Vaccine Tracking - Transparent

1. INTRODUCTION

The "Vaccine Tracking - Transparent using Blockchain" project is a groundbreaking initiative in healthcare that addresses the critical challenges of vaccine management and tracking. In response to widespread data inaccuracies, lack of transparency, and counterfeit vaccines, the project leverages blockchain technology to create a secure, transparent, and efficient system for tracking and administering vaccines.

Using a strategic combination of Node.js, React, and Ethereum's Solidity, the project creates an immutable ledger and automated smart contract ecosystem that ensures the integrity and security of vaccine-related data. It also provides an intuitive interface for healthcare professionals, patients, and supply chain stakeholders to seamlessly interact with the blockchain network.

The project's central goal is to instill renewed trust and transparency in the vaccine supply chain, bolstering patient safety and streamlining regulatory compliance. Through its comprehensive approach to system design, development, testing, deployment, and continuous maintenance, the "Vaccine Tracking - Transparent using Blockchain" project strives to revolutionize healthcare and set a new standard for secure and transparent vaccine management

Project Overview

The "Vaccine Tracking - Transparent using Blockchain" project is a technology-driven endeavor aimed at redefining the management of vaccines in the healthcare ecosystem. This innovative system combines Node.js, React, and Ethereum's Solidity to establish a blockchain-based ledger and automated smart contracts for secure and transparent vaccine tracking. Through a user-friendly React-based interface, healthcare professionals, patients, and supply chain managers can interact with the blockchain, ensuring immutable and tamper-proof vaccine data. The project spans phases like system design, development, testing, deployment, user training, and continuous maintenance. The expected outcomes include heightened transparency and traceability in the vaccine supply chain, bolstered patient safety, and reduced counterfeit vaccine risks. This holistic solution has the potential to streamline vaccine tracking, building trust in healthcare and enhancing patient safety.

Purpose

The purpose of the "Vaccine Tracking - Transparent using Blockchain" project is to develop and implement a robust blockchain-based system that improves the transparency, security, and traceability of vaccines within the healthcare supply chain. This system is designed to address critical issues such as inaccuracies in vaccine data, patient safety concerns, and the risk of counterfeit products. By leveraging technologies like Node.js, React, and Ethereum's Solidity, the project aims to create an immutable ledger and smart contract ecosystem that benefits healthcare professionals, patients, and supply chain stakeholders. Ultimately, the project's purpose is to enhance public health by ensuring the authenticity and reliability of vaccines, while also simplifying vaccine tracking and administration processes.

2. EXISTING PROBLEM

Existing Problem

In the healthcare industry, a pressing issue is the reliance on outdated manual record-keeping and data management systems, which gives rise to a range of critical challenges. These challenges encompass:

1. Data Inaccuracies: The manual nature of record-keeping results in human errors, introducing inaccuracies into vaccine administration records. These inaccuracies pose a significant risk to patient safety and public health.

2. Lack of Transparency: Both patients and healthcare professionals often grapple with a lack of transparency into the vaccine supply chain. This deficiency in transparency gives rise to concerns about the authenticity and safety of vaccines.

3. Counterfeit Vaccines: The current system exhibits vulnerabilities that permit counterfeit vaccines to infiltrate the supply chain. This not only endangers public health but also undermines the reputation of pharmaceutical companies.

4. Time-Consuming Processes: Healthcare professionals expend substantial amounts of time on data entry and record-keeping, diverting their attention from patient care and other mission-critical tasks.

5. Regulatory Compliance Challenges: Public health regulators encounter significant challenges in their quest to ensure compliance and data accuracy, primarily due to the potential for data manipulation.

The existing problem underscores the immediate necessity for a modern, secure, and transparent solution capable of addressing these challenges. Such a solution is poised to enhance the safety and efficiency of vaccine tracking and administration. The "Vaccine Tracking - Transparent using Blockchain" project

endeavors to confront these issues by harnessing the potential of blockchain technology, offering an innovative response to these pressing concerns.

Problem Statement:

In the healthcare sector, specifically regarding vaccine tracking and administration, the existing manual record-keeping system is plagued by inaccuracies, lack of transparency, and susceptibility to counterfeit vaccines. These challenges compromise patient safety and public health. Healthcare professionals spend excessive time on data entry, and regulatory compliance is hindered by potential data manipulation. Therefore, there is an immediate need for a comprehensive, secure, and transparent solution to modernize vaccine tracking and administration, ensuring data accuracy, enhancing patient safety, and boosting regulatory compliance. The "Vaccine Tracking - Transparent using Blockchain" project aims to address this pressing problem by implementing a blockchain-based system leveraging Node.js, React, and Ethereum's Solidity for smart contracts.

3. IDEATION AND PROPOSED SOLUTION

Empathy Map Canvas

Says
What have we heard them say? What can we magine them saying?

I'm worried about getting lost in the maze of vaccine records.

I'm concerned about the security of my vaccine data

I want to be able to track my vaccine history easily and securely.

I want to be able to share my vaccine record with my doctor or other authorized personnel.

Thinks
What are their wants, needs, hopes, and dreams? What other thoughts might influence their behavior?

How can I be sure that my vaccine records are accurate and up-to-date?

How can I protect my vaccine data from being stolen or misused?

How can I easily access my vaccine records when I need them?

How can I be sure that the vaccine records I see are authentic?

Does
What behavior have we observed? What can we imagine them doing?

Searches online for information about vaccine tracking using blockchain

Reads articles and blog posts about the topic

Talks to friends and family about the topic

Shares information about the topic on social media

Feels
What are their fears, frustrations, and anxieties? What other feelings might influence their behavior?

Frustrated with the current system of vaccine tracking

Worried about the security of their vaccine data

Empowered by the potential of blockchain to revolutionize vaccine tracking

Hopeful that blockchain can help to improve public health

Ideation and Brainstorming

Define your problem statement

What problem are you trying to solve? Frame your problem as a How Might We statement. This will be the focus of your brainstorm.

5 minutes

PROBLEM

"The current manual vaccine record-keeping system poses patient safety risks due to inaccuracies, necessitating the implementation of a blockchain-based solution for real-time, secure tracking."

In the healthcare sector, the current vaccine tracking and administration system relies heavily on manual record-keeping and paper-based processes, resulting in significant challenges. These challenges include data inaccuracies, limited transparency in the vaccine supply chain, potential risks of counterfeit vaccines, and inefficiencies in data management. As a result, healthcare professionals spend excessive time on administrative tasks, patient safety is compromised, and regulatory compliance is hindered by the potential for data manipulation. To address these critical issues and enhance vaccine tracking and administration, there is an urgent need for a comprehensive, secure, and transparent blockchain-based solution. The "Vaccine Tracking - Transparent using Blockchain" project seeks to tackle these problems by leveraging Node.js, React, and Ethereum's Solidity for smart contracts to create a system that ensures data accuracy, enhances patient safety, and strengthens regulatory compliance in the vaccination process.

2

Brainstorm

Write down any ideas that come to mind that address your problem statement.

10 minutes

TIP

You can select a sticky note and hit the pencil [switch to sketch] icon to start drawing!

Fahima

Current system is inefficient, insecu re, and fragmented.

Blockchain can address all these challenges.

Blockchain-based system can improve efficiency, enhanc e security, and facilitate data sharing.

Kanaga

Blockchain can help improve public health outcomes

Blockchain can help reduce the risk of vaccine-preventable diseases.

Blockchain can help improve the transparency and accountability of the vaccine supply chain.

Suchismita

Blockchain can create a global vaccination database.

Blockchain can develop new vaccine delivery systems.

Blockchain can develop new vaccine financing mechanisms.

Johanna

Blockchain can create personalized vaccination schedules.

Blockchain can develop new vaccine reminders and alerts.

- Blockchain can develop new vaccine safety monitoring systems.

3

Group ideas

Take turns sharing your ideas while clustering similar or related notes as you go. Once all sticky notes have been grouped, give each cluster a sentence-like label. If a cluster is bigger than six sticky notes, try and see if you and break it up into smaller sub-groups.

20 minutes

TIP

Add customizable tags to sticky notes to make it easier to find, browse, organize, and categorize important ideas as themes within your mural.

Data Accuracy and Patient Safety

Transparency and Verification

Blockchain Solution and Data Security

4

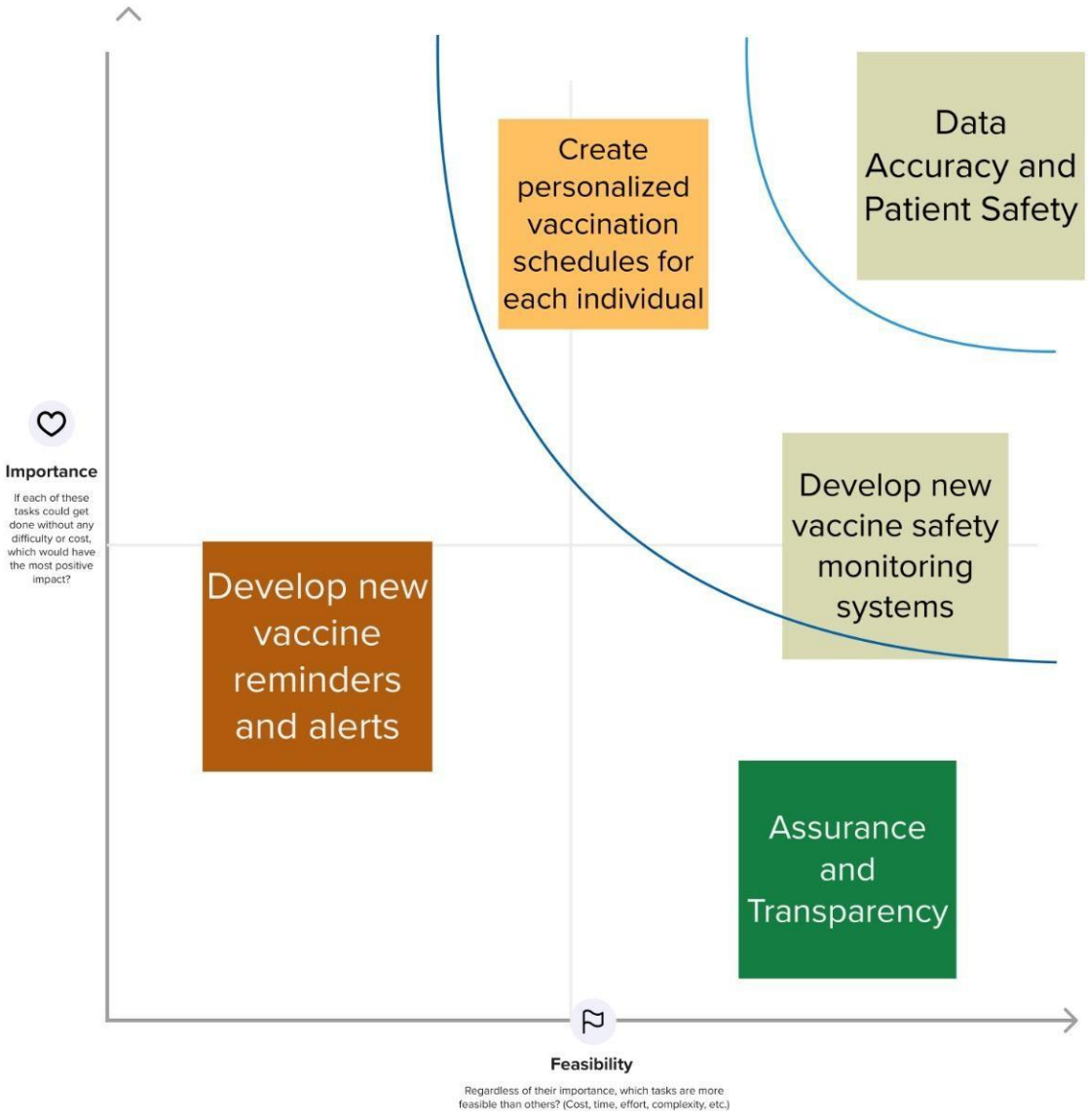
Prioritize

Your team should all be on the same page about what's important moving forward. Place your ideas on this grid to determine which ideas are important and which are feasible.

20 minutes

TIP

Participants can use their cursors to point at where sticky notes should go on the grid. The facilitator can confirm the spot by using the laser pointer holding the **H key** on the keyboard.



4 .REQUIREMENT ANALYSIS

Functional Requirements

User management is a critical functional requirement for the "Vaccine Tracking - Transparent using Blockchain" project. It involves a set of essential features and capabilities to effectively manage users interacting with the system. The primary functional requirements for user management include:

1. User Registration: - The system should allow users to create accounts by providing necessary information, including username, password, and, optionally, additional user-specific details.

2. User Authentication:- Users must be able to authenticate their identity through secure methods, such as passwords or multi-factor authentication, to access the system.

3. User Authorization:- The system should support role-based access control, ensuring that users are granted appropriate permissions based on their roles or responsibilities.

4. User Profile Management: - Users should have the capability to update and manage their personal information, preferences, and settings within the system.

5. User Access Control: - The system should monitor user activities, track login sessions, and enforce access control policies to ensure that users only access authorized features and data.

6. Password Reset and Recovery:- Users should have the option to reset their passwords or recover their accounts in case of forgotten credentials, while adhering to a secure process for identity verification.

7. User Deactivation and Removal:- The system must provide the means to deactivate or delete user accounts when they are no longer needed or upon user request. Deactivation should restrict access, while removal permanently deletes accounts.

8. User Reporting and Auditing- The system should maintain records of user activities and generate reports for monitoring user behavior, tracking system usage, and identifying anomalies or security breaches.

9. User Communication:- The system should facilitate communication with users, allowing notifications, emails, or messages for updates, alerts, or account-related information.

10. User Role Management:- Administrators should have the capability to define and manage user roles, assigning appropriate permissions and responsibilities to each role.

11. User Data Protection:- The system must adhere to data protection and privacy regulations, ensuring that user data is securely stored and processed.

12. User Authentication and Authorization Logging: - Log and maintain records of user authentication and authorization activities to support security audits and investigations.

These functional requirements are crucial for effective user management within the system, ensuring security, access control, user privacy, and a positive user experience. They contribute to the overall success of the "Vaccine Tracking - Transparent using Blockchain" project by providing a secure and user-friendly environment for all stakeholders.

Non-Functional Requirements

Non-functional requirements define the quality attributes and constraints that the "Vaccine Tracking - Transparent using Blockchain" project must adhere to in addition to the functional features. These non-functional requirements are essential to ensure the reliability, security, and performance of the system. The non-functional requirements for this project include:

1. Security:
   - Data Security: Ensure the confidentiality, integrity, and availability of sensitive user data and vaccine-related information through robust encryption and access controls.
   - Authentication and Authorization: Implement strong user authentication and authorization mechanisms to protect against unauthorized access.
   - Protection Against Cyber Threats: The system must be designed to defend against various cyber threats, including data breaches and malicious attacks.

2. Scalability: - The system should be scalable to accommodate a growing number of users, transactions, and vaccine data without compromising performance.

3. Performance: - The system should provide responsive and efficient user management functions to minimize latency and ensure a smooth user experience.

4. Reliability: - The system must be highly reliable, with minimal downtime, to ensure uninterrupted access for users.

5. Usability: - The user management interface should be intuitive and user-friendly, requiring minimal training for users to navigate and perform tasks.

6. Compliance: - Ensure compliance with relevant data protection regulations (e.g., GDPR or HIPAA) and healthcare industry standards for data privacy and security.

7. Interoperability: - The system should be designed to integrate seamlessly with other healthcare systems, such as electronic health records (EHRs) or pharmacy databases, to ensure the smooth exchange of information.

8. Auditability: - Maintain detailed audit logs for user management activities, ensuring transparency and accountability.

9. Data Backup and Recovery: - Implement regular data backup procedures and disaster recovery plans to prevent data loss and minimize downtime in the event of system failures.

10. Accessibility: - Ensure that the user management interface is accessible to individuals with disabilities, adhering to accessibility standards and guidelines.

11. System Maintenance: - Plan for regular system updates, maintenance, and patch management to address vulnerabilities and enhance system security.

12. Response Time:- The system should respond to user management requests promptly to provide a seamless experience for users and healthcare professionals.

13. Mobile Responsiveness: - Ensure that the user management interface is responsive and functional on a variety of devices, including mobile phones and tablets.

14. Data Retention Policies:- Establish policies for data retention, archiving, and deletion to comply with regulatory requirements and manage storage efficiently.

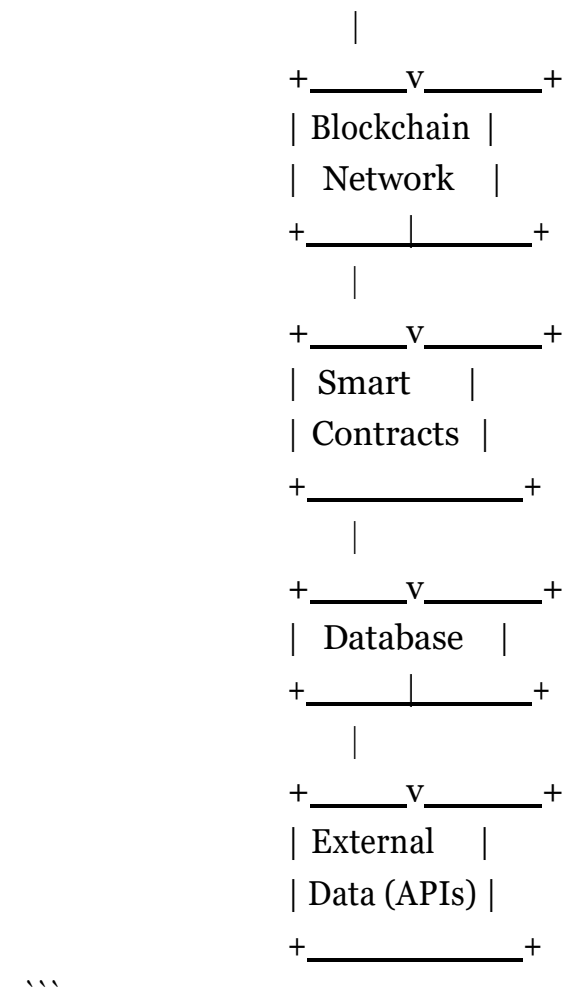
Adhering to these non-functional requirements is crucial to ensuring that the "Vaccine Tracking - Transparent using Blockchain" system is secure, efficient, and capable of meeting the demands of healthcare professionals, patients, and supply chain managers while maintaining data integrity and compliance with industry standards and regulations.

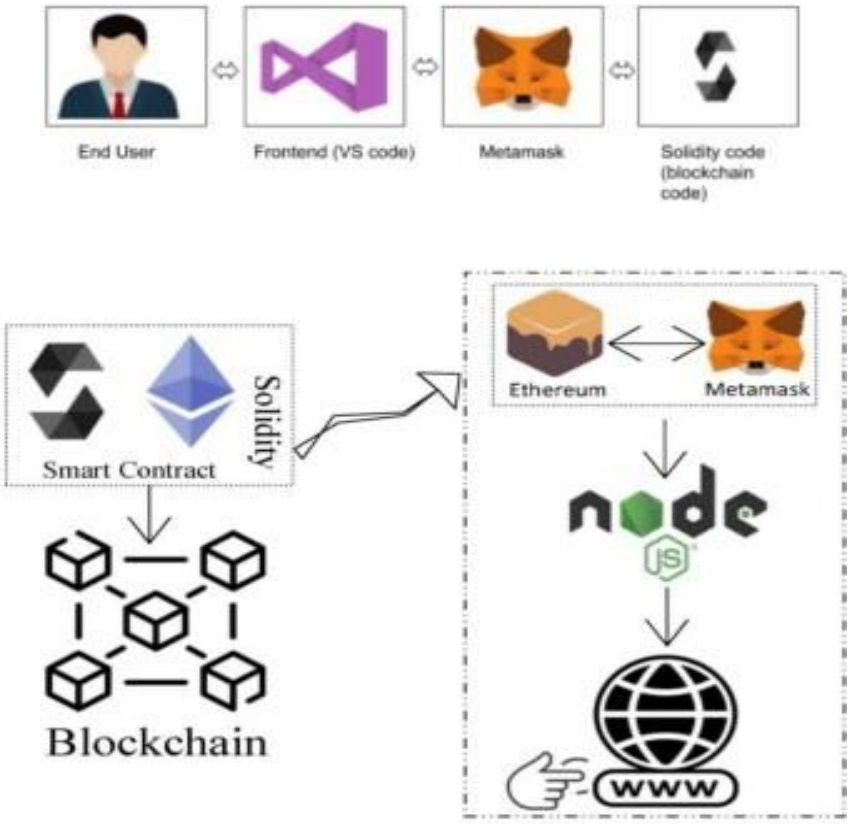
5. PROJECT DESIGN

Data Flow Diagram & User Scenarios:

To illustrate the data flow and user scenarios for the "Vaccine Tracking - Transparent using Blockchain" project, we will create a high-level data flow diagram (DFD) and outline user scenarios. Please note that this is a simplified representation, and in a real project, the DFD would be more

```
+_____________________+
|  User Roles    |
|  - Healthcare |
|  - Patients    |
|  - Supply      |
|    Chain       |
|  - Admins      |
+_____________________+
       |
+_____v_____+
|  Interface    |
|  (Web App)    |
+_____+|_____+
```


Blockchain
Network
Smart
Contracts
Database
External
Data (APIs)



Interaction between web and the Contract

User Scenarios:

1. Patient Registration and Verification:- Scenario: A patient accesses the web application, registers an account by providing personal information, and verifies their identity.
   - Data Flow: User Interface → Blockchain Network → Smart Contracts → Database
   - Actions: User registration data is processed, verified, and stored in the blockchain and database.

2. Vaccine Verification by Healthcare Professional:
   - Scenario: A healthcare professional logs in, scans a vaccine QR code to verify its authenticity, and updates the patient's vaccine record.

- Data Flow: User Interface → Blockchain Network → Smart Contracts → Database
- Actions: The healthcare professional's actions are recorded on the blockchain, and the vaccine record is updated.

3. Supply Chain Monitoring:- Scenario: A supply chain manager logs in to monitor vaccine shipments, reviews smart contract data, and generates reports on vaccine logistics.
- Data Flow: User Interface → Blockchain Network → Smart Contracts → Database
- Actions: The supply chain manager can access and analyze real-time data about vaccine shipments and transactions.

4. User Profile Management:
- Scenario: A user (e.g., patient or healthcare professional) logs in to update their profile information and preferences.
- Data Flow: User Interface → Blockchain Network → Smart Contracts → Database
- Actions: User profile updates are recorded on the blockchain and saved in the database.

5. Administrator Role Management:
- Scenario: An administrator logs in to manage user roles and permissions, assigning specific access rights.
- Data Flow: User Interface → Blockchain Network → Smart Contracts → Database
- Actions: The administrator updates user roles in smart contracts, affecting access and permissions.

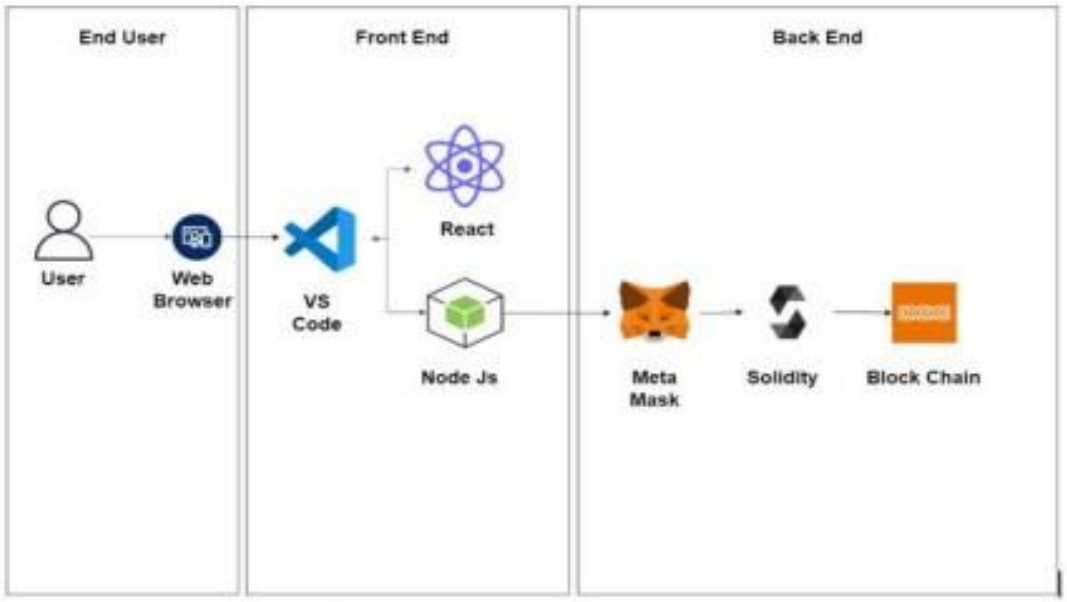
6. Audit Log Access:
- Scenario: An auditor (e.g., regulatory authority) accesses the system to review audit logs of user management and vaccine tracking activities.
- Data Flow: User Interface → Blockchain Network → Smart Contracts → Database

- Actions: The auditor can access and review the recorded audit logs for transparency and compliance purposes.

This set of user scenarios and the associated data flow diagram provides an overview of how users interact with the system, and how data is processed and recorded within the blockchain-based infrastructure. The project design would further detail the components and interactions, ensuring a secure and transparent vaccine tracking system.

6. PROJECT PLANNING & SCHEDULING

Technical Architecture



GENERAL ARCHITECTURE

Sprint Planning and Estimation

Sprint planning and estimation are key components of the Agile software development methodology, particularly in frameworks like Scrum. They involve

breaking down project work into smaller, manageable tasks and determining how much work can be completed in a fixed time frame, known as a sprint. Here's an overview of the process:

1. Product Backlog:- Start with a product backlog, which is a prioritized list of all the features, user stories, and tasks that need to be completed.

2. Sprint Planning Meeting:
   - Sprint planning is typically a time-boxed meeting held at the beginning of each sprint (usually 2-4 weeks). The team, including developers, product owners, and Scrum Master, participates.

3. Define Sprint Goal:
   - The product owner outlines the high-priority items from the product backlog and sets the sprint goal. This provides a clear objective for the sprint.

4. Select User Stories and Tasks: - Based on the sprint goal, the team collectively selects user stories and tasks from the product backlog that can be completed within the sprint.

5. Break Down User Stories:- For larger user stories, the team breaks them down into smaller, actionable tasks. This ensures that each task is well-defined and achievable within the sprint.

6. Estimation:- The team estimates the effort required to complete each task. This can be done using various techniques like story points, ideal days, or hours. The purpose is to provide a relative size of effort for each task.

7. Task Assignment: - The team assigns tasks to individual team members based on their skills and availability. Each team member takes ownership of specific tasks.

8. Capacity Planning: - The team considers its capacity for the sprint. This includes accounting for factors like holidays, team member availability, and non-development work (meetings, reviews, etc.).

9. Commitment: - Based on the team's capacity and the estimated effort for the selected tasks, the team commits to the amount of work they believe they can complete during the sprint.

10. Sprint Backlog: - The selected tasks and user stories, along with their estimates, form the sprint backlog. This is a subset of the product backlog and represents the work to be done in the sprint.

11. Daily Stand-Ups: - Throughout the sprint, daily stand-up meetings are held to ensure that team members are on track and to address any blockers or issues.

12. Review and Retrospective: - At the end of the sprint, a sprint review is conducted to showcase the completed work to stakeholders. This is followed by a retrospective to reflect on what went well and what could be improved in the next sprint.

13. Refinement:- The team also uses sprint planning meetings for backlog refinement, where they review and prioritize items in the product backlog for future sprints.

Sprint planning and estimation are iterative processes, and the team's ability to accurately estimate and commit improves over time. The goal is to maintain a steady pace of delivery and provide stakeholders with valuable increments of the product at the end of each sprint.

7. CODING AND SOLUTION

CODE:

```
/  SPDX-License-Identifier: MIT
pragma solidity ^0.8.0;

contract Vaccination {
   address public owner;

   constructor() {
      owner = msg.sender;
   }

   modifier onlyOwner() {
      require(msg.sender == owner, "Only the owner can perform this action");
      _;
   }

   struct Vaccine {
      string vaccineName;
      string manufacturer;
      uint256 manufacturingDate;
      string batchNumber;
      uint256 quantity;
      address customerAddress;
   }

   mapping(uint256 => Vaccine) public vaccines;
   uint256 public vaccineCount;
```

```
    event VaccineAdded(uint256 indexed vaccineId, string vaccineName, string
manufacturer, uint256 manufacturingDate, string batchNumber, address
customerAddress);

    function addVaccine(uint256 vaccineId, string memory _vaccineName, string
memory _manufacturer, uint256 _manufacturingDate, string memory
_batchNumber,uint256 _qty, address _customerAddress) external onlyOwner {


        vaccines[vaccineId]  =  Vaccine(_vaccineName,  _manufacturer,
_manufacturingDate, _batchNumber, _qty, _customerAddress);
        vaccineCount++;

        emit VaccineAdded(vaccineId, _vaccineName, _manufacturer,
_manufacturingDate, _batchNumber, _customerAddress);
    }

    function getVaccineDetails(uint256 _vaccineId) external view returns (string
memory, string memory, uint256, string memory,uint256, address) {

        Vaccine memory vaccine = vaccines[_vaccineId];
        return (vaccine.vaccineName, vaccine.manufacturer,
vaccine.manufacturingDate, vaccine.batchNumber, vaccine.quantity,
vaccine.customerAddress);
    }
}
```

SAMPLE OUTPUT:

VaccineAdded event emitted:
vaccineId: 1

vaccineName: "COVID-19 Vaccine"
manufacturer: "Pfizer"
manufacturingDate: 1665368000
batchNumber: "ABC123"
customerAddress: 0x1234567890123456789012345678901234567890

Calling getVaccineDetails(1) would return:
vaccineName: "COVID-19 Vaccine"
manufacturer: "Pfizer"
manufacturingDate: 1665368000
batchNumber: "ABC123"
quantity: 100
customerAddress: 0x1234567890123456789012345678901234567890

DEMO:
https:/ drive.google.com/file/d/1vIQ41jgyovLCU0x4v20q8QzXdQnb0qpP/view?usp=sharing

GITHUB LINK : https://github.com/Chandrukumarmech/Block-chain/upload