VISVESVARAYA TECHNOLOGICAL UNIVERSITY

Jnana Sangama, Belagavi–590018,Karnataka.

Project report on

"Block chain-based Traceability in Agri-Food Supply Chain Management: A Practical Implementation"

Submitted in partial fulfillment of the requirement for the award of degree of

BACHELOR OF ENGINEERING

in

COMPUTER SCIENCE & ENGINEERING

Submitted by:-

BHEEMASHREE	**(1AK20CS013)**
ASHWINI	**(1AK20CS008)**
SHRUTHI CD	**(1AK20CS077)**
KAVYSHREE BS	**(1AK20CS029)**

Under the guidance of

Dr. PUSHPA R B.E., MTech.., PhD.

Professor and Head, Department of CSE.

Akshaya Institute of Technology , Tumakuru

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

AKSHAYA INSTITUTE OF TECHNOLOGY

Lingapura ,Tumkur-Koratagere Road,Tumkur-572106.

2023-24

AKSHAYA INSTITUTE OF TECHNOLOGY

Lingapura,Tumkur-KoratagereRoad,Tumkur-572106.

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

CERTIFICATE

This is to certify that the Project work entitled **"Blockchain-based Traceability in Agri-Food Supply Chain Management: A Practical Implementation"** is a bonafide work Carried out by **Bheemashree (1AK20CS013), Shruthi C D (1AK20CS077), Kavyashree (1AK20CS029), Ashwini (1AK20CS029) in** partial fulfillment for the award of degree in Bachelor of Engineering in Computer Science of Visvesvaraya Technological University, Belagavi during the academic year of 2023-24. It is certified that all correction/suggestions indicated by internal assessment have been incorporated in the report deposited in departmental library. The project work has been approved as it satisfies the academic requirements in respect of Project work prescribed for the said degree.

Signature of the Guide:	**Signature of the HOD:**	**Signature of the Principal:**
Dr. Pushpa R PhD ., Professor & Head Department of CSE, AIT, Tumakuru	**Dr . Pushpa R** PhD., Professor & Head, Department of CSE, AIT, Tumakuru.	**Dr. K V Sreenivas Rao** PhD., Principal, AIT, Tumakuru

Name of the Examiners	**Signature**
1) _____	_____
2)	_____ _____

AKSHAYA INSTITUTE OF TECHNOLOGY

Lingapura,Tumkur-KoratagereRoad,Tumkur-572106.

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

DECLARATION

We Are students of Eighth Semester B.E at the Department of Computer Science and Engineering, the Akshaya Institute of Technology, Tumakuru declare that the project entitled

"**Block chain-based Traceability in Agri-Food Supply Chain Management: A Practical Implementation**" has been presented by us and submitted in partial fulfillment of course requirements for the award of degree in Bachelor of Engineering in Computer Science and Engineering discipline of Visvesvaraya Technological University, Belagavi during the academic years in 2023-2024

BHEEMASHREE (1AK20CS013)

ASHWINI (1AK20CS008)

SHRUTHI C D (1AK20CS077)

KAVYSHREE B S (1AK20CS029)

ACKNOWLEDGEMENT

The satisfaction and euphoria that accompany the successful completion of any task would be incomplete without the mention of the people who made it possible, whose constant guidance and encouragement crowned our efforts with success.

I/we have great pleasure in expressing my gratitude to my respected chairman **Dr. Shivakumar K S** for providing me with great infrastructure and well-furnished labs

I/we take this opportunity to express my profound gratitude to our respected Principal **Dr. K V Sreenivas Rao** for his constant support and encouragement.

I/we am/are grateful to the Head of the Department, **Dr. Pushpa R**, Department of CSE, for her unfailing encouragement and suggestion given to me in the course of my *Project/Mini Project/Technical seminar/Internship* work.

Guidance plays a very important role in successful completion of the Project report on time. I/we also convey my gratitude to **Dr. Pushpa R** **PhD.**, Assistant Professor, Department of CSE, for having constantly monitored the development of this report.

Finally, a note of thanks to the Department of Computer Science Engineering, both teaching and non-teaching staff for their co-operation extended to me.

CONTENTS

ABSTRACT

This project embarks on a journey into the frontier of multi-disease prediction, leveraging the prowess of machine learning (ML) methodologies. With a focused approach, three distinctive algorithms - K-Neighbors Classifier, Gaussian Naive Bayes, and Convolutional Neural Network (CNN) - are employed to decode the complex interplay between symptoms and diseases. Operating within a framework where symptoms are meticulously categorized into 11 classes, the system strives to unravel the diagnostic puzzle by offering predictions for a diverse spectrum of 7 potential ailments. Amidst the exploration, the project's evaluation framework places a spotlight on algorithmic accuracy, underlining the pivotal role it plays in steering the predictive prowess of the system. Through an intricate web of experimentation and analysis, this project endeavors to cast a revealing light on the performance landscape of the K-Neighbors Classifier, Gaussian Naive Bayes, and Convolutional Neural Network within the domain of disease prediction. By dissecting the accuracy metrics of each algorithm in the context of symptom-based disease prognostication, invaluable insights are unearthed, illuminating their practical efficacy and adaptability within real-world healthcare settings. As the curtains draw on this scholarly endeavor, the implications resonate far and wide, promising to recalibrate the trajectory of disease diagnosis processes towards heightened efficiency and unwavering reliability. It's a transformative journey poised to usher in a new era where machine learning becomes an indispensable ally in the realm of clinical decision-making

CHAPTER-1

INTRODUCTION

In recent years, the problems of agri-food safety and farmer income have received widespread attentions The issues of agri-food safety may occur in each part of agri-food supply chains (ASCs), while inefficient management strategies of ASCs would lead to low profits. However, many factors may constrain the normal work of ASCs. First, due to the complex structure of ASCs, it is hard to record the full circulation information of agri-food products while ensuring that the information will never be tampered with. Second, the shift of consumer preferences has become the main barrier of precisely determining the production and storage of agri-food products with the consideration of profit maximization. Such uncertainties and dynamics undoubtedly increase the toughness of designing an efficient ASC framework. To address these problems, the effective traceability and management for agri- food products in ASCs have become urgently necessary On one hand, to guarantee the agri- food safety, the information of agri-food products in ASCs including production, processing, storage, distribution, and retail should be collected and recorded when establishing a mechanism of product traceability However, most of the traditional traceability solutions of ASCs rely on a centralized system maintained by a trusted third party, which may suffer from the potential single-node failure and security threats such as data tampering and information leakage. Blockchain, a distributed, append-only, and tamper-proof ledger, offers an effective architecture for reliable transactions on the Bitcoin network without the control of a centralized third party. Each information recorded in a blockchain should be verified by the majority of participants to reach a global consensus, which ensures the information source with auditability and transparency. Moreover, there is no need for a blockchain-based traceability solution to connect to a remote cloud data center because it only requires the stable connection among adjacent participants. Therefore, the blockchain technology can be used to realize reliable product traceability in supply chains, which has recently become a new research direction and attracted many research interests. For example, Tian [10] proposed a traceability system for ASCs with the radio frequency identification (RFID) and blockchain technologies, where RFID-based devices and blockchains are used for collecting

and storing data, respectively. Furthermore, the author designed another traceability system for ASCs based on the Hazard Analysis and Critical Control Points (HACCP), blockchain and Internet-of-Things (IoT) technologies. Toyoda et al. Proposed a blockchain-based product ownership management system (POMS), which can be used to prevent counterfeit products in supply chains. Caro et al. developed the Agri Block IoT, a blockchain-based traceability solution, which can acquire the agri-food data of production and consumption from IoT devices along ASCs. Mao et al. designed a blockchain-based credit evaluation system to optimize the supervision and management of food supply chains, which collected the credit via smart contracts and made analysis by using the long short-term memory (LSTM). Lin et al. proposed an information and communications technology (ICT) based system by integrating the blockchain technology. T se et al. discussed the application of the blockchain technology to the food supply chain and made comparisons with traditional traceability systems. and Monfared proposed that the application of blockchains can help raise trust levels of supply chains by using transparent and traceable transactions.

CHAPTER-2

LITERATURE SURVEY

Blockchain is an emerging meta-technology and considered a new institutional technology with the potential to change the governance of vertically integrated food supply chains. This paper investigates the effects on coordination mechanisms in vertically cooperating agri-food networks that result from the implementation of different blockchain technology platform types (BCTPT). The research is based on an extensive literature overview and exploratory use cases of BCTPT implementations in the agri-food industry which are presented to illustrate the applicability of the findings. Our analysis shows that BCTPT predominantly differentiate through the coordination mechanisms exerting of power, information sharing, decision-making, and collective learning benefits. We also reveal that blockchain use cases with high success rates typically operate in a vertical ecosystem where a focal firm assumes the responsibility for coordinating the activities in the supply chain network. These use cases are typically operationalized in tracking and tracing applications as well as in provenance-based information provision, which either operate in vertically coordinated private blockchain or consortium-type blockchain platforms. We conclude that the choice of a specific BCTPT with its respective coordination mechanisms is a key determinant of the economic success of the intended use case, the efficient management of the supply chain network, and eventually for the chosen digital business model. This paper will close a research gap, as the potential impacts of different blockchain technology platform types on digital agri-food business models and its supply chain management have scarcely been researched.

Supply chains are evolving into automated and highly complex networks and are becoming an important source of potential benefits in the modern world. At the same time, consumers are now more interested in food product quality. However, it is challenging to track the provenance of data and maintain its traceability throughout the supply chain network. The traditional supply chains are centralized and they depend on a third party for trading. These centralized systems lack transparency, accountability and auditability. In our proposed solution, we have presented a complete solution for blockchain- based Agriculture and Food (Agri-Food) supply chain. It leverages the key features of blockchain and smart contracts, deployed over blockchain network. Although blockchain provides immutability of data and records in the network, it still fails to solve some major problems in supply chain management like credibility of the involved entities, accountability of the trading process and traceability of the products. Therefore, there is a need of a reliable system that ensures traceability, trust and delivery mechanism in Agri-Food supply chain. In the proposed system, all transactions are written to blockchain which ultimately uploads the data to Interplanetary File Storage System (IPFS). The storage system returns a hash of the data which is stored on blockchain and ensures efficient, secure and reliable solution. Our system provides smart contracts along with their algorithms to show interaction of entities in the system. Furthermore, simulations and evaluation of smart contracts along with the security and vulnerability analyses are also presented in this work.

The fundamental purpose of agri-food supply chain management is to restrict opportunism caused by information asymmetry. Traditional Chinese agri-food supply chain management introduces a contract mechanism and a trust mechanism to manage the uncertainty of the agri-food quasi-organization. However, it is almost impossible to improve the efficiency of transactions and maintain agri-food supply chain stability in the case of asymmetric information. Nowadays, blockchain, Internet of Things technology and big data drive the agri-food supply chain into a vast smart network which would break the information constraints. This paper analyzes the coupling between blockchain-based digital system and the agri-food supply chain. In addition, this paper presents two cases from China, indicating that the proposed blockchain-based system can achieve disruptive transformation in agri-food supply chain management.

The implementation of the blockchain technology in the agri-food supply chains is in its introductory phase. Lead companies, often retailers, introduce this technology for specific objectives, such as assuring traceability or improving sales and reputation. At the same time, the technology could impact much more broadly the performances of food chains. Little is known about this impact as the evidence provided in the literature is scarce and mostly focused on specific indicators. This paper addresses this gap assessing the impact of the blockchain technology on food supply chains from an explorative perspective. An integrated conceptual framework is proposed which includes a broad set of performance dimensions discussed in the literature: efficiency, flexibility, responsiveness, food quality, and transparency of supply chains. These dimensions are assessed using a case study, consisting of three supply chains where a large European retailer has promoted the blockchain adoption.

The recent, exponential rise in adoption of the most disparate Internet of Things (IoT) devices and technologies has reached also Agriculture and Food (Agri-Food) supply chains, drumming up substantial research and innovation interest towards developing reliable, auditable and transparent traceability systems. Current IoT-based traceability and provenance systems for Agri-Food supply chains are built on top of centralized infrastructures and this leaves room for unsolved issues and major concerns, including data integrity, tampering and single points of failure. Blockchains, the distributed ledger technology underpinning cryptocurrencies such as Bitcoin, represent a new and innovative

technological approach to realizing decentralized trustless systems. Indeed, the inherent properties of this digital technology provide fault-tolerance, immutability, transparency and full traceability of the stored transaction records, as well as coherent digital representations of physical assets and autonomous transaction executions. This paper presents AgriBlockIoT, a fully decentralized, blockchain-based traceability solution for Agri-Food supply chain management, able to seamless integrate IoT devices producing and consuming digital data along the chain. To effectively assess Agri Block IoT, first, we defined a classical use-case within the given vertical domain, namely from-farm-to-fork. Then, we developed and deployed such use-case, achieving traceability using two different blockchain implementations, namely Ethereum and Hyperledger Sawtooth. Finally, we evaluated and compared the performance of both the deployments, in terms of latency, CPU, and network usage, also highlighting their main pros and cons.

CHAPTER-3

PROBLEM STATEMENT AND OBJECTIVES

1. Lack of Transparency: The multi-layered nature of the agri-food supply chain leads to opacity, making it difficult to trace the origin and journey of food products. This lack of transparency can result in inefficiencies, fraud, and difficulty in verifying the authenticity of products.

2. Data Fragmentation: Information is often siloed across different stakeholders, each using disparate systems and standards. This fragmentation hinders seamless data sharing and creates barriers to end-to-end traceability.

3. Manual and Paper-based Processes: Many supply chain operations still rely on manual and paper-based
methods for recording and sharing information. This not only increases the risk of human error but also slows down the process of data retrieval and verification.
4. Food Safety Risks: In the event of a contamination or foodborne illness outbreak, the lack of efficient traceability systems delays response times and complicates the process of identifying and isolating the affected products, posing significant public health risks.

‘

OBJECTIVES

1.Provide End-to-End Traceability*: Enable all stakeholders to trace the journey of food products from farm to table with real-time visibility.

2. *Ensure Data Integrity and Security*: Leverage the immutable nature of blockchain to prevent data tampering and ensure that all recorded information is accurate and reliable.

3. *Facilitate Seamless Data Sharing*: Create a unified platform where data can be securely shared among all participants in the supply chain, breaking down silos and improving collaboration.

4. *Automate and Streamline Processes*: Reduce reliance on manual and paper-based processes by automating data entry and retrieval through smart contracts and IoT integrations.

EXISTING SYSTEM

1. In Existing system, In Before, to by a product from farmer to customer in directly. Soin this process there is no money growth in future profit for another client.
2. If Add some Client means the Selling Money will be increases.
3. Differences among the product rate from Farmer to Customer.

CHAPTER-4

TECHNOLOGY USED

4.1 Requirement Analysis

1. **Software Environment**

2. **Java Technology**

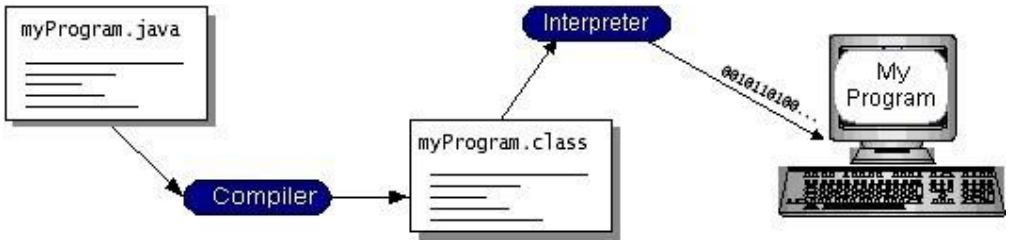
- Java technology is both a programming language and a platform.

A. The Java Programming Language

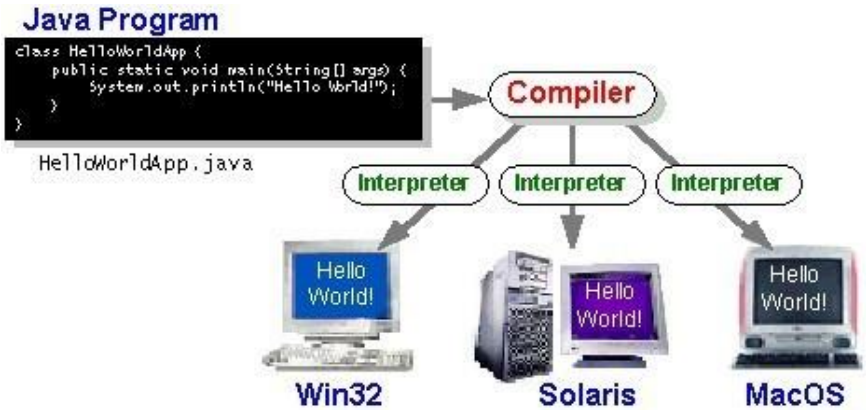
- The Java programming language is a high-level language that can be characterized by all of the following buzzwords:
  - Simple
  - Architecture neutral
  - Object oriented
  - Portable
  - Distributed
  - High performance
  - Interpreted
  - Multithreaded
  - Robust
  - Dynamic
  - Secure

- -

With most programming languages, you either compile or interpret a program so that you can run it on your computer. The Java programming language is unusual in that a program is both compiled and interpreted. With the compiler, first you translate a program into an intermediate language called *Java byte codes* —the platform independent codes interpreted by the interpreter on the Java platform. The interpreter parses and runs each Java byte code instruction on the computer. Compilation happens just once; interpretation occurs each time the program is executed. The following figure illustrates how this works.



You can think of Java byte codes as the machine code instructions for the *Java Virtual Machine* (Java VM). Every Java interpreter, whether it's a development tool or a Web browser that can run applets, is an implementation of the Java VM. Java byte codes help make "write once, run anywhere" possible. You can compile your program into byte codes on any platform that has a Java compiler. The byte codes can then be run on any implementation of the Java VM. That means that as long as a computer has a Java VM, the same program written in the Java programming language can run on Windows 2000, a Solaris workstation, or on an iMac.



3. The Java Platform

- A *platform* is the hardware or software environment in which program runs. We've already mentioned some of the most popular platforms like Windows 2000, Linux, Solaris, and MacOS. Most platforms can be described as a combination of the operating system and hardware. The Java platform differs from most other platforms in that it's a software-only platform that runs on top of other hardware-based platforms.

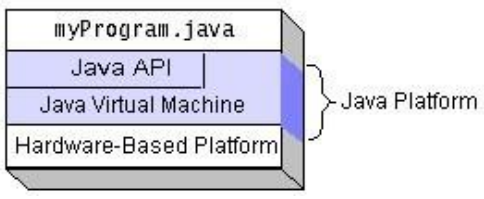
The Java platform has two components:

- The *Java Virtual Machine* (Java VM)
- The *Java Application Programming Interface* (Java API)

You've already been introduced to the Java VM. It's the base for the Java platform and is ported onto various hardware-based platforms.

The Java API is a large collection of ready-made software components that provide many useful capabilities, such as graphical user interface (GUI) widgets. The Java API is grouped into libraries of related classes and interfaces; these libraries are known as *packages*. The next section, What Can Java Technology Do? Highlights what functionality some of the packages in the Java API provide.

The following figure depicts a program that's running on the Java platform. As the figure shows, the Java API and the virtual machine insulate the program from the hardware.



Native code is code that after you compile it, the compiled code runs on a specific hardware platform. As a platform-independent environment, the Java platform can be a bit slower than native code. However, smart compilers, well-tuned interpreters, and just-in-time byte code compilers can bring performance close to that of native code without threatening portability.

4.2 What Can Java Technology Do?

The most common types of programs written in the Java programming language are *applets* and *applications*. If you've surfed the Web, you're probably already familiar with applets. An applet is a program that adheres to certain conventions that allow it to run within a Java- enabled browser.

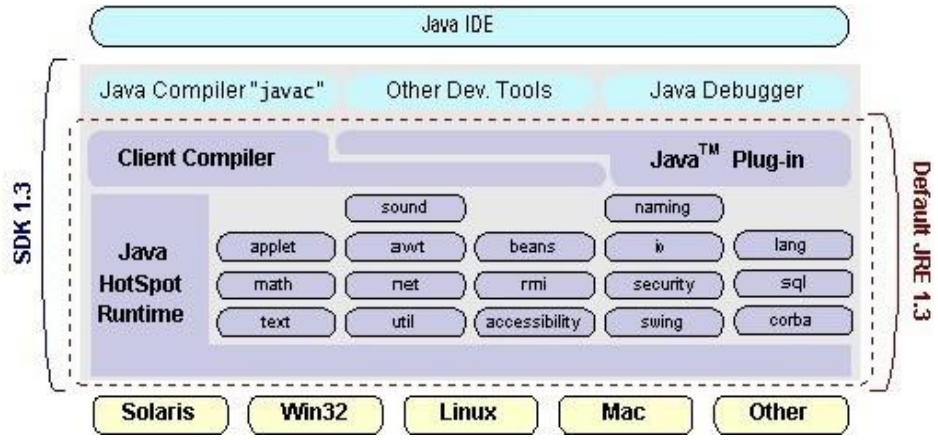
However, the Java programming language is not just for writing cute, entertaining applets for the Web. The general-purpose, high-level Java programming language is also a powerful software platform. Using the generous API, you can write many types of programs.

An application is a standalone program that runs directly on the Java platform. A special kind of application known as a **server** serves and supports clients on a network. Examples of servers are Web servers, proxy servers, mail servers, and print servers. Another specialized program is a **servlet**. A servlet can almost be thought of as an applet that runs on the server side. Java Servlets are a popular choice for building interactive web applications, replacingthe use of CGI scripts. Servlets are similar to applets in that they are runtime extensions of applications. Instead of working in browsers, though, servlets run within Java Web servers, configuring or tailoring the server. How does the API support all these kinds of programs? It does so with packages of software components that provides a wide range of functionality. Every full implementation of the Java platform gives you the following features:

- **The essentials**: Objects, strings, threads, numbers, input and output, data structures, system properties, date and time, and so on.
- **Applets**: The set of conventions used by applets.
- **Networking**: URLs, TCP (Transmission Control Protocol), UDP (User Datagram Protocol) sockets, and IP (Internet Protocol) addresses.

- **Internationalization**: Help for writing programs that can be localized for users worldwide. Programs can automatically adapt to specific locales and be displayed in the appropriate language.
- **Security**: Both low level and high level, including electronic signatures, public and private key management, access control, and certificates.
- **Software components**: Known as Java Beans ™, can plug into existing component architectures.
- **Object serialization**: Allows lightweight persistence and communication via Remote Method Invocation (RMI).
- **Java Database Connectivity (JDBC™)**: Provides uniform access to a wide range of relational databases.

The Java platform also has APIs for 2D and 3D graphics, accessibility, servers, collaboration, telephony, speech, animation, and more. The following figure depicts what is included in the Java 2 SDK.



4.3 How Will Java Technology Change My Life?

We can't promise you fame, fortune, or even a job if you learn the Java programming language. Still, it is likely to make your programs better and requires

less effort than other languages. We believe that Java technology will help you do the following:

- **Get started quickly**: Although the Java programming language is a powerful object-oriented language, it's easy to learn, especially for programmers already familiar with C or C++.
- **Write less code**: Comparisons of program metrics (class counts, method counts, and so on) suggest that a program written in the Java programming language can be four times smaller than the same program in C++.
- **Write better code**: The Java programming language encourages good coding practices, and its garbage collection helps you avoid memory leaks. Its object orientation, its JavaBeans component architecture, and its wide-ranging, easily extendible API let you reuse other people's tested code and introduce fewer bugs.
- **Develop programs more quickly**: Your development time may be as much as twice as fast versus writing the same program in C++. Why? You write fewer lines of code and it is a simpler programming language than C++.
- **Avoid platform dependencies with 100% Pure Java**: You can keep your program portable by avoiding the use of libraries written in other languages. The 100% Pure Java ™ Product Certification Program has a repository of historical process manuals, white papers, brochures, and similar materials online.
- **Write once, run anywhere**: Because 100% Pure Java programs are compiled into machine-independent byte codes, they run consistently on any Java platform.
- **Distribute software more easily**: You can upgrade applets easily from a central server. Applets take advantage of the feature of allowing new classes to be loaded "on the fly," without recompiling the entire program.

ODBC

Microsoft Open Database Connectivity (ODBC) is a standard programming interface for application developers and database systems providers. Before ODBC became a de facto standard for Windows programs to interface with database systems, programmers had to use proprietary languages for each database they wanted to connect to. Now, ODBC has made the choice of the database system almost irrelevant from a coding perspective, which is as it should be. Application developers have much more important things to worry about than the syntax that is needed to port their program from one database to another when business needs suddenly change.

Through the ODBC Administrator in Control Panel, you can specify the particular database that is associated with a data source that an ODBC application program is written to use. Think of an ODBC data source as a door with a name on it. Each door will lead you to a particular database. For example, the data source named Sales Figures might be a SQL Server database, whereas the Accounts Payable data source could refer to an Access database. The physical database referred to by a data source can reside anywhere on the LAN.

The ODBC system files are not installed on your system by Windows 95. Rather, they are installed when you setup a separate database application, such as SQL Server Client or Visual Basic 4.0. When the ODBC icon is installed in Control Panel, it uses a file called ODBCINST.DLL. It is also possible to administer your ODBC data sources through a stand-alone program called ODBCADM.EXE. There is a 16-bit and a 32-bit version of this program and each maintains a separate list of ODBC data sources.

From a programming perspective, the beauty of ODBC is that the application can be written to use the same set of function calls to interface with any data source, regardless of the database vendor. The source code of the application doesn't change whether it talks to Oracle or SQL Server. We only mention these two as an example. There are ODBC drivers available for several dozen popular database systems. Even

or SQL Server). The loading of the ODBC drivers is transparent to the ODBC application program. In a client/server environment, the ODBC API even handles many of the network issues for the application programmer.

The advantages of this scheme are so numerous that you are probably thinking there must be some catch. The only disadvantage of ODBC is that it isn't as efficient as talking directly to the native database interface. ODBC has had many detractors make the charge that it is too slow. Microsoft has always claimed that the critical factor in performance is the quality of the driver software that is used. In our humble opinion, this is true. The availability of good ODBC drivers has improved a great deal recently. And anyway, the criticism about performance is somewhat analogous to those who said that compilers would never match the speed of pure assembly language. Maybe not, but the compiler (or ODBC) gives you the opportunity to write cleaner programs, which means you finish sooner. Meanwhile, computers get faster every year.

JDBC

In an effort to set an independent database standard API for Java; Sun Microsystems developed Java Database Connectivity, or JDBC. JDBC offers a generic SQL database access mechanism that provides a consistent interface to a variety of RDBMSs. This consistent interface is achieved through the use of "plug-in" database connectivity modules, or drivers. If a database vendor wishes to have JDBC support, he or she must provide the driver for each platform that the database and Java run on.

To gain a wider acceptance of JDBC, Sun based JDBC's framework on ODBC. As you discovered earlier in this chapter, ODBC has widespread support on a variety of platforms. Basing JDBC on ODBC will allow vendors to bring JDBC drivers to market much faster than developing a completely new connectivity solution.

JDBC was announced in March of 1996. It was released for a 90 day public review that ended June 8, 1996. Because of user input, the final JDBC v1.0 specification was released soon after.

The remainder of this section will cover enough information about JDBC for you to know what it is about and how to use it effectively. This is by no means a complete overview of JDBC. That would fill an entire book.

JDBC Goals

Few software packages are designed without goals in mind. JDBC is one that, because of its many goals, drove the development of the API. These goals, in conjunction with early reviewer feedback, have finalized the JDBC class library into a solid framework for building database applications in Java.

The goals that were set for JDBC are important. They will give you some insight as to why certain classes and functionalities behave the way they do. The eight design goals for JDBC are as follows:

1. **SQL Level API**

The designers felt that their main goal was to define a SQL interface for Java. Although not the lowest database interface level possible, it is at a low enough level for higher-level tools and APIs to be created. Conversely, it is at a high enough level for application programmers to use it confidently. Attaining this goal allows for future tool vendors to "generate" JDBC code and to hide many of JDBC's complexities from the end user.

2. **SQL Conformance**

SQL syntax varies as you move from database vendor to database vendor. In an effort to support a wide variety of vendors, JDBC will allow any query

statement to be passed through it to the underlying database driver. This allows the connectivity module to handle non-standard functionality in a manner that is suitable for its users.

3. **JDBC must be implemental on top of common database interfaces**
The JDBC SQL API must "sit" on top of other common SQL level APIs. This goal allows JDBC to use existing ODBC level drivers by the use of a software interface. This interface would translate JDBC calls to ODBC and vice versa.

4. Provide a Java interface that is consistent with the rest of the Java system

Because of Java's acceptance in the user community thus far, the designers feel that they should not stray from the current design of the core Java system.

5. Keep it simple

This goal probably appears in all software design goal listings. JDBC is no exception. Sun felt that the design of JDBC should be very simple, allowing for only one method of completing a task per mechanism. Allowing duplicate functionality only serves to confuse the users of the API.

6. Use strong, static typing wherever possible

Strong typing allows for more error checking to be done at compile time; also, less error appear at runtime.

7. Keep the common cases simple

Because more often than not, the usual SQL calls used by the programmer are simple SELECT's, INSERT's, DELETE's and UPDATE's, these queries should be simple to perform with JDBC. However, more complex SQL statements should also be possible.

Finally, we decided to proceed the implementation using Java Networking. And

for dynamically updating the cache table we go for MS Access database.

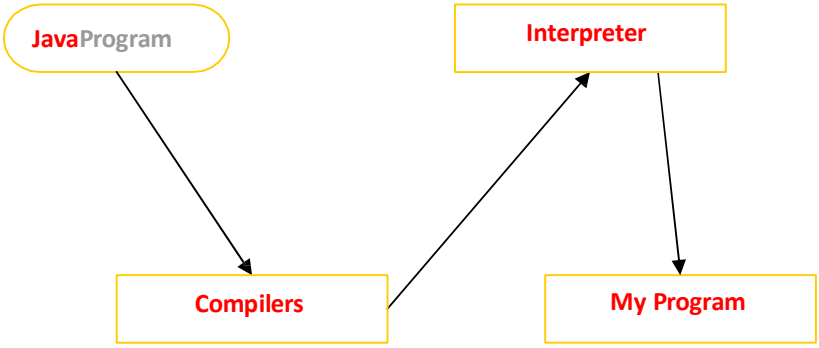
Java ha two things: a programming language and a platform.

Java is a high-level programming language that is all of the following

Simple	: Architecture-neutral
Object-oriented	; Portable
Distributed	: High-performance
Interpreted	: Multithreaded
Robust	: Dynamic
Secure	

Java is also unusual in that each Java program is both compiled and interpreted. With a compile you translate a Java program into an intermediate language called Java byte codes the platform-independent code instruction is passed and run on the computer.

Compilation happens just once; interpretation occurs each time the program is executed. The figure illustrates how this works.

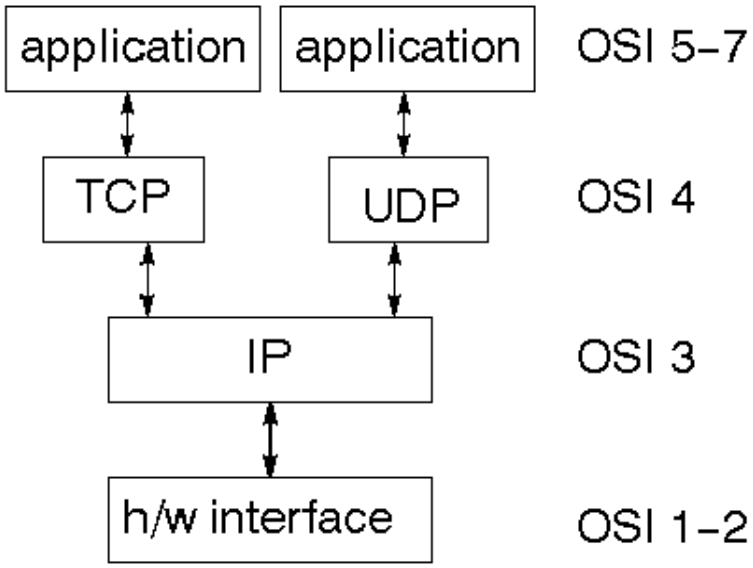


You can think of Java byte codes as the machine code instructions for the Java Virtual Machine (Java VM). Every Java interpreter, whether it's a Java development tool or a Web browser that can run Java applets, is an implementation of the Java VM. The Java VM can also be implemented in hardware.

Java byte codes help make "write once, run anywhere" possible. You can compile your Java program into byte codes on my platform that has a Java compiler. The byte codes can then be run any implementation of the Java VM. For example, the same Java program can run Windows NT, Solaris, and Macintosh.

TCP/IP stack:

The TCP/IP stack is shorter than the OSI one.



TCP is a connection-oriented protocol; UDP (User Datagram Protocol) is a connectionless protocol.

IP datagram's

The IP layer provides a connectionless and unreliable delivery system. It considers each datagram independently of the others. Any association between datagram must be supplied by the higher layers. The IP layer supplies a checksum that includes its own header. The header includes the source and destination addresses. The IP layer handles routing through an

Internet. It is also responsible for breaking up large datagram into smaller ones for transmission and reassembling them at the other end.

UDP

UDP is also connectionless and unreliable. What it adds to IP is a checksum for the contents of the datagram and port numbers. These are used to give a client/server model - see later.

TCP

TCP supplies logic to give a reliable connection-oriented protocol above IP. It provides a virtual circuit that two processes can use to communicate.

1. Internet addresses

In order to use a service, you must be able to find it. The Internet uses an address scheme for machines so that they can be located. The address is a 32 bit integer which gives the IP address. This encodes a network ID and more addressing. The network ID falls into various classes according to the size of the network address.

2. Network address

Class A uses 8 bits for the network address with 24 bits left over for other addressing. Class B uses 16 bit network addressing. Class C uses 24 bit network addressing and class D uses all 32.

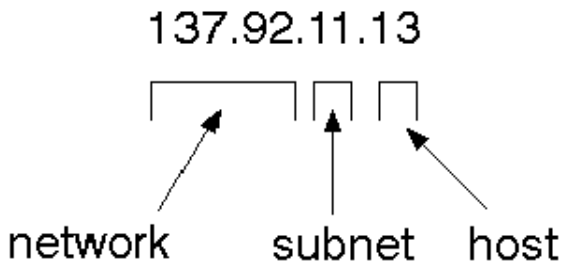
3. Subnet address

Internally, the UNIX network is divided into sub networks. Building 11 is currently on one sub network and uses 10-bit addressing, allowing 1024 different hosts.

1. Host address

8 bits are finally used for host addresses within our subnet. This places a limit of 256 machines that can be on the subnet.

2. Total address



The 32 bit address is usually written as 4 integers separated by dots

3. Port addresses

A service exists on a host, and is identified by its port. This is a 16 bit number. To send a message to a server, you send it to the port for that service of the host that it is running on. This is not location transparency! Certain of these ports are "well known".

4. Sockets

A socket is a data structure maintained by the system to handle network connections.

socket is created using the call socket. It returns an integer that is like a file descriptor. In fact, under Windows, this handle can be used with Read File and Write File functions.

```
#include <sys/types .h>

#include <sys/socket  .h>

Int socket(int family, int type, int protocol);
```

Here "family" will be AF_INET for IP communications, protocol will be zero, and type will depend on whether TCP or UDP is used. Two processes wishing to communicate over a network create a socket each. These are similar to two ends of a pipe - but the actual pipe does not yet exist.

5. J Free Chart

J Free Chart is a free 100% Java chart library that makes it easy for developers to display professional quality charts in their applications. J Free Chart's extensive feature set includes:

A consistent and well-documented API, supporting a wide range of chart types;

A flexible design that is easy to extend, and targets both server-side and client-side applications;

Support for many output types, including Swing components, image files (including PNG and JPEG), and vector graphics file formats (including PDF, EPS and SVG);

J Free Chart is "open source" or, more specifically, free software. It is distributed under the terms of the GNU Lesser General Public Licence (LGPL), which permits use in proprietary applications.

1. Map Visualizations

Charts showing values that relate to geographical areas. Some examples include: (a) population density in each state of the United States, (b) income per capita for each country in Europe, (c) life expectancy in each country of the world. The tasks in this project include:

Sourcing freely redistributable vector outlines for the countries of the world, states/provinces in particular countries (USA in particular, but also other areas);

Creating an appropriate dataset interface (plus default implementation), a rendered, and integrating this with the existing XY Plot class in J Free Chart;

Testing, documenting, testing some more, documenting some more.

2. Time Series Chart Interactivity

Implement a new (to J Free Chart) feature for interactive time series charts --- to display a separate control that shows a small version of ALL the time series data, with a sliding "view"

3. Dashboards

There is currently a lot of interest in dashboard displays. Create a flexible dashboard mechanism that supports a subset of J Free Chart types (dials, pies, thermometers, bars, and lines/time series) that can be delivered easily via both Java Web Start and an applet.

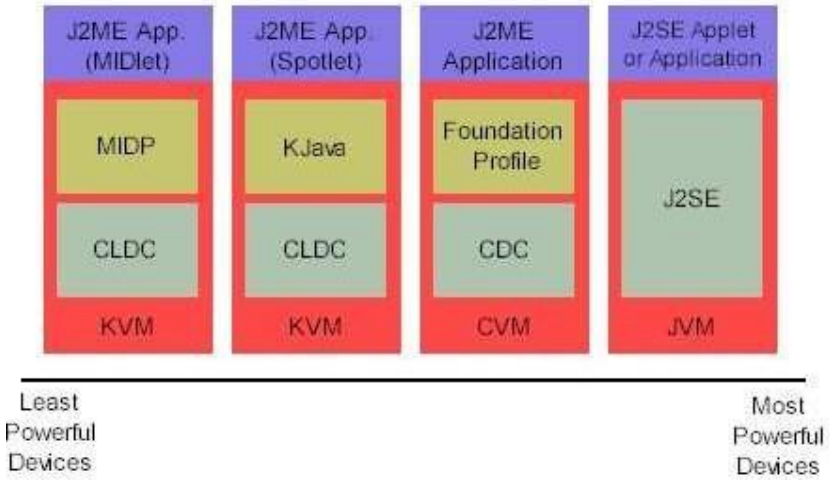
4. Property Editors

The property editor mechanism in Free Chart only handles a small subset of the properties that can be set for charts. Extend (or reimplement) this mechanism to provide greater end-user control over the appearance of the charts.

J2ME (Java 2 Micro edition):-

Sun Microsystems defines J2ME as "a highly optimized Java run-time environment targeting a wide range of consumer products, including pagers, cellular phones, screen- phones, digital set-top boxes and car navigation systems." Announced in June 1999 at the Java One Developer Conference, J2ME brings the cross-platform functionality of the Java language to smaller devices, allowing mobile wireless devices to share applications. With J2ME, Sun has adapted the Java platform for consumer products that incorporate or are based on small computing devices.

1. General J2ME architecture



J2ME uses configurations and profiles to customize the Java Runtime Environment (JRE). As a complete JRE, J2ME is comprised of a configuration, which determines the JVM used, and a profile, which defines the application by adding domain-specific classes. The configuration defines the basic run-time environment as a set of core classes and a specific JVM that run on specific types of devices. We'll discuss configurations in detail in the The profile defines the application; specifically, it adds domain-specific classes to the J2ME configuration to define certain uses for devices. We'll cover profiles in depth in the The following graphic depicts the relationship between the different virtual machines, configurations, and profiles. It also draws a parallel with the J2SE API and its Java virtual machine. While the J2SE virtual machine is generally referred to as a JVM, the J2ME virtual machines, KVM and CVM, are subsets of JVM. Both KVM and CVM can be thought of as a kind of Java virtual machine -- it's just that they are shrunken versions of the J2SE JVM and are specific to J2ME.

2. Developing J2ME applications

Introduction In this section, we will go over some considerations you need to keep in mind when developing applications for smaller devices. We'll take a look at the way the compiler is invoked when using J2SE to compile J2ME applications. Finally, we'll explore packaging and deployment and the role pre verification plays in this process.

3. Design considerations for small devices

Developing applications for small devices requires you to keep certain strategies in mind during the design phase. It is best to strategically design an application for a small device before you begin coding. Correcting the code because you failed to consider all of the "gotchas" before developing the application can be a painful process. Here are some design strategies to consider:

* Keep it simple. Remove unnecessary features, possibly making those features a separate, secondary application.

Smaller is better. This consideration should be a "no brainer" for all developers. Smaller applications use less memory on the device and require shorter installation times. Consider packaging your Java applications as compressed Java Archive (jar) files.

* Minimize run-time memory use. To minimize the amount of memory used at run time, use scalar types in place of object types. Also, do not depend on the garbage collector. You should manage the memory efficiently yourself by setting object references to null when you are finished with them. Another way to reduce run-time memory is to use lazy instantiation, only allocating objects on an as-needed basis. Other ways of reducing overall and peak memory use on small devices are to release resources quickly, reuse objects, and avoid exceptions.

4. Configurations overview

The configuration defines the basic run-time environment as a set of core classes and a specific JVM that run on specific types of devices. Currently, two configurations exist for J2ME, though others may be defined in the future:

* **Connected Limited Device Configuration (CLDC)**is used specifically with the KVM for 16-bit or 32-bit devices with limited amounts of memory. This is the configuration (and the virtual machine) used for developing small J2ME applications. Its size limitations make CLDC more interesting and challenging (from a development point of view) than CDC. CLDC is also the configuration that we will use for developing our drawing tool application. An example of a small wireless device running small applications is a Palm hand-held computer.

* **Connected Device Configuration (CDC)**is used with the C virtual machine (CVM) and is used for 32-bit architectures requiring more than 2 MB of memory. An example of such a device is a Net TV box.

5. J2ME profiles

What is a J2ME profile?

As we mentioned earlier in this tutorial, a profile defines the type of device supported. The Mobile Information Device Profile (MIDP), for example, defines classes for cellular phones. It adds domain-specific classes to the J2ME configuration to define uses for similar devices. Two profiles have been defined for J2ME and are built upon CLDC: K Java and MIDP. Both K Java and MIDP are associated with CLDC and smaller devices. Profiles are built on top of configurations. Because profiles are specific to the size of the device (amount of memory) on which an application runs, certain profiles are associated with certain configurations.

A skeleton profile upon which you can create your own profile, the Foundation Profile, is available for CDC.

Profile 1: K Java

K Java is Sun's proprietary profile and contains the K Java API. The K Java profile is built on top of the CLDC configuration. The K Java virtual machine, KVM, accepts the same byte codes and class file format as the classic J2SE virtual machine. K Java contains a Sun-specific API that runs on the Palm OS. The K Java API has a great deal in common with the J2SE Abstract Windowing Toolkit (AWT). However,

because it is not a standard J2ME package, its main package is com. Sun .k java. We'll learn more about the K Java API later in this tutorial when we develop some sample applications.

Profile 2: MIDP

MIDP is geared toward mobile devices such as cellular phones and pagers. The MIDP, like K Java, is built upon CLDC and provides a standard run-time environment that allows new applications and services to be deployed dynamically on end user devices. MIDP is a

common, industry-standard profile for mobile devices that is not dependent on a specific vendor. It is a complete and supported foundation for mobile application

development. MIDP contains the following packages, the first three of which are core CLDC packages, plus three MIDP-specific packages.

* Java .lang

* java.io

* java. util

* javax.microedition.io

* java x .micro edition

* java x. micro edition

* java x .micro edition

INPUT DESIGN:

The input design is the link between the information system and the user. It comprises the developing specification and procedures for data preparation and those steps are necessary to put transaction data in to a usable form for processing can be achieved by inspecting the computer to read data from a written or printed document or it can occur by having people keying the data directly into the system. The design of input focuses on controlling the amount of input required, controlling the errors, avoiding delay, avoiding extra steps and keeping the process simple. The input is designed in such a way so that it provides security and ease of use with retaining the privacy. Input Design considered the following things:

- What data should be given as input?
- How the data should be arranged or coded?
- The dialog to guide the operating personnel in providing input.
- Methods for preparing input validations and steps to follow when error occur.

OBJECTIVES

1. Input Design is the process of converting a user-oriented description of the input into a computer-based system. This design is important to avoid errors in the data input process and show the correct direction to the management for getting correct information from the computerized system.

2. It is achieved by creating user-friendly screens for the data entry to handle large volume of data. The goal of designing input is to make data entry easier and to be free from errors. The data entry screen is designed in such a way that all the data manipulates can be performed. It also provides record viewing facilities.

3. When the data is entered it will check for its validity. Data can be entered with the help of screens. Appropriate messages are provided as when needed so that the user will not be in maize of instant. Thus, the objective of input design is to create an input layout that is easy to follow

OUTPUT DESIGN

A quality output is one, which meets the requirements of the end user and presents the information clearly. In any system results of processing are communicated to the users and to other system through outputs. In output design it is determined how the information is to be displaced for immediate need and also the hard copy output. It is the most important and direct source information to the user. Efficient and intelligent output design improves the system's relationship to help user decision-making.

1. Designing computer output should proceed in an organized, well thought out manner; the right output must be developed while ensuring that each output element is designed so that people will find the system can use easily and effectively. When analysis design computer output, they should Identify the specific output that is needed to meet the requirements.

2. Select methods for presenting information.

3. Create document, report, or other formats that contain information produced by the system.

The output form of an information system should accomplish one or more of the following objectives.

- Convey information about past activities, current status or projections of the
- Future.
- Signal important events, opportunities, problems, or warnings.
- Trigger an action.
- Confirm an action

System Requirements:

Hardware Requirements:

1. System : Intel i3.
2. Hard Disk : 40 GB.
3. Floppy Drive : 1.44 Mb.
4. Monitor : 15 VGA Colour.
5. Mouse : Logitech.
6. Ram : 512 Mb.

Software Requirements:

1. Operating system : Windows XP.
2. Coding Language : JAVA
3. Data Base : MYSQL
4. IDE : Netbeans IDE

Design Process

Proposed System

- We evaluate how well the suggested DR-SCM method for managing ASCs performs by contrasting it with two established techniques: the computational cost of hashes with various numbers of blocks and Q-learning methods.
- Displays the benefits (i.e. profits) of various ASC management strategies in various scenarios. With the exception of the heuristic method, the benefits of the various strategies all rise as the number of episodes rises.

1. Advantages

- High Performance and More secure
- Based on the Hash value key to One person to person it will be more secure.

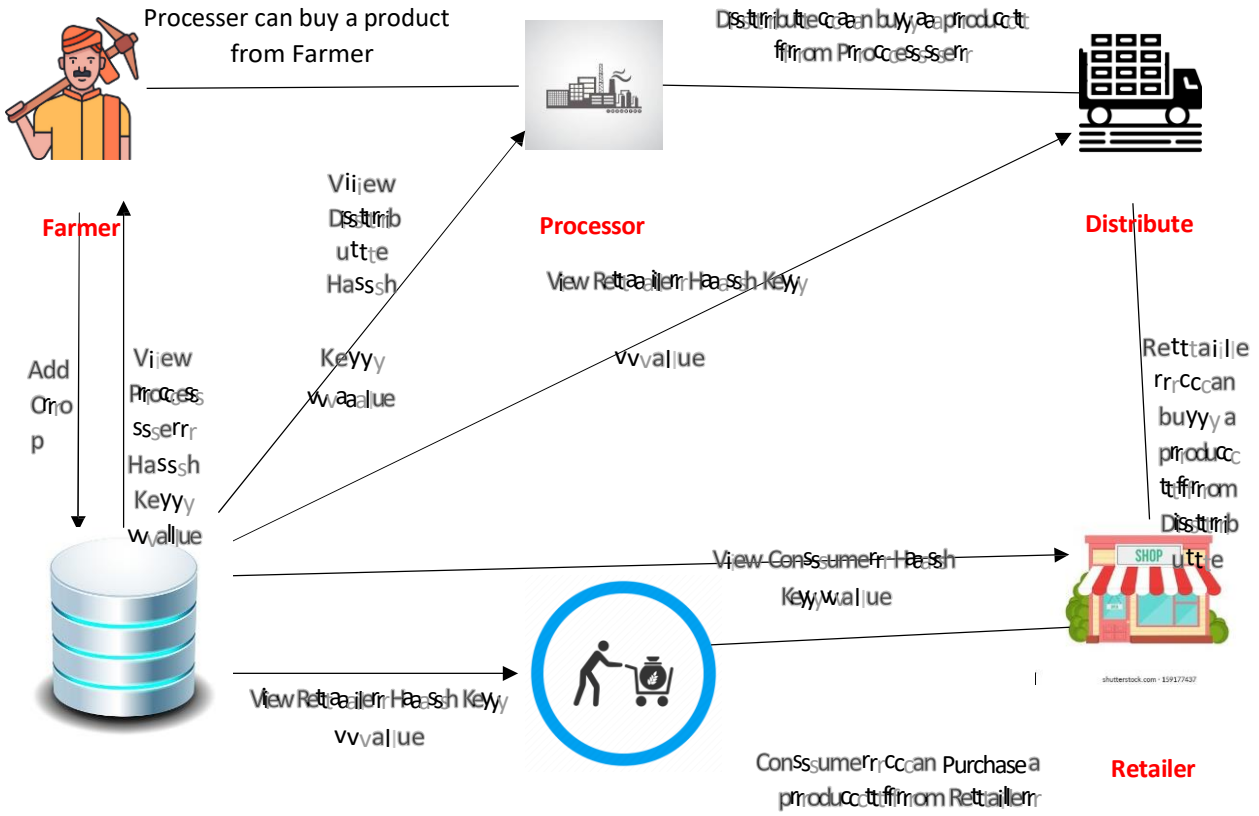
2. Algorithms

1. Collaborative filtering

Collaborative filtering is a family of algorithms where there are multiple ways to find similar users or items and multiple ways to calculate rating based on ratings of similar users

CHAPTER-5

METHODOLOGY



Dataset

Consumer

SYSTEM ARCHITECURE

MODULE DESCRIPTION

1. In this project have five modules:

- Farmer
- Processor
- Distributed
- Retailer
- Consumer
- Authority

2. Farmer

- Register the account with the basic information
- Login account with the correct username and password
- Farmer can add crop
- Once the processer can buy the product. Farmer can view the hash key value view status
- Logout

3. Processor

- Register the account with the basic information
- Login account with the correct username and password
- Processer can buy the product
- Once the Distributor can buy the product. Processor can view the Hash key value view status
- Logout

4. Distributed

- Register the account with the basic information
- Login account with the correct username and password
- Distributor can buy the product
- Once the Retailer can buy the product. Distributor can view the Hash key value view status
- Logout

5. Retailer

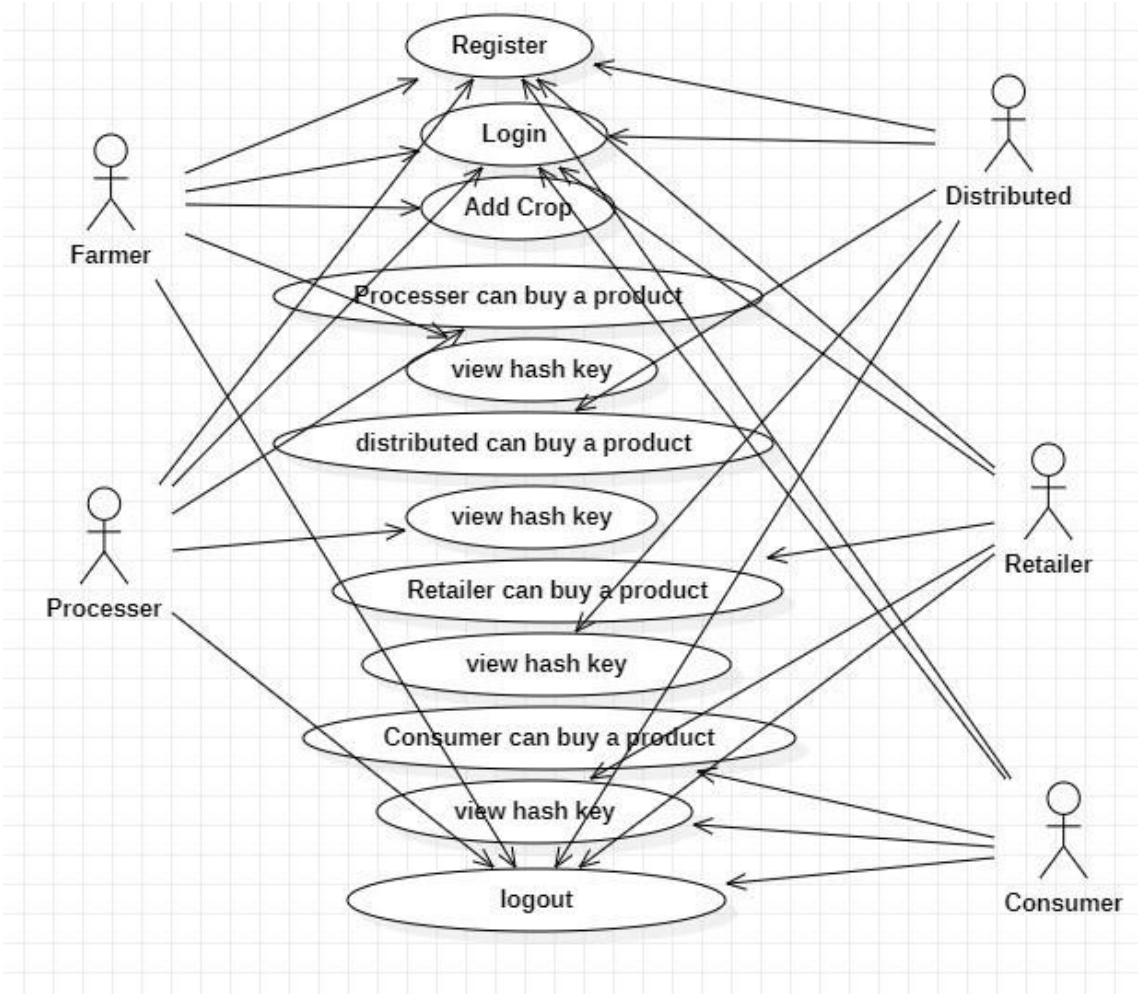
- Register the account with the basic information
- Login account with the correct username and password
- Retailer can buy the product
- Once the Consumer can buy the product. Retailer can view the Hash key value view status
- Logout

6. Consumer

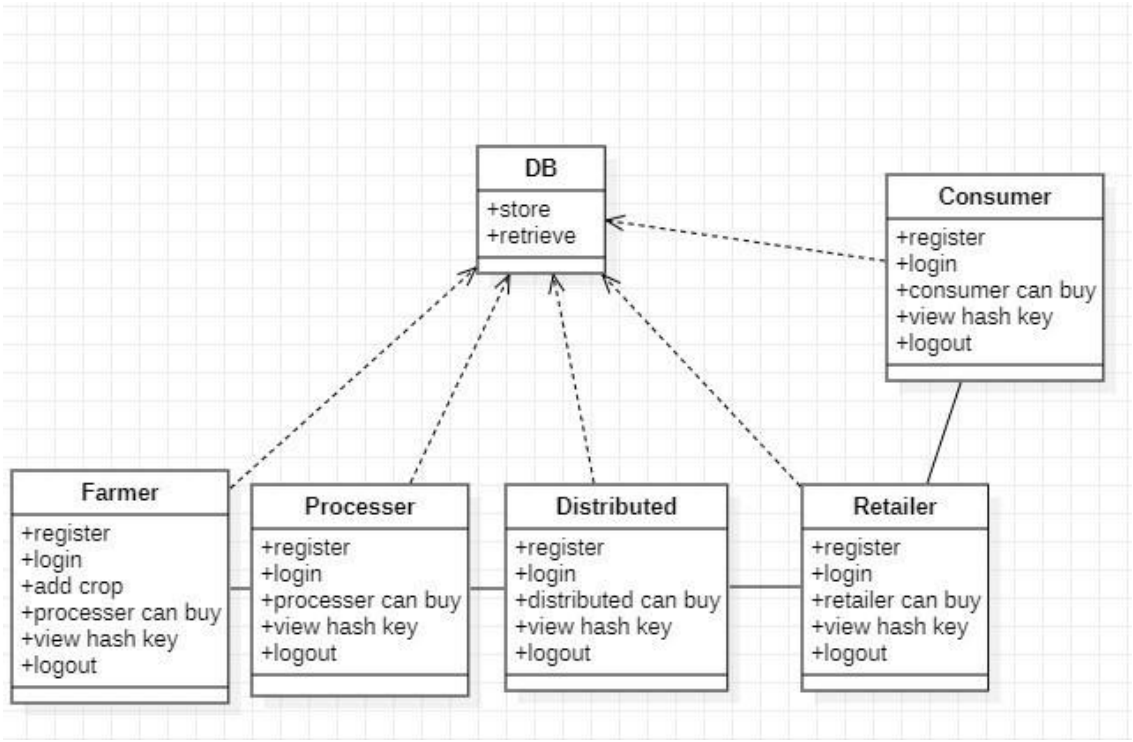
- Register the account with the basic information
- Login account with the correct username and password
- Consumer can buy the product
- Once the Consumer can Purchase the product. Consumer can view the Retailer Hash key value view status
- Logout

UML DIAGRAM:

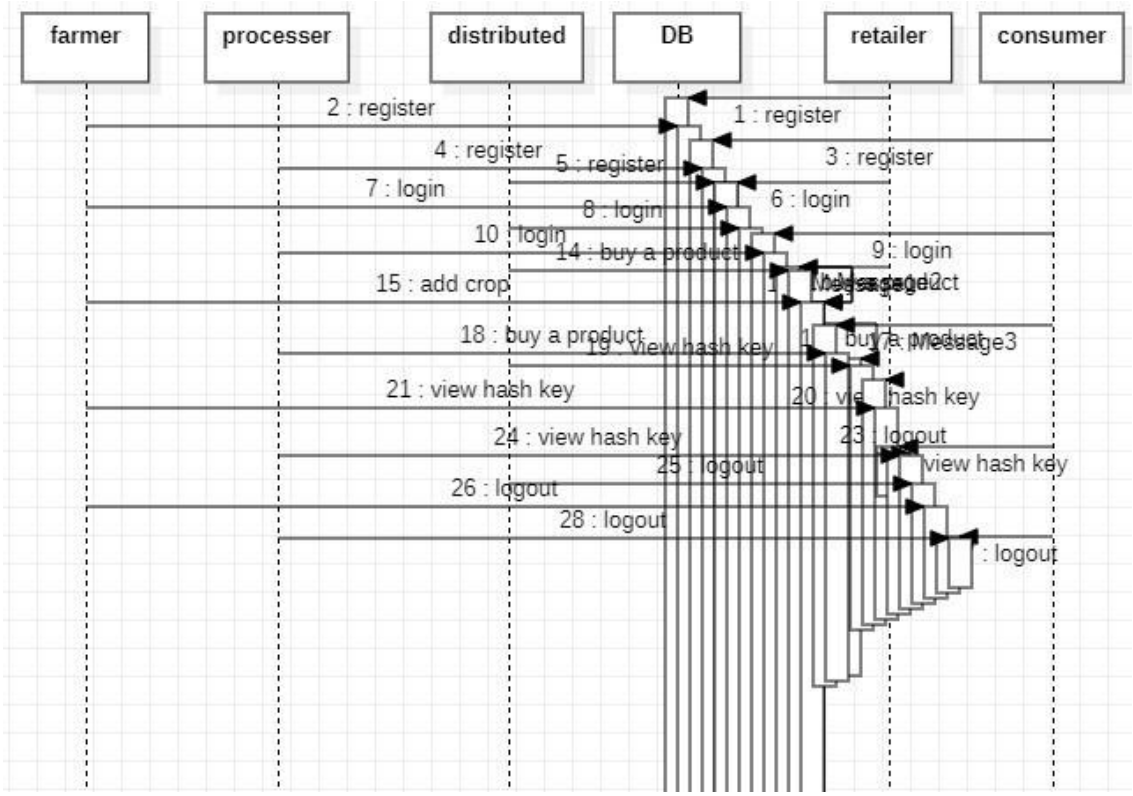
USE CASE DIAGRAM:



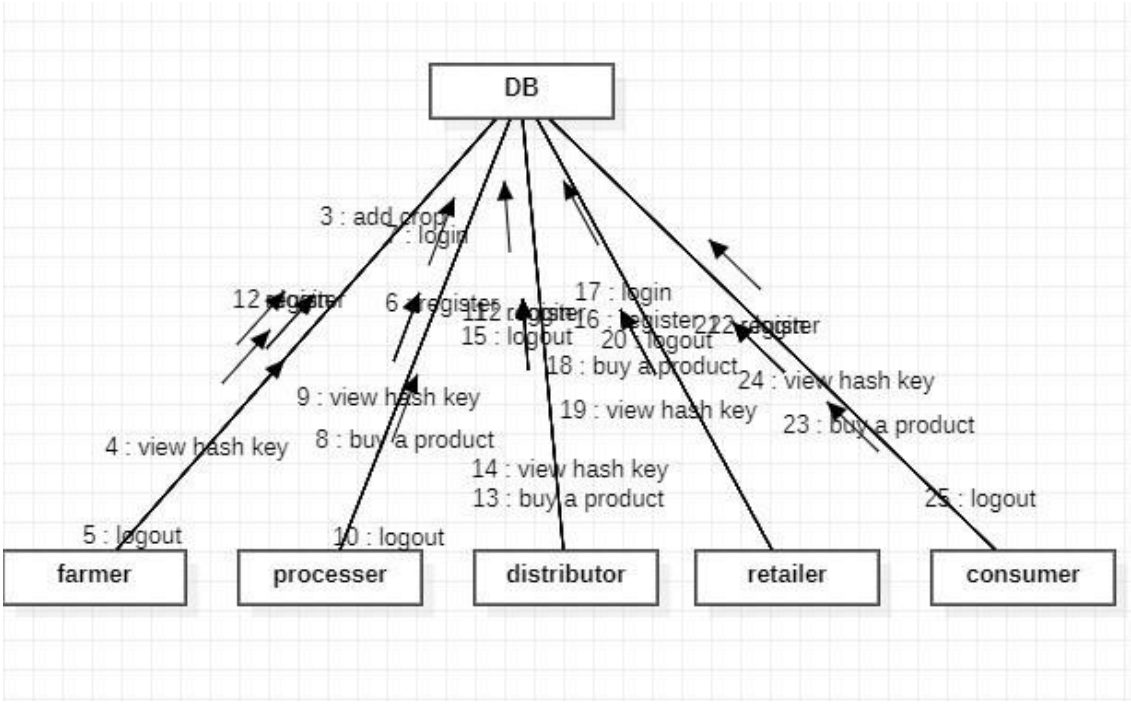
CLASS DIAGRAM:



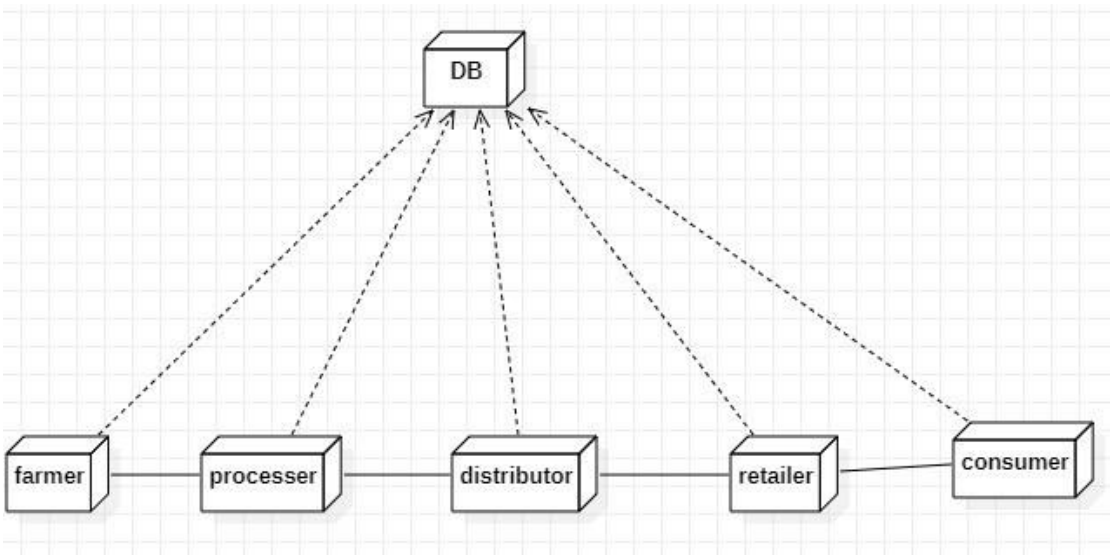
SEQUENCE DAIGRAM:



COLABORATION DAIAGRAM:

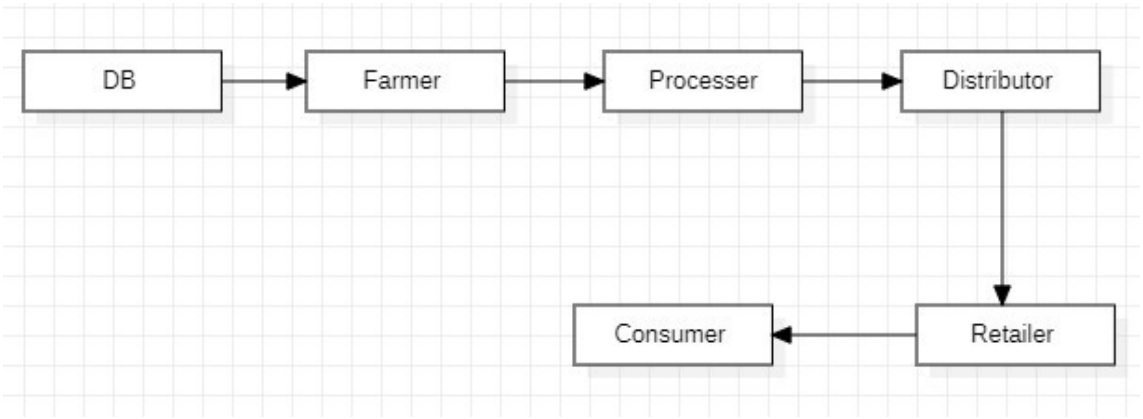


DEPLOYMENT DAIGRAM:

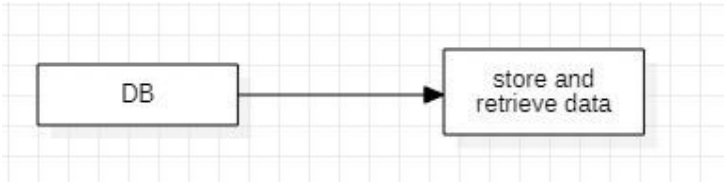


DATAFLOW DIAGRAM:

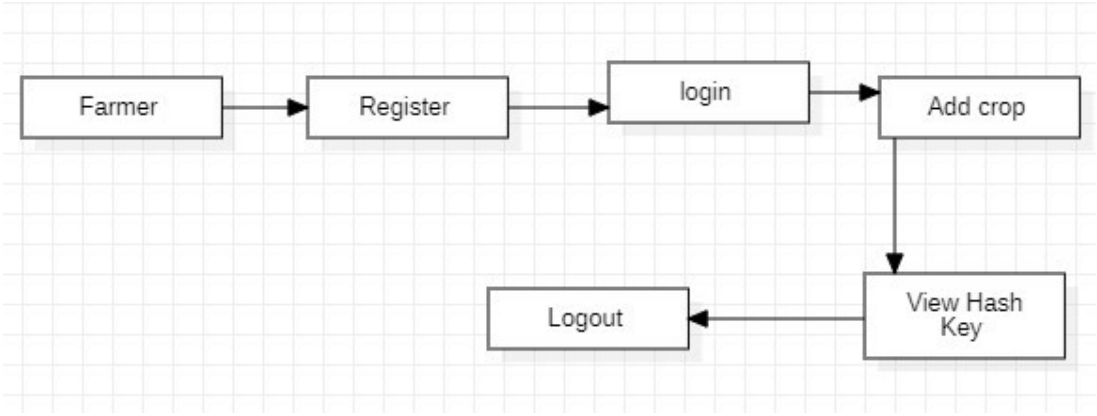
LEVEL 0:



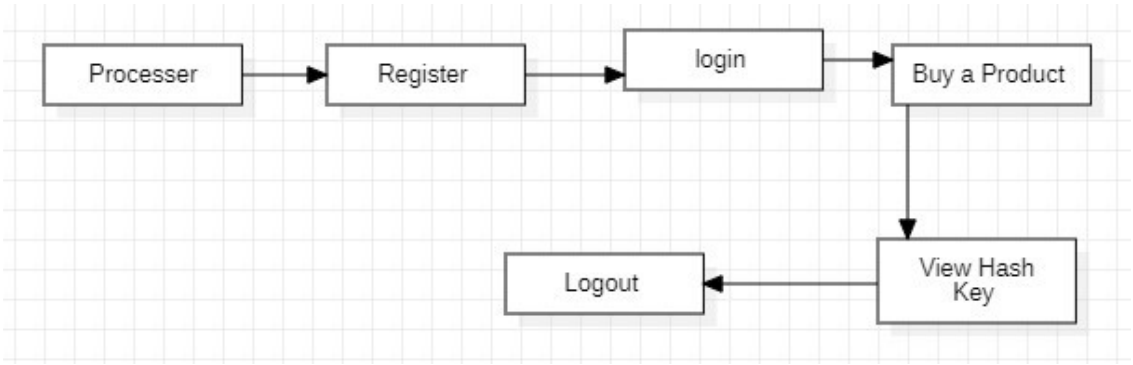
LEVEL 1:



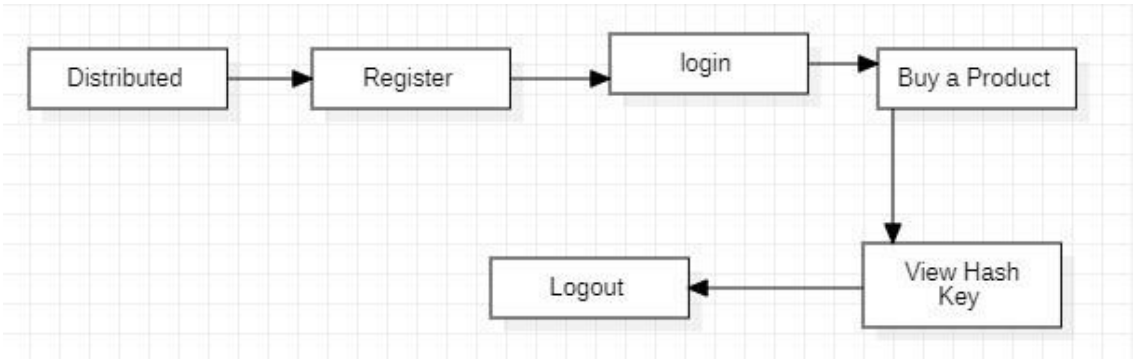
LEVEL 2:



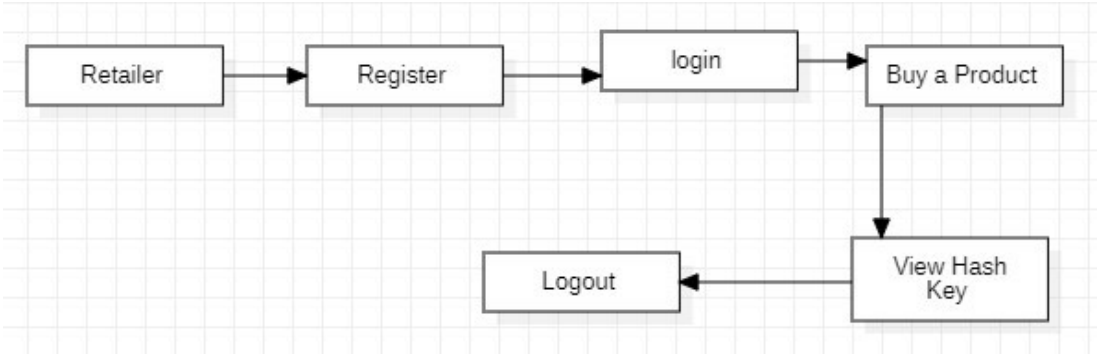
LEVEL 3:



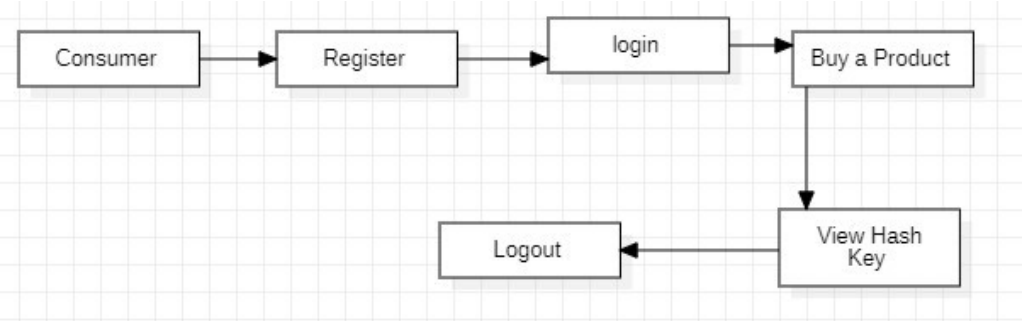
LEVEL 4:



LEVEL 5:



LEVEL 5:



CHAPTER-6

SYSTEM TESTING

The purpose of testing is to discover errors. Testing is the process of trying to discover every conceivable fault or weakness in a work product. It provides a way to check the functionality of components, sub-assemblies, assemblies and/or a finished product It is the process of exercising software with the intent of ensuring that the

Software system meets its requirements and user expectations and does not fail in an unacceptable manner. There are various types of test. Each test type addresses a specific testing requirement.

5.1 TYPES OF TESTS

1. Unit testing

Unit testing involves the design of test cases that validate that the internal program logic is functioning properly, and that program inputs produce valid outputs. All decision branches and internal code flow should be validated. It is the testing of individual software units of the application .it is done after the completion of an individual unit before integration. This is a structural testing, that relies on knowledge of its construction and is invasive. Unit tests perform basic tests at component level and test a specific business process, application, and/or system configuration.

Unittests ensure that each unique path of a business process performs accurately to the documented specifications and contains clearly defined inputs and expected results.

2. Integration testing

Integration tests are designed to test integrated software components to determine if they actually run as one program. Testing is event driven and is more concerned with the basic outcome of screens or fields. Integration tests demonstrate that although the components were individually satisfaction, as shown by successfully unit testing, the combination of components is correct and consistent. Integration testing is specifically aimed at exposing the problems that arise from the combination of components.

3. Functional test

Functional tests provide systematic demonstrations that functions tested are available as specified by the business and technical requirements, system documentation, and user manuals.

Functional testing is centered on the following items:

Valid Input : identified classes of valid input must be accepted.

Invalid Input : identified classes of invalid input must be rejected.

Functions : identified functions must be exercised.

Output : identified classes of application outputs must be exercised.

Systems/Procedures: interfacing systems or procedures must be invoked.

Organization and preparation of functional tests is focused on requirements, key functions, or special test cases. In addition, systematic coverage pertaining to identify Business process flows; data fields, predefined processes, and successive processes must be considered for testing. Before functional testing is complete, additional tests are identified and the effective value of current tests is determined.

4. System Test

System testing ensures that the entire integrated software system meets requirements. It tests a configuration to ensure known and predictable results. An example of system testing is the configuration oriented system integration test. System testing is based on process descriptions and flows, emphasizing pre-driven process links and integration points.

5. White Box Testing

White Box Testing is a testing in which in which the software tester has knowledge of the inner workings, structure and language of the software, or at least its purpose. It is purpose. It is used to test areas that cannot be reached from a black box level.

6. Black Box Testing

Black Box Testing is testing the software without any knowledge of the inner workings, structure or language of the module being tested. Black box tests, as most other kinds of tests, must be written from a definitive source document, such as specification or requirements document, such as specification or requirements document. It is a testing in which the

software under test is treated, as a black box. you cannot "see" into it. The test provides inputs and responds to outputs without considering how the software works.

Unit Testing:

Unit testing is usually conducted as part of a combined code and unit test phase of the software lifecycle, although it is not uncommon for coding and unit testing to be conducted as two distinct phases.

Test strategy and approach

Field testing will be performed manually and functional tests will be written in detail.

Test objectives

- All field entries must work properly.
- Pages must be activated from the identified link.
- The entry screen, messages and responses must not be delayed.

Features to be tested

- Verify that the entries are of the correct format
- No duplicate entries should be allowed
- All links should take the user to the correct page.

Integration Testing

Software integration testing is the incremental integration testing of two or more integrated software components on a single platform to produce failures caused by interface defects.

The task of the integration test is to check that components or software applications, e.g. components in a software system or – one step up – software applications at the company level – interact without error.

Test Results: All the test cases mentioned above passed successfully. No defects encountered.

Acceptance Testing

User Acceptance Testing is a critical phase of any project and requires significant participation by the end user. It also ensures that the system meets the functional requirements.

CHAPTER-7

SYSTEM STUDY

Feasibility Study

The feasibility of the project is analyzed in this phase and business proposal is put forth with a very general plan for the project and some cost estimates. During system analysis the feasibility study of the proposed system is to be carried out. This is to ensure that the proposed system is not a burden to the company. For feasibility analysis, some understanding of the major requirements for the system is essential.

Three key considerations involved in the feasibility analysis are

1. ECONOMICAL FEASIBILITY
2. TECHNICAL FEASIBILITY
3. SOCIAL FEASIBILITY

A. Economical Feasibility

This study is carried out to check the economic impact that the system will have on the organization. The amount of fund that the company can pour into the research and development of the system is limited. The expenditures must be justified. Thus, the developed system as well within the budget and this was achieved because most of the technologies used are freely available. Only the customized products had to be purchased.

B. Technical Feasibility

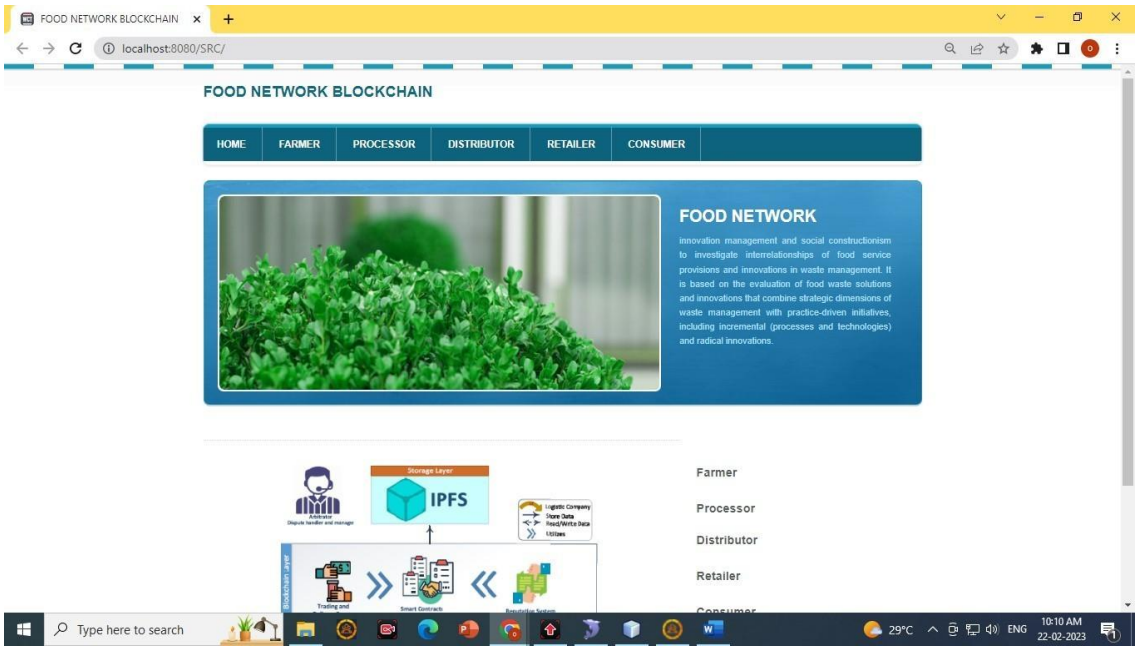
This study is carried out to check the technical feasibility, that is, the technical requirements of the system. Any system developed must not have a high demand on the available technical resources. This will lead to high demands on the available technical resources. This will lead to high demands being placed on the client. The developed system must have a modest requirement, as only minimal or null changes are required for implementing this system.

C. Social Feasibility

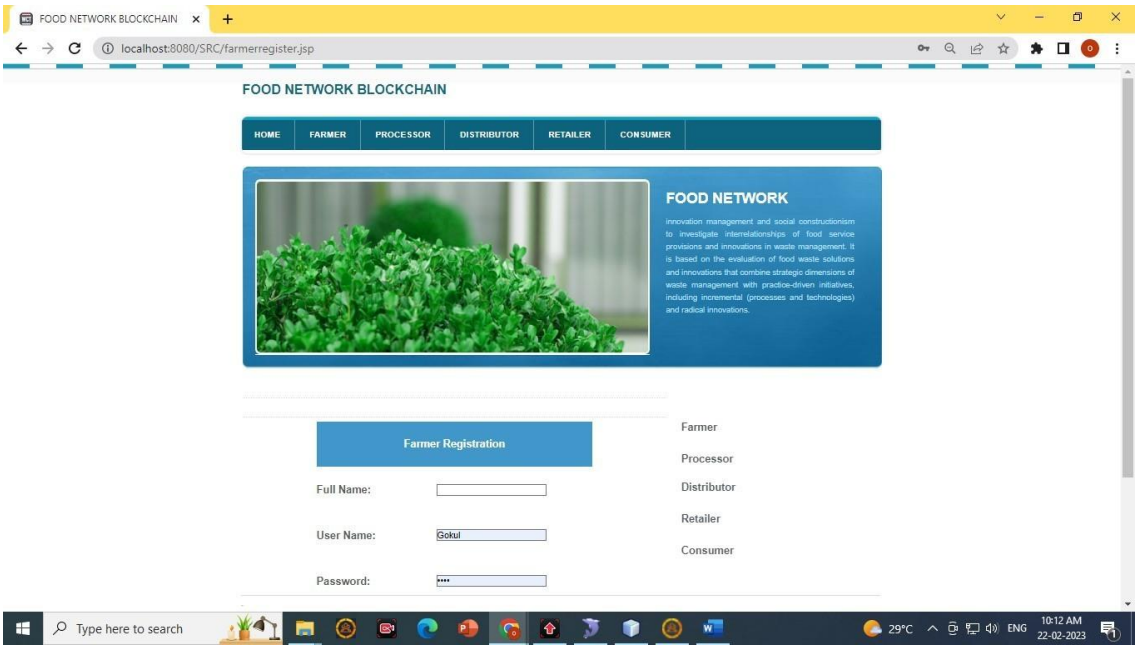
The aspect of study is to check the level of acceptance of the system by the user. This includes the process of training the user to use the system efficiently. The user must not feel threatened by the system, instead must accept it as a necessity. The level of acceptance by the users solely depends on the methods that are employed to educate the user about the system and to make him familiar with it. His level of confidence must be raised so that he is also able to make some constructive criticism, which is welcomed, a she is the final user of the system.

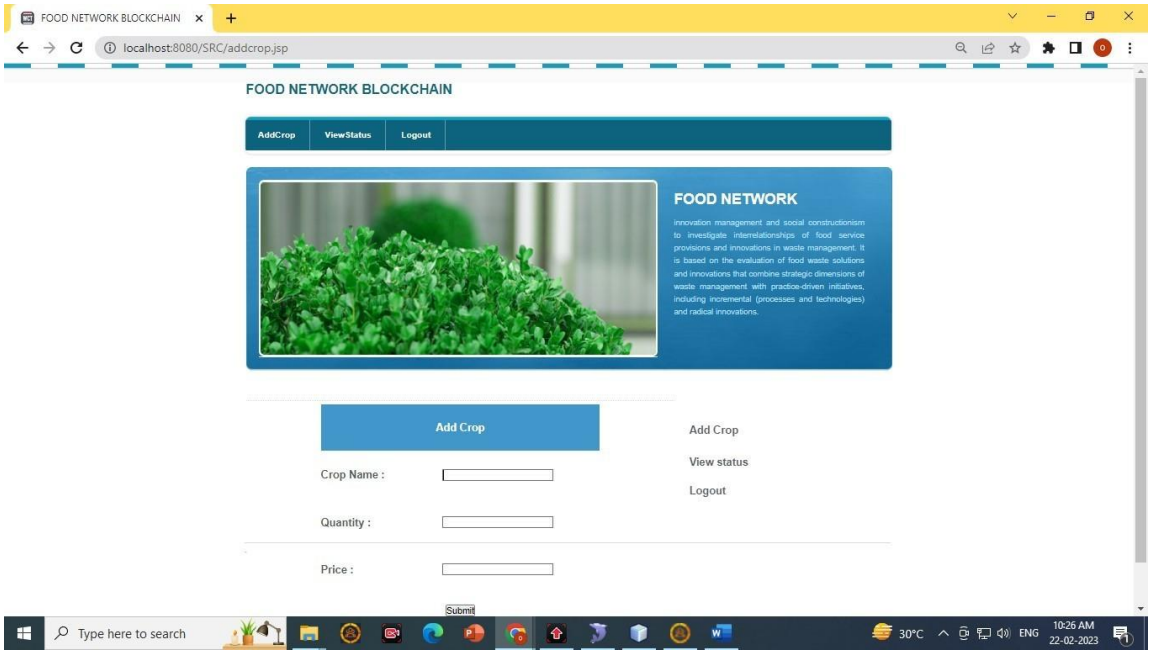
CHAPTER--8

RESULTS

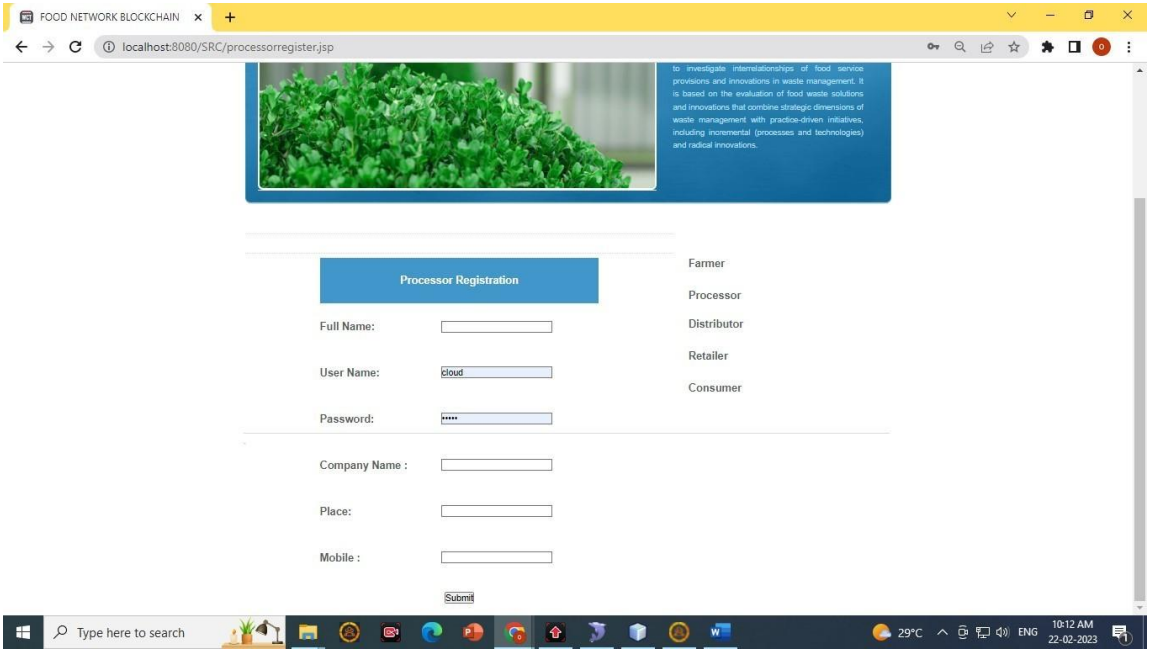


Login Page

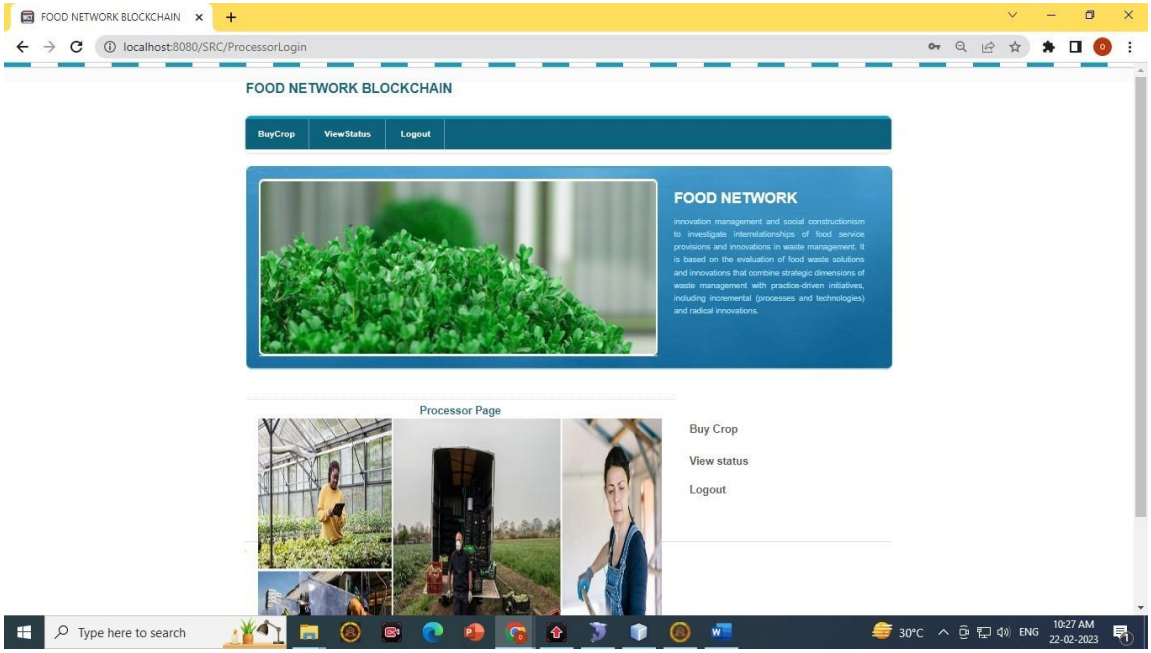




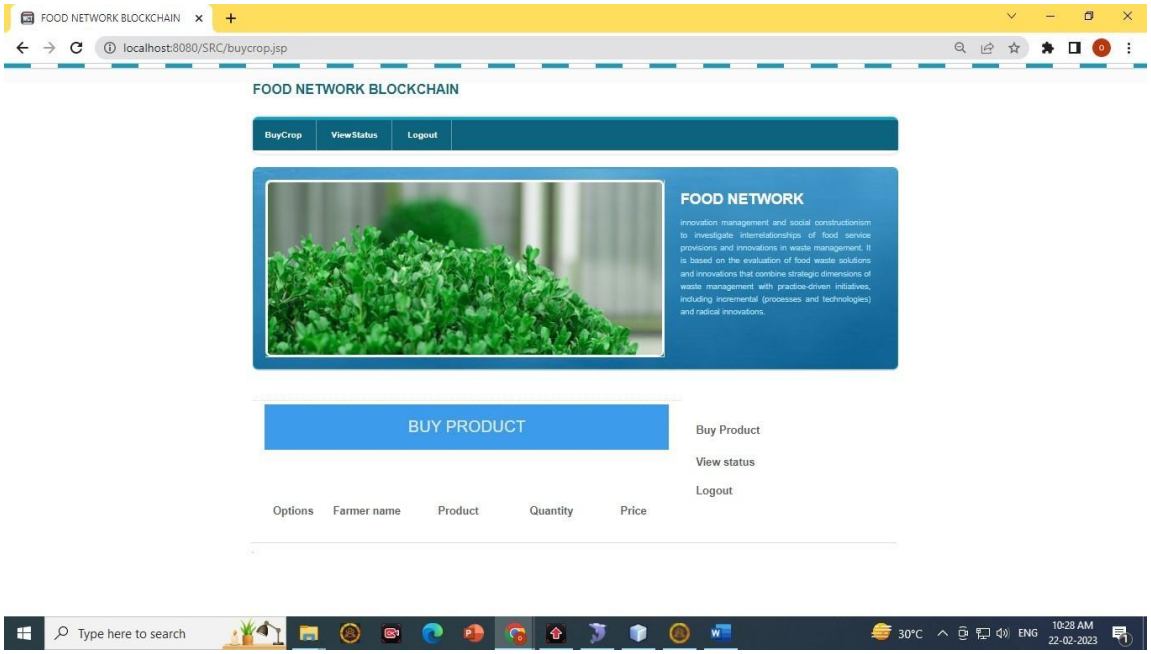
Add Crop



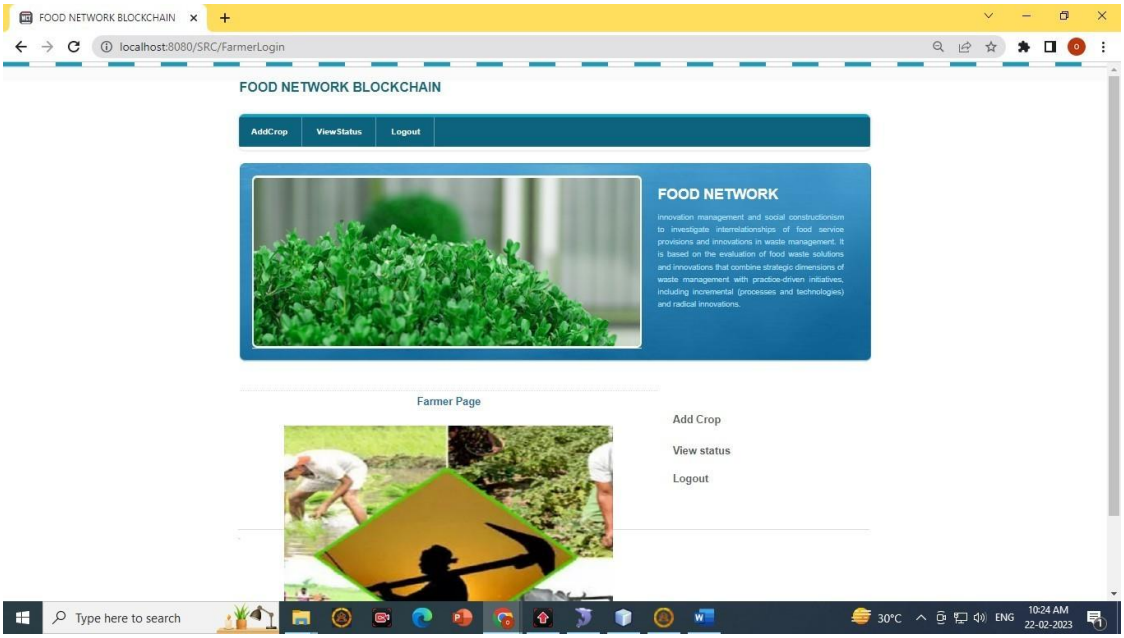
Processor Registration



Processor Page



Buy Product


FOOD NETWORK BLOCKCHAIN
localhost:8080/SRC/FarmerLogin
FOOD NETWORK BLOCKCHAIN
AddCrop
ViewStatus
Logout
FOOD NETWORK
innovation management and social constructionism to investigate interrelationships of food service provisions and innovations in waste management. It is based on the evaluation of food waste solutions and innovations that combine strategic dimensions of waste management with practice-driven initiatives, including incremental (processes and technologies) and radical innovations.
Farmer Page
Add Crop
View status
Logout
Type here to search
30°C
ENG
10:24 AM
22-02-2023

CHAPTER-9

CONCLUSION AND FUTURE ENHANCEMENT

Design a blockchain-based framework to guarantee the agri-food safety with product traceability in ASC systems. DR-SCM method to make decisions on the production and storage of agri-food products for optimizing product profits in ASCs. The extensive simulation experiments verify the effectiveness of the proposed blockchain-based framework and the DR-SCM method for ASC optimization. More specifically, the results show that the proposed blockchain-based ASC framework can well guarantee reliable product traceability.

REFERENCES

[1] D.-Y. Lin, C.-J. Juan, and C.-C. Chang, ‘‘Managing food safety with pricing, contracts and coordination in supply chains,’’ IEEE Access, vol. 7, pp. 150892–150909, 2019.

[2] H. Fan, ‘‘Theoretical basis and system establishment of China food safety intelligent supervision in the perspective of Internet of Things,’’ IEEE Access, vol. 7, pp. 71686–71695, 2019.

[3] M. Toledo-Hernández, T. Tscharntke, A. Tjoa, A. Anshary, B. Cyio, and T. C. Wanger, ‘‘Hand pollination, not pesticides or fertilizers, increases cocoa yields and farmer income,’’ Agricult.,Ecosyst. Environ., vol. 304, Dec. 2020, Art. no. 107160.

[4] J. Himmelstein, A. Ares, D. Gallagher, and J. Myers, ‘‘A meta-analysis of intercropping in Africa: Impacts on crop yield, farmer income, and integrated pest management effects,’’ Int. J. Agricult. Sustainability, vol. 15, no. 1, pp. 1–10, Jan. 2017.

[5] Y. Dong, Z. Fu, S. Stankovski, S. Wang, and X. Li, ‘‘Nutritional quality and safety traceability system for China’s leafy vegetable supply chain based on fault tree analysis and QR code,’’ IEEE Access, vol. 8, pp. 161261–161275, 2020.

[6] C. Ganeshkumar, M. Pachayappan, and G. Madanmohan, ‘‘Agri-food supply chain management:Literature review,’’ Intell. Inf. Manage., vol. 9, no. 2, pp. 68–96, 2017.

[7] Q. Lin, H. Wang, X. Pei, and J. Wang, ‘‘Food safety traceability system based on blockchain andEPCIS,’’ IEEE Access, vol. 7, pp. 20698–20707, 2019.

[8] H. Feng, X. Wang, Y. Duan, J. Zhang, and X. Zhang, ‘‘Applying blockchain technology to improve agri-food traceability: A review of development methods, benefits and challenges,’’ J. Cleaner Prod., vol. 260, Jul. 2020, Art. no. 121031.

[9] S. Nakamoto, ‘‘Bitcoin: A peer-to-peer electronic cash system,’’ White Paper, 2008. Accessed: Jun. 26, 2020. [Online]. Available: https://bitcoin.org/bitcoin.pdf

[10] F. Tian, ‘‘An agri-food supply chain traceability system for China based on RFID & blockchain technology,’’ in Proc. 13th Int. Conf. Service Syst. Service Manage. (ICSSSM), Jun. 2016, pp. 1–6.

[11] F. Tian, ‘‘A supply chain traceability system for food safety based on HACCP, blockchain & Internet of Things,’’ in Proc. 14th Int. Conf. Service Syst. Service Manage. (ICSSSM), Jun. 2017, pp.1–6.

[12] K. Toyoda, P. T. Mathiopoulos, I. Sasase, and T. Ohtsuki, ‘‘A novel blockchain-based product ownership management system (POMS) for anti- counterfeits in the post supply chain,’’ IEEE Access,vol. 5, pp. 17465–17477, 2017.

[13] M. P. Caro, M. S. Ali, M. Vecchio, and R. Giaffreda, ‘‘Blockchain-based traceability in agri-foodsupply chain management: A practical implementation,’’ in Proc. IoT Vertical Topical Summit Agricult. Tuscany (IOT Tuscany), May 2018, pp. 1–4.

[14] D. Mao, F. Wang, Z. Hao, and H. Li, ‘‘Credit evaluation system based on blockchain for multiple stakeholders in the food supply chain,’’ Int. J. Environ. Res. Public Health, vol. 15, no. 8, p. 1627, Aug. 2018.

[15] Y.-P. Lin, J. Petway, J. Anthony, H. Mukhtar, S.-W. Liao, C.-F. Chou, and Y.-F. Ho, ‘‘Blockchain: The evolutionary next step for ICT e-agriculture,’’ Environments, vol. 4, no. 3, p. 50, Jul. 2017.

[16] D. Tse, B. Zhang, Y. Yang, C. Cheng, and H. Mu, ‘‘Blockchain application in food supply information security,’’ in Proc. IEEE Int. Conf. Ind. Eng. Eng. Manage. (IEEM), Dec. 2017, pp. 1357–1361.

[17] S. A. Abeyratne and R. P. Monfared, ‘‘Blockchain ready manufacturing supply chain using distributed ledger,’’ Int. J. Res. Eng. Technol., vol. 5, no. 9, pp. 1–10, Sep. 2016.