A

PROJECT REPORT ON

BLOCKCHAIN BASED MANAGEMENT FOR ORGAN DONATION AND TRANSPLANTATION

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

MASTER OF COMPUTER APPLICATIONS

By

P. BALAJI
(21BF1F0094)

Under the esteemed guidance of

Mr. J.Kiran Kumar M.C.A (Ph.D)
Assistant Professor

DEPARTMENT OF MASTER OF COMPUTER APPLICATIONS

SRIVENKATESWARACOLLEGEOFENGINEERING
(AUTONOMOUS)
(Approved by AICTE, New Delhi & Affiliated to JNTUA, Anantapuramu)
Accredited by NAAC with 'A' Grade
Opp.LIC Training Centre, Karakambadi Road, TIRUPATI–517507

2021-2023

***

SRI VENKATESWARA COLLEGE OF ENGINEERING (AUTONOMOUS)

(Approved by AICTE, New Delhi & Affiliated to JNTUA, Anantapuramu)

Accredited by NAAC with 'A' Grade

Opp.LIC Training Centre, Karakambadi Road, TIRUPATI – 517507

DEPARTMENT OF MASTER OF COMPUTER APPLICATIONS

2021-2023

CERTIFICATE

This is to certify that the project report entitled **"BLOCKCHAIN BASED MANAGEMENT FOR ORGAN DONATION AND TRANSPLANTATION"** *is a bonafide record of the project work done and submitted by*

P. BALAJI

(21BF1F0094)

for the partial fulfillment of the requirements for the award of **MASTER OF COMPUTER APPLICATIONS** *Degree from SRI VENKATESWARA COLLEGE OF ENGINEERING (AUTONOMOUS) Affiliated to JNT University Anantapur, Anantapuramu.*

GUIDE **HEAD OF THE DEPARTMENT**

INTERNAL EXAMINER **EXTERNAL EXAMINER**

ACKNOWLEDGEMENT

I am thankful to my guide **Mr. J. Kiran Kumar, Assistant Professor** for her valuable guidance and encouragement. His helping attitude and suggestions have helped me in the successful completion of the project**.**

I would like to express my sincere thanks to **Dr. E. Sreedevi,** Professor, Head of the Department of MCA, for her kind help and encouragement during the course of my study and in the successful completion of the project work.

I express my sincere thanks to **Dr. N. Sudhakar Reddy**, Principal, S.V College of Engineering, Tirupati.

Successful completion of any project cannot be done without proper support and encouragement. My sincere thanks to the Management for providing all the necessary facilities during the Course of my study.

I would like to thank my parents and friends, who have the greatest contributions in all my achievements, for the great care and blessings in making me successful in all my endeavors.

I would like to express my deep gratitude to all those who helped me directly or indirectly to transform an idea into my working project.

P. BALAJI

(21BF1F0094)

DECLARATION

I hereby declare that the project entitled **"BLOCKCHAIN BASED MANAGEMENT FOR ORGAN DONATION AND TRANSPLANTATION"** submitted to the Department Of COMPUTER APPLICATIONS, **S.V.COLLEGE OF ENGINEERING, TIRUPATI** in partial fulfillment of requirements for the award of the degree of **MASTER OF COMPUTER APPLICATIONS**.

This project is the result of my own effort and it has not been submitted to any other University or Institution for the award of any degree other than specified above.

Signature of the student

P. BALAJI

(21BF1F0094)

ABSTRACT

Today's organ donation and transplantation systems pose different requirements and challenges in terms of registration, donor-recipient matching, organ removal, organ delivery, and transplantation with legal, clinical, ethical, and technical constraints. Therefore, an end-to-end organ donation and transplantation system is required to guarantee a fair and efficient process to enhance patient experience and trust. In this paper, we propose a private Ethereum blockchain-based solution to enable organ donation and transplantation management in a manner that is fully decentralized, secure, traceable, auditable, private, and trustworthy. We develop smart contracts and present six algorithms along with their implementation, testing, and validation details.

TABLE OF CONTENTS

LIST OF FIGURES

1. INTRODUCTION

Organ failure or damage occurs due to an injury or a disease. It affects the quality of life and, in some cases, leads to death. Donating an organ is one of humanity's most honorable actions to save the lives of patients through organ transplantation. For a successful transplant, the organ must be in acceptable working conditions with donor-recipient matching, and its removal should not pose a life-threatening risk to the donor [1]. The first successful organ donation occurred with a kidney transplant between twin brothers in 1954 [2]. Since then, the annual number of transplants has steadily increased. However, the demand for organ donations still exceeds the number of donors [3]. In fact, while waiting for an organ transplant, twenty people die every day, and a new patient is added to the waiting list in every ten minutes [4]. More importantly, accessing the organ donation waiting list is a basic requirement for organ allocation. Referral for transplantation can be affected by both geographical and socioeconomic factors. Therefore, the allocation process on the waiting list should not discriminate against certain groups of patients [4].

Organ donation is conducted in two different ways, including deceased donation and living donation. Figure 1 illustrates the typical flow chart for donating an organ and transplanting it to a patient. First, the donor is examined by the hospital transplant team, and if the donor is deceased, a brain death test is performed. Meanwhile, if the donor is still alive, doctors examine the donor and ensure that the donor is fit for live donation. Then, all medical records are reported to the procurement organizer. The procurement organizer is responsible for evaluating the donor's condition to decide if he is a fit donor and ensuring that the donor is properly registered in the medical system. Next, if the evaluation shows that the donor is eligible for donation, the procurement organizer sends all the data to the organ transplantation organizer. This step can be performed only if the donor gives consent to donate to an anonymous person. After that, the matching process between the available donors and patients on the waiting list is performed by the organ transplantation organizer. As a result, a ranked list is generated as an output and provided to the transplantation surgeons. Next, the transplant surgeon decides whether the organ is appropriate for the patient based on various considerations, such as the donor's medical records and the current health of the prospective recipient. Later, when a transplant surgeon accepts the donated organ, the donor's surgeon is notified to remove the donated organ. Finally, the donated organ is transported to the patient's hospital and received by the transplant surgeon. However, suppose the situation is for a live donor and it has been planned to donate to a known person by name. In that case, the data will go directly to the transplant surgeon to start the surgery of removing and transplanting the donated organ [6], [7].

In the past, when a patient died or was near death, the organ procurement organization and hospital worked together to do an initial medical test to decide if the patient could be an organ donor. This call takes around 15 minutes, and only 6% of these calls result in possible organ donors' being identified. Over the years, this phone call has been replaced by an instant message generated by central computer systems that store all the data required for this process [8]. However, the core issue with this strategy is that the security and validity of such data are entirely dependent on the transplantation centers' ability to keep their systems secure and identify potential harm to donors and recipients. The accuracy of the wait-list data is largely dependent on people's faith and trust in these centers' ability to keep it secure from hackers and fraudulent employees [9]. Moreover, transparency is another challenge affecting the success of the organ donation process. According to World Health Organization (WHO) reports, up to 10% of transplanted organs may have been obtained unethically via organ trafficking, but the exact numbers are unknown [10]. The lack of transparency in the current system among participants leads to illegal organ trade and purchases and medical professionals engaging in unethical practices [11]. Moreover, there are hospitals that take advantage of the patient's need for the organ and offer the opportunity to transfer the organ to those who pay a higher amount to the hospital while ignoring the patient with the highest priority on the waiting list [12], [13]. In addition, current transplant systems are also frequently slow, which is unacceptable in such a critical and life-threatening scenario. Such systems are hardly up to date with the minimum security standards. So far, there has recently been a surge in security breaches affecting user privacy and system integrity. In general, modern systems manage data through the use of standard databases; however, most hospitals, health ministries, and other medical facilities lack a standardized data communication
system [1].

1.1 SCOPE OF THE PROJECT

The scope of Blockchain-based management for organ donation and transplantation is vast and holds significant potential to address various challenges in the current organ donation and transplantation processes. Blockchain technology can bring transparency, security, efficiency, and traceability to the organ donation ecosystem.

Here are some key aspects of its scope:

Transparency and Trust: Blockchain's distributed and immutable ledger ensures transparency in the organ donation and transplantation process. All transactions and changes related to organ donation, allocation, and transplantation can be recorded on the blockchain, providing a transparent view of the entire process. This transparency helps build trust among stakeholders, including donors, recipients, healthcare providers, and regulatory authorities.

Organ Tracking and Traceability: With blockchain, the complete journey of an organ, from the time of donation to transplantation, can be tracked and recorded. This feature ensures that all necessary information about the organ's origin, transportation, and handling is accessible to authorized parties, reducing the risk of organ trafficking and ensuring the quality and safety of organs used for transplantation.

Data Security and Privacy: Blockchain's cryptographic features and decentralized nature make it highly secure against unauthorized access and tampering. Medical data related to organ donors and recipients can be stored on the blockchain, ensuring privacy and compliance with data protection regulations.

Decentralization and Accessibility: Blockchain networks can be decentralized, allowing multiple healthcare organizations, hospitals, and transplantation centers to participate and share data securely. This decentralization leads to better accessibility to information, faster matching of donors and recipients, and more efficient organ allocation processes.

Smart Contracts for Automation: Smart contracts are self-executing contracts with predefined rules and conditions. In the context of organ donation, smart contracts can automate various processes, such as organ matching, allocation, and consent verification. This automation can lead to faster organ transplantation processes and reduce administrative overhead.

Incentive Mechanisms: Blockchain-based systems can incorporate incentive mechanisms to encourage organ donation. Donors or their families can be rewarded with tokens or other benefits through smart contracts, promoting more active participation in organ donation programs.

Global Collaboration: Blockchain can facilitate secure data sharing and collaboration among healthcare organizations, research institutions, and regulatory bodies worldwide. This global collaboration can lead to a more efficient exchange of knowledge, best practices, and research findings related to organ donation and transplantation.

Medical Records Management: Blockchain can be utilized to securely store and share medical records related to organ donors and recipients. Having access to accurate and up-to-date medical records can expedite the matching process and ensure better outcomes for transplantation procedures.

Despite its significant potential, there are challenges to implementing blockchain-based solutions for organ donation and transplantation. These include regulatory compliance, standardization of data formats, interoperability with existing healthcare systems, and the need for broad adoption across the medical community.

Overall, Blockchain-based management for organ donation and transplantation has the potential to revolutionize the field by enhancing transparency, security, and efficiency, ultimately leading to increased organ availability and improved patient outcomes

1.2 OBJECTIVE

The primary objective of Blockchain-based management for organ donation and transplantation is to enhance the efficiency, transparency, security, and traceability of the organ donation process. By leveraging blockchain technology, the objective is to overcome various challenges and improve the overall organ donation and transplantation ecosystem.

Here are the main objectives:

Improve Transparency: Blockchain's distributed ledger ensures that all transactions related to organ donation, allocation, and transplantation are recorded in a transparent and tamper-resistant manner. This transparency helps build trust among stakeholders, including donors, recipients, medical professionals, and regulatory authorities.

Ensure Data Security and Privacy: Blockchain employs cryptographic techniques to secure sensitive medical data. Medical records related to organ donors and recipients can be stored on the blockchain in a privacy-preserving manner, ensuring that only authorized parties can access the data.

Enable Traceability and Accountability: Blockchain enables the complete traceability of organ donation processes, from the time of donation to transplantation. This traceability helps ensure that the organ's journey is recorded and that the organs are used for legitimate and ethical purposes, reducing the risk of organ trafficking.

Facilitate Efficient Organ Matching and Allocation: With blockchain, the process of matching organ donors with suitable recipients can be streamlined. Smart contracts can be used to automate the organ allocation process, ensuring that the right organs reach the right recipients in a timely manner.

Encourage Global Collaboration: Blockchain technology enables secure data sharing and collaboration among healthcare organizations, research institutions, and regulatory bodies worldwide. This global collaboration can lead to the exchange of best practices, knowledge, and research findings related to organ donation and transplantation.

Enhance Donor Confidence and Participation: By providing a transparent and secure system, blockchain-based management can enhance donor confidence in the organ donation process. Smart contracts can also incentivize organ donation, encouraging more individuals to participate in organ donation programs.

Improve Organ Transplant Outcomes: Faster and more efficient organ allocation processes facilitated by blockchain can lead to improved transplant outcomes for recipients. By reducing administrative overhead and increasing transparency, blockchain can contribute to better patient care.

Reduce Medical Errors: Blockchain's decentralized nature and consensus mechanisms can help prevent single points of failure and reduce the risk of medical errors in organ donation and transplantation.

Ensure Compliance with Regulations: Blockchain-based management systems can be designed to comply with relevant regulations and standards in the healthcare industry. This compliance helps ensure that the organ donation process adheres to legal and ethical requirements.

By achieving these objectives, Blockchain-based management for organ donation and transplantation aims to create a more robust, ethical, and accessible organ donation ecosystem, ultimately saving more lives and improving the quality of life for those in need of organ transplants.

1.3 DESCTIPTION OF THE PROJECT

Blockchain-based management for organ donation and transplantation is a technological approach that leverages blockchain technology to enhance the efficiency, transparency, security, and traceability of the entire organ donation process, from organ registration to transplantation. The system aims to address several challenges faced by the traditional organ donation and transplantation ecosystem, including the need for transparency, trust, and accountability.

Here's a description of how blockchain-based management for organ donation and transplantation works:

Donor Registration: Organ donors can register their willingness to donate organs on the blockchain platform. The donor's information, medical history, and consent to donation are recorded in a secure and immutable manner. This data is accessible only to authorized healthcare professionals and organizations.

Organ Tracking and Traceability: Once an organ donor is registered, the entire journey of the organ is recorded on the blockchain. This includes the details of the organ's procurement, transportation, preservation, and transplantation. Each transaction is timestamped and linked to the previous one, creating a transparent and traceable trail of the organ's history.

Smart Contracts for Organ Allocation: Smart contracts are programmable self-executing contracts that run on the blockchain. In the context of organ donation, smart contracts can be used to automate the organ matching and allocation process. When a suitable recipient is identified, the smart contract automatically triggers the allocation process, ensuring efficient and timely organ transplantation.

Data Privacy and Security: Medical data related to organ donors and recipients are stored in a decentralized manner on the blockchain. The data is encrypted and accessible only to authorized parties with appropriate keys, ensuring patient privacy and data security.

Consent Verification: Consent verification is a critical aspect of organ donation. Blockchain-based management allows for efficient verification of the donor's consent status. Donor consent is securely recorded on the blockchain, ensuring that the organs are used for transplantation only with explicit consent.

Decentralization and Global Collaboration: Blockchain networks are decentralized, allowing multiple healthcare organizations, hospitals, and transplantation centers to participate and share data securely. This decentralization fosters global collaboration, leading to a more efficient exchange of knowledge and best practices.

Enhanced Donor Confidence: The transparency and traceability provided by blockchain-based management can enhance donor confidence in the organ donation process. Donors can be assured that their organs are used ethically and reach the intended recipients.

Prevention of Organ Trafficking: The immutable nature of blockchain prevents unauthorized alterations to the organ data, reducing the risk of organ trafficking and ensuring the legitimacy of organ transactions.

Auditability and Compliance: Blockchain-based management provides an audit trail of all organ-related activities, ensuring compliance with regulatory requirements and improving accountability within the organ donation and transplantation ecosystem.

Overall, the use of blockchain in organ donation and transplantation can revolutionize the field by offering a secure, efficient, and transparent platform that fosters trust among all stakeholders involved in the process. By addressing the challenges of the current system, blockchain-based management has the potential to save more lives and improve patient outcomes in organ transplantation.

2. LITERARTURE SURVEY

As of my last update in September 2021, there were no specific literature surveys dedicated solely to blockchain-based management for organ donation and transplantation. However, there were research papers, articles, and studies exploring the potential of blockchain technology in the healthcare domain and some mentioning its applications in organ donation and transplantation. Keep in mind that the landscape of research and literature is continuously evolving, so there may be new publications after my last update.

To find the most recent and comprehensive literature on this topic, I recommend conducting a search on academic databases, such as Google Scholar, PubMed, IEEE Xplore, or ACM Digital Library, using relevant keywords like "blockchain," "organ donation," "transplantation," "healthcare," and "decentralized systems."

Here are a few papers and articles related to blockchain and organ transplantation:

Khezr, S., Moniruzzaman, M., & Mahmud, M. S. (2020). A comprehensive review on blockchain technology: Insights from a healthcare perspective. Journal of Medical Systems, 44(7), 1-23.

Rahaman, M. S., Jayakumar, M., & Duraiswamy, K. (2019). Blockchain and organ transplantation. Journal of Clinical and Diagnostic Research, 13(5), VL01.

Barman, S., Das, A. K., De, D., Chattopadhyay, S., & Das, A. K. (2019). A blockchain framework for efficient organ transplantation. Proceedings of the 11th International Conference on Advanced Computing (ICoAC), 16-20.

Azaria, A., Ekblaw, A., Vieira, T., & Lippman, A. (2016). MedRec: Using blockchain for medical data access and permission management. Proceedings of the 2nd International Conference on Open and Big Data (OBD), 25-30.

Ding, S., & Sheng, Q. Z. (2018). A Decentralized Blockchain-Based Organ Transplant Management System. In IEEE International Conference on Smart Cloud (SmartCloud), 213-218.

Remember to assess the relevance, credibility, and applicability of the information in each paper or article for your specific research context. Additionally, explore the citations and references within those publications for further relevant research on the topic.

3. SYSTEM ANALYSIS

3.1 EXISTING SYSTEM:

The authors in [17] developed a multi-agent software platform to represent the information workflow model among donor hospitals, regulators, and recipient hospitals. This platform optimizes the pre-transplantation tasks, which can improve the process efficiency. In addition, it allows storing potential donor information and improves direct communication among all participants in the organ transplantation process. An information workflow was simulated using the developed platform, and it was estimated that the saved time might be between three to five hours.

The TransNet in [18] is a system using scanning technology for barcodes at the point of organ recovery to assist in labeling, packaging, and tracking organs and other biological materials for transplantation. It involves supplementing the labeling system with an application developed and a portable barcode printer corresponding with DonorNet. During organ recovery, procurement coordinators will use the operating room's system to print labels and scan all organs to be transported. Similarly, many supply chain management solutions have relied on barcodes, RFID tags, and Electronic Product Codes (EPC) for identifying and sharing product information to facilitate the tracking of items through various phases [19].

Finally, the authors in [20] proposed a manageable mechanism, MIN, for the online matching of deceased organs to donors to improve efficiency and fairness in selecting patients within the current system in Australia. The MIN mechanism simply designates an arriving organ to a patient that minimizes |KDPI-EPTS|, tie-breaking by time on the waiting list and later randomly. The Kidney Donor Patient Inde (KDPI) estimates the quality of the organ. On the other hand, the Expected Post-Transplant Survival Score (EPTS) measures the life quality of the recipient after the transplant. After testing, the results showed that the MIN mechanism outperforms the current mechanism under consideration by the Organ and Tissue Authority in Australia.

The authors in [23] proposed an organ donation decentralized app using blockchain technology. Patients use a web application to register their information, including their medical ID, organ type, blood type, and state. The system would operate on a first-in, first-out (FIFO) approach, with the exception of a patient being in a critical state. It offered better security, added transparency, and a much faster system. However, it should be modified when used in different regions according to their regulations and needs. Similarly, the authors in [24], developed a web-based application using FIFO to choose an organ donor for each actual patient seeking a transplant, and in the case of an emergency, that patient is given priority. Furthermore, an organ donation and transplantation application utilizing blockchain has been proposed in [12], where the registered hospital accepts the registered donors and registers the recipients to match them with a suitable donor based on the request.

Moreover, in [25], a use case for blockchain in organ donation has been developed. Simply, the process begins with the donor signing a smart contract for organ donation and the patient ling a transplant request. Both papers are verified and hashed by a registered doctor or nurse, who then creates a verified mismatching pair and announces it over the network. The network finds a match and sends it to a doctor for approval. If a match is found, the doctor approves, and the next step is for the doctor to generate a hash. If the doctor generates a hash, the verified matched pair becomes part of the blockchain. Finally, doctors and healthcare professionals
are given all the information they need to prepare for the logistics of the surgery.

Disadvantage:

- ❖ The system is not implemented blockchain based organ donation which leads less security and less communication between hospitals and donors..
- ❖ The system is not implemented an auto-matching process between the donor and recipient through a smart contract based on certain criteria.

3.2 PROPOSED SYSTEM:

- ❖ The system proposes a private Ethereum blockchain-based solution that ensures organ donation and transplantation management in a manner that is decentralized, secure, reliable, traceable, auditable, and trustworthy.
- ❖ The system develops smart contracts that register actors and ensure data provenance through producing events for all the necessary actions that occur during the organ donation and transplantation stages. The smart contracts code is made publicly available on Github.1

- The system develops an auto-matching process between the donor and recipient through a smart contract based on certain criteria.
- The system presents six algorithms along with their full implementation, testing, and validation details.
- The system conducts security analysis to determine that the proposed solution is secure against common security attacks and vulnerabilities. We compare our solution with the existing solutions to show its novelty. Our proposed solution is general and may be easily adjusted to meet the needs of a variety of related applications.

Advantage:

- The system is implemented an organ donation based on blockchain techniques which is more fast and secure.
- In the proposed system, the system is implemented an automatic process of human organ donation.

3.3 HARDWARE & SOFTWARE REQUIREMENTS:

Minimum hardware requirements:

System	: Pentium –IV
Hard Disk	: 40 GB.
Ram	: 4 GB

Minimum software requirements:

Operating System	: Windows XP.
Coding Language	: Java/J2EE(JSP,Servlet)
Front End	: J2EE
Back End	: MySQL

4. SYSTEM DESIGN

4.1 INPUT DESIGN:

In an information system, input is the raw data that is processed to produce output. During the input design, the developers must consider the input devices such as PC, MICR, OMR, etc. Therefore, the quality of system input determines the quality of system output. Well-designed input forms and screens have following properties −

- It should serve specific purpose effectively such as storing, recording, and retrieving
- the information.
- It ensures proper completion with accuracy.
- It should be easy to fill and straightforward.
- It should focus on user’s attention, consistency, and simplicity.
- All these objectives are obtained using the knowledge of basic design principles
- regarding −
    - What are the inputs needed for the system?
    - How end users respond to different elements of forms and screens.

Objectives for Input Design:

The objectives of input design are −

- To design data entry and input procedures
- To reduce input volume
- To design source documents for data capture or devise other data capture methods
- To design input data records, data entry screens, user interface screens, etc.
- To use validation checks and develop effective input controls.

4.2 OUTPUT DESIGN:

The design of output is the most important task of any system. During output design, developers identify the type of outputs needed, and consider the necessary output controls and prototype report layouts.

Objectives of Output Design:

- To develop output design that serves the intended purpose and eliminates the production of unwanted output.
- To develop the output design that meets the end user’s requirements.

- To deliver the appropriate quantity of output.
- To form the output in appropriate format and direct it to the right person.
- To make the output available on time for making good decisions.

4.3 UML DIAGRAMS

UML stands for Unified Modeling Language. UML is a standardized general-purpose modeling language in the field of object-oriented software engineering. The standard is managed, and was created by, the Object Management Group.

The goal is for UML to become a common language for creating models of object oriented computer software. In its current form UML is comprised of two major components: a Meta-model and a notation. In the future, some form of method or process may also be added to; or associated with, UML.

The Unified Modeling Language is a standard language for specifying, Visualization, Constructing and documenting the artifacts of software system, as well as for business modeling and other non-software systems.

The UML represents a collection of best engineering practices that have proven successful in the modeling of large and complex systems.

The UML is a very important part of developing objects oriented software and the software development process. The UML uses mostly graphical notations to express the design of software projects.

GOALS:

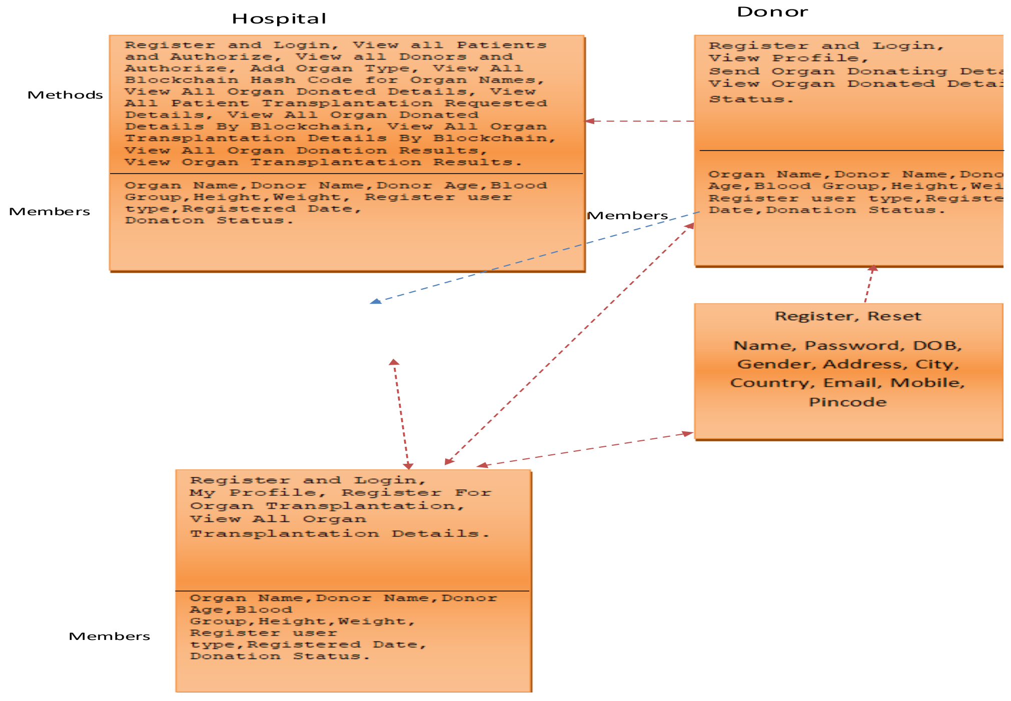
The Primary goals in the design of the UML are as follows:

1. Provide users a ready-to-use, expressive visual modeling Language so that they can develop and exchange meaningful models.
2. Provide extendibility and specialization mechanisms to extend the core concepts.
3. Be independent of particular programming languages and development process.
4. Provide a formal basis for understanding the modeling language.
5. Encourage the growth of OO tools market.
6. Support higher level development concepts such as collaborations, frameworks, patterns and components.
7. Integrate best practices.

USE CASE DIAGRAM:

A use case diagram in the Unified Modeling Language (UML) is a type of behavioral diagram defined by and created from a Use-case analysis. Its purpose is to present a graphical overview of the functionality provided by a system in terms of actors, their goals (represented as use cases), and any dependencies between those use cases.

USE CASE:

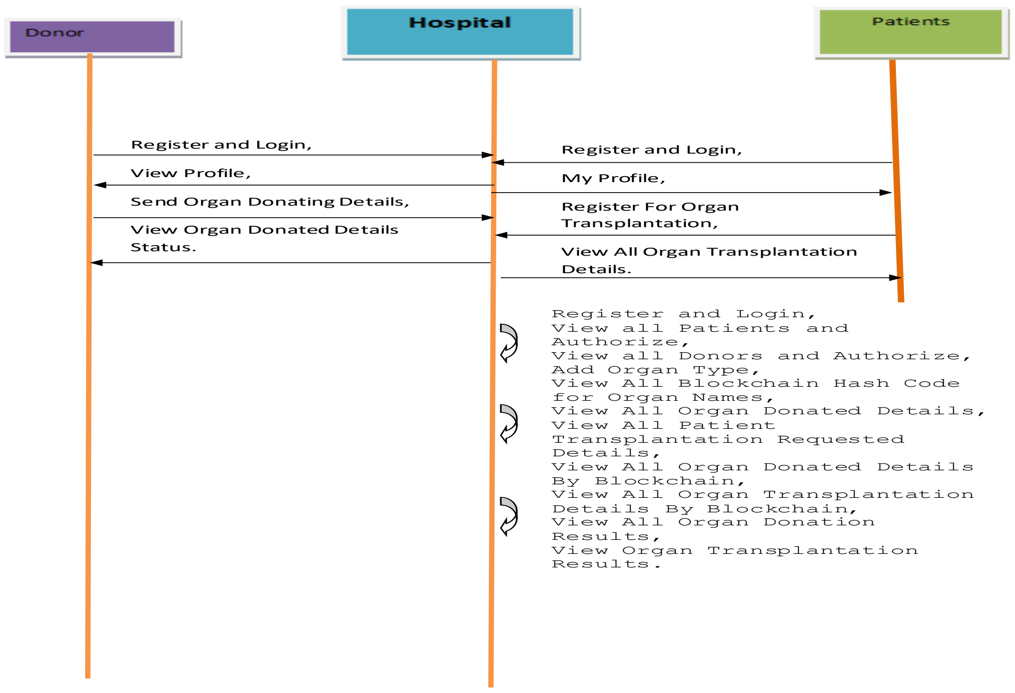

Hospital
Methods
Register and Login, View all Patients and Authorize, View all Donors and Authorize, Add Organ Type, View All Blockchain Hash Code for Organ Names, View All Organ Donated Details, View All Patient Transplantation Requested Details, View All Organ Donated Details By Blockchain, View All Organ Transplantation Details By Blockchain, View All Organ Donation Results, View Organ Transplantation Results.
Members
Organ Name,Donor Name,Donor Age,Blood Group,Height,Weight, Register user type,Registered Date, Donaton Status.
Donor
Register and Login, View Profile, Send Organ Donating Deta View Organ Donated Detai Status.
Members
Organ Name,Donor Name,Dono Age,Blood Group,Height,Wei Register user type,Registe Date,Donation Status.
Register, Reset
Name, Password, DOB, Gender, Address, City, Country, Email, Mobile, Pincode
Register and Login, My Profile, Register For Organ Transplantation, View All Organ Transplantation Details.
Members
Organ Name,Donor Name,Donor Age,Blood Group,Height,Weight, Register user type,Registered Date, Donation Status.

CLASS DIAGRAM:

In software engineering, a class diagram in the Unified Modeling Language (UML) is a type of static structure diagram that describes the structure of a system by showing the system's classes, their attributes, operations (or methods),

SEQUENCE DIAGRAM:

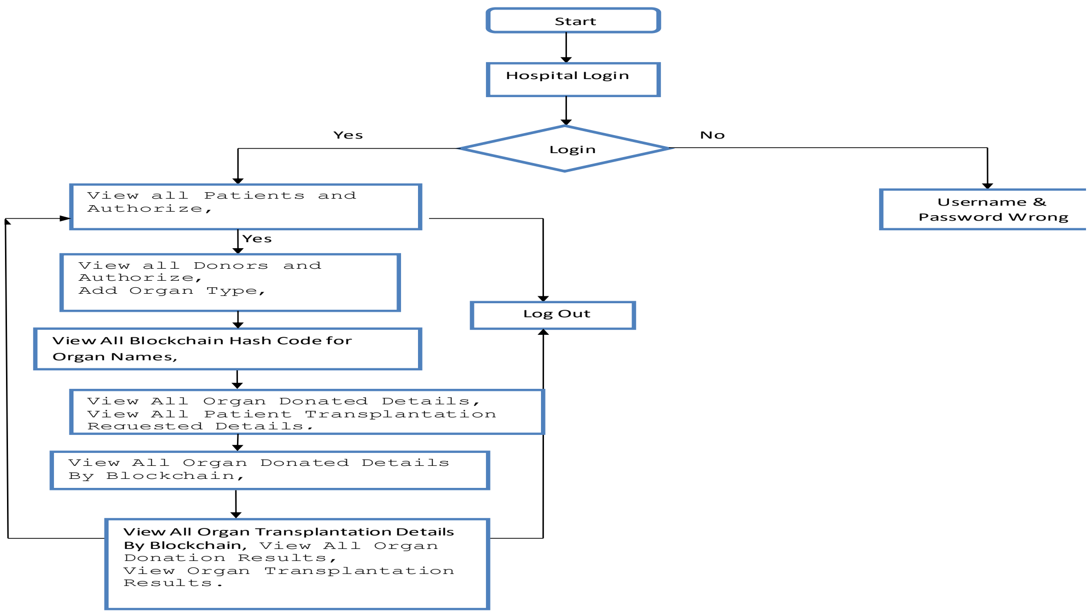
A sequence diagram in Unified Modeling Language (UML) is a kind of interaction diagram that shows how processes operate with one another and in what order. It is a construct of a Message Sequence Chart. Sequence diagrams are sometimes called event diagrams, event scenarios, and timing diagrams.


Donor
Hospital
Patients
Register and Login,
View Profile,
Send Organ Donating Details,
View Organ Donated Details Status.
Register and Login,
My Profile,
Register For Organ Transplantation,
View All Organ Transplantation Details.
Register and Login,
View all Patients and Authorize,
View all Donors and Authorize,
Add Organ Type,
View All Blockchain Hash Code for Organ Names,
View All Organ Donated Details,
View All Patient Transplantation Requested Details,
View All Organ Donated Details By Blockchain,
View All Organ Transplantation Details By Blockchain,
View All Organ Donation Results,
View Organ Transplantation Results.

ACTIVITY DIAGRAM:

Activity diagrams are graphical representations of workflows of stepwise activities and actions with support for choice, iteration and concurrency. In the Unified Modeling Language, activity diagrams can be used to describe the business and operational step-by-step workflows of components in a system. An activity diagram shows the overall flow of control.

Flow Chart 2: Hospital


Start
Hospital Login
Login
Yes
No
Username & Password Wrong
View all Patients and Authorize,
Yes
View all Donors and Authorize, Add Organ Type,
View All Blockchain Hash Code for Organ Names,
View All Organ Donated Details, View All Patient Transplantation Requested Details,
View All Organ Donated Details By Blockchain,
View All Organ Transplantation Details By Blockchain, View All Organ Donation Results, View Organ Transplantation Results.
Log Out

5. IMPLEMENTATION

5.1 DESCRIPTION OF MODULES:

- **Donors**
- **Patients**
- **Hospital**

Donors

In this module, the Donor will register and login then uploads their organ donor data to the Hospital and will do the following operations such as View Profile, Send Organ Donating Details, View Organ Donated Details Status.

Patients

In this module, patients logs in by using his/her user name and password. After Login User will do some operations such as My Profile, Register For Organ Transplantation, View All Organ Transplantation Details.

Hospital

The Hospital manages Hospital records to provide organ storage service for donation and transplantation and also performs the following operations such as View all Patients and Authorize, View all Donors and Authorize,Add Organ Type, View All Blockchain Hash Code for Organ Names,View All Organ Donated Details,View All Patient Transplantation Requested Details,View All Organ Donated Details By Blockchain,View All Organ Transplantation Details By Blockchain, View All Organ Donation Results,View Organ Transplantation Results.

Implementing a blockchain-based management system for organ donation and transplantation can bring numerous benefits, including increased transparency, data security, and efficiency in the organ allocation process. Below are some potential modules that could be incorporated into such a system:

Identity Verification Module:

This module ensures that both organ donors and recipients are verified, authenticated, and have their identities securely stored on the blockchain. It could involve integrating biometric data, government-issued IDs, and medical records to prevent fraudulent activities and ensure the accuracy of patient information.

Donor Registration Module:

This module facilitates the registration of potential organ donors onto the blockchain. Donors could express their consent for organ donation, provide relevant medical history, and specify any preferences regarding the allocation of their organs.

Recipient Registration Module:

Similarly, this module allows patients in need of organ transplantation to register on the blockchain. It captures essential medical data and matches recipients with suitable donors based on compatibility, medical urgency, and other relevant criteria.

Organ Matching Module:

The organ matching module utilizes smart contracts to match registered donors with suitable recipients based on factors such as blood type, tissue compatibility, geographic location, and medical urgency. This automated process ensures fair and efficient allocation of organs.

Organ Transportation Module:

This module tracks the transportation of donated organs from the donor's location to the recipient's hospital. It ensures the organs are handled properly and delivered in a timely manner, minimizing the risk of organ rejection due to delays.

Consent Tracking Module:

Consent for organ donation is a critical aspect of the process. This module records and manages the consent status of both donors and recipients, providing transparency and accountability throughout the organ allocation process.

Medical Records Module:

The blockchain can securely store and manage the medical records of donors and recipients, enabling authorized medical professionals to access relevant health information in real-time, leading to better decision-making and more accurate matches.

Audit and Compliance Module:

To meet regulatory requirements and ensure the system's integrity, an audit and compliance module can be implemented. It tracks all activities on the blockchain, allowing for transparent audits and ensuring adherence to applicable laws and guidelines.

Post-Transplantation Monitoring Module:
This module enables post-transplantation monitoring, recording the progress and outcomes of organ transplantations. It helps evaluate the success rates of transplants and enhances the system's overall performance.

Reporting and Analytics Module:
A reporting and analytics module can provide insights into the effectiveness and efficiency of the organ donation and transplantation process. By analyzing the data collected on the blockchain, stakeholders can make informed decisions to improve the system continuously.

It's important to note that building a blockchain-based organ donation and transplantation management system involves careful consideration of legal, ethical, and technical aspects. Collaborating with healthcare professionals, legal experts, and blockchain developers will be crucial to designing a system that is secure, effective, and ethically sound.

5.2 ALGORITHM:

Ensuring scalability and managing complexity of the models is also a challenge associated with any modelling approach. Indeed, our models are enriched with fairly complex behavioral information defined in terms of state invariants.
To reduce the complexity, we do not require the system analyst to model the entire functional and security behavior of private cloud in one diagram. Our approach can be used to represent and validate only those scenarios that are considered to be critical by the experts.
The proposed semi-automated approach aimed at helping the cloud developers and security experts to identify the security loopholes in the implementation by relying on modelling rather than manual code inspection or testing.

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

You take the following aes steps of encryption for a 128-bit block:

Derive the set of round keys from the cipher key.

Initialize the state array with the block data (plaintext).

Add the initial round key to the starting state array.

Perform nine rounds of state manipulation.

Perform the tenth and final round of state manipulation.

Copy the final state array out as the encrypted data (ciphertext).

The reason that the rounds have been listed as "nine followed by a final tenth round" is because the tenth round involves a slightly different manipulation from the others.

The block to be encrypted is just a sequence of 128 bits. AES works with byte quantities so we first convert the 128 bits into 16 bytes. We say "convert," but, in reality, it is almost certainly stored this way already. Operations in RSN/AES are performed on a two-dimensional byte array of four rows and four columns. At the start of the encryption, the 16 bytes of data.

FEASIBILITY STUDY

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

Three key considerations involved in the feasibility analysis are

- ECONOMICAL FEASIBILITY
- TECHNICAL FEASIBILITY
- SOCIAL FEASIBILITY

Economic feasibility:

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

Technical feasibility:

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

Social feasibility:

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

5.3 OVER VIEW OF IMPLEMENTATION OF LANGUAGE:

Java: Java is a set of computer software and specifications developed by Sun Microsystems, which was later acquired by the Oracle Corporation, that provides a system for developing application software and deploying it in a cross platform computing environment. Java that is object oriented programming is used in a wide variety of computing platforms from embedded devices and mobile phones to enterprise servers and supercomputers. Java applets, which are less common than standalone Java applications, run in secure, sandboxed environments to provide many features of native applications and can be embedded in HTML pages. Java can run on many different operating systems. This makes Java platform independent. Java does this by making the Java compiler turn code into Java byte code instead of machine code. This means that when the program is executed, the Java Virtual Machine interprets the byte code and translates it into machine code.

Features of The Language Used:

In my project, I have chosen Java language for developing the code.

About Java

Initially the language was called as "oak" but it was renamed as "Java" in 1995. The primary motivation of this language was the need for a platform independent (i.e., architecture neutral) language that could be used to create software to be embedded in various consumer electronic devices.

- Java is a programmer's language.
- Java is cohesive and consistent.
- Except for those constraints imposed by the Internet environment, Java gives the programmer, full control.

Finally, Java is to Internet programming where C was to system programming. Importance of Java to the Internet:

Java has had a profound effect on the Internet. This is because; Java expands the Universe of objects that can move about freely in Cyberspace. In a network, two categories of objects are transmitted between the Server and the Personal computer.

They are: Passive information and Dynamic active programs. The Dynamic, Self executing programs cause serious problems in the areas of Security and probability. But, Java addresses those concerns and by doing so, has opened the door to an exciting new form of program called the Applet.

Java can be used to create two types of programs:

Applications and Applets: An application is a program that runs on our Computer under the operating system of that computer. It is more or less like one creating using C or C++. Java's ability to create Applets makes it important. An Applet is an application designed to be transmitted over the Internet and executed by a Java – compatible web browser. An applet is actually a tiny Java program, dynamically downloaded across the network, just like an image. But the difference is, it is an intelligent program, not just a media file. It can react to the user input and dynamically change. Features Of Java

Security:

Every time you that you download a "normal" program, you are risking a viral infection. Prior to Java, most users did not download executable programs frequently, and those who did scanned them for viruses prior to execution. Most users still worried about the possibility of infecting their systems with a virus. In addition, another type of malicious program exists that must be guarded against. This type of program can gather private information, such as credit card numbers, bank account balances, and passwords. Java answers both these concerns by providing a "firewall" between a network application and your computer.

When you use a Java-compatible Web browser, you can safely download Java applets without fear of virus infection or malicious intent.

Portability:

For programs to be dynamically downloaded to all the various types of platforms connected to the Internet, some means of generating portable executable code is needed .As you will see, the same mechanism that helps ensure security also helps create portability. Indeed, Java's solution to these two problems is both elegant and efficient.

The Byte code:

The key that allows the Java to solve the security and portability problems is that the output of Java compiler is Byte code. Byte code is a highly optimized set of instructions designed to be executed by the Java run-time system, which is called the Java Virtual Machine (JVM). That is, in its standard form, the JVM is an interpreter for byte code.

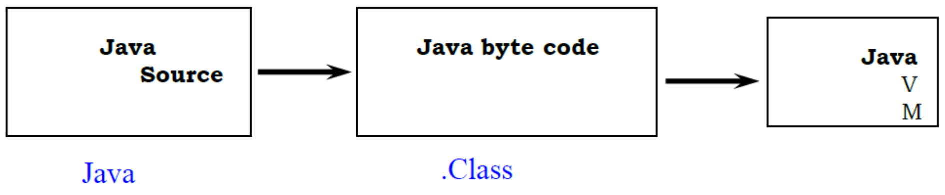
Translating a Java program into byte code helps makes it much easier to run a program in a wide variety of environments. The reason is, once the run-time package exists for a given system, any Java program can run on it.

Although Java was designed for interpretation, there is technically nothing about Java that prevents on-the-fly compilation of byte code into native code. Sun has just completed its Just In Time (JIT) compiler for byte code. When the JIT compiler is a part of JVM, it compiles byte code into executable code in real time, on a piece-by piece, demand basis. It is not possible to compile an entire Java program into executable code all at once, because Java performs various run-time checks that can be done only at run time. The JIT compiles code, as it is needed, during execution.

Java, Virtual Machine (JVM):

Beyond the language, there is the Java virtual machine. The Java virtual machine is an important element of the Java technology. The virtual machine can be embedded within a web browser or an operating system. Once a piece of Java code is loaded onto a machine, it is verified. As part of the loading process, a class loader is invoked and does byte code verification makes sure that the code that's has been generated by the compiler will not corrupt the machine that it's loaded on. Byte code verification takes place at the end of the compilation process to make sure that is all accurate and correct. So byte code verification is integral to the compiling and executing of Java code.

Overall Description



Picture showing the development process of JAVA Program

Java programming uses to produce byte codes and executes them. The first box indicates that the Java source code is located in a. Java file that is processed with a Java compiler called javac. The Java compiler produces a file called a. class file, which contains the byte code. The. Class file is then loaded across the network or loaded locally on your machine into the execution environment is the Java virtual machine, which interprets and executes the byte code.

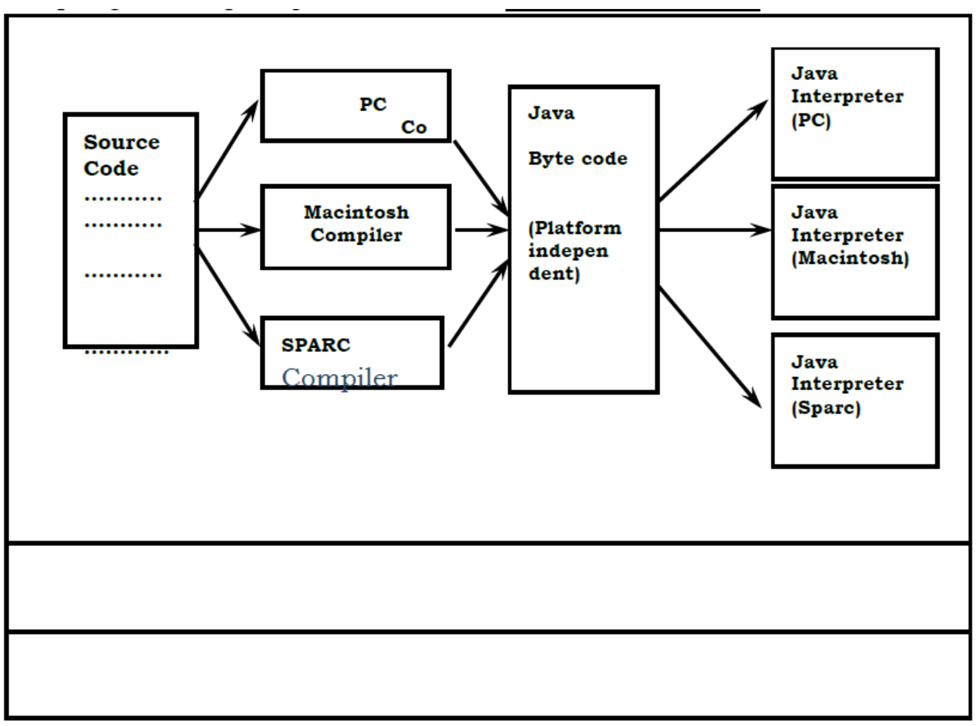
Java Architecture:

Java architecture provides a portable, robust, high performing environment for development. Java provides portability by compiling the byte codes for the Java Virtual Machine, which is then interpreted on each platform by the run-time environment. Java is a dynamic system, able to load code when needed from a machine in the same room or across the planet.

Compilation of code:

When you compile the code, the Java compiler creates machine code (called byte code) for a hypothetical machine called Java Virtual Machine (JVM). The JVM is supposed to execute the byte code. The JVM is created for overcoming the issue of portability. The code is written and compiled for one machine and interpreted on all machines. This machine is called Java Virtual Machine.

Compiling and interpreting Java Source Code FIG. 5.4.1



During run-time the Java interpreter tricks the byte code file into thinking that it is running on a Java Virtual Machine. In reality this could be a Intel Pentium Windows 95 or Sun SARC station running Solaris or Apple Macintosh running system and all could receive code from any computer through Internet and run the Applets. Simple:

Java was designed to be easy for the Professional programmer to learn and to use effectively. If you are an experienced C++ programmer, learning Java will be even easier. Because Java inherits the C/C++ syntax and many of the object oriented features of C++. Most of the confusing concepts from C++ are either left out of Java or implemented in a cleaner, more approachable manner. In Java there are a small number of clearly defined ways to accomplish a given task.

Object-Oriented:

Java was not designed to be source-code compatible with any other language. This allowed the Java team the freedom to design with a blank slate. One outcome of this was a clean usable, pragmatic approach to objects. The object model in Java is simple and easy to extend, while simple types, such as integers, are kept as high-performance non-objects.

Robust:

The multi-platform environment of the Web places extraordinary demands on a program, because the program must execute reliably in a variety of systems. The ability to create robust programs was given a high priority in the design of Java. Java is strictly typed language; it checks your code at compile time and run time. Java virtually eliminates the problems of memory management and deallocation, which is completely automatic. In a well-written Java program, all run time errors can –and should –be managed by your program.

JAVASCRIPT:

JavaScript is a script-based programming language that was developed by Netscape Communication Corporation. JavaScript was originally called Live Script and renamed as JavaScript to indicate its relationship with Java. JavaScript supports the development of both client and server components of Web-based applications. On the client side, it can be used to write programs that are executed by a Web browser within the context of a Web page. On the server side, it can be used to write Web server programs that can process information submitted by a Web browser and then updates the browser's display accordingly

Even though JavaScript supports both client and server Web programming, we prefer JavaScript at Client side programming since most of the browsers supports it. JavaScript is almost as easy to learn as HTML, and JavaScript statements can be included in HTML documents by enclosing the statements between a pair of scripting tags

<SCRIPTS>..</SCRIPT>.

<SCRIPT LANGUAGE = "JavaScript">

JavaScript statements

</SCRIPT>

Here are a few things we can do with JavaScript :

- Validate the contents of a form and make calculations.
- Add scrolling or changing messages to the Browser's status line.
- Animate images or rotate images that change when we move the mouse over them.
- Detect the browser in use and display different content for different browsers.
- Detect installed plug-ins and notify the user if a plug-in is required.

We can do much more with JavaScript, including creating entire application.

J a v a v s j a v a s c r i p t :

JavaScript and Java are entirely different languages. A few of the most glaring differences are:

- Java applets are generally displayed in a box within the web document; JavaScript can affect any part of the Web document itself.
- While JavaScript is best suited to simple applications and adding interactive features to Web pages; Java can be used for incredibly complex applications.

There are many other differences but the important thing to remember is that JavaScript and Java are separate languages. They are both useful for different things; in fact they can be used together to combine their advantages.

A D V A N T A G E S

- JavaScript can be used for Sever-side and Client-side scripting.
- It is more flexible than VBScript.
- JavaScript is the default scripting languages at Client-side since all the browsers supports it.

Hyper Text Markup Language

Hypertext Markup Language (HTML), the languages of the World Wide Web (WWW), allows users to produces Web pages that include text, graphics and pointer to other Web pages (Hyperlinks).

HTML is not a programming language but it is an application of ISO Standard 8879, SGML (Standard Generalized Markup Language), but specialized to hypertext and adapted to the Web. The idea behind Hypertext is that instead of reading text in rigid linear structure, we can easily jump from one point to another point. We can navigate through the information based on our interest and preference. A markup language is simply a series of elements, each delimited with special characters that define how text or other items enclosed within the elements should be displayed. Hyperlinks are underlined or emphasized works that load to other documents or some portions of the same document.

HTML can be used to display any type of document on the host computer, which can be geographically at a different location. It is a versatile language and can be used on any platform or desktop.

HTML provides tags (special codes) to make the document look attractive. HTML tags are not case-sensitive. Using graphics, fonts, different sizes, color, etc., can enhance the presentation of the document. Anything that is not a tag is part of the document itself.

Basic HTML Tags :

<!-- -->

Specifies comments

<A>……….</A>

Creates hypertext links

<B>……….</B>

Formats text as bold

<BIG>……….</BIG>

Formats text in large font.

<BODY>…</BODY>

Contains all tags and text in the HTML document

<CENTER>...</CENTER>

Creates text

<DD>…</DD>

Definition of a term

<DL>...</DL>

Creates definition list

<FONT>…</FONT>

Formats text with a particular font

<FORM>...</FORM>

Encloses a fill-out form

<FRAME>...</FRAME>

Defines a particular frame in a set of frames

<H#>…</H#>

Creates headings of different levels

<HEAD>...</HEAD>

Contains tags that specify information about a document

<HR>...</HR>

Creates a horizontal rule

<HTML>…</HTML>

Contains all other HTML tags

<META>...</META>

Provides meta-information about a document

<SCRIPT>…</SCRIPT>

Contains client-side or server-side script

<TABLE>…</TABLE>

Creates a table

<TD>…</TD>

Indicates table data in a table

<TR>…</TR>

Designates a table row

<TH>…</TH>

Creates a heading in a table

6. TESTING

The purpose of testing is to discover errors. Testing is the process of trying to discover every conceivable fault or weakness in a work product. It provides a way to check the functionality of components, sub- assemblies, assemblies and/or a finished product It is the process of exercising software with the intent of ensuring that the Software system meets its requirements and user expectations and does not fail in an unacceptable manner. There are various types of test. Each test type addresses a specific testing requirement.

6.1 TYPES OF TESTS

Unit testing

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

Integration testing

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

Functional test

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

Functional testing is centered on the following items:

Valid Input	: identified classes of valid input must be accepted.
Invalid Input	: identified classes of invalid input must be rejected.
Functions	: identified functions must be exercised.
Output	: identified classes of application outputs must be exercised.

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

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

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.

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.

Black Box Testing

Black Box Testing is testing the software without any knowledge of the inner workings, structure or language of the module being tested. Black box tests, as most other kinds of tests, must be written from a definitive source document, such as specification or requirements document, such as specification or requirements document. It is a testing in which the software under test is treated, as a black box .you cannot “see” into it. The test provides inputs and responds to outputs without considering how the software works.

Unit Testing:

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

6.2 TEST STRATEGY AND APPROACH

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

Test objectives

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

Features to be tested

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

Integration Testing

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

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

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

Acceptance Testing

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

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

7. RESULT

As of my last update in September 2021, the implementation of blockchain-based management for organ donation and transplantation was still in its early stages, and there were limited real-world deployments. However, research and pilot studies have shown promising results and highlighted the potential benefits of using blockchain in this domain. Here are some potential results and benefits of implementing blockchain-based management for organ donation and transplantation:

Improved Transparency and Traceability: Blockchain's distributed ledger provides a transparent and tamper-resistant record of all organ-related activities, from donor registration to transplantation. This enhanced transparency helps build trust among stakeholders and reduces the risk of fraud or organ trafficking.

Enhanced Data Security and Privacy: Blockchain technology employs cryptographic techniques to secure medical data, ensuring that sensitive information related to organ donors and recipients is stored in a privacy-preserving manner. This can improve data security and patient privacy.

Efficient Organ Allocation: Smart contracts can automate the organ matching and allocation process, leading to faster and more efficient organ transplantation. This can potentially reduce waiting times for patients in need of organ transplants and improve overall patient outcomes.

Decentralization and Global Collaboration: Blockchain networks are decentralized, allowing multiple healthcare organizations, hospitals, and transplantation centers to participate and share data securely. This fosters global collaboration and information exchange, leading to better organ matching and allocation across different regions.

Prevention of Data Manipulation: The immutable nature of blockchain prevents unauthorized alterations to the organ data, ensuring the integrity of the information and reducing the risk of errors or discrepancies in the transplantation process.

Increased Donor Confidence: The transparency and traceability provided by blockchain-based management can increase donor confidence in the organ donation process. Donors can have more assurance that their organs will be used ethically and reach the intended recipients.Streamlined Consent Verification: Blockchain can facilitate efficient consent verification by securely recording donor consent on the blockchain. This helps ensure that organs are used for transplantation only with explicit consent. Potential Cost Savings: By streamlining administrative processes and reducing the need for intermediaries, blockchain-based management can potentially lead to cost savings in the organ donation and transplantation ecosystem.It's important to note that while blockchain technology shows great promise, there are still challenges and considerations, such as regulatory compliance, interoperability with existing healthcare systems, and the need for broad adoption among medical professionals and organizations.

Since the field of blockchain and organ donation is evolving, there might be more recent research and real-world implementations that have provided additional insights and results. To stay up-to-date, it is recommended to explore the latest literature and developments in this area.

8.CONCLUSION

In this paper, we have proposed a private Ethereum blockchain-based solution that manages organ donation and transplantation in a decentralized, accountable, auditable, traceable, secure, and trustworthy manner. We developed smart contracts that ensure the data provenance by recording events automatically. We present six algorithms with their implementation, testing, and validation details. We analyze the security of the proposed solution to guarantee that smart contracts are protected against common attacks and vulnerabilities. We compare our solution to other blockchain-based solutions that are currently available. We discuss how our solution can be customized with minimal effort to meet the needs of other systems experiencing similar problems. In the future, our solution can be improved by developing an end-to end DApp. Furthermore, the smart contracts can be deployed and tested on a real private Ethereum network. Finally, the Quorum platform can provide better confidentiality because transactions among entities can only be viewed by specific participants and nobody else, which is not the case in our solution, where transactions between two participants are viewed by other actors authorized in the private blockchain

9. APPENDIX

The appendixes are not always considered part of the actual Requirements Specification and are not always necessary. They may include Sample input/output formats, descriptions of cost analysis studies, or results of user surveys;
Supporting or background information that can help the readers of the Requirements

9.1 SAMPLE CODE:

```
import java.util.ArrayList;
import java.util.Date;
class Block {
    private int index;
    private String previousHash;
    private String hash;
    private long timestamp;
    private String data;
    public Block(int index, String previousHash, long timestamp, String data) {
        this.index = index;
        this.previousHash = previousHash;
        this.timestamp = timestamp;
        this.data = data;
        this.hash = calculateHash();
    }
    public String calculateHash() {
        // In a real implementation, you would use a cryptographic hash function to
calculate the hash.
        return "hash value"; // Placeholder value for simplicity.
    }

    // Getters and setters
}
class Blockchain {
    private ArrayList<Block> chain;
    public Blockchain() {
        chain = new ArrayList<>();
        // Create the genesis block (first block in the chain).
        Block genesisBlock = new Block(0, "0", new Date().getTime(), "Genesis
Block");
        chain.add(genesisBlock);
    }
    public void addBlock(String data) {
        int index = chain.size();
        String previousHash = chain.get(index - 1).getHash();
        long timestamp = new Date().getTime();
        Block newBlock = new Block(index, previousHash, timestamp, data);
        chain.add(newBlock);
    }
    public boolean isChainValid() {
        for (int i = 1; i < chain.size(); i++) {
            Block currentBlock = chain.get(i);
            Block previousBlock = chain.get(i - 1);

            if (!currentBlock.getHash().equals(currentBlock.calculateHash())) {
                return false;
            }
            if (!currentBlock.getPreviousHash().equals(previousBlock.getHash())) {
                return false;
            }
        }
        return true;
    }
}
public class BlockchainExample {
    public static void main(String[] args) {
        Blockchain blockchain = new Blockchain();
        // Simulate organ donation and transplantation data
        String donorData = "Donor: John Doe, Organ: Kidney";
        String recipientData = "Recipient: Jane Smith, Organ: Kidney";
        // Add blocks to the blockchain
        blockchain.addBlock(donorData);
        blockchain.addBlock(recipientData);
        // Check if the blockchain is valid
        System.out.println("Is blockchain valid? " + blockchain.isChainValid());
        // Print the blockchain
        for (Block block : blockchain.getChain()) {
            System.out.println("Index: " + block.getIndex());
            System.out.println("Timestamp: " + block.getTimestamp());
            System.out.println("Data: " + block.getData());
            System.out.println("Previous Hash: " + block.getPreviousHash());
            System.out.println("Hash: " + block.getHash());
            System.out.println("------------------------------");
        }
    }
}
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<title>Authorize Donor Users</title>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
<link href="css/style.css" rel="stylesheet" type="text/css" />
<link rel="stylesheet" type="text/css" href="css/coin-slider.css" />
<script type="text/javascript" src="js/cufon-yui.js"></script>
<script type="text/javascript" src="js/cufon-aller.js"></script>
<script type="text/javascript" src="js/jquery-1.4.2.min.js"></script>
<script type="text/javascript" src="js/script.js"></script>
<script type="text/javascript" src="js/coin-slider.min.js"></script>
<style type="text/css">
<!--
.style1 {font-size: 14px}
.style2 {font-size: 24px}
.style5 {color: #3A94CD}
```

```
.style7 {color: #CC0000}
.style9 {color: #006600}
.style10{color:#FF3300}
.style22{color:#000000}
-->
</style>
</head>
<body>
<div class="main">
  <div class="header">
    <div class="header_resize">
      <div class="menu_nav">
        <ul>
          <li class="active"><a href="index.html"><span>Home Page</span></a></li>
                          <li><a href="HospitalLogin.jsp">Hospital</a></li>
                          <strong></strong>
          <li><a href="CompanyLogin.jsp">Donor</a></li>
          <li><a href="UserLogin.jsp"><span>Patients</span></a></li>
        </ul>
      </div>
      <div class="logo">
        <h1 class="style1"><a href="index.html" class="style2">Blockchain Based
Management for Organ Donation and Transplantation</a></h1>
      </div>
      <div class="clr"></div>
      <div class="slider">
       <div id="coin-slider"> <a href="#"><img src="images/slide1.jpg" width="970"
height="305" alt="" /> </a> <a href="#"><img src="images/slide2.jpg" width="970"
height="305" alt="" /> </a> <a href="#"><img src="images/slide3.jpg" width="970"
height="305" alt="" /> </a> </div>
      </div>
      <div class="clr"></div>
    </div>
  </div>
  <div class="content">
    <div class="content_resize">
      <div class="mainbar">
        <div class="article">
          <h2>Authorize Donor Users...<span class="style5"></span></h2>
          <p class="infopost">&nbsp;</p>
          <div class="clr"></div>

          <div class="post_content">
                          <table width="592" border="1" align="left"  cellpadding="0"
cellspacing="0"  ">
                        <tr>
                      <td  width="37"  valign="middle" height="34" style="color:
#2c83b0;"><div align="center" class="style57 style56
style7"><b>ID</b></div></td>
                      <td  width="116" valign="middle" height="34" style="color:
#2c83b0;"><div align="center" class="style57 style56 style7"><b>User
Image</b></div></td>
                      <td  width="109" valign="middle" height="34" style="color:
#2c83b0;"><div align="center" class="style57 style56 style7"><b>User
Name</b></div></td>
                      <td  width="116" valign="middle" height="34" style="color:
#2c83b0;"><div align="center" class="style57 style56
style7"><b>Email</b></div></td>
                      <td  width="100" valign="middle" height="34" style="color:
#2c83b0;"><div align="center" class="style57 style56
style7"><b>Address</b></div></td>
                      <td  width="81"  valign="middle" height="34" style="color:
#2c83b0;"><div align="center" class="style7 style57
style56"><b>Status</b></div></td>
                    </tr>
                    <%@ include file="connect.jsp" %>
                    <%
                    String s1,s2,s3,s4,s5,s6,s7;
                    int i=0;
                    try
                    {
                            String query="select * from cuser";
                            Statement st=connection.createStatement();
                            ResultSet rs=st.executeQuery(query);
                            while ( rs.next() )
                            {
                                    i=rs.getInt(1);
                                    s1=rs.getString(3);
                                    s2=rs.getString(5);
                                    s3=rs.getString(7);
                                    s4=rs.getString(9);
                                    %>
                <tr>
               <td height="0" align="center"  valign="middle"><div align="center"
class="style5 style37 style54 style55 style22"><span class="style22">
                <%out.println(i);%>
               </span></div></td>
               <td width="116" rowspan="1" align="center" valign="middle" ><div
class="style5 style37 style54 style55 style22" style="margin:10px 13px 10px 13px;"
> <a class="#" id="img1" href="#" >
                 <input  name="image" type="image"
src="user_Pic.jsp?picture=<%="cuserimage"%>&id=<%=i%>" style="width:90px;
height:90px;" />
               </a> </div></td>
       <td height="0" align="center"  valign="middle"><div align="center"
class="style5 style20 style37 style54 style55 style22">
                  <span class="style10">
                  <b><%out.println(s1);%></b>
                  </span></div></td>
               <td height="0" align="center"  valign="middle"><div align="center"
class="style5 style20 style37 style54 style55 style22">
```

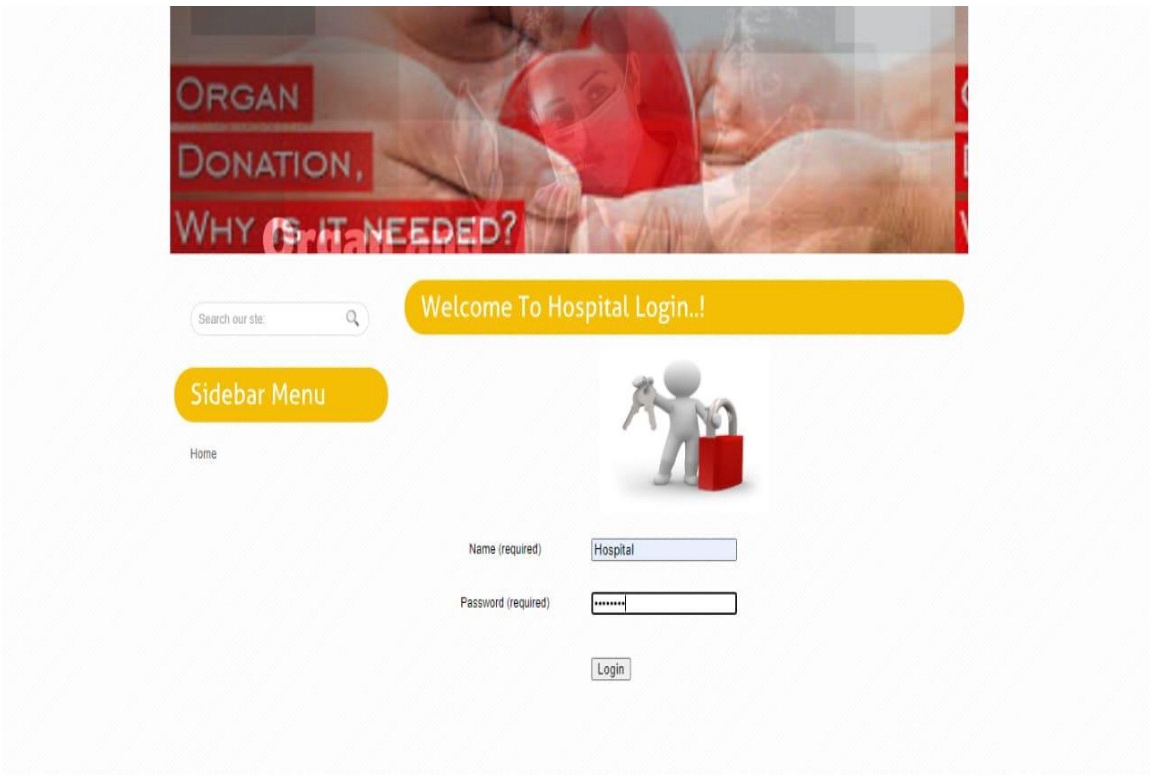
```
                    <span class="style22">
                    <%out.println(s2);%>
                    </span></div></td>
                <td height="0" align="center"  valign="middle"><div align="center"
class="style5 style20 style37 style54 style55 style22">
                    <span class="style22">
                    <%out.println(s3);%>
                    </span></div></td>
                <%
                            if(s4.equalsIgnoreCase("waiting"))
                            {
                            %>
                <td valign="middle" height="0"
style="color:#000000;"align="center"><div align="center" class="style22 style5
style20 style30 style37">
                   <div align="center" class="style20 style37 style46"><a
href="A_UserStatus.jsp?type=<%="cuser"%>&id=<%=i%>" class="style30 style32
style9"><b>waiting</b></a></div>
                </div></td>
                <%
                            }
                            else
                            {
                            %>
                <td width="17" height="0" align="center"  valign="middle"><div
align="center" class="style22 style5 style20 style37 style55 style58">
                 <b><%out.println(s4);%></b>
                </div></td>
                <%
                            }

                            %>
               </tr>
               <%
```

```
                                        }

                                        connection.close();
                                }
                                catch(Exception e)
                                {
                                        out.println(e.getMessage());
                                }
                                %>
     </table>
                            <p class="style19">&nbsp;</p>
                      <p>&nbsp;</p>

                      <p><a href="Hospital_Main.jsp"
class="style16">Back</a></p>

                   </div>
    <div class="clr"></div>
   </div>
  </div>
  <div class="sidebar">
   <div class="searchform">
    <form id="formsearch" name="formsearch" method="post" action="#">
     <span>
     <input name="editbox_search" class="editbox_search" id="editbox_search"
maxlength="80" value="Search our ste:" type="text" />
     </span>
     <input name="button_search" src="images/search.gif" class="button_search"
type="image" />
    </form>
   </div>
   <div class="clr"></div>
   <div class="gadget">
    <h2 class="star"><span>Hospital</span> Menu</h2>
```

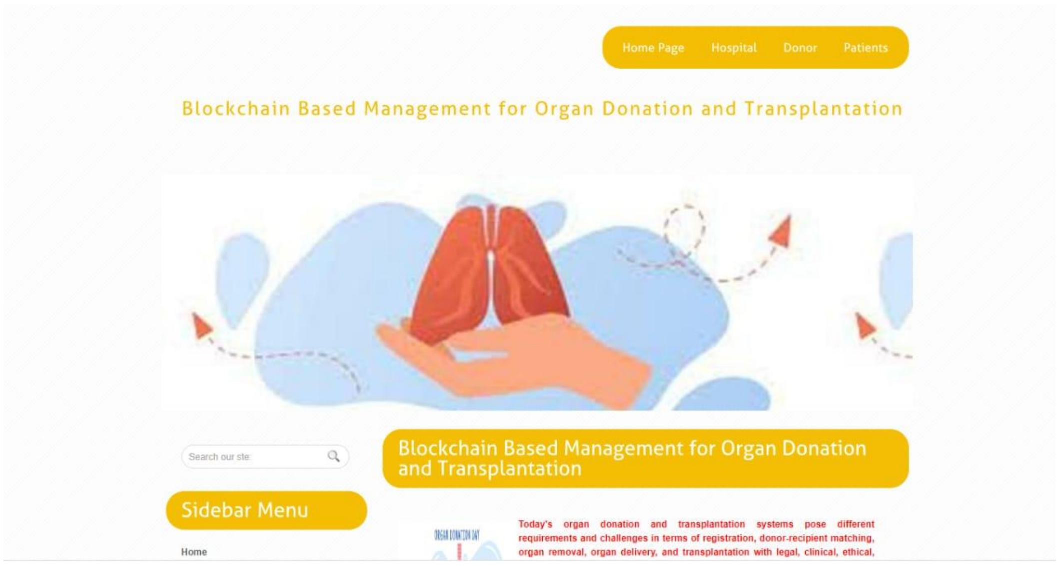
```
          <div class="clr"></div>
          <ul class="sb_menu">
            <li><a href="Hospital_Main.jsp">Home</a></li>
                                    <li><a href="HospitalLogin.jsp">Logout</a></li>
          </ul>
        </div>
      </div>
      <div class="clr"></div>
    </div>
  </div>
  <div class="fbg">
    <div class="fbg_resize">
      <div class="clr"></div>
    </div>
  </div>
  <div class="footer">
    <div class="footer_resize">
      <div style="clear:both;"></div>
    </div>
  </div>
</div>
<div align=center></a></div></body>
</html>
```

9.2 SCREEN SHOTS:

1.log-in page

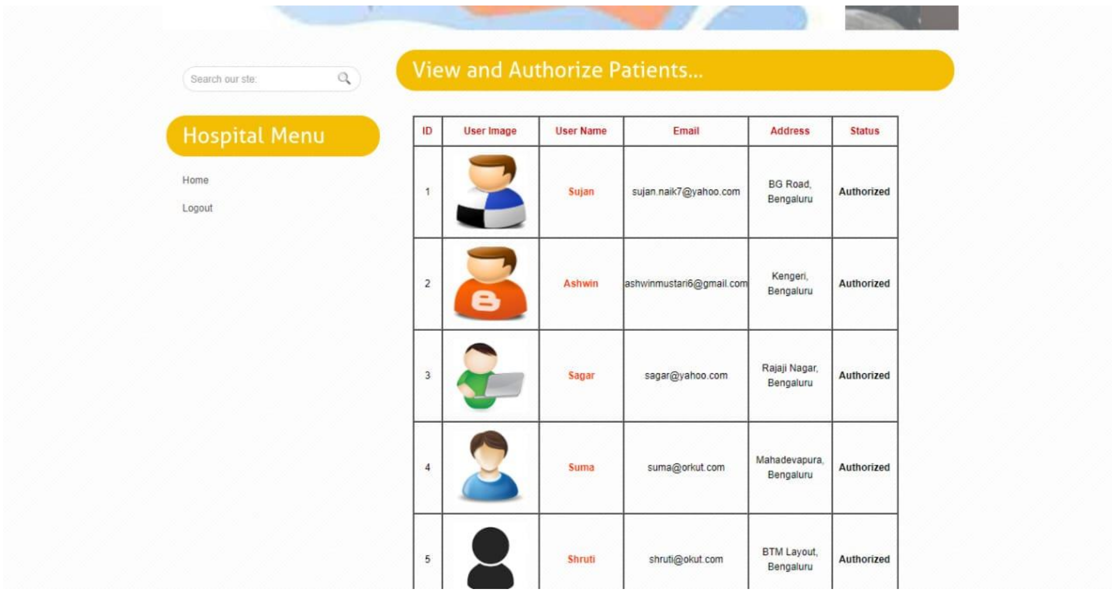


2.main menu


Home Page
Hospital
Donor
Patients
Blockchain Based Management for Organ Donation and Transplantation
Search our site:
Blockchain Based Management for Organ Donation and Transplantation
Sidebar Menu
Home
ORGAN DONATION DAY

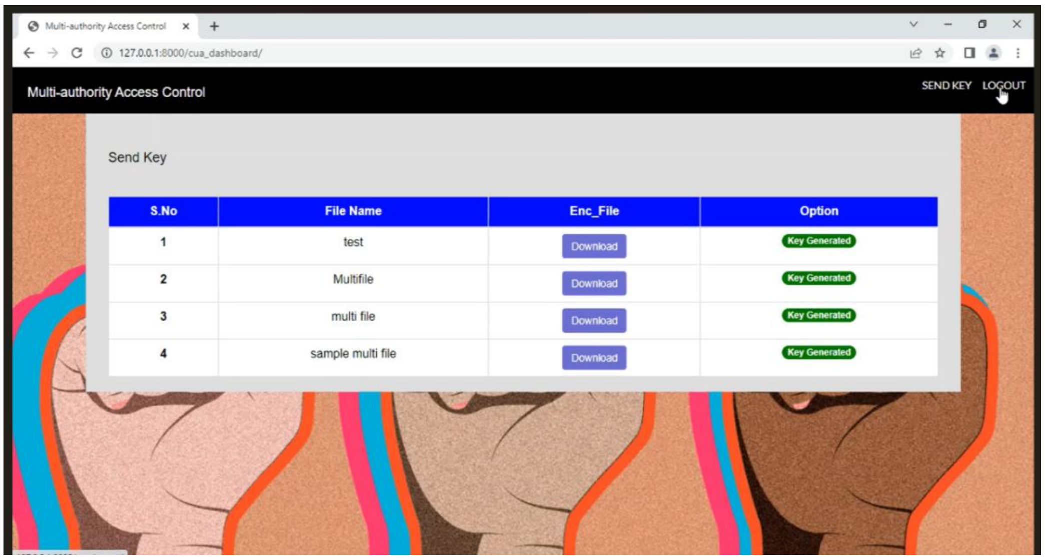
Today's organ donation and transplantation systems pose different requirements and challenges in terms of registration, donor-recipient matching, organ removal, organ delivery, and transplantation with legal, clinical, ethical,

3. List of organs and Autorized Patients



View and Authorize Patients...

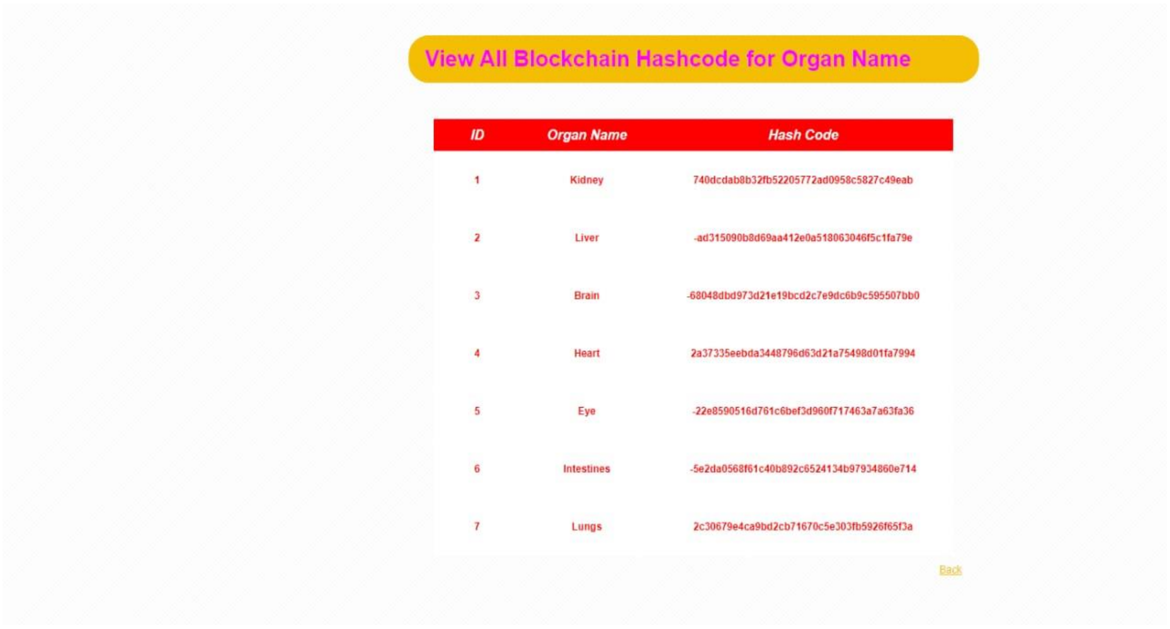
ID	User Image	User Name	Email	Address	Status
1		Sujan	sujan.naik7@yahoo.com	BG Road, Bengaluru	Authorized
2		Ashwin	ashwinmustari6@gmail.com	Kengeri, Bengaluru	Authorized
3		Sagar	sagar@yahoo.com	Rajaji Nagar, Bengaluru	Authorized
4		Suma	suma@orkut.com	Mahadevapura, Bengaluru	Authorized
5		Shruti	shruti@okut.com	BTM Layout, Bengaluru	Authorized



Multi-authority Access Control
127.0.0.1:8000/cua_dashboard/
Multi-authority Access Control
SEND KEY
LOGOUT

Send Key

S.No	File Name	Enc_File	Option
1	test	Download	Key Generated
2	Multifile	Download	Key Generated
3	multi file	Download	Key Generated
4	sample multi file	Download	Key Generated

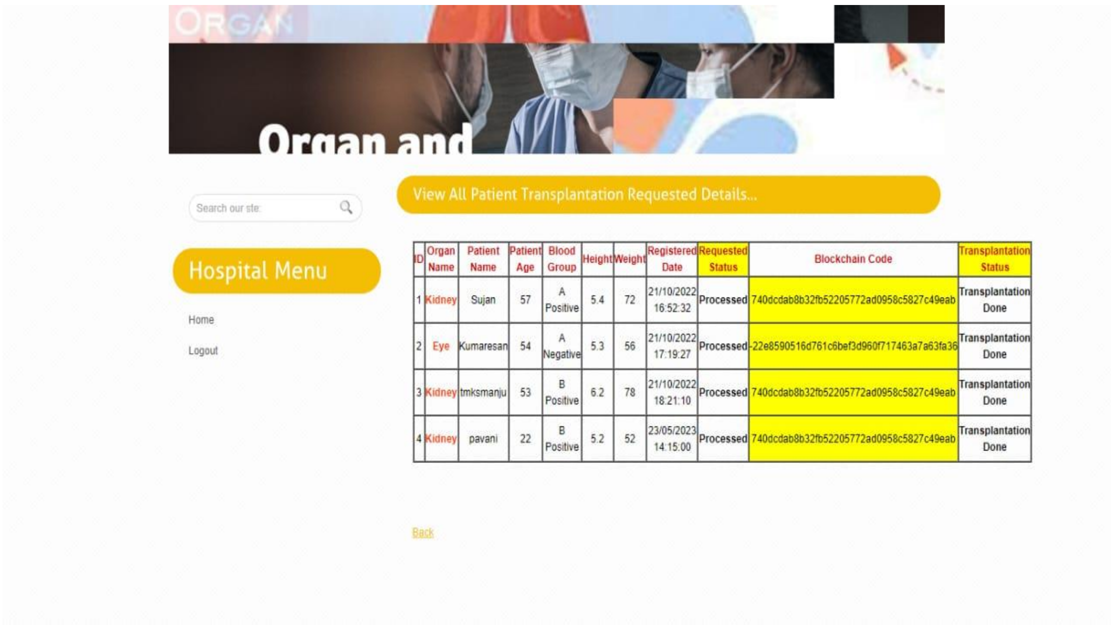


View All Blockchain Hashcode for Organ Name

ID	Organ Name	Hash Code
1	Kidney	740dcdab8b32fb52205772ad0958c5827c49eab
2	Liver	-ad315090b8d69aa412e0a518063046f5c1fa79e
3	Brain	-68048dbd973d21e19bcd2c7e9dc6b9c595507bb0
4	Heart	2a37335eebda3448796d63d21a75498d01fa7994
5	Eye	-22e8590516d761c6bef3d960f717463a7a63fa36
6	Intestines	-5e2da0568f61c40b892c6524134b97934860e714
7	Lungs	2c30679e4ca9bd2cb71670c5e303fb5926f65f3a

Back

4. LIST OF REQUEST DETAILS

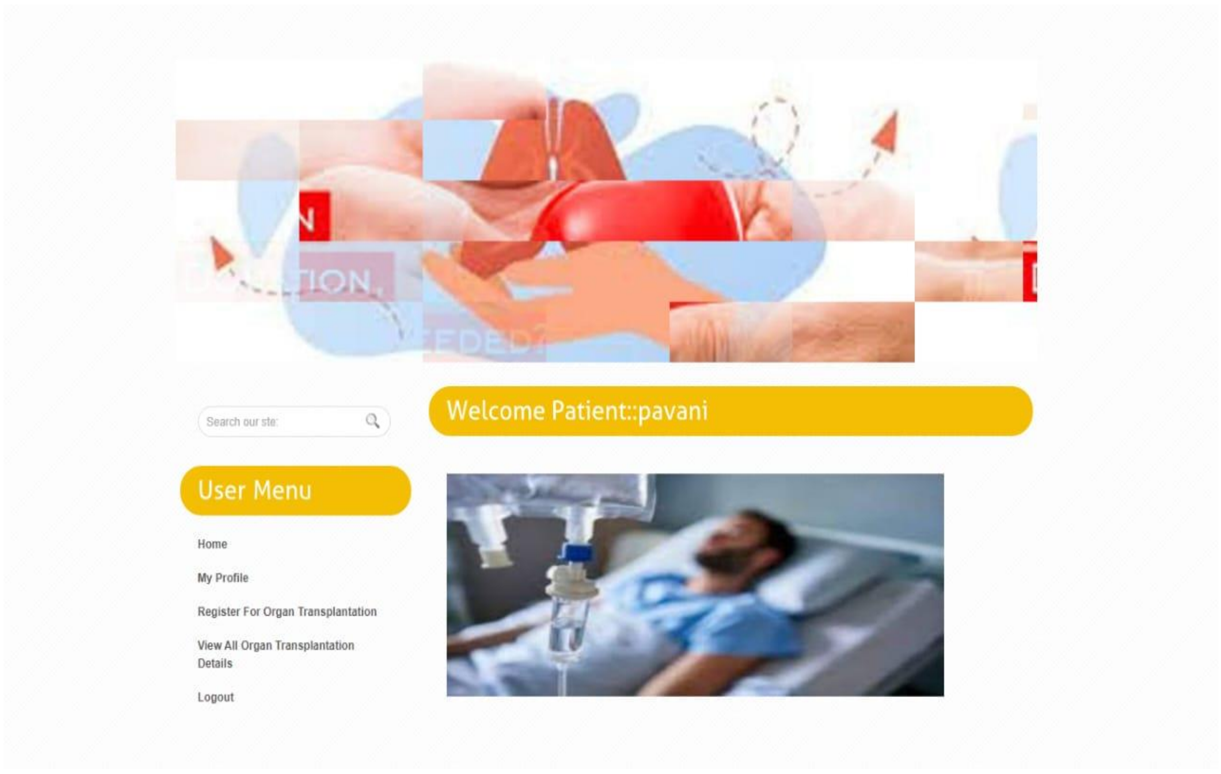


View All Patient Transplantation Requested Details...

ID	Organ Name	Patient Name	Patient Age	Blood Group	Height	Weight	Registered Date	Requested Status	Blockchain Code	Transplantation Status
1	Kidney	Sujan	57	A Positive	5.4	72	21/10/2022 16:52:32	Processed	740dcdab8b32fb52205772ad0958c5827c49eab	Transplantation Done
2	Eye	Kumaresan	54	A Negative	5.3	56	21/10/2022 17:19:27	Processed	-22e8590516d761c6bef3d960f717463a7a63fa36	Transplantation Done
3	Kidney	tmksmanju	53	B Positive	6.2	78	21/10/2022 18:21:10	Processed	740dcdab8b32fb52205772ad0958c5827c49eab	Transplantation Done
4	Kidney	pavani	22	B Positive	5.2	52	23/05/2023 14:15:00	Processed	740dcdab8b32fb52205772ad0958c5827c49eab	Transplantation Done

Back

5. ORGAN FOR PATIENT


Search our site:
Welcome Patient::pavani
User Menu
Home
My Profile
Register For Organ Transplantation
View All Organ Transplantation Details
Logout

10. REFERENCES

[1] L. A. Dajim, S. A. Al-Farras, B. S. Al-Shahrani, A. A. Al-Zuraib, and R. Merlin Mathew, ``Organ donation decentralized application using blockchain technology," in Proc. 2nd Int. Conf. Comput. Appl. Inf. Secur. (ICCAIS), May 2019, pp. 14, doi: 10.1109/cais.2019.8769459.

[2] A. Powell. (Mar. 18, 2019). A Transplant Makes History. Harvard Gazette. [Online]. Available: https://news.harvard.edu/gazette/story/2011/09/atransplant-makes-history/

[3] Organ Donation Facts and Info: Organ Transplants. Accessed: Apr. 18, 2021. [Online]. Available: https://my.clevelandclinic.org/health/ articles/11750-organ-donation-and-transplantation

[4] (Mar. 21, 2019). Facts and Myths About Transplant. Accessed: Apr. 21, 2021. [Online]. Available: https://www.americantransplant foundation.org/about-transplant/facts-and-myths/

[5] Organ Procurement and Transplantation Network. Accessed: Apr. 18, 2021. [Online]. Available: https://optn.transplant.hrsa.gov/ resources/ethics/ethical-principles-in-the-allocation-of-humanorgans/

[6] How Donation Works. Accessed: Jan. 7, 2022. [Online]. Available: https://www.organdonor.gov/learn/process

[7] UFO Themes. (Aug. 1, 2017). Organ Donation and Transplanta- tion in Germany. Plastic Surgery Key. [Online]. Available: https://plasticsurgerykey.com/organ-donation-and-transplantation-in-germany/

[8] Harvard Business Review. (Dec. 13, 2021). Electronic Health Records Can Improve the Organ Donation Process. Accessed: Apr. 8, 2022. [Online]. Available: https://hbr.org/2021/12/electronic-health-records-can-improvethe-organ-donation-process

[9] U. Jain, ``Using blockchain technology for the organ procurement and transplant network," San Jose State Univ., San Jose, CA, USA, Tech. Rep., 2020, doi: 10.31979/etd.g45p-jtuy.

[10] M. He, A. Corson, J. Russo, and T. Trey, ``Use of forensic DNA testing to trace unethical organ procurement and organ trafcking practices in regions that block transparent access to their transplant data," SSRN Electron. J., 2020, doi: 10.2139/ssrn.3659428.

[11] Livemint. The Illegal Organ Trade Thrives in India-and it isn't Likely to End Soon. Accessed: Dec. 21, 2021. [Online]. Available:https://www.livemint.com/Politics/pxj4YasmivrvAhanv6OOCJ/Whyorgan-trafficking-thrives-in-India.html

[12] D. P. Nair. (2016). Organ is Free, Transplant Cost is Problem. [Online]. Available:https://timesondia.indiatimes.com/life-style/healthtness/health-news/Organ-is-free-transplant-cost-isproblem/articleshow/54014378.cms

[13] P. Ranjan, S. Srivastava, V. Gupta, S. Tapaswi, and N. Kumar, ``Decentralised and distributed system for organ/tissue donation and transplantation," in Proc. IEEE Conf. Inf. Commun. Technol., Dec. 2019, pp. 16, doi: 10.1109/cict48419. 2019.9066225.

[14] V. Puggioni. (Feb. 26, 2022). An Overview of the Blockchain Develop-ment Lifecycle. Cointelegraph. Accessed: Apr. 8, 2022. [Online]. Available: https://cointelegraph.com/explained/an-overview-of-the-blockchaindevelopment-lifecycle

[15] History of Blockchain. Accessed: Apr. 8, 2022. [Online]. Available: https://www.icaew.com/technical/technology/blockchain-andcryptoassets/blockchain-articles/what-is-blockchain/histor