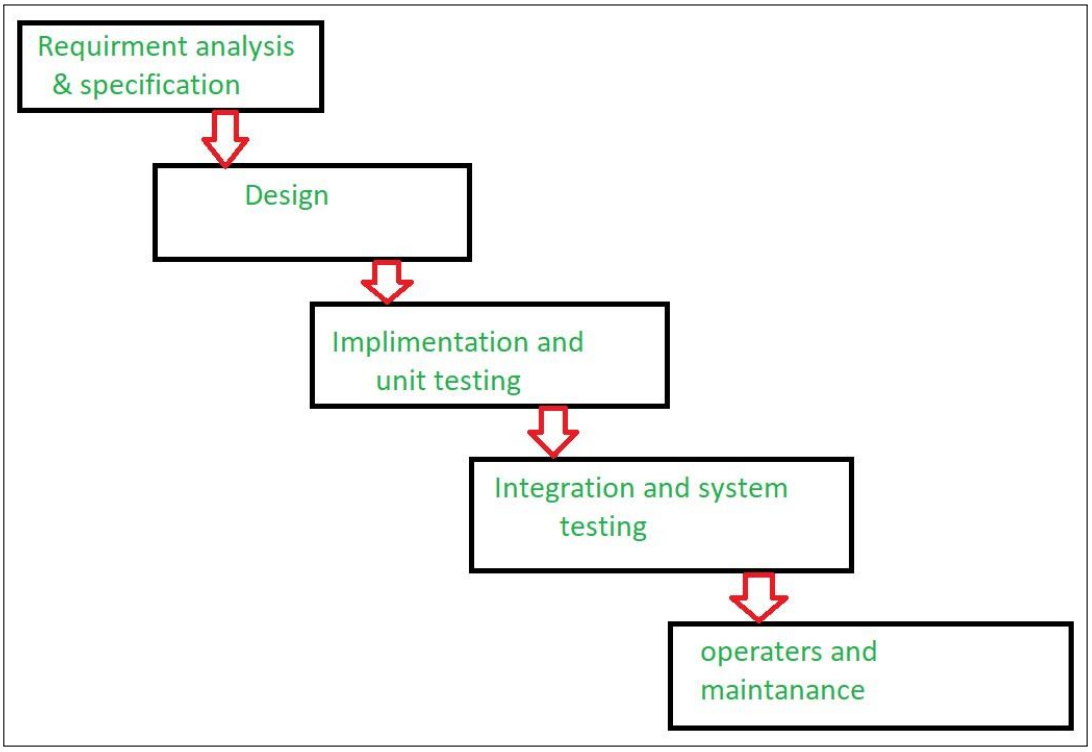


Figure 3.1: Waterfall Model

CHAPTER4

SYSTEMDESIGN

4.1 SYSTEM ARCHITECTURE

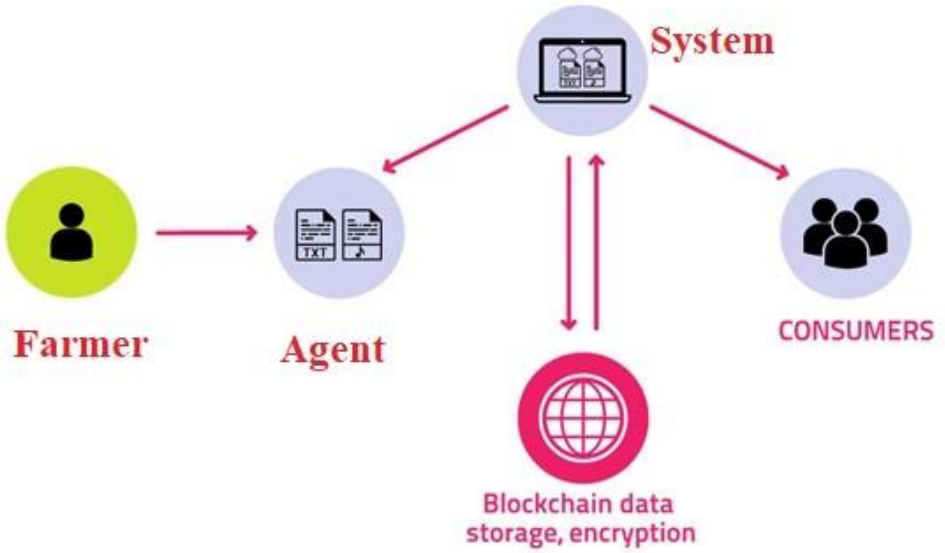
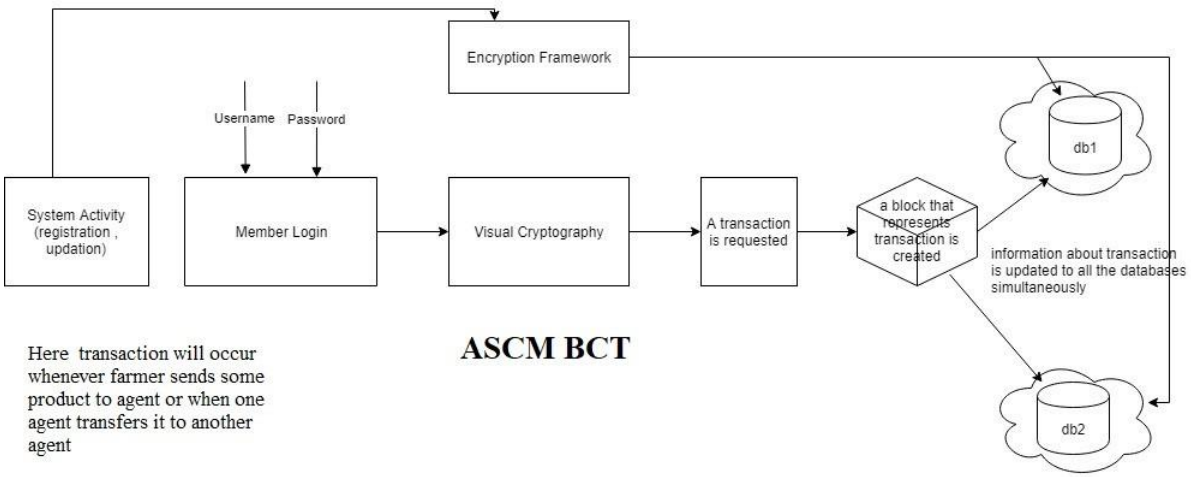


Figure 4.1: System Architecture

Whenever any transaction will occur in the system , the record of that transaction is maintained in the form of hash value in a block. Each next block will get attached to the previous block and in this way a virtual block chain will occur. The hash value of a current block is generated using the data of a current block and the hash of the previous block. In this way if any of the block is tempered the subsequent all the block's hash must be changed . Such multiple copies are maintained at different servers , which will assure the data security and confidentiality. As everything is through application interface, it will maintain the transparency in the agricultural supply chain management.

Figure 4.2: System Flow



4.2 MATHEMATICAL MODEL

Let

S be Closed system defined as, S = Ip, Op, Ss, Su, Fi, A

To select the input from the system and perform various actions from the set of actions A so that Su state can be attained.

S=Ip,Op,Ss,Su,Fi,A

Where,

IP1=Username,Password, image

Set of actions=A=F1,F2,F3,F4

Where

F1= Send Mail

F2= Merge Images

F3= Encrypt Database

F4= Generate Hash

S=Set of users

Ss=rest state, registration state, login state

Su- success state is successful analysis

Fi- failure state

Objects:

1) Input1: Ip1 = Username, Password

2) Input2 : Ip2= image from mail

1) Output1 : Op1 = Transaction Record 2) Output2 : Op2 = Encrypted Database

3) Output3 : Op3 = Hash Codes.

4.3 DATA FLOW DIAGRAMS

A data flow diagram (DFD) is a graphical representation of the “flow” of data through an information system, modeling its process aspects. A DFD is often used as a preliminary step to create an overview of the system, which can later be elaborated.

DFDs can also be used for the visualization of data processing.

4.3.1 Level 0 Data Flow Diagram

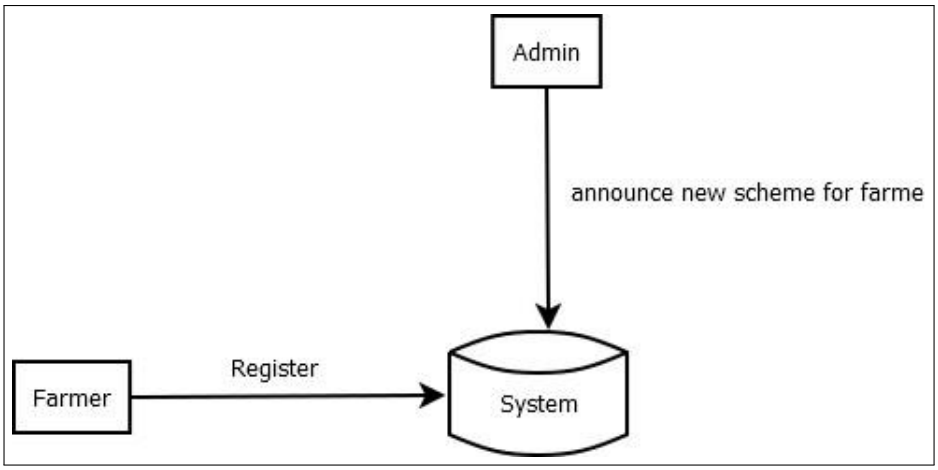


Figure 4.3: Level 0 Data Flow Diagram

4.3.2 Level 1 Data Flow Diagram

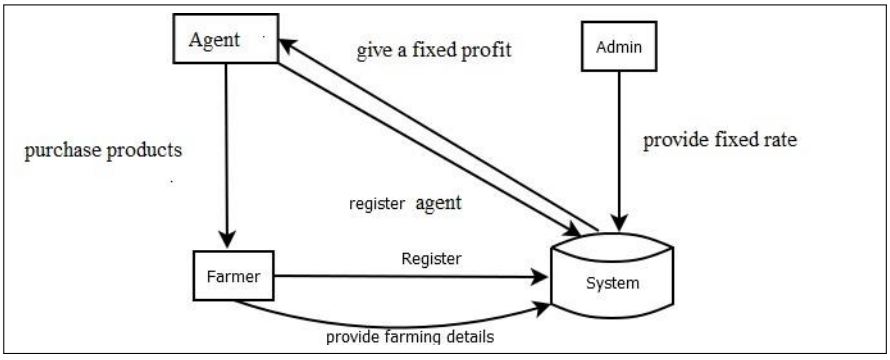


Figure 4.4: Level 1 Data Flow Diagram

4.4 ENTITY RELATIONSHIP DIAGRAMS

An entity relationship diagram (ERD) shows the relationships of entity sets stored in a database. An entity in this context is an object, a component of data. An entity set is a collection of similar entities. These entities can have attributes that define its properties. By defining the entities, their attributes, and showing the relationships between them, an ER

diagram illustrates the logical structure of databases.

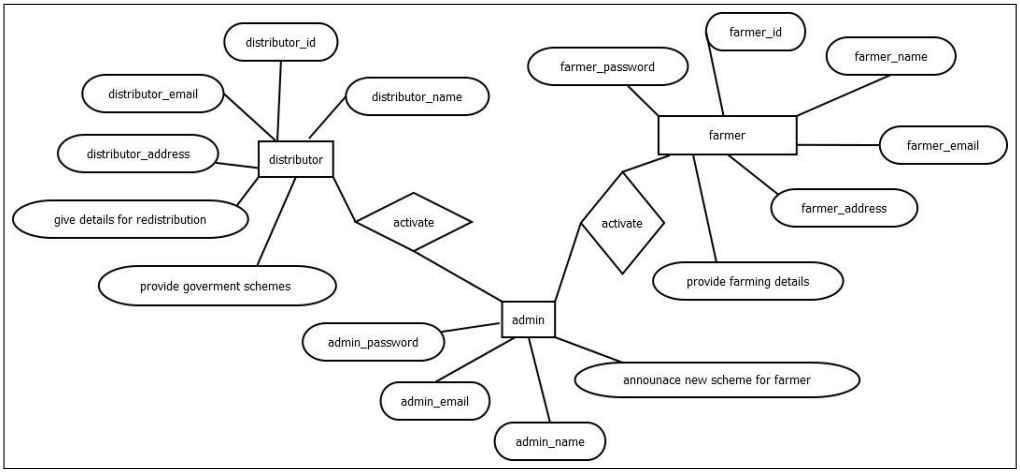


Figure 4.5: Entity Relationship Diagrams

4.5 UML DIAGRAMS

4.5.1 Class Diagram

A class diagram in the world of Unified Modeling Language or UML can be defined as a type of static structure diagram which mainly defines the structure of a system. It works by showing the system's classes and their attributes and operations or methods also the relationships among objects.

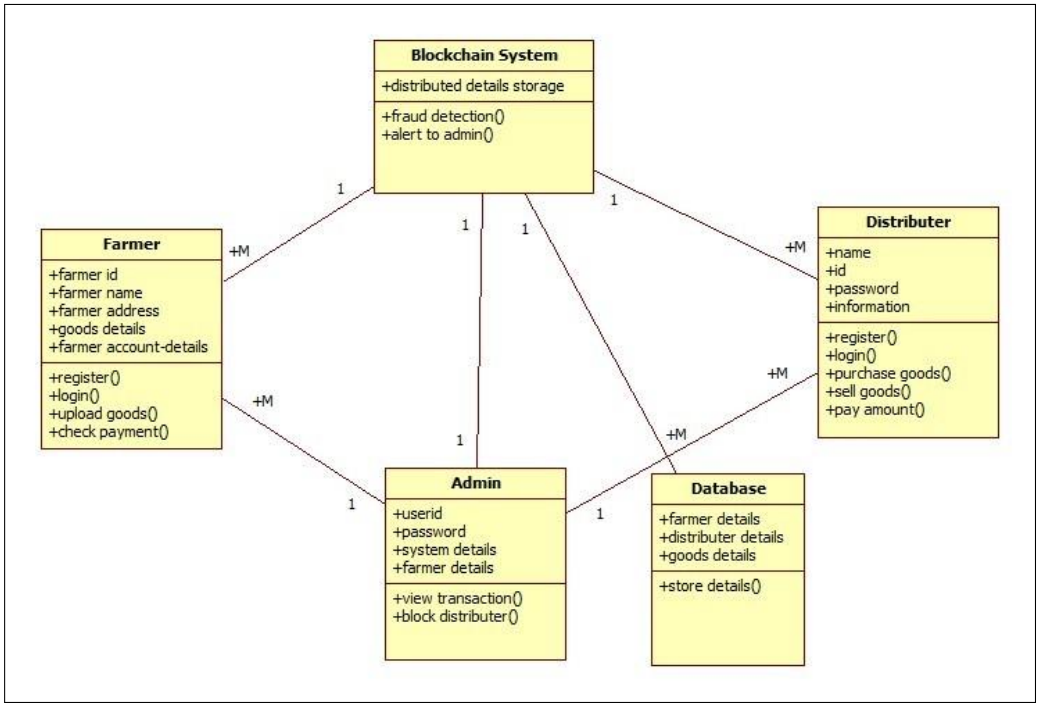


Figure 4.6: Class Diagram

4.5.2 Use Case Diagram

Dynamic behavior is most important aspect to capture the model of any system. Dynamic behavior can be defined as the behavior of the system when it is running or operating. Static behavior is not sufficient to model a system rather dynamic behavior is more important than static behavior.

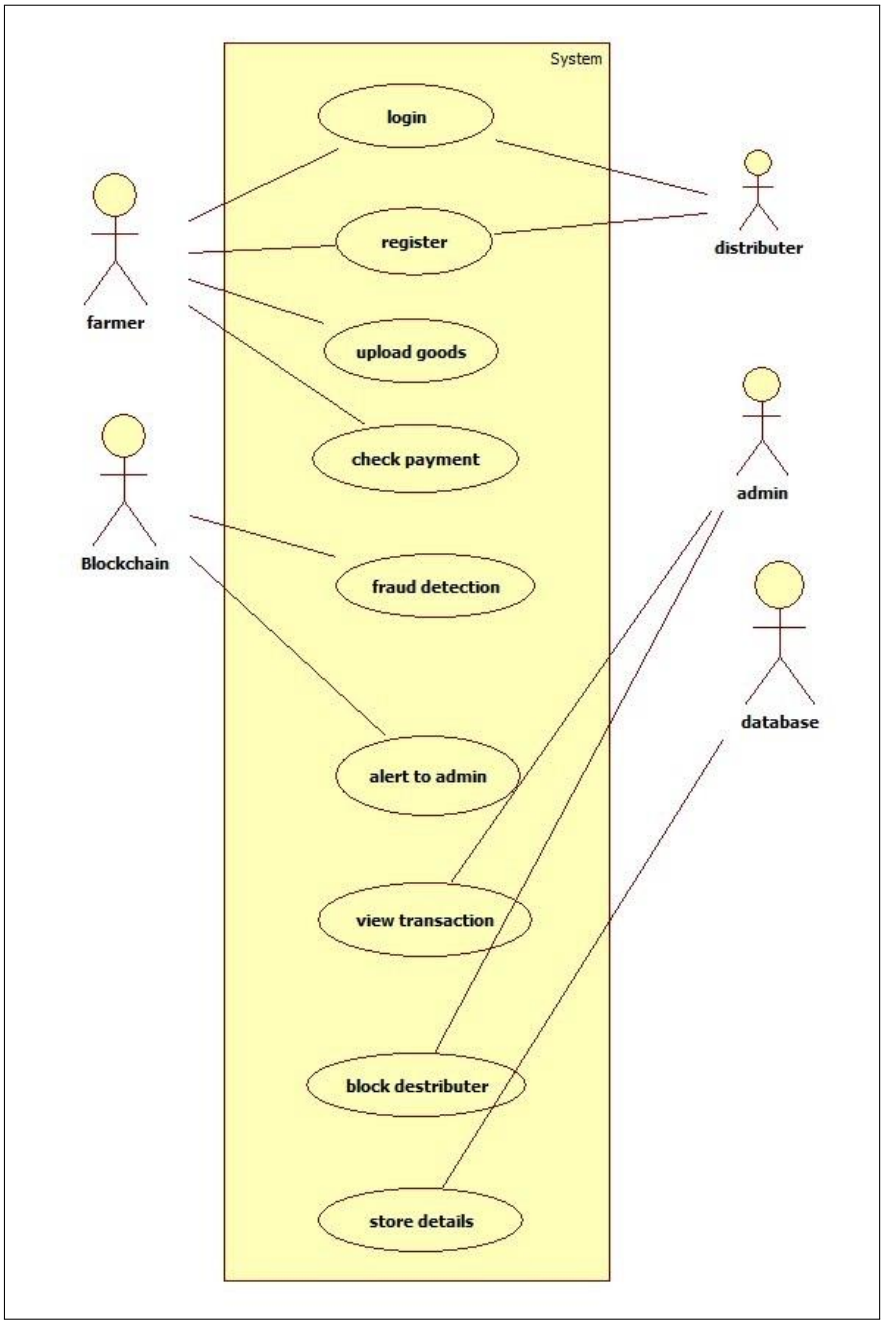


Figure 4.7: Use Case Diagram

4.5.3 Sequence Diagram

Sequence diagrams can be used to provide a graphical representation of object interactions or object coordination over the time. These basically displays a actor or user, and the objects and components they

interact with in the execution of a use case. The sequence diagrams displays the own of messages from one object to another object, and as such correspond to the methods and events supported by a class/object.

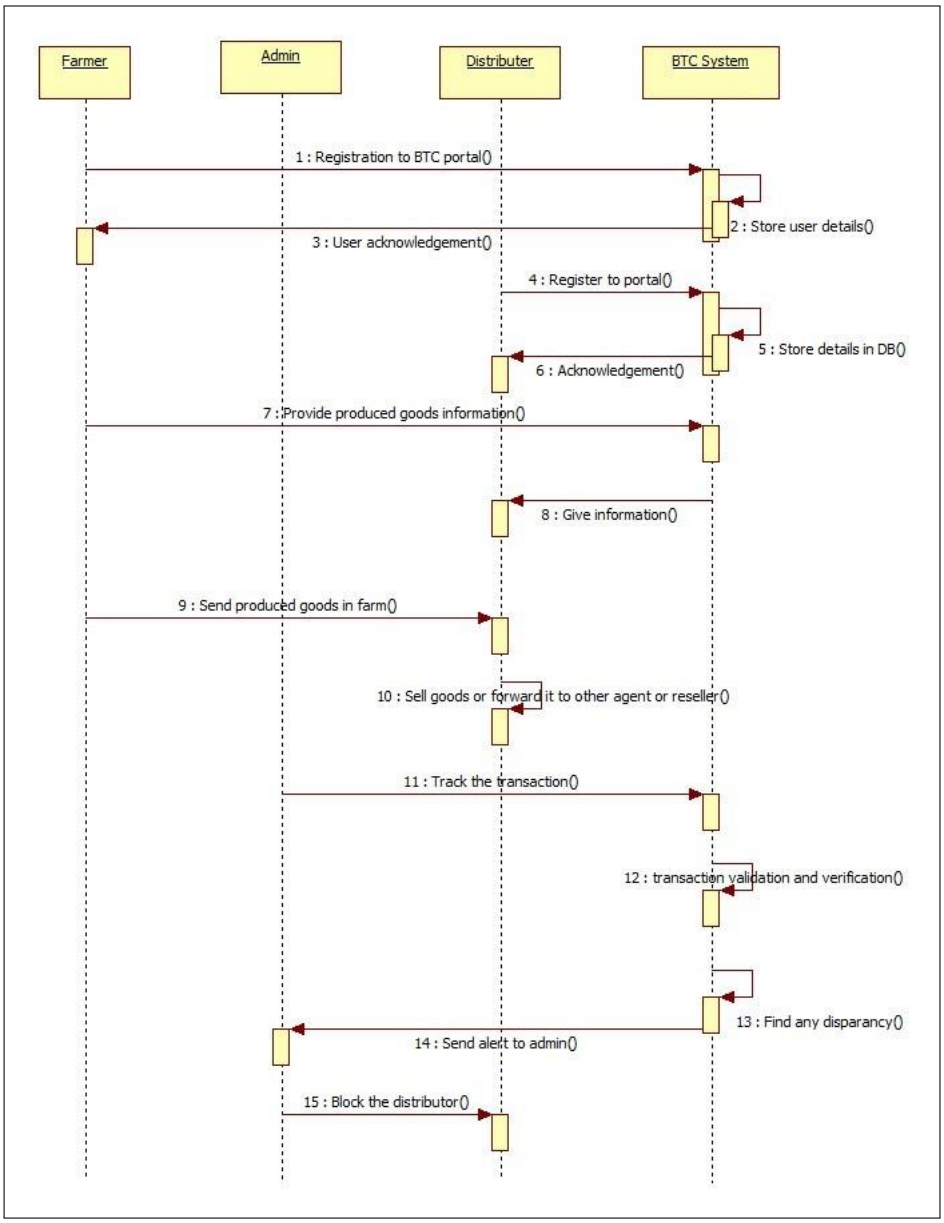


Figure 4.8: Sequence Diagram

4.5.4 Activity Diagram

Activity diagram can be defined as a flowchart to display the flow from one activity to another activity. These activities could be described as an operation of the system. The control flow usually is drawn from one operation of application to another. This can be branched or sequential, or concurrent also. Activity diagrams can deal with all or many type of flow control and used different elements such as join or fork.

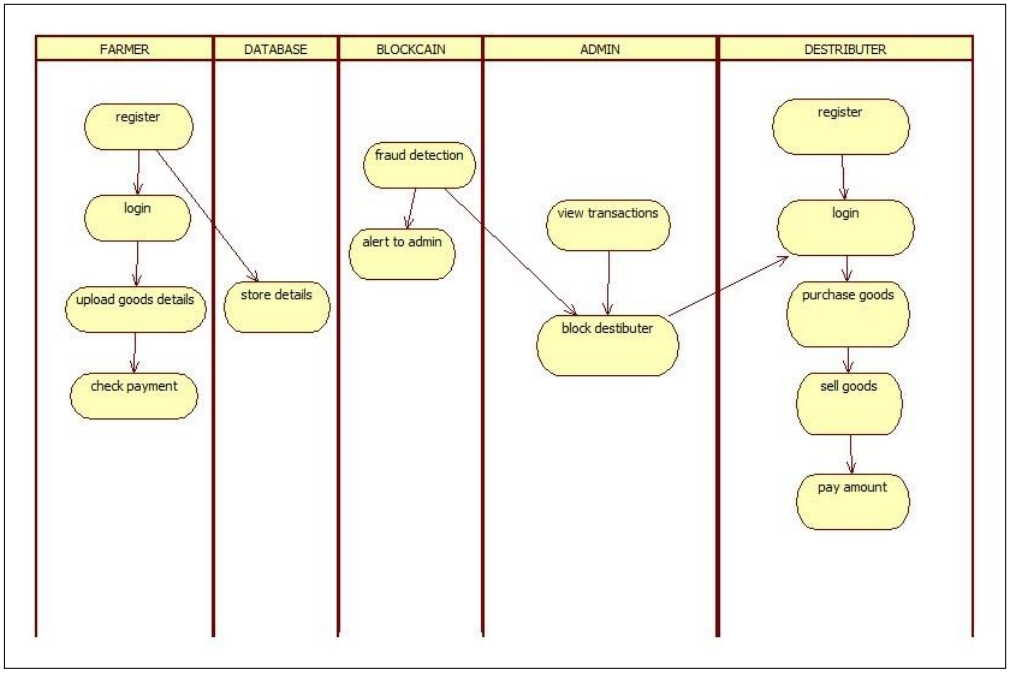


Figure 4.9: Activity Diagram

CHAPTER5 PROJECTPLAN

Phase	Task	Description
Phase 1	Analysis	Analyze the information related to Project Topic
Phase 2	System Design	Assign the module and design the process flow Control
Phase 3	Implementation	Implement the code for all the modules and integrate all the modules
Phase 4	Testing	Test the code and overall process weather the process works properly Test the code and over all process weather the process works properly
Phase 5	Maintenance	Modification of a software product after delivery to improve performance or maintainability.

5.0.1 Reconciled Estimates

5.1 PROJECT ESTIMATE

Sr.No.	Milestone Name	Milestone Description
1.	Requirement Analysis	Complete specification of system
2.	High level design	Identify the modules and the different entities and their relationships

3.	Detailed design	GUI design, program specification etc
4.	Build	Writing code for different modules
5.	Testing	Test the different modules together
6.	Final Review and Deployment	Checking all the requirements are fulfilled

Table 5.1: Project Estimate

5.1.1 COCOMO Model

Cocomo (Constructive Cost Model) is a regression model based on LOC, i.e number of Lines of Code. It is a procedural cost estimate model for software projects and often used as a process of reliably predicting the various parameters associated with making a project such as size, effort, cost, time and quality. It was proposed by Barry Boehm in 1970 and is based on the study of 63 projects, which make it one of the best-documented models. The key parameters which define the quality of any software products, which are also an outcome of the Cocomo are primarily Effort & Schedule:

- Effort: Amount of labor that will be required to complete a task. It is measured in person-months units.
- Schedule: Simply means the amount of time required for the completion of the job, which is, of course, proportional to the effort put. It is measured in the units of time such as weeks, months.

Different models of Cocomo have been proposed to predict the cost estimation at different levels, based on the amount of accuracy and correctness required. All of these models can be applied to a variety of projects, whose characteristics determine the value of constant to be used in subsequent calculations. These characteristics pertaining to different system types are mentioned below.

Boehm's definition of organic, semidetached, and embedded systems:

1. Organic – A software project is said to be an organic type if the team size required is adequately small, the problem is well understood and has been solved in the past and also the team members have a nominal experience regarding the problem.

2. Semi-detached – A software project is said to be a Semi-detached type if the vital characteristics such as team-size, experience, knowledge of the various programming environment lie in between that of organic and Embedded. The projects classified as Semi-Detached are comparatively less familiar and difficult to develop compared to the organic ones and require more experience and better guidance and creativity. Eg: Compilers or different Embedded Systems can be considered of Semi-Detached type.
3. Embedded – A software project with requiring the highest level of complexity, creativity, and experience requirement fall under this category. Such software requires a larger team size than the other two models and also the developers need to be sufficiently experienced and creative to develop such complex models.

All the above system types utilize different values of the constants used in Effort Calculations. Types of Models: COCOMO consists of a hierarchy of three increasingly detailed and accurate forms. Any of the three forms can be adopted according to our requirements. These are types of COCOMO model:

1. Basic COCOMO Model The first level, Basic COCOMO can be used for
   quick and slightly rough calculations of Software Costs. Its accuracy is somewhat restricted due to the absence of sufficient factor considerations.
   The basic COCOMO model estimates the software development effort using only lines of code.

$$E = a(KLOC)^b$$

   newline Where,
   E is the efforts applied by person in months, a = 3.0 and b = 1.12 , then
   KLOC=2.25
   Hence Efforts = 3.0 (1.8)1.12 ,
   E = 5.79 Person-month
   E = 6 Person-month
   Total of 6 Person-Month are required to complete the project successfully.

$$D = cb(E)^d b$$

   Where,
   D = Development time in chronological months, cb = 2.5 and db = 0.35, and
   E =6 Person-Month
   Hence, Development Time= 2.5 (1.8)0.35
   D = 3.07 months

The approximate duration of project is 3 months.

P = E/D

Where,

P = Number of persons to accomplish project.

Hence, Number of Persons required completing the project

P = 6 / 3

P = 2 persons

Therefore 2 persons are required to successfully complete the project on sched-

ule.

2. Intermediate COCOMO Model Intermediate COCOMO takes these Cost Drivers into account and Detailed COCOMO additionally accounts for the influence of individual project phases, i.e in case of Detailed it accounts for both these cost drivers and also calculations are performed phase wise henceforth producing a more accurate result. These two models are further discussed below.
3. Detailed COCOMO Model In detailed cocomo, the whole software is divided into different modules and then apply COCOMO in different modules to estimate effort and then sum the effort.

Project Cost-

The model followed is the Constructive Cost Model(COCOMO) for estimating the effort required in completing the project. Like all the estimation models, the COCOMO Model requires sizing information. This information can be specified in the form of:

- Object Point
- Function Point
- Lines of source Code (KLOC) for our project, This work uses the sizing information in the form Lines of Source Code.
- Total lines of code for our project, KLOC =1.8K (approx.).
- Cost of each person per month, Cp=Rs.11,000/- (Per person-month)

So, $C = 3 \times Cp = 3 \times 11000 = 33{,}000-$

Therefore, the cost pf project is 33,000+10000(cost of camera approx) = 43,000/-

(approx).

5.1.2 Reconciled Estimates

The part of the project will be hardware, which need to implement in our system on besides, also need to estimates the cost of the application which are designing keeping in mind the following factor:

- Its market demand, what it has got to offer to the customer

- Its relevance in the current world.
- The extent to which it can adhere to its objective of secured data transmission.

5.1.3 Project Resources

1. Designer: To design system and perform requirement gathering.

2.Developer: To develop system and provide to tester for testing

5.2 RISK MANAGEMENT

:

Risk Identification

For risks identification, review of scope document, requirements specifications and schedule is done. Answers to questionnaire revealed some risks. Each risk is categorized as per the categories mentioned in [?].You can refereed following risk identification questionnaire.

1. Have top software and customer managers formally committed to support the project? Answer : Yes
2. Are end-users enthusiastically committed to the project and the system/product to be built? Answer : Yes
3. Are requirements fully understood by the software engineering team and its customers? Answer : Yes
4. Have customers been involved fully in the definition of requirements? Answer : Yes
5. Do end-users have realistic expectations? Answer : Yes
6. Does the software engineering team have the right mix of skills? Answer : Yes
7. Are project requirements stable? Answer : Yes
8. Is the number of people on the project team adequate to do the job? Answer : Yes
9. Do all customer/user constituencies agree on the importance of the project and on the requirements for the system/product to be built? Answer : Yes

5.2.2 NP Hard

A problem is NP-hard if solving it in polynomial time would make it possible to solve all problems in class NP in polynomial time. Some NP-hard problems are also in NP (these are called ”NP-complete”), some are not. If you could reduce an

NP problem to an NP-hard problem and then solve it in polynomial time, you could solve all NP problems. Also, there are decision problems in NP-hard but are not NP-complete, such as the infamous halting problem

5.2.3 Risk Analysis

- Technical Risk: The probability of loss incurred through the execution of a technical process in which the outcome is uncertain. Untested engineering, technological or manufacturing procedures entail some level technical risk that can result in the loss of time, resources, and possibly harm to individuals and facilities. Like mobile phone battery off, network error in user and server, multiple requests at time. • Operational Risk: Operational risk is the prospect of loss resulting from inadequate or failed procedures, systems or policies. Employee errors. Systems failures. Fraud or other criminal activity. Any event that disrupts business processes. Like user registration, login, send request to service provider.
- Schedule Risk: Schedule risk is the risk that the project takes longer than scheduled. It can lead to cost risks, as longer projects always cost more, and to performance risk, if the project is completed too late to perform its intended tasks fully.
- Business Risk: Business risk is the possibilities a company will have lower than anticipated profits or experience a loss rather than taking a profit. Business risk is influ- enced by numerous factors, including sales volume, per-unit price, input costs, competition, and the overall economic climate and government regulations.

5.3 PROJECT SCHEDULE

:

5.3.1 Project task set

Major Tasks in the Project stages are:

Priority (High to low)	Risks	Back-up plan
1	Schedule	Overtime

2	Operational	Validation
3	Business	Marketing
4	Technical	-

- Task 1: Requirement Gathering
- Task 2: Literature Survey
- Task 3: System Design
- Task 4: Functionality Implementation
- Task 5: Testing

5.3.2 Timeline Chart

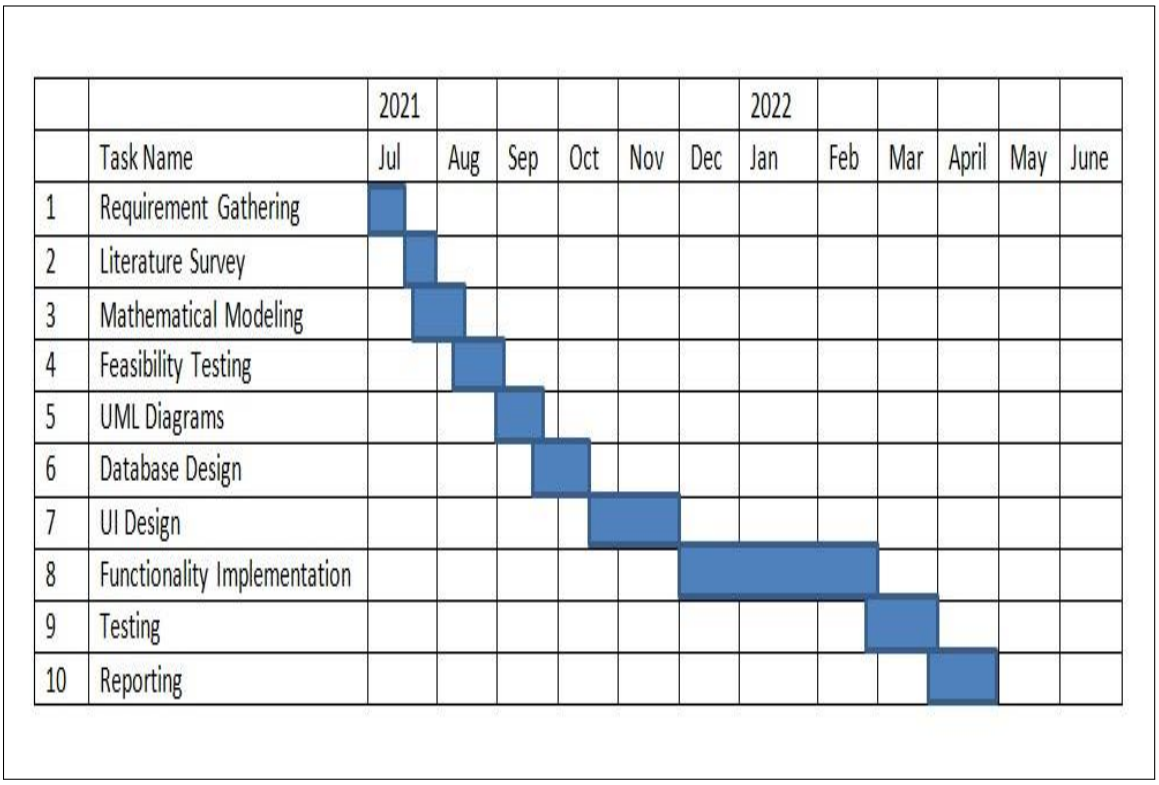


Figure 5.1: Timeline Chart

5.4 TEAM ORGANIZATION

5.4.1 Team Structure

Whatever activities are done related to the project that we all showing all details log to our guide. All the reporting are noted to the guide.

Work Task	Description	Duration
Literature Search	Related work done for conceptual data similarity	6 weeks
System analysis	Critical analysis and comparison of technologies studied and results achieved in research	4 weeks
Design and Planning	Modeling and design and dataset searching or creation	8 weeks
Implementation	Divided into phases	
Phase A	Implementation module 1	2 weeks
Phase B	Implementation module 2	2 weeks
Phase C	Implementation module 3	2 weeks
System Testing	Test system quality, fix errors if any and improve if needed. Test system for different data sets	3 weeks
Final Report	Prepare and upload Initial Report	2 weeks
Initial Report	Prepare and upload Initial Report	2 weeks

Table 5.2: Time line Chart

CHAPTER6

PROJECTIMPLEMENTATION

6.1 OVERVIEW OF PROJECT MODULES

BCT: First and foremost, blockchain is a public electronic ledger built around a P2P system that can be openly shared among disparate users to create an unchangeable record of transactions, each time-stamped and linked to the previous one. Every time a set of transactions is added, that data becomes another block in the chain (hence, the name). Blockchain can only be updated by consensus between participants in the

system, and once new data is entered it can never be erased. It is a write-once, append-many technology, making it a verifiable and auditable record of each and every transaction. Famer will transfer the products to the agent through the application interface, agent in turn will transfer any product to another agent through application interface only. Also the record of each and every transaction will be maintained at different places which will maintain transparency also the database is secured through AES. System login is secured through visual cryptography.

6.2 TOOLS AND TECHNOLOGIES USED:

JDK 1.8 Installation 1. Double click jdk-8-ea-bin-b32-windows-i586 to run the installation program. JDK License dialog displayed. Accept the license in order to install JDK.

2. The JRE Custom setup dialog enables you to choose a custom directory for JRE Files. 3. The complete dialog indicates a successful installation.

Net Beans IDE 7.3.1 Installation To install the software:

1. After the download completes, run the installer. For Windows, the installer executable file has the .exe extension. Double-click the installer file to run it. 2. If you downloaded the All bundle, you can customize your installation. Per- form the following steps at the Welcome page of the installation wizard:

a. Click Customize.

b. In the Customize Installation dialog box, make your selections.

- At the Welcome page of the installation wizard, click Next. At the License agreement page, review the license agreement, click the acceptance check box, and click Next. At the JUnit License Agreement page, decide if you want to install JUnit and click the appropriate option, click Next. At the NetBeans IDE installation page, do the following: • Accept the default installation directory for the NetBeans IDE or specify an- other directory. Note: The installation directory must be empty and the user profile you are using to run the installer must have read/ write permissions for this directory. • Accept the default JDK installation to use with the NetBeans IDE or select a different installation from the drop-down list. If the installation wizard did not find a compatible JDK installation to use with the Net-Beans IDE, your JDK is not installed in the default location. In this case, specify the path to an installed JDK and click Next, or cancel the current installation. After installing the required JDK version you can restart the installation.
- If you are installing Apache Tomcat, on its installation page, accept the default installation directory or specify another installation location. Click Next. • At the Summary page, verify that the list of components to be installed is correct and that you have adequate

space on your system for the installation. • Click Install to begin the installation. • At the Setup Complete page, provide anonymous usage data if desired, and click Finish.

MySQL Database

Microsoft SQL Server is a relational database management system developed by Mi- crosoft. As a database server, it is a software product with the primary function of storing and retrieving data as requested by other software applications which may run either on the same computer or on another computer across a network (includ- ing the Internet). Microsoft markets at least a dozen different editions of Microsoft SQL Server, aimed at different audiences and for workloads ranging from small single-machine applications to large Internet-facing applications with many concur- rent users.

6.3 ALGORITHM DETAILS:

Algorithm:

AES is used to encrypt the database.

The encryption process uses a set of specially derived keys called round keys. These are applied, along with other operations, on an array of data that holds exactly one block of data, the data to be encrypted. This array we call the state array.

STEPS:

- Derive the set of round keys from the cipher key.
- Initialize the state array with the block data (plaintext).
- Add the initial round key to the starting state array.
- Perform nine rounds of state manipulation.
- Perform the tenth and final round of state manipulation
- Copy the final state array out as the encrypted data (ciphertext).

6.3.2. SHA 256:Hash Function

SHA-256 (secure hash algorithm, FIPS 182-2) is a cryptographic hash function with digest length of 256 bits. It is a keyless hash function; that is, an MDC (Manipulation Detection Code). A message is processed by blocks of 512 = 16 ×

32 bits, each block requiring 64 rounds A cryptographic hash (sometimes called ‘digest’) is a kind of ‘signature’ for a text or a data file. SHA-256 generates an almost-unique 256-bit (32-byte) signature for a text.A hash is not ‘encryption’ – it cannot be decrypted back to the original text (it is a ‘one-way’ cryptographic function, and is a fixed size for any size of source text). This makes it suitable when it is

appropriate to compare 'hashed' versions of texts, as opposed to decrypting the text to obtain the original version.

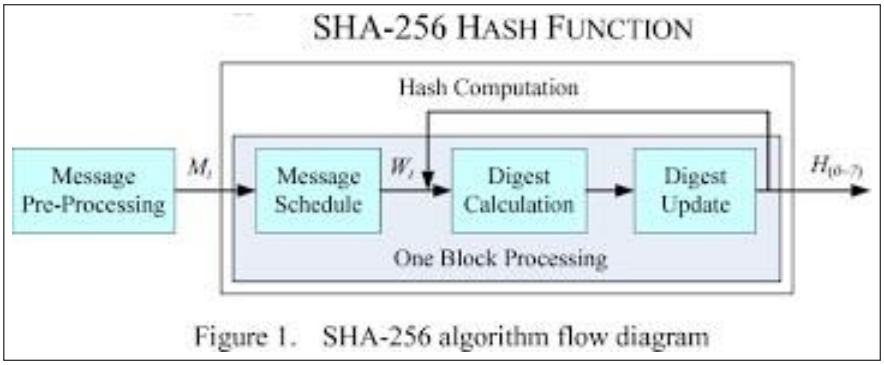


Figure 6.1: SHA 256 Diagram

CHAPTER7

CONCLUSION

:

Thus we are are going to implement a prototype web based software application in Java for application of BCT in supply chain management . We will implement block chain features such as: 1. Decentralization 2. Visual Cryptography 3. Hash Algorithm 4. Encrypted Database. Thus it is possible to track agricultural supply chain and to give minimum price for agricultural products.

7.1 FUTURE SCOPE

In future we will try for sponsorship from government and will implement a project on large scale with some domain and hosting space online.

7.2 APPLICATIONS

1. Farmers
2. Government Organizations
3. Banking Sector.
4. Educational System

CHAPTER8

REFERENCES

1. L. Guo, C. Zhang, J. Sun, Y. Fang. “A privacy-preserving attribute based authentication System for Mobile Health Networks”, IEEE Transactions on Mobile Computing, 2014.
2. A. Abbas, S. Khan, “ A review on the state-of-the-art privacy preserving approaches in e-health clouds”, IEEE Journal of Biomedical Health Informatics, 2014.
3. J. Yang, J. Li, Y. Niu, “ A hybrid solution for privacy preserving medi-cal data sharing in the cloud environment ”, Future Generation Computer Systems, 2015.
4. V. Goyal, O. Pandey, A. Sahai, B. Waters, “ Attribute-based encryption for fine-grained access control of encrypted data ”, Proc. 13thm ACM Conf. Computer and Comm. Security (CCS06), 2006.
5. R. Ostrovsky, A. Sahai, B.Waters, “ Attribute-based encryption with nonmonotonic access structures ”, in: Proceedings of the 14th ACM Confer-ence on Computer and Communications Security, ACM, 2007.
6. J. Han, W. Susilo, Y. Mu. “ Improving privacy and security in decentralized cipher text-policy attribute-based encryption ”, IEEE Transactions on on Information Forensics and Security, 2015.
7. M. Li, S. Yu, Y. Zheng, K. Ren, W. Lou, “ Scalable and secure sharing of personal health records in cloud computing using attribute based encryp-tion

“Supply Chain Management in Agriculture using Blockchain”

”, IEEE transactions on parallel and distributed systems, 2013.
8. M. Green, S. Hohenberger, B. Waters, “ Outsourcing the decryption of ABE ciphertexts ”, in Proc. USENIX Security Symp., San Francisco, CA, USA, 2011.
9. J. Lai, R. H. Deng, C. Guan, J. Weng, “Attribute-based encryption with verifiable outsourced decryption ”, IEEE Trans. Inf. Forensics Security, Aug. 2013.
10. B. Qin, R. H. Deng, S. Liu, S. Ma, “Attribute-based encryption with efficient verifiable outsourced decryption ”, IEEE Trans. Inf. Forensics Security, JULY. 2015.