A DECENTRALIZED APPLICATION FOR SECURE MESSAGING IN A TRUSTLESS ENVIRONMENT

PROJECT BATCH NUMBER:CSEMPBNUM_K10

PROJECT TEAM

K. SHASHANK SINGH (2215316131)

B. SUMANTH REDDY (2215316165)

CHANDU GIRI SHASHANK (2215316117)

T. CHAITANYA (2215316156)

Abstract

Blockchain technology has been seeing widespread interest as a means to ensure the integrity, confidentiality and availability of data in a trustless environment. They are designed to protect data from both internal and external cyberattacks by utilizing the aggregated power of the network to resist malicious efforts. In this project, we will create our own decentralized messaging application utilizing the Ethereum Whisper protocol. Our application will be able to send encrypted messages both securely and anonymously. We will utilize the Ethereum platform to deploy our blockchain network. This application would be resistant to most suppression tactics due to its distributed nature and adaptability of its communication protocol.

Introduction

In recent times, it is becoming increasingly vital that communication not only be secure but also anonymous. The presence of mass surveillance programs as well as cyberattacks focused on compromising messaging applications highlight the need for maintaining the anonymity of communicating parties. In this article, we developed a decentralized messenger application utilizing the Ethereum Whisper protocol. Using our application, two users can engage in secure and anonymous communication which is encrypted end-to-end and resistant to network traffic analysis.

Background and definitions

- *Secure Messaging*

A significant amount of current electronic communication is still placed over a number of legacy protocols such as SMS/GSM, SMTP and centralized messengers which were not designed with end-to-end security as a requirement. These methods routinely broadcast recipient and sender information and therefore provide limited anonymity capabilities. In addition, they are more prone to suppression due to the storage and transmission requirements by intermediate servers.

- *Blockchain*

A blockchain is an append-only distributed database operating within a P2P network whereby each peer has a partial or full copy of the blockchain. Due to their distributed nature, blockchains are highly available and fault to learn even if a large scale attack is mounted on the network. Changes in the blockchain are conducted through transactions which are broadcasted and verified by all the nodes in the network. After verification the change is appended to the blockchain.

First block in the blockchain is called the **Genesis Block** and has no **Prev Block Hash** value.

- Django Framework

Django is an extremely popular and fully featured server-side web framework, written in Python. This module shows you why Django is one of the most popular web server frameworks, how to set up a development environment, and how to start using it to create your own web applications.

Ridiculously Fast:

Django was designed to help developers take application from the concept to completion as quickly as possible.

Reassuringly Secure:

Django takes security seriously and helps developers avoid many common security mistakes.

Exceedingly Scalable:

Some of the busiest sites on the Web leverage Django's ability to quickly and flexibly scale

SYSTEM DESIGN

PROBLEM STATEMENTS

End-to-End encryption: Only the users should be able to decipher the data being stored or communicated.

Anonymous Sender: The source of a particular message cannot be attributed to a specific entity.

Anonymous recipient: The destination of a particular message cannot be attributed to a specific entity.

Anonymous Participants: The set of participants engaged in a conversation cannot be determined.

Unlinkability: Only the participants engaged in a conversation are able to associate two or more protocol messages as belonging to the same conversation.

Resistant to Flood and Spam attacks: Bulk messaging and DoS attacks should not be able to significantly impact the availability of the system.

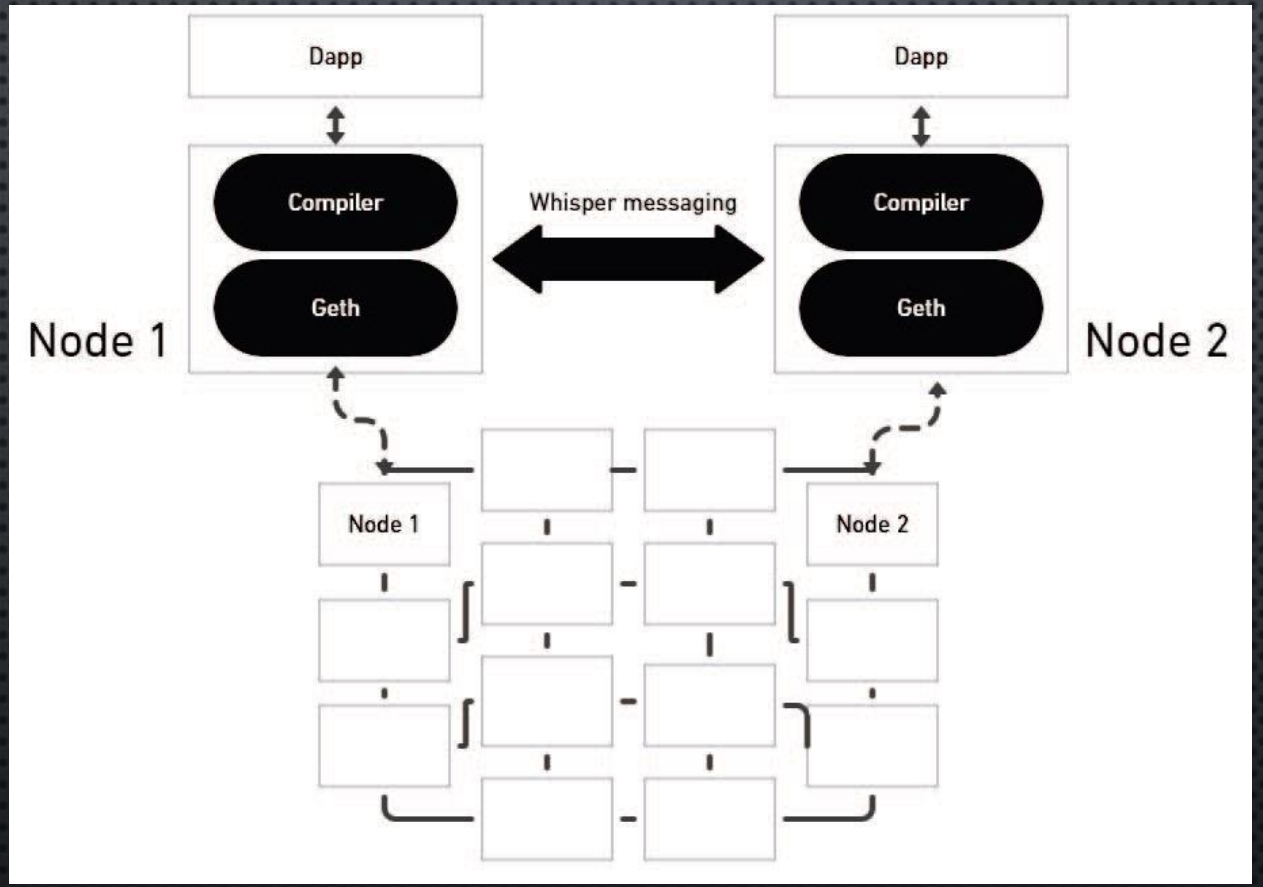
Plausible deniability: It should be possible for the any entity to deny having sent a particular message.

- *Software Architecture*

The client uses Geth, an Ethereum client, to run a node and to serve as an interface to interact with the Whisper network. The front end consists of a web application built on Node.js and was chosen due to the abundant support for web3.js,

1) *Geth:* the command line interface for running a Go implementation of a full ethereum node.
2) *Node.js:* a JavaScript runtime environment used for building and executing event-driven, scalable applications.
3) *React.js:* a JavaScript library used for building user interfaces.
4) *Redux.js:* a JavaScript library used for managing application state.
5) *Application*
    a) *Account Management:* included various functions such as creating accounts and adding/removing friends.
    b) *Messaging:* allows the user to send encrypted messages to peers in the Whisper network.
6) *web3.js:* is an Ethereum JavaScript API used to interact with the Ethereum and Whisper Network through a local Ethereum or Whisper node.
7) *JSON-RPC:* is a light-weight, stateless remote procedure call (RPC) protocol.
8) *Whisper:* it is Ethereum's P2P communication protocol for decentralized applications.

SOFTWARE NETWORK ARCHITECTURE



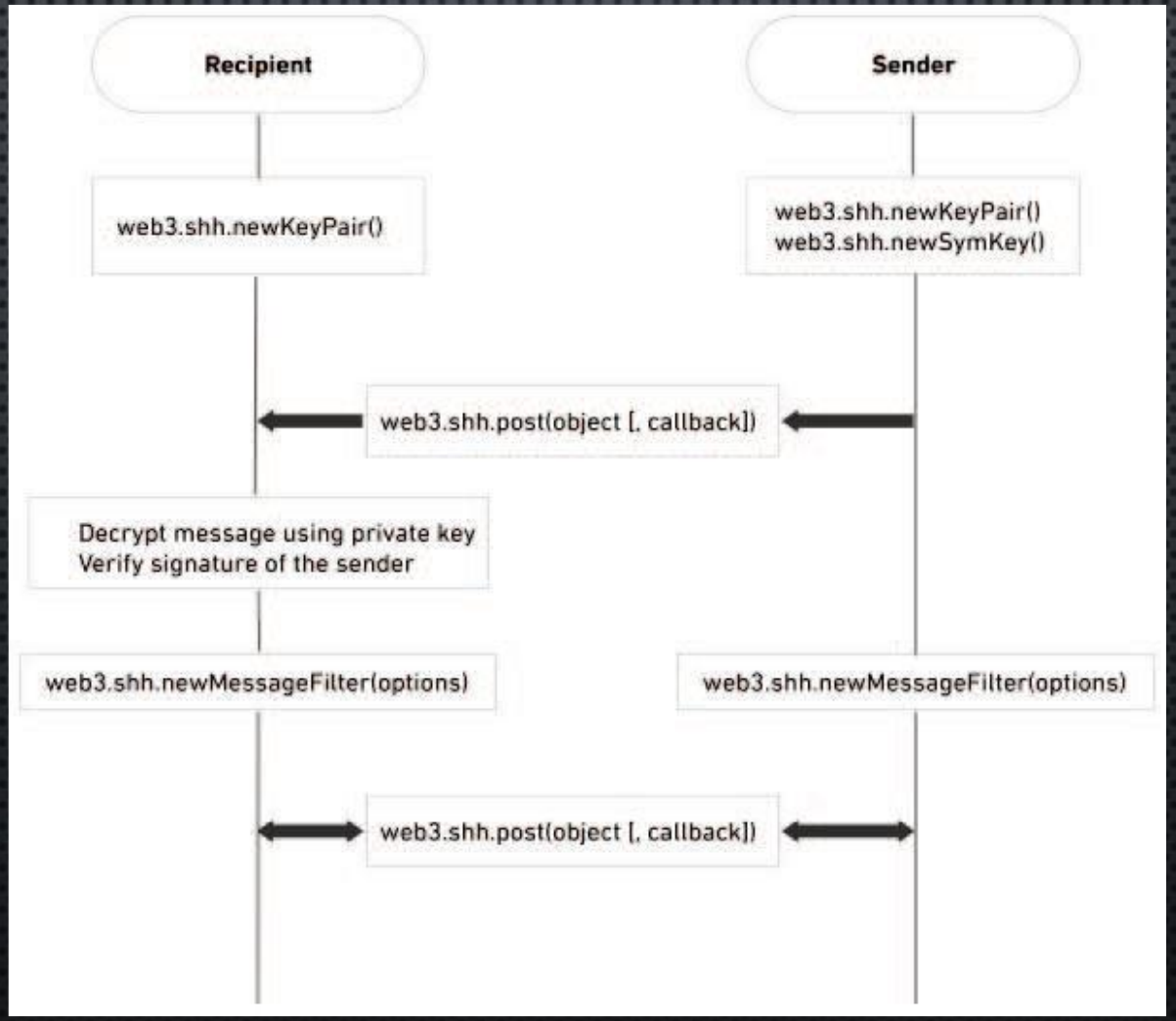
- *Operation*

Both the sender and recipient should have a generated asymmetric key pair. The sender initially transmits a message which is asymmetrically encrypted with the recipient's public key. The message contains a randomly generated symmetric key and a partial Topic for each participant in the chat. Topics are probabilistic filters known as bloom filters used to partially classify the message without compromising security. Nodes can filter for messages that are probably meant for them using the partial Topic.

After the recipient decrypts the message using their private key. The recipient verifies the identity of the sender and then retrieves the symmetric keys in the message.

The receiver subscribes for messages with the first partial Topic and the sender subscribes for messages with the second partial Topic. The sender encrypts outgoing messages using the first symmetric key while the receiver encrypts outgoing messages with the latter symmetric key.

MESSAGE TRANISMISSION PROCESS



Consensus Algorithm
Proof-of-Work
(PoW)

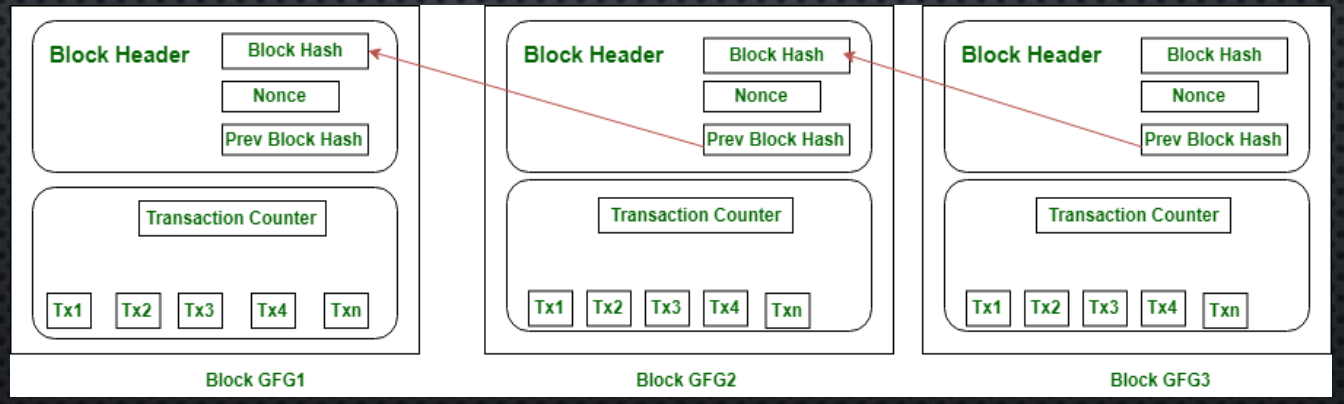
Proof of Work consensus is the mechanism of choice for the majority of cryptocurrencies currently in circulation.

Principle: A solution that is difficult to find but is easy to verify.

The **purpose** of a consensus mechanism is to bring all the nodes in agreement, that is, trust one another, in an environment where the nodes don't trust each other.

First block in the blockchain is called the **Genesis Block** and has no **Prev Block Hash** value.

In PoW, each node of the network is calculating a hash value of the block header. The block header contains a nonce and miners would change the nonce frequently to get different hash values. The consensus requires that the calculated value must be equal to or smaller than a certain given value. When one node reaches the target value, it would broadcast the block to other nodes and all other nodes must mutually confirm the correctness of the hash value. If the block is validated, other miners would append this new block to their blockchains.



working of PoW algorithm

Encryption algorithm

Sha-512 is a function of cryptographic algorithm Sha-2, which is an evolution of famous Sha-1. Sha-512 is very close to its "brother" Sha-256 except that it used 1024 bits "blocks", and accept as input a 2^128 bits maximum length string. Sha-512 also has others algorithmic modifications in comparison with Sha-256. Sha-512 is very secure, but also takes a lot of database space.

As the sender's signature is part of the message and can only be accessed after decryption, the signature can only be accessed by the recipient. Subsequent messages between the sender and recipient are encrypted using the shared symmetric keys.

After every block of 1024 bits goes through the message processing phase, i.e. the last iteration of the phase, we get the final 512 bit Hash value of our original message. So, the intermediate results are all used from each block for processing the next block. And when the final 1024 bit block has finished being processed, we have with us the final result of the SHA-512 algorithm for our original message.

Resistance to Denial of Service Attacks

Since every Whisper message is routed to every node that it is able to reach, the network might be susceptible to

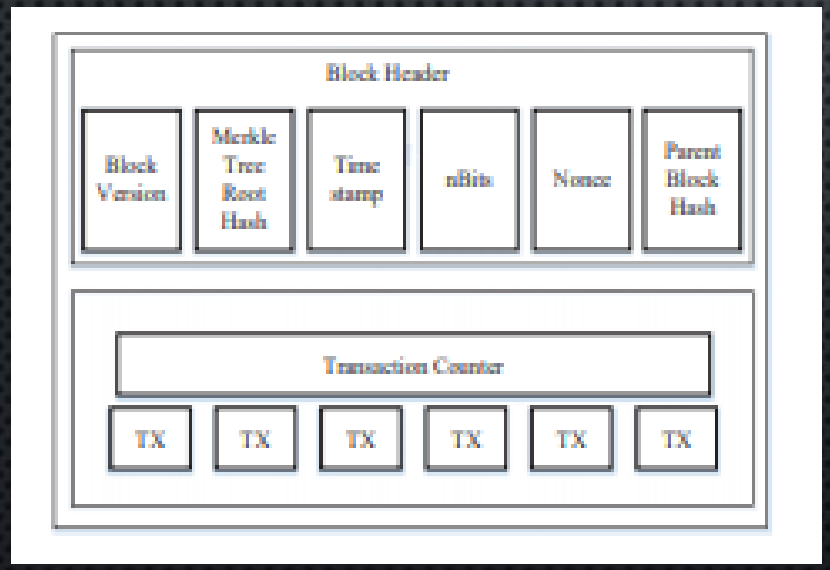
two types of attacks:

1. Expiry Attack: Messages are set to have a long Time-to-Live (TTL) on the envelope.
2. Flood Attack: The network is repeatedly sent messages.

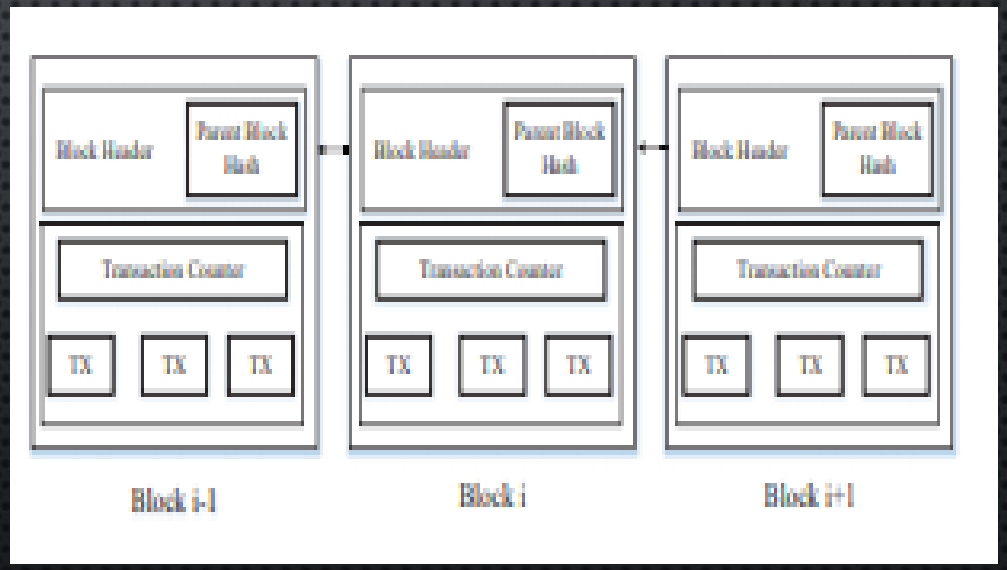
An expiry attack is averted by using a system that rates messages by taking into account the message size, TTL and PoW. Messages that have a smaller size, lower TTL and higher PoW are considered to have a higher rating. This rating impacts how long the message is stored as well as its forwarding priority. Lower rated messages would be removed first in the event of a DoS attack, thus preventing an expiry attack.

BLOCKCHIAN ARCHITECTURE

A BLOCKCHAIN IS A SEQUENCE OF BLOCKS, WHICH HOLDS A COMPLETE LIST OF TRANSACTION RECORDS LIKE CONVENTIONAL PUBLIC LEDGER. FIGURE ILLUSTRATES AN EXAMPLE OF A BLOCKCHAIN. IT IS WORTH NOTING THAT UNCLE BLOCKS (CHILDREN OF THE BLOCK'S ANCESTORS) HASHES WOULD ALSO BE STORED IN THE ETHEREAL BLOCKCHAIN. THE FIRST BLOCK OF A BLOCKCHAIN IS CALLED GENESIS BLOCK WHICH HAS NO PARENT BLOCK.



BLOCK STRUCTURE



CONTINUOUS SEQUENCE OF BLOCKS

Block: A block consists of the block header and the block body as shown in Figure. In particular, the block header includes:

- (i) Block version: indicates which set of block validation rules to follow.
- (ii) Merkle tree root hash: the hash value of all the transactions in the block.
- (iii) Timestamp: current time as seconds in the universal time since January 1, 1970.
- (iv) NBits: target threshold of a valid block hash.
- (v) Nonce: a 4-byte field, which usually starts with 0 and increases for every hash calculation (will be explained in detail in Section III).
- (vi) Parent block hash: a 256-bit hash values that point to the previous block. The block body is composed of a transaction counter and transactions. The maximum number of transactions that a block can contain depends on the block size and the size of each transaction. The Blockchain uses an asymmetric cryptography mechanism to validate the authentication of transactions. Digital signature based on asymmetric cryptography is used in an untrustworthy environment. Next, briefly illustrate the digital signature.

Digital Signature: Each user owns a pair of private key and public key. The private key that shall be kept in confidentiality is used to sign the transactions. The digitally signed transactions are broadcasted throughout the whole network.

The typical digital signature is involved with two phases: signing phase and verification phase. For instance, a user Alice wants to send another user Bob a message.

(1) In the signing phase, Alice encrypts her data with her private key and sends Bob the encrypted result and original data.

(2) In the verification phase, Bob validates the value with Alice's public key. In that way, Bob could easily check if the data has been tampered or not.

KEY CHARACTERISTICS OF BLOCKCHAIN:

***DECENTRALIZATION*:** IN CONVENTIONAL CENTRALIZED TRANSACTION SYSTEMS, EACH TRANSACTION NEEDS TO BE VALIDATED THROUGH THE CENTRAL TRUSTED AGENCY (E.G., THE CENTRAL BANK), INEVITABLY RESULTING IN THE COST AND THE PERFORMANCE BOTTLENECKS AT THE CENTRAL SERVERS. IN CONTRAST TO THE CENTRALIZED MODEL, THE THIRD PARTY IS NO LONGER NEEDED IN BLOCKCHAIN. CONSENSUS ALGORITHMS IN BLOCKCHAIN ARE USED TO MAINTAIN DATA CONSISTENCY IN A DISTRIBUTED NETWORK.

***PERSISTENCY*:** TRANSACTIONS CAN BE VALIDATED QUICKLY AND INVALID TRANSACTIONS WOULD NOT BE ADMITTED BY HONEST MINERS. IT IS NEARLY IMPOSSIBLE TO DELETE OR ROLLBACK TRANSACTIONS ONCE THEY ARE INCLUDED IN THE BLOCKCHAIN. BLOCKS THAT CONTAIN INVALID TRANSACTIONS COULD BE DISCOVERED IMMEDIATELY.

***ANONYMITY*:** EACH USER CAN INTERACT WITH THE BLOCKCHAIN WITH A GENERATED ADDRESS, WHICH DOES NOT REVEAL THE REAL IDENTITY OF THE USER. NOTE THAT BLOCKCHAIN CANNOT GUARANTEE THE PERFECT PRIVACY PRESERVATION DUE TO THE INTRINSIC CONSTRAINT.

Taxonomy of Blockchain systems

Current Blockchain systems is categorized roughly into three types:

1.public Blockchain

2.private Blockchain

3.consortium Blockchain.

In the public Blockchain, all records are visible to the public and everyone could take part in the consensus process. Differently, only a group of pre-selected nodes would participate in the consensus process of a consortium Blockchain. As for the private Blockchain, only those nodes that come from one specific organization would be allowed to join the consensus process.

A private Blockchain is regarded as a centralized network since it is fully controlled by one organization. The consortium Blockchain constructed by several organizations is partially decentralized since only a small portion of nodes would be selected to determine the consensus.

The comparison among the three types of Blockchains is listed in.

Consensus determination: In the public Blockchain, each node could take part in the consensus process. And only a selected set of nodes is responsible for validating the block in the consortium Blockchain. As for the private chain, it is fully controlled by one organization and the organization could determine the final consensus.

Read permission: Transactions in a public Blockchain are visible to the public while it depends when it comes to a private Blockchain or a consortium Blockchain.

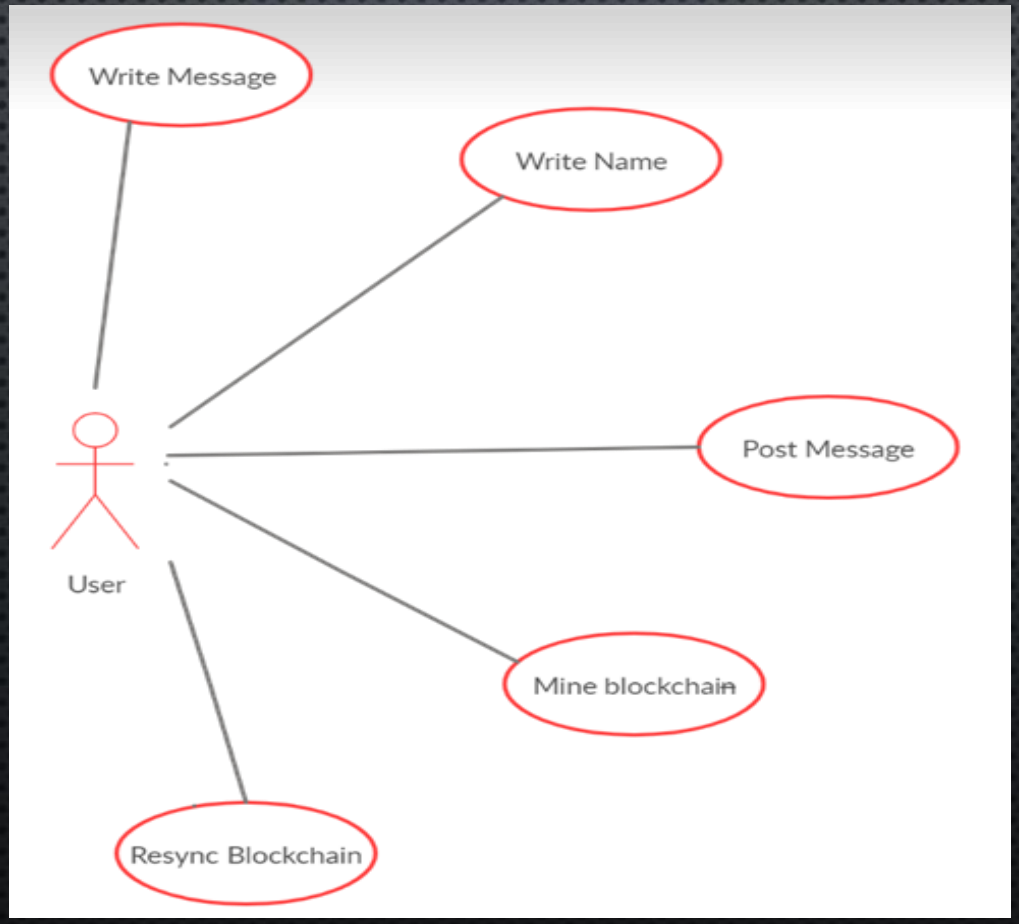
Immutability: Since records are stored on a large number of participants, it is nearly impossible to tamper transactions in a public Blockchain. Differently, transactions in a private Blockchain or a consortium Blockchain could have tampered easily as there is only a limited number of participants.

Efficiency: It takes plenty of time to propagate transactions and blocks as there are a large number of nodes on the public Blockchain network. As a result, transaction throughput is limited and the latency is high. With fewer validates, consortium Blockchain and private Blockchain could be more efficient.

SYSTEM METHODOLOGY

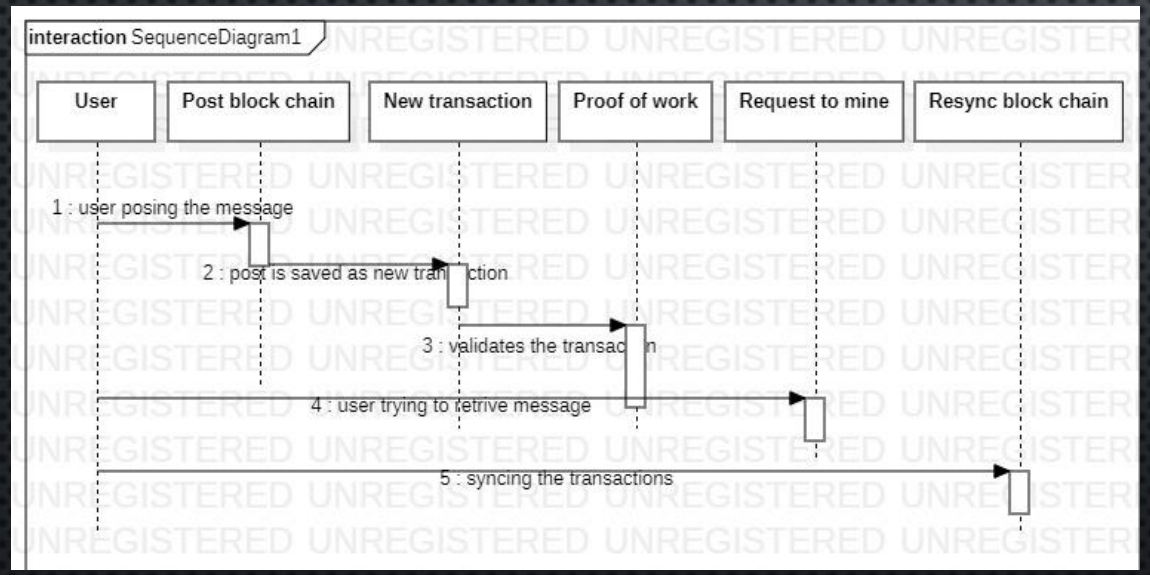
USE CASE DIAGRAM

THE USE CASE DIAGRAM SHOWS ALL THE ACTIONS OF THE USER WHICH ARE TO BE PERFORMED BY THE USER WHILE OPERATING THE APPLICATION.

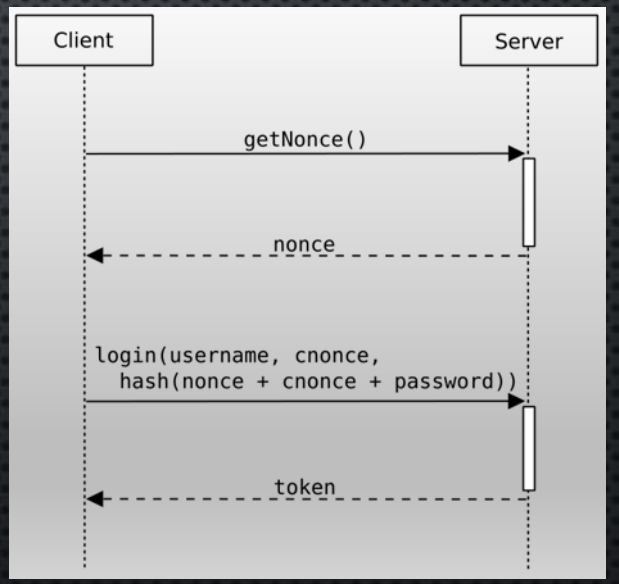


SEQUENCE DIAGRAM

THIS DIAGRAM SHOWS THE SEQUENCE IN WHICH THE APPLICATION IS RUNNING SO IN THIS FIRST THE USER ENTERS THE MESSAGE WHICH IS TO BE MINED AND THEN HE POSTS IT AND THE ALGORITHM AT BACKEND VALIDATES IT AND THEN WHEN USER REQUESTS TO MINE IT IS MINED AND ON RESYNCING THE BLOCKCHAIN THE MESSAGE IS DISPLAYED. EVEN IN THE BELOW DIAGRAM THE CLIENT IS THE USER AND THIS HAPPENS AT THE BACKEND WITH ALGORITHM INSIDE.



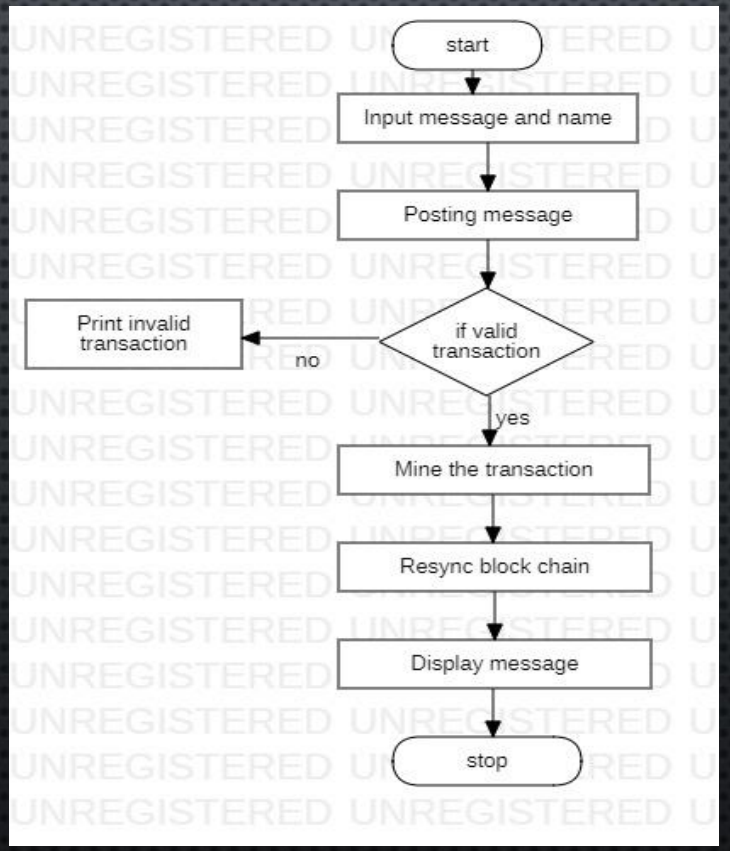
SEQUENCE DIAGRAM OF THE APPLICATION



SEQUENCE OF PROJECT IN BACKEND

ACTIVITY DIAGRAM

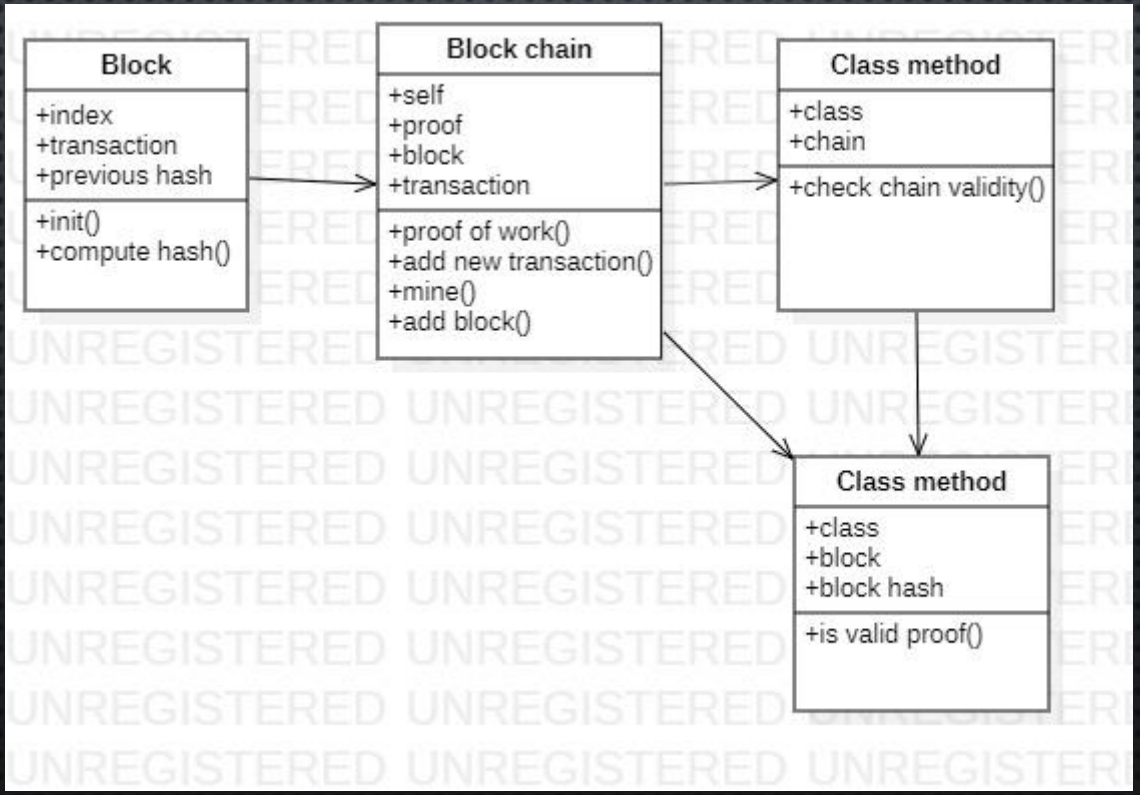
THE DATA IN THE PROJECT FLOWS RIGHT INTO THE ALGORITHM OF THE BLOCKCHAIN FROM THE USER'S POST WHICH IS TO BE MINED AND THEN AFTER THE VALIDATION OF THE TRANSACTION THE BLOCKCHAIN HAS TO BE RESYNCED IN ORDER TO DISPLAY THE MINED MESSAGE FROM THE USER.



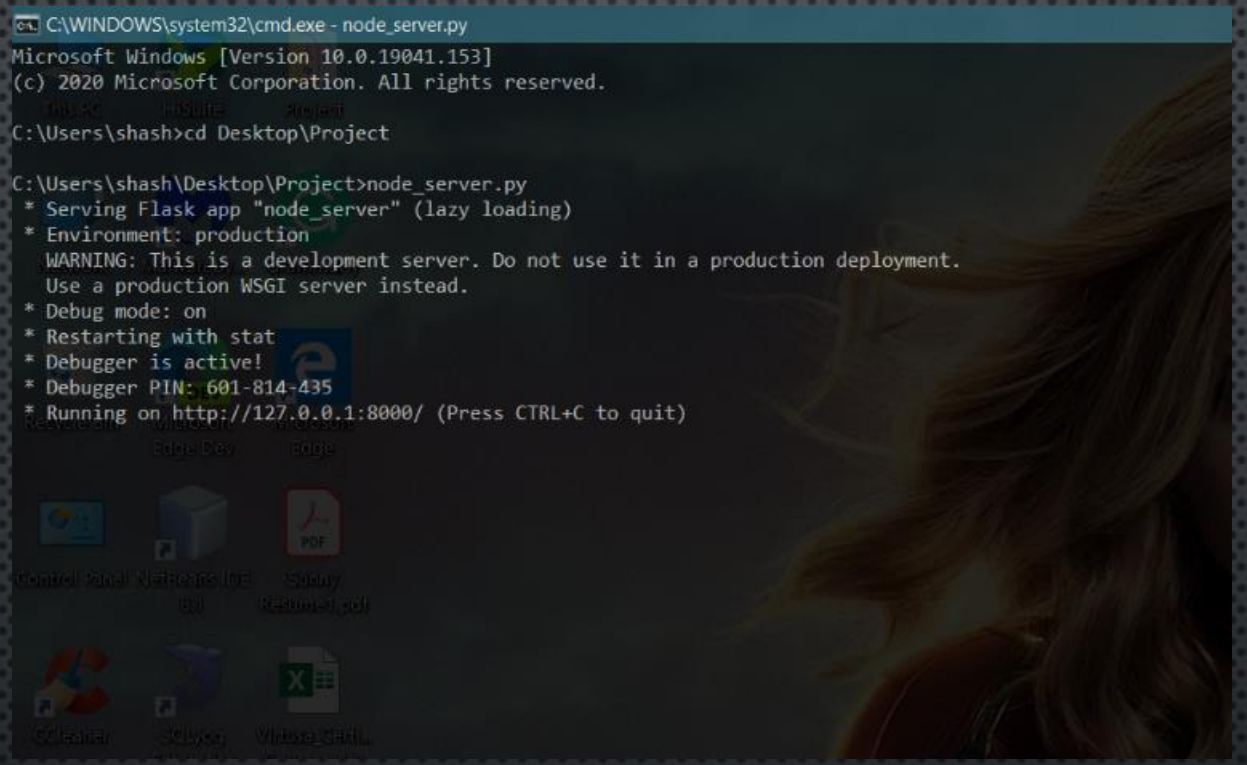
Class diagram

The below mentioned classes are only few classes which are prominent in the blockchain technology

Which contains some primary methods and parameters without which the algorithms won't have the apt functionality though.

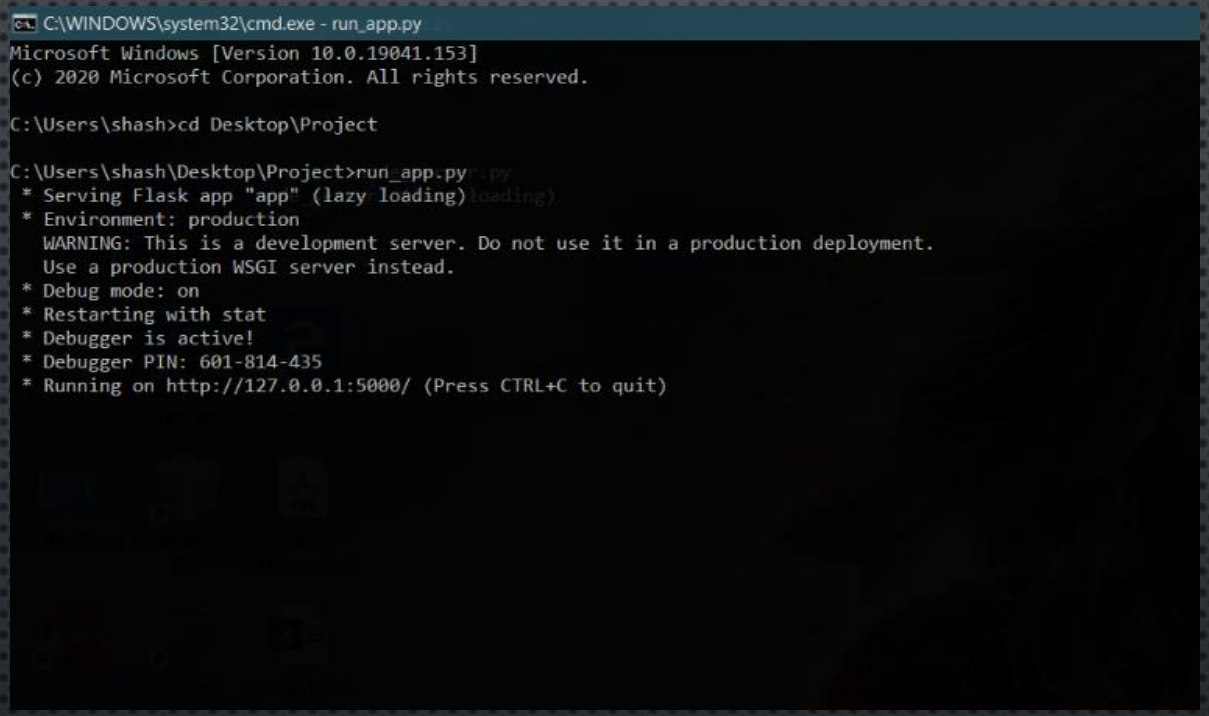


LAUNCHING THE PROJECT



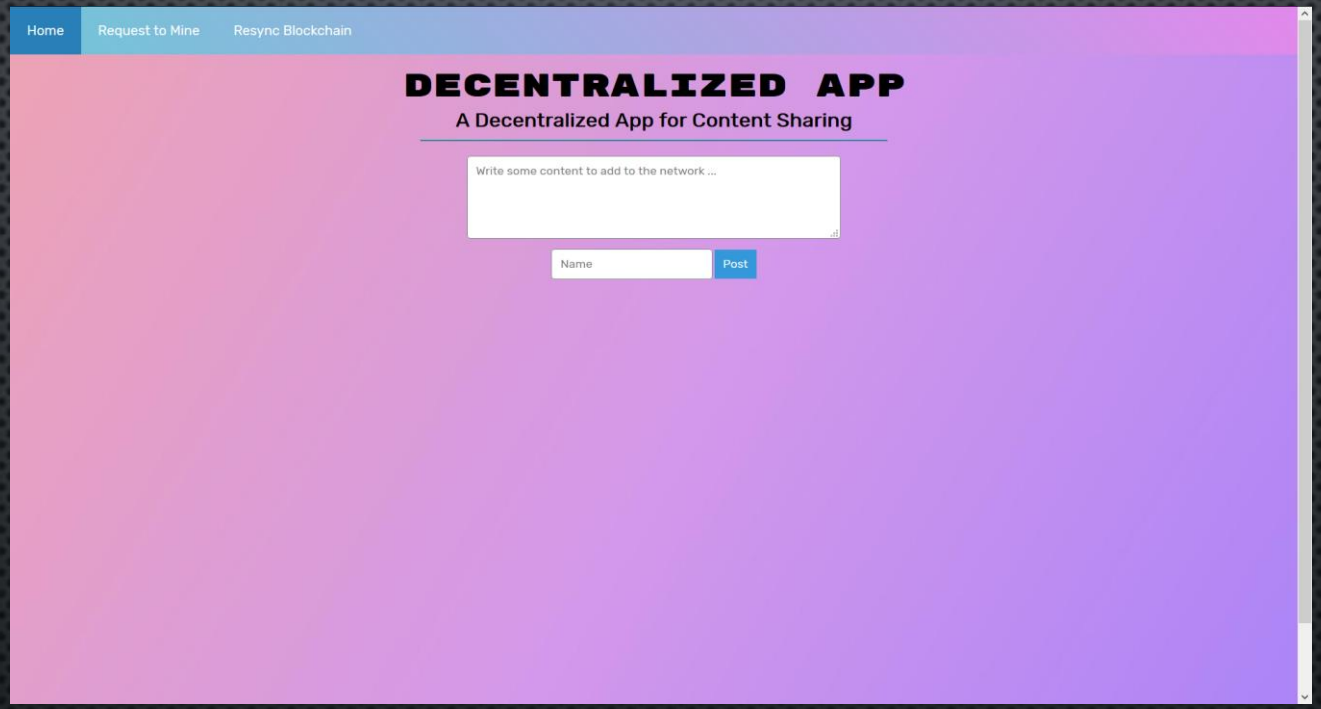
INITIALLY THE LOCAL SERVER IS RUN FOR A BACKEND SUPPORT WHERE THE PROJECT RUNS. THE SERVER CAN BE ACCESSED THROUGH THE GIVEN URL DISPLAYED ON THE TERMINAL.

Launching the application on the server



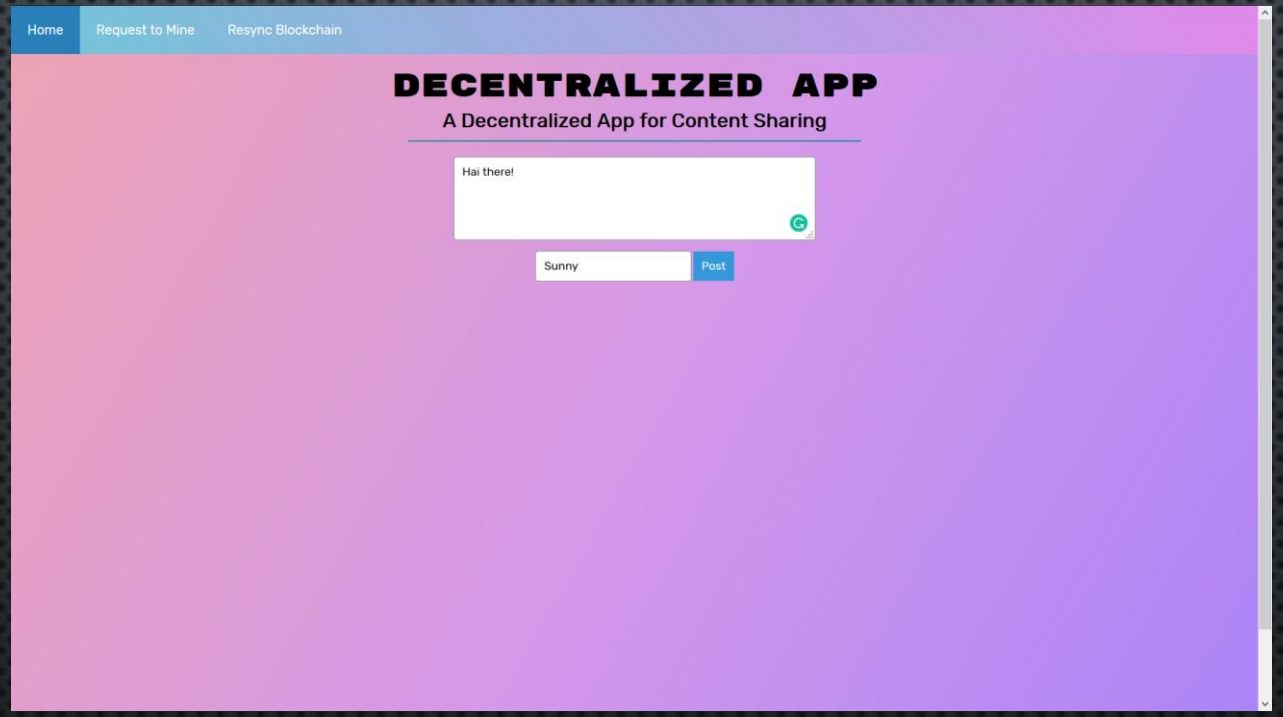
Here when the application is launched on the server started previously we have an URL from where the main application can be accessed through any web browser.

USER INTERACTION PAGE



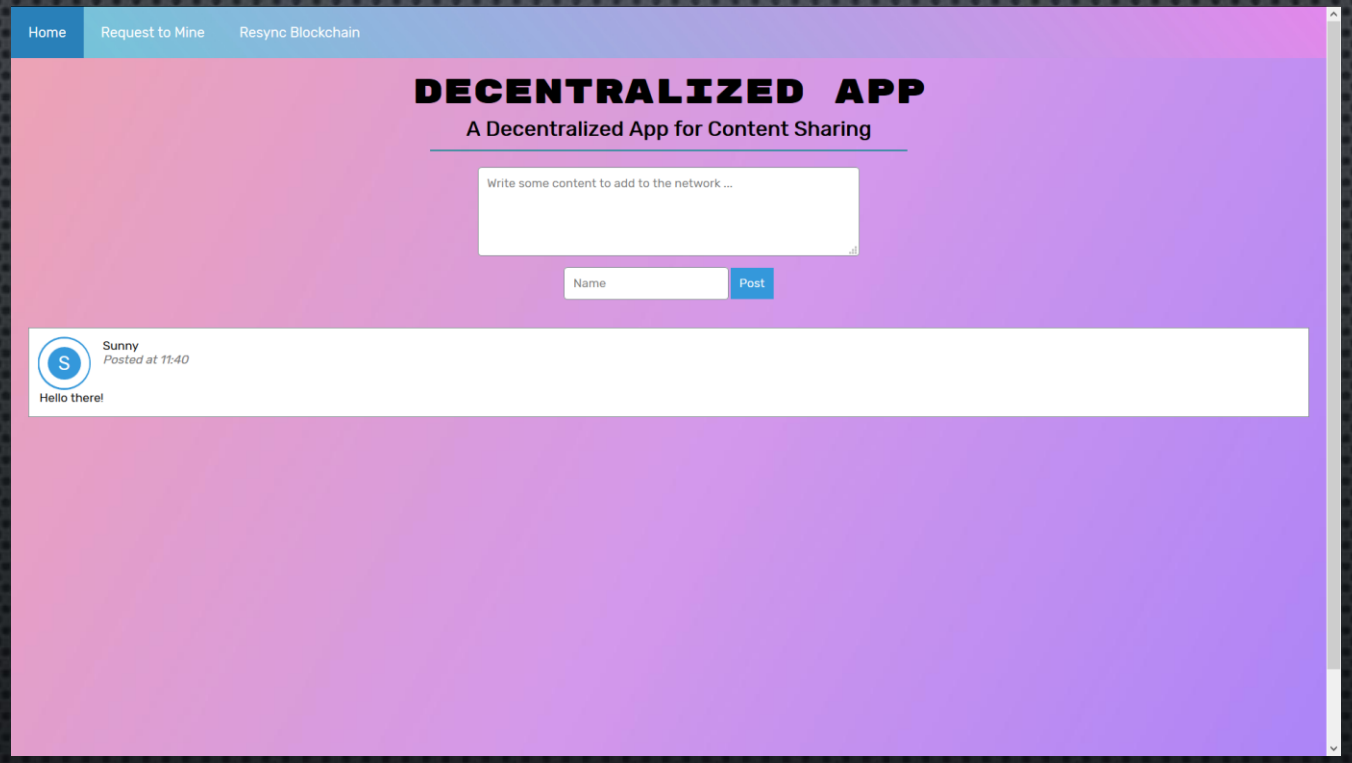
THE INITIAL PAGE WHICH WOULD BE SEEN BY THE USER IS SEEMED TO BE AS SHOWN IN THE ABOVE FIGURE WHERE THE ACTIONS PREFERRED BY THE USER CAN BE PERFORMED.

POSTING A MESSAGE WITH REGISTERING A NAME



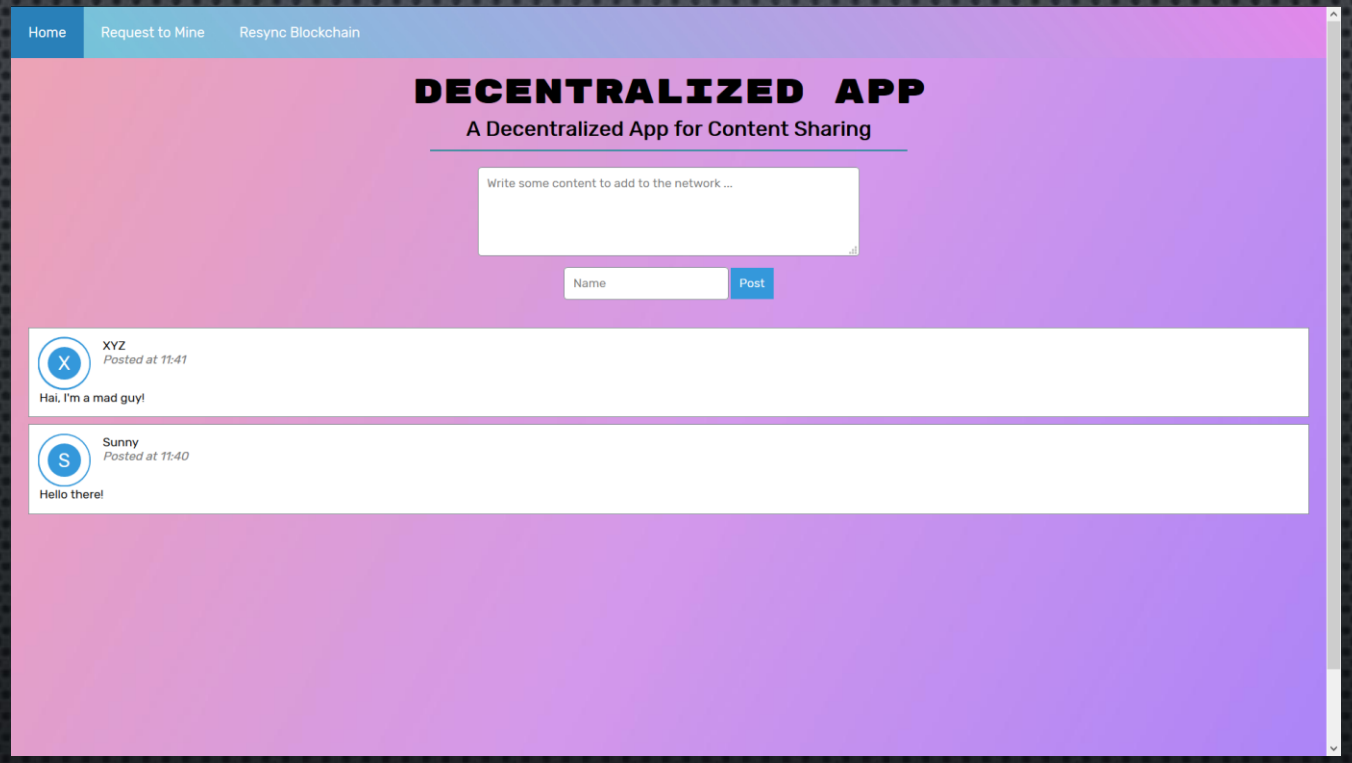
FIRSTLY ANY MESSAGE IS WRITTEN IN THE TEXT AREA PROVIDED IN THE PAGE AND THEN AFTER ENTERING ANY NAME PREFERRED BY THE USER, THE USER CAN POST IT AND THE BLOCKCHAIN IS MINED IN THE BACK-END BY THE REGISTERED NAME.

OUTPUT AFTER RESYNCHRONIZING



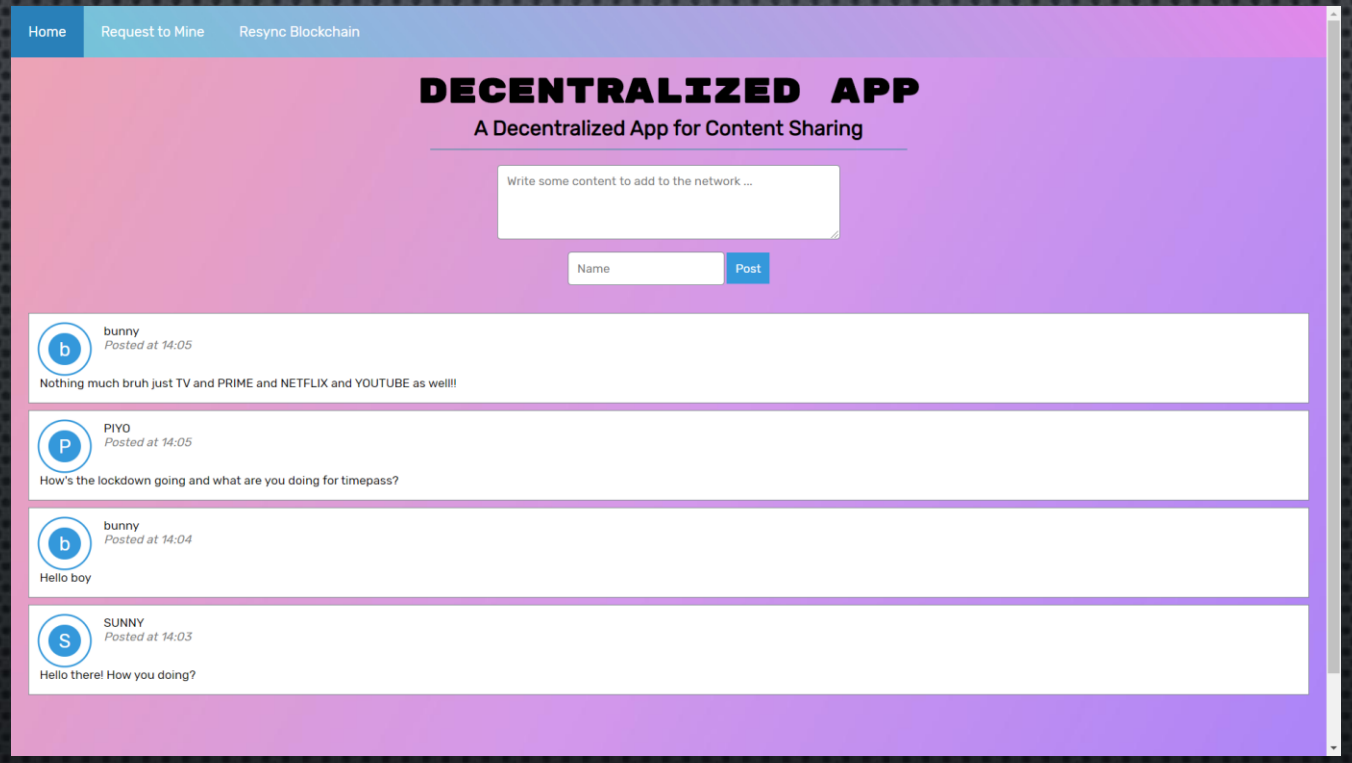
AFTER WE REQUEST TO MINE THE BLOCKCHAIN WHICH IS POSTED BY THE USER THE BLOCKCHAIN IS MINED AND THEN SYNCHRONIZED WHEN THE BUTTON RESYNC BLOCKCHAIN IS CLICKED AND THEN THE MESSAGE IS DISPLAYED AS ABOVE.

SAME STEPS WERE REPEATED



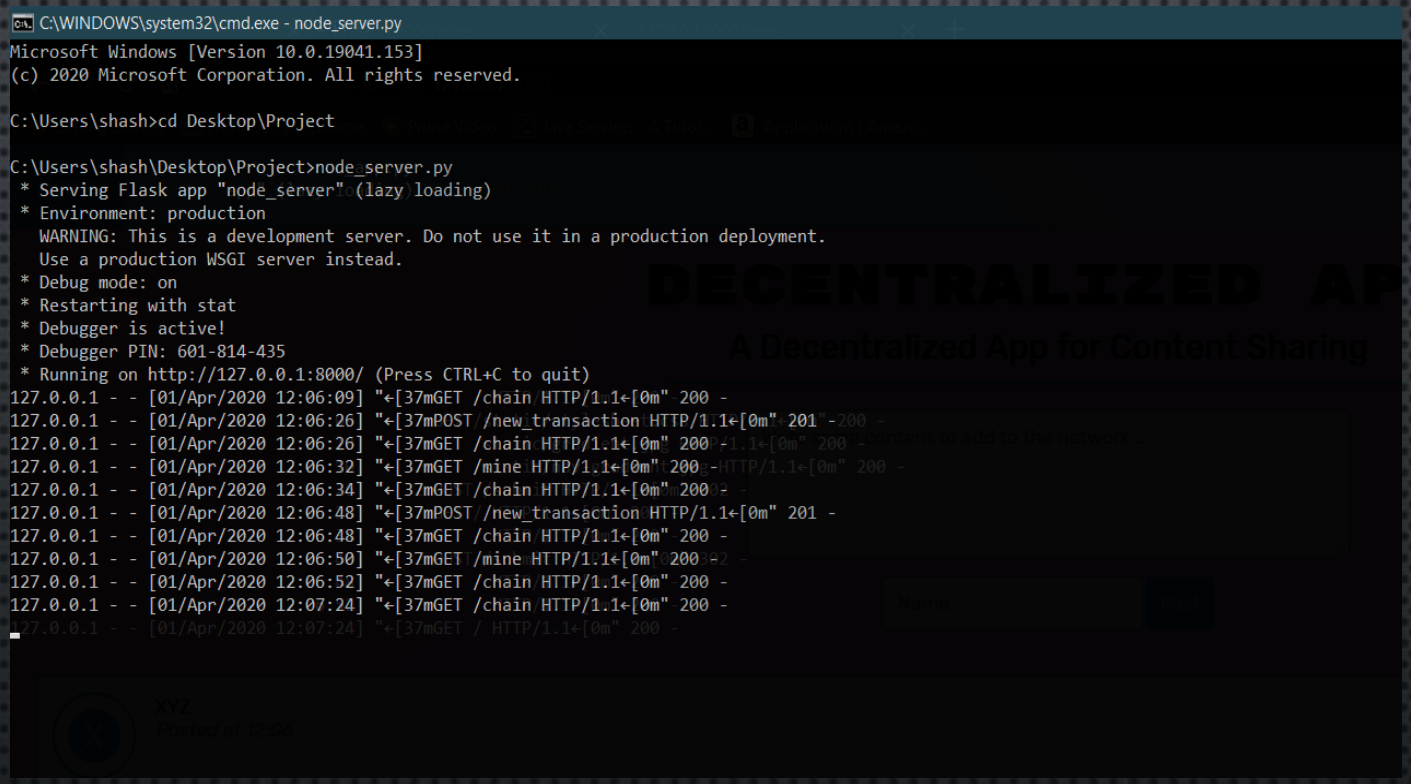
AFTER WE HAVE RESYNCHRONIZED THE BLOCKCHAIN OF THE SECOND USER AS WELL THEN THE FINAL OUTPUT WOULD BE LIKE SHOWN AND THE NUMBER OF MESSAGES IS DIRECTLY PROPORTIONAL TO NUMBER OF USERS POSTING THE BLOCKCHAIN.

FINAL OUTPUT OF THE PROJECT



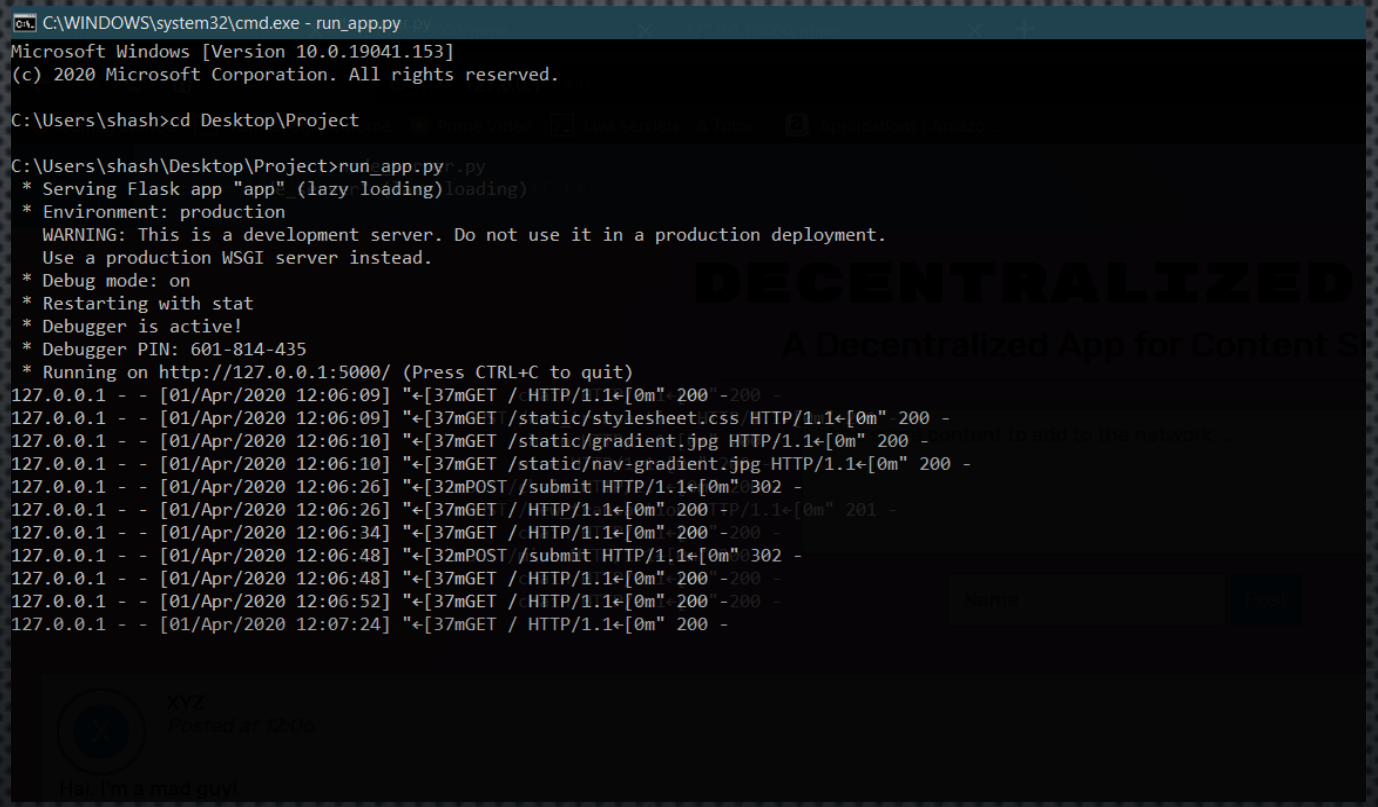
AFTER WE HAVE RESYNCHRONIZED THE BLOCKCHAIN OF THE ALL USERS AS WELL THEN THE FINAL OUTPUT WOULD BE LIKE SHOWN AND THE NUMBER OF MESSAGES IS DIRECTLY PROPORTIONAL TO NUMBER OF USERS POSTING THE BLOCKCHAIN

ACTIONS LOGGED BY THE LOCAL SERVER



ALL THE ACTIONS PERFORMED BY THE USER IN THE APPLICATION ARE REFLECTED IN THE BACKEND SERVER AND CAN BE SEEN WHAT HAS BEEN DONE BUT NOT WITH WHAT IT HAS BEEN DONE WITH.

ACTIONS LOGGED BY THE APPLICATION



WITH THIS WE CAN KEEP A TRACK OF PAGE DIRECTIONS OF THE APPLICATION RUNNING ON THE LOCAL SERVER.

CONCLUSIONS

The concept of blockchain technology is used for trading the renewable energy system in an environment among the neighbors in the peer-to-peer network.

Blockchain technology with cryptographic embedded to support the security issues can become a possible solution for the future to create a secure trading renewable energy system in the environment among the neighbors.

The energy and commodity transaction life cycle, even for simple transactions, involves a multitude of processes within each company and across market participants.

Blockchain turns both currencies and commodities into a digital form without relying on middleman which allows one person to trade with another.

A secure and anonymous decentralized messaging application is capable of sending end-to-end encrypted messages while ensuring that the identity of the sender and receiver are anonymous even with the presence of an adversary controlling most of the network.

Future Scope

An unexplored issue was accounting for the possibility of the user being offline or an unexpected network failure. The messages intended for a specific user could expire during this interval which would mean that the user would not be able to retrieve those messages. One solution could be a decentralized mail server capable of resending the expired messages to the network with sufficient POW. However, since the mail server would not know whether the recipient had received the message it would have to continue doing so indefinitely while new messages would continue being added. It is unlikely that any mail server could be capable of managing the appropriate POW and in any case, would result in a DoS attack on the Whisper Network.

An alternative solution might be to send the expired message directly to the node upon reconnection however this would result in the exposure of their identity as a recipient. Even if this could be solved by routing the message through the Whisper network upon an anonymous legitimate request by the recipient, there is an issue with storage and incentivizing storage by the participants in the network.

THANK YOU!!