A

Major Project

On

**DECENTRALIZATION AND THE SECURITY ISSUES IN BLOCKCHAIN**

**ENABLED INTERNET OF THINGS**

(Submitted in partial fulfilment of the requirements for the award of Degree)

BACHELOR OF TECHNOLOGY

in

COMPUTER SCIENCE AND ENGINEERING

By

## G. SHRIYA(197R1A0576)

## P. DIVYA(197R1A0597)

## V. VANI(197R1A05B4)

Under the Guidance

Of

**V NARESH KUMAR**

(Associate Professor)



**CMR TECHNICAL CAMPUS**

UGC AUTONOMOUS

Accredited by NAAC, NBA, Permanently Affiliated to JNTUH,

Approved by AICTE, New Delhi Recognized Under Section 2(f) & 12(B)

of the UGCAct.1956, Kandlakoya(V) Medchal Road, Hyderabad- 501401.

**2019 - 2023**

**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING**



## CERTIFICATE

This is to certify that the project entitled as “**DECENTRALIZATION AND SECURITY ISSUES IN BLOCKCHAIN ENABLED INTERNET OF THINGS**” is submitted by G.SHRIYA(197R1A0576), P. DIVYA (197R1A0597) AND V. VANI (197R1A05B4) in partial Fulfilment of the requirements for the award of the degree of B. Tech in Computer Science and Engineering to the CMR Technical Campus, Hyderabad, is a record of bonafide work carried out by his/her under our guidance and supervision during the year 2022 – 2023.

The results embodied in this thesis have not been submitted to any other University or Institute for the award of any degree or diploma.

#### 

#### V. NARESH KUMAR Dr. A. Raji Reddy

Associate Professor

**INTERNAL GUIDE**

#### DIRECTOR

**Dr. K. SRUJAN RAJU**

**HOD**

# Submitted viva examination held on

#### EXTERNAL EXAMINER

**ACKNOWLEDGEMENT**

Apart from the efforts of us, the success of any project depends largely on the encouragement and guidelines of many others. We take this opportunity to express our gratitude to the people who have been instrumental in the successful completion of this project.

We take this opportunity to express my deep gratitude and regard to my guide **Mr. V. Naresh Kumar**, Associate Professor for his exemplary guidance, monitoring and constant encouragement throughout the project work. The blessing, help and guidance given by him shall carry us a long way in the journey of life on which we are about to embark.

We also take this opportunity to express a deep sense of gratitude to Project Review Committee (PRC) **Dr. Punyaban Patel, Ms. Shilpa, Dr. M. Subha Mastan Rao & J. Narasimha Rao** for their cordial support, valuable information and guidance, which helped us in completing this task through various stages.

We are also thankful to **Dr. K. Srujan Raju**, Head of the Department of Computer Science and Engineering for providing encouragement and support for completing this project successfully.

We are obliged to **Dr. A. Raji Reddy**, Director for being cooperative throughout the course of this project. We also express our sincere gratitude to **Sri. Ch. Gopal Reddy**, Chairman for providing excellent infrastructure and a nice atmosphere throughout the course of this project.

The guidance and support received from all the members of **CMR Technical Campus** who contributed to the completion of the project. We are grateful for their constant support and help.

Finally, we would like to take this opportunity to thank our family for their constant encouragement, without which this assignment would not be completed. We sincerely acknowledge and thank all those who gave support directly and indirectly in the completion of this project.

**G Shriya (197R1A0576)**

**P Divya (197R1A0597)**

**V Vani (197R1A05B4)**

**ABSTRACT**

This project is describing about security issues for patient health record stored in Decentralized (data will be maintained at multiple peer or systems) Blockchain server. Now-a-days patient data can be shared between multiple hospitals, insurance agencies, Government and lab. Due to sharing of data there will be data security issues raised for patient as this data can be misuse by agency peoples or attackers may steal this data. To overcome from this problem author is using Decentralized Blockchain server which maintain data as blocks of trees and at each transaction all previous hash code will be verified and if verification successful then data will be consider as intact and if data changes then Blockchain server will notify that system in under attack and it gather data from another working node. Due to this transaction hash code verification and immutable data storage make Blockchain secure and trustable in current market. To further secure data author is using encryption algorithms before storing data in Blockchain and author has describe many traditional and new encryption algorithms such as DES and many more. Python currently does not support ABE algorithm so I am using AES algorithm to encrypt patient data before storage.

**i**

**LIST OF FIGURES/TABLES**

**FIGURE NO FIGURE NAME PAGE NO**

Figure 4.6

Figure 4.5.1

|  |  |  |
| --- | --- | --- |
| Figure 4.1 | Project Architecture of  decentralization and security issues in blockchain enabled internet of things | 8 |
| Figure 4.3 | Use Case diagram of decentralization and security issues in blockchain enabled internet of things | 10 |
| Figure 4.4 | Class Diagram of decentralization and security issues in blockchain enabled internet of things | 11 |
| Figure 4.4.1 | Class Diagram of decentralization and security issues in blockchain enabled internet of things | 12 |
| Figure 4.5 | Sequence Diagram of decentralization and security issues in blockchain enabled internet of things  Sequence Diagram of decentralization and security issues in blockchain enabled internet of things  Activity diagram of Decentralization and security Issues in Blockchain Enabled Internet Of Things | 13  14  15 |
|  |  |  |
|  |  |  |

**ii**

**LIST OF SCREENSHOTS**

**FIGURE NO FIGURE NAME PAGE NO**

Figure 6.1 New User Signup 22

Figure 6.2 Patient profile creation 22

Figure 6.3 New Agency creation 23

Figure 6.4 Patient Login 23

Figure 6.5 Access Patient shared data 24

**iii**

**TABLE OF CONTENTS PAGE NO**

**ABSTRACT**

**LIST OF FIGURES/TABLES**

**LIST OF SCREENSHOTS**

1. **INTRODUCTION**
   1. PROJECT SCOPE
   2. PROJECT PURPOSE
   3. PROJECT FEATURES
2. **LITREATURE SURVEY**

2.1 BLENDCAC: A BLOCKCHAIN-ENABLED DECENTRALIZED

CAPABILITY-BASED ACCESS CONTROL FOR IOTS

2.1.1 BLOCKCHAIN MEETS IOT: AN ARCHITECTURE FOR SCALABLE

ACCESS

2.1.2 A SURVEY ON THE ADOPTION OF BLOCKCHAIN IN IOT:

CHALLENGES AND SOLUTIONS

2.1.3 BLOCKCHAIN-ENABLED EDGE INTELLIGENCE FOR IOT:

BACKGROUND, EMERGING TRENDS AND OPEN ISSUE

**3. SYSTEM ANALYSIS**

3.1 EXISTING SYSTEM

3.1.1 DISADVANTAGES OF EXISTING SYSTEM

3.2 PROPOSED SYSTEM

3.2.1 ADVANTAGES OF PROPOSED SYSTEM

* 1. FEASIBILITY

3.3.1 ECONOMIC FEASIBILITY

3.3.2 TECHNICAL FEASIBILITY

3.3.3 SOCIAL FEASIBILITY

3.3.4 HARDWARE AND SOFTWARE REQUIREMENTS

1. **ARCHITECTURE**
   1. PROJECT ARCHITECTURE
   2. MODULES
   3. USE CASE DIAGRAM
   4. CLASS DIAGRAM
   5. SEQUENCE DIAGRAM
   6. ACTIVITY DIAGRAM

i

ii

iii

1

1

1

2

3

3

4

5

5

5

6

6

6

6

7

7

8

9

10

11

13

15

16

22

25

25

25

26

27

27

28

29

**5. IMPLEMENTATION**

1. **RESULTS**
2. **TESTING**
   1. TYPES OF TESTING
      1. UNIT TESTING
      2. INTEGRATED TESTING
      3. FUNCTIONAL TESTING
   2. TEST CASES

7.2.1 CLASSIFICATION

1. **CONLCUSION**
2. **BIBLIOGRAPHY**

9.1 REFERENCES

**1. INTRODUCTION**

**Decentralization and the security issues in blockchain enabled internet of things**

**1. INTRODUCTION**

### PROJECT SCOPE

This project is titled as “decentralization and security issues in blockchain enabled internet of things”.The main objective is to build a supervised machine learning model for Forecasting value of a vehicle based on multiple attributes. The system that is being built must be future based that i.e future wise prediction must be possible. Providing graphical comparisons to provide a better view

### PROJECT PURPOSE

Blockchain has become a valuable research area. Blockchain is a distributed database system that can record all information about transactions. It can achieve the required security and reliability in an untrusted environment without relying on central participants. Blockchain will enable IoT devices to send data to the blockchain ledger to prevent tampering and counterfeiting. This fits perfectly into the basic functions and architecture of the IoT. However, blockchain technology is still in its infancy, and its integration with the IoT is still particularly challenging.

### PROJECT FEATURES

This technology is used for various purposes. It stores the data of the patients in a blockchain server. The user will be entering their details about the disease and user will be able to give the access to their data to only whom they want to share with such as Insurance companies, doctors or Government Agencies. The data is secured and privacy is maintained.

**CMRTC**

**1**

**2. LITERATURE SURVEY**

**Decentralization and the security issues in blockchain enabled internet of things**

**2.** **LITERATURE SURVEY**

**2.1 BlendCAC: A BLockchain-Enabled Decentralized Capability-Based**

**Access Control for IoTs**

**AUTHORS:**  Xu, R.; Chen, Y.; Blasch, E.; Chen,

**ABSTRACT:** The prevalence of Internet of Things (IoT) allows heterogeneous embedded smart devices to collaboratively provide smart services with or without human intervention. While leveraging the large-scale IoT-based applications like Smart Gird or Smart Cities, IoT also incurs more concerns on privacy and security. Among the top security challenges that IoT face, access authorization is critical in resource sharing and information protection. One of the weaknesses of today's access control (AC) is the centralized authorization server, which can be the performance bottleneck or the single point of failure. In this paper, BlendCAC, a blockchain-enabled decentralized capability-based AC is proposed for the security of IoTs. The BlendCAC aims at an effective access control processes to devices, services and information in large scale IoT systems. Based on the blockchain network, a capability delegation mechanism is suggested for access permission propagation. A robust identity-based capability token management strategy is proposed, which takes advantage of a smart contract for registration, propagation and revocation of the access authorization. In the proposed BlendCAC scheme, IoT devices are their own master to control their resources instead of being supervised by a centralized authority. Implemented and tested on a Raspberry Pi device and on a local private blockchain network, the experimental results demonstrate the feasibility of the proposed BlendCAC approach to offer a decentralized, scalable, lightweight and fine-grained AC solution to IoT systems.

**CMRTC**

**2**

**Decentralization and the security issues in blockchain enabled internet of things**

**2.1.1 Blockchain Meets IoT: An Architecture for Scalable Access**

**Management in IoT**

**AUTHORS:**  Novo, O

**ABSTRACT:** The Internet of Things (IoT) is stepping out of its infancy into full maturity and establishing itself as a part of the future Internet. One of the technical challenges of having billions of devices deployed worldwide is the ability to manage them. Although access management technologies exist in IoT, they are based on centralized models which introduce a new variety of technical limitations to manage them globally. In this paper, we propose a new architecture for arbitrating roles and permissions in IoT. The new architecture is a fully distributed access control system for IoT based on blockchain technology. The architecture is backed by a proof of concept implementation and evaluated in realistic IoT scenarios. The results show that the blockchain technology could be used as access management technology in specific scalable IoT scenarios.

**2.1.2 A Survey on the Adoption of Blockchain in IoT: Challenges and**

**Solutions**

**AUTHORS:**  Oscar Novo

**ABSTRACT:** The Internet of Things (IoT) is stepping out of its infancy into full maturity and establishing itself as a part of the future Internet. One of the technical challenges of having billions of devices deployed worldwide is the ability to manage them. Although access management technologies exist in IoT, they are based on centralized models which introduce a new variety of technical limitations to manage them globally. In this paper, we propose a new architecture for arbitrating roles and permissions in IoT. The new architecture is a fully distributed access control system for IoT based on blockchain technology.

**CMRTC**

**3**

**Decentralization and the security issues in blockchain enabled internet of things**

**2.1.3 Blockchain-Enabled Edge Intelligence for IoT: Background, Emerging**

**Trends and Open Issue**

**AUTHORS:**  Victor C. M. Leung

**ABSTRACT:** Blockchain, a distributed ledger technology (DLT), refers to a list of records with consecutive time stamps. This decentralization technology has become a powerful model to establish trust among trustless entities, in a verifiable manner. Motivated by the recent advancement of multi-access edge computing (MEC) and artificial intelligence (AI), blockchain-enabled edge intelligence has become an emerging technology for the Internet of Things (IoT). We review how blockchain-enabled edge intelligence works in the IoT domain, identify the emerging trends, and suggest open issues for further research. To be specific: (1) we first offer some basic knowledge of DLT, MEC, and AI; (2) a comprehensive review of current peer-reviewed literature is given to identify emerging trends in this research area; and (3) we discuss some open issues and research gaps for future investigations. We expect that blockchain-enabled edge intelligence will become an important enabler of future IoT, providing trust and intelligence to satisfy the sophisticated needs of industries and society.

**CMRTC**

**4**

**Decentralization and the security issues in blockchain enabled internet of things**

**3. SYSTEM ANALYSIS**

**3.1 EXISTING SYSTEM**

In Existing system, we are using normalization database which can easily tampered of patient details. Security, privacy, and data storage issues have become key issues in IoT systems. The current centralized architecture model in IoT systems is difficult to scale to meet the needs of future IoT systems. To solve the above problems, blockchain has become a valuable research area. Blockchain is a distributed database system that can record all information about transactions. It can achieve the required security and reliability in an untrusted environment without relying on central participants. Blockchain will enable IoT devices to send data to the blockchain ledger to prevent tampering and counterfeiting.

* + 1. **DISADVANTAGES**

1. Due to sharing of data there will be data security issues raised for patient as this data can be misuse by agency peoples or attackers may steal this data.
2. To avoid all these limitations and make the working more accurately the system needs to be computerized in a better way.

**3.2 PROPOSED SYSTEM**

Decentralized Blockchain server which maintain data as blocks of trees and at each transaction all previous hash code will be verified and if verification successful then data will be consider as intact and if data changes then Blockchain server will notify that system in under attack and it gather data from another working node. Due to this transaction hash code verification and immutable data storage make Blockchain secure and trustable in current market. To further secure data author is using encryption algorithms before storing data in Blockchain and author has describe many traditional and new encryption algorithms such as ABE, IBE, CPABE and many more. Python currently does not support ABE algorithm so I am using AES algorithm to encrypt patient data before storage.

**5**

**CMRTC**

**Decentralization and the security issues in blockchain enabled internet of things**

* + 1. **ADVANTAGES**

1. If data changes then Blockchain server will notify that system in under attack and it gather data from another working node.
2. At each transaction all previous hash code will be verified and if verification

successful then data will be consider as intact.

* 1. **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.

* + 1. **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.

**3.3.2 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.

**CMRTC**

**6**

**Decentralization and the security issues in blockchain enabled internet of things**

**3.3.3 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.

* + 1. **HARDWARE AND SOFTWARE REQUIREMENTS**

**HARDWARE REQUIREMENTS:**

Hardware interfaces specifies the logical characteristics of each interface between the software product and the hardware components of the system. The following are hardware requirements.

* SYSTEM : Pentium IV 2.4 GHz Or Higher
* RAM : 512 Mb Or More
* SPACE ON HARD DISK : 128 GB or More

**SOFTWARE REQUIREMENTS:**

Software Requirements specifies the logical characteristics of each interface and software

components of the system.

The following are some software requirements.

* OPERATING SYSTEM :Windows
* CODING LANGUAGE : Python 3.7

**7**

**CMRTC**

**4. ARCHITECTURE**

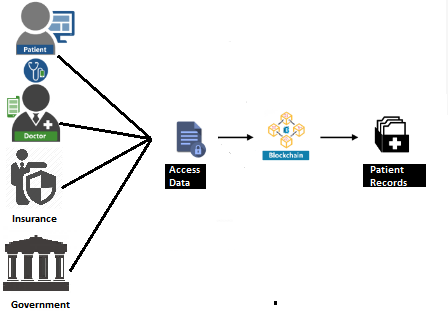
**Decentralization and the security issues in blockchain enabled internet of things**

**4. ARCHITECTURE**

**4.1 PROJECT ARCHITECTURE**

This project architecture shows the procedure followed for classification, starting from input to final prediction.





**Fig 4.1: Project architecture of Decentralization and security Issues in Blockchain Enabled Internet Of Things**

**8**

**CMRTC**

**Decentralization and the security issues in blockchain enabled internet of things**

**4.2 MODULES**

This project can be run by following Modules

* **Users:**

Patients who create their medical profiles and give permission access to Healthcare agents and this permission can be controlled by access control program to decide which users allowed to access patient data.

* **Healthcare Agents:**

Agents can be doctors, insurance companies or government users and here government can access patient data to know how many peoples are suffering from which disease. Insurance companies can access this data to decide to give insurance policy to patients or not and doctors can access this data for treatment.

* **Cloud Storage:**

Here Blockchain encrypted data storage will be consider as cloud storage as we don’t

have any cloud server so we are storing data in Blockchain server.

**CMRTC**

**9**

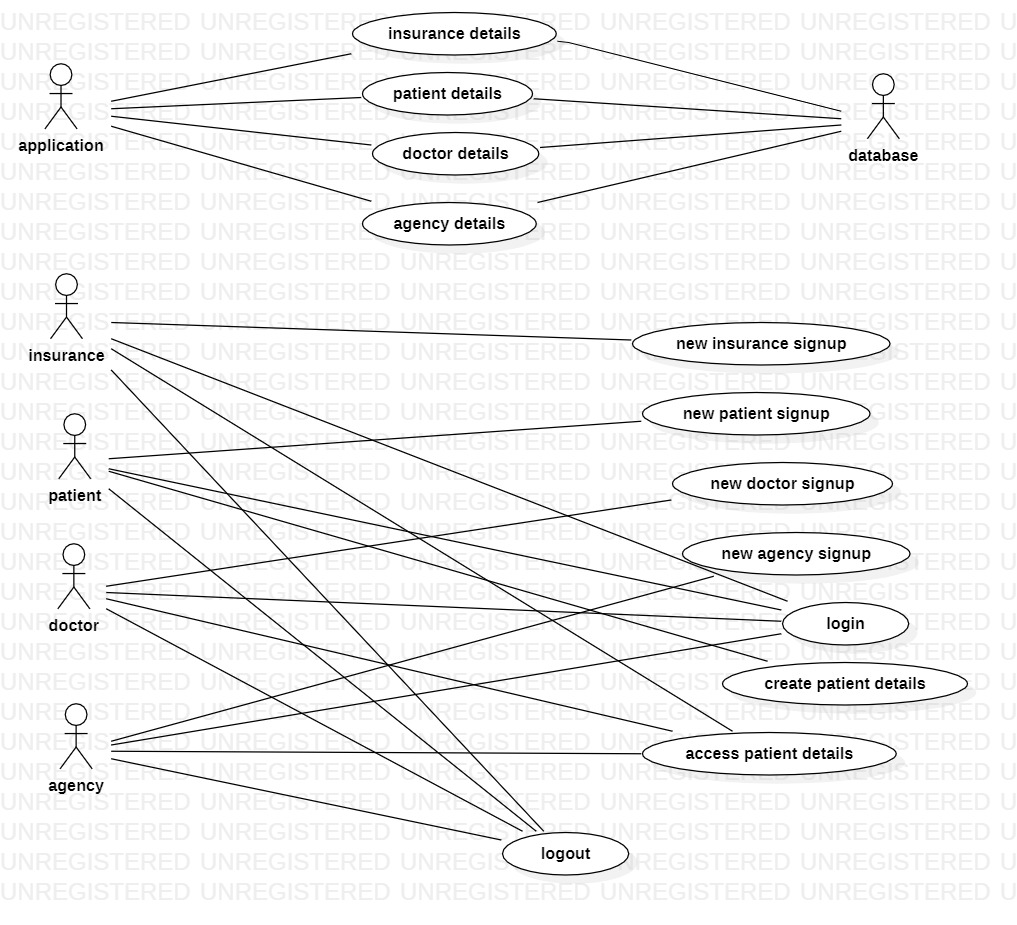
**Decentralization and the security issues in blockchain enabled internet of things**

**4.3 USE CASE DIAGRAM**

In the Use Case diagram, we have basically one actor who is the user in the trained

model.

A Use Case diagram is a graphical depiction of a user’s possible interactions with a system. A use case diagram shows various Use Case’s and different types of user’s the system has. The Use Case’s are represented by either circles or ellipses. The actor are often shown as stick figures.



**Fig 4.3: Use Case diagram of Decentralization and security Issues in Blockchain Enabled Internet Of Things**

**CMRTC**

**10**

**Decentralization and the security issues in blockchain enabled internet of things**

**4.4 CLASS DIAGRAM**

Class diagram is a type of static structure diagram that describes the structure of a system by showing the systems classes ,their attributes, operations(Methods),and their relationships among objects.

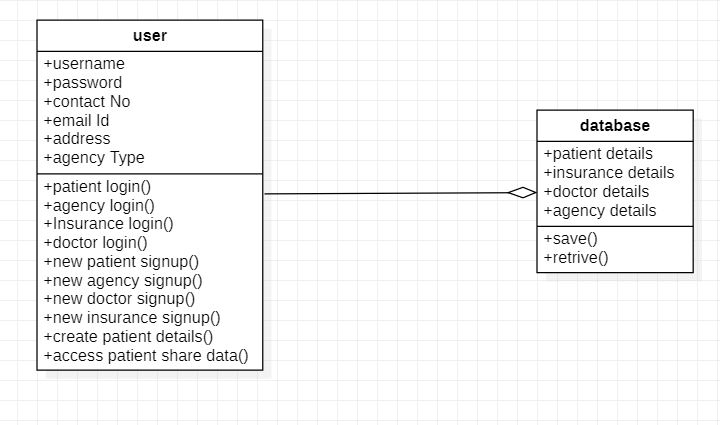


**Fig 4.4: Class diagram of Decentralization and security Issues in Blockchain Enabled Internet Of Things**

**CMRTC**

**11**

**Decentralization and the security issues in blockchain enabled internet of things**



**Fig 4.4.1: Class diagram of Decentralization and security Issues in Blockchain Enabled Internet Of Things**

**12**

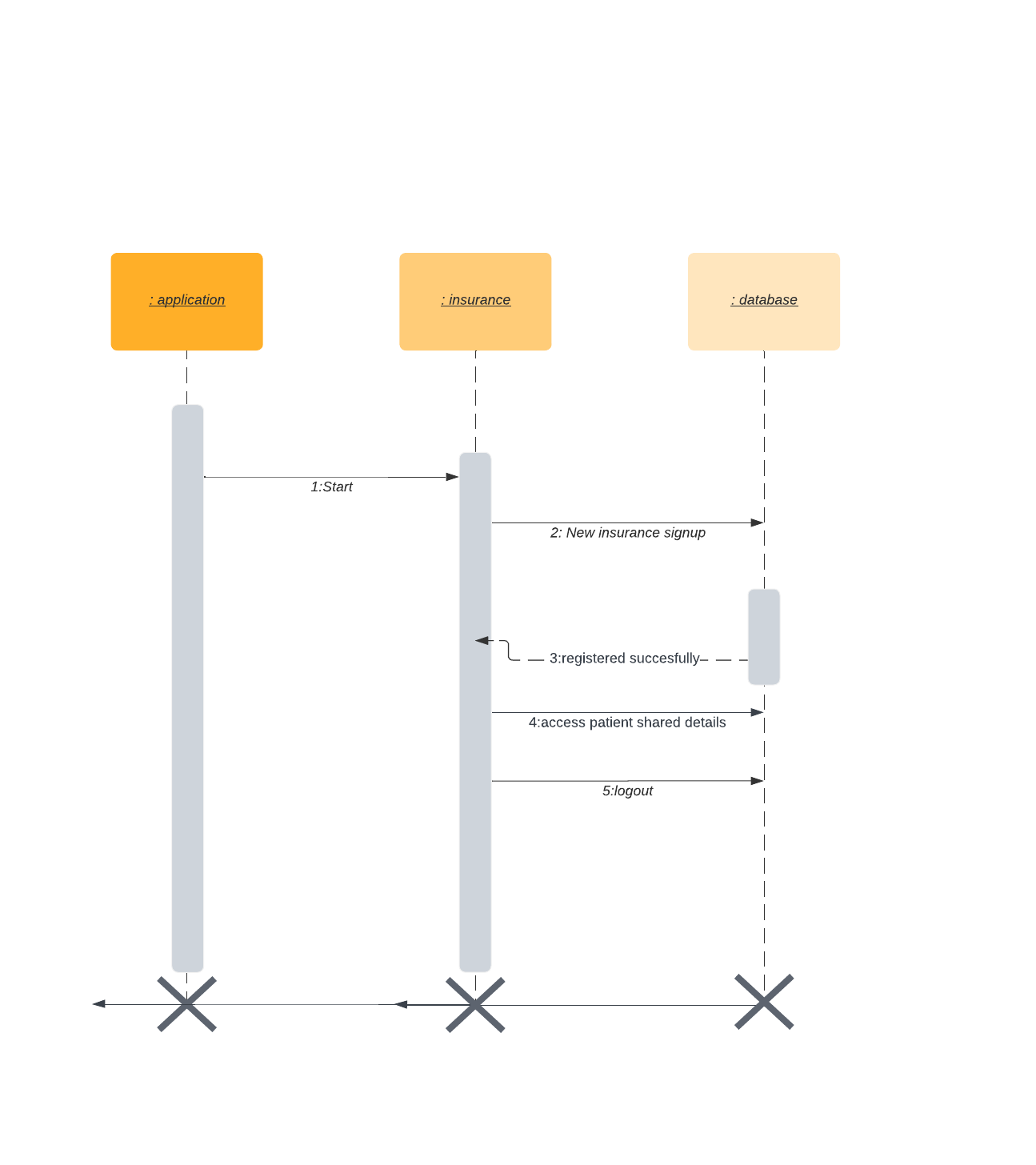
**CMRTC**

**Decentralization and the security issues in blockchain enabled internet of things**

**4.5 SEQUENCE DIAGRAM**



A Sequence diagram shows object interactions arranged in time sequence. It depicts the objects involved in the scenario and the sequence of messages exchanged between the objects needed to carry out the functionality of the scenario.

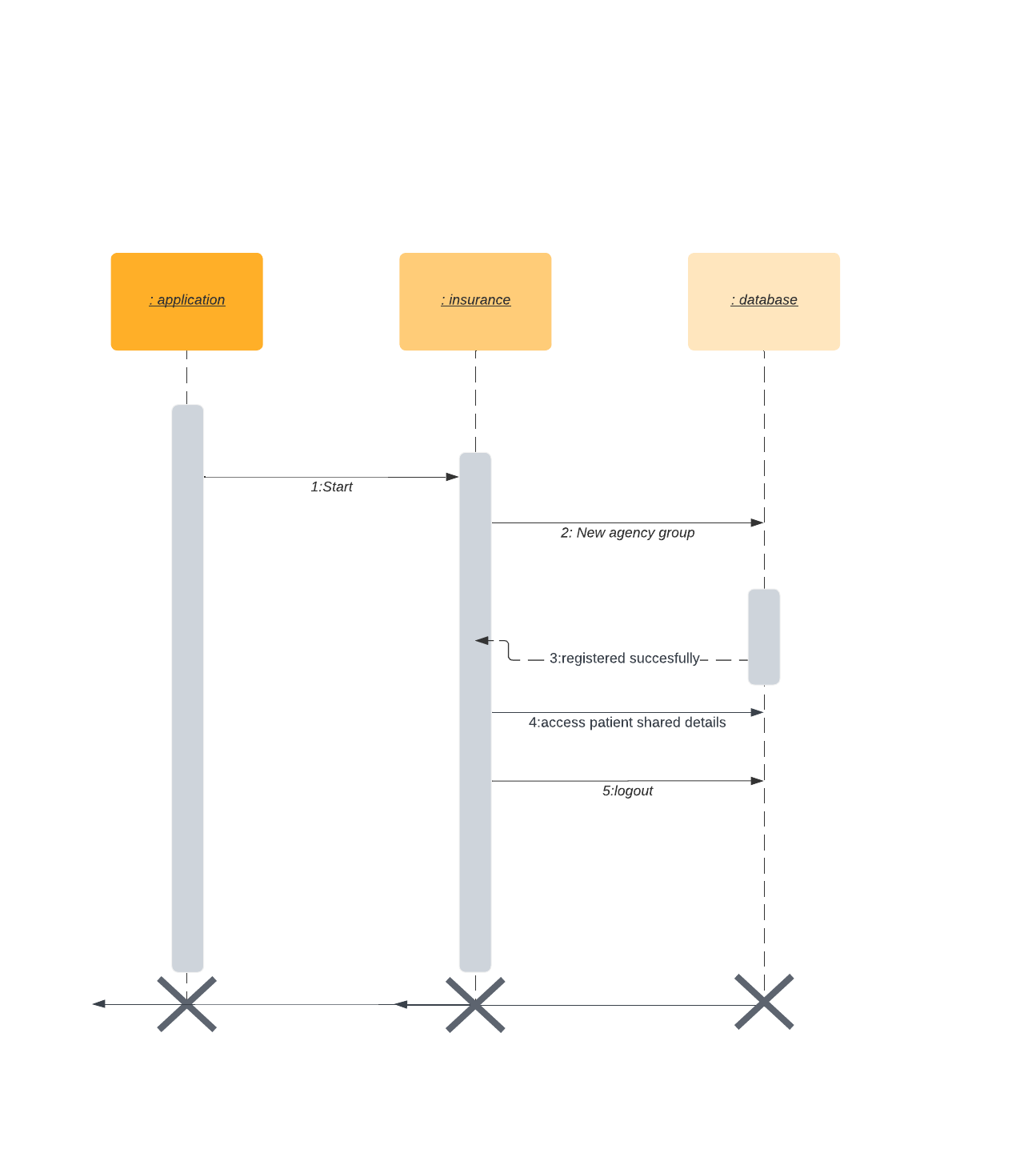
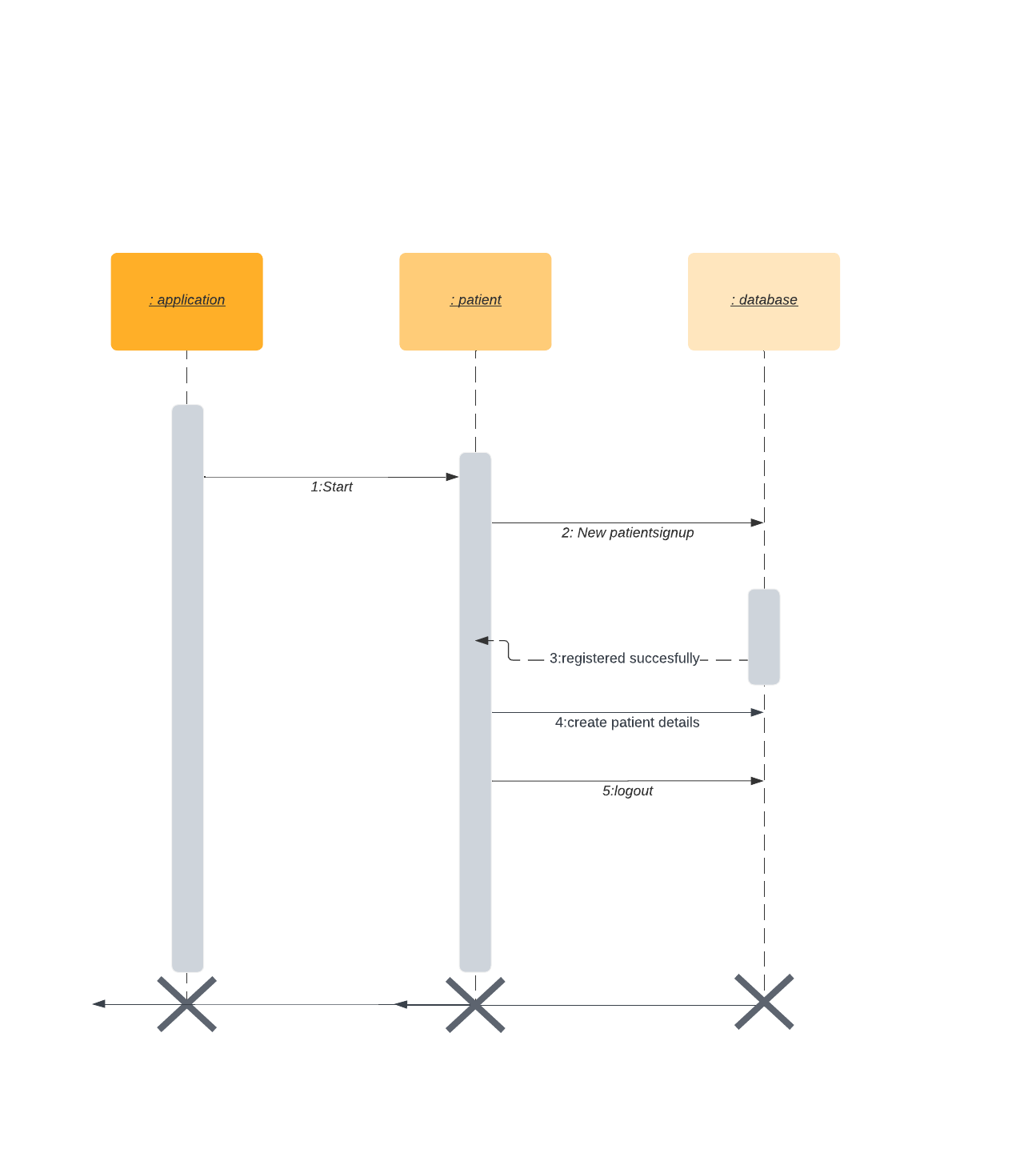
****

**Fig 4.5: Sequence diagram of Decentralization and security Issues in Blockchain Enabled Internet Of Things**

**CMRTC**

**13**

**Decentralization and the security issues in blockchain enabled internet of thing**

****

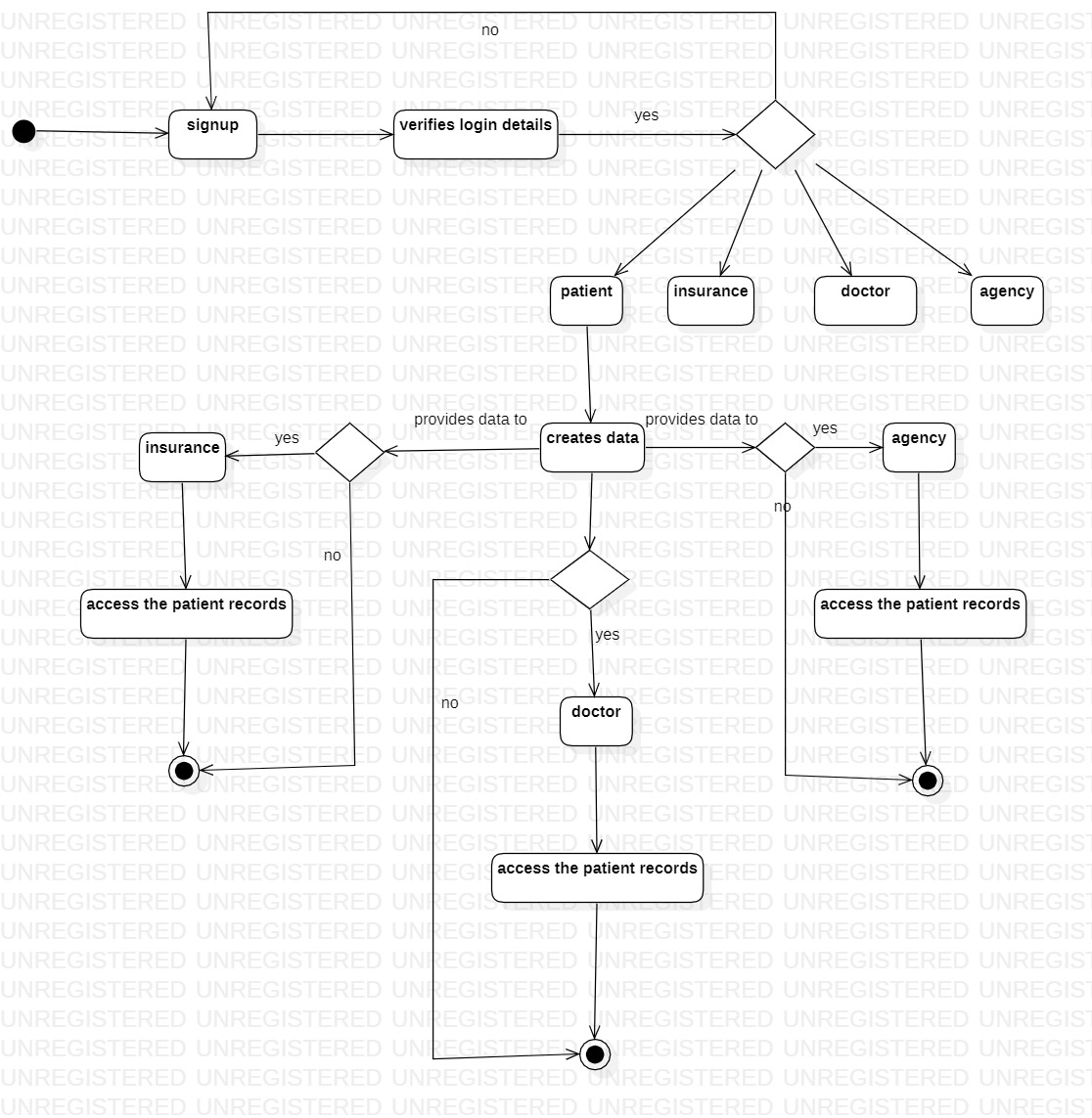
**Fig 4.5.1: Sequence diagram of Decentralization and security Issues in Blockchain Enabled Internet Of Things**

**CMRTC**

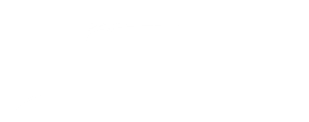
**14**

**Decentralization and the security issues in blockchain enabled internet of things**

**4.6 ACTIVITY DIAGRAM**

Activity diagram are graphical representations of work flows of step wise activities and actions with support for choice, iteration and, concurrency. They can also include elements showing the flow of data between activities through one or more data stores.

**Fig 4.6: Activity diagram of Decentralization and security Issues in Blockchain Enabled Internet Of Things**



**CMRTC**

**15**

**5. IMPLEMENTATION**

**Decentralization and the security issues in blockchain enabled internet of things**

**Sample code**

from django.shortcuts import render

from django.template import RequestContext

from django.contrib import messages

from django.http import HttpResponse

from Blockchain import \*

from Block import \*

import datetime

import pyaes, pbkdf2, binascii, os, secrets

import base64

blockchain = Blockchain()

if os.path.exists('blockchain\_contract.txt'):

with open('blockchain\_contract.txt', 'rb') as fileinput:

blockchain = pickle.load(fileinput)

fileinput.close()

def getKey(): #generating key with PBKDF2 for AES

password = "s3cr3t\*c0d3"

passwordSalt = '76895'

key = pbkdf2.PBKDF2(password, passwordSalt).read(32)

return key

def encrypt(plaintext): #AES data encryption

aes = pyaes.AESModeOfOperationCTR(getKey(), pyaes.Counter(31129547035000047302952433967654195398124239844566322884172163637846056248223))

ciphertext = aes.encrypt(plaintext)

return ciphertext

def decrypt(enc): #AES data decryption

aes = pyaes.AESModeOfOperationCTR(getKey(), pyaes.Counter(31129547035000047302952433967654195398124239844566322884172163637846056248223))

decrypted = aes.decrypt(enc)

return decrypted

def AccessData(request):

if request.method == 'GET':

return render(request, 'AccessData.html', {})

def index(request):

if request.method == 'GET':

return render(request, 'index.html', {})

def CreateProfile(request):

if request.method == 'GET':

**CMRTC**

**16**

**Decentralization and the security issues in blockchain enabled internet of things**

return render(request, 'CreateProfile.html', {})

def Agency(request):

if request.method == 'GET':

return render(request, 'Agency.html', {})

def Patient(request):

if request.method == 'GET':

return render(request, 'Patient.html', {})

def AgencySignup(request):

if request.method == 'GET':

return render(request, 'AgencySignup.html', {})

def AgencySignupAction(request):

if request.method == 'POST':

user = request.POST.get('t1', False)

password = request.POST.get('t2', False)

contact = request.POST.get('t3', False)

email = request.POST.get('t4', False)

address = request.POST.get('t5', False)

agency = request.POST.get('t6', False)

record = 'none'

for i in range(len(blockchain.chain)):

if i > 0:

b = blockchain.chain[i]

data = b.transactions[0]

data = base64.b64decode(data)

data = str(decrypt(data))

data = data[2:len(data)-1]

print(data)

arr = data.split("#")

if arr[0] == "agency":

if arr[1] == user:

record = "exists"

break

if record == 'none':

data = "agency#"+user+"#"+password+"#"+contact+"#"+email+"#"+address+"#"+agency

enc = encrypt(str(data))

enc = str(base64.b64encode(enc),'utf-8')

blockchain.add\_new\_transaction(enc)

hash = blockchain.mine()

b = blockchain.chain[len(blockchain.chain)-1]

print("Encrypted Data : "+str(b.transactions[0])+" Previous Hash : "+str(b.previous\_hash)+" Block No : "+str(b.index)+" Current Hash : "+str(b.hash))

bc = "Encrypted Data : "+str(b.transactions[0])+" Previous Hash : "+str(b.previous\_hash)+"<br/>Block No : "+str(b.index)+"<br/>Current Hash : "+str(b.hash)

blockchain.save\_object(blockchain,'blockchain\_contract.txt')

context= {'data':'Signup process completd and record saved in Blockchain with below

**CMRTC**

**17**

**Decentralization and the security issues in blockchain enabled internet of things**

hashcodes.<br/>'+bc}

return render(request, 'AgencySignup.html', context else:

context= {'data':username+'Username already exists'}

return render(request, 'AgencySignup.html', context)

def PatientLogin(request):

if request.method == 'POST':

pid = request.POST.get('t1', False)

strdata = '<table border=1 align=center width=100%><tr><th><font size='' color=black>Patient ID</th><th><font size='' color=black>Patient Name</th>'

strdata+='<th><font size='' color=black>Age</th><th><font size='' color=black>Problem Description</th><th><font size='' color=black>Profile Date</th><th><font size='' color=black>Access Control</th>'

strdata+='<th><font size='' color=black>Gender</th><th><font size='' color=black>Contact No</th><th><font size='' color=black>Address</th><th><font size='' color=black>Block Chain Hashcode</th></th></tr><tr>'

for i in range(len(blockchain.chain)):

if i > 0:

b = blockchain.chain[i]

data = b.transactions[0]

data = base64.b64decode(data)

data = str(decrypt(data))

data = data[2:len(data)-1]

print(data)

row = data.split("#")

if row[0] == "patients" and row[1] == pid:

strdata+='<td><font size='' color=black>'+str(row[1])+'</td><td><font size='' color=black>'+row[2]+'</td><td><font size='' color=black>'+str(row[3])+'</td><td><font size='' color=black>'+str(row[4])+'</td><td><font size='' color=black>'+str(row[5])+'</td><td><font size='' color=black>'+row[6]+'</td><td><font size='' color=black>'+row[7]+'</td><td><font size='' color=black>'+row[8]+'</td><td><font size='' color=black>'+row[9]+'</td><td>'+str(b.hash)+'</td></tr><tr>'

context= {'data':strdata}

return render(request, 'ViewData.html', context)

def AgencyLogin(request):

if request.method == 'POST':

username = request.POST.get('t1', False)

password = request.POST.get('t2', False)

utype='none'

for i in range(len(blockchain.chain)):

if i > 0:

b = blockchain.chain[i]

data = b.transactions[0]

data = base64.b64decode(data)

data = str(decrypt(data))

data = data[2:len(data)-1]

print(data)

**CMRTC**

**18**

**Decentralization and the security issues in blockchain enabled internet of things**

arr = data.split("#")

if arr[0] == "agency":

if arr[1] == username and arr[2] == password:

utype = arr[6]

break

if utype != 'none':

file = open('session.txt','w')

file.write(utype)

file.close()

context= {'data':'welcome '+username}

return render(request, 'AgencyScreen.html', context)

else:

context= {'data':'login failed'}

return render(request, 'Agency.html', context)

def PatientDataAccess(request):

if request.method == 'GET':

user = ''

with open("session.txt", "r") as file:

for line in file:

user = line.strip('\n')

file.close()

strdata = '<table border=1 align=center width=100%><tr><th><font size='' color=black>Patient ID</font></th><th><font size='' color=black>Patient Name</th>'

strdata+='<th><font size='' color=black>Age</th><th><font size='' color=black>Problem Description</th><th><font size='' color=black>Profile Date</th>'

strdata+='<th><font size='' color=black>Access Control</th><th><font size='' color=black>Gender</th><th><font size='' color=black>Contact No</th></th></tr><tr>'

for i in range(len(blockchain.chain)):

if i > 0:

b = blockchain.chain[i]

data = b.transactions[0]

data = base64.b64decode(data)

data = str(decrypt(data))

data = data[2:len(data)-1]

row = data.split("#")

if row[0] == "patients":

arr = row[6].split(" ")

if len(arr) == 1:

if arr[0] == user:

strdata+='<td>'+str(row[1])+'</td><td>'+row[2]+'</td><td>'+str(row[3])+'</td><td>'+str(row[4])+'</td><td>'+str(row[5])+'</td><td>'+row[6]+'</td><td>'+row[7]+'</td><td>'+row[8]+'</td></tr><tr>'

if len(arr) == 2:

if arr[0] == user or arr[1] == user:

**CMRTC**

**19**

**Decentralization and the security issues in blockchain enabled internet of things**

strdata+='<td>'+str(row[1])+'</td><td>'+row[2]+'</td><td>'+str(row[3])+'</td><td>'+str(row[4])+'</td><td>'+str(row[5])+'</td><td>'+row[6]+'</td><td>'+row[7]+'</td><td>'+row[8]+'</td></tr><tr>'

if len(arr) == 3:

if arr[0] == user or arr[1] == user or arr[2] == user: strdata+='<td>'+str(row[1])+'</td><td>'+row[2]+'</td><td>'+str(row[3])+'</td><td>'+str(row[4])+'</td><td>'+str(row[5])+'</td><td>'+row[6]+'</td><td>'+row[7]+'</td><td>'+row[8]+'</td></tr><tr>'

context= {'data':strdata}

return render(request, 'ViewAccessData.html', context)

def getProfileID():

count = 0

for i in range(len(blockchain.chain)):

if i > 0:

b = blockchain.chain[i]

data = b.transactions[0]

data = base64.b64decode(data)

data = str(decrypt(data))

data = data[2:len(data)-1]

print(data)

arr = data.split("#")

if arr[0] == "patientprofile":

count = count + 1

count = count + 1

return count

def CreateProfileData(request):

if request.method == 'POST':

name = request.POST.get('t1', False)

age = request.POST.get('t2', False)

problem = request.POST.get('t3', False)

access\_list = request.POST.getlist('t4', False)

gender = request.POST.get('t5', False)

contact = request.POST.get('t6', False)

address = request.POST.get('t7', False)

count = 0

access = ''

for i in range(len(access\_list)):

access+=access\_list[i]+" "

access = access.strip()

count = getProfileID()

now = datetime.datetime.now()

current\_time = now.strftime("%Y-%m-%d %H:%M:%S")

data = "patients#"+str(count)+"#"+name+"#"+age+"#"+problem+"#"+str(current\_time)+"#"+str(access)+"#"+gender+"#"+contact+"#"+address

enc = encrypt(str(data))

**CMRTC**

**20**

**Decentralization and the security issues in blockchain enabled internet of things**

enc = str(base64.b64encode(enc),'utf-8')

blockchain.add\_new\_transaction(enc)

hash = blockchain.mine()

b = blockchain.chain[len(blockchain.chain)-1]

print("Encrypted Data : "+str(b.transactions[0])+" Previous Hash : "+str(b.previous\_hash)+" Block No : "+str(b.index)+" Current Hash : "+str(b.hash))

bc = "Encrypted Data : "+str(b.transactions[0])+" Previous Hash : "+str(b.previous\_hash)+"<br/>Block No : "+str(b.index)+"<br/>Current Hash : "+str(b.hash)

blockchain.save\_object(blockchain,'blockchain\_contract.txt')

context= {'data':'Signup process completd and record saved in Blockchain with below hashcodes.<br/>'+bc}

return render(request, 'CreateProfile.html', context)

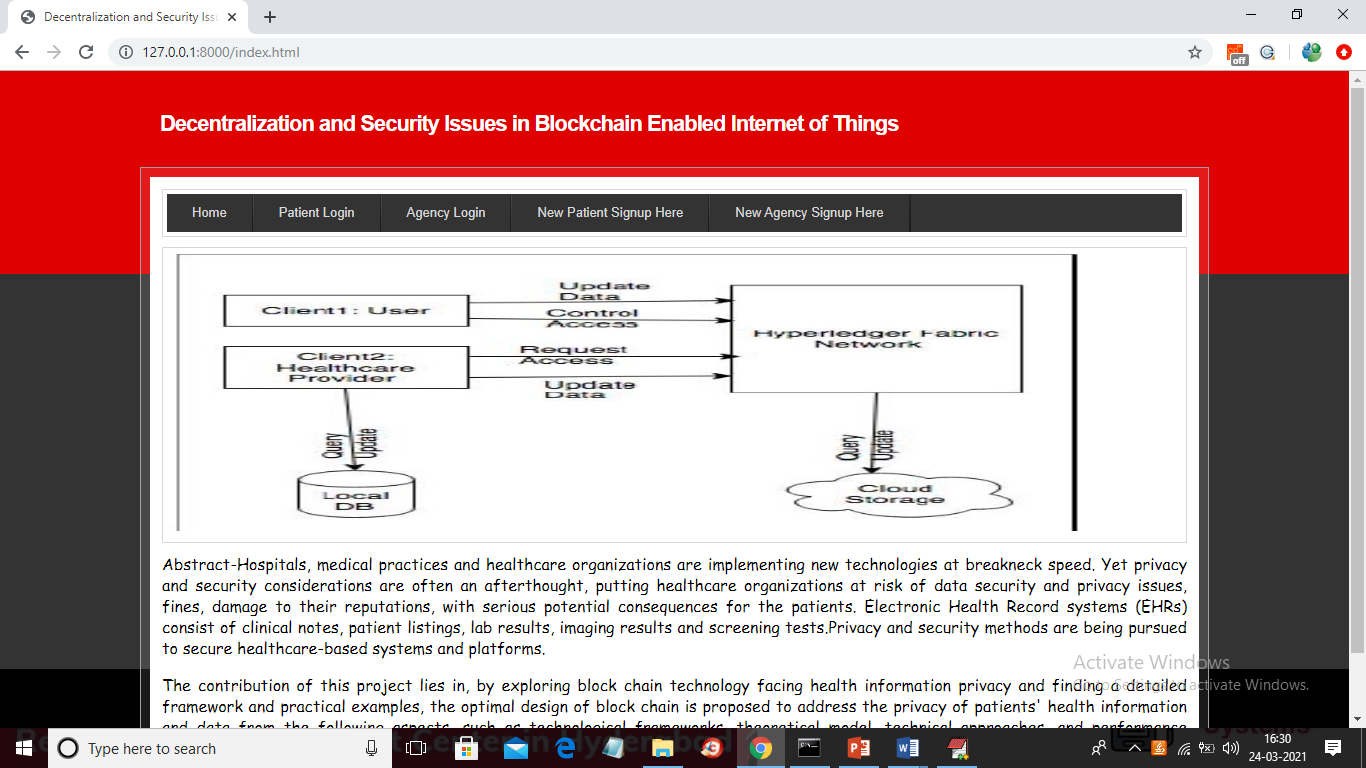
**CMRTC**

**21**

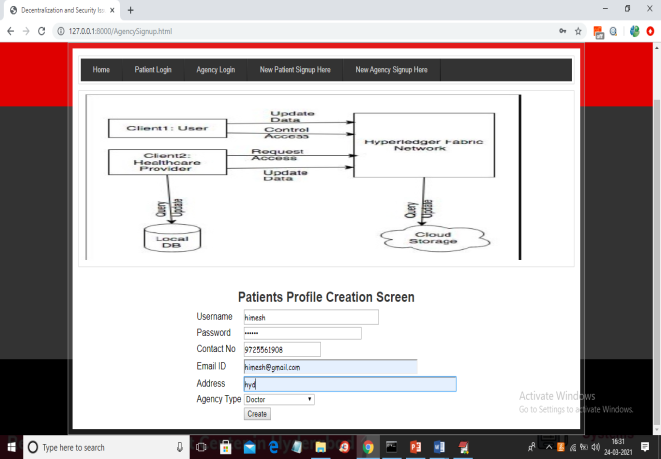
**6. RESULTS**

**Decentralization and the security issues in blockchain enabled internet of things**

**6. SCREENSHOTS**



**Fig 6.1: New Agency Signup**

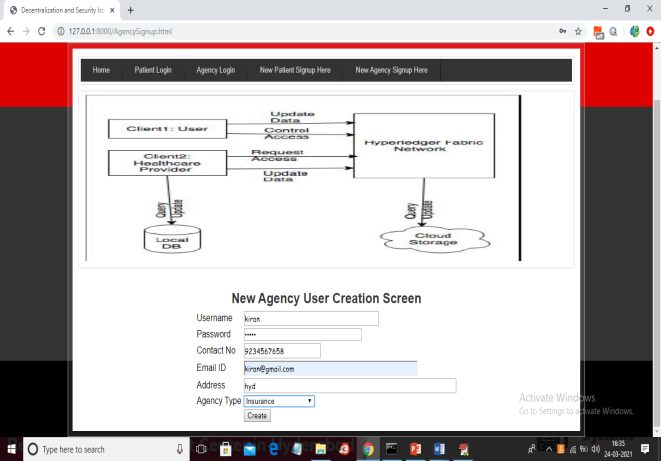


**Fig 6.2: Patient Profile Creation**

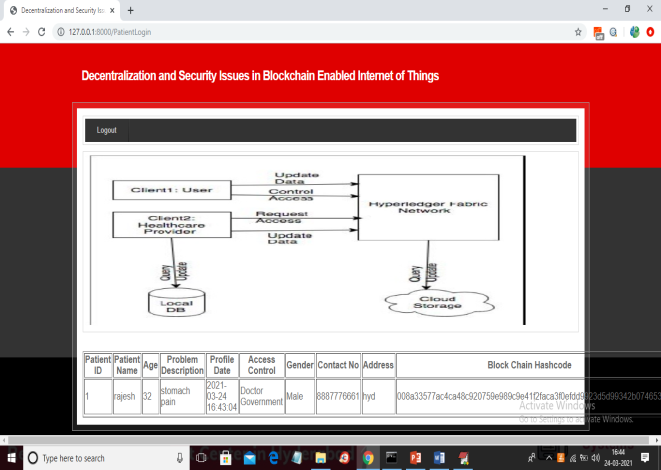
**22**

**CMRTC**

**Decentralization and the security issues in blockchain enabled internet of things**



**Fig 6.3: New Agency Creation**

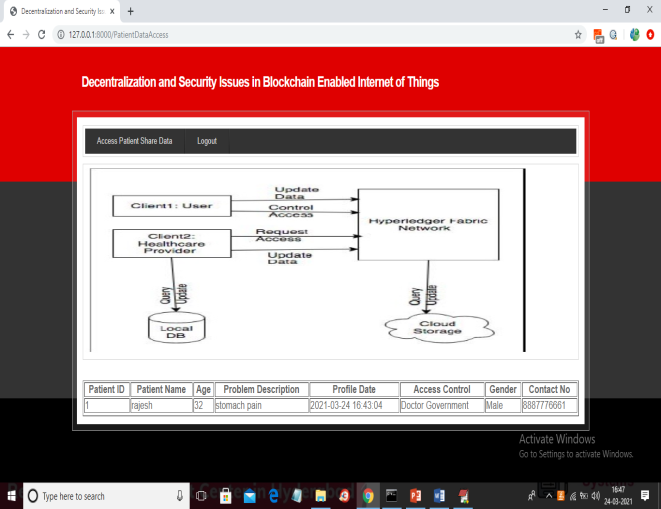


**Fig 6.4: Patient Login**

**CMRTC**

**23**

**Decentralization and the security issues in blockchain enabled internet of things**



**Fig 6.5: Access Patient Share Data**

**CMRTC**

**24**

**7. TESTING**

**Decentralization and the security issues in blockchain enabled internet of things**

### 7. TESTING

**INTRODUCTION TO TESTING**

An estimate says that 50% of whole software development process should be tested. The errors that are occurred may destroy the entire software. Software testing is done while coding by the developers and through testing is conducted by testing experts at various level of code such as module testing, program testing, in-house testing and testing the product at user’s end. Early discovery of errors and their remedy is the key to reliable software.

### TYPES OF TESTING

### 7.1.1 UNIT TESTING

Unit testing involves the design of test cases that validate that the internal programlogic 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.

### 7.1.2 INTEGRATION TESTING

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

**CMRTC**

**25**

**Decentralization and the security issues in blockchain enabled internet of things**

### 7.1.3 FUNCTIONAL TESTING

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.

**26**

**CMRTC**

**Decentralization and the security issues in blockchain enabled internet of things**

### 7.2 TEST CASES

**7.2.1 CLASSIFICATION**

|  |  |  |  |
| --- | --- | --- | --- |
| **TEST ID** | **TEST CASE NAME** | **TEST INPUT** | **TEST RESULT** |
| 1. | Patient  Sign Up | If entered “valid” details  Such as Name, Email, Address, Disease and to whom patient have to share data  . | PASS |
| 2. | Patient  Sign Up | If entered “In-valid” details Such as Name, Email, Address, Disease and to whom patient have to share data | FAIL |
| 3. | Agency  Sign Up | If entered “valid” details  Such as Username, Password, Contact, Email, Address, Agency type | PASS |
| 4. | Agency  Sign Up | If entered “In-valid” details Such as Username, Password, Contact, Email, Address, Agency type. | FAIL |
| 5. | Agency  Login | If entered “valid” Username and Password. | PASS |
| 6. | Agency  Login | If entered “In-valid” Username and Password | FAIL |

**27**

**CMRTC**

**8. CONCLUSION**

**Decentralization and the security issues in blockchain enabled internet of things**

**8. CONCLUSION**

IoT security and privacy are critical success factors for meeting the high expectations of the technology to transform many aspects of our society and economy. Our proposed blockchain based IoT architecture handles most security and privacy threats, while considering the resource-constraints of many IoT devices. Our qualitative overhead analysis of the the architecture has shown that it has constant performance overhead at best, and at worst most of its transactions scale with the number of clusters in the network, rather than the number of nodes.

Decentralized Blockchain server which maintain data as blocks of trees and at each transaction all previous hash code will be verified and if verification successful then data will be consider as intact and if data changes then Blockchain server will notify that system in under attack and it gather data from another working node. Due to this transaction hash code verification and immutable data storage make Blockchain secure and trustable in current market.

**CMRTC**

**28**

**9. BIBLIOGRAPHY**

**Decentralization and the security issues in blockchain enabled internet of things**

**9. BIBLIOGRAPHY**

**9.1 REFERENCES**

1. Ouaddah, A.; Mousannif, H.; Elkalam, A.A.; Ouahman, A.A. Access control in the Internet of Things: Big challenges and new opportunities. Comput. Netw. 2017, 112, 237–262.

2. Aldowah, H.; Rehman, S.U.; Umar, I. Security in Internet of Things: Issues, Challenges and Solutions. In International Conference of Reliable Information and Communication Technology; Springer: Cham, Switzerland, 2018; pp. 396–405.

3. Ourad, A.Z.; Belgacem, B.; Salah, K. Using blockchain for IOT access control and authentication management. In InternationalConference on Internet of Things 2018 June; Springer: Cham, Switzerland, 2018; pp. 150–164.

4. Ravidas, S.; Lekidis, A.; Paci, F.; Zannone, N. Access control in Internet-of-Things: A survey. J. Netw. Comput. Appl. 2019,144, 79–101.

5. Ouaddah, A.; Mousannif, H.; Ouahman, A.A. Access control models in loT: The road ahead. In Proceedings of the 2015 IEEE/ACS 12th International Conference of Computer Systems and Applications (AICCSA), Marrakech, Morocco, 17–20 November 2016;pp. 272–277.

6. Jia, J.; Qiu, X.; Cheng, C. Access control method for web of things based on role and SNS. In Proceedings of the 2012 IEEE

12th International Conference on Computer and Information Technology, Chengdu, China, 27–29 October 2012; pp. 316–321.

**CMRTC**

**29**