GTÜ CSE 495&496

BLOCKCHAIN BASED SECURE MESSAGING APPLICATION

π2Pcath

Muhammed Bedir ULUCAY

Project Consultant:

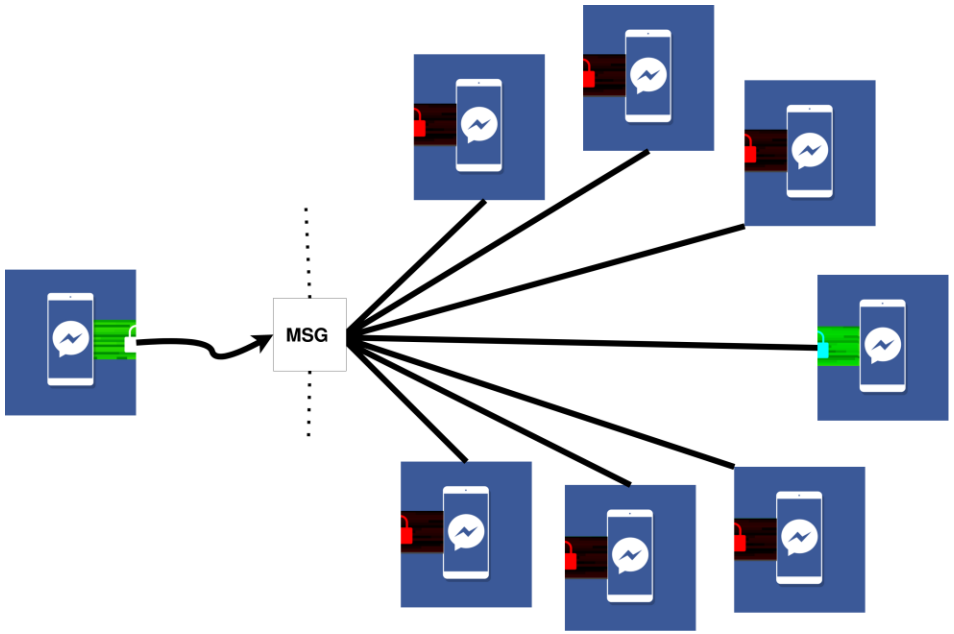
Prof. Dr. Ibrahim SOGUKPINAR

December 2022

CONTENTS

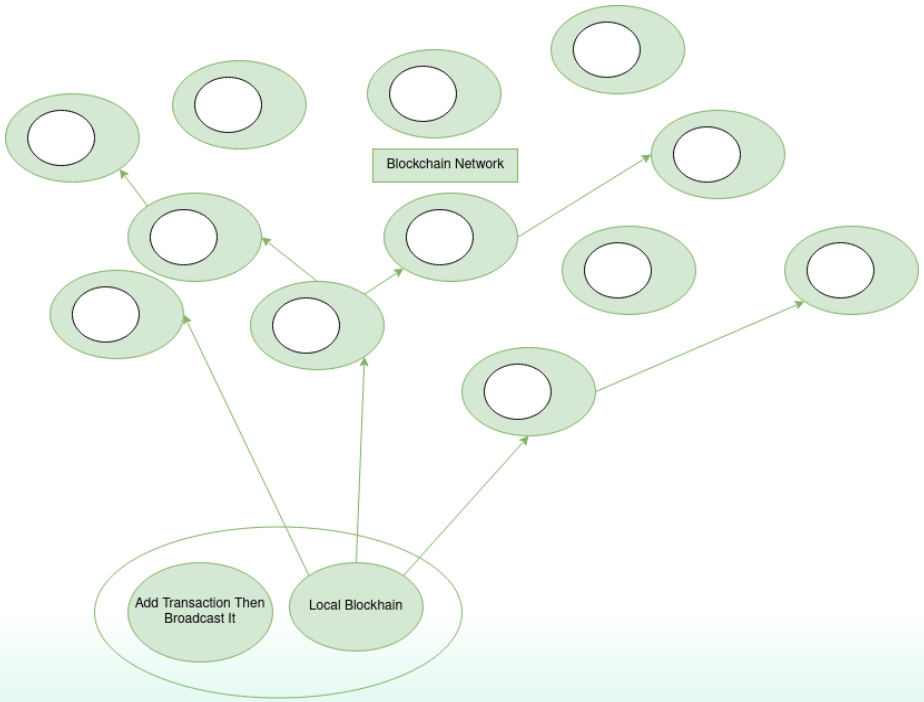
SCHEME AND DESCRIPTION OF THE PROJECT

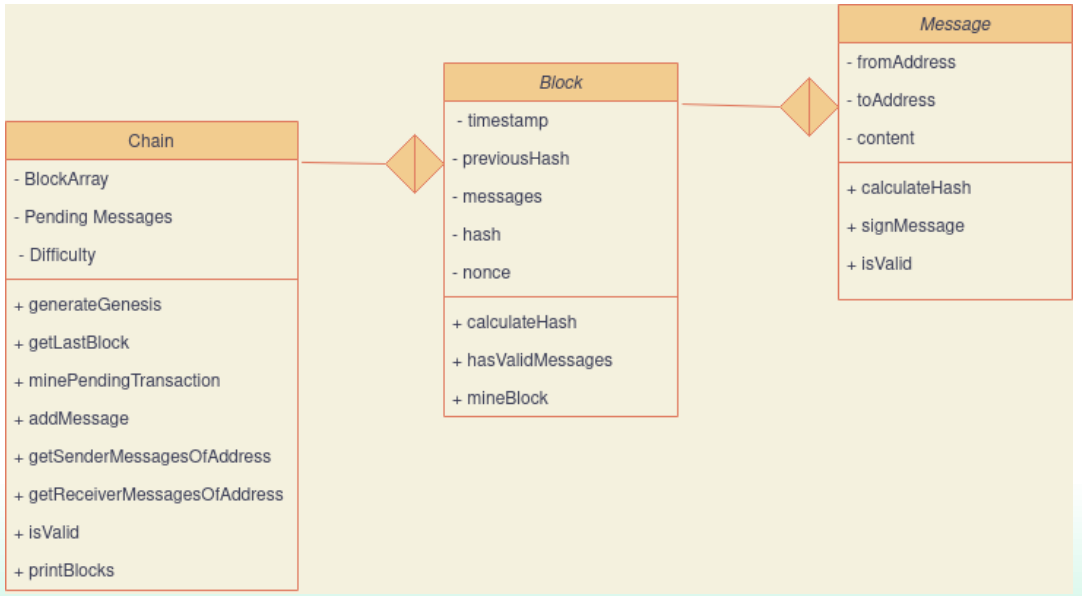
An application where people can securely communicate with each other as a peer to peer. To create a reliable and unchangeable communication channel between users using blockchain and cryptology. In this way, users can be absolutely sure that their data cannot be accessed by third parties or malicious people and from the recipient side.

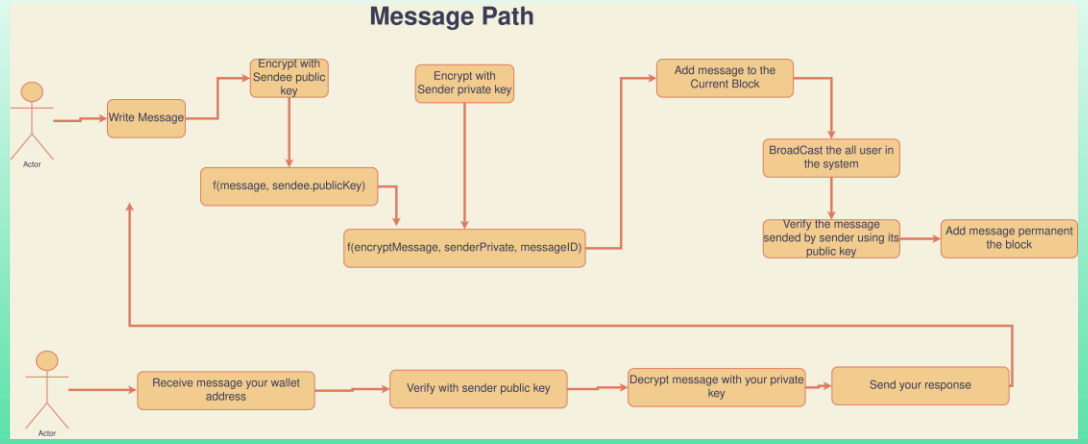


PROJECT DESIGN PLAN

People will communicate
via smart contracts and a blockchain network







Similar Project:

- Dust
- Status
- E-Chat
- BeeChat
- OpenChat
- BChat

PROJECT REQUIREMENTS - 1

- It must encode Messages asymmetrically according to the algorithm to be selected.
- It must create the necessary blocks for new messages
- I needs to research algorithms for unique number values to be used in the system.
- I needs to create a list of smart contracts required for the blockchain chain
- I should research the languages that I can use for the mobile application.
- I should decrease the size of the data so that it is appropriate for the users
- It needs to increase the message sending speed above a certain threshold.
- It must create the necessary unique data for each user locally so that others cannot access it.

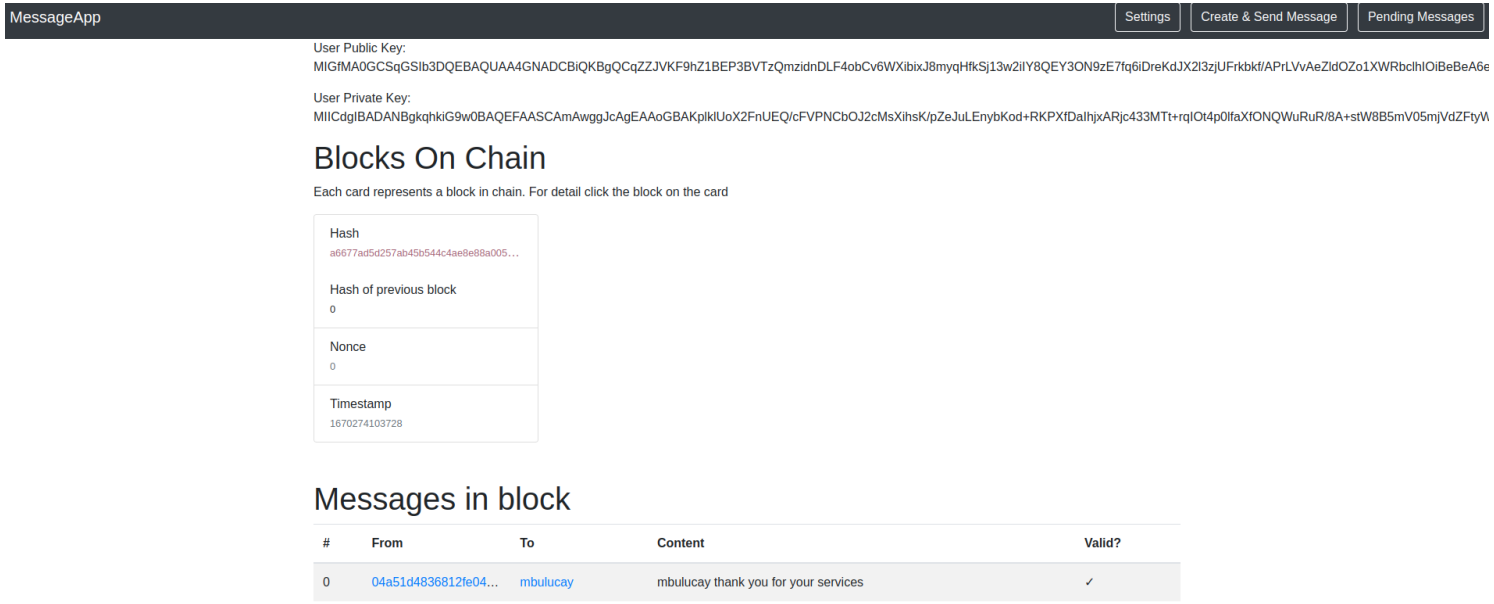
PROJECT REQUIREMENTS - 2

- I need mobile Cross platform library
- Go / Rust, JS, NodeJS
- Solidity for smart contracts of the blockchain structure
- 2 phone for testing
- Cryptographic hash algorithms
- Algorithms that generate 256-bit unique numbers
- Meta mask extension for connect the Ethereum blockchain network
- Ethereum mobile boilerplate
- Truffle Framework for local chain
- Libp2p, web3.js, Parity

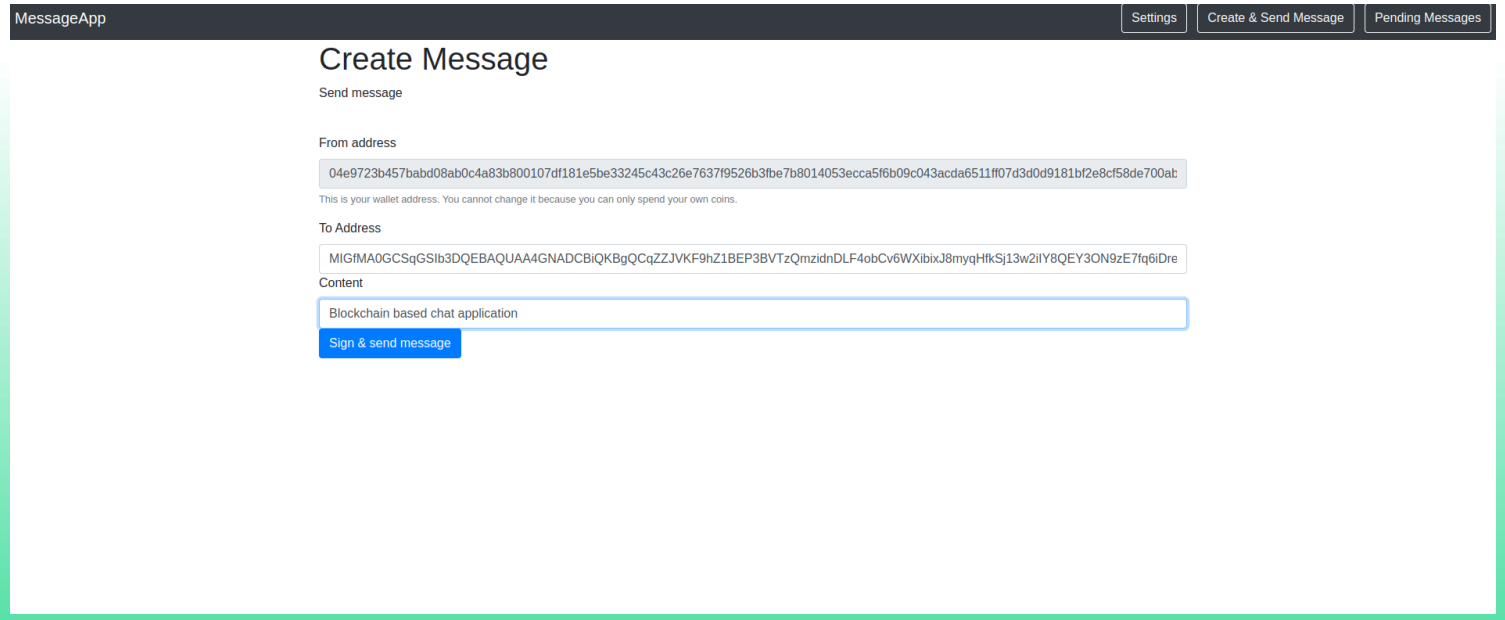
THE WORK DONE

- Build a local blockchain structure with message and block classes
- Build an interface for interaction with the blockchain
- Asymmetric encryption and decryption for a user
- Creating and sending message from a user to an address (public key)
- Mine block for pending messages and hashing with the new blocks
- Make sign a transaction to not change
- Read the article about designing a blockchain chat application.
- I learn the solidity, JS and angular for building application

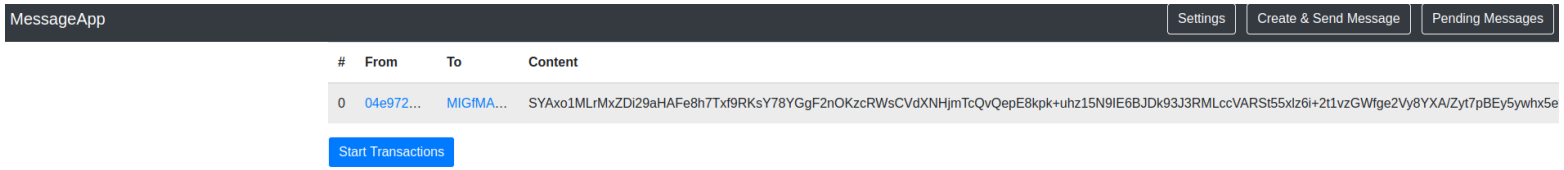
IMPLEMENTATION DESIGN AND TEST



Main page of the application:
We see genesis block and its message contents listed below the page

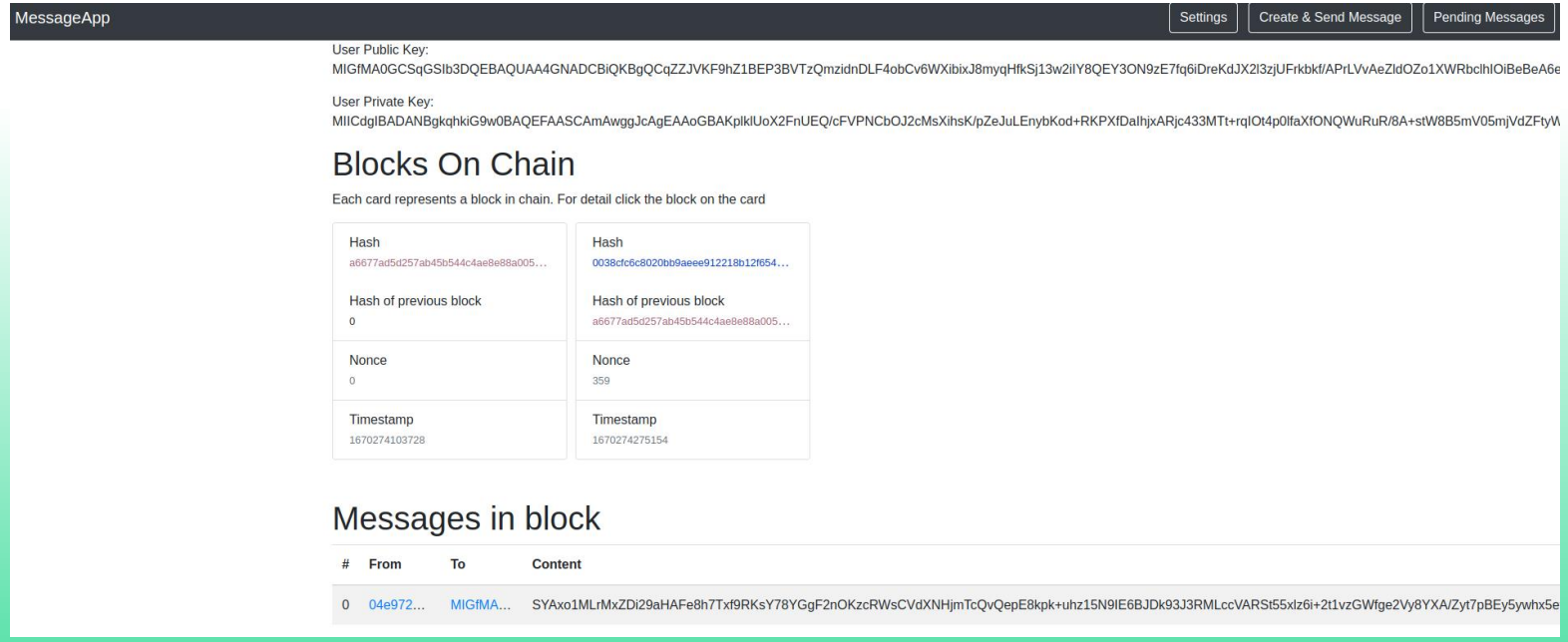


Message creation page:
Sending message and sender address and receiver address and filing the content of the messages.
The button sending transaction to pending transaction



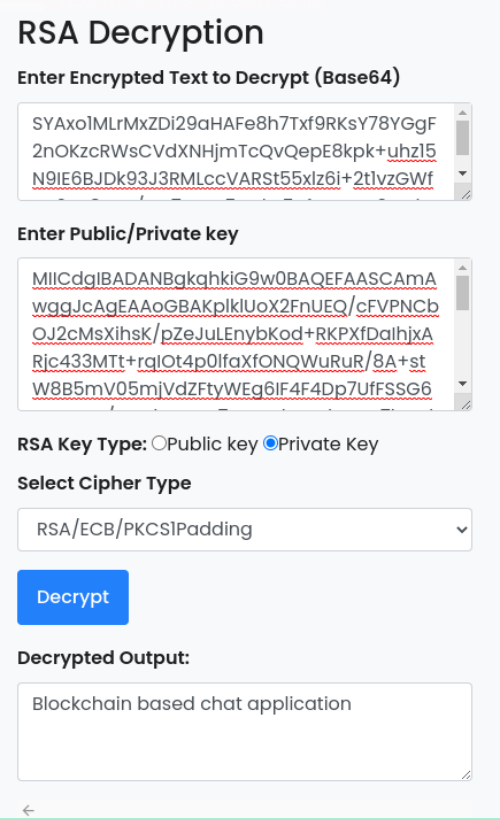
Mine the new block with the pending transaction never to be changed again.

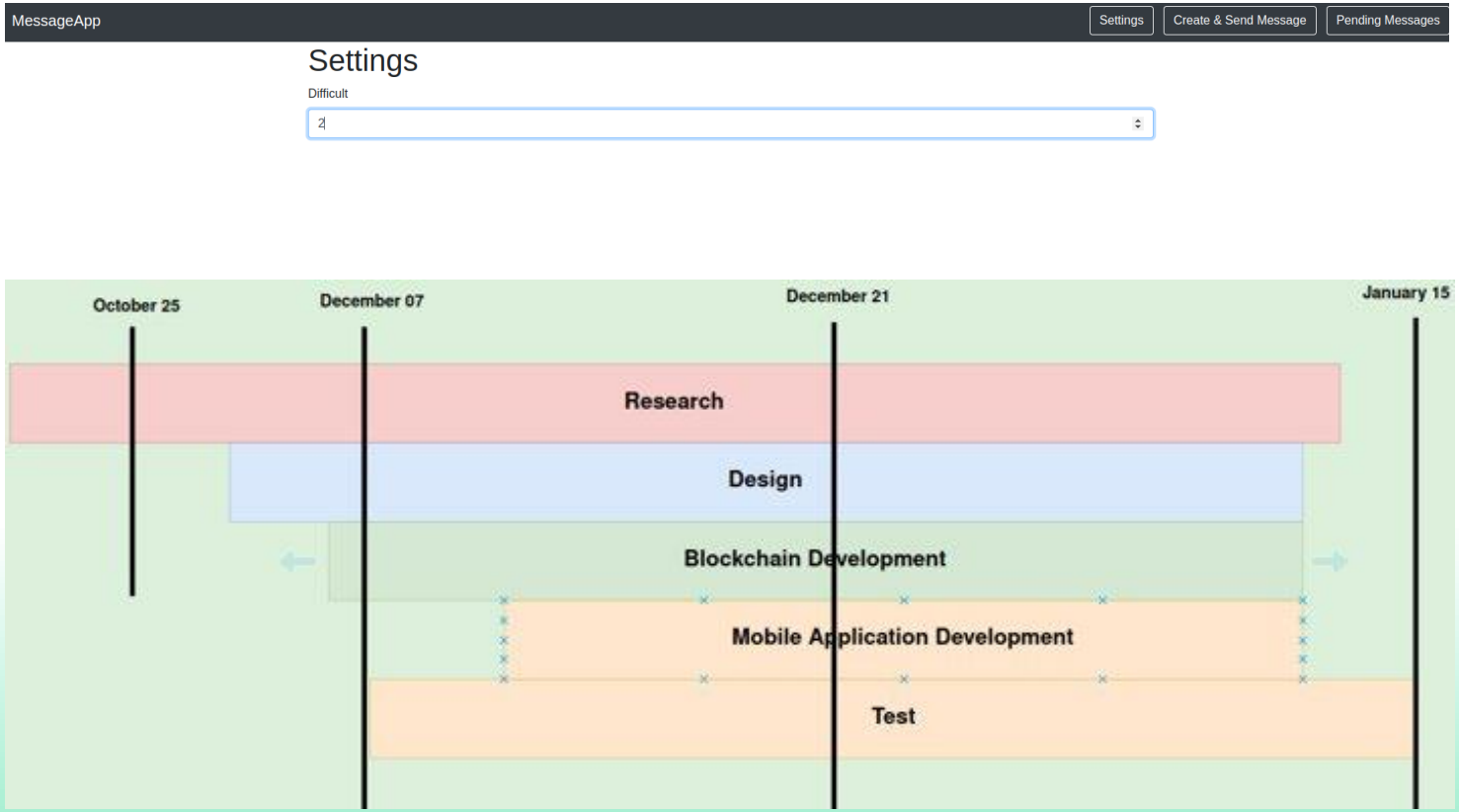
You can see the content of the message is encrypted. Just the owner can see the main text.



Mined block and added to the chain. The hashing is also working with previous hash.

Validate the encrypted message





For Code:
https://github.com/mbulucay/Blockchain/tree/main/OwnBlockchain/angular/MessageApp

GEBZE TEKNİK ÜNİVERSİTESİ

THINGS TO DO

- Creating a similar application using the smart contract.
- Build local blockchain network using truffle/ganache framework.
- Learn Ethereum blockchain network usage.
- Deploy own blockchain on Ethereum blockchain network.
- Build an interface for the application mobile/web.
- Create an Api services for broadcasting messages.
- Connect Meta mask extension with using tools(ganache/blockchain network)
- Learn Go or rust for the Api services
- Lots of research about blockchain

GEBZE TEKNİK ÜNİVERSİTESİ

SUCCESS CRITERIA

- Making message transaction below 4 second (human psychological)
- Usage and signing of the program after installation in 10 minute.
- Loading message transactions from blockchain under 2 minute.
- When a text search is requested, all messages containing the text written in less than 2 minute can be displayed to the user
- (depends on the size of blockchain)

REFERENCES

- https://www.researchgate.net/publication/318131748_An_Overview_of_Blockchain_Technology_Architecture_Consensus_and_Future_Trends
- https://www.irjet.net/archives/V7/i5/IRJET-V7I5531.pdf
- https://github.com/machinomy/awesome-non-financial-blockchain#readme
- https://scholarworks.calstate.edu/concern/theses/qj72pb04f?locale=en
- https://bitcoin.org/bitcoin.pdf
- https://github.com/TristanBilot/blockchain-chat-app
- https://www.youtube.com/watch?v=hYip_Vuv8J0&t=261s
- https://www.youtube.com/watch?v=bBC-nXj3Ng4&t=1065s
- https://www.youtube.com/watch?v=ZEApLtE8KkE&list=PLxz5IdaTYSOUmhECFNN-WfGeJrzlnXn_a
- https://www.blockchain.com/tr/explorer
- https://www.youtube.com/watch?v=yubzJw0uiE4&t=261s
- https://medium.com/adamant-im/how-decentralized-blockchain-messenger-works-b9932834a639
- https://medium.com/@BeFastTV/top-blockchain-messaging-apps-crypto-messengers-28893e5f908f
- https://hal.archives-ouvertes.fr/hal-02180329/document/
- https://docs.soliditylang.org/en/v0.8.17/introduction-to-smart-contracts.html
- https://remix.ethereum.org
- https://www.researchgate.net/publication/328160285_Survey_of_Consensus_Protocols
- https://blog.harmony.one/peer-discovery-in-harmony-network/
- https://rejolut.com/blog/creating-your-own-blockchain-network/