**NHẬN XÉT CỦA GIẢNG VIÊN**

**MỤC LỤC**

Mục lục hình 3

Danh mục từ viết tắt 4

Chương I: Giới thiệu 5

I. Tính cấp thiết của đề tài 5

II. Tính phổ biến của đề tài 6

Chương II: Kiến thức nền tảng 10

I. Tổng quan về Blockchain 10

I.1 Lịch sử Blockchain 10

I.2 Công nghệ Blockchain 11

I.3 Tính năng của Blockchain 11

I.4 Xây dựng hệ thống block 12

I.4.1 Mặt cơ sở dữ liệu 12

I.4.2 Mặt khối 13

I.4.3 Hàm băm 14

I.4.4 Các node 15

I.5 Transaction 17

I.6 Tính an toàn 18

I.7 Tính thúc đẩy 18

II. Tổng quan về IPFS 18

II.1 Lịch sử IPFS 20

II.2 Cách thức làm việc của IPFS 20

II.3 Lợi ích 21

Chương III: Thiết kế hệ thống 23

I. Lưu đồ Use-case 23

II. Lưu đồ hoạt động 33

III. Lưu đồ tuần tự 37

Chương IV: Thực nghiệm 40

I. Cài đặt môi trường 40

II. Hoạt động của hệ thống 44

III. Test Case 45

IV. Kết quả thực nghiệm 47

Chương V: Kết luận 48

Tài liệu tham khảo 49

Phụ lục 50

**MỤC LỤC HÌNH**

Hình 1: Hình thức giao dịch Blockchain không cần bên thứ ba 1

Hình 2: Chuỗi các Block nối kết 11

Hình 3: Quy trình hoạt động của giao dịch chi trả 13

Hình 4: Các thuộc tính của block 14

Hình 5: Cơ chế của hàm băm 15

Hình 6: Các transaction mới được thêm vào vùng xử lý “Pending Facts” 16

Hình 7: Các node tìm kiếm block mới (Node 433 làm chủ đạo) 16

Hình 8: Cơ chế hash của block 598 với số nonce 1 11

Hình 9: Cơ chế hash của block 598 với số nonce 2 12

Hình 10: Cơ chế hash của block 598 với số nonce 11316 15

Hình 11: Các transaction mới được thêm vào vùng xử lý “Pending Facts” 16

Hình 12: Các node tìm kiếm block mới (Node 433 làm chủ đạo) 17

Hình 13: Phương thức chi trả bằng transaction 18

Hình 14: Hình thức làm việc giữa HTTP và IPFS 20

Hình 15: Use-case gửi - rút tiền tổng quan 23

Hình 16: Use-case tính năng quản lý tài khoản 26

Hình 17: Use-case chức năng gửi tiền 28

Hình 18: Use-case xem lịch sử giao dịch 30

Hình 19: Use-case tính năng rút tiền 31

Hình 20: Lưu đồ hoạt động quản lý tài khoản 33

Hình 21: Lưu đồ hoạt động chức năng gửi tiền 34

Hình 22: Lưu đồ hoạt động hiển thị lịch sử giao dịch 35

Hình 23: Lưu đồ hoạt động chức năng rút tiền 36

Hình 24: Lưu đồ tuần tự tổng quan 37

Hình 25: Lưu đồ tuần tự mô tả quản lý tài khoản 38

Hình 26: Lưu đồ tuần tự mô tả tính năng gửi tiền 38

Hình 27: Lưu đồ tuần tự mô tả tính năng xem lịch sử giao dịch 39

Hình 28: Lưu đồ tuần tự mô tả tính năng rút tiền 39

Hình 29: Giao diện cài đặt ứng dụng MetaMask trên Chrome 41

Hình 30: Giao diện thiết lập mật khẩu tài khoản MetaMask 42

Hình 31: Giao diện MetaMask sau khi thiết lập tài khoản mới 43

Hình 32: Giao diện Command Prompt sau khi cài đặt Chocolatey 43

Hình 33: Thư viện cho IPFS sau khi cài đặt trên Git Bash 44

**DANH MỤC TỪ VIẾT TẮT**

1. USD United States Dollars

2. RBC Royal Bank of CanadaBusiness

3. CNBC Consumer News and Business Channel

4. IBM International Business Machines

5. HSBC Hongkong and Shanghai Banking Corporation

6. KBC Kredietbank ABB Insurance CERA Bank

7. P2P Peer-To-Peer

8. IPFS InterPlanetary File System

9. DAG Directed Acyclic Graphs

10. PoW Proof-of-Work

11. ID Identification

12. RAM Random Access Memory

13. VGA Video Graphics Array

14. PHP Hypertext Preprocessor

15. XAMPP Cross-Platform - Apache - MariaDB - PHP - Perl

16. HTTP Hypertext Transfer Protocol

**CHƯƠNG I: GIỚI THIỆU**

**I. Tính cấp thiết của đề tài**

Bước sang thế kỉ 21, chưa bao giờ nền kinh tế thế giới phải đối diện với nhiều biến chuyển lớn như hiện nay, trong đó phải kể đến sự phát triển của công nghệ 4.0 mở đầu cho xu thế tương tác kĩ thuật số và trao đổi dữ liệu trong không gian máy tính, xúc tiến cho sự ra đời của công nghệ Blockchain. Trong những năm gần đây, nhu cầu bảo mật thông tin đang trở nên bức xúc hơn bao giờ hết, đòi hỏi quá trình trao đổi thông tin phải được đảm bảo bí mật, và công nghệ Blockchain là ứng cử viên sáng giá để giải quyết vấn đề đó. [1]

Nói đến việc trao đổi thông tin thì các ngân hàng trên toàn cầu đã và đang đẩy mạnh phát triển các hệ thống thanh toán kỹ thuật số để đảm bảo cho các giao dịch tài chính được an toàn. Nếu giải pháp sử dụng công nghệ Blockchain được áp dụng trong các hệ thống này thì các ngân hàng có thể tiết kiệm hàng tỷ USD chi phí và ngăn chặn được các hành vi gian lận, tiêu biểu như việc ngân hàng Hoàng gia Canada (RBC) đang thử nghiệm dùng Blockchain cho các khoản thanh toán giữa các chi nhánh ở Mỹ và Canada; ngoài ra CNBC cũng đưa tin hãng IBM đang xây dựng công nghệ Blockchain dành riêng cho 7 ngân hàng lớn nhất châu Âu (gồm Deutsche Bank, HSBC, KBC, Natixis, Rabobank, Societe Generale và Unicredit) để tăng hiệu quả giao thương quốc tế cho các doanh nghiệp nhỏ và vừa; ngân hàng Wells Fargo và Commonwealth Bank of Australia cũng từng dùng Blockchain để xử lý và thực hiện các chuyến xuất khẩu cotton từ Mỹ sang Trung Quốc,… [1]

Một trong những vấn đề quan trọng đặt ra cho việc phát triển chương trình xây dựng trên hệ thống Blockchain là đảm bảo thông tin bảo mật cho người dùng. Khi nhắc đến lợi ích của người dùng, ta có thể kể đến nhiều lĩnh vực như thương mại, giáo dục, y tế, bầu cử, chứng thực,… nhưng lĩnh vực ngân hàng là nơi tiêu biểu nhất thể hiện rõ lợi ích của người dùng được quan tâm như thế nào. Ngân hàng có nhiều hoạt động giao dịch từ việc gửi tiền tiết kiệm cho đến đầu tư quy mô lớn nhỏ vào các dự án quan trọng trong cuộc sống. Công nghệ Blockchain đang nổi lên như là cách thức để cho phép các tổ chức thực hiện và xác minh các giao dịch tài chính trên mạng ngay lập tức mà không cần xác thực tập trung. Theo cách thức truyền thống thì các giao dịch và thanh toán trong ngân hàng dựa vào xác thực tập trung hoặc thông qua trung gian để tạo ra cũng như cho phép hoạt động thanh toán. Blockchain cho phép một mạng lưới các máy tính phân tán có thể đạt được sự đồng thuận mà không cần thông qua bên trung gian. [1]

**II. Tính phổ biến của đề tài**

Hiện nay các chương trình Blockchain phục vụ cho các hoạt động ngân hàng lớn nhỏ đang trở nên ngày một nhiều, và sự đa dạng của chúng đi kèm với nhiều dạng cấu trúc khác nhau, cùng với phương thức hoạt động khác biệt.

Cấu trúc thiết lập chương trình phổ biến nhất thường xoay quanh việc áp dụng môi trường Truffle – chương trình kiểm thử đặc biệt dành cho Blockchain - hoặc Hyperledger – dự án mã nguồn mở được phát triển bởi Linux cho phép người dùng ứng dụng mọi loại hình Blockchain trong mọi ngành nghề, lĩnh vực khác nhau. Tuy nhiên, để cấu trúc xương sống của chương trình có thể chạy được thì ngoài môi trường ra thì các chương trình yêu cầu phải có sự trợ giúp từ ngôn ngữ lập trình như PHP, Java, C#; chương trình hỗ trợ xây dựng chương trình trực tiếp trên trình duyệt như Web3, NodeJS; và gói tin phụ trợ Webpack dành cho mọi phiên bản trên trình duyệt.

Phần phức tạp nhất của các chương trình ngân hàng áp dụng kĩ thuật Blockchain hiện nay chính là xây dựng ví điện tử phục vụ cho tiền điện tử của khách hàng. Để làm được điều đó, mỗi chương trình phải xây dựng thuật toán hash để mã hóa mã nhận dạng và điều chỉnh hệ thống tự động sao lưu thông tin của mỗi khách hàng vào các block để đưa lên Blockchain. Bên cạnh đó, các chương trình hiện đại đòi hỏi phải thiết lập thuật toán ***timing*** đối với các chương trình có quy mô lớn và ***rehash*** để lấy lại mật khẩu đối với các chương trình có quy mô nhỏ hơn. [1]

Cụ thể hơn thì đây là ví dụ điển hành về cách xây dựng chương trình:

▪ Giai đoạn I: Xây dựng smart contract đơn giản

⮚ Bước 1: Thiết lập môi trường TruffleSuite để thử nghiệm bằng việc nhập dòng lệnh sau trên Git Bash: *npm install truffle* hoặc có thể làm theo hướng dẫn mục Tutorial trên

⮚ Bước 2: Tạo folder đồ án và truy nhập vào bằng Git Bash.

⮚ Bước 3: Kế đến sử dụng chương trình Sublime Text và xây dựng contract mới trong thư mục contracts của đồ án.

⮚ Bước 4: Vào file truffle.js và nhập dòng lập trình truy cập vào mạng testnet Ropsten trên trang Infura cá nhân đã đăng ký.

⮚ Bước 5: Sử dụng Git Bash và chạy cài đặt testrpc bằng dòng lệnh:

*npm install -g ethereumjs-testrpc*

⮚ Bước 6: Mở cửa sổ Git Bash mới và chạy dòng lệnh hướng mạng Ropsten:

*truffle deploy --network=“ropsten”*

⮚ Bước 7: Nhập dòng lệnh chạy chương trình lần đầu: *npm run dev*

⮚ Bước 8: Truy cập đường dẫn localhost để xem chương trình.

▪ Giai đoạn II: Nối kết Frontend – Backend

⮚ Bước 1: Sử dụng chương trình Sublime Text và mở thư mục đồ án

⮚ Bước 2: Tạo các file sender, withdraw, user, records với mỗi tên dưới dạng 2 loại html và js, sau đó bổ sung thư viện vào các file có đuôi .js.

⮚ Bước 3: Thiết lập chức năng hiển thị vào file html và lệnh hoạt động cho file js.

⮚ Bước 4: Mở Command Prompt và cài đặt IPFS bằng dòng lệnh:

*choco install ipfs*

⮚ Bước 5: Mở Git Bash ở thư mục đề án và nhập dòng lệnh sau để xây dựng lại chương trình:

*truffle compile*

⮚ Bước 6: Ta cho chạy thử chương trình ở trang localhost mới:

*npm run dev*

▪ Giai đoạn III: Phát triển các chức năng nâng cao và kiểm thử tổng kết

⮚ Bước 1: Sử dụng chương trình Sublime Text và mở thư mục đồ án

⮚ Bước 2: Thiết lập dòng lệnh chức năng gộp transaction và hẹn giờ gửi lên block trên Blockchain.

⮚ Bước 3: Cho xây dựng lại chương trình bằng dòng lệnh:

*truffle compile*

⮚ Bước 4: Mở trang localhost mới và kiểm chứng thành quả:

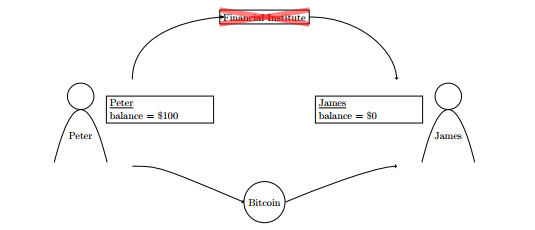
*npm run dev*

**CHƯƠNG II: KIẾN THỨC NỀN TẢNG**

**I. Tổng quan về Blockchain**

Blockchain có thể hiểu là một dạng cơ sở dữ liệu phân tán được dùng để duy trì một danh sách chứa các record liên tục tăng trưởng, chúng được biết đến là khối (Block) được gán con dấu thời gian và liên kết với khối trước đó. [2]

Blockchain còn đóng vai trò là một ledger phân tán mở, có nhiệm vụ ghi nhận và kiểm chứng các giao dịch giữa các bên một cách hiệu quả. [2]



**Hình 1: Hình thức giao dịch Blockchain không cần bên thứ ba**

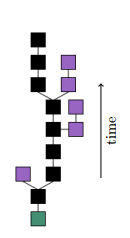
**I.1. Lịch sử Blockchain**

Blockchain và đồng Bitcoin được biết đến là thành quả sáng tạo từ một người Nhật tên là Satoshi Nakamoto.

Theo Nakamoto, Blockchain vốn được ông cho phát triển sớm vào những năm 90 thông qua việc học hỏi các thuật toán từ R.C.Merkle; và đến năm 2008 thì Nakamoto mới giới thiệu đến Bitcoin – một dạng tiền tệ kỹ thuật số được phát hành dưới dạng phần mềm mã nguồn mở, mở ra thời kỳ công nghệ tiền điện tử được xây dựng trên hệ thống kiểu ngang hàng (Peer-to-Peer). [2]

**I.2. Công nghệ Blockchain**

Blockchain được xây dựng trên giao thức đồng thuận (Consensus protocol) quy định cách thông tin được xây dựng cấu trúc và phương thức truyền nhận dữ liệu giữa các thiết bị trong hệ thống để đạt đến sự đồng thuận về trạng thái của chính hệ thống đó.

Ta cho trường hợp sau: Peter hứa sẽ giao 100$ cho cả James và Sarah thông qua Blockchain, sau khi đã kiểm định thì người đầu tiên request là James sẽ lãnh trọn 100$, buộc Peter phải thất hứa với người còn lại, chính là Sarah. [2]

**Thời gian**

**Hình 2: Chuỗi các Block nối kết**

Xét từ hình trên, Blockchain được kết thành một chuỗi vô tận từ dưới lên, trong đó block đầu tiên (Màu lục) là block duy nhất tiền nhiệm cho các block sau cùng.

Các block màu đen là các block thường, trong khi đó các block màu tím được hình thành khi có 2 block được tạo ra cùng thời điểm, với cùng kích cỡ. [2]

**I.3. Tính năng của Blockchain**

Blockchain ra đời nhằm giải quyết vấn đề chi tiêu gấp đôi (Double-spending) liên quan đến sự tin tưởng trong giao dịch giữa 2 bên mua và bán mà không cần thông qua bên thứ ba vốn chứa nhiều rủi ro, gian lận trong quá trình trao đổi. [2]

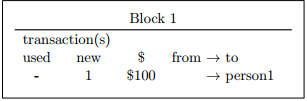
Nhờ tính chất của Blockchain, mọi giao dịch giữa 2 bên chỉ diễn ra đúng 1 lần, không có sự đảo ngược hay can thiệp từ bất kì nguồn nào vào quá trình đó. [2]

**I.4. Xây dựng hệ thống Block**

I.4.1. Mặt cơ sở dữ liệu

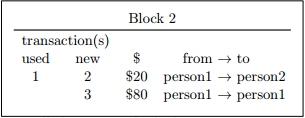
Khác với các database thông thường, database của Blockchain chỉ tồn tại khi đóng vai trò là một ledger chứa các thông tin giao dịch trước đó. [2]

Blockchain sẽ có nhiệm vụ tính toán lại “Số dư” mỗi khi transaction kết thúc, đồng nghĩa với việc nó sẽ theo dõi các giao dịch chi trả đã hết hạn hoặc không hoạt động (Active). [2]



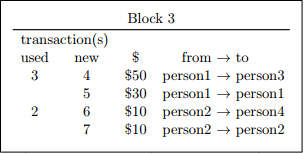
**Hình 3: Transaction không xảy ra giao dịch**

Tại transaction 1, block 1 có xảy ra hoạt động tạo mới thông tin người dùng nhưng không xảy ra hoạt động giao dịch.



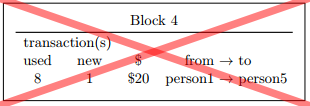
**Hình 4: Transaction với hoạt động chi trả giữa hai tài khoản**

Tại transaction 2, block 2 xuất hiện hoạt động chi trả của transaction 1 đến person2, transaction 3 là số tiền gửi về cho person1.



**Hình 5: Transaction với hoạt động chi trả giữa nhiều tài khoản**

Từ hình trên, các transaction 4 đến transaction 7 là hoạt động giao dịch giữa person1, person2, person3 và person4.



**Hình 6: Transaction bị chặn**

Do transaction 1 đã được dùng ở block 1 nên ở block 4 nó không chấp nhận dù person1 còn số dư để giao dịch với person5.

I.4.2. Mặt khối

Các thành phần của mỗi block gồm:

+ Số khối

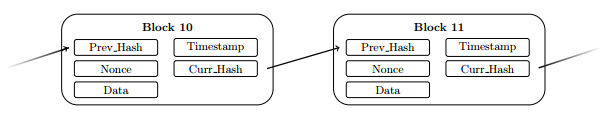
+ Dữ liệu các ***transaction***

+ Hàm băm (Hash) của block trước

+ Mốc thời gian tạo

+ Số ngẫu hứng (Nonce)

+ Hash của block hiện hành



**Hình 7: Các thuộc tính của block**

Trong đó, số Nonce được hiểu là một con số sinh ra để dùng cho cơ chế đồng thuận, được dùng như cơ chế đếm (Counter) để bảo đảm mỗi transaction chỉ được xử lý 1 lần duy nhất. Với hệ thống kiểu tập trung thì việc cập nhật trạng thái là bình thường, nhưng với hệ thống phân tán là cả 1 vấn đề, do vậy cần một cơ chế đồng thuận để mọi người có thể thống nhất tạo ra được 1 order của các transaction theo đúng ý muốn. [2]

Ngoài ra, các block làm việc dựa trên nguyên lý vận hành của Proof-of-work - là thuật toán đồng thuận được dùng để xác nhận giao dịch và tạo ra các block mới nên các miner thường dùng thuật toán này để cạnh tranh nhau giải toán các giao dịch trên Internet nhằm lãnh thưởng. [2]

I.4.3. Hàm băm

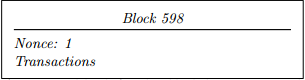
Hash là một thuật toán khó, đòi hỏi các miner phải đáp ứng các bước sau:

+ Bước 1: Một con số ngẫu nhiên phải được đoán (Nonce).

+ Bước 2: Nonce sẽ được thêm vào phần cuối tất cả dữ liệu của block.

+ Bước 3: Mọi dữ liệu sẽ được áp dụng hash bằng phương thức SHA-256.

+ Bước 4: Nếu hash bắt đầu bằng 1 con số xác định trước đó thì sẽ tìm ra block, ngược lại phải quay về bước 1.

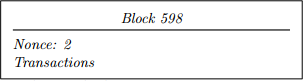


**Hình 8: Cơ chế hash của block 598 với số nonce 1**

Để dãy hash thích hợp cho bất kì block phải có 4 số 0 đầu, và block 598 không phải ngoại lệ, ở đây ta có dãy:

*"****bac6****d67daf63c7a06bab569adeadab130d332ed4c870da314d87f6f1b4c8a409"*

🡺 Dãy có 4 số đầu không phù hợp.

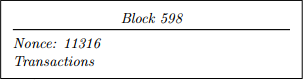
****

**Hình 9: Cơ chế hash của block 598 với số nonce 2**

Hash ở đây cho ra là:

*"****6ee6****b0c4aa8e6aaa369dfaee1379a59cec8797e7fc8ad1a358d57e1a87a1466d"*

🡺 Vẫn không phù hợp.



**Hình 10: Cơ chế hash của block 598 với số nonce 11316**

Kết quả ở block 598 với số nonce mới nhất đây là:

*"****0000****15783b764259d382017d91a36d206d0600e2cbb3567748f46a33fe9297cf“*

🡺 Phù hợp

I.4.4. Các node

Ta hiểu một miner (Hay 1 node) là một CPU đang thử giải 1 thuật toán khó để tìm 1 block mới, với các bước làm như sau:

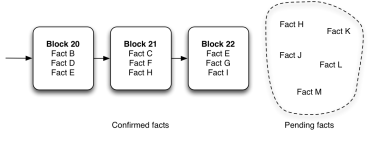
+ Bước 1: Các transaction mới sẽ được phát tới tất cả các node.

+ Bước 2: Mỗi node thu thập các transaction thành mỗi block.

+ Bước 3: Mỗi node hoạt động dựa trên việc tìm kiếm một proof-of-work cho block của nó.

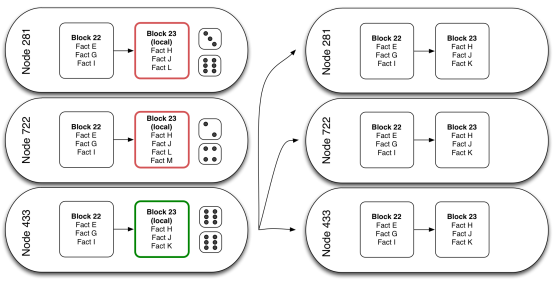
+ Bước 4: Mỗi node làm việc dựa trên một proof-of-work có chức năng phát block đến toàn bộ node.

+ Bước 5: Các node chấp nhận block chỉ khi các transaction trong đó hợp lệ và chưa được chi tiêu.

+ Bước 6: Các node biểu hiện sự chấp nhận đối với block bằng việc tạo block tiếp theo trong chuỗi, sử dụng hash của block đã được công nhận trước đó.

**Hình 11: Các transaction mới được thêm vào vùng xử lý “Pending Facts”**

Vùng “Pending Facts” ở đây đóng vai trò là nơi trông giữ cho các transaction chưa tìm được block thích hợp, trong khi đó các fact đã tìm được block ổn định sẽ được đưa lên node trên Blockchain và mã hóa bằng hash.

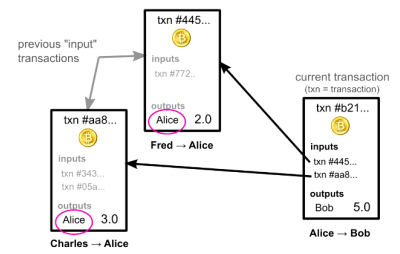
****

**Hình 12: Các node tìm kiếm block mới (Node 433 làm chủ đạo)**

Các node thường sẽ lấy một node làm chủ đạo sao cho block có chứa thành phần tương tự như node đầu, dẫn đến việc một số fact bị đào thải ra vùng Pending Facts.

**I.5. Transaction**

Transaction trong Blockchain không giới hạn mọi giá trị, tuy nhiên một số ứng dụng Blockchain sẽ tiến hành mã hóa chỉ khi các điều kiện đáp ứng yêu cầu đã chỉ định. Đối với tiền giao dịch điện tử, Blockchain thay vì theo dõi số dư và lưu trữ chúng thì nó sẽ tính toán lượng chi tiêu của các transaction đã thực hiện. [2]

****

**Hình 13: Phương thức chi trả bằng transaction**

Bằng phương thức này, thông tin khoản giao dịch của người dùng được mã hóa thành chuỗi mã mới, góp phần cho độ bảo mật của phương thức Blockchain trở nên tin cậy hơn.

**I.6. Tính an toàn**

Những lý do khiến Blockchain đáng tin cậy trong bảo mật như [2]:

+ Tính minh bạch: Bất kỳ ai có máy tính kết nối Internet đều có thể kiểm tra mọi thứ trong chuỗi block có hợp lệ hay không.

+ PoW: Nếu thay đổi block tiền nhiệm thì buộc người dùng phải liên tục thay đổi các block sắp tới.

+ Công nghệ mã hóa và giải thuật hash.

**I.7. Tính thúc đẩy**

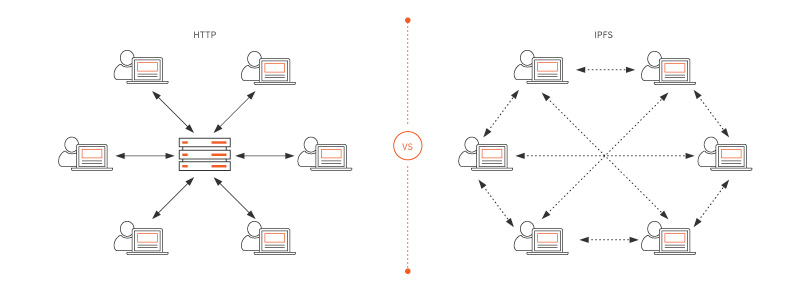
Tính năng này hoạt động dựa trên việc hao tổn năng lượng bằng thuật toán hash và để giúp duy trì sự an toàn cho Blockchain, nó cần nguồn nhân lực.

Tính năng cho phép trả tiền đối với các tài khoản có công trong việc khai phá các block, đặc biệt hơn là các tài khoản đó sẽ kiếm được nguồn thu nhập cao, từ đó lôi kéo nhiều người vào các mỏ trại Blockchain nhiều hơn. [2]

**II. Tổng quan về IPFS**

Blockchain trở thành một cơn sốt đối với các nhà đầu tư doanh nghiệp cũng như các ngân hàng lớn nhất nhì toàn cầu, tuy nhiên, xét theo góc độ công nghệ, Blockchain gặp không ít thách thức đến từ chính bản thân nó, bằng chứng hiện tại là các cơ chế đồng thuận chuỗi công việc đã làm giảm tốc độ giao dịch xuống gần mức tê liệt. Điều này làm cho việc lưu trữ dữ liệu hoặc các tệp lớn trên blockchain không bắt đầu. Nếu Blockchain không thể duy trì các chuỗi văn bản nhỏ chỉ đơn giản là ghi lại chuyển khoản cân bằng giữa hai bên, thì việc lưu trữ các tệp hoặc hình ảnh lớn trên Blockchain bị giới hạn và việc chuyển giao thông tin bị trì trệ liên tiếp. Để tiếp tục giải quyết vấn đề mới này, IPFS đã ra đời và tiếp nhận vai trò quản lý lưu giữ các thông tin đa dạng về kích thước và thể loại.

IPFS là một giao thức phân phối mã nguồn mở của giao thức hypermedia hoạt động dựa trên nội dung và danh tính. Cụ thể hơn, nó sẽ phân phối dữ liệu theo hình thức P2P, hay còn gọi là mạng ngang hàng (mạng đồng đẳng). Trong đó, các hoạt động của IPFS chủ yếu dựa vào khả năng tính toán băng thông của tất cả các máy tham gia chứ không tập trung vào một phần nhỏ các máy chủ trung tâm như các giao thức khác. Để giải thích rõ hơn cho điều này, nhóm nhà phát triển đã mô tả cách thức hoạt động của IPFS sẽ tương tự như BitTorrent, đồng nghĩa với mỗi máy tính tham gia trong mạng lưới của nó sẽ đảm nhận cả việc download lẫn upload dữ liệu mà không cần có sự có mặt của một máy chủ trung tâm. [3]



**Hình 14: Hình thức làm việc giữa HTTP và IPFS**

So với HTTP, IPFS mang lại tiềm năng lớn nhờ cải thiện được tốc độ truyền tải, tránh sự phụ thuộc vào các máy chủ và có khả năng cải thiện cấu trúc của nền tảng Internet. Những thuật ngữ chuyên ngành và cách thức hoạt động rắc rối có thể sẽ phù hợp hơn với dân IT, nhưng lại là những yếu tố quá xa vời đối với người dùng Internet. Để biết rõ tiềm năng thật sự của IPFS, chúng ta cần biết thêm IPFS sẽ mang tới điều gì cho người dùng, cách nó cải tiến hơn so với HTTP và lí do nên chuyển qua sử dụng IPFS. [3]

**II.1. Lịch sử IPFS**

Trong năm 2014, giao thức IPFS đã tận dụng Bitcoin và hạ tầng mạng để lưu trữ dữ liệu không thay đổi được, loại bỏ các tập tin trùng lặp trên mạng và lấy thông tin địa chỉ để truy cập các node lưu trữ để tìm kiếm các tập tin trong mạng.

IPFS được triển khai trong chương trình Go và JavaScript vốn đã tồn tại, và hiện tại nó đang được triển khai trong Python. Việc triển khai lên Go được coi là cách thực hiện tham chiếu khi các đặc tả chính thức đang dần được quan tâm phát triển. [4]

**II.2. Cách thức làm việc của IPFS**

Về cơ bản, IPFS là một khái niệm tương tự với World Wide Web mà chúng ta biết ngày nay, nhưng tương tự như hình thức một BitTorrent đơn lẻ trao đổi các đối tượng trong một kho lưu trữ Git duy nhất. Các tệp được phân phối thông qua giao thức dựa trên BitTorrent. Quan trọng hơn, IPFS hoạt động như một sự kết hợp giữa Kodemila, BitTorrent và Git để tạo ra một hệ thống con phân tán của Internet.

Thiết kế của giao thức cung cấp phiên bản trước đây của Internet tương đồng với Git. Mỗi tệp và tất cả các khối trong đó được cung cấp một mã định danh duy nhất - chính là một mã hash. Các bản sao bị xóa trên mạng và phiên bản cũ được theo dõi ở mọi tệp. Điều này dẫn đến nội dung có sẵn liên tục, nơi các trang web không biến mất do máy chủ bị hỏng hoặc máy chủ lưu trữ web bị gián đoạn. Hơn nữa, tính xác thực của nội dung được đảm bảo thông qua cơ chế này và khi tra cứu các tệp, về cơ bản bạn yêu cầu mạng tìm các nút lưu trữ nội dung đằng sau hàm hash duy nhất được liên kết với nội dung đó.

Các liên kết giữa các nút trong IPFS có dạng băm mật mã, và điều này là có thể vì kiến trúc dữ liệu Merkle DAG của nó. Những lợi ích của Merkle DAG cho IPFS bao gồm:

+ Nội dung địa chỉ - Nội dung có một số nhận dạng duy nhất là băm mật mã của tệp.

+ Không sao chép - Không thể sao chép các tệp có cùng nội dung và chỉ được lưu trữ một lần.

+ Bằng chứng giả mạo - Dữ liệu được xác minh bằng tổng kiểm tra của nó, vì vậy nếu thay đổi băm, thì IPFS sẽ biết dữ liệu bị giả mạo.

Mỗi node chỉ lưu trữ nội dung mà nó quan tâm và lập chỉ mục thông tin cho phép nó tìm ra ai đang lưu trữ cái gì. Khuôn khổ cho IPFS về cơ bản loại bỏ sự cần thiết cho các máy chủ tập trung để cung cấp nội dung trang web cho người dùng. Cuối cùng, khái niệm này hoàn toàn có thể đẩy giao thức HTTP thành không liên quan và cho phép người dùng truy cập nội dung cục bộ, ngoại tuyến. Thay vì tìm kiếm các máy chủ như với cơ sở hạ tầng hiện tại của Internet, người dùng sẽ tìm kiếm ID duy nhất (Băm mật mã), cho phép hàng triệu máy tính phân phối tệp cho bạn thay vì chỉ một máy chủ.

**II.3. Lợi ích**

Bằng việc cung cấp kiến ​​trúc P2P mới cùng việc phân tán cho Internet đi kèm với cấu trúc phức tạp của nó, nhiều lợi ích có thể kể đến trong mọi thứ từ tiết kiệm tài chính trong lưu trữ và dải băng thông để tích hợp với các mạng Blockchain phân tán. [3]

Lợi thế rõ ràng đi kèm với mô hình lưu trữ phân tán của IPFS áp dụng cho việc lưu trữ dữ liệu hiệu quả hơn và không thay đổi, lâu dài cùng với nó. Không còn các trang web sẽ được chuyển xuống các thông báo lỗi 404 theo chu kỳ do các máy chủ bị hỏng hoặc chuỗi bị gián đoạn của các liên kết HTTP bị gián đoạn. Hơn nữa, những lợi thế đáng kể có sẵn cho các nhà nghiên cứu về hiệu quả, đặc biệt là những người cần phân tích và phân tích các tập dữ liệu rất lớn. Với sự phổ biến của Big Data trong khoa học hiện đại, hiệu suất nhanh và lưu trữ phân tán dữ liệu do IPFS lo liệu sẽ trở nên thích hợp để thúc đẩy tiến bộ công tác giao dịch.

Các nhà cung cấp dịch vụ và người tạo nội dung cũng có thể giảm đáng kể chi phí hoạt động của họ liên quan đến việc phân phối lượng lớn dữ liệu cho khách hàng. Nói lặp đi lặp lại, hiện tại vấn đề của mô hình này là sự cản trở bởi việc tăng chi phí băng thông và các nhà cung cấp dữ liệu bị tính thêm phí cho các thỏa thuận ngang hàng. Các chi phí liên quan đến việc phân phối nội dung thông qua cơ sở hạ tầng đều tập trung của các mạng được kết nối với nhau có chỉ số tăng và qua đó tạo ra một môi trường mang tính hiệu quả, quan trọng và tập trung hơn nữa trong nỗ lực vượt qua những thách thức kể trên.

Ngoài ra, việc tập trung máy chủ như HTTP dẫn đến việc rình mò của chính phủ, tăng tỷ lệ tấn công DDoS, kiểm duyệt ISP và bán dữ liệu cá nhân; ở trường hợp của IPFS thì các đợt tấn công thường rất khó xảy ra tùy vào khả năng của nhà phát triển.

Như Juan Benet, tác giả của IPFS đã nói “Nội dung trên IPFS có thể di chuyển qua bất kỳ người trung gian không tin cậy nào mà không từ bỏ quyền kiểm soát dữ liệu hoặc gây nguy hiểm đến người dùng”. [3]

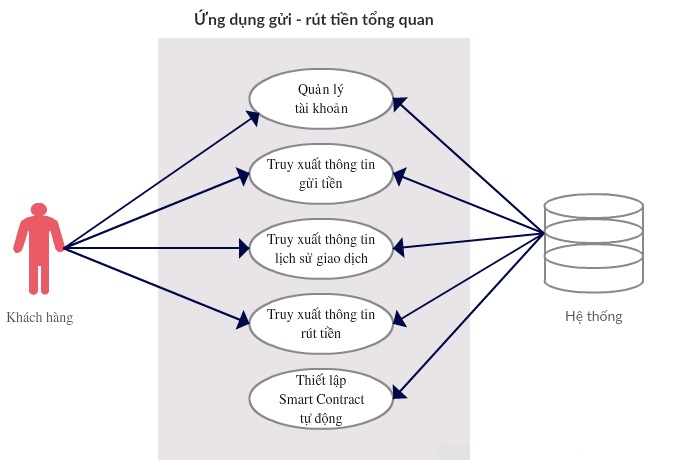
Cuối cùng, sự tích hợp của IPFS với công nghệ Blockchain dường như hoàn toàn phù hợp. Khi sử dụng IPFS trong một giao dịch blockchain, bạn có thể đặt các liên kết vĩnh viễn, không thay đổi. Dấu thời gian bảo mật dữ liệu của bạn mà không cần phải lưu trữ dữ liệu trên chuỗi, điều này dẫn đến việc giảm khối lượng blockchain và cung cấp một phương pháp thuận tiện cho các giải pháp chuỗi an toàn để giúp ngăn chặn quy mô.

IPFS đang được đưa vào một số nền tảng tiền điện tử và có khả năng giúp ngành công nghiệp mở rộng quy mô bằng cách cung cấp kiến trúc hệ thống tệp ngang hàng và phân tán cần thiết làm nền tảng để hỗ trợ sự phát triển của nền tảng tiền điện tử. [4]

**CHƯƠNG III: THIẾT KẾ HỆ THỐNG**

Nhằm đảm bảo định hướng của nhóm luôn rõ ràng và chính xác, chương trình do nhóm xây dựng được lên kế hoạch cơ cấu toàn bộ tính năng và hoạt động dưới dạng lưu đồ, với mỗi lưu đồ biểu thị hướng hoạt động và mục đích của ứng dụng được phát triển.

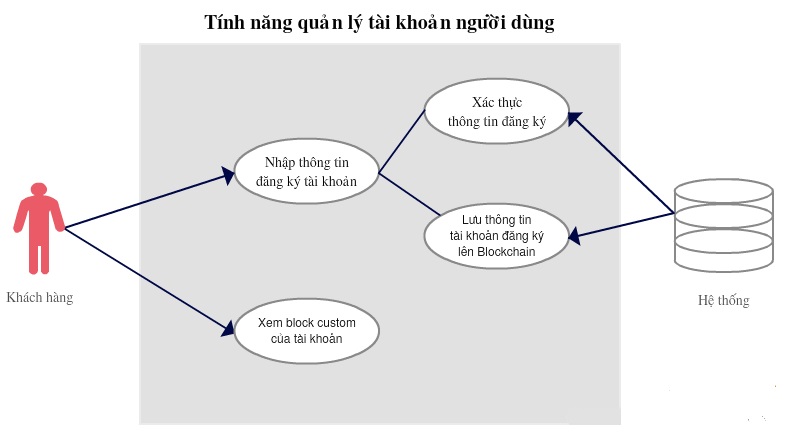
**I. Lưu đồ Use-case**

****

**Hình 15: Use-case gửi - rút tiền tổng quan**

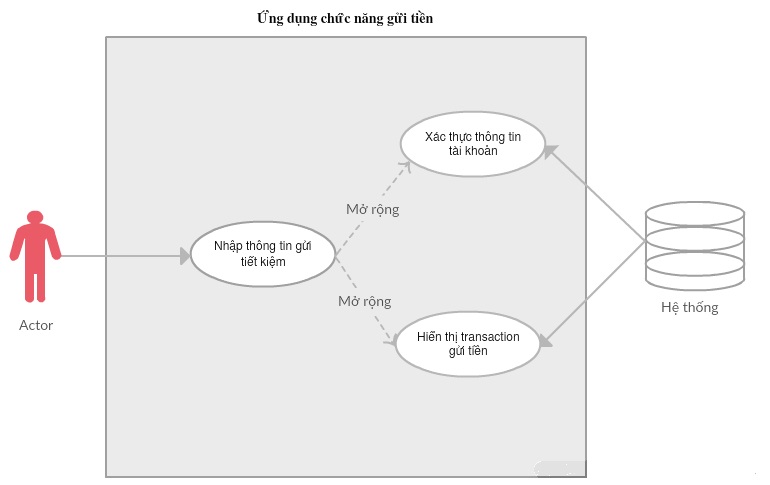
**Bảng 1: Bảng mô tả use-case gửi - rút tiền tổng quan**

|  |  |  |
| --- | --- | --- |
| **Tên lưu đồ** | Lưu đồ Use-case gửi – rút tiền tổng quan | |
| **Tác nhân** | 1. Khách hàng  2. Hệ thống | |
| **Mô tả chi tiết** | - Khách hàng truy cập các mục như: Manage Account, Sending, Records và Withdrawal để thiết lập hoạt động gửi tiền tiết kiệm. | |
| **Điều kiện trước** | - Khách hàng phải truy cập vào trang của ứng dụng. | |
| **Điều kiện sau** | - Khách hàng đăng ký được tài khoản trên ứng dụng ngân hàng.  - Khách hàng có thể xem giao dịch đã thực hiện hoàn tất.  - Khách hàng thực hiện công việc gửi – rút tiền trên ứng dụng ngân hàng. | |
| **Các luồng sự kiện** | **Hành động** | **Phản hồi của**  **hệ thống** |
| - Chọn mục Manage Account  - Nhập thông tin đăng ký tài khoản ngân hàng  - Nhấp nút gửi lên hệ thống  - Xác nhận phí transaction  - Chọn mục Sending  - Nhập số tiền cần gửi  - Nhập hạn gửi  - Xác nhận phí transaction  - Chọn mục Records  - Chọn mục Withdrawal  - Nhập số tiền cần rút  - Nhập hạn tiền đã gửi  - Xác nhận phí transaction | - Kiểm tra tài khoản trên hệ thống  - Hiển thị transaction đăng ký  - Lưu thông tin lên Blockchain  - Hiển thị block dữ liệu tài khoản  - Kiểm tra thông tin tài khoản IPFS và MetaMask  - Kiểm tra thông tin gửi tiền  - Hiển thị transaction gửi tiền  - Xác nhận tiền gửi  - Lưu thông tin giao dịch trên MetaMask  - Cập nhật thông tin tài khoản khách hàng  - Kiểm tra thông tin tài khoản IPFS và MetaMask  - Kiểm tra thông tin tài khoản IPFS và MetaMask  - Kiểm tra giao dịch gửi tiền  - Kiểm tra giao dịch rút tiền  - Hiển thị danh sách Requests đã thực hiện  - Kiểm tra thông tin tài khoản IPFS và MetaMask  - Kiểm tra thông tin rút tiền  - Hiển thị transaction rút tiền  - Xác nhận tiền rút  - Lưu thông tin giao dịch trên MetaMask  - Cập nhật thông tin tài khoản khách hàng |

****

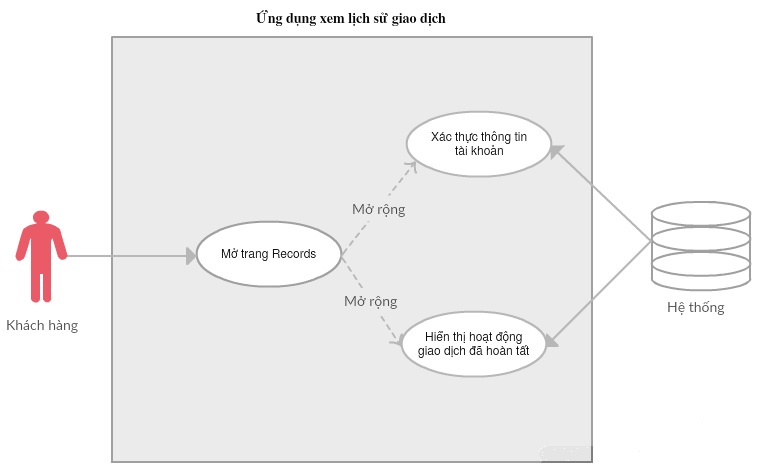
**Hình 16: Use-case tính năng quản lý tài khoản**

|  |  |  |
| --- | --- | --- |
| **Tên lưu đồ** | Lưu đồ Use-case quản lý tài khoản | |
| **Tác nhân** | 1. Khách hàng  2. Hệ thống | |
| **Mô tả chi tiết** | - Khách hàng sẽ tiến hành truy cập mục Quản lý tài khoản (Manage Account) | |
| **Điều kiện trước** | - Khách hàng phải vào trình duyệt và truy cập vào trang chủ ứng dụng.  - Khách hàng truy cập vào mục Quản lý tài khoản (Manage Account)  - Thông tin tài khoản khách hàng đã được xác minh | |
| **Điều kiện sau** | - Thông tin tài khoản đăng ký sẽ được lưu lại trên hệ thống.  - Một block chứa tên khách, mã hash và đoạn thông tin của khách hàng sẽ hiển thị trên trang. | |
| **Các luồng sự kiện** | **Hành động** | **Phản hồi của**  **hệ thống** |
| - Chọn mục Manage Account  - Nhập thông tin đăng ký tài khoản ngân hàng  - Nhấp nút gửi lên hệ thống  - Xác nhận phí transaction | - Kiểm tra tài khoản trên hệ thống  - Hiển thị transaction đăng ký  - Lưu thông tin lên Blockchain  - Hiển thị block dữ liệu tài khoản |

****

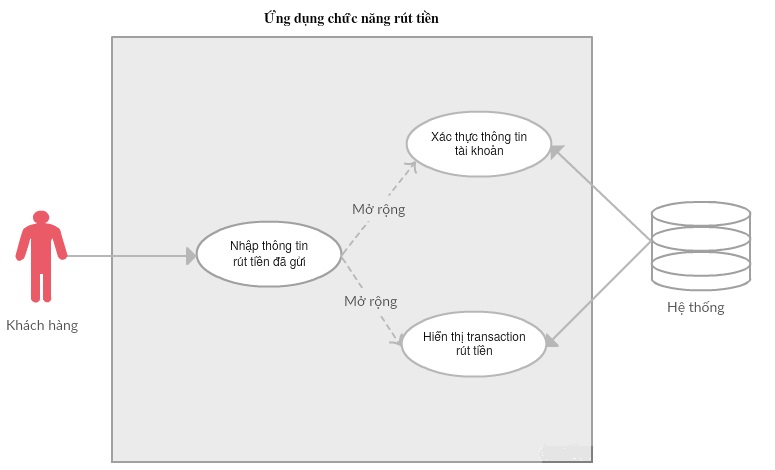
**Hình 17: Use-case chức năng gửi tiền**

|  |  |  |
| --- | --- | --- |
| **Tên lưu đồ** | Lưu đồ Use-case gửi tiền | |
| **Tác nhân** | 1. Khách hàng  2. Hệ thống | |
| **Mô tả chi tiết** | - Khách hàng truy cập mục gửi tiền (Sending).  - Khách hàng tiến hành nhập thông tin số tiền cần gửi và kì hạn gửi. | |
| **Điều kiện trước** | - Khách hàng đã đăng ký tài khoản trên ứng dụng.  - Khách hàng sở hữu tài khoản MetaMask.  - Khách hàng có tài sản lớn hơn 0. | |
| **Điều kiện sau** | - Khách hàng gửi được tiền vào hệ thống ngân hàng.  - Số tiền khách hàng giảm bằng số tiền khách hàng đã đưa.  - Hệ thống sẽ lưu hồ sơ hoạt động giao dịch gửi tiền | |
| **Các luồng sự kiện** | **Hành động** | **Phản hồi của**  **hệ thống** |
| - Chọn mục Sending  - Nhập số tiền cần gửi  - Nhập hạn gửi  - Xác nhận phí transaction | - Kiểm tra thông tin tài khoản IPFS và MetaMask.  - Kiểm tra thông tin gửi tiền  - Hiển thị transaction gửi tiền  - Xác nhận tiền gửi  - Lưu thông tin giao dịch trên MetaMask  - Cập nhật thông tin tài khoản khách hàng |

****

**Hình 18: Use-case xem lịch sử giao dịch**

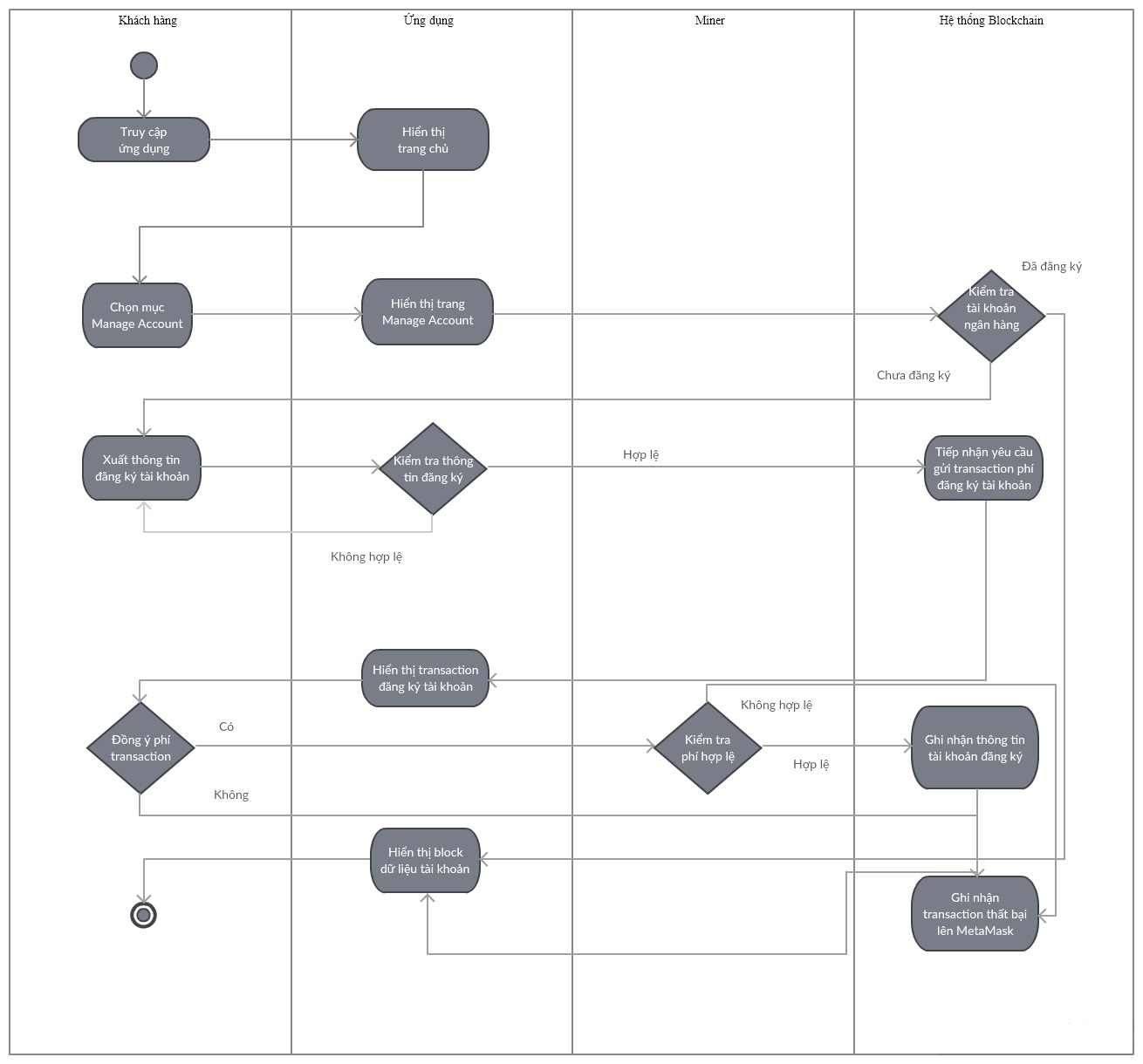
|  |  |  |
| --- | --- | --- |
| **Tên lưu đồ** | Lưu đồ Use-case lịch sử giao dịch | |
| **Tác nhân** | 1. Khách hàng  2. Hệ thống | |
| **Mô tả chi tiết** | - Khách hàng xem thông tin giao dịch mà khách hàng đã hoàn tất trước đây. | |
| **Điều kiện trước** | - Khách hàng đã đăng ký tài khoản trên ứng dụng.  - Khách hàng sở hữu tài khoản MetaMask. | |
| **Điều kiện sau** |  | |
| **Các luồng sự kiện** | **Hành động** | **Phản hồi của**  **hệ thống** |
| - Chọn mục Records | - Kiểm tra thông tin tài khoản IPFS và MetaMask.  - Kiểm tra giao dịch gửi tiền  - Kiểm tra giao dịch rút tiền  - Hiển thị danh sách Requests đã thực hiện |

****

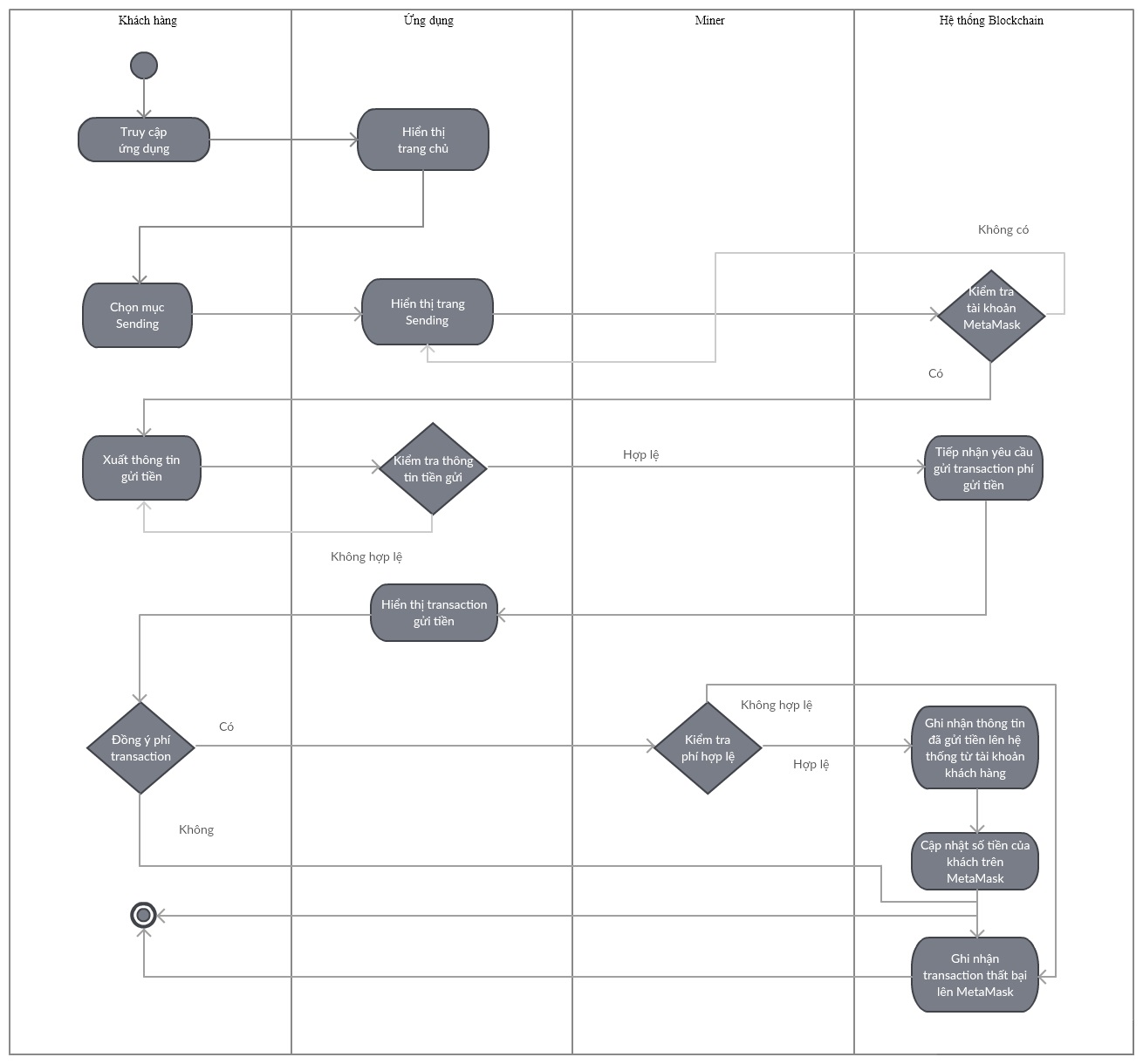
**Hình 19: Use-case tính năng rút tiền**

|  |  |  |
| --- | --- | --- |
| **Tên lưu đồ** | Lưu đồ Use-case gửi tiền | |
| **Tác nhân** | 1. Khách hàng  2. Hệ thống | |
| **Mô tả chi tiết** | - Khách hàng truy cập mục gửi tiền (Withdrawal).  - Khách hàng tiến hành nhập thông tin số tiền cần rút và kì hạn tiền đã gửi trước đó. | |
| **Điều kiện trước** | - Khách hàng đã đăng ký tài khoản trên ứng dụng.  - Khách hàng sở hữu tài khoản MetaMask. | |
| **Điều kiện sau** | - Khách hàng rút được tiền từ hệ thống ngân hàng.  - Số tiền khách hàng tăng bằng số tiền khách hàng đã rút.  - Hệ thống sẽ lưu hồ sơ hoạt động giao dịch rút tiền. | |
| **Các luồng sự kiện** | **Hành động** | **Phản hồi của**  **hệ thống** |
| - Chọn mục Withdrawal  - Nhập số tiền cần rút  - Nhập hạn tiền đã gửi  - Xác nhận phí transaction | - Kiểm tra thông tin tài khoản IPFS và MetaMask.  - Kiểm tra thông tin rút tiền  - Hiển thị transaction rút tiền  - Xác nhận tiền rút  - Lưu thông tin giao dịch trên MetaMask  - Cập nhật thông tin tài khoản khách hàng |

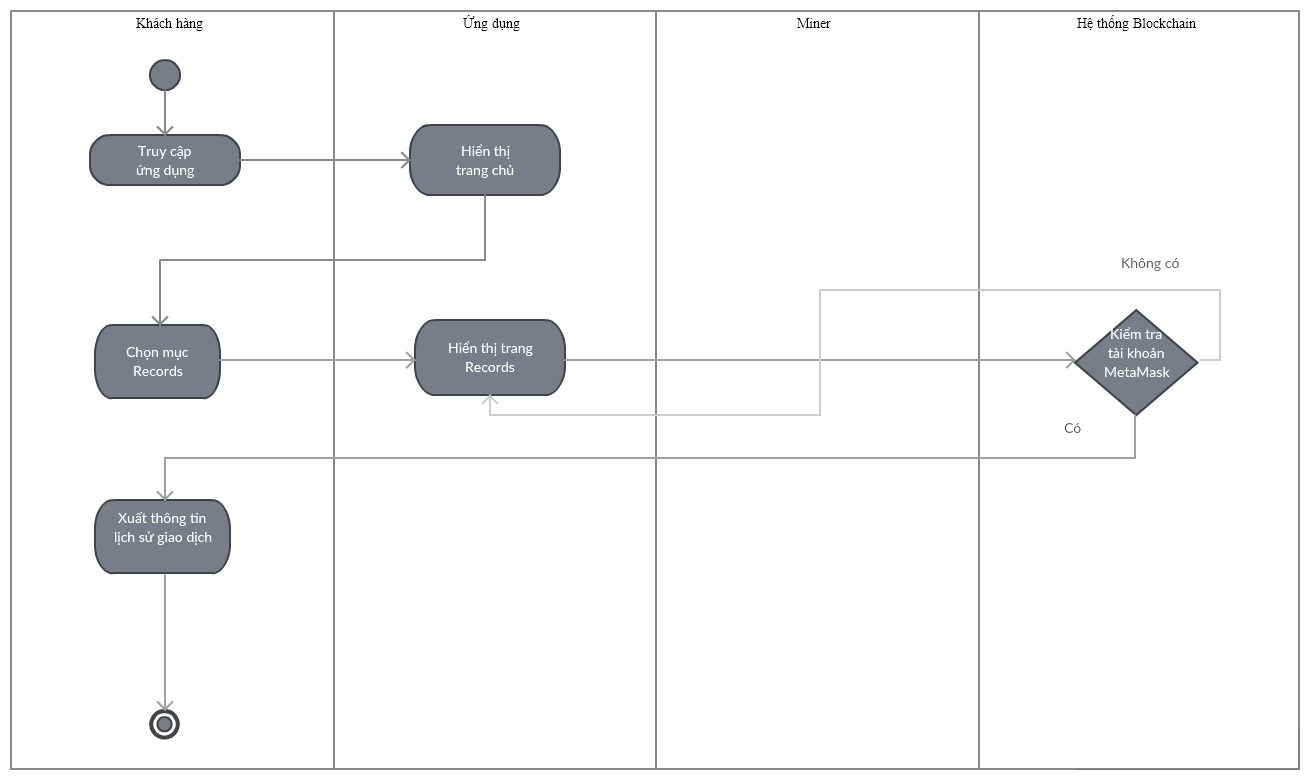
**II. Lưu đồ hoạt động**

****

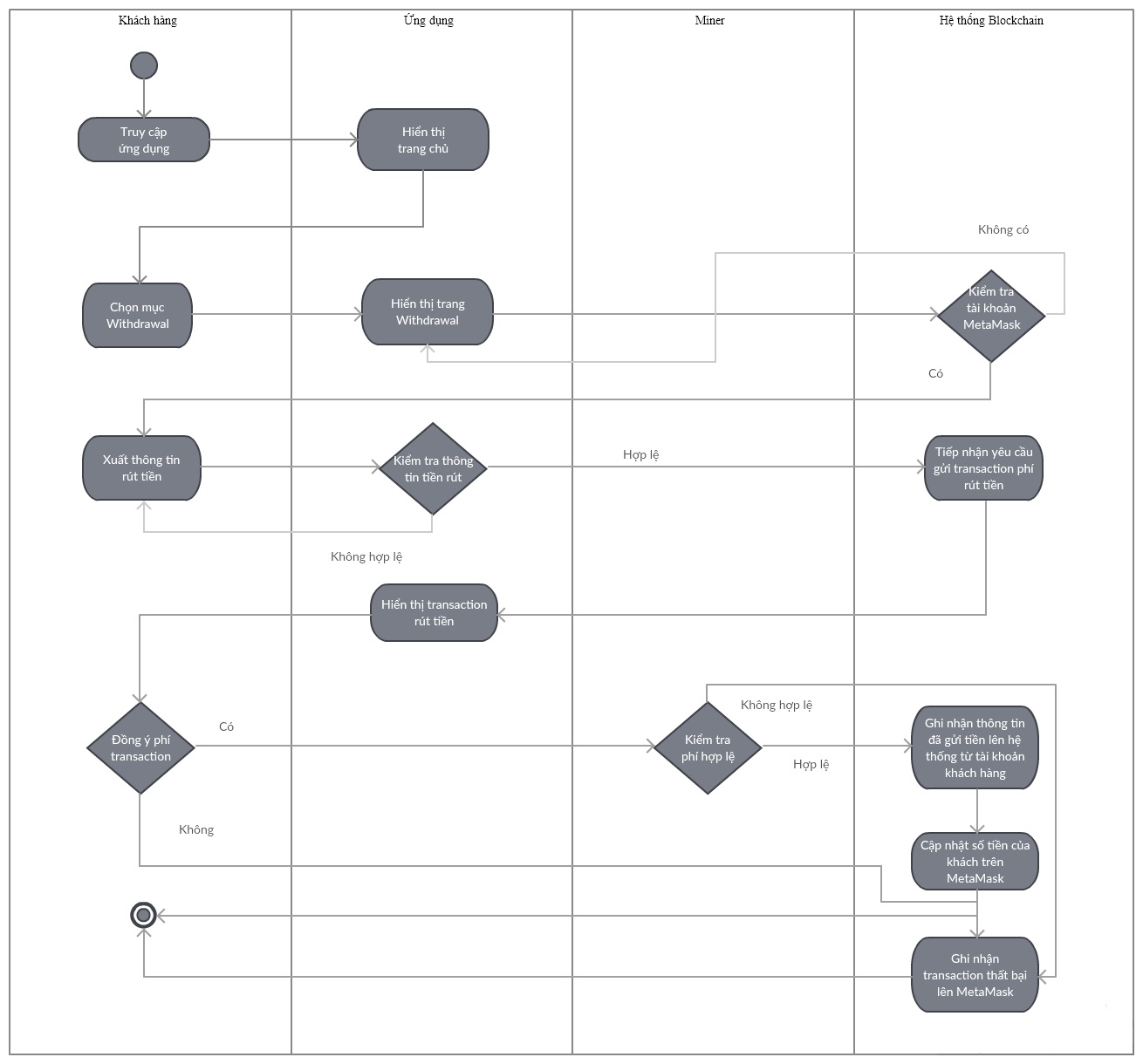
**Hình 20: Lưu đồ hoạt động quản lý tài khoản**

****

**Hình 21: Lưu đồ hoạt động chức năng gửi tiền**

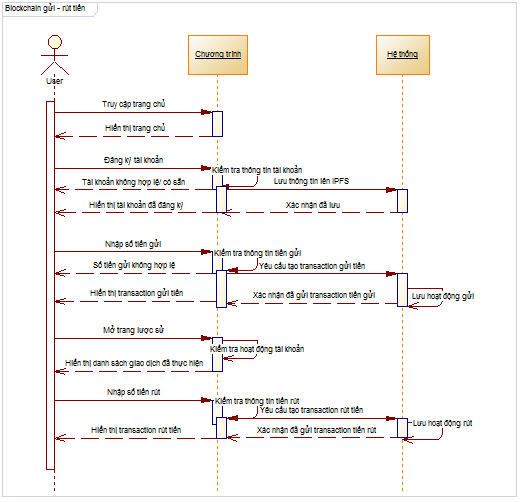
****

**Hình 22: Lưu đồ hoạt động hiển thị lịch sử giao dịch**

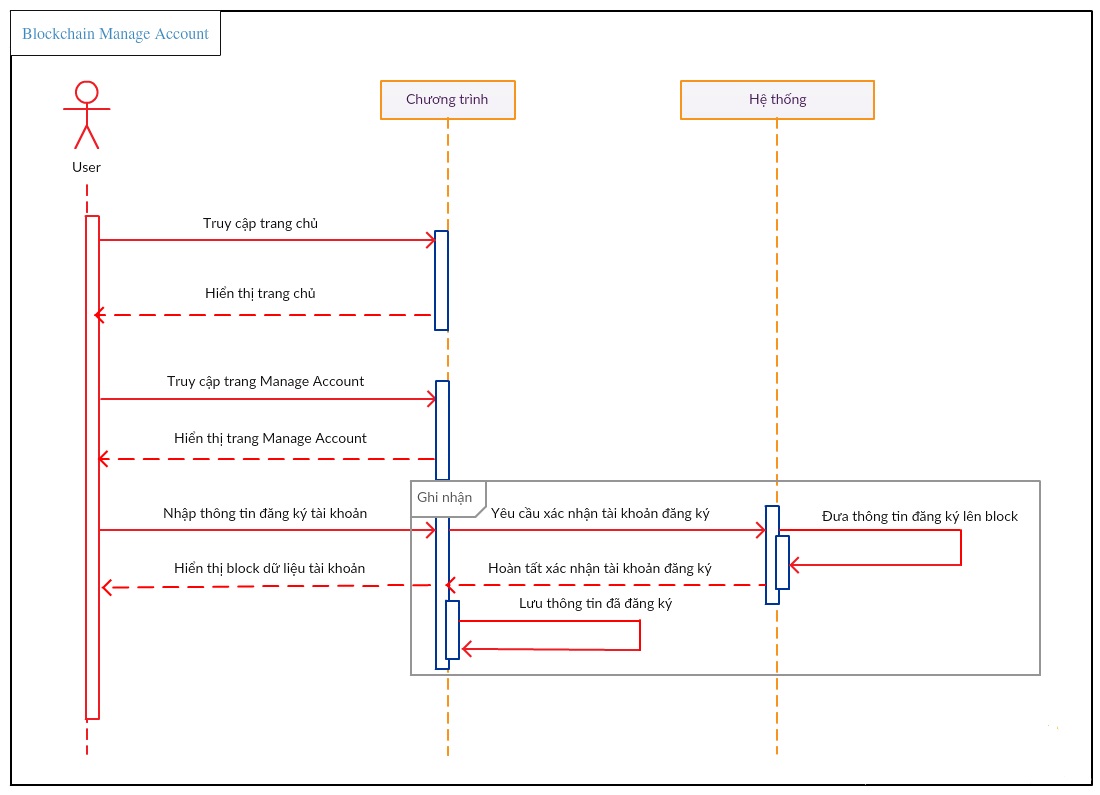
****

**Hình 23: Lưu đồ hoạt động chức năng rút tiền**

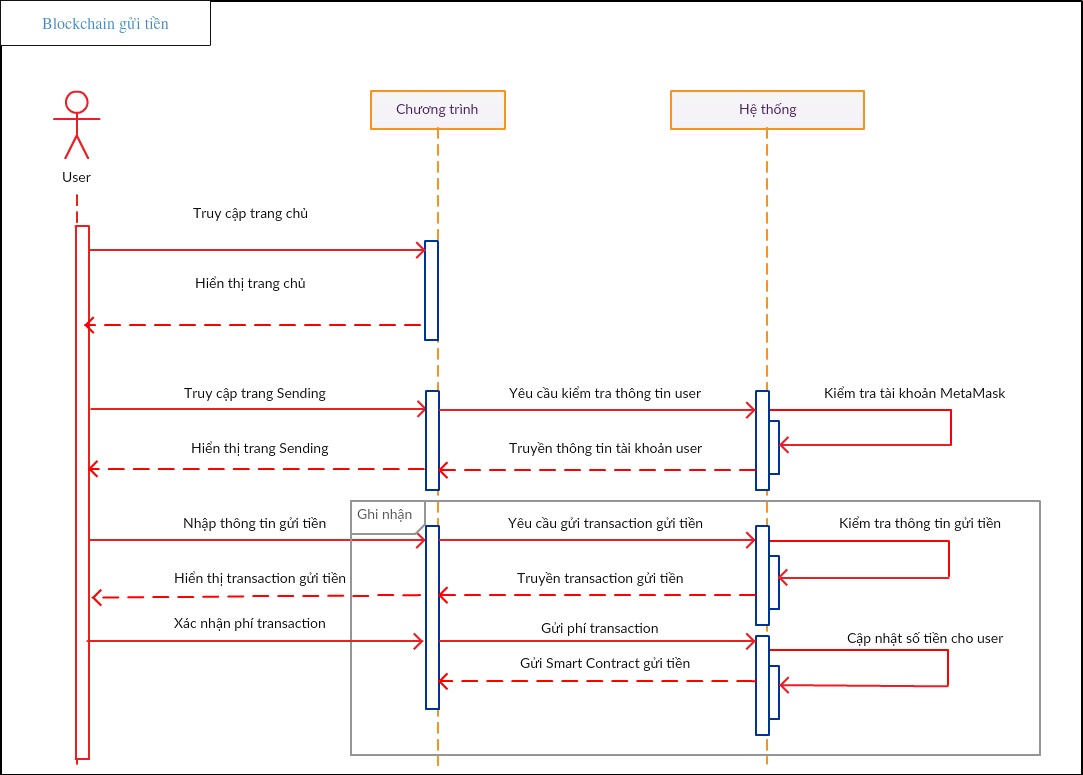
**III. Lưu đồ tuần tự**



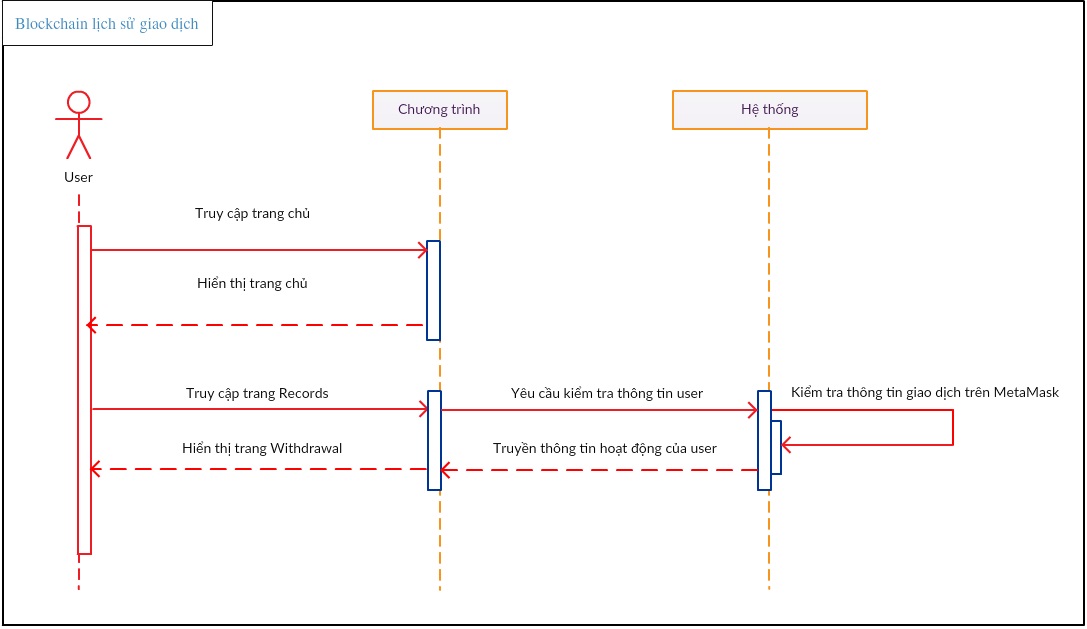
**Hình 24: Lưu đồ tuần tự tổng quan**

****

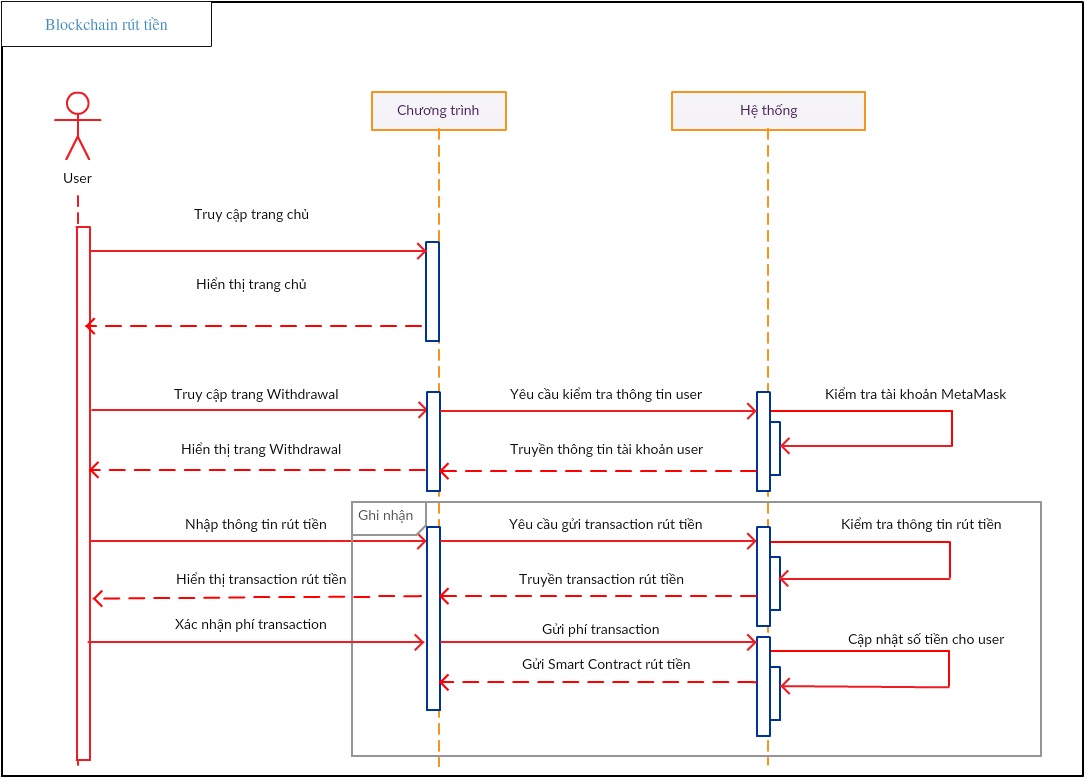
**Hình 25: Lưu đồ tuần tự mô tả quản lý tài khoản**

****

**Hình 26: Lưu đồ tuần tự mô tả tính năng gửi tiền**

****

**Hình 27: Lưu đồ tuần tự mô tả tính năng xem lịch sử giao dịch**

****

**Hình 28: Lưu đồ tuần tự mô tả tính năng rút tiền**

**CHƯƠNG IV: THỰC NGHIỆM**

**I. Cài đặt môi trường**

Để ứng dụng có thể chạy bình thường và cho ra kết quả theo ý muốn thì cần phải có môi trường làm việc, sau đây là phần giới thiệu môi trường làm việc cũng như công cụ xây dựng ứng dụng Blockchain cho đề tài.

I.1. Thành phần tổng quan

**-** Phần cứng:

+ Bộ xử lý: Intel® Core ™ i7 – 4510U CPU @ 2.00 GHz - 2.60 GHz

+ Bộ nhớ RAM: 8.00 GB

+ Hệ điều hành: WIN 8.1 PRO 64-bit

+ Đầu nối VGA 15-pin

+ Bàn phím laptop DELL

+ Chuột laptop DELL và chuột Logitech cảm ứng không dây

- Phần mềm:

+ Sublime Text 3

+ Google Chrome

+ Git Bash

+ Google Drive

+ MetaMask

+ IPFS

+ NodeJS

+ Web3

+ XAMPP

+ Webpack 3.0

+ Infura

- Dữ liệu:

+ Thông tin tài khoản người dùng

+ Tài sản người dùng (Tiền điện tử - Đồng ETH)

+ Thông tin giao dịch

+ Lịch sử giao dịch

+ Phí transaction

- Ngôn ngữ: PHP – JavaScript

I.2. Chương trình hỗ trợ

I.2.1. MetaMask

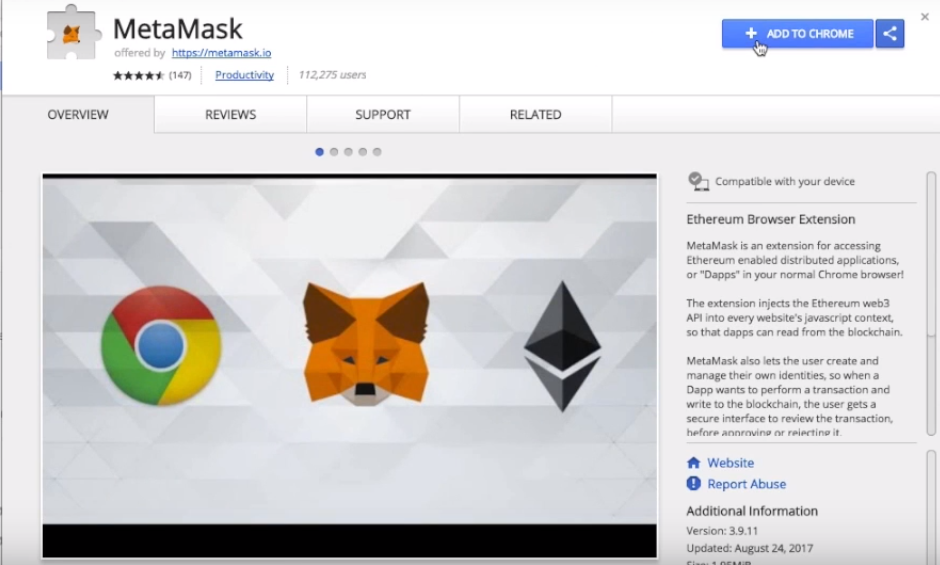
MetaMask là đường dẫn cho phép truy cập vào trang web phân tán trong trình duyệt của chính bạn. Ngoài ra Metamask còn cho phép bạn chạy các ứng dụng khác như Ethereum dApps ngay trong trình duyệt mà bạn không cần khởi động toàn bộ node Ethereum dẫn đến trình duyệt trở nên nặng nề và chậm chạp.

Các bước cài đặt MetaMask như sau:

Bước 1: Truy cập trang chủ MetaMask: *https://metamask.io/*

Bước 2: Chọn Get Chrome Extension để tải về trình duyệt Chrome.

Bước 3: Nhấp vào nút ADD TO CHROME màu xanh bên tay phải trên cùng.



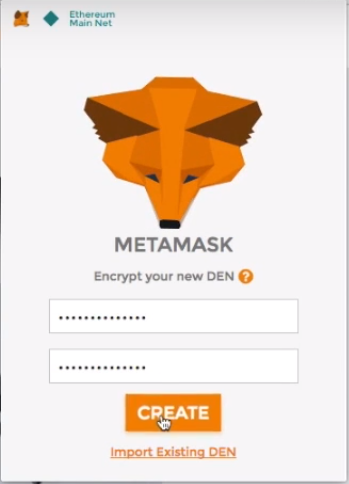
**Hình 29: Giao diện cài đặt ứng dụng MetaMask trên Chrome**

Bước 4: Tiếp đến nhấp nút Add Extension ở cửa sổ hiện ra để hoàn tất cài đặt.

Các bước tạo tài khoản MetaMask như sau:

Bước 1: Sau khi cài đặt MetaMask, nhấp vào biểu tượng MetaMask trên thanh công cụ bên phải.

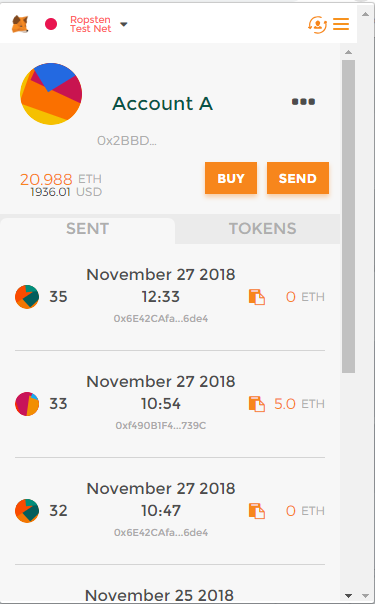
Bước 2: Giao diện MetaMask sẽ mở ra với các trang điều lệ, sau đó kéo dọc hết các điều lệ và nhấp nút Accept để hoàn tất điều lệ.



**Hình 30: Giao diện thiết lập mật khẩu tài khoản MetaMask**

Bước 3: Kế đến, MetaMask hiển thị trang nhập mật khẩu mới cho phép người dùng điền thông tin mật khẩu khi truy cập lần đầu, sau đó nhấp nút CREATE; nếu người dùng trước đó đã có lưu dòng mã từ dưới dạng file DEN thì có thể nhấp vào dòng Import Existing DEN để tải lên thông tin tài khoản cũ cho ứng dụng.

Bước 4: Một bảng hiển thị hiện ra cùng với một dòng các từ ngẫu nhiên, ở đây người dùng sẽ tiến hành sao chép các từ đó để phòng trường hợp sử dụng tài khoản cũ, và sau khi sao chép xong, người dùng nhấp vào nút I’VE COPIED IT SOMEWHERE SAFE để hoàn tất tạo tài khoản trên MetaMask.



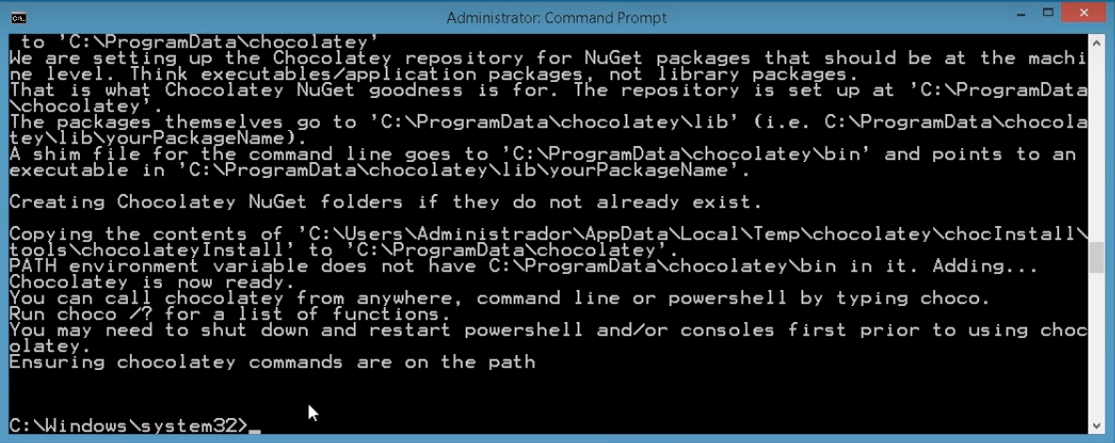
**Hình 31: Giao diện MetaMask sau khi thiết lập tài khoản mới**

II.2.2. IPFS

IPFS với hình thức là một giao thức hypermedia ngang hàng để làm cho web nhanh hơn, an toàn hơn và mở hơn. Ngoài ra, chính hệ thống này là lí do giúp cho các trang web, ứng dụng không chỉ ở Blockchain mà ở tất cả loại hình khác áp dụng nhằm làm tăng tuổi thọ của chương trình cũng như chống lại mọi hình thức gian lận, ăn cắp thông tin đến từ bên ngoài.

Để chạy được IPFS trên máy tính bàn, ta cài đặt Chocolatey bằng cách mở chương trình Command Prompt và nhập dòng lệnh sau:

*@"%SystemRoot%\System32\WindowsPowerShell\v1.0\powershell.exe" -NoProfile -InputFormat None -ExecutionPolicy Bypass -Command "iex ((New-Object System.Net.WebClient).DownloadString('https://chocolatey.org/install.ps1'))" && SET "PATH=%PATH%;%ALLUSERSPROFILE%\chocolatey\bin"*



**Hình 32: Giao diện Command Prompt sau khi cài đặt Chocolatey**

Kế đến, công việc cài đặt IPFS được thực hiện qua các bước sau:

Bước 1: Mở chương trình Command Prompt và nhập dòng lệnh sau:

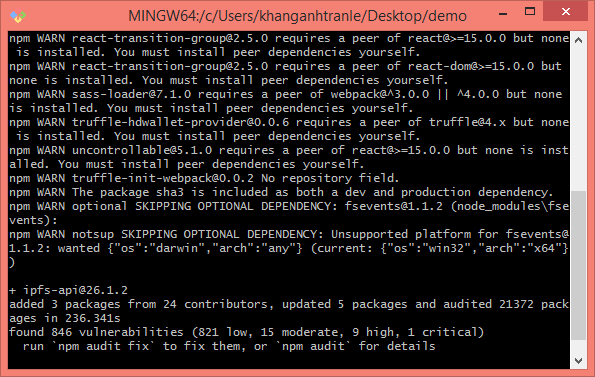
*choco install ipfs*

Bước 2: Sau đó, tiến hành cấu hình IPFS lần lượt bằng các dòng lệnh sau:

*ipfs config --json API.HTTPHeaders.Access-Control-Allow-Origin "[\"\*\"]"  
ipfs config --json API.HTTPHeaders.Access-Control-Allow-Credentials "[\"true\"]"  
ipfs daemon*

Bước 3: Sau khi cấu hình xong, ta cài đặt thư viện IPFS cho thư mục chương trình bằng Git Bash và hoàn tất với dòng lệnh sau:

*npm install --save ipfs-api*



**Hình 33: Thư viện cho IPFS sau khi cài đặt trên Git Bash**

**II. Hoạt động của hệ thống**

Về tổng quan, hệ thống của chương trình hoạt động được thể hiện ở các bước như sau:

Bước 1: Hệ thống hiển thị trang chủ khi người dùng truy cập trang

Bước 2: Khi người dùng truy cập trang Manage Account, hệ thống sẽ vừa hiển thị trang, vừa kiểm tra tài khoản khách hàng trên ứng dụng ngân hàng.

Bước 3: Khi người dùng nhấp nút Send sau khi hoàn tất điền thông tin đăng ký tài khoản, hệ thống sẽ kiểm tra thông tin mà khách hàng xuất.

Bước 4: Nếu người dùng nhập sai thì hệ thống sẽ không thay đổi, còn nếu nhập đúng thì một bảng transaction hiện ra, cho phép người dùng xác nhận phí đăng ký tài khoản.

Bước 5: Sau khi hoàn tất nhập phí transaction, hệ thống sẽ lưu thông tin của khách hàng lên Blockchain, sau đó hiển thị block dữ liệu trên trang Manage Account.

Bước 6: Khi người dùng truy cập trang Sending, hệ thống sẽ kiểm tra thông tin chủ yếu trên MetaMask và hiển thị giao diện gồm thông tin người dùng cùng với form nhập tiền gửi.

Bước 7: Khi người dùng nhấp nút Send sau khi hoàn tất điền thông tin gửi tiền, hệ thống sẽ kiểm tra thông tin mà khách hàng xuất.

Bước 8: Nếu người dùng nhập sai thì hệ thống ghi nhận transaction thất bại trên tài khoản MetaMask, còn nếu nhập đúng thì một bảng transaction hiện ra, cho phép người dùng xác nhận phí rút tiền.

Bước 9: Sau khi hoàn tất nhập phí transaction, hệ thống sẽ lưu thông tin giao dịch của khách hàng, sau đó cập nhật số tiền khách có trên MetaMask.

Bước 10: Khi người dùng truy cập trang Records, hệ thống sẽ vừa hiển thị trang, vừa kiểm tra tài khoản khách hàng trên ứng dụng ngân hàng.

Bước 11: Hệ thống hiển thị danh sách các Requests về thông tin hoạt động giao dịch hoàn tất của khách hàng.

Bước 12: Khi người dùng truy cập trang Withdrawal, hệ thống sẽ vừa hiển thị trang, vừa kiểm tra tài khoản khách hàng trên ứng dụng ngân hàng.

Bước 13: Khi người dùng nhấp nút Withdraw sau khi hoàn tất điền thông tin rút tiền, hệ thống sẽ kiểm tra thông tin mà khách hàng xuất.

Bước 14: Nếu người dùng nhập sai thì hệ thống sẽ ghi nhận transaction thất bại trên tài khoản MetaMask, còn nếu nhập đúng thì một bảng transaction hiện ra, cho phép người dùng xác nhận phí rút tiền.

Bước 15: Sau khi hoàn tất nhập phí transaction, hệ thống sẽ lưu thông tin giao dịch của khách hàng, sau đó cập nhật số tiền khách có trên MetaMask.

**III. Test Case**

|  |  |
| --- | --- |
| **Tên đề tài: Ứng dụng Blockchain trong gửi tiền tiết kiệm tại ngân hàng** | **Test Case** |
| Test Case ID: Account A | Thiết kế: Khang, Hiếu, Ân |
| Mức độ thử nghiệm: Vừa | Ngày thiết kế: 16/12/2018 |
| Tên Module: Quản lý tài khoản (Manage Account) | Thử nghiệm bởi: Khang |
| Tiêu đề: Thử nghiệm tính năng tự động quản lý tài khoản đã đăng ký trên trang. | Ngày thử: 16/12/2018 |
| Mô tả: Xác minh thông tin tài khoản có tồn tại hay không. | Kết quả: Đạt |

|  |  |
| --- | --- |
| **Tên đề tài: Ứng dụng Blockchain trong gửi – rút tiền tiết kiệm tại ngân hàng** | **Test Case** |
| Test Case ID: Account B | Thiết kế: Khang, Hiếu, Ân |
| Mức độ thử nghiệm: Vừa | Ngày thiết kế: 16/12/2018 |
| Tên Module: Tính năng gửi – rút tiền (Sending và Withdrawal) | Thử nghiệm bởi: Khang |
| Tiêu đề: Thử nghiệm khả năng gửi – rút tiền dựa trên số tài sản mà người dùng có. | Ngày thử: 16/12/2018 |
| Mô tả: Tiến hành công việc gửi tiền tiết kiệm vào ứng dụng ngân hàng | Kết quả: Đạt |

|  |  |
| --- | --- |
| **Tên đề tài: Ứng dụng Blockchain trong gửi tiền tiết kiệm tại ngân hàng** | **Test Case** |
| Test Case ID: Account C | Thiết kế: Khang, Hiếu, Ân |
| Mức độ thử nghiệm: Vừa | Ngày thiết kế: 16/12/2018 |
| Tên Module: Tính năng xem lịch sử hoạt động giao dịch (Records) | Thử nghiệm bởi: Khang |
| Tiêu đề: Thử nghiệm tính năng hiển thị các hoạt động giao dịch đã hoàn tất. | Ngày thử: 16/12/2018 |
| Mô tả: Hệ thống hiển thị thông tin các giao dịch Requests đã hoàn tất. | Kết quả: Chưa đạt do hiển thị chậm các hoạt động Request mới hoàn tất. |

**IV. Kết quả thực nghiệm**

Sau khi thử nghiệm, chương trình, về mặt bằng chung, đã đáp ứng các yêu cầu cơ bản về mục tiêu của nhóm cũng như của khóa học Seminar lần này.

Tuy nhiên, khi nói về chức năng nâng cao, nhóm dường như không thể khiến cho chương trình tự động hóa việc gộp toàn bộ thông tin giao dịch trong cùng một khoảng thời gian và đẩy lên block nhằm hạn chế chi phí transaction nói chung và lãng phí thời gian ở phía người dùng nói riêng.

**CHƯƠNG V: KẾT LUẬN**

**I. Lời kết**

Sau gần 4 tháng học tập và nghiên cứu về lĩnh vực Blockchain, nhóm chúng tôi đã đúc kết được một số kiến thức hữu ích về các thành phần như số Nonce, thuật toán hash và cơ chế đồng thuận PoW là yếu tố then chốt trong công tác bảo mật cũng như cách Blockchain vận hành trong việc hình thành chuỗi hash để mã hóa thông tin giao dịch cũng như các thông tin nhạy cảm trong transaction.

Về cơ bản, chương trình do nhóm phát triển thực hiện đúng yêu cầu mục tiêu của nhóm đề ra ban đầu chính là phục vụ công việc gửi và rút tiền tiết kiệm dành cho người dùng, cụ thể là khách hàng, đồng thời thiết lập thành công smart contract cũng như IPFS dành cho việc lưu trữ thông tin giao dịch.

Tuy nhiên, bản thân nhóm có nhiều mặt hạn chế trong việc phát triển chương trình, dẫn đến việc hao phí thời gian vào những lỗi bug không cần thiết, ảnh hưởng đến việc thử nghiệm test-case và phát triển một số chức năng nâng cao khác.

**II. Hướng phát triển**

Chương trình do nhóm phát triển có thể được dùng để xây dựng bổ sung một số thành phần sau:

- Về tính năng:

+ Tính năng cho vay ngân hàng nhiều phía.

+ Tính năng chuyển đổi dạng tiền tệ.

+ Tính năng in thông tin giao dịch và lịch sử giao dịch.

+ Tính năng lưu chữ ký điện tử dành cho doanh nghiệp.

+ Tính năng chuyển giao đa tài sản giữa người dùng.

+ Tính năng quét mã QRCODE cho việc đăng ký, đăng nhập tài khoản.

- Về hệ thống: Có thể đẩy chương trình lên MainNet thay cho TestNet hiện hành cùng với các tính năng bổ sung nâng cao theo yêu cầu người dùng.

**TÀI LIỆU THAM KHẢO**

[1] http://www.sapcoin.net/blockchain/blockchain-by-melanie-swan.pdf

[2] https://beta.vu.nl/nl/Images/werkstuk-bruyn\_tcm235-862258.pdf

[3] https://blockonomi.com/interplanetary-file-system

[4] https://medium.com/@ConsenSys/an-introduction-to-ipfs-9bba4860abd0

**PHỤ LỤC**

- Phát triển chương trình là điều cần thiết, và thứ quan trọng hơn chính là việc thu nhận kiến thức từ mọi nơi, bao gồm cả từ Internet, mạng xã hội và con người.

- Để giúp cho mọi người tiếp cận gần hơn với công nghệ Blockchain, nhóm xin đưa ra đường dẫn chương trình nhằm giúp cho mọi người đều có khả năng phát triển ứng dụng Công nghệ thông tin nói chung và công nghệ Blockchain nói riêng.

- Github: *https://github.com/anhkhangle/BLOCKCHAIN\_GDGUITIETKIEM\_Nhom01\_2018*