PROJECT REPORT

DRUG TRACEABILITY

TEAM ID	NM2023TMID04410
PROJECT NAME	PROJECT-DRUG TRACEABILITY

NAME	REGISTER NUMBER
A.SUGITHA	812720104028
S.THANGAPANDIAN	812720104031
V.JOTHESHWARAN	812718104301
C.KARTHIKEYAN	812720104311

Drug traceablity

ABSTRACT

Drug traceability system is essentially important for public drug security and business of pharmaceutical companies, which aims to track or trace where the drug has been and where it has gone along the drug supply chain. Traditional centralized server-client technical solutions have been far from satisfying for their bad performances in data authenticity, privacy, system resilience and flexibility. We have proposed an entirely new blockchain system for drug traceability. This system is more secure and scalable than other alternatives on the market today. In addition, the proposed system is able to effectively prune its storage, resulting in a finally stable and usable blockchain storage solution.

Furthermore, the blockchain-based drug traceability system provides transparent and immutable records of each drug's journey, ensuring the integrity of the supply chain and safeguarding public health. By leveraging the decentralized nature of blockchain technology, the system eliminates the reliance on a single point of failure, enhancing system resilience and minimizing the risk of data manipulation or unauthorized access.

With its enhanced privacy features, the proposed system protects sensitive information while still allowing authorized stakeholders to access necessary data for regulatory compliance and quality assurance. Overall, this innovative blockchain solution offers a robust and efficient approach to drug traceability, addressing the limitations of traditional centralized systems and advancing the security and efficiency of pharmaceutical operations. .

CONTENTS

ABREVATION

Acryonym	Defination
DFD	Data Flow Diagram.
BC	Block Chain.
SDLC	Software Development Life Cycle.
UML	Unified Modeling Language.

List of Figures

Chapter 1

INTRODUCTION

1.1 Project Title

Drug traceability in healthcare supply chain using blockchain

1.2 Project Overview

In this project, we aim to develop a blockchain-based drug traceability solution using Java programming language. The goal is to improve transparency, security, and efficiency in tracking pharmaceutical products within the healthcare supply chain. We will design and implement a Java-based blockchain framework, integrate it with existing systems, and evaluate its performance, security, and scalability. Case studies or pilot implementations will be conducted to validate the solution's effectiveness. Ultimately, we will compare the blockchain solution with traditional approaches to assess its advantages in terms of cost-effectiveness, traceability accuracy, and overall supply chain efficiency.

1.3 Technical Keywords

Block chain, Drug Traceability, Security, Data Security

1.4 Motivation of Project

- Healthcare supply chain is a complex network of several independent entities that include raw material suppliers, manufacturer, distributor, pharmacies, hospitals and patients.

- Tracking supplies through this network is non-trivial due to several factors includinglack of information, centralized control and competing behaviour among stakeholders.

- Such complexity not only results in inefficiencies such as those highlighted through COVID-19 pandemic but can also aggravate the challenge of mitigating against counterfeit drugs as these can easily permeate the healthcare supply chain.

1.5 Problem definition

The healthcare supply chain lacks transparency, security, and efficiency in tracking pharmaceutical products, leading to risks of counterfeit drugs, compromised patient safety, and regulatory non-compliance. Traditional approaches suffer from manual recordkeeping, fragmented information, and limited traceability. There is a need for a secure and efficient drug traceability system using blockchain technology and Java programming language to ensure authenticity, quality, and provenance of drugs, while addressing scalability and data integrity challenges.

1.6 Research Objective

The objective of this research project is to develop and evaluate a blockchain-based solution using Java programming language for drug traceability in the healthcare supply chain. The primary goal is to enhance transparency, security, and efficiency in tracking pharmaceutical products from their origin to the end consumers, while leveraging the decentralized and immutable nature of blockchain technology.

1.7 Methodologies of Problem Solving

- Requirement Analysis: Conduct a thorough analysis of the healthcare supply chain requirements, including stakeholders, data flow, and regulatory compliance, to understand the specific needs and challenges.

- Design and Development: Design a blockchain-based drug traceability system using Java programming language, including smart contracts and integration with existing systems. Develop the system iteratively, ensuring modularity, security, and scalability.
- Performance Evaluation: Conduct simulations and benchmarking experiments to evaluate the system's performance, scalability, and efficiency under various transaction volumes and network conditions.

- Security and Privacy Measures: Implement robust security measures to protect sensitive data and ensure compliance with regulations like HIPAA. Conduct vulnerability assessments and implement appropriate measures to mitigate risks.

- Validation and Feedback: Collaborate with healthcare organizations or pharmaceutical companies to conduct case studies or pilot implementations,

gathering feedback and validating the effectiveness and usability of the solution in real-world scenarios.

- Comparative Analysis: Compare the blockchain-based drug traceability system with traditional approaches, considering factors such as cost-effectiveness, traceability accuracy, counterfeit prevention, and overall supply chain efficiency.

Chapter 2

LITERATURE SURVEY

Literature survey is the most important step in any kind of research. Before start developing we need to study the previous papers of our domain which we are working and on the basis of study we can predict or generate the drawback and start working with the reference of previous papers.

In this section, we briefly review the related work on Block chain technology.

R.Alvaro-Hermana, J. Fraile-Ardanuy, P. J. Zufiria, L. Knapen, and D. Janssens present the concept between two arrangements of electric vehicles, which fundamentally diminish the effect of the charging procedure on the power framework amid business hours. This trading approach is also economically beneficial for all the users involved in the trading process. An activity-based approach is used to predict the daily agenda and trips of a synthetic population for Flanders (Belgium) [1].

Y. Xiao, D. Niyato, P. Wang, and Z. Han provide a study of the possible flow and functional factors that enable DET in communication networks. Various design issues on how to implement DET in practice are discussed. An ideal approach is created for delay-tolerant remote controlled correspondence organizes in which every remote powered device can masterminded its information transmission and energy exchanging activities as indicated by present and future vitality accessibility [2].

J. Kang, R. Yu, X. Huang, S. Maharjan, Y. Zhang, and E. Hossain presents a work to accomplishes request reaction by giving motivating forces to releasing PHEVs to adjust nearby power request out of their own self-interests. Be that as it may, since exchange security and security insurance issues show genuine difficulties, they investigate a promising consortium block-chain innovation to enhance exchange security without dependence on a confided in outsider. A restricted P2P Electricity Trading framework with Consortium block- chain (PETCON) strategy is proposed to represent detailed activities of limited P2P power exchanging [3].

N. Z. Aitzhan and D. Svetinovic presents a work that address the issue of providing transaction security in decentralized smart grid energy trading without confidence on trusted third parties. We have developed a proof-of-concept for decentralized energy trading system using blockchain technology, multi-signatures, and anonymous encrypted messaging flows, enabling peers to anonymously negotiate energy prices and securely perform trading transactions [4].

M. Mihaylov, S. Jurado, N. Avellana, K. Van Moffaert, I. M. de Abril, and A. Now presents a work that shows decentralized computerized cash, called NRG-coin. Prosumers in the smart grid framework exchange privately made sustainable power source utilizing NRG-coins, the estimation of which is indented on an open cash trade advertise. Like Bit-coins, this money proposes various favorable circumstances over fiat cash, however not at all like Bit-coins it is made by infusing vitality into the matrix, as opposed to giving vitality on computational influence. Likewise, they make a novel exchanging worldview for purchasing and offering environmentally friendly power vitality in the smart grid network [5].

S. Barber et al presents a work that Bit-coin is isolated computerized cash which has pulled in a significant number of clients. They play out a top to bottom examination to comprehend what made Bit-coin so effective, while many years of research on cryptographic e-money have not prompt a vast scale appropriation. They ask additionally how Bit-coin could turn into a decent contender for seemingly perpetual stable money [6].

I. Alqassem et al presents a work that Bit-coin is constantly improved by an open source network, and different Bit-coin libraries, APIs, and elective usage are being created. All things considered, there is no up and coming convention contrast or design portrayal since the authority whitepaper was distributed. The work demonstrates an a la mode convention detail and design investigation of the Bit-coin framework. We play out this examination as the initial move towards determination of the cryptographic currency reference design [7].

K. Croman et al presents a work that the expanding fame of block-chain-based digital forms of money has made versatility an essential and earnest obligation. The work ponders how essential and incidental bottlenecks in Bit-coin restrict the ability of its present distributed overlay system to help generously higher throughputs and lower latencies. These outcomes propose that re-parameterization of square size and interruption ought to be seen just as a first augmentation toward accomplishing people to come, highstack block-chain conventions, and real advances will moreover require a fundamental reevaluating of specialized ways [8].

G. W. Peters and E. Panayipresents a work which give a diagram of the idea of blockchain innovation and its capacity to disturb the universe of managing an account through encouraging worldwide cash settlement, shrewd contracts, mechanized keeping money records and advanced resources. In such manner, they first give a concise outline of the center parts of this innovation, and in addition the second-age contract-based improvements [9].

L. Luu et al presents a work which gives another circulated understanding convention for authorization less block-chains called ELASTICO. ELASTICO scales exchange rates straightly with accessible estimation for mining: the more the calculation control in the system, the higher the quantity of exchange squares chosen per unit time. ELASTICO is

productive in its system messages and permit complex foes of up to one-fourth of the aggregate computational power [10].

Chapter 3

LITERATURE OUTCOME

The literature survey reveals several relevant studies on blockchain technology in various domains. Specifically, in the context of the project, the surveyed papers focus on different aspects of blockchain, such as decentralized energy trading, transaction security, scalability, and disruptive potential in banking.

These studies provide valuable insights and approaches that can be applied to the development of a blockchain-based drug traceability system in the healthcare supply chain. The surveyed literature highlights the benefits of blockchain, including enhanced transparency, secure transactions, and the elimination of the need for trusted third parties.

The surveyed papers also identify challenges and propose solutions related to scalability, security, and the design of consensus protocols in blockchain networks. These insights can inform the development process and help address potential drawbacks in the design and implementation of the drug traceability system.

Overall, the literature survey provides a strong foundation for the project, offering a comprehensive understanding of blockchain technology and its applications in various domains. This knowledge will guide the development of an effective and robust blockchain-based drug traceability solution in the healthcare supply chain.

Chapter 4

SOFTWARE REQUIREMENT AND SPECIFICATION

4.1 Assumption and Dependencies

Requirement analysis results in the specification of software's operational characteristics indicates software's interface with other system elements and establish constraints that software must meets. Requirement analysis allows the software engineer(sometime called Analyst or Modeler in this role) to elaborate on basis requirements during earlier requirement engineering task and build models that depict user scenarios, functional activities, problem classes and their relationship, system and class behavior and the flow of data as it is transformed. The requirements analysis task is a process of discovery, refinement, modeling and specification.

4.1.1 Project Scope

A blockchain system for drug traceability and regulation is presented. As time goes on, it rebuilds the entire service architecture, ensuring the authenticity and privacy of traceability data, while at the same time, achieving a finally stable blockchain storage There have also been presented algorithms that mirror the practical workflow of the medication supply chain.

4.1.2 User classes and characteristics

Our system have main three classes

i) Input data

ii) processing: System can train the dataset for note character and number and when system get the input data that time they perform there relevant operation as preprocess-

ing, feature extraction, classification and segmentation.

iii) Output: System have main class as output class where final result is show on screen as an output of recognize data.

4.2 Functional Requirements

The functional requirements for a system describe what system do.

i) The developed system should recognize tracaibility.

ii) System shall show the error message to the user when given input is not in the required format. iii) System must provide the quality of service to user.

4.3 External Interface Requirements

4.3.1 User Interfaces

The system specifies the user interfaces are as follows:

i) Open application. ii) Login. iii) Supply chain

4.3.2 Hardware Interfaces

The entire system interface with java and library.

4.3.3 Software Interfaces

The system works on database so system fetch the data.

4.3.4 Communication Interfaces

The system should also use standard protocols for image processing so we are uses various library for our project.

4.4 Non-Functional Requirements

Non-functional requirements are not directly related to the functional behavior of the system.

4.4.1 Performance Requirement

i) System must be user friendly, simple and interactive.

ii) The user interface is designed in such way that novice users with little knowledge of library, should be able to access this application.

iii) iii) Users are required to have some knowledge regarding library module.

4.5 System Requirements

4.5.1 Database Requirement

We use own database for our project were we use mysql database.

4.5.2 Software Requirement

i) Operating System:
Windows 8 or higher.

ii) Platform: Eclipse.

iii) Technologies used: Java.

4.5.3 Hardware Specifications

i) Processor: i3 or higher.

ii) Processor speed: 2.0 GHz.

iii) RAM: 8 GB. iv) Disk Space: 100
GB or higher.

Chapter 5

SYSYTEM DESIGN

5.1 System Architecture

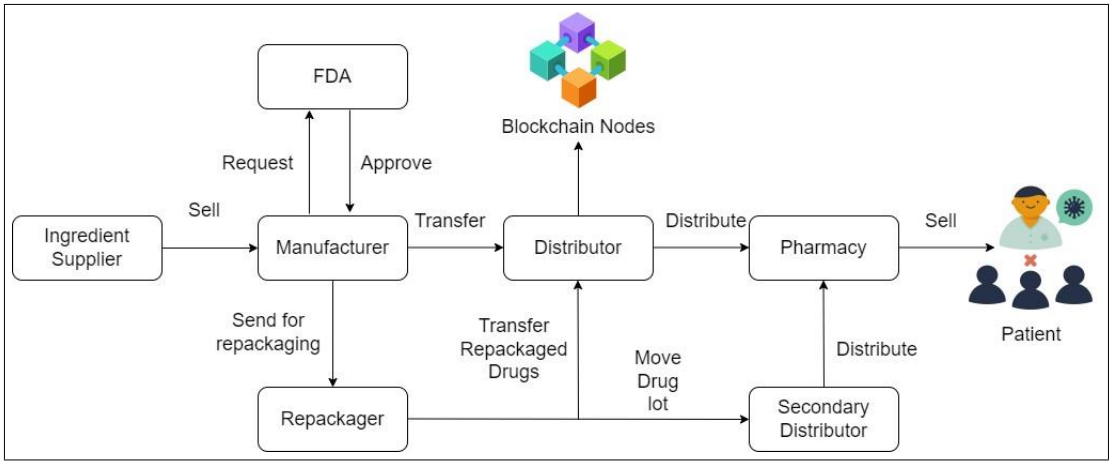


Figure 5.1: System Architecture

Our method identifies and involves significant stakeholders in the medication supply chain, such as the FDA, suppliers, manufacturers, distributors, pharmacies, and patients, whereas the FDA, suppliers, manufacturers, and wholesalers are the only ones involved.We make a concerted effort to identify and disentangle linkages between stakeholders, on-chain resources, smart contracts, and decentralised storage systems, which is currently lacking.We use smart contracts technology to achieve real-time, seamless traceability with push alerts, reducing the need for human intervention and, as a result, unnecessary delays.Each drug Lot is given its own smart contract, which generates an event whenever there is a change in ownership and sends a list of events to the app user.

5.2 Data Flow Diagram

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.

5.2.1 DFD

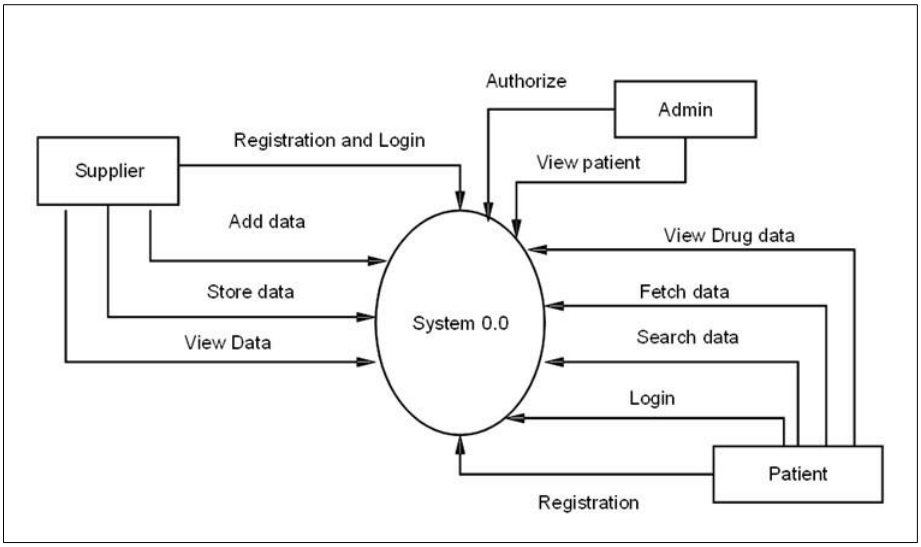


Figure 5.2: DFD

5.3 UML Diagram

5.3.1 Use Case Diagram

A Use Case Diagram consists of set of elements and the relationships between them. It depicts all the scenarios, regarding how our application interacts with users and other external systems to achieve the goals of application. The main components of a use case diagram include actors, use cases and their relationships. The use case is an external view of the system that represents some actions that the user performs to get a job done.

Actors are the users who interact with the application.

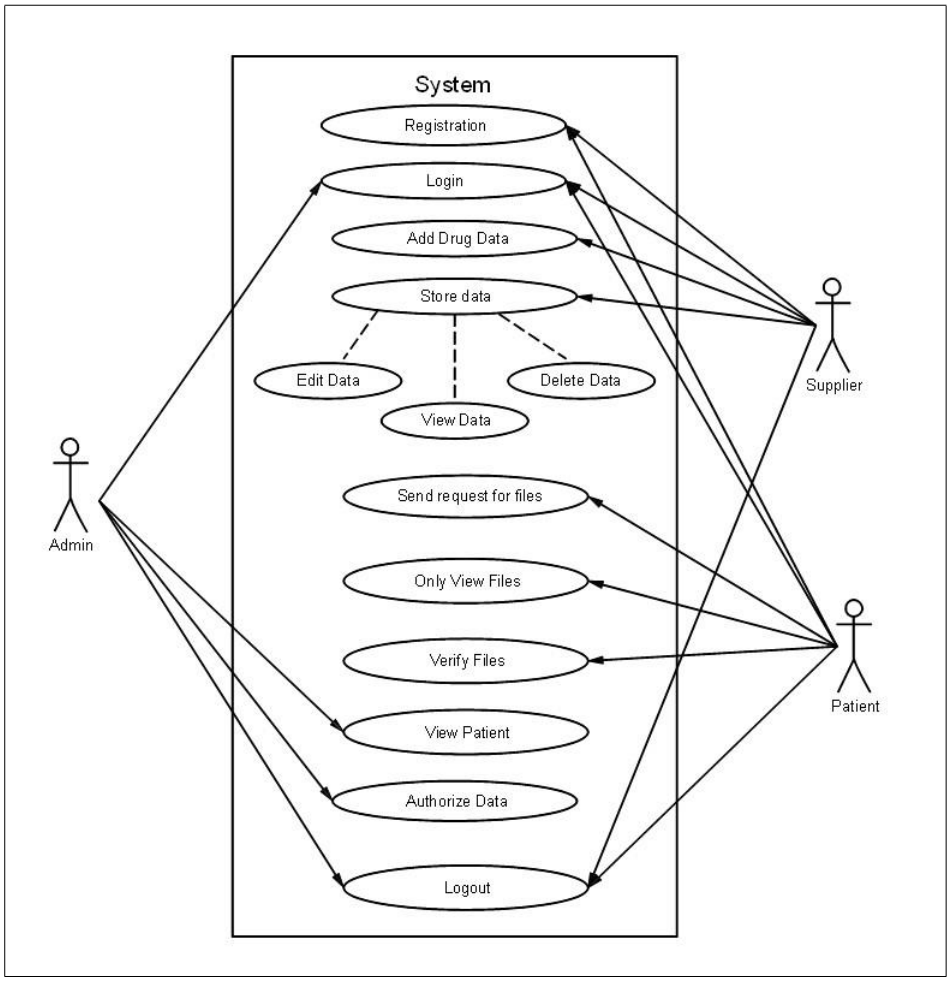


Figure 5.3: Use Case Diagram

5.3.2 Activity Diagram

Activity diagram is basically a flowchart to represent the flow from one activity to another activity. The activity can be described as an operation of the system. The main element of an activity diagram is the activity itself. An activity is a function performed by the system. After identifying the activities, we need to understand how they are associated with constraints and conditions.

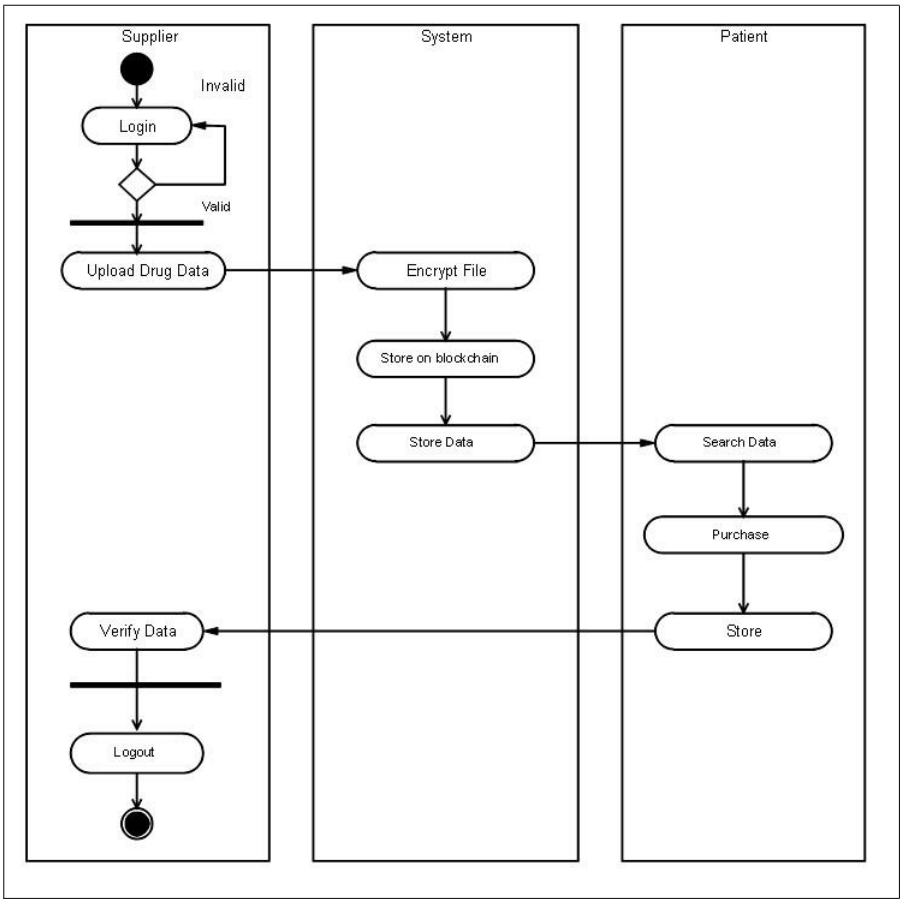


Figure 5.4: Activity Diagram

5.4 Implementation Steps

5.4.1 Message-Digest Algorithm

The MD5 message-digest algorithm is a widely used cryptographic hash function producing a 128-bit (16-byte) hash value, typically expressed in text format as a 32 digit hexadecimal number. MD5 has been utilized in a wide variety of cryptographic applications, and is also commonly used to verify data integrity.
Steps:

1. A message digest algorithm is a hash function that takes a bit sequence of any length and produces a bit sequence of a fixed small length.
2. The output of a message digest is considered as a digital signature of the input data.
3. MD5 is a message digest algorithm producing 128 bits of data.
4. It uses constants derived to trigonometric Sine function.
5. It loops through the original message in blocks of 512 bits, with 4 rounds of operations for each block, and 16 operations in each round.

6. Most modern programming languages provides MD5 algorithm as built-in functions

5.4.2 MD5 Implementation

In the Java web application for drug traceability, the MD5 algorithm can be used to create an encrypted hash value after drug transaction. Here is an outline of the implementation process.

Steps :

1. Obtain the drug information: Retrieve the necessary information about the newly created drug, such as its name, manufacturer, batch number, production date, and any other relevant details.

2. Concatenate the drug information: Concatenate all the drug information into a single string. For example, you can combine the drug name, manufacturer, batch number, and production date into a single string.

3. Convert the string to bytes: Convert the concatenated string into bytes using the appropriate character encoding, such as UTF-8.

4. Create an MD5 instance: Instantiate an MD5 algorithm implementation in Java, such as the java.security.MessageDigest class.

5. Update the MD5 instance with the byte array: Pass the byte array obtained from the concatenated string to the MD5 instance using the update() method. This will feed the data into the MD5 algorithm for hashing.

6. Generate the hash value: Generate the hash value by invoking the digest() method on the MD5 instance. This will produce a byte array representing the hash value.

7. Convert the hash value to a hexadecimal string: Convert the byte array hash value to a hexadecimal string representation using a conversion method, such as javax.xml.bind.DatatypeConverter.printHexBinary().

8. Store the hash value: Store the generated hash value in a secure and tamper-proof location, such as a database or a blockchain.

5.4.3 Existing System and Need for System

The existing system in the healthcare supply chain for drug traceability is plagued with numerous challenges that undermine transparency, security, and efficiency. Manual record-keeping, fragmented information silos, and centralized databases hinder real-time visibility and data interoperability, leading to risks such as counterfeit drugs, compromised patient safety, and regulatory non-compliance. These limitations create opportunities for fraudulent activities, product diversion, and errors in the tracking process. Therefore, there is an urgent need for a robust and secure drug traceability system that can ensure the authenticity, quality, and provenance of pharmaceutical products throughout the supply chain.

By leveraging blockchain technology and utilizing the Java programming language, the proposed system aims to address these challenges and provide a transparent, tamperproof, and efficient solution for tracking drugs. The blockchain-based system will enhance data integrity, eliminate the need for intermediaries, and enable real-time visibility for stakeholders, ensuring the safe and reliable distribution of pharmaceutical products while mitigating risks associated with counterfeit drugs and regulatory non-compliance.

5.5 Limitation of Existing System

- Lack of transparency: The existing system suffers from a lack of transparency and real-time visibility, making it difficult to track and verify the authenticity, quality, and provenance of pharmaceutical products throughout the supply chain.

- Manual record-keeping: Reliance on manual record-keeping introduces the risk of errors, delays, and inconsistencies in tracking drug movements, which can hinder timely identification of supply chain issues and pose risks to patient safety.

- Fragmented information silos: Fragmented information repositories and data silos make it challenging to achieve seamless data interoperability and hinder effective collaboration among stakeholders in the healthcare supply chain.

- Centralized databases and intermediaries: The reliance on centralized databases and intermediaries introduces vulnerabilities, such as the risk of data breaches, unauthorized access, and compromised integrity and confidentiality of sensitive information.

- Lack of standardization: Inconsistent data formats, incompatible systems, and limited data sharing standards among different stakeholders hamper the efficient exchange of information and hinder comprehensive traceability insights.

- Limited scalability: The existing system may struggle to handle the increasing complexity and scale of the healthcare supply chain, leading to potential bottlenecks, inefficiencies, and delays in tracking drug movements.

Chapter 6

METHODOLOGIES OF PROBLEM SOLVING

6.1 Project Estimate

Use Waterfall model and associated streams derived from assignments 1, 2, 3, 4 and 5 (Annex A and B) for estimation.

6.1.1 Reconciled Estimates

- Development Effort: Based on the complexity of the project, the estimated development effort is approximately 800 person-hours. This includes tasks such as system design, blockchain implementation, smart contract development, user interface development, testing, and integration.

- Timeframe: Considering the development effort and resource availability, the estimated timeframe for completing the project is approximately 12 weeks. This timeframe includes development, testing, bug fixing, and documentation.

- Resource Allocation: The project requires a dedicated team of developers proficient in Java and blockchain technologies. The estimated resource allocation includes two experienced Java developers, one blockchain developer, and a quality assurance specialist.

- Infrastructure and Tools: The project requires a suitable development environment, including servers, databases, and blockchain frameworks. Additionally, development tools such as IDEs (Integrated Development Environments), testing frameworks, and version control systems will be utilized.

- Project Costs: The project costs are estimated based on resource allocation, infrastructure requirements, and any additional expenses related to software licenses, cloud services, or external consulting if required. The estimated project cost is 20000 .Rs

- Risks and Mitigation: Potential risks and challenges, such as technology complexities, interoperability issues, or regulatory compliance, should be identified and addressed during the development process. Mitigation strategies, such as thorough testing, continuous monitoring, and adherence to best practices, will be implemented to minimize these risks.

6.2 Risk Management Analysis W. R. T. Np HardAnalysis

6.2.1 Risk Management

Given a failure case viz. Q, Using cloud computing Technology and mobile computing and saving energy by offloading tasks to cloud, we derived an algorithm for this problem as follows:

i) What steps do we have to take to prove a problem Q is NP-Complete?

ii) Pick a known NP-Complete problem P.

iii) Reduce P to Q.

iv) Describe a transformation that maps instances of P to instances of Q, . "yes" for Q = "yes" for P.

V) Prove the transformation works.
vi) Prove it runs in polynomial time

This is considered with specifying equipment and software that will successful satisfy the user requirement the technical needs of the system may vary considerably but might include

i)The facility to produce outputs in a given time.

ii)Response time under certain conditions. iii) Ability to process a certain column of transaction at a particular speed. iv) Our goal is to divide some data D (e.g., the safe combination) into pieces D1, D2. . .,Dn in such a way that:

v) The Knowledge of any k or more Di pieces makes D easily computable. vi) We divide our secret into pieces by picking a random degree polynomial

Q(x)= a0 + a1X + a2 Xˆ2 + a3 Xˆ3 +. . . .+ a(k-1) Xˆ(k-1) The APK Parsing algorithm is NP type. Because we can get and verify the solution set. Hence the problem statement involving it are NP-Complete.

6.2.2 Risk Identification

For risks identification, a review of the scope document, requirements specifications, and schedule is done. Answers to the questionnaire revealed some risks. Each risk is categorized as per the categories. Please refer table for all the risks. You can be refereed the following risk identification questionnaire.

i) Have top software and customer managers formally committed to supporting the project?YES

ii) Are end-users enthusiastically committed to the project and the system/product to be built? YES

iii) Are requirements fully understood by the software engineering team and its customers? YES

iv) Have customers been involved fully in the definition of requirements? YES

v) Do end-users have realistic expectations? Partial

vi) Does the software engineering team have the right mix of skills? YES

vii) Are project requirements stable? YES

viii) Is the number of people on the project team adequate to do the job? YES

ix) Do all customer/user constituencies agree on the importance of the project and on the requirements for the system/product to be built? : YES

6.3 Project Schedule

6.3.1 Project Task Set

Major Tasks in the Project stages are:

i) Task 1: To develop the problem under consideration and justify feasibility using concepts of knowledge canvas and IDEA matrix.

ii) Task 2: Project problem statement feasibility assessment using NP-Hard, NP-Complete or satiability issues using modern algebra and/or relevant mathematical models.

iii) Task 3: Use of divide and conquer strategies to exploit distributed/parallel/concurrent processing of the above to identify objects, morphisms, overloading in functions (if any), and functional relations and any other dependencies (as per requirements).

iv) Task 4: Use of above to draw functional dependency graphs and relevant Software modelling methods, techniques including UML diagrams or other necessities using appropriate tools.

v) Task 5: Testing of project problem statement using generated test data (using mathematical models, GUI, Function testing principles, if any) selection and appropriate use of testing tools, testing of UML diagram's reliability.

6.3.2 Task Network

- Project Initiation: Define project objectives, scope, and deliverables. Identify key stakeholders and establish project team roles and responsibilities.

- Requirement Gathering: Conduct in-depth requirements gathering sessions with domain experts and stakeholders to understand the specific needs and challenges of drug traceability in the healthcare supply chain.

- Literature Review: Perform a comprehensive literature survey to study existing research and solutions related to blockchain-based drug traceability and identify best practices and potential challenges.

- System Design: Based on the gathered requirements, design the architecture, data models, and user interfaces for the blockchain-based drug traceability system. Define the necessary components and their interactions.

- Blockchain Development: Implement the blockchain infrastructure using Java-based blockchain frameworks such as Hyperledger Fabric or Ethereum. Develop smart contracts to handle drug tracking, authentication, and data integrity.
- User Interface Development: Design and develop user-friendly interfaces for different stakeholders to interact with the system, including healthcare providers, manufacturers, distributors, and regulatory authorities.

- Integration and Testing: Integrate the blockchain system with existing healthcare supply chain systems and perform rigorous testing to ensure system functionality, data accuracy, and security. Conduct unit tests, integration tests, and user acceptance tests.

- Deployment: Prepare the system for deployment by configuring the necessary infrastructure, setting up servers, databases, and blockchain nodes. Ensure compatibility and scalability of the system.

- Data Migration: If applicable, migrate existing data from the legacy system to the new blockchain-based system, ensuring data integrity and maintaining historical records.

- Training and Documentation: Provide training sessions to end-users on how to use the system effectively. Prepare comprehensive documentation, including user manuals, technical guides, and system architecture documentation.

Chapter 7

ANALYSIS MODEL

7.1 Analysis Model

7.1.1 Planning and Requirement Analysis

i) System Design:- The requirement specifications from the first phase are studied in this phase and the system design is prepared. System Design helps in specifying hardware and system requirements and also helps in defining overall system architecture. The software code to be written in the next stage is created now.

ii) Implementation:- With inputs from system design, the system is first developed in small programs called units, which are integrated in the next phase. Each unit is developed and tested for its functionality which is referred to as Unit Testing.

iii) Integration and Testing:- All the units developed in the implementation phase are integrated into a system after testing of each unit. The software designed, needs to go through constant software testing to find out if there are any flaw or errors. Testing is done so that the client does not face any problem during the installation of the software.
iv) Deployment of System:- Once the functional and non-functional testing is done, the product is deployed in the customer environment or released into the market.

v) Maintenance:- This step occurs after installation, and involves making modifications to the system or an individual component to alter attributes or improve performance. These modifications arise either due to change requests initiated by the customer, or defects uncovered during live use of the system. Client is provided with regular maintenance and support for the developed software. All these phases are cascaded to each other in which progress is seen as flowing steadily downwards (like a waterfall) through the phases. The next phase is started only after the defined set of goals are achieved for previous phase and it is signed off, so the name Waterfall Model.

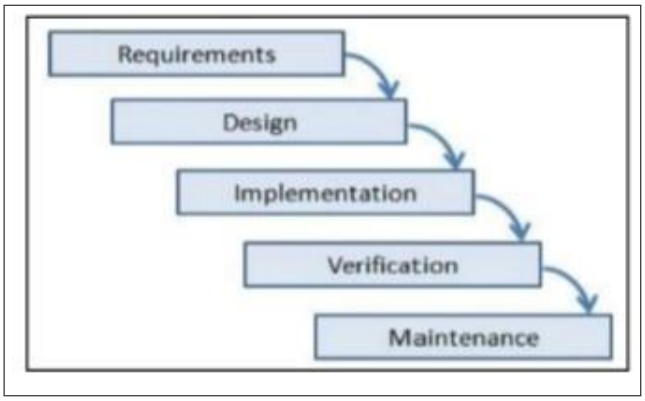


Figure 7.1: SDLC Process

7.2 System Implementation Plan

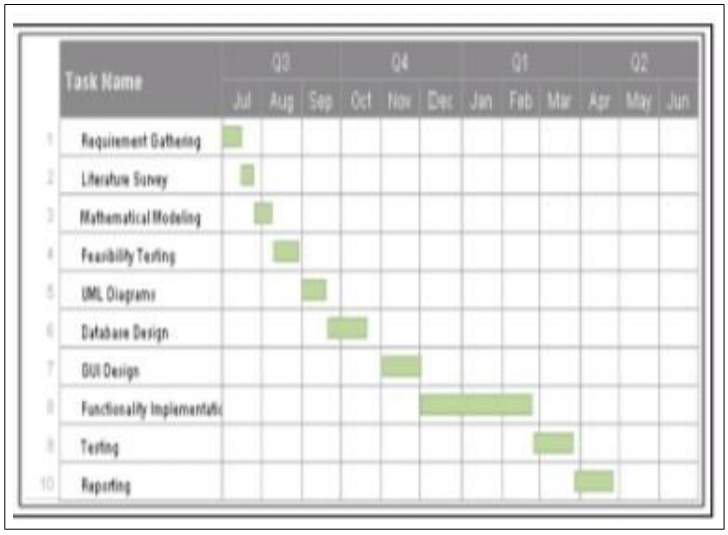


Figure 7.2: Plan of Project Execution

7.3 Coding and Development

7.3.1 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 the component level and test a specific business process, application, and/or system configuration. Unit tests ensure that each

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

7.3.2 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 satisfactory, as shown by successful 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.

7.3.3 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:

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

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

iii)Functions : identified functions must be exercised.

iv)Output : identified classes of application outputs must be exercised.

v)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.

7.3.4 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.

7.3.5 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.

7.3.6 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.

7.3.7 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.

7.3.8 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.

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

7.4 Test cases and Test Results

Testing of project problem statement using generated test data (using mathematical models, GUI, Function testing principles, if any) selection and appropriate use of testing tools, testing of UML diagram's reliability.

7.4.1 Module-ID:-01

Modules to be tested:-Registration

1. Enter the case insensitive Username click on Submit button.

Expected: It should display error.

2. Enter the case sensitive Username click on Submit button.

Expected: It should accept.

3. Enter the case insensitive Password click on Submit button.

Expected: It should display error.

4. Enter the case sensitive Password click on Submit button.

Expected: It should accept.

5. Enter the case insensitive Mobile Number click on Submit button.

Expected: It should display error.

6. Enter the case sensitive Mobile Number click on Submit button.

Expected: It should accept.

7. Enter the wrong address and click on Submit button.

Expected: It should display error.

8. Enter the correct address and click on Submit button.

Expected: It should accept.

Test Case_ID	Description	Test case I/P	Actual Result	Expected result	Test case criteria (P/F)
101	Enter the case insensitive Username click on Submit button.	Username	Error comes	Error Should come	P
102	Enter the case sensitive Username click on Submit button.	Username	Accept	Accept Username	P
201	Enter the case insensitive Password click on Submit button.	Password	Error comes	Error Should come	P
202	Enter the case sensitive Password click on Submit button	Password	Accept	Accept	P
301	Enter the case insensitive Mobile Number click on Submit button	Mobile Number	Error comes	Error Should come	P
302	Enter the case sensitive Mobile Number click on Submit button.	Mobile Number	Accept	Accept	P

Table 7.1: Test Cases

Test Case_ID	Description	Test case I/P	Actual Result	Expected result	Test case criteria (P/F)
001	Enter the correct username and wrong password click on Login button.	Username Password	Error comes	Error Should come	P
002	Enter the wrong username and correct password click on Login button,	Username Password	Error comes	Error Should come	P
003	Enter the correct username and password and click on Login button.	Username Password	Accept	Accept	P

Table 7.2: Test Cases

7.4.2 Module-ID:-2

Modules to be tested:- Login

1. Enter the correct username and wrong password click on Submit button.

Expected: It should display error.

2. Enter the wrong username and correct password and click on Submit button.

Expected: It should display error.

3. Enter the correct username and password and click on Login button.

Expected: It should display welcome page.

4. After login with valid credentials click on back button.

Expected: The page should be expired.

5. After login with valid credentials copy the URL and paste in another browser.

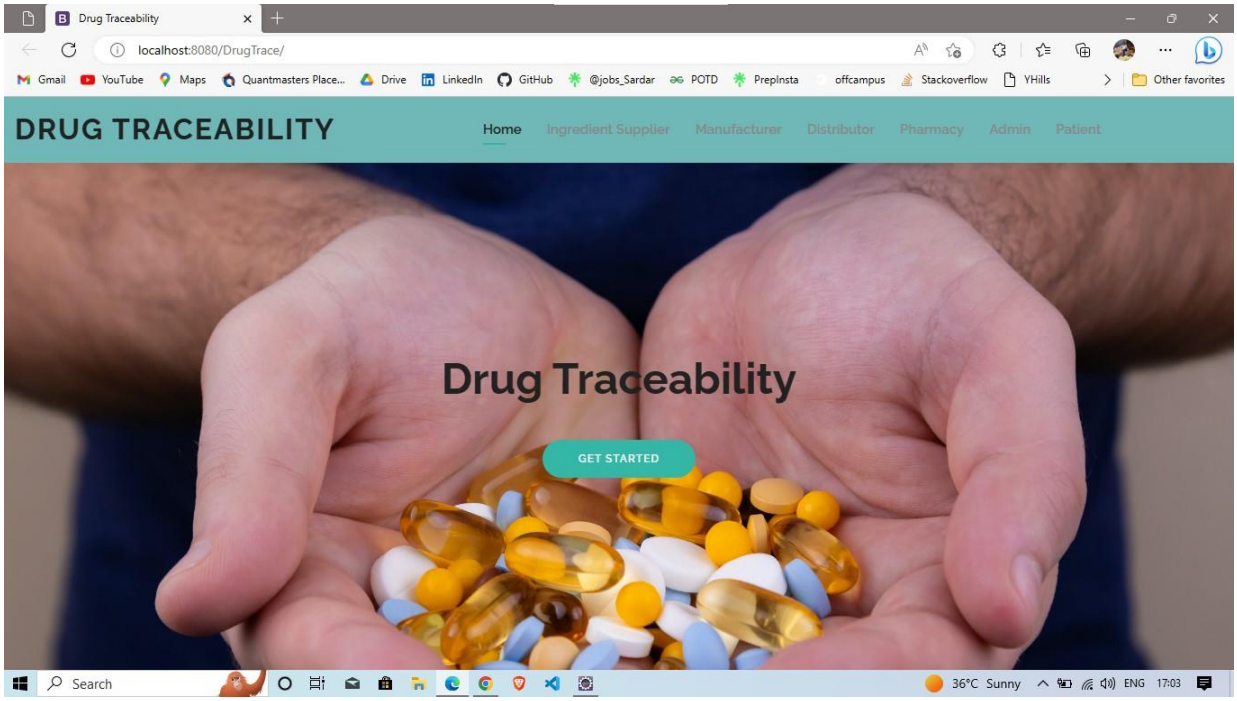
Expected: It should not display the user's welcome page.

6. Check the password with Lower case and upper case.

Expected: Password should be case sensitive.

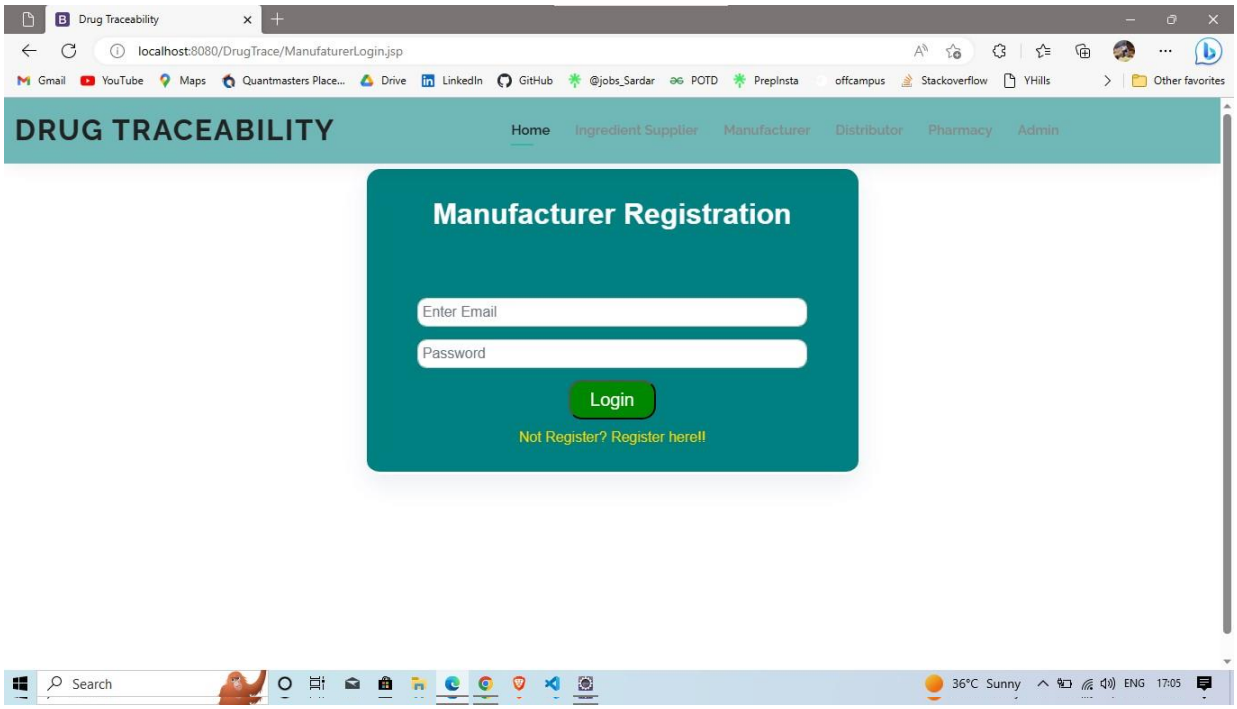
7.5 Outputs and Results

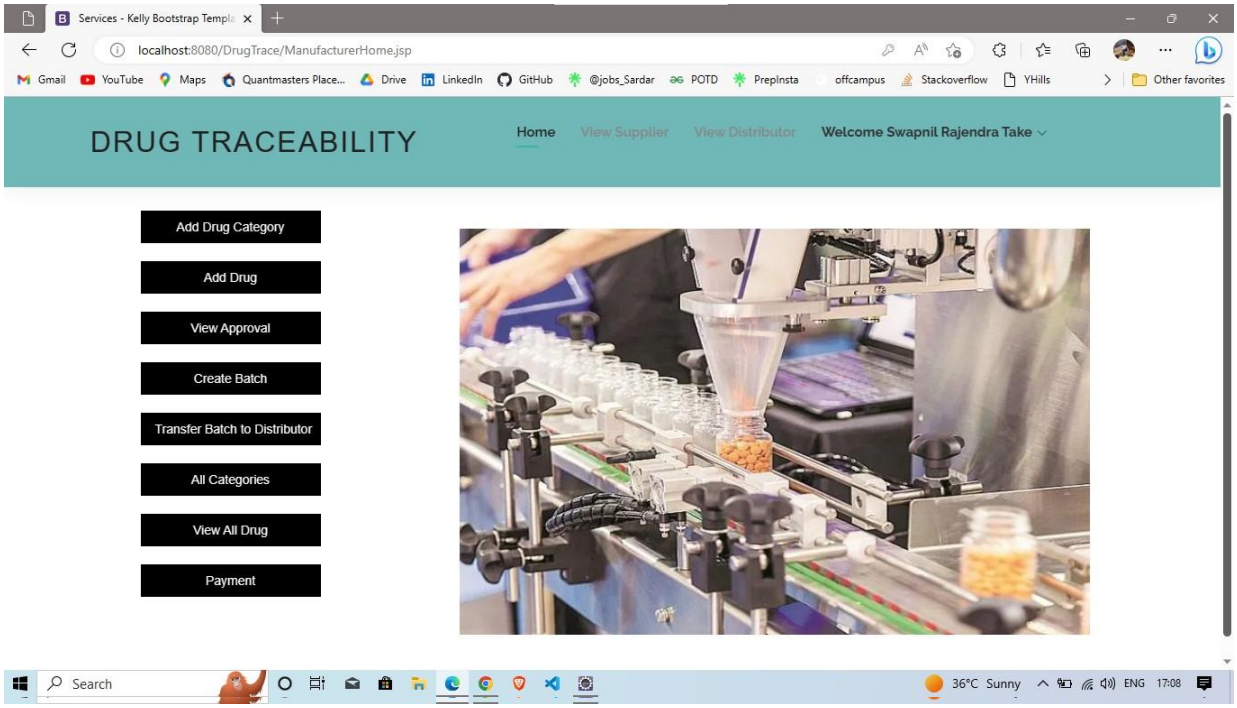
7.5.1 Home Module



7.5.2

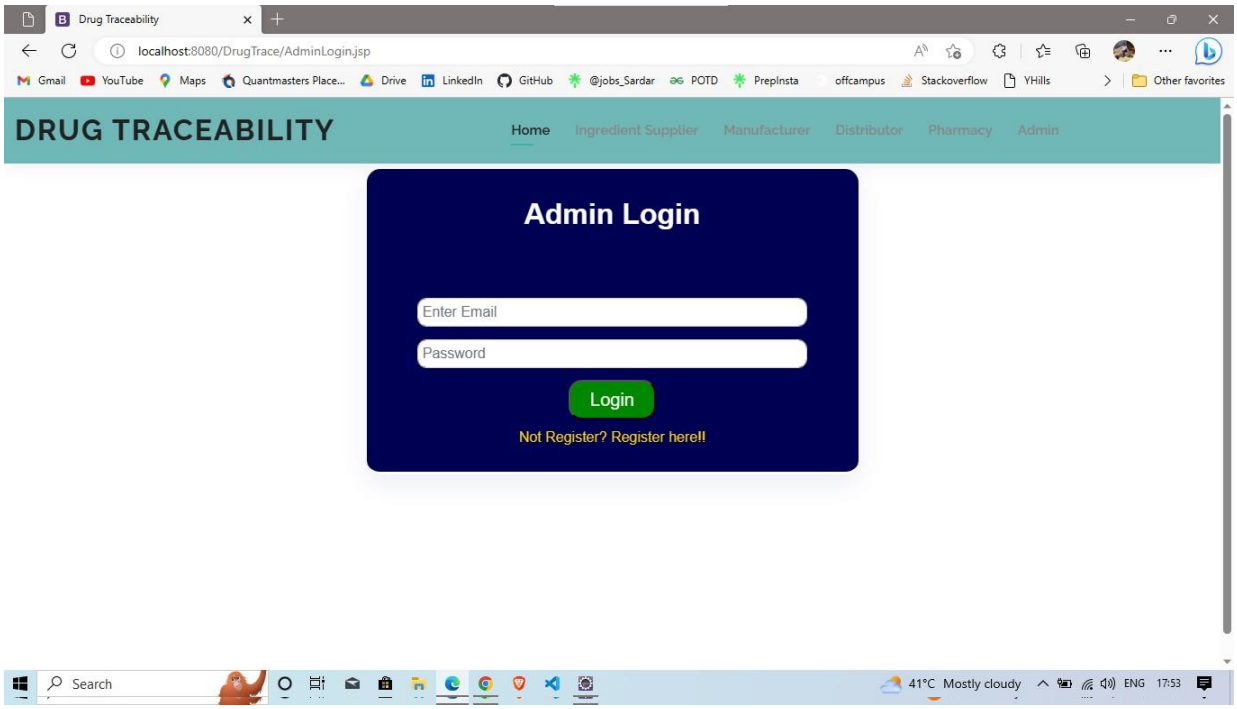
Supplier Module Manufacturer Module Distributor Module

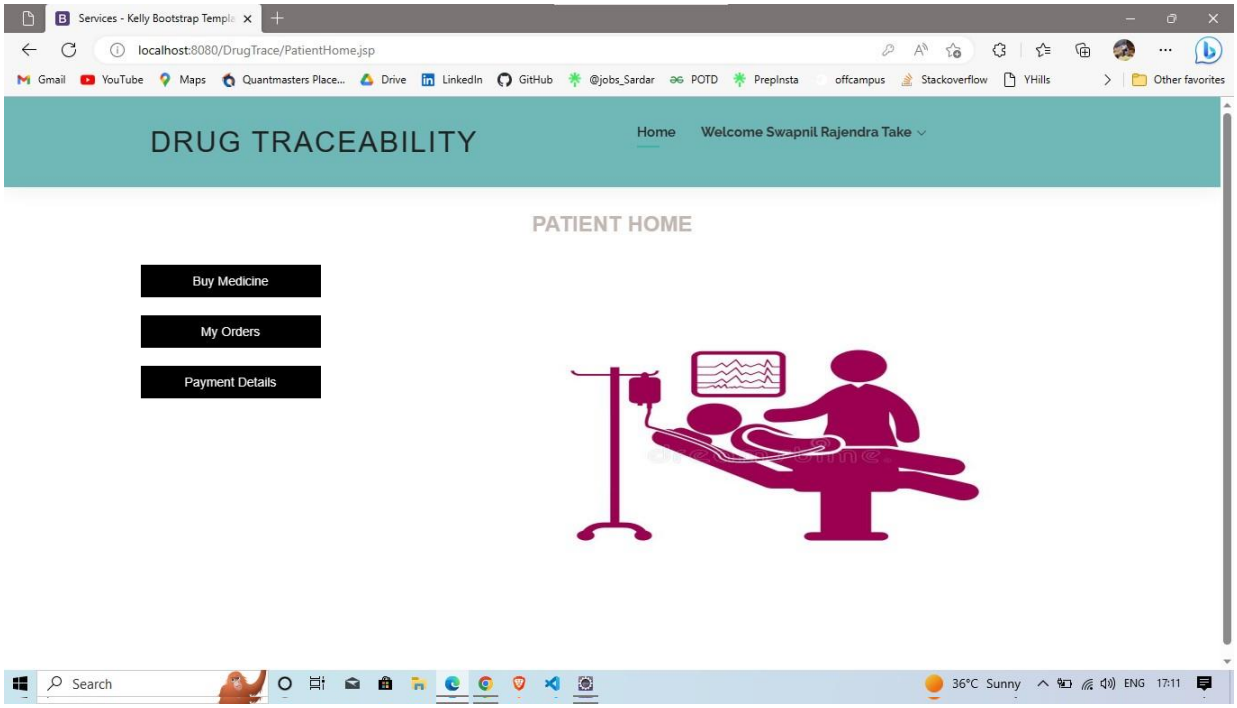




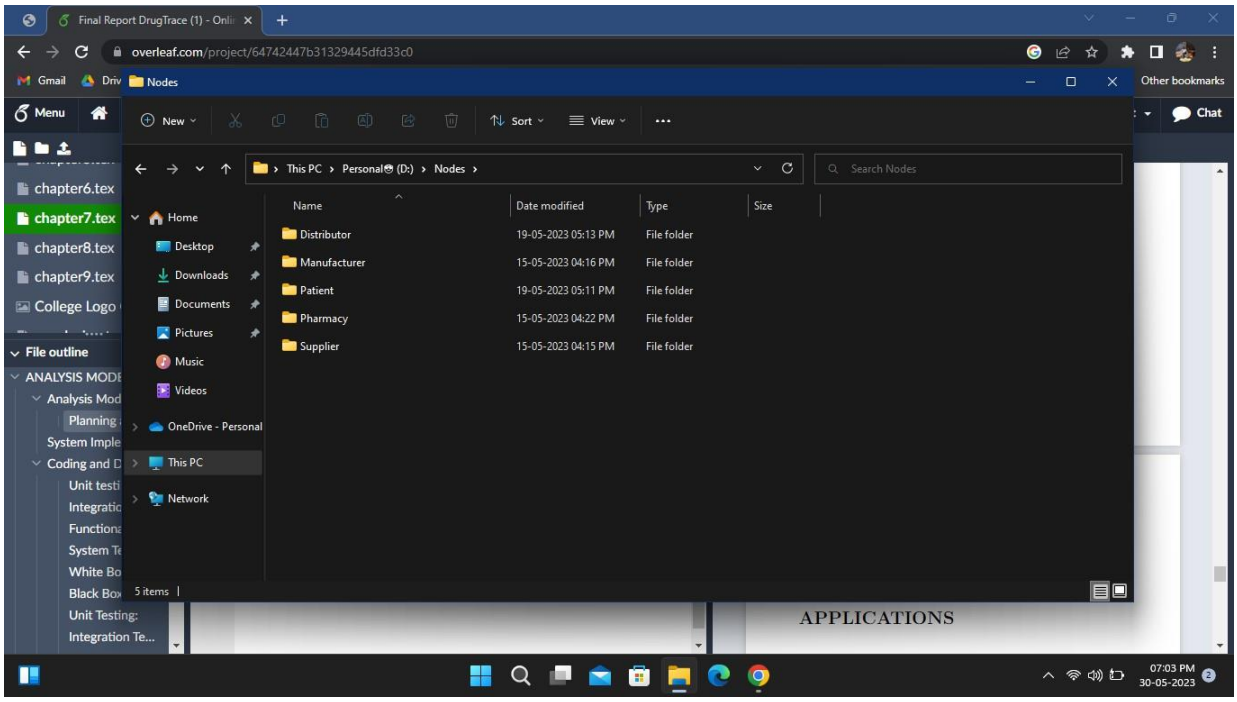
7.5.3

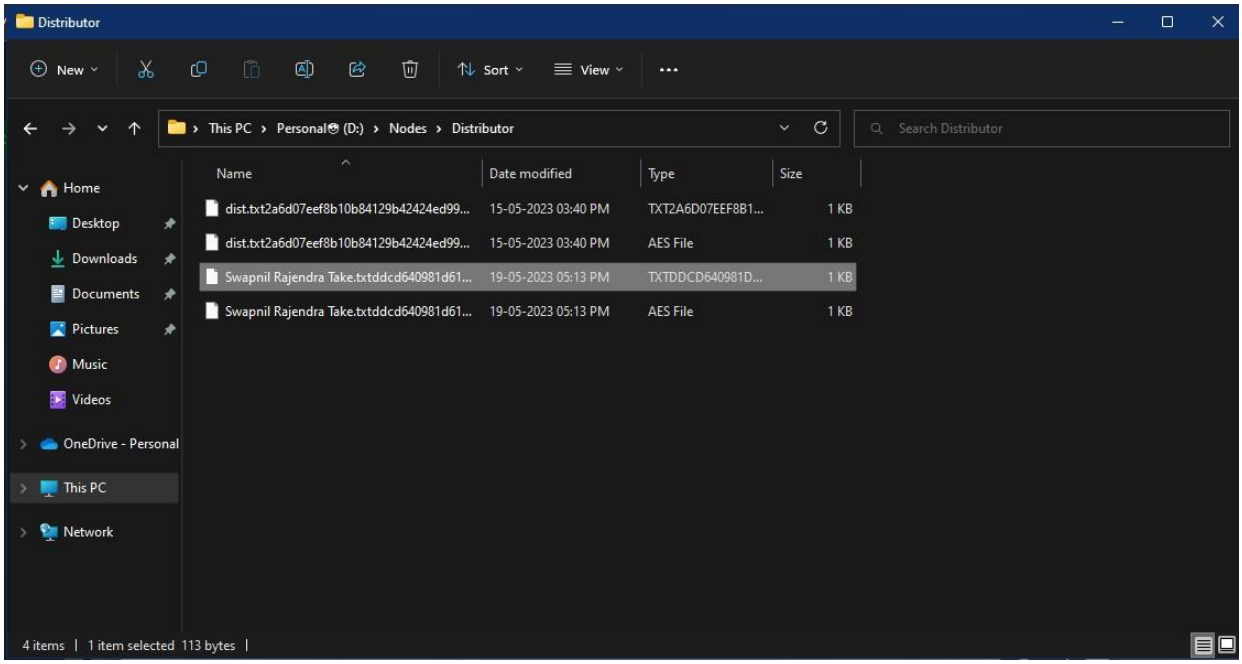
Pharmacy Module Admin Module Patients Module Generated





7.5.4 Values


Final Report DrugTrace (1) - Onli
overleaf.com/project/64742447b31329445dfd33c0
Nodes
New
Sort
View
This PC › Personal (D:) › Nodes ›
Search Nodes
Name
Date modified
Type
Size
Distributor 19-05-2023 05:13 PM File folder
Manufacturer 15-05-2023 04:16 PM File folder
Patient 19-05-2023 05:11 PM File folder
Pharmacy 15-05-2023 04:22 PM File folder
Supplier 15-05-2023 04:15 PM File folder
Home
Desktop
Downloads
Documents
Pictures
Music
Videos
OneDrive - Personal
This PC
Network
5 items
chapter6.tex
chapter7.tex
chapter8.tex
chapter9.tex
College Logo
File outline
APPLICATIONS
07:03 PM
30-05-2023


Distributor
New
Sort
View
This PC › Personal (D:) › Nodes › Distributor
Search Distributor
Name
Date modified
Type
Size
dist.txt2a6d07eef8b10b84129b42424ed99... 15-05-2023 03:40 PM TXT2A6D07EEF8B1... 1 KB
dist.txt2a6d07eef8b10b84129b42424ed99... 15-05-2023 03:40 PM AES File 1 KB
Swapnil Rajendra Take.txtddcd640981d61... 19-05-2023 05:13 PM TXTDDCD640981D... 1 KB
Swapnil Rajendra Take.txtddcd640981d61... 19-05-2023 05:13 PM AES File 1 KB
Home
Desktop
Downloads
Documents
Pictures
Music
Videos
OneDrive - Personal
This PC
Network
4 items | 1 item selected 113 bytes

Chapter 8

ADVANTAGES

8.1 Advantages

i) We propose a blockchain-based solution for the pharmaceutical supply chain that provides security, traceability, immutability, and accessibility of data provenance for pharmaceutical drugs.

ii) We design a smart contract capable of handling various transactions among pharmaceutical supply chain stakeholders.

iii) We present, implement and test the smart contract that denes the working principles of our proposed solution.

iv) We conduct security and cost analysis to evaluate the performance of the proposed blockchain-based solution.

Chapter 9

APPLICATIONS

9.1 Applications

- Counterfeit Drug Detection: The system can help in detecting counterfeit drugs by ensuring the authenticity and provenance of pharmaceutical products. The immutable nature of blockchain records and secure tracking mechanisms enable stakeholders to verify the origin and quality of drugs, reducing the risk of counterfeit products entering the supply chain.

- Supply Chain Transparency: The project enables real-time visibility into the healthcare supply chain, allowing stakeholders to track the movement of drugs from manufacturers to distributors, pharmacies, and ultimately to patients. This transparency improves accountability, reduces the likelihood of drug diversion, and enhances overall supply chain efficiency.

- Regulatory Compliance: The system assists in ensuring compliance with regulatory requirements and standards in the pharmaceutical industry. By providing an auditable and transparent record of drug transactions, the project helps stakeholders demonstrate adherence to regulatory guidelines, thereby mitigating the risk of non-compliance and potential penalties.

- Enhanced Patient Safety: With improved traceability, the system contributes to enhanced patient safety by reducing the risks associated with counterfeit or substandard drugs. Patients can have confidence in the authenticity and quality of the medications they receive, leading to better health outcomes

Chapter 10

10.1 Strengths and Weaknesses

The blockchain-based drug traceability system developed using Java has several strengths. Firstly, it provides a decentralized and tamper-proof ledger, ensuring the integrity and transparency of drug supply chain data.

This enhances trust and accountability among stakeholders, including pharmaceutical companies, regulatory authorities, and consumers. Secondly, the system offers enhanced security and privacy features, protecting sensitive information while still allowing authorized access.

Thirdly, the system leverages the scalability and robustness of blockchain technology, ensuring the efficient management of a large volume of drug traceability data.

However, there are also certain weaknesses of the system that need to be addressed. One major concern is the scalability of blockchain networks, as the system's performance may degrade as the number of transactions and participants increases. This can lead to slower transaction times and higher costs. Additionally, the energy consumption associated with blockchain technology is a significant consideration, as it requires substantial computational power and contributes to environmental impact. Furthermore, the complexity of blockchain implementation and the learning curve associated with understanding its principles and functionalities may pose challenges for adoption and integrate system.

10.2 Limitations of System

The real-world deployment of the blockchain-based drug traceability system faces several limitations and constraints. Firstly, regulatory and legal frameworks may not be fully supportive or well-defined, making it challenging to ensure compliance and acceptance of the system across different jurisdictions. Additionally, interoperability with existing systems and technologies used in the pharmaceutical industry can be a constraint, as integration efforts and data exchange standards need to be established. The system's reliance on internet connectivity and infrastructure is also a potential limitation, particularly in regions with limited technological infrastructure or unstable network conditions.

Furthermore, the system's implementation cost and resource requirements can be a significant constraint, especially for smaller pharmaceutical companies or regions with limited financial resources. Moreover, the participation and cooperation of all stakeholders in the drug supply chain, including manufacturers, distributors, pharmacists, and regulatory bodies, are crucial for the system's effectiveness. Resistance to change and lack of collaboration among stakeholders can impede the successful deployment and adoption of the system.

10.3 Potential Future Enhancements

There are several areas for potential future enhancements and extensions to the blockchain-based drug traceability system. Firstly, integrating Internet of Things (IoT) devices and sensors can enhance real-time tracking and monitoring of drug shipments, providing more accurate and comprehensive data. This can further improve supply chain transparency and efficiency. Additionally, incorporating smart contracts into the system can automate contract execution and enforcement, streamlining processes and reducing administrative burdens.

10.4 Ethical Considerations In Drug Traceability

The use of blockchain technology in drug traceability raises important ethical considerations. One such consideration is the privacy and security of patients' personal information contained within the blockchain. Safeguards must be in place to ensure that sensitive data is handled appropriately, with access restricted to authorized parties and in compliance with data protection regulations. Additionally, the system must consider the ethical implications of data ownership and control, ensuring that patients have control over their own data and can provide informed consent for its use

Chapter 11

CONCLUSION

In this system, we have investigated the challenge of drug traceability within pharmaceutical supply chains highlighting its significance especially to protect against counterfeit drugs. We have developed and evaluated a blockchain-based solution for the pharmaceutical supply chain to track and trace drugs in a decentralized manner. Specifically, our proposed solution leverages cryptographic fundamentals underlying blockchain technology to achieve tamper-proof logs of events within the supply chain and utilizes smart contracts within the Aws blockchain to achieve automated recording of events that are accessible to all participating stakeholders.

Chapter 12

PUBLISHED RESEARCH PAPER

WITH CERTIFICATE

12.1 Research Papers

12.1.1 IJIRSET

Drug Traceability in Healthcare Supply Chain Using Blockchain: A Review

International Journal of Innovative Research in Science, Engineering and Technology (IJIRSET) · Dec 12, 2022

(http://www.ijirset.com/upload/2022/december/30*DrugNC.pdf*)

12.1.2 IJRPR

Drug Traceability in Healthcare Supply Chain system using Blockchain.

International Journal of Research Publication and Reviews (IJRPR). May 21, 2023 (https://ijrpr.com/uploads/V4ISSUE5/IJRPR13289.pdf)

References

1. F. Jamil, L. Hang, K. Kim, and D. Kim, “A novel medical blockchain model for drug supply chain integrity management in a smart hospital,” Electronics, vol. 8 , p. 505, Apr. 2019, doi: 10.3390 /electronics 8050505.

2. Y. Huang, J.Wu, and C. Long, “Drugledger:Apractical blockchain system for drug traceability and regulation,” in Proc. IEEE Conf. Internet Things, Jul./Aug. 2018 , pp. 1137 1144.

3. S. Delgado-Segura, C. P´erez-Sol`a, G. Navarro-Arribas, and J. HerreraJoancomart´ı, “Analysis of the bitcoin UTXO set,” in Financial Cryptography and Data Security (Lecture Notes in Computer Science), vol. 10958, A. Zohar, Ed. Berlin, Germany: Springer, 2019, pp. 78 91.

4. K. M. Khan, J. Arshad, and M. M. Khan, “Investigating performance constraints for blockchain based secure e-voting system,” Future Gener. Comput. Syst., vol.

   105, pp. 13 26, Apr. 2020.

5. D. Vujicic, D. Jagodic, and S. Randic, “Blockchain technology, bitcoin, and Ethereum:

   A brief overview,” in Proc. 17th Int. Symp. Infoteh- Jahorina (INFOTEH), East Sarajevo, Srpska, Mar. 2018, pp. 1 6, doi: 10.1109 /INFOTEH. 2018.8345547.

6. . Jiang, J. Cao, H. Wu, Y. Yang, M. Ma, and J. He, “BlocHIE: A blockchain- based platform for healthcare information exchange,” in Proc. IEEE Int. Conf. Smart Comput. (SMARTCOMP), Jun. 2018, pp. 49–56.

7. W. J. Gordon and C. Catalini, “Blockchain technology for healthcare: Facilitating the transition to patient-driven interoperability,” Comput. Struct. Biotechnol J., vol. 16, pp. 224–230, 2018.

8. Anjia yang, Wei Lu, Yue Zhang, Lin Hou” CrowdBC: A Blockchain-based Decentralized a Framework for Crowdsourcing” IEEE , vol. 7, May.2018.

9. S. Wang, Y. Zhang, and Y. Zhang, “A blockchain-based framework for data sharing with fine-grained access control in decentralized storage systems,” IEEE Access, vol.

   6, pp. 38437–38450, Jun.2018.

10. M. Steichen, R. Norvill, B. F. Pontiveros, and W. Shbair,“Blockchainbased, decentralized access control for IPFS,” in Proc. IEEE Blockchain, Jul. 2018, pp.

1499–1506.