A Project Report on

A Comprehensive Blockchain Based Web Framework For Blood Banks

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

Bachelor of Engineering

in

Information Technology

by

Chinmay Dharap(18104062)
Prajwal Sonar(18104019)
Tejas Jadhav(18104030)

Under the Guidance of

Prof. Kiran Deshpande

Department of Information Technology
NBA Accredited
A.P. Shah Institute of Technology
G.B.Road,Kasarvadavli, Thane(W), Mumbai-400615
UNIVERSITY OF MUMBAI

Academic Year 2021-2022

Approval Sheet

This Project Report entitled ***"A Comprehensive Blockchain Based Web Framework For Blood Banks"*** Submitted by ***Chinmay Dharap (18104062), Prajwal Sonar (18104019), Tejas Jadhav (18104030)*** is approved for the partial fulfillment of the requirement for the award of the degree of ***Bachelor of Engineering*** in ***Information Technology*** from ***University of Mumbai***.

Prof. Kiran Deshpande
Guide

Prof. Kiran Deshpande
Head, Department of Information Technology

Place:A.P.Shah Institute of Technology, Thane
Date:

CERTIFICATE

This is to certify that the project entitled ***"A Comprehensive Blockchain Based Web Framework For Blood Banks"*** submitted by ***"Chinmay Dharap" (18104060), Prajwal Sonar (18104019), Tejas Jadhav (18104030)*** for the partial fulfillment of the requirement for award of a degree ***Bachelor of Engineering*** in ***Information Technology***, to the University of Mumbai is a bonafide work carried out during academic year 2021-2022.

(Prof. Kiran Deshpande)
Guide

Prof. Kiran Deshpande
Head, Department of Information Technology

Dr. Uttam D.Kolekar
Principal

External Examiner(s)

1.

2.

Place:A.P.Shah Institute of Technology, Thane
Date:

Acknowledgement

We have great pleasure in presenting the report on **A Comprehensive Blockchain Based Web Framework For Blood Banks** We take this opportunity to express our sincere thanks towards our guide **Prof. Kiran Deshpande** for providing the technical guidelines and suggestions regarding line of work. We would like to express our gratitude towards his constant encouragement, support and guidance throughout the development of project.

We thank **Prof. Vishal S. Badgujar** BE project co-ordinator, Department of IT, APSIT for being encouraging throughout the course and for guidance.

We also thank the entire faculties of department for their invaluable help rendered during the course of this work. We wish to express our deep gratitude towards all our colleagues of APSIT for their encouragement.

Chinmay Dharap(18104062)

Prajwal Sonar(18104019)

Tejas Jadhav(18104030)

Declaration

We declare that this written submission represents our ideas in our own words and where others' ideas or words have been included, We have adequately cited and referenced the original sources. We also declare that We have adhered to all principles of academic honesty and integrity and have not misrepresented or fabricated or falsified any idea/data/fact/source in our submission. We understand that any violation of the above will be cause for disciplinary action by the Institute and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been taken when needed.

Chinmay Dharap (18104062)
Prajwal Sonar (1810409)
Tejas Jadhav (18104030)

Date:

Abstract

Our aim is to improve the existing Blood Bank system through our project. We propose an innovative blood bank system based on blockchain technology using a distributed ledger structure and transaction execution process of the situation where the collected blood reaches the final stage - hospitals. Existing centralized blood management systems have various drawbacks including lack of information on blood bags, inability to reflect real-time updates in details and the trust factor. As the blockchain technologies offer the possibility to maintain a transparent blood bank system and since the data cannot be changed and tampered with, it help overcome these drawbacks. In addition, the system helps keep the donor updated regarding blood bag usage. This strategy aims to make the proposed system as transparent as possible. This strategy enables hospitals located distant from blood banks to obtain blood supplies in an emergency. Since blood has a certain storage period, it is expected that it will be possible to fulfill the demand of blood bags by supplying and using the bags that are nearing their expiration. Not only does it solve the primary purpose that is it helps to save lives, but it will also act as a waste management strategy for Blood by using near expiration bags first.

Contents

List of Figures

Chapter 1

Introduction

In today's modern era all the work is planned or implemented by using some kind of newest technology. The usage of the emerging technologies like blockchain allows us to take benefit of their distinctive feature. Blockchain is dispensed and duplicated ledger shared throughout all the nodes of a network. However, all the data becomes completely tamper-proof and available from any of the nodes. Each node has a copy of the same blockchain, providing traceability that is transparent and trusted. Existing blood bank management systems can use emerging technologies like blockchain to develop a reliable system that can assist humans to accomplish complicated tasks. Although, while using them, some loopholes still remain unnoticed. These loopholes can give rise to detrimental side effects. In India, currently, more than 90% of blood donations are carried out in camps set up by various organizations. While all the donated blood goes within the testing stage and safe blood is separated, sometimes unsafe blood can also infiltrate this test and reach the patients infecting them. Though this problem may go unnoticed, the consequences are fatal. Furthermore, there are several important challenges while setting up a blood repository. That mainly contains the identification of donors, authentication, and most importantly, separation of donors based on their past health records. This makes it extremely necessary to have a decentralized system for blood donors. In this project work, by the adoption of the Blockchain, we aim at delivering rather simplified yet quality-assured support for maintaining and monitoring the blood bag details collected from various sources and tracking the physical conditions of those bags as well. While the current traditional blood bank system is thwarted by some fundamental flaws like ambivalence and distrust among the network, some parts are difficult to handle despite the fact of they are being noticed. Firstly, the wastage of blood occurs due to less correlation between demand and supply for blood bags. An organization cannot take risk of reducing the collection of blood, based on the assumption that since the current requirement is low, it will also remain low going forward. An emergency can occur anytime, hence it calls for a decisive measure of proper management which can be done using a few variations as mentioned in our proposed blockchain-based web framework for blood supply management. Secondly, the factor of mistrust amongst the entities in the entire network can be a major discomfort for their constituents to function smoothly.[4] These posing challenges can be solved using our system. We aim to develop the blockchain-based implementation of the traditional supply chain management systems where traceability and efficient management is intended for the proper management of the entire distribution. In the case of conventional supply chain management systems, when a product gets launched into the market, the consumers are generally skeptical and may have some ambivalent opinions regarding the quality. This factor of ‘distrust’ is a very common issue faced and is one of the major drawbacks of using traditional supply chain management systems. Moreover, this distrust factor is of primal significance when it is encountered in blood donations and thus should be fixed as early as possible. Thus there is a necessity for a system where a donor can convincingly get the location of the blood he donated and on the corollary, the receiver of the respective blood will also want to be ensured that the blood that he is about to be injected with, is 100% free of any detrimental factors. As blockchain is immutable, transactions happening via it cannot be tampered with. Therefore, it helps build trust and encourages the use of blockchain for social events such as blood donation[1].

Chapter 2

Literature Review

Blood supply management systems should be reliable and trustable and that is where blockchain plays a huge role[1]. As blockchain can provide the ability to remain unchanged to transactions, it helps in maintaining transparency and keeping these transactions decentralized. This serves to be of great importance as far as blood bank systems are concerned since tampering with these transactions is not possible. Blood supply management systems currently face a major issue handling the data related to blood bags, donors, receivers, etc. which can be solved using blockchain as it provides a list of blocks on which this data can be stored and fetched easily. Blockchain offers important traits like decentralization, persistency, anonymity, and audibility that lowers the cost and boost the efficiency of the system[1]. Research on the blood cold chain has been carried by different methods. Davis et al. proposed an RFID-based system that dynamically manages the information related to blood[4]. Few problems which were faced by the existing system for blood information management were pointed out in one of the papers. For example, information that can be stored in the bar code is limited, and the information is not reflected in the blood source in realtime, trusting the particular actors on updating legit information on the system, etc.[4]. Jabbarzadeh et al. Suggested a sturdy network design model for supply chain of blood if a calamity occurs.[5]. In today's era of a globalized supply chain of goods and services, the supply chain has now involved various actors and entities from different parts of the globe who have never seen each other and may not trust the genuineness of either one or multiple parties/actors. With such complex issues in hand, the main concerns are the lack of transparency and traceability[7]. This is where Blockchain technologies can help us create more efficient and effective supply chains with the above concerns minimized or even completely diminished in some cases. The technological world has taken receipt of Blockchain Technology in the past years. A blockchain is a type of database that is noncentralized, reliable, and perfect when used to avoid any fraudulent activities[8]. It stores the data across the network which makes the occurrence of malpractices is difficult. Blockchain is now used in the healthcare industry to protect patient privacy, procure untampered history, and complete transactions using cryptocurrency[8]. Using this advancement, health records of patients cannot be lost or meddled. The same concept can be applied and visualized in a Blockchain-based blood supply management system. Payments can also be done by using a virtual wallet based on Ethereum cryptocurrency.

Chapter 3

Objectives

The following are the objectives we are aiming to achieve by the deployment of our proposed system:

- Security : It is extremely important to ascertain the quality of blood details including the expiry of blood from any node of the supply chain. Every node contains a copy of Blockchain, since Blockchain has disturbed ledger data structure. This architecture makes this system incredibly secure and tamper-proof because if anyone tampers with a single block then the whole chain will become invalid and we can verify from other nodes. Tracing each blood unit in the blood donation process is possible using Blockchain. The bag can be given a distinct identifier to be identified distinctively. Smart contracts validate the statistics and then enter it into the Blockchain. Thereafter, the system becomes full-proof. This way integrity, and immutability are hugely ensured across the blood ecosystem.

- Transparency : Being a decentralized platform, one of the most beneficial advantages of blockchain is transparency. All the nodes will have a copy of the blockchain which makes it easy to verify the donor history from each node in the blockchain. Sometimes even with the proper blood filtration, some unsafe blood can penetrate the testing phase which can further infect the receiver. Though this problem may go unnoticed the consequences are fatal. To overcome this problem it is required to separate the donors based on their past health records. So anyone can verify if the donor has a reliable quality of blood by checking the donor's medical history. That is possible because of the unique identity given by the Blockchain to the particular donor which is completely hack-proof. It eliminates the possibility of the distrust. As every node in blockchain contains all transactions, it can be viewed by anyone from any node. This makes the system 100% transparent.

- Availability : Along with the other things it is necessary that people should know the availability of blood bags. The absence of proper updates can harm patients. Because of blockchain it will be simple to verify the availability of blood regarding blood banks. It resolves the issue regarding the absence of coordination in demand and supply. Most of the time 30% of the patients don't get the components that they need, and on the other hand, 10-12% of the components get wasted due to expiry. Most blood donation drives are carried out based on the occasion with less co-relation with the actual blood demand. Because of one platform, we can bring all the entities together in the blood ecosystem so that the availability of blood will be known to each entity particularly to the hospitals. Hence, depending on the availability, blood donation camps can be held.

Chapter 4

Project Design

4.1 Existing System

In India existing system involves procurement of data and resources by the organizers and generating public invitations to donors and encouraging them to donate blood. Once blood is donated, the organizers hand it over to any blood banks or hospitals. The existing system for blood donations mostly involves blood donation camps organized by various non-profitable organizations or under the banner of political agendas or even under blood banks themselves. These kinds of exercises are mostly held in urban sectors and suburban townships. In rural parts, blood donation and management are managed by political representatives. Ideally, the intent or motive of such camps is the welfare of society but political agendas and displaying of the virtuous character of representatives lead to tampering of motives leading to mismanagement and tampering of data, blood, resources, or multiple of theses in combination. This, in a way, destroys the good intent of the donor and when such data or blood tampering incidents are revealed to the general public, they are discouraged to donate blood even to genuine organizations. Currently, we have online web platforms like Friends2Support, eraktkosh, etc for blood bank management systems. These platforms work without the implementation of Blockchain technology. All the existing web frameworks work without using Blockchain.

4.2 Proposed System

In order to maintain transparency in the entire system/network, our solution proposes the use of Blockchain. That is, integrating blockchain in the supply chain management network where each and every entity can be tracked as per their movement. The inclusion of Blockchain solves the issue of 'distrust'. By providing total transparency to its users, the entities involved in the network that are not be acquainted with one another and may become a potential cause for dissension during the phase of payment transactions. Our system resolves this problem, with the help of Smart Contracts, a feature of blockchain, that helps take control of the security aspect of these transactions. Only when a certain criterion is satisfied, the Smart Contract is fired, releasing the due payment. Secondly, all the data on a blockchain is stored securely through cryptographic hash function and the ledger is owned by each node in the network. This makes it very difficult for any intruder to manipulate the ledger data, ensuring protection against any possible mutations or fraudulent manipulation of the data stored in so far, thus establishing a common base of trust and immutability Also by exposing the supply chain ledger publicly, it helps gain the trust of blood donors. Our solution also provides a remedial strategy for improper waste management, which is one of the factors that deter donors away from this system. Using the data uploaded on the BC network, the internal entities can have access to the transparent data and consequently prioritize the process of dispatching the blood bags that are nearing its expiry. The bags which are nearing their expiry are flagged and are placed as ready to be dispatched in a case when an urgent call from a hospital is received.

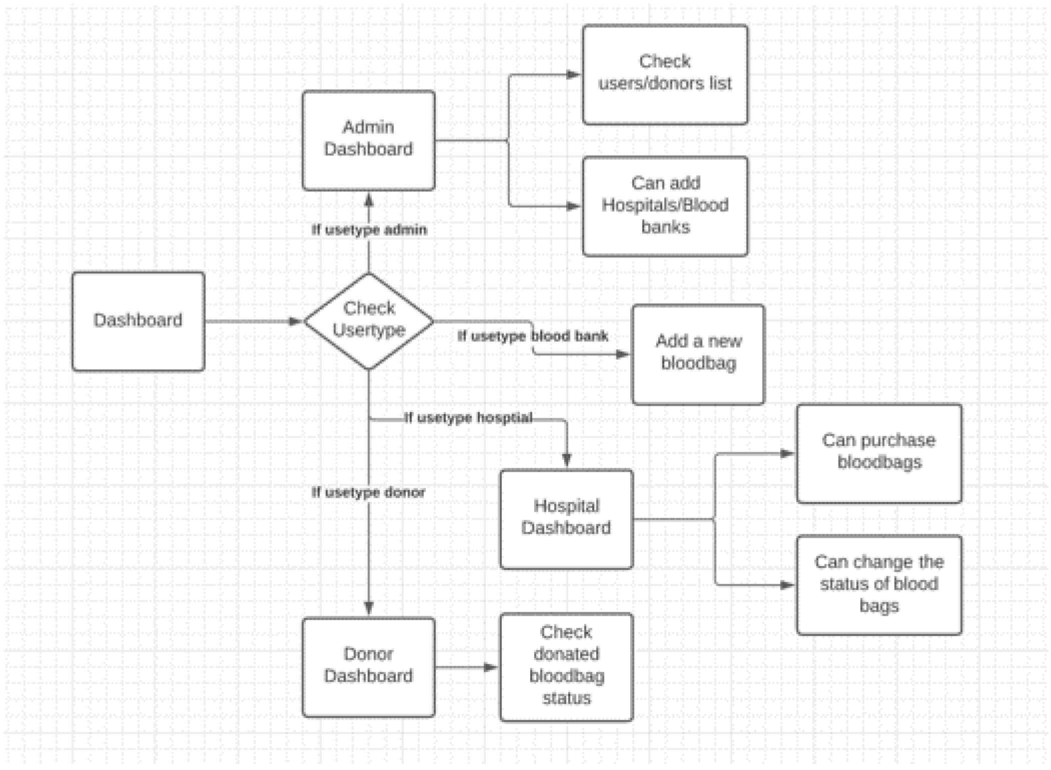


Figure 4.1: Flow of Proposed System

4.3 UML Diagrams

4.3.1 Use Case Diagram

Use case diagrams are used to gather the requirements of a system including internal and external influences. These requirements are mostly design requirements. Hence, when a system is analyzed to gather its functionalities, use cases are prepared and actors are identified. In this use case diagram, There are 4 major actors in our framework i.e Administrator, Blood bank, Hospital and Donor. Administrator has the authority to add new accounts of Blood Banks and Hospital. Hospital can purchase new blood bags and can also change the status of the usage of blood. Blood Banks can add new blood bags with their details.

4.3.2 Activity Diagram

Activity diagram is basically a flowchart to represent the flow from one activity to another activity. Activity diagrams deal with all type of flow control by using different elements such as fork, join, etc. In this activity diagram, we will discuss the system flow. When the system is started , it will identify the user type and user according to their respective metamask wallet address. Therefore, after the user is identified the system will redirect them to their respective page . The User can perform a particular set of tasks depending on the type of user, for e.g. An Admin can add blood banks, hospitals to the system and can view the list of users that are present in the system. This is the control flow for the system.

4.3.3 Sequence Diagram

Sequence Diagrams are interaction diagrams that detail how operations are carried out. They capture the interaction between objects in the context of a collaboration. Sequence Diagrams are time-focused and they show the order of the interaction visually by using the vertical axis of the diagram to represent the time what messages are sent and when. First, the Admin will add Blood banks and Hospitals to the system. Then Blood bank can add Blood bags and also can view the list of donors. The hospital requests available Blood bags from the system to purchase them and can also change the usage status of the Blood bag. The donor can request Blood bag details(i.e. whether the Blood bag is been used or it got expired). Admin can also view the list of all users present on the system.

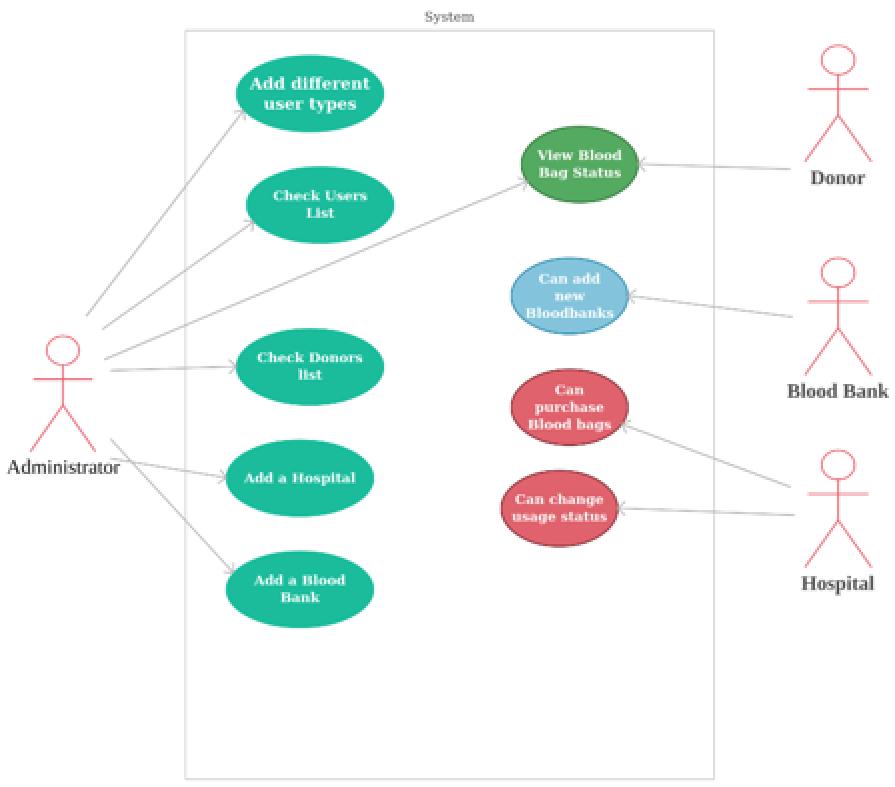


Figure 4.2: Use Case Diagram

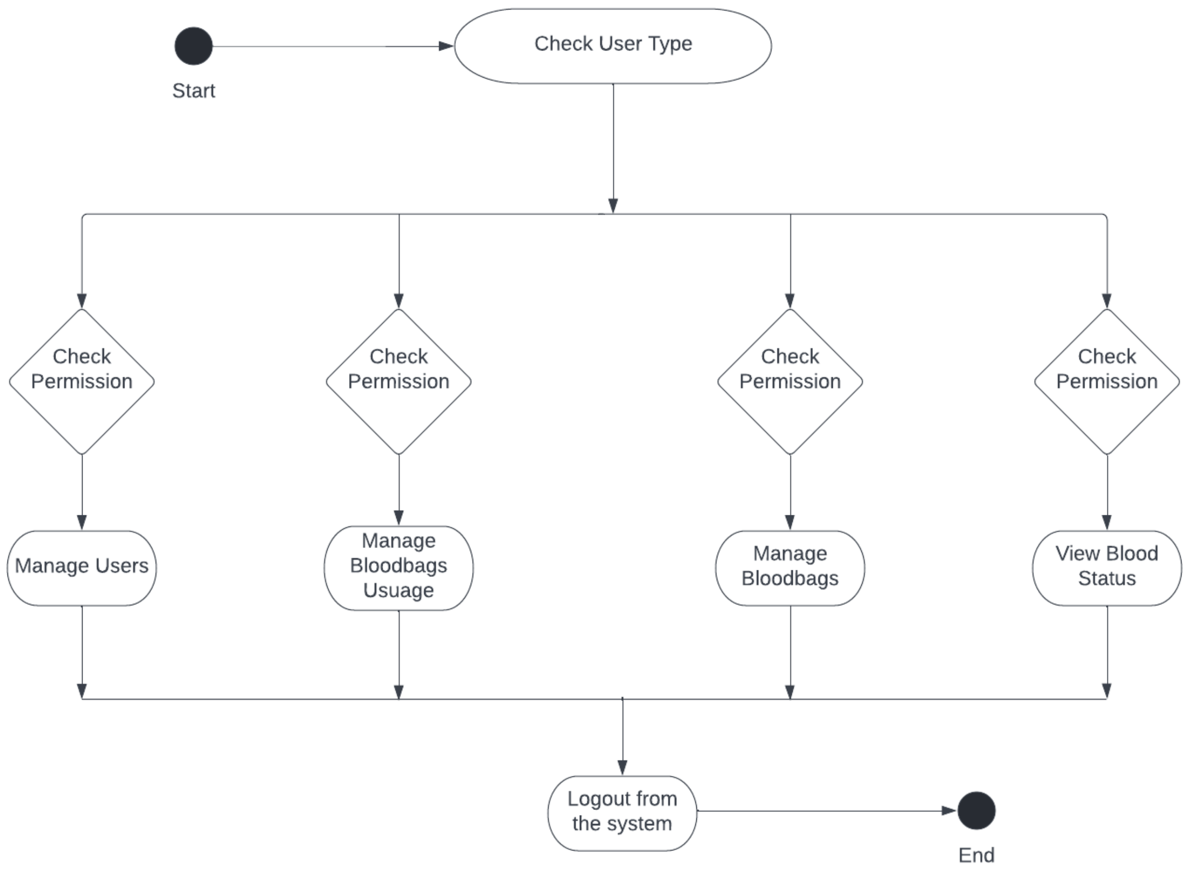


Figure 4.3: Activity Diagram

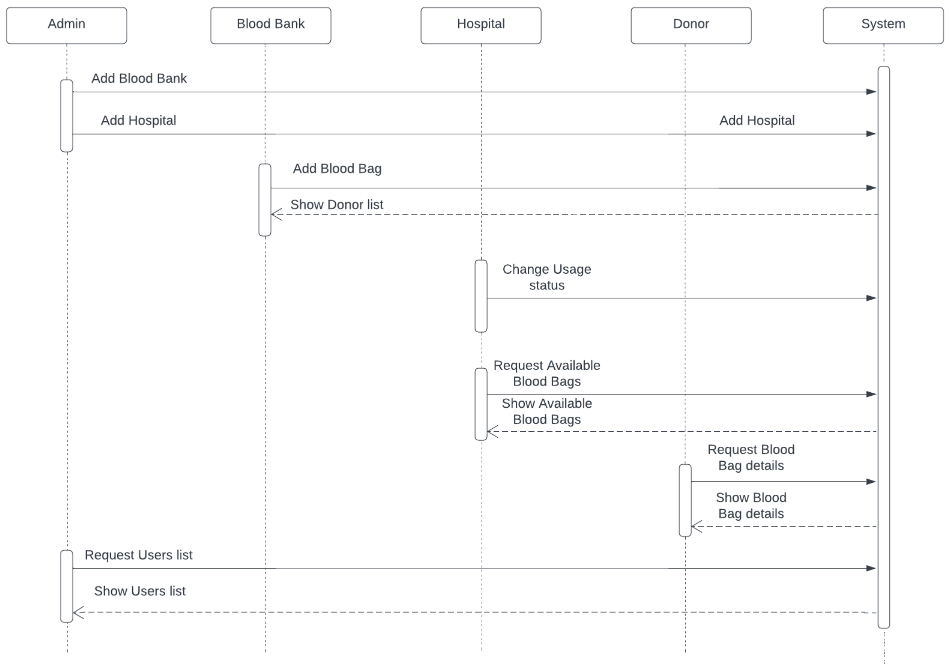


Figure 4.4: Sequence Diagram

Chapter 5

Project Implementation

5.1 Technology Stack

For the proposed system we are using the following technology stack

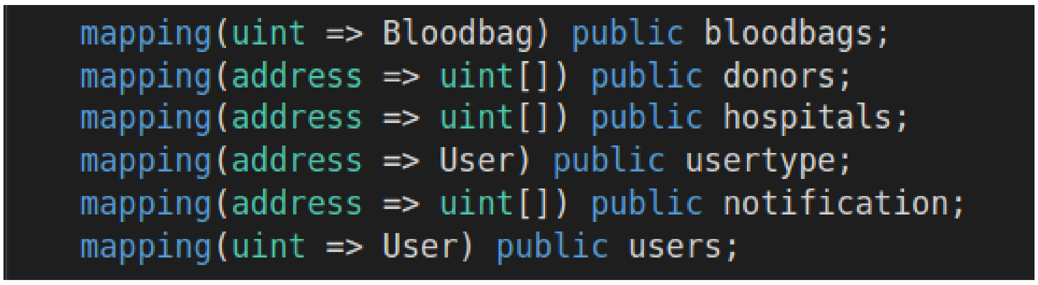
1. Solidity programming language for smart contract creation
2. Ganache (Local blockchain for testing and development)
3. React.js, Node server and npm for web application
4. Metamask Wallet for handling cryptocurrency

5.2 Application Implementation

5.2.1 System Architecture

Our system, heavily relies on the 'Smart Contracts' written in programming solidity. Shown below are the snippets of our main contract – 'Blood.sol'.

- **Mappings**:
  Mappings work as dictionary or hash tables where data is stored in key-value pair[10]. Here we have 6 main mappings:



```
    mapping(uint => Bloodbag) public bloodbags;
    mapping(address => uint[]) public donors;
    mapping(address => uint[]) public hospitals;
    mapping(address => User) public usertype;
    mapping(address => uint[]) public notification;
    mapping(uint => User) public users;
```

Figure 5.1: Mappings

1. **Bloodbags** - This is a very general list of bloodbags. It stores and returns the entire structure(object) of a Bloodbag, whose ‘id’ will be equal to the uint (integer) provided to the mapping.

2. **Donors** - The Donors mapping will store and return an array, consisting of the bloodbag id(s) corresponding to each and every donor for a particular Blood Collection Center. Based on the address of the Bank provided, the list of donors for that center will be retrieved.

3. **Hospitals** - This mapping stores and returns an array, consisting of the bloodbag id(s) that are currently in possession and owned by the given address’s hospital.

4. **Usertype** - This is the type of mapping used to bifurcate different users in the system. It is used specifically to provide the client’s information, such as the type of user. For example -

   - ID 1 is for Admin
   - ID 2 is for Donor
   - ID 3 is for Blood banks
   - ID 4 is for Hospitals

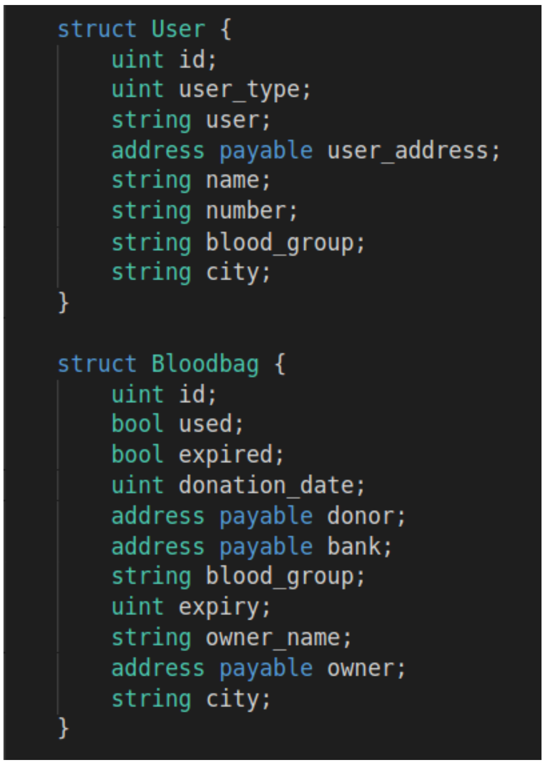
   This ‘usertype’ mapping helps the system to dynamically recognize the type of client, who is visiting the webpage. Using usertype and also address of the client, we can provide a fluent access to the data belonging to the user/client

5. **Notification** - This mapping stores and returns an array, consisting of the bloodbag id(s) that have been used by the hospital(s). It is then used to notify respective donors about their donated blood. It is initially set to false by Blood Bank admin and updated to true by Hospital Admin.

6. **Users** - Lastly, the ‘users’ mapping is used to contain the list of entire user network, including the donors, hospitals, and the banks. It helps to keep the access of id of every user in a simplistic yet efficient manner.

5.2.2 Structs

Every system has to use a type of data structure that is efficient enough to handle the application of the system and suffice with its use-case. Our system comprises of two different



```
struct User {
    uint id;
    uint user_type;
    string user;
    address payable user_address;
    string name;
    string number;
    string blood_group;
    string city;
}

struct Bloodbag {
    uint id;
    bool used;
    bool expired;
    uint donation_date;
    address payable donor;
    address payable bank;
    string blood_group;
    uint expiry;
    string owner_name;
    address payable owner;
    string city;
}
```

Figure 5.2: Structs

'structs'.

1. Users - The user struct is a skeleton holding certain parameters for each and every user that is registered on our network. These value are following:
– Id - Used for listing and identification purposes.
– user type – It helps denote the category of the user (Bank/Hospital/Donor).
– user – This includes the description of given user.
– user's network address – This contains the payable address of the user which will be used for transactions.
– unique name – For listing and Identification Purposes.

2. Bloodbag - The bloodbag struct holds the information unique to each and every bag created/donoted to the blood bank, that enters into our network.
– Id - Used for listing and identification purposes
– Donation date – It is factored in for the authenticity of the bloodbag created. It helps to calculate the expiry date of donated blood.
– Used – A boolean field to check if bag is used or not.
– Address (Donor) – Used to maintain the link to donor, inorder for the donor certificate's tracking.
– Address(bank) – Used for maintaining the relation to the bank that has generated the bag.
– Blood Group – Used for Medical purposes.
– Expiry – Used for secure medical Purposes.
– Owner name – Used to track the current dynamic ownership/holder of the bag in the system.

– Address Owner – Used to track the purchase details for the owner where the payment is transacted.

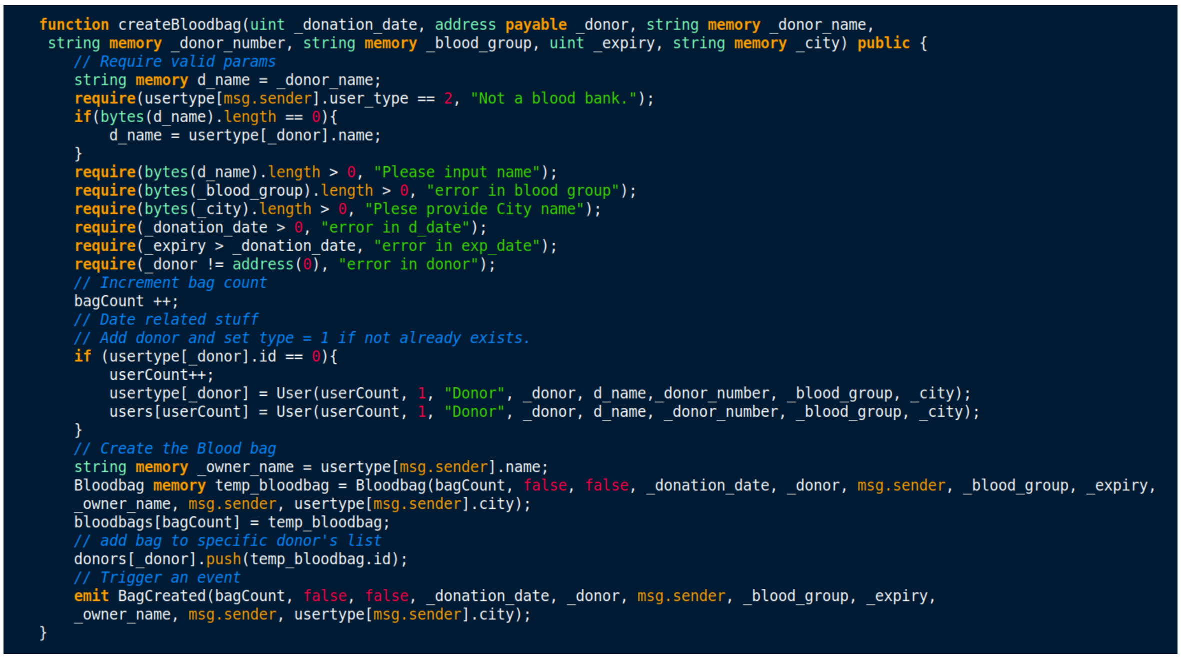
5.2.3 Functions

• **Admin Functions:** These are the functions designed for the admin panel to carry out the back-end tasks namely:

1. createBank – The admin will assign the registration 'creation' of every blood bank.
2. createHosp – The admin will handle the registrations for every hospital that get affiliated with our network.

• **System Functions:**

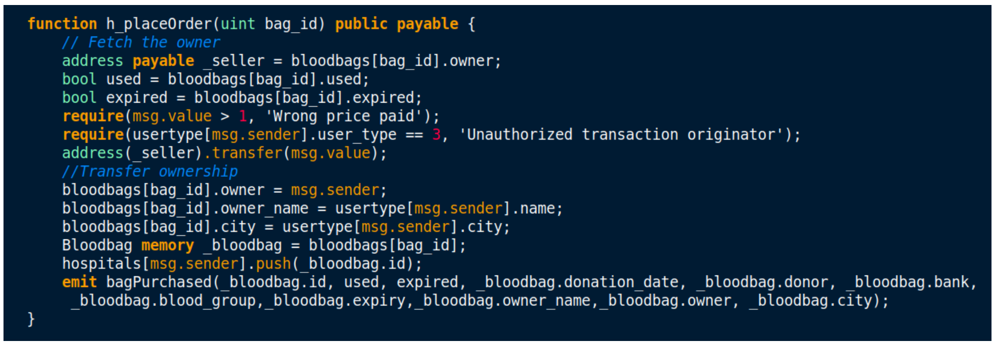
1. createBloodbag – The banks on receiving the donations made by the donor, will log in to their systems and register the details on the blood bag.



```
function createBloodbag(uint _donation_date, address payable _donor, string memory _donor_name,
 string memory _donor_number, string memory _blood_group, uint _expiry, string memory _city) public {
    // Require valid params
    string memory d_name = _donor_name;
    require(usertype[msg.sender].user_type == 2, "Not a blood bank.");
    if(bytes(d_name).length == 0){
        d_name = usertype[_donor].name;
    }
    require(bytes(d_name).length > 0, "Please input name");
    require(bytes(_blood_group).length > 0, "error in blood group");
    require(bytes(_city).length > 0, "Plese provide City name");
    require(_donation_date > 0, "error in d_date");
    require(_expiry > _donation_date, "error in exp_date");
    require(_donor != address(0), "error in donor");
    // Increment bag count
    bagCount ++;
    // Date related stuff
    // Add donor and set type = 1 if not already exists.
    if (usertype[_donor].id == 0){
        userCount++;
        usertype[_donor] = User(userCount, 1, "Donor", _donor, d_name,_donor_number, _blood_group, _city);
        users[userCount] = User(userCount, 1, "Donor", _donor, d_name, _donor_number, _blood_group, _city);
    }
    // Create the Blood bag
    string memory _owner_name = usertype[msg.sender].name;
    Bloodbag memory temp_bloodbag = Bloodbag(bagCount, false, false, _donation_date, _donor, msg.sender, _blood_group, _expiry,
    _owner_name, msg.sender, usertype[msg.sender].city);
    bloodbags[bagCount] = temp_bloodbag;
    // add bag to specific donor's list
    donors[_donor].push(temp_bloodbag.id);
    // Trigger an event
    emit BagCreated(bagCount, false, false, _donation_date, _donor, msg.sender, _blood_group, _expiry,
    _owner_name, msg.sender, usertype[msg.sender].city);
}
```

Figure 5.3: Function for creating a blood bag

2. h placeOrder – The hospitals, when in need of new blood bags, will order for the purchase of bloodbags from the nearby Banks and other Hospitals. This transaction will be performed by this function.



```
function h_placeOrder(uint bag_id) public payable {
    // Fetch the owner
    address payable _seller = bloodbags[bag_id].owner;
    bool used = bloodbags[bag_id].used;
    bool expired = bloodbags[bag_id].expired;
    require(msg.value > 1, 'Wrong price paid');
    require(usertype[msg.sender].user_type == 3, 'Unauthorized transaction originator');
    address(_seller).transfer(msg.value);
    //Transfer ownership
    bloodbags[bag_id].owner = msg.sender;
    bloodbags[bag_id].owner_name = usertype[msg.sender].name;
    bloodbags[bag_id].city = usertype[msg.sender].city;
    Bloodbag memory _bloodbag = bloodbags[bag_id];
    hospitals[msg.sender].push(_bloodbag.id);
    emit bagPurchased(_bloodbag.id, used, expired, _bloodbag.donation_date, _bloodbag.donor, _bloodbag.bank,
     _bloodbag.blood_group,_bloodbag.expiry,_bloodbag.owner_name,_bloodbag.owner, _bloodbag.city);
}
```

Figure 5.4: Function for placing a blood bag order

3. useBag – Whenever the hospital uses a blood bag from its inventory, the donor of that blood bag will be notified. The bag ID will be pushed at the back of the notification array corresponding to the donor ID of same bag.

5.2.4 Working

• The framework uses the Metamask wallet to handle the crypto currency. In order to make any transaction with the web application all the entites will need an account on metamask. If an entity does not have an account in the system and tries to interact with web application then it will not allow that entity to use the application as the metamask account is a mandatory requirement.
• By using Metamask we can skip the traditional login/register part in this system as metamask handles all the accounts by default. Therefore, overall system becomes simple, fast but secure too, as users just have to log into there wallet account to proceed.[12]
• The framework will automatically detect the usertype and according to that it will redirect to the desired page. For example - if a blood bank accesses the application, it will redirected to the bank's page where it allows them to add new bloodbags.
• Admin can only add new blood banks and hospitals manually. Admin can also check donor's list by blood group. Bloodbank can add new blood bags if available. After adding new blood bags it will be shown in hospital's available blood bag's list from where any hospital can purchase the blood bags.
• Once hospital buys the bag it will be shown in the hospital's inventory as available for use. If a bag gets used for a patient then hospital will change the bag's usage status as used in the web application.
• Donor can also check the blood bag's expiry date and usage status by accessing the application.

Chapter 6

Testing

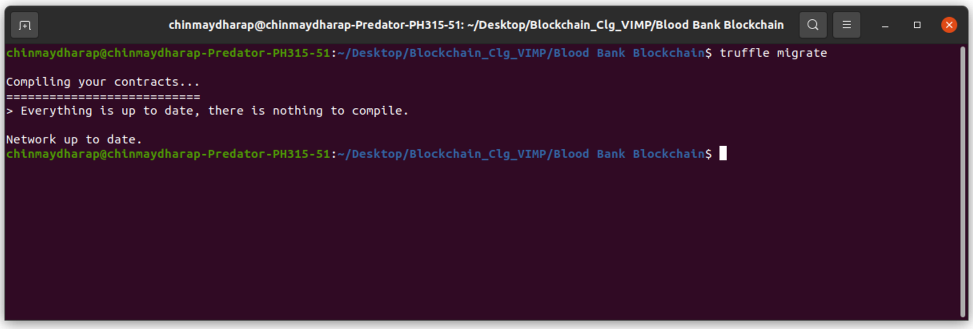
6.1 Functional Testing

6.1.1 Unit Testing

Unit testing is used in Truffle (an Ethereum development framework) using plugins Chai and Mocha to test various transactions taking place in the smart contract. Each test function is executed as a single transaction, in order of appearance in the test file (as shown in the previous section). Assertion functions provided by truffle/Assert. ... This is to make writing solidity unit tests easier, and should allow for more extensibility in the future with less hassle. We decided to go with the Truffle framework as it comes standard with an automated testing framework to make testing the contracts a breeze. This framework lets us write simple and manageable tests in two different ways:

- In JavaScript and TypeScript, for exercising your contracts from the outside world, just like your application.
- In Solidity, for exercising your contracts in advanced, bare-to-the-metal scenarios.

The transactions and blocks can be viewed on Ganache while conducting the tests. It is included in the truffle suite to develop applications and run tests. Given below is the output of the testing code shown in the previous section.



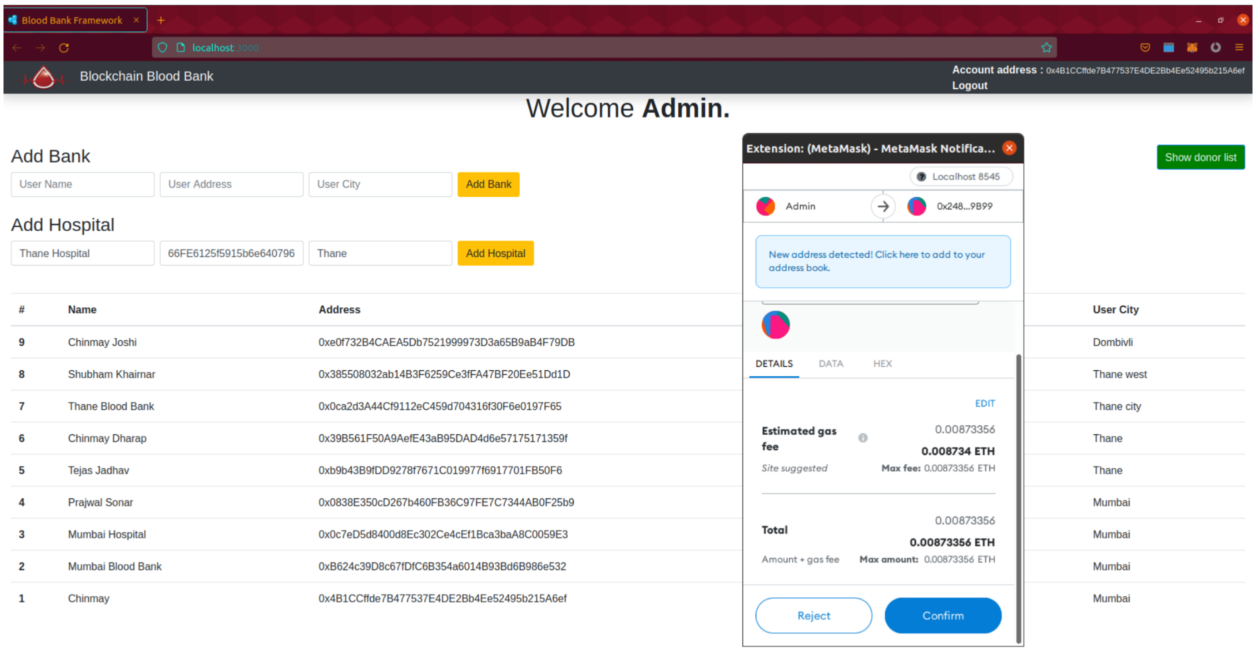
```
chinmaydharap@chinmaydharap-Predator-PH315-51: ~/Desktop/Blockchain_Clg_VIMP/Blood Bank Blockchain
chinmaydharap@chinmaydharap-Predator-PH315-51:~/Desktop/Blockchain_Clg_VIMP/Blood Bank Blockchain$ truffle migrate

Compiling your contracts...
===========================
> Everything is up to date, there is nothing to compile.

Network up to date.
chinmaydharap@chinmaydharap-Predator-PH315-51:~/Desktop/Blockchain_Clg_VIMP/Blood Bank Blockchain$
```

Figure 6.1: Smart contract compilation

6.1.2 Integration Testing

Integration testing is the phase in software testing in which individual software modules are combined and tested as a group. Integration testing is conducted to evaluate the compliance of a system or component with specified functional requirements. It occurs after unit testing and before system testing.

After compiling the smart contract now we need to integrate it with our front-end web application in order to achieve the final results i.e a fully working blockchain based web application. To do this we are going to use node server and reactjs. Once done with the designing and connectivity of web app to the contract we can test the application's user interface.



Account address : 0x4B1CCffde7B477537E4DE2Bb4Ee52495b215A6ef

Welcome **Admin.**

Add Bank

User Name	User Address	User City

Add Hospital

Thane Hospital	66FE6125f5915b6e640796	Thane

#	Name	Address	User City
9	Chinmay Joshi	0xe0f732B4CAEA5Db7521999973D3a65B9aB4F79DB	Dombivli
8	Shubham Khairnar	0x385508032ab14B3F6259Ce3fFA47BF20Ee51Dd1D	Thane west
7	Thane Blood Bank	0x0ca2d3A44Cf9112eC459d704316f30F6e0197F65	Thane city
6	Chinmay Dharap	0x39B561F50A9AefE43aB95DAD4d6e57175171359f	Thane
5	Tejas Jadhav	0xb9b43B9fDD9278f7671C019977f6917701FB50F6	Thane
4	Prajwal Sonar	0x0838E350cD267b460FB36C97FE7C7344AB0F25b9	Mumbai
3	Mumbai Hospital	0x0c7eD5d8400d8Ec302Ce4cEf1Bca3baA8C0059E3	Mumbai
2	Mumbai Blood Bank	0xB624c39D8c67fDfC6B354a6014B93Bd6B986e532	Mumbai
1	Chinmay	0x4B1CCffde7B477537E4DE2Bb4Ee52495b215A6ef	Mumbai

Extension: (MetaMask) - MetaMask Notifica...

Localhost 8545

Admin → 0x248...9B99

New address detected! Click here to add to your address book.

Estimated gas fee	0.00873356
	0.008734 ETH
Site suggested	**Max fee:** 0.00873356 ETH
Total	0.00873356
	0.00873356 ETH
Amount + gas fee	**Max amount:** 0.00873356 ETH

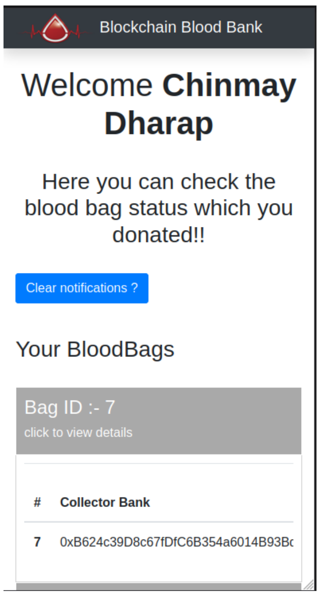
Figure 6.2: Complete Web Framework

6.2 Non Functional Testing

6.2.1 Compatibility Testing

Compatibility testing is a part of non-functional testing conducted on application software to ensure the application's compatibility with different computing environment.

The proposed framework uses the solidity smart contracts as it's back-end. As it's a web application it is responsive for all the desktop resolutions. It is also not limited to the desktop only as Metamask also provides the necessary support for android/ios devices.


Blockchain Blood Bank

Welcome **Chinmay Dharap**

Here you can check the blood bag status which you donated!!

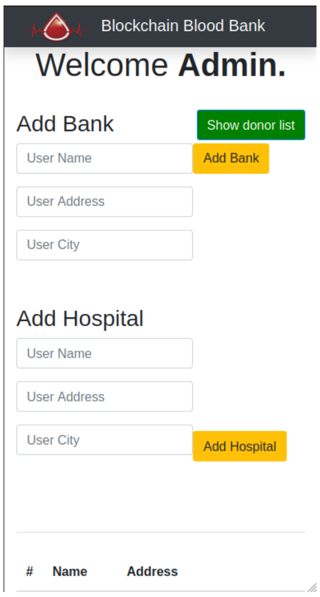
Your BloodBags

Bag ID :- 7

click to view details

#	Collector Bank
7	0xB624c39D8c67fDfC6B354a6014B93B

Figure 6.3: Donor Dashboard


Blockchain Blood Bank

Welcome **Admin.**

Add Bank

Add Hospital

#	Name	Address

Figure 6.4: Admin Dashboard

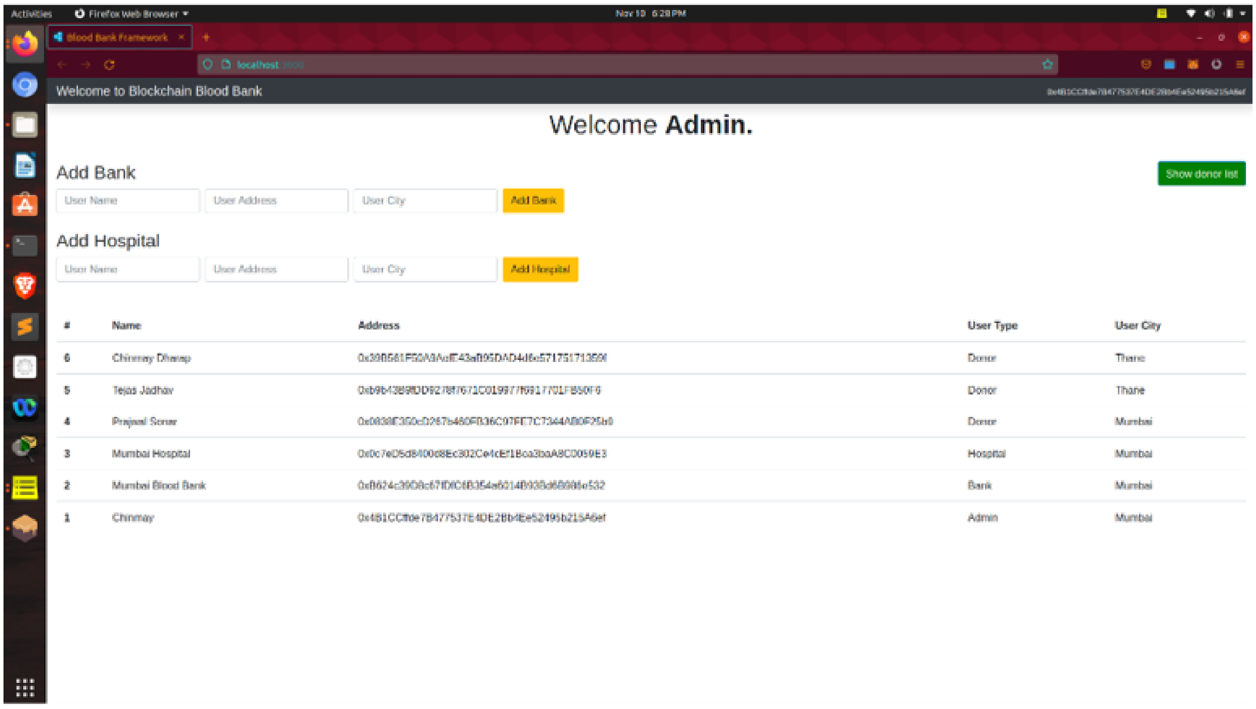
Chapter 7

Result

The end result after integrating the solidity smart contract with the web application would be a blockchain based web framework which is ready for the end users. This has four main web pages i.e admin dashboard, blood bank dashboard, hospital dashboard and lastly donor's dashboard. All these pages are elaborated below.

7.1 Admin Screen

Admin can view the list of Users of the Blood bank system. He/She can add a new Blood Bank and a new Hospital. Therefore only approved Blood Bank and Hospital by the admin can exist on the system . He/She can also view a list only of donors.



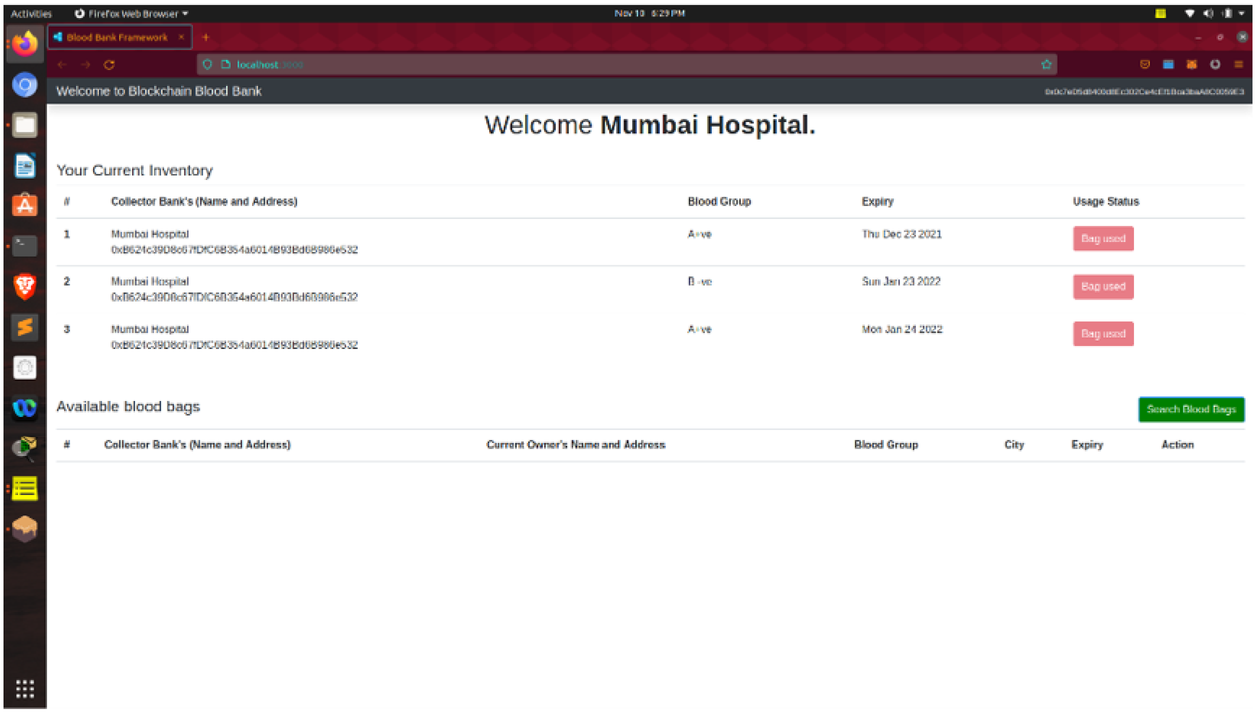
Welcome **Admin.**

#	Name	Address	User Type	User City
6	Chinmay Dhamp	0x39B561F50A9AdE43aB95DAD4d6c57175171359f	Donor	Thane
5	Tejas Jadhav	0xb9b4389fDD9278f7671C019977f6917701FB50F6	Donor	Thane
4	Prajwal Sonar	0x0838E350cD267b480FB36C97FE7C7344A80F25b0	Donor	Mumbai
3	Mumbai Hospital	0x0c7eD5d8400d8Ec302Ce4cEf1Bca3baA8C0059E3	Hospital	Mumbai
2	Mumbai Blood Bank	0xf8624c39D8c67fDfC6B354e6014B935d6B686e532	Bank	Mumbai
1	Chinmay	0x4B1CCffde7B477537E4DE2Bb4Ee52495b215A6ef	Admin	Mumbai

Figure 7.1: Admin Dashboard

7.2 Hospital Dashboard

A hospital can view the available Blood bags or can search for the Blood bags available. Also a hospital can change the usage status of the blood bag.



Welcome **Mumbai Hospital.**

Your Current Inventory

#	Collector Bank's (Name and Address)	Blood Group	Expiry	Usage Status
1	Mumbai Hospital 0xB621c39D8c67fDfC6B354a6014B93Bd6B986e532	A+ve	Thu Dec 23 2021	Bag used
2	Mumbai Hospital 0xB624c39D8c67fDfC6B354a6014B93Bd6B986e532	B -ve	Sun Jan 23 2022	Bag used
3	Mumbai Hospital 0xB621c39D8c67fDfC6B354a6014B93Bd6B986e532	A+ve	Mon Jan 24 2022	Bag used

Available blood bags

#	Collector Bank's (Name and Address)	Current Owner's Name and Address	Blood Group	City	Expiry	Action

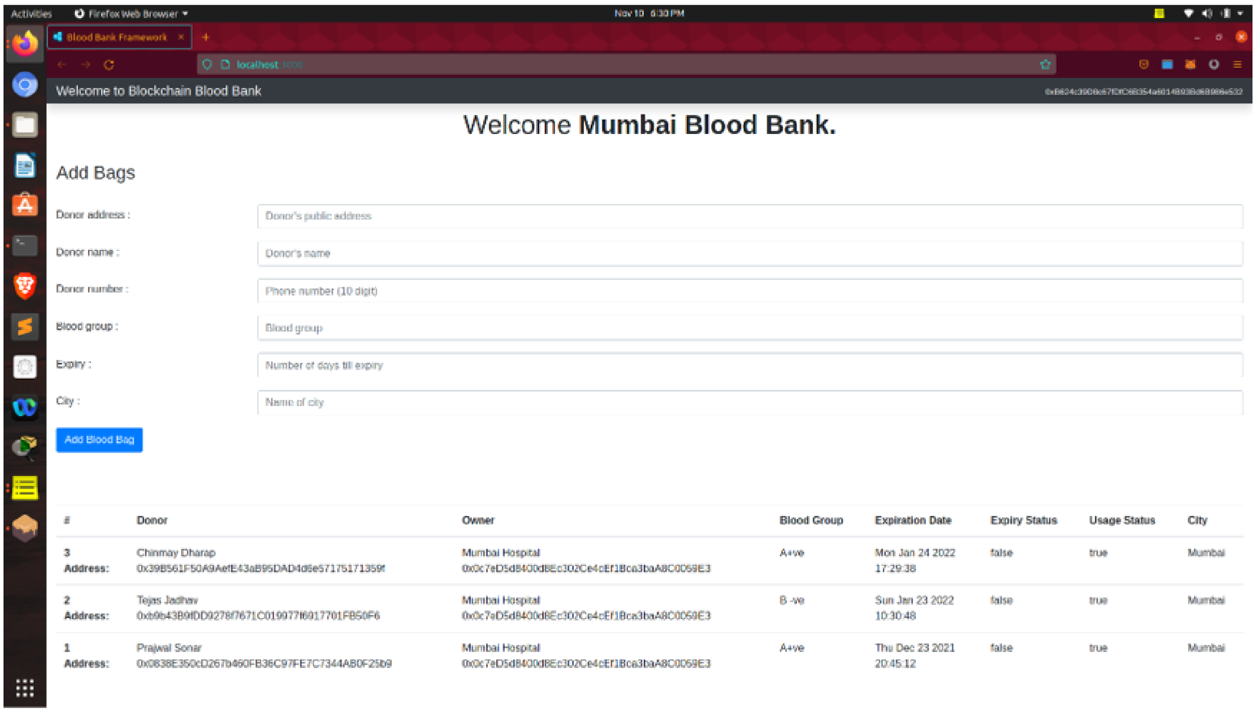
Figure 7.2: Hospital Dashboard

7.3 Blood bank Dashboard

A Blood Bank can add new blood bags by entering all the donor details along with the blood details, also can view all the information about the blood bags they have collected.

7.4 Donor Dashboard

This is the Donor's dashboard where donors can keep the track on their donated blood. They can also check the usage status and expiry status.



Welcome to Blockchain Blood Bank 0xB624c39D8c67fDfC6B354a6014B93Bd6B986e532

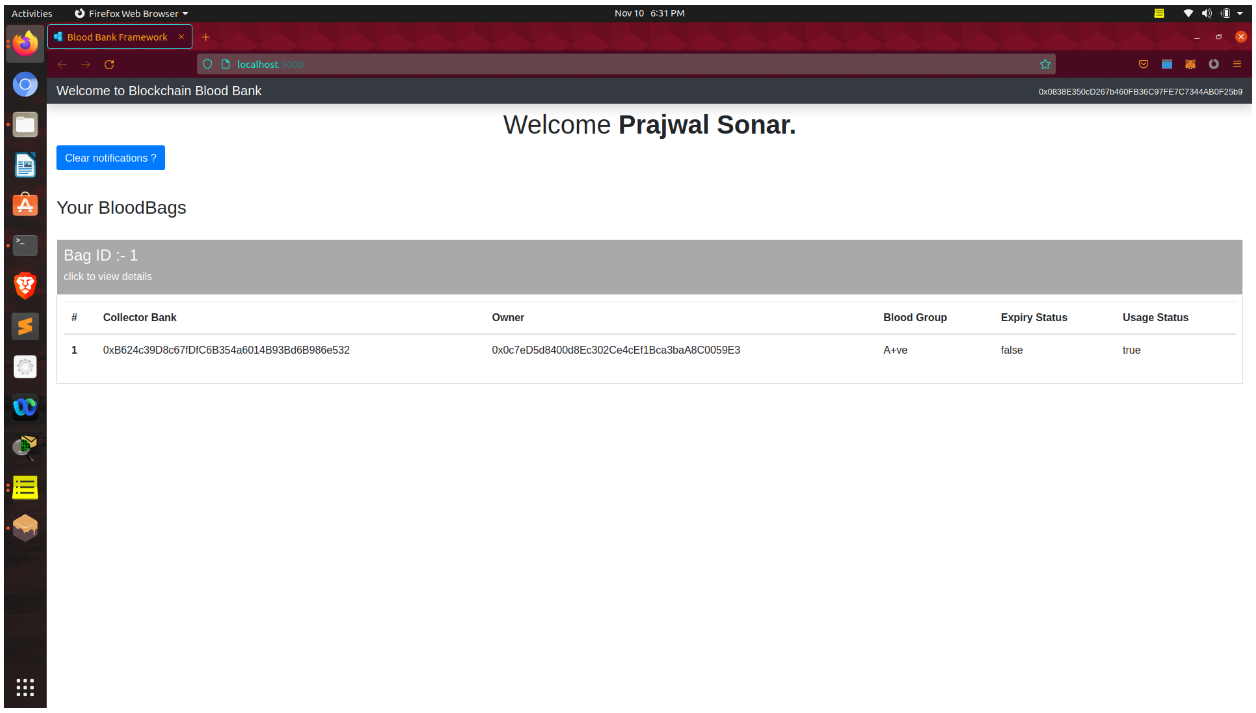
Welcome **Mumbai Blood Bank.**

Add Bags

Donor address :	Donor's public address
Donor name :	Donor's name
Donor number :	Phone number (10 digit)
Blood group :	Blood group
Expiry :	Number of days till expiry
City :	Name of city

#	Donor	Owner	Blood Group	Expiration Date	Expiry Status	Usage Status	City
3 **Address:**	Chinmay Dharap 0x39B561F50A9AefE43aB95DAD4d6e57175171359f	Mumbai Hospital 0x0c7eD5d8400d8Ec302Ce4cEf1Bca3baA8C0059E3	A+ve	Mon Jan 24 2022 17:29:38	false	true	Mumbai
2 **Address:**	Tejas Jadhav 0xb9b43B9fDD9278f7671C019977f6017701FB50F6	Mumbai Hospital 0x0c7eD5d8400d8Ec302Ce4cEf1Bca3baA8C0059E3	B -ve	Sun Jan 23 2022 10:30:48	false	true	Mumbai
1 **Address:**	Prajwal Sonar 0x0838E350cD267b460FB36C97FE7C7344AB0F25b9	Mumbai Hospital 0x0c7eD5d8400d8Ec302Ce4cEf1Bca3baA8C0059E3	A+ve	Thu Dec 23 2021 20:45:12	false	true	Mumbai

Figure 7.3: Blood bank Dashboard



Welcome to Blockchain Blood Bank 0x0838E350cD267b460FB36C97FE7C7344AB0F25b9

Welcome **Prajwal Sonar.**

Your BloodBags

Bag ID :- 1

click to view details

#	Collector Bank	Owner	Blood Group	Expiry Status	Usage Status
1	0xB624c39D8c67fDfC6B354a6014B93Bd6B986e532	0x0c7eD5d8400d8Ec302Ce4cEf1Bca3baA8C0059E3	A+ve	false	true

Figure 7.4: Donor Dashboard

Chapter 8

Conclusions and Future Scope

By using blockchain we can achieve three main objectives - Security, Transparency, and Availability. Because of its distributed ledger architecture, it is completely tamper-proof. Thus we have made a web-based framework for blood banks that use blockchain in the backend to store the data of blood donations. This is a centralized platform for the blood ecosystem so that it can solve the issue of improper coordination in demand and supply and avoid misunderstandings between blood banks and hospitals. It also helps donors to check the blood bag's usage and expiry status. The platform proposed will help avoid any kind of interference in blood donation drives for personal benefits. The system that has been proposed will help in avoiding the wastage of blood by offering a unified platform for the exchange of blood and its derivatives between blood banks which brings more transparency to the process of blood donation by tracking the blood trail. It can be beneficial in areas where political interference and chances of data tampering for personal benefits are high. As this system will include a web application, it becomes more user-friendly and can be reached to more population.

Bibliography

[1] Z. Zheng, S. Xie, H. Dai, X. Chen and H. Wang, ”An Overview of Blockchain Technology: Architecture, Consensus, and Future Trends,” 2017 IEEE International Congress on Big Data (BigData Congress), 2017, pp. 557-564, doi: 10.1109/BigDataCongress.2017.85.

[2] Sylim, Patrick & Liu, Fang & Marcelo, Alvin & Fontelo, Paul. “Blockchain technology for detecting falsified and substandard drugs in the pharmaceuticals distribution system”. In: JMIR Research Protocols. Vol. 7. (2018).

[3] Feng Tian. “An agri-food supply chain traceability system for China based on RFID & blockchain technology”. In: Service Systems and Service Management (ICSSSM), 13th International Conference on. IEEE, pp. 1–6, (2016).

[4] Davis, R. & Geiger, Bradley & Gutierrez, Alfonso & Heaser, Julie & Veeramani, Dharmaraj. “Tracking blood products in blood centers using radio frequency identification: A comprehensive assessment”. In: Vox Sanguinis, vol. 91, pp. 50-60, (2009).

[5] Jabbarzadeh, F. Behnam and S. Stefan. “Dynamic supply chain network design for the supply of blood in disasters: A robust model with real-world application”. In: Transportation Research Part E: Logistics and Transportation Review, vol. 70, pp. 225- 244, (2014).

[6] K. M. Giannoutakis et al., ”A Blockchain Solution for Enhancing Cybersecurity Defence of IoT,” 2020 IEEE International Conference on Blockchain (Blockchain), 2020, pp. 490-495, doi: 10.1109/Blockchain50366.2020.00071.

[7] K. M. Giannoutakis et al., ”A Blockchain Solution for Enhancing Cybersecurity Defence of IoT,” 2020 IEEE International Conference on Blockchain (Blockchain), 2020, pp. 490-495, doi: 10.1109/Blockchain50366.2020.00071.

[8] P. Tasatanattakool and C. Techapanupreeda, ”Blockchain: Challenges and applications,” 2018 International Conference on Information Networking (ICOIN), 2018, pp. 473-475, doi: 10.1109/ICOIN.2018.8343163.

Appendices

These following appendices mention the procedure for installation of platforms required in the proposed system:

Appendix-A: Node Server installation Commands in Ubuntu

1. sudo apt update

2. sudo apt install nodejs

3. node -v

4. sudo apt install npm

Appendix-B: Ganache and Truffle Download and Installation

1. Download Ganache AppImage File for the Ubuntu from
https://trufflesuite.com/ganache/

2. Simply Run the file to start Ganache

3. To install truffle run the following command:
npm install truffle -g

Appendix-C: Metamask Wallet Download and Installation

1. Download the Metamask Firefox plugin from
https://addons.mozilla.org/en-US/firefox/addon/ether-metamask/