VIETNAM NATIONAL UNIVERSITY – HO CHI MINH CITY

THE INTERNATIONAL UNIVERSITY

SCHOOL OF COMPUTER SCIENCE AND ENGINEERING



**CHARITY FUNDRAISING APPLICATION BUILT ON ETHEREUM BLOCKCHAIN**

By

Ho Minh Huy

A thesis submitted to the School of Computer Science and Engineering

in partial fulfillment of the requirements for the degree of

Bachelor of Computer Science

Ho Chi Minh city, Vietnam

2018

**CHARITY FUNDRAISING APPLICATION BUILT**

**ON ETHEREUM BLOCKCHAIN**

APPROVED BY:

Tran Thanh Tung, Ph.D.

THESIS COMMITTEE

**ACKNOWLEDGMENTS**

I would like to express my very great appreciation to Dr. Tran Thanh Tung for his valuable and constructive suggestions and guidance during the planning and development of this research work. His constant encouragement and support helped me to achieve my goal.

Table of Contents

[LIST OF TABLES 5](#_Toc512869996)

**[LIST OF FIGURES](#_Toc512869997)** [6](#_Toc512869997)

[ABSTRACT 7](#_Toc512869998)

**iv**

# LIST OF TABLES

# **LIST OF FIGURES**

# ABSTRACT

Blockchain is a recent technology which provides a alternative type of database, structured as a chain of blocks of data , which is managed autonomously using a peer-to-peer network and distributed timestamping server. By design, a blockchain is unchanged. At the time researching this study, there are 3 major blockchain protocols such as Bitcoin, Ethereum and Ripple Consensus Network.

Current approaches of making payment through applications, especially on websites, is that payers have to trust and provide their private information to the middleman which is a bank or a payment service to valid transactions from their accounts and transfer their money to the final destinations. It is too risky. Not only in making payment but also in funding, the fundraiser owner does not directly receive funds from donators. Both sender and receiver are able to keep track of their transactions. But, it is not the way they expected.

This research approach is to apply Blockchain technology to create a decentralized application running on Ethereum private blockchain. This application has its own custom Token exchange and fund between owners and fundraisers. Besides, it is also required HTML including CSS, Javascript, Jquery, Bootstrap and Web3js for the front-end. The local server is run using Express module in Nodejs and a distributed database, which is MongoDB, to store fundraisers contents.

# CHAPTER I

## INTRODUCTION

Blockchain technology is amazingly trendy nowadays. This technique was authentically described in 1991 by a group of researcher and its purpose was to “timestamp” digital documents in order to prevent them from being tampered or put on a date that was earlier than which they were written. But until 2009, Satoshi Nakamoto adapted this technology to run a the digital cryptocurrency called Bitcoin, Blockchain has become widespread.

As its name, a blockchain is a database structure that can be described as a chain of blocks. Each block contains some data, the hash of that block and the hash of the previous block. The data that is stored inside a block depends on the type of the blockchain. For Bitcoin blockchain, the details about a transaction such as the sender, the receiver and amount of coins are packed and put into a block. The hash of a block is an alphanumeric string, which can be comparable to a fingerprint. It identifies a block and all of its contents. Just as a fingerprint, a hash is always unique. Once a block is created, its hash is calculated. If there is any change inside the block, it will be easily detected because the hash of that block is also changed. Therefore, it no longer is the same block. The third element inside each block is the hash of the previous block. This technique dramatically creates a chain of blocks and makes blockchain is so secure. In a blockchain, if any middle block is tampered, the hash of that block will be recalculated. In turn, that will make its adjacent block and all the following blocks invalid since they no longer store a valid hash of the previous one. Hence, changing a single block will cause all the hashes of other blocks to be recalculated in order to make the blockchain valid again.

In the area of technological breakthroughs, computers these days are so sophisticated that can calculate hundreds of thousands of hashes per second. The recalculation of all hashes of other blocks as a consequence of meddling with a middle block can be done easily. To due with this situation, blockchain has a mechanism named Proof-Of-Work. It helps to slow down the creation of new blocks by offering a prerequisite that must be fulfilled before a new block is added to the chain. For Bitcoin, it takes 10 minutes to calculate the required proof-of-work and add a new block to the chain. In case a block in a chain is tampered, it requires to recalculate not only all the hashes of other blocks but also the proof-of-work of each block.

There is one more way that most of blockchains secure themselves: being distributed. Instead of using a central entity to manage the chain, blockchain uses a peer-to-peer network. Anyone is allowed to join and became a node of the network. When someone runs as a full-node, he or she gets the full copy of the blockchain. The node can use this to verify that everything is still in order. When someone in the network successfully creates a new block and add it to the chain, it is called “mining”. That new block is sent to every nodes on the network. Each node then verifies the block to make sure that it has not been tampered with. All the node in this network create consensus. They make agreement about which blocks are valid and which are not. Blocks that are tampered with will be rejected by other nodes in the network. To successfully tamper with a blockchain, it requires to tamper with all blocks on the chain, redo the proof-of-work for each block and take control of more than 50% number of nodes on the network so that the tampered block could become accepted. And it is almost impossible to do so. In the end, after everything is checked out, each node update their copy by adding this block to the end of the chain. The one who mines the block is called miner. Mining would be explained in details later in Chapter II-Literature Review of this report. So, the security of blockchain comes from its creative use of hasing, proof-of-work mechanism and the consensus among all nodes in the peer-to-peer network.

Blockchain technology is constantly evolving. The born of Smart Contracts is one of the recent developments in this field. They are very popular nowadays. The term “smart contract” was first used by Nick Szabo in 1997, long before Bitcoin. He wanted to use a distributed ledger to store contracts. Now, smart contracts are similar to contracts in real world. The only difference is that they are completely digital because they are actually programs stored inside a blockchain. Bitcoin coin can be seen as a blockchain smart contract in some way. The purpose of creating smart contracts is to rebuild a community exchange system that does not require a third-party such as bank, middle organizations, etc. Programmers can write a smart contract which contains conditions and put them into a blockchain. For example, a sender A want to send an amount of money to receiver B, A and B have to trust a middleman ( a bank ) to handle their money correctly, ensure transactions are valid, non-fraudulent and successful. Neither sender nor receiver has to pay transaction fee to the bank. With a smart contract, transactions can be validated if all conditions are reached and the money will be sent directly to the receiver account address within a micro transaction fee. Otherwise, transactions will not be confirmed. And because the smart contract are stored on a blockchain, everything is completely distributed. This technique could remove the role of the middlemen or a third-party, which means no one is in control of the someone’s trading properties, and make a trust-less system become possible.

Are smart contracts trustful ? Because they are stored on a blockchain they inherit some sophisticated properties. Smart contracts are immutable and distributed. Once a smart contract is created, it can never be changed. Hence, no one can go behind the back and edit the code of a smart contract. The output of a smart contract in a blockchain is validated by everyone on the network, which means nobody could force one smart contract to release a wanted output because other nodes on the network will spot this attempt and mark it as invalid. This ability is related to smart contract distributed property. Tampering with smart contracts becomes almost impossible. Smart contracts can be applied to many different things: on crowdfunding, automatic payments, processing claims of insurances, delivery payments and so on. Present, there are a handful of blockchains which support smart contracts, but the biggest one is Ethereum.

The main goal of this research is to build a Charity Fundraising System called Decentralized Application, which is a combination of smart contracts, and deployed to Ethereum blockchain testnet - a simulation of real Ethereum blockchain - in order to make all the fundraisers and their funding transactions in this system become immutable, indelible and transparent.

# CHAPTER II

## LITERATURE REVIEW

**Ethereum:**

Mining, or generating, a new block is not a piece of cake. This process required miners to spend resources since it is very difficult. Imaging that there is a pool of records. A miner now pick out some records and then pack them into a block, which is called candidate block. THe

\_What is ethereum ?

\_what are its characteristics ?

\_What are in the ethereum Block ?

\_What does a transaction contain ?

\_The workflows of pushing a transaction to a block and valid that block.

\_What make it more powerful than Bitcoin

\_ SmartContract and ERC20 token

**Geth:**

\_What is geth

\_How it works, how it connects ( communicate ) to the ethereum blockchain

**Metamask:**

\_What is metamask ?

\_What is the different between metamask and geth ?

Solidity and Remix IDE:

\_Ethereum Virtual Machine

\_Remix IDE with representative features. and some Solidity basic syntax.

**HTML:**

\_basic knowledge of HTML

**Javascript and Web3js:**

\_Introduce Javascript

\_describe in details Web3js : Functions and their return results.

**Summary**

.

# CHAPTER III

## METHODOLOGY

**Overview Of Methodology :**

Describe the workflows along with the reality

**Detail Of Methodology:**

A private blockchain

Solidity code compile

SmartContract Deployment

ERC223 token and ERC223 SmartContrarct

Dapp front-end

# CHAPTER IV

## IMPLEMENTATION

**Input/Output data:**

**Implementation settings:**

\_Install Programming environment( Ethereum, Geth, required module,…)

**Implementation steps:**

\_Running a private node

\_Mining some ether

\_Deploy SmartContract ( Customized Token, Charity SmartContract, )

\_Running Dapp from its front-end and using metamask to interact with it.

# CHAPTER V

## RESULT / DISCUSSION

Showing result

# CHAPTER VI

## CONCLUSION

**1. Summary**

Advantages and disadvantages of using blockchain technology .

**2.Future work:**

Some promising improvements.

# LIST OF REFERENCES

# APPENDICES