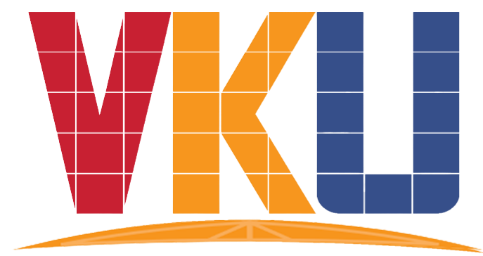
ĐẠI HỌC ĐÀ NẴNG

TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG VIỆT HÀN

KHOA KHOA HỌC MÁY TÍNH



ĐỒ ÁN TỐT NGHIỆP

Sàn giao dịch blockchain

SINH VIÊN : Phan Đức Hải 18IT319

: Nguyễn Văn Nam 18IT340

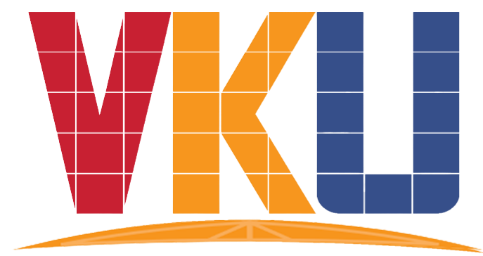
LỚP SH : 18T5

GVHD : TS.Nguyễn Đức Hiển

Đà Nẵng, 12 tháng 2022

TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG VIỆT HÀN

KHOA KHOA HỌC MÁY TÍNH



ĐỒ ÁN TỐT NGHIỆP

Sàn giao dịch blockchain

SINH VIÊN : Phan Đức Hải 18IT319

: Nguyễn Văn Nam 18IT340

LỚP SH : 18T5

GVHD : TS.Nguyễn Đức Hiển

Đà Nẵng, 12 tháng 2022

**Đà Nẵng, 03/2022**

.................................................................................................................................

.................................................................................................................................

.................................................................................................................................

.................................................................................................................................

.................................................................................................................................

.................................................................................................................................

.................................................................................................................................

.................................................................................................................................

.................................................................................................................................

.................................................................................................................................

.................................................................................................................................

.................................................................................................................................

.................................................................................................................................

.................................................................................................................................

*Đà Nẵng, ngày tháng 12 năm 2022*

Giảng viên hướng dẫn

**TS. Nguyễn Đức Hiển**

.................................................................................................................................

.................................................................................................................................

.................................................................................................................................

.................................................................................................................................

.................................................................................................................................

.................................................................................................................................

.................................................................................................................................

.................................................................................................................................

.................................................................................................................................

.................................................................................................................................

.................................................................................................................................

.................................................................................................................................

.................................................................................................................................

.................................................................................................................................

.................................................................................................................................

.................................................................................................................................

.................................................................................................................................

.................................................................................................................................

.................................................................................................................................

.................................................................................................................................

.................................................................................................................................

..............................................................................................................................

## TÓM TẮT

Tên đề tài: Xây dựng ứng dụng flashloan trên mạng thử nghiệm Kovan Sinh viên thực hiện 1: Phan Đức Hải

Mã số SV:18IT319

Lớp: 18IT5

Sinh viên thực hiện 2: Nguyễn Văn Nam

Mã số SV:18IT340 Lớp: 18IT5

Tóm tắt đề tài: Flashloan có thể cho vay mà không cần tài sản thế chấp, không cần chứng minh thu nhập, chỉ cần một tài khoản ví tiền điện tử và một khoản tiền nhỏ để làm phí khởi tạo giao thức vay.Vì flashloan là một hình thức vay diễn ra trên blockchain và toàn bộ quá trình vay và thao tác các hành động của người dùng yêu cầu tồn tại trong một giao dịch (transaction) duy nhất nên đảm bảo tính toàn vẹn của khoản vay. Ngoài phí giao thức thì không phát sinh thêm bất kỳ khoản phí nào mà người dùng phải trả và giao dịch diễn ra nhanh chóng nên không có lãi suất theo thời gian mà đó chỉ là phí giao thức mỗi lần khởi tạo đã được đề cập ở trên.

Trong đồ án này, em sẽ làm sáng tỏ khả năng cho phép vay mà không ràng buộc của hình thức vay flashloan. Em sẽ làm việc với Smart Contract để xây dựng ứng dụng cho phép vay flashloan và dùng khoản tiền vay được dể tạo ra thu nhập dựa trên hình thức chênh lệch giá của các đồng tiền điện tử đang lưu hành.

Trong suốt thời gian thực hiện và hoàn thành đồ án này, em đã nhận được sự giúp đỡ và hướng dẫn tận tình của các Thầy, các Cô và các bạn trong Khoa Khoa học máy tính Trường Đại học công nghê thông tin và truyền thông Việt Hàn - Đại học Đà Nẵng. Em xin gửi lời cảm ơn chân thành và sâu sắc tới các Thầy Cô trong Khoa đã giảng dạy và truyền đạt những kiến thức cần thiết, những kinh nghiệm quý báu cho em có thể thực hiện đồ án này.

Trước tiên, em muốn cảm ơn người hướng dẫn đồ án của em, TS. Nguyễn Đức Hiển đã hướng dẫn, giúp đỡ, và thảo luận đã cho em kiến thức và nhiệt tình để thiết kế và phát triển ứng dụng này.

Do kiến thức và kinh nghiệm của bản thân còn hạn chế nên việc thực hiện không thể tránh khỏi những thiếu sót. Rất mong được sự đóng góp ý kiến của thầy cô trong bộ môn công nghệ thông tin trường Đại học công nghê thông tin và truyền thông Việt Hàn để đồ án của em được hoàn thiện hơn.

Cuối cùng, em xin chân thành cảm ơn thầy cô trong khoa Công nghệ thông tin trường Đại học công nghê thông tin và truyền thông Việt Hàn, đặc biệt là TS. Nguyễn Đức Hiển đã hướng dẫn, giúp đỡ em hoàn thành đồ án tốt nghiệp này.

Em xin chân thành cảm ơn!

Đà Nẵng, ngày 16 tháng 12 năm 2022

Sinh viên (Nhóm trưởng)

Nguyễn Văn Nam

[MỞ ĐẦU 10](#_Toc97325979)

[1. Tổng quan về đề tài 10](#_Toc97325980)

[2. Mục đích và ý nghĩa của đề tài 10](#_Toc97325981)

[2.1. Mục đích 10](#_Toc97325982)

[2.2. Ý nghĩa 10](#_Toc97325983)

[2.2.1. Phương pháp thực hiện 11](#_Toc97325984)

[2.2.2. Bố cục đồ án 11](#_Toc97325985)

[Chương 1. CƠ SỞ LÝ THUYẾT 12](#_Toc97325986)

[1.1. Nghiên cứu lý thuyết 12](#_Toc97325987)

[1.1.1. Tổng quan về ReactJS 12](#_Toc97325988)

[1.1.2. Ethereum 16](#_Toc97325989)

[1.1.3. Smart contract 18](#_Toc97325990)

[1.1.4. Tổng quan về Solidity 19](#_Toc97325991)

[1.1.5. KeystoneJS 20](#_Toc97325992)

[1.1.6. GraphQL 22](#_Toc97325993)

[1.1.7. Phần kết luận 24](#_Toc97325994)

[Chương 2. PHÂN TÍCH VÀ THIẾT KẾ HỆ THỐNG 25](#_Toc97325995)

[2.1. Các chức nặng chính 25](#_Toc97325996)

[2.1.1. Xem thông tin thị trường tiền điện tử 25](#_Toc97325997)

[2.1.2. Tạo ví điện tử lưu trữ Ethereum 25](#_Toc97325998)

[2.1.3. Tạo, thêm tổng cung cho token riêng của bạn 25](#_Toc97325999)

[2.1.4. Vay nhanh với tùy chỉnh tham số 26](#_Toc97326000)

[2.1.5. Lưu lịch sử vay nhanh và thống kê 26](#_Toc97326001)

[2.1.6. Quản lý đăng nhập để truy xuất thông tin lịch sử giao dịch 26](#_Toc97326002)

[2.1.7. Mời bạn bè tham gia trải nghiệm ứng dụng 26](#_Toc97326003)

[2.2. Sơ đồ ca sử dụng 27](#_Toc97326004)

[2.2.1. Tổng thể sơ đồ 27](#_Toc97326005)

[2.2.2. Xem thông tin thị trường tiền điện tử 28](#_Toc97326006)

[2.2.3. Tạo ví điện tử lưu trữ Ethereum 29](#_Toc97326007)

[2.2.4. Tạo, thêm tổng cung cho token riêng của bạn 30](#_Toc97326008)

[2.2.5. Vay nhanh với tùy chỉnh tham số và lưu lịch sử giao dịch 31](#_Toc97326009)

[2.3. Sơ đồ tuần tự 32](#_Toc97326010)

[2.3.1. Xem thông tin thị trường tiền điện tử 32](#_Toc97326011)

[2.3.2. Tạo ví điện tử lưu trữ Ethereum 32](#_Toc97326012)

[2.3.3. Tạo, thêm tổng cung cho token riêng của bạn 33](#_Toc97326013)

[2.3.4. Vay nhanh với tùy chỉnh tham số và lưu lịch sử giao dịch 34](#_Toc97326014)

[2.4. Sơ đồ lớp 34](#_Toc97326015)

[2.5. Cơ sở dữ liệu 35](#_Toc97326016)

[Chương 3: TRIỂN KHAI VÀ CHẠY THỬ NGHIỆM 37](#_Toc97326017)

[3.1. Môi trường triển khai 37](#_Toc97326018)

[3.2. Triển khai hệ thống 37](#_Toc97326019)

[3.3. Kết quả demo và hướng dẫn sử dụng chương trình 43](#_Toc97326020)

[3.2.1. Xem thông tin thị trường tiền điện tử 43](#_Toc97326021)

[3.2.2. Tạo ví điện tử lưu trữ Ethereum 44](#_Toc97326022)

[3.2.3. Tạo, thêm tổng cung cho token riêng của bạn 45](#_Toc97326023)

[3.2.4. Vay nhanh với tùy chỉnh tham số 46](#_Toc97326024)

[3.2.5. Lưu lịch sử vay nhanh và thống kê 51](#_Toc97326025)

[3.2.6. Quản lý đăng nhập để truy xuất thông tin lịch sử giao dịch 53](#_Toc97326026)

[3.2.7. Mời bạn bè tham gia trải nghiệm ứng dụng 54](#_Toc97326027)

[3.2.8. Đánh giá 55](#_Toc97326028)

[KẾT LUẬN 56](#_Toc97326029)

[1. Thành tựu 56](#_Toc97326030)

[2. Hướng phát triển 56](#_Toc97326031)

[TÀI LIỆU THAM KHẢO 57](#_Toc97326032)

## DANH SÁCH CÁC BẢNG, HÌNH VẼ

### Danh sách các hình vẽ

[Hình 1.1 Jordan Walke, người sáng lập ra ReactJS. 12](#_Toc96891718)

[Hình 1.2 Component trong ReactJS. 13](#_Toc96891719)

[Hình 1.3 Sử dụng state trong Component. 14](#_Toc96891720)

[Hình 1.4 Sơ đồ vòng đời của component trong ReactJS 14](#_Toc96891721)

[Hình 1.5 Ethereum 16](#_Toc96891722)

[Hình 1.6 Solidity 19](#_Toc96891723)

[Hình 1.7 KeystoneJS 20](#_Toc96891724)

[Hình 1.9 Khác biệt giữa Rest và GraphQL 23](#_Toc96891725)

[Hình 2.1 Sơ đồ tổng thể hệ thống 27](#_Toc96891726)

[Hình 2.2 Sơ đồ ca sử dụng xem thông tin thị trường tiền điện tử 28](#_Toc96891727)

[Hình 2.3 Sơ đồ ca sử dụng tạo ví điện tử ethereum 29](#_Toc96891728)

[Hình 2.4 Sơ đồ ca sử dụng tương tác với token 30](#_Toc96891729)

[Hình 2.5 Sơ đồ ca sử dụng chức năng vay flash loan 31](#_Toc96891730)

[Hình 2.6 Sơ đồ tuần tự xem thông tin thị trường tiền điện tử 32](#_Toc96891731)

[Hình 2.7 Sơ đồ tuần tự chức năng tạo ví điện tử ethereum 32](#_Toc96891732)

[Hình 2.8 Sơ đồ vòng tuần tự tạo token 33](#_Toc96891733)

[Hình 2.9 Sơ đồ tuần tự thêm tổng cung cho token 33](#_Toc96891734)

[Hình 2.10 Sơ đồ tuần tự chức năng vay flash loan 34](#_Toc96891735)

[Hình 2.11 Sơ đồ lớp hệ thống 34](#_Toc96891736)

[Hình 2.11 Sơ đồ cơ sở dữ liệu hệ thống 36](#_Toc96891737)

[Hình 3.1 Cấu trúc hàm vay flash loan của giao thức 38](#_Toc96891738)

[Hình 3.2 Cấu trúc hàm swap token ở Uniswap V1 39](#_Toc96891739)

[Hình 3.3 Cấu trúc hàm swap token của Uniswap V2 40](#_Toc96891740)

[Hình 3.4 Cặp giao dịch ITF-DAI đã được thêm ở uniswap va sushiswap 40](#_Toc96891741)

[Hình 3.5 Sơ đồ hoạt động của ứng dụng flash loan 41](#_Toc96891742)

[Hình 3.6 Sơ đồ luồng hoạt động của code 42](#_Toc96891743)

[Hình 3.7 Giá chênh lệch của GRT token trên các sàn tập trung 42](#_Toc96891744)

[Hình 3.8 Giao diện xem thông tin thị trường tiền điện tử 43](#_Toc96891745)

[Hình 3.9 Giao diện tạo ví điện tử ethereum 44](#_Toc96891746)

[Hình 3.10 Giao diện mint token 45](#_Toc96891747)

[Hình 3.11 Giao diện khởi tạo khoản vay flash loan 46](#_Toc96891748)

[Hình 3.12 Giao diện chọn loại coin muốn vay 46](#_Toc96891749)

[Hình 3.13 Giao diện chọn số lượng coin muốn vay 47](#_Toc96891750)

[Hình 3.14 Giao diện thêm và tùy chỉnh các hoán đổi coin 47](#_Toc96891751)

[Hình 3.15 Giao diện lựa chọn sàn giao dịch 48](#_Toc96891752)

[Hình 3.16 Giao diện xác nhận các thiết lập 48](#_Toc96891753)

[Hình 3.17 Giao diện chờ kết quả 49](#_Toc96891754)

[Hình 3.18 Xác thực giao dịch trên metamask 49](#_Toc96891755)

[Hình 3.19 Kết quả giao dịch thành công trên kovan.etheescan.io 50](#_Toc96891756)

[Hình 3.20 Kết quả giao dịch thất bại trên kovan.etheescan.io 50](#_Toc96891757)

[Hình 3.21 Giao diện lịch sử giao dịch 51](#_Toc96891758)

[Hình 3.22 Giao diện thống kê dữ liệu lịch sử các giao dịch 51](#_Toc96891759)

[Hình 3.23 Giao diện lịch sử của giao dịch thành công 52](#_Toc96891760)

[Hình 3.24 Giao diện lịch sử giao dịch thất bại 52](#_Toc96891761)

[Hình 3.25 Giao diện thông tin cá nhân 53](#_Toc96891762)

[Hình 3.26 Giao diện mời thêm bạn bè 54](#_Toc96891763)

[Hình 3.27 Giao diện tổng hợp kết quả mời bạn bè 54](#_Toc96891764)

### Danh sách các bảng

[Bảng 2. 1 Bảng cơ sở dữ liệu Customer 35](#_Toc97326065)

[Bảng 2. 2 Bảng cơ sở dữ liệu History 35](#_Toc97326066)

[Bảng 2. 3 Bảng cơ sở dữ liệu Referral 36](#_Toc97326067)

[Bảng 2. 4 Bảng cơ sở dữ liệu Image 36](#_Toc97326068)

[Bảng 3. 1 So sánh các mạng thử nghiệm trên Ethereum 37](#_Toc97326069)

[Bảng 3. 2 Tham số truyền của hàm tokenToTokenSwapInput 39](#_Toc97326070)

[Bảng 3. 3 Tham số truyền của hàm swapExactTokenForTokens 40](#_Toc97326071)

## DANH SÁCH CÁC KÝ HIỆU, CHỮ VIẾT TẮT

CHỮ VIẾT TẮT:

|  |  |  |
| --- | --- | --- |
| **Số thứ tự** | **Chữ viết tắt** | **Diễn giải** |
| 1 | API | Application Programming Interface |
| 2 | HTML | Hypertext Markup Language |
| 3 | HTTP | Hypertext Transfer Protocol |
| 4 | REST | Representational State Transfer |
| 5 | MVC | Model, View, Controller |
| 6 | AJAX | Asynchronous JavaScript and XML |
| 7 | POW | Proof Of Work |
| 8 | POS | Proof Of Stake |
| 9 | POA | Proof Of Authority |
| 10 | CEX | Centralized Exchanges |
| 11 | DEX | Decentralized exchange |
| 12 | NFT | Non-fungible token |

# MỞ ĐẦU

## Tổng quan về đề tài

Theo cách truyền thống, muốn vay được một khoản tiền từ ngân hàng thì yêu cầu người vay phải cần có tài sản thế chấp hoặc thông tin có thể chứng minh được thu nhập có thể hoàn trả khoản vay đúng hạn.Vậy nếu những người nghèo, hoàn cảnh khó khăn không có tài sản có khả năng thế chấp và không đạt điều kiện thu nhập thì họ sẽ không được vay bất cứ khoản tiền nào (không đề cập các trường hợp chính sách). Ứng dụng flashloan có thể giải quyết được vấn đề này trên hệ thống mạng lưới blockchain.

## Mục đích và ý nghĩa của đề tài

### Mục đích

Nhằm giải quyết nhu cầu cho vay mà không cần trãi qua các thủ tục hay ràng buộc phức tạp, mang lại sự linh hoạt, an toàn và minh bạch trong dịch vụ vay vốn, đề tài này đã giải quyết được 3 vấn đề chính như sau :

* + Giúp cho người có nhu cầu vay vốn để đầu tư hay tạo ra lợi nhuận bằng một hình thức nào khác trên thị trường tiền điện tử.
  + Giúp cho người dùng sử dụng tiền điện tử đang vay thế chấp một tài sản A bất kỳ thay thế tài sản thế chấp B bất kỳ khác
  + Giúp cho việc đáo hạn các khoản vay có thế chấp tài sản khác trở nên dễ dàng và linh hoạt

Trong đồ án này, chỉ tập trung vào khả năng tạo ra lợi nhuận bằng hình thức chênh lệch giá, sử dụng cách giao dịch mua bán trên các sàn phi tập trung.

### Ý nghĩa

Flash loan vô tình làm khuếch đại lỗ hổng cũng như hạn chế của Oracle hiện tại, dẫn đến việc hàng loạt nền tảng bị tấn công flash loan. Tuy nhiên, điều này một phần đến từ hạn chế của các chỉ số giá Oracle chứ không hoàn toàn là lỗi của hình thức flash loan này

Hiện tại, Flash loan còn khá mới mẻ, chỉ mới ra mắt từ đầu năm 2020 nhưng ý nghĩa của nó đem lại cho thị trường tiền điện tử rất lớn. Tính linh hoạt và tùy biến của khoản vay này là một mục mới vào không gian tài chính phi tập trung, nhưng chúng chắc chắn sẽ tạo nên một tương lai lâu dài. Khái niệm về các khoản cho vay phi tập trung, chỉ được thực thi bằng mã, mở ra một bước tiến cách mạng trong một hệ thống tài chính mới

Ngoài ra, vì đồ án đang xây dựng trên mạng thử nghiệm Kovan testnet nên là môi trưởng trải nghiệm dịch vụ vay flash loan. Nhằm giúp những người mới, muốn tham gia vào thị trường tiền điện tử có một sân chơi thích ứng mà không tốn một khoản phí nào.

### Phương pháp thực hiện

* Phương pháp phân tích tổng hợp từ tài liệu từ nguồn tài liệu trên mạng
* Phương pháp phân tích thiết kế hệ thống theo hướng đối tượng
* Phương pháp thử nghiệm, đánh giá kết quả.

### Bố cục đồ án

Đồ án bao gồm các nội dung sau:

*Mở đầu*

*Chương 1: Cơ sở lý thuyết*

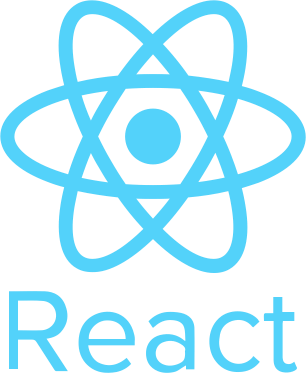
*Chương 2: Phân tích và thiết kế hệ thống Chương 3: Triển khai và đánh giá kết quả Kết luận và hướng phát triển.*

# Chương 1. CƠ SỞ LÝ THUYẾT

## Nghiên cứu lý thuyết

### Tổng quan về ReactJS

* + - 1. ***Giới thiệu***

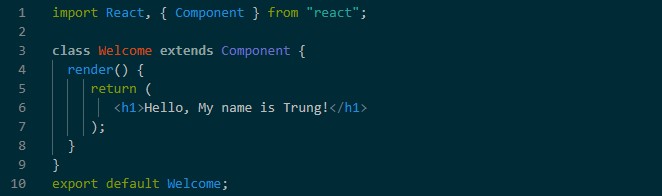
React (hay ReactJS, React.js) là một thư viện Javascript mã nguồn mở để xây dựng các thành phần giao diện có thể tái sử dụng. Nó được tạo ra bởi *Jordan Walke*, một kỹ sư phần mềm tại Facebook. Người bị ảnh hưởng bởi XHP (Một nền tảng thành phần HTML cho PHP). React lần đầu tiên được triển khai cho ứng dụng Newsfeed của Facebook năm 2011, sau đó được triển khai cho Instagram.com năm 2012. Nó được mở mã nguồn (open-sourced) tại JSConf US tháng 5 năm 2013 [7].

Hình 1.1 Jordan Walke, người sáng lập ra ReactJS.

Đặc tính của ReactJS:

* Tư tưởng của ReactJS là xây dựng lên các components có tính tái sử dụng, dễ dàng cho việc chia nhỏ vấn đề, testing. Nó giúp dễ dàng quản lý, mở rộng hệ thống, điều này nếu là Angular JS thì đòi hỏi cấu trúc, cách viết code phải thật tối ưu.
* ReactJS luôn giữ các components ở trạng thái stateless (nhiều nhất có thể) khiến ta dễ dàng quản lý bởi nó chẳng khác gì một trang static HTML. Bản thân các components này không có trạng thái (state) nó nhận đầu vào từ bên ngoài và chỉ hiển thị ra dựa vào các đầu vào đó, điều này giải thích tại sao nó mang tính tái sử dụng và dễ dàng cho việc test.
  + - 1. ***Các đặc trưng của ReactJS***

Component:

* + - * + Là một đoạn mã ngắn có ý nghĩa thể hiện một chức năng hay đối tượng nhất định, có thể tái sử dụng.
        + Một ứng dụng React có thể sử dụng hàng chục, hoặc hàng trăm các component liên kết và tương tác với nhau.
        + Ta có thể tạo ra một component bằng cách extends từ Component như sau:

Hình 1.2 Component trong ReactJS.

* + - * + Phương thức quan trọng nhất là render(), phương thức này được trigger khi component đã sẵn sàng để được render lên trên trang. Trong hàm này, sẽ trả về mô tả cho việc muốn React render cái gì lên trên trang. Hàm render chính là mô tả cụ thể của giao diện người dùng tại bất cứ thời điểm nào. Vì thế nếu dữ liệu thay đổi, React sẽ quan tâm đến việc cập nhật lại giao diện với dữ liệu tương ứng.

Props:

* + - * + Thông tin được truyền từ component này sang component khác được gọi là props.
        + Props của một component có thể là một object, function, string, number

…. Nó chứa thông tin về thành phần đó.

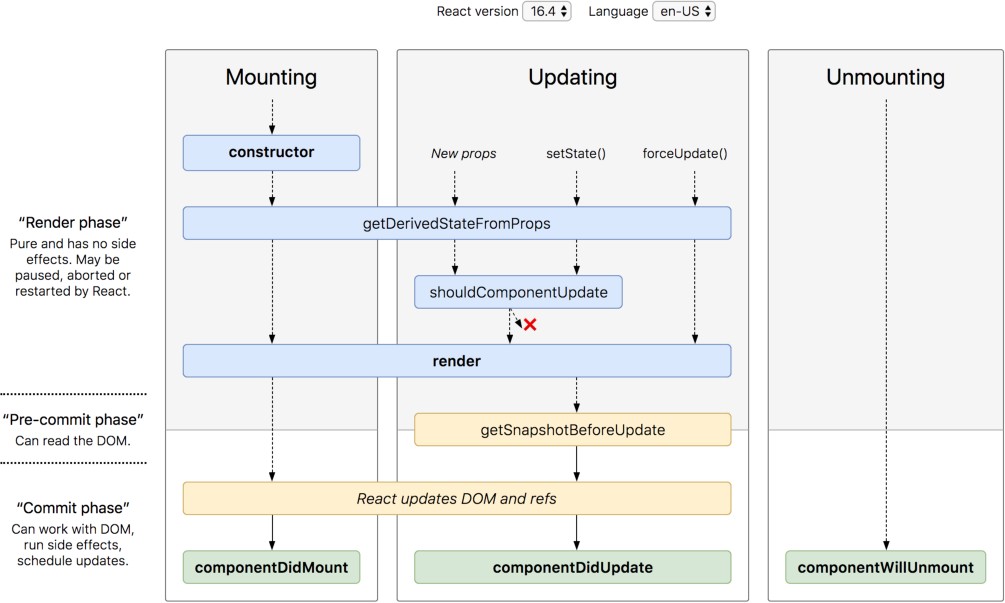
State:

* + - * + Một component trong React có 2 cách để lấy thông tin props và state. Không như props, state của 1 component không được truyền từ bên ngoài vào. Một component sẽ tự quyết định state của chính nó. Để tạo ra state cho component, ta sẽ phải khai báo nó trong hàm constructor.



Hình 1.3 Sử dụng state trong Component.

* + - * + This.state ở đây là một đối tượng. Để truy cập đến các state ta sử dụng cú pháp: this.state.name\_of\_property. Và cách phổ biến để thay đổi state của một component là dùng hàm setState().

Lyfecycle:

Hình 1.4 Sơ đồ vòng đời của component trong ReactJS

* + - * + Một thành phần đặc trưng không kém Props và State của React đó là LifeCycle. LifeCycle là những phương thức được viết sẵn truy cập đến

từng thời điểm trong một vòng đời của component, từ đó có thể triển khai ý tưởng một cách dễ dàng mà linh động.

* + - * + **contructor(props):** Hàm này là hàm khởi tạo, là hàm chạy ngay sau khi component được tạo ra, đầu vào của hàm này là props (tức là thuộc tính của component được dùng để truyền dữ liệu từ bên ngoài vào). Trong hàm này cũng có thể khởi tạo state cho component.
        + **Render:** Là hàm bắt buộc, vì đây là đầu ra của component. Nó chịu trách nhiệm trả về một đối tượng JSX những gì muốn hiển thị ra bên ngoài.
        + **componentDidMount:**Là hàm được gọi ngay sau khi giao diện được render xong, có nghĩa là khi giao diện hiện lên sẽ gọi hàm này. Trong hàm này có thể dùng để lấy dữ liệu từ api, server… và setState dữ liệu ra.
        + **componentWillUnmount:**Hàm này được gọi khi component loại bỏ DOM, thực hiện các thao tác dọn dẹp, loại bỏ những phần tử dư thừa và hủy các timer…
        + **componentDidUpdate(prevProps, prevState):**Hàm này được gọi ngay sau khi component được render lại.
      1. ***Ưu điểm và nhược điểm của ReactJS***

Ưu điểm:

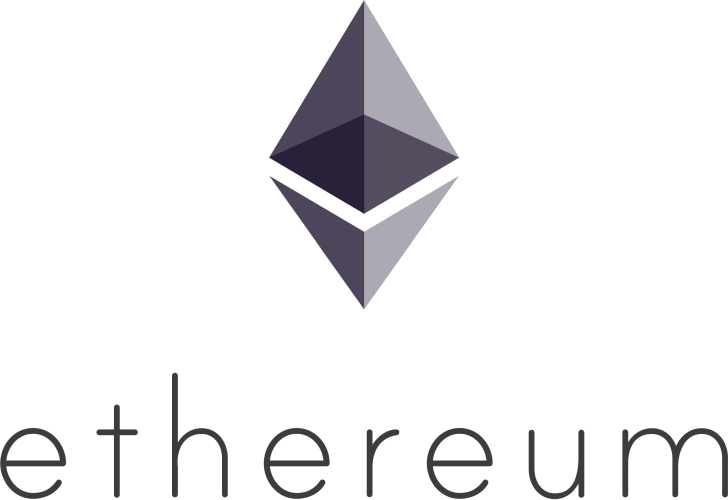
* + - * + ReactJs cực kì hiệu quả: Reactjs tạo ra cho chính nó DOM ảo – nơi mà các component thực sự tồn tại trên đó. Điều này sẽ giúp cải thiện hiệu suất rất nhiều. Reactjs cũng tính toán những thay đổi nào cần cập nhật len DOM và chỉ thực hiện chúng. Điều này giúp Reactjs tránh những thao tác cần trên DOM mà nhiều chi phí.
        + ReactJs giúp việc viết các đoạn code JS dễ dàng hơn: Nó dùng cú pháp đặc biệt là JSX (Javascript mở rộng) cho phép ta trộn giữa code HTML và Javascript. Ta có thể thêm vào các đoạn HTML vào trong hàm render mà không cần phải nối chuỗi. Đây là đặc tính thú vị của Reactjs. Nó sẽ chuyển đổi các đoạn HTML thành các hàm khởi tạo đối tượng HTML bằng bộ biến đổi JSX.
        + Reactjs có nhiều công cụ phát triển: Khi bắt đầu Reactjs, đừng quên cài đặt ứng dụng mở rộng của Chrome dành cho Reactjs. Nó giúp sửa lỗi chương trình dễ dàng hơn. Sau khi cài đặt ứng dụng này, người dùng sẽ có cái nhìn trực tiếp vào virtual DOM như thể đang xem cây DOM thông thường.
        + Render tầng server: Một trong những vấn đề với các ứng dụng đơn trang là tối ưu SEO và thời gian tải trang. Nếu tất cả việc xây dựng và hiển thị trang đều thực hiện ở client, thì người dùng sẽ phải chờ cho trang được khởi tạo và hiển thị lên. Điều này thực tế là chậm. Hoặc nếu giả sử người dung vô hiệu hóa Javascript thì sao? Reactjs là một thư viện component, nó có thể vừa render ở ngoài trình duyệt sử dụng DOM và cũng có thể render bằng các chuỗi HTML mà server trả về.
        + Làm việc với vấn đề test giao diện: Nó cực kì dễ để viết các test case giao diện vì virtual DOM được cài đặt hoàn toàn bằng JS.
        + Hiệu năng cao đối với các ứng dụng có dữ liệu thay đổi liên tục, dễ dàng cho bảo trì và sửa lỗi.

Nhược điểm:

* + - * + Reactjs chỉ phục vụ cho tầng View. React chỉ là View Library nó không phải là một MVC framework như những framework khác. Đây chỉ là thư viện của Facebook giúp render ra phần view. Vì thế React sẽ không có phần Model và Controller, mà phải kết hợp với các thư viện khác. React cũng sẽ không có 2-way binding hay là Ajax.
        + Tích hợp Reactjs vào các framework MVC truyền thống yêu cầu cần phải cấu hình lại.
        + React khá nặng nếu so với các framework khác, React có kích thước tương tương với Angular (Khoảng 35kb so với 39kb của Angular). Trong khi đó Angular là một framework hoàn chỉnh.

### Ethereum

* + - 1. **Giới thiệu**

****

Hình 1.5 Ethereum

Ethereum (ETH) là một nền tảng chạy trên nền tảng công nghệ Blockchain. Được phát triển bởi Vitalik Buterin vào cuối năm 2013 và nó được thêm vào chức năng smart contract.

ETH hay cũng có thể gọi là Ether là đơn vị tiền tệ trong hệ thống Blockchain của Ethereum. Chính cái này là thứ mà chúng ta có thể giao dịch mua bán trên các sàn giao dịch hiện tại.

* + - 1. **Cách đọc giao dịch của Ethereum**
      * **TxHash –** Còn được gọi là ID giao dịch, TxHash là cách để tìm kiếm một giao dịch cụ thể blockchain Ethereum.
      * **Block height –** Là số thứ tự của khối mà giao dịch được thêm vào. Tức là thợ đào sẽ xử lý giao dịch của bạn khi họ đào Ethereum. Nói cách khác số đó biểu thị giao dịch đó được diễn ra ở block số mấy trong chuỗi blockchain.
      * **TimeStamp –** Thời điểm xử lý giao dịch.
      * **Value -** Số lượng tiền ETH trong chuyến giao dịch đó.
      * **Gas limit –** Số Gas tối đa mà bạn sẵn sàng trả khi thực hiện giao dịch.
      * **Gas Used by Txn –** Số Gas thực dùng.
      * **Gas price –** Giá của Gas, giá này được quyết định bởi thợ đào.
      * **Actual Tx Cost/Fee –** Phí bạn phải trả, phí này được tính bằng Gas Price \* Gas Used.
      * **Nonce –** Số lần mà địa chỉ đó đã giao dịch. Ví dụ bạn chuyển lui tiền 100 lần thì con số đó sẽ là 100.
      1. **Một số khái niệm cần nắm của Ethereum**

1. **Ví Ethereum**

Cũng giống như Bitcoin, ví Ethereum là loại ví dùng để lưu trữ Ethereum nhằm phục vụ cho việc giao dịch, mua bán, trao đổi và cất trữ đồng tiền này.

1. **Địa chỉ Ethereum**

Địa chỉ Ethereum cũng là khái niệm tương tự với địa chỉ Bitcoin, là dãy chữ số để người khác có thể chuyển ETH vào ví của bạn. Khi cần chuyển ETH đi, bạn cần biết địa chỉ ví Ethereum của người đó.

1. **Token Ethereum – ERC-20**

Token Ethereum được phát hành trên nền tảng tiền điện tử để phục vụ cho một mục đích cụ thể.

ERC-20 định nghĩa một danh sách chung các quy tắc cho các mã thông báo Ethereum để tuân theo hệ sinh thái Ethereum lớn hơn, cho phép các nhà phát triển dự đoán chính xác sự tương tác giữa các chuỗi. Các quy tắc này bao gồm cách các mã thông báo được chuyển giữa các địa chỉ và cách truy cập dữ liệu trong mỗi mà thông báo.

### Smart contract

* + - 1. **Giới thiệu**

Smart contract (hợp đồng thông minh) là một giao thức quản lý hợp đồng. Nó là một bộ giao thức đặc biệt với mục tiêu là để đóng góp, xác nhận hay tiến hành quá trình đàm phán và thực hiện hợp đồng. Smart contract cho phép chúng ta triển khai giao dịch không cần thông qua một bên thứ ba trung gian. Những giao dịch này hoàn toàn dễ dàng truy dấu và không thể bị can thiệp hoặc đảo chiều. Smart contract chứa trong mình toàn bộ những thông tin chi tiết về các điều khoản và thực hiện chúng một cách tự động.

* + - 1. **Lợi ích của Smart contract**

Smart Contract là một ứng dụng tận dụng tất cả những điểm mạnh của công nghệ Blockchain mang lại vì vậy nó có rất nhiều lợi ích, dưới đây là các lợi ích chính của nó.

1. **Tự động hóa**: Quá trình được thực hiện hợp đồng là tự động bằng. Đồng thời bạn chính là người tạo hợp đồng, không còn phải phụ thuộc vào môi giới, luật sư hay bất kì ai khác. Như vậy, nó cũng xóa bỏ những nguy cơ đến từ bên thứ ba.
2. **Không bị thất lạc**: Tài liệu của bạn được mã hóa trên một cuốn sổ cái chung, có nghĩa là không thể bị thất lạc. Với Blockchain, tất cả những người bạn đều có lưu trữ lại tài liệu của bạn.
3. **An toàn**: Blockchain sẽ đảm bản sự an toàn cho tài liệu của bạn. Không một hacker nào có thể đe dọa đến chúng.
4. **Tốc độ**: Hợp đồng thông minh sử dụng các ngôn ngữ lập trình, code phần mềm để tự động hóa các điều khoản, tiết kiệm hàng tiếng đồng hồ cho những công việc không cần thiết.
5. **Tiết kiệm**: Hợp đồng thông minh tiết kiệm cho bạn hàng đống tiền nhờ xóa bỏ khâu trung gian.
6. **Chính xác**: Các hợp đồng tự động không chỉ nhanh và rẻ hơn mà còn tránh được các lỗi thường thấy khi viết giấy tờ.
   * + 1. **Cần những gì để tạo nên một Smart contract**
   1. **Chủ thể hợp đồng:** Smart Contract phải được cấp khả năng truy cập đến sản phẩm/dịch vụ liệt kê trong hợp đồng để có thể tự động khóa hay mở khóa chúng.
   2. **Chữ kí điện tử:** Tất cả các bên tham gia vào Smart Contract đều phải đồng ý triển khai thỏa thuận bằng các khóa cá nhân (chữ kí điện tử) của họ.
   3. **Điều khoản hợp đồng:** Điều khoản trong Smart Contract có dạng là một chuỗi các hoạt động. Các bên tham gia hợp đồng đều phải ký chấp nhận.
   4. **Nền tảng phân quyền:** Smart Contract sau khi hoàn tất sẽ được tải lên Blockchain của nền tảng phân quyền tương ứng và được phân phối về cho các node của nền tảng ấy.

### Tổng quan về Solidity

* + - 1. **Giới thiệu**



Hình 1.6 Solidity

Solidity là một ngôn ngữ cấp cao, hướng đối tượng để thực hiện các hợp đồng thông minh. Hợp đồng thông minh là các chương trình chi phối hành vi của các tài khoản trong trạng thái Ethereum.

Solidity chịu ảnh hưởng của C ++, Python và JavaScript và được thiết kế để nhắm mục tiêu Máy ảo Ethereum (EVM).

Với Solidity, bạn có thể tạo hợp đồng cho các mục đích sử dụng như bỏ phiếu, gây quỹ cộng đồng, đấu giá mù và ví đa chữ ký.

* + - 1. **Kiểu dữ liệu**
  1. ***Mảng***

Biến struct cho phép chúng ta xác định nhiều cấu trúc data hơn. Solidity cũng có các mảng. Bạn có thể tạo các mảng cố định hoặc linh hoạt tùy ý. Mảng bằng struct là linh hoạt vì không xác định chiều dài. Vì thế chúng ta có thể cộng thêm các đối tượng vào vô số mà không có giới hạn.

* 1. ***Phiên bản Pragma***

Tất cả các mã nguồn đều bắt đầu với một phiên bản “version pragma“-một tuyên bố của phiên bản của trình biên dịch Solidity. Đây là để ngăn chặn các vấn đề với các phiên bản trình biên dịch tương lai có khả năng giới thiệu những thay đổi có thể phá vỡ code của người dùng.

* 1. ***Địa chỉ - Address***

Các blockchain Ethereum được tạo thành từ các tài khoản, ví dụ tài khoản ngân hàng. Một tài khoản có một số dư của Ether (đơn vị tiền tệ được sử dụng trên Ethereum blockchain), bạn có thể gửi và nhận thanh toán Ether cho các tài khoản khác, giống như tài khoản ngân hàng của bạn có thể chuyển khoản tiền vào các tài khoản ngân hàng khác. Và mỗi tài khoản sẽ có địa chỉ. An FCM implementation includes two main components for sending and receiving:

Ví dụ: Ethereum blockchain được tạo nên từ các địa chỉ. Mỗi tài khoản có một address riêng. Nó có dạng dưới đây:

0x0cE440255306E921F41612C46F1v6df9Cc969183.

Mỗi địa chỉ có một lượng Ether nhất định, là số tiền ảo được dùng trên blockchain, và có thể cho và nhận Ether từ các địa chỉ khác.

* 1. ***Mappings***

Mappings là một cách khác để lưu trữ dữ liệu có tổ chức trong Solidity. Mappings là một kho lưu trữ giá trị chính để lưu trữ và tìm kiếm dữ liệu.

* 1. ***Memory và Storage***

Trong Solidity, có 2 chỗ lưu biến: trong storage hoặc trong memory. Một biến lưu trên memory chỉ là tạm thời, nó còn khi các hàm được dùng đến, sau đó sẽ bị xoá. Một biến lưu trên storage thì ở đó vĩnh viễn. Bạn cũng không cần phải lo lắng về việc biến được lưu ở đâu, vì Solidity sẽ xử lý nó giúp bạn.

Tuy nhiên, trong một số trường hợp, bạn muốn cần chi tiết chỗ lưu một biến nhất định. Solidity sẽ hiển thị cho bạn bằng từ khóa memory và storage.

* + 1. **KeystoneJS**

**A picture containing icon

Description automatically generated**

Hình 1.7 KeystoneJS

* + - 1. **Giới thiệu:**

KeystoneJS là một framework mã nguồn mở dành cho việc phát triển các database-driven websites, ứng dụng và APIs bằng Nodejs

Keystone sẽ cấu hình Express - máy chủ web thực tế Nodejs cho bạn và kết nối với cơ sở dữ liệu MongoDB của bạn bằng Mongoose (Mongoose là thư viện mô hình hóa dữ liệu đối tượng (object data modeling – ODM) cung cấp môi trường mô hình hóa nghiêm ngặt cho dữ liệu của bạn, thực thi cấu trúc khi cần mà vẫn duy trì tính linh hoạt giúp MongoDB trở nên mạnh mẽ)

* + - 1. **Tính năng:**
* **Dynamic Routes**: Keystone khởi tạo với những thành phần tốt nhất để thiết lập ứng dụng MV \* cho bạn và giúp dễ dàng quản lý các Templates, Views và Routes.
* **Database Fields:** IDs, Strings, Booleans, Dates và Numbers là các khối xây dựng cơ sở dữ liệu của bạn. Keystone xây dựng trên những thứ này với các loại trường thực tế, hữu ích như name, email, password, address, image và các trường quan hệ (và hơn thế nữa)
* **Admin UI được tạo tự động:** Dù trong lúc đang xây dựng ứng dụng hoặc khi ứng dụng được vận hành như một hệ thống quản lý nội dung cơ sở dữ liệu, UI của Keystone sẽ giúp bạn tiết kiệm thời gian và giúp quản lý dữ liệu của bạn dễ dàng.
* **Code đơn giản hơn:** Đôi khi, có thể code phức tạp để làm những việc đơn giản. Keystone giữ cho mọi thứ đơn giản – như tải dữ liệu trước khi hiển thị nó trong chế độ View.
* **Xử lý Form:** Bạn muốn xác thực một Form, tải lên một hình ảnh và cập nhật cơ sở dữ liệu của bạn với một dòng duy nhất? Keystone có thể làm điều đó, dựa trên các mô hình dữ liệu bạn đã xác định.
* **Quản lý Session:** Keystone cung cấp sẵn các tính năng xác thực và quản lý Session, bao gồm tự động mã hóa cho các trường mật khẩu.
  + - 1. **Yêu cầu cần có:**
* Hãy chắc chắn rằng bạn đã cài đặt Node.js JavaScript runtime. Keystone 4 đã được thử nghiệm nhiều nhất với Node 6, nhưng phải tương thích với các bản phát hành sản xuất mới hơn. Chúng tôi khuyên bạn nên sử dụng các bản phát hành Node LTS (Long Term Support) cho vòng đời hỗ trợ mở rộng và tính ổn định của chúng.
* Bạn cũng sẽ cần truy cập vào máy chủ cơ sở dữ liệu MongoDB được cài đặt cục bộ hoặc được lưu trữ từ xa. Keystone 4 phải tương thích với các bản phát hành sản xuất hiện tại của MongoDB.
* Bạn sẽ cần kiến thức làm việc với JavaScript, Node.js và npm để sử dụng Keystone. Làm quen cơ bản với các khái niệm cơ sở dữ liệu và MongoDB cũng sẽ hữu ích.

Hiện tại dự án đang sử dụng Keystone v5 và bản v6 đang trong quá trình phát triển, chưa tối ưu và ổn định để xử dụng.

* + 1. **GraphQL**

**A picture containing logo

Description automatically generated**

*Hình 1.8 GraphQL*

* + - 1. **Giới thiệu:**

GraphQL là ngôn ngữ thao tác và truy vấn dữ liệu nguồn mở cho API, cung cấp cho client 1 cách thức dễ dàng để request chính xác những gì họ cần, giúp việc phát triển API dễ dàng hơn theo thời gian. GraphQL được Facebook phát triển nội bộ vào năm 2012 trước khi phát hành công khai vào năm 2015.

GraphQL bao gồm 3 điểm đặc trưng bao gồm:

* Cho phép client xác định chính xác những dữ liệu gì họ cần
* GraphQL làm cho việc tổng hợp dữ liệu từ nhiều nguồn dễ dàng hơn
* Sử dụng một type system để khai báo dữ liệu.
  + - 1. **Đặc điểm của GraphQL**
  1. **Hiệu quả, mạnh mẽ & tiện lợi**

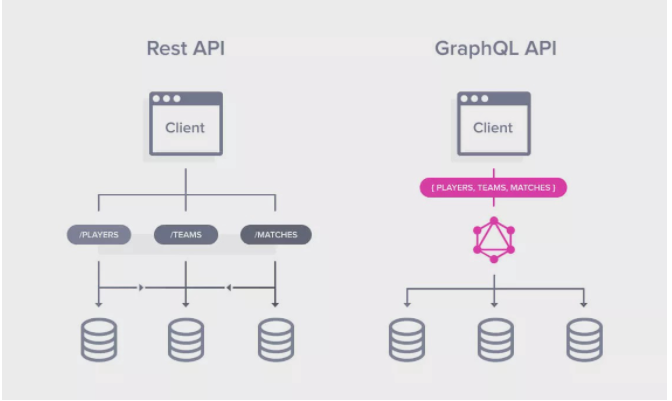
Để bạn dễ hình dung về điểm mạnh của GraphQL, chúng ta cần nói đến REST (Representational State Transfer) — loại giao thức API phổ biến mà GraphQL muốn hướng đến thay thế.

Trước khi có GraphQL, các lập trình viên thường sử dụng REST để truy vấn dữ liệu. Tuy nhiên, vấn đề của REST là dữ liệu của nó thường sẽ trả về quá nhiều, dư thừa so với thông tin được yêu cầu; có khi lại quá ít. Dù là trường hợp nào thì hiệu suất của ứng dụng cũng sẽ bị ảnh hưởng rất nhiều.

Với GraphQL thì bạn yêu cầu thông tin nào từ API thì chỉ sẽ ra đúng thứ bạn cần, không hơn không kém.

* 1. **Tính năng Defining Schema (Xác định Schema) và Type System (Kiểu Hệ thống)**
     + GraphQL có một hệ thống riêng dành cho nó được sử dụng để xác định schema của một api. Tất cả type được liệt kê trong một API thì được viết trong schema thì sử dụng GraphQL Schema Definition Language(SDL) (SDL là cách ngắn gọn nhất để kê khai thông tin trong một GraphQL schema. Thực chất, SDL là một phần trong đặc tả GraphQL chính thức).
     + Bạn có thể hiểu schema này như là bản quy định giữa 2 phía client và server để xác định cách client truy cập dữ liệu từ client như thế nào.
  2. **Fetching Data**

Cách GraphQL nạp dữ liệu khác với REST



Hình 1.9 Khác biệt giữa Rest và GraphQL

Trong REST thì mỗi nguồn thông tin là 1 endpoint, với mỗi endpoint bạn phải thực hiện 1 lệnh gọi dữ liệu khác nhau, và dữ liệu mỗi endpoint trả về sẽ có thể nhiều hơn hoặc ít hơn dữ liệu bạn yêu cầu. Trong khi đó, với GraphQL, bạn chỉ cần thực hiện lệnh gọi đến 1 endpoint và hệ thống sẽ làm hết phần còn lại, trả về thông tin của 3 nguồn cho bạn.

### Phần kết luận

Bằng cách nghiên cứu và tìm hiểu về các công nghệ trên, em đã áp dụng thành công các khái niệm và cơ chế hoạt động của chúng trong đồ án lần này để tạo ra một ứng dụng khá đầy đủ và chuyên nghiệp..

Một vài công nghệ có thể không mới, nhưng chúng được sử dụng rộng rãi và là xu hướng cho ngành công nghệ phát triển phần mềm. Do đó, hiểu khái niệm là rất quan trọng, nó giúp em có thể áp dụng đúng công nghệ cho từng dự án, để nâng cao hiệu quả và khả năng sử dụng

# Chương 2. PHÂN TÍCH VÀ THIẾT KẾ HỆ THỐNG

Chương này sẽ đi vào chi tiết của các yêu cầu, mô tả các yêu cầu phi chức năng, các ràng buộc thiết kế và các yếu tố cần thiết khác để cung cấp một bản mô tả đầy đủ và hoàn thiện về các yêu cầu cho ứng dụng. Nó sẽ bao gồm sơ đồ ca sử dụng, sơ đồ tuần tự, sơ đồ lớp và cơ sở dữ liệu. Từ đó sẽ hiển thị được tổng quan về những chức năng mà ứng dụng có thể đáp ứng. Ngoài ra, nó còn xác định kiến trúc, module và dữ liệu cho một hệ thống để đáp ứng các yêu cầu đã nêu ra.

Quá trình thiết kế hệ thống là cung cấp đủ dữ liệu, thông tin chi tiết về hệ thống, và là một yếu tố của hệ thống để cho phép thực hiện các hành vi phù hợp với các thể kiến trúc cũng như được định nghĩa trong các mô hình, các module và quan điểm của kiến trúc hệ thống. Nó cho thấy các thành phần của ứng dụng, cấu trúc của các bảng dữ liệu, mối quan hệ của các yếu tố tạo nên hệ thống.

**2.1. Các chức nặng chính**

**2.1.1. Xem thông tin thị trường tiền điện tử**

Ở tính nặng này, người dùng có thể dễ dàng tìm thấy thông tin biến động, chênh lệch giá của các đồng coin đang lưu hành trên một số sàn CEX lớn hiện nay như Binance,, Huobi, Gate.io và các sàn DEX như Uniswap, Sushiswap, PancakeSwap. Từ đó giúp người dùng có thể nhanh chóng cập nhật thông tin và quyết định xem mình nên sắp xếp chiến thuật đầu tư sử dụng khoản tiền vừa vay sao cho hợp lý.

**2.1.2. Tạo ví điện tử lưu trữ Ethereum**

Chỉ cần một thao tác đơn giản là người dùng đã có thể tạo ngay cho mình một ví điện tử lưu trữ các đồng tiền điện tử. Sau khi tạo ví thành công ứng dụng sẽ sinh ra một mã khóa bí mật cá nhân và 12 cụm từ ghi nhớ để khôi phục sau này mỗi khi quên mật khẩu. Người dùng bắt buộc phải nhớ (lưu lại ở một nơi an toàn) để có thể nhập ví này vào những ứng dụng khác. Trường hợp người dùng quên thì sẽ mất ví này vĩnh viễn, không có cách nào để khôi phục.

Để khôi phục ví, ta sẽ sử dụng metamask (đây là một extension của Google giúp quản lý phiên đăng nhập và tương tác ví với các thành phần khác trên mạng lưới blockchain) và sử dụng 12 cụm từ ghi nhớ để khôi phục ví của bạn.

**2.1.3. Tạo, thêm tổng cung cho token riêng của bạn**

Thật tuyệt vời khi có thể tự tạo cho mình một mã token riêng của bạn trên ứng dụng này. Chỉ cần cung cấp thông tin về tên, kí hiệu, sau khi tạo thành công, bạn đã tự sở hữu cho mình 1 token riêng cực kỳ đơn giản.

Tổng cung được hiểu đơn giản là khoản tiền được cung ứng cho thị trường, tức lượng token có thể lưu thông mua bán.

Để thêm tổng cung cho token của bạn (hay còn gọi là mint token) thì có thể tạo ra một số lượng token hữu hạn tùy ý mà ví có thể lưu trữ.

**2.1.4. Vay nhanh với tùy chỉnh tham số**

Chức nặng chính của đồ án này là khả năng vay một token với số lượng lớn và dùng nó cho một chuỗi hoạt động nào đó trên blockchain để đem lại lợi nhuận và toàn bộ quá trình lúc vay đến lúc trả chỉ thao tác trên một giao dịch duy nhất nên hoàn toàn an toàn và minh bạch cho khoản vay.

***Các bước tiến hành vay:***

B1: Chọn token cần vay và số lượng cần vay token đó

B2: Thêm các hoạt động giao dịch dựa trên kiến thức về thị trường tiền điện tử. Bước này cần xác định cũng một loại token thì giá ở sàn nào cao hơn, sàn nào thấp hơn để đưa ra chiến lược phù hợp nhất.

+ Chọn token có biến động giá cao

+ Chọn sàn có hỗ trợ token đó

B3: Vay và thanh toán khoản phí khởi tạo giao thức. Nếu chiến lược đúng đắn thì ứng dụng sau khi vay và hoàn thành các hoat động của bạn, nó sẽ trả lại phần token đã vay( kèm phí vay). Phần lợi nhuận còn lại sẽ tự động chuyển về ví của bạn.

**2.1.5. Lưu lịch sử vay nhanh và thống kê**

Toàn bộ quá trình vay chỉ diễn ra trong một giao dịch duy nhất nên nếu chiến lược đầu tư thất bại thì giao dịch đó sẽ bị thất bại và không hiển thị bất cứ thông tin gì trên các ứng dụng quét giao dịch như etherscan hay bscscan.

Và vì dữ liệu chỉ hiển thị khi vay và thao tác các hoạt động thành công, đem về khoản lợi nhuận nên thật khó khăn khi theo dõi cũng như phân tích và thông kế dữ liệu.

Chức năng lưu lịch sử nhằm mục đích lưu lại lịch sử các giao dịch và giữ lại số liệu cần thiết để thống kê một cách tổng quan về quá trình tham gia vay của người dùng.

**2.1.6. Quản lý đăng nhập để truy xuất thông tin lịch sử giao dịch**

Để cá nhân hóa dữ liệu trên ứng dụng và phục vụ cho các dịch vụ phát triển thêm sau này, việc đăng nhập để lưu dữ liệu về tên, hình ảnh, hay thậm chí là các token NFT từ ứng dụng sẽ giúp trãi nghiệm người dùng trở nên thân thiện và dễ sử dụng hơn.

Thao tác đăng nhập chỉ đơn giản là kết nới với ví điện tử, ứng dụng sẽ hiện thị đúng với thông tin của tài khoản ví điện tử đó.

**2.1.7. Mời bạn bè tham gia trải nghiệm ứng dụng**

Nhằm tăng tính tương tác của ứng dụng thì chức năng mời bạn bè sử dụng mã mời cá nhân là một cách mà ứng dụng thu hút thêm người dùng. Mời càng nhiều bạn bè sử dụng thì ứng dụng sẽ thống kê và trong tương lai sẽ tặng phàn quà như token hay món quà NFT phiên bản giới hạn nào đó.

**2.2. Sơ đồ ca sử dụng**

**2.2.1. Tổng thể sơ đồ**

Ứng dụng sẽ có các tính năng như sau:

* Xem thông tin thị trường tiền điện tử
* Tạo ví điện tử lưu trữ Ethereum
* Tạo, thêm tổng cung cho token riêng của bạn
* Vay nhanh với tùy chỉnh tham số
* Lưu lịch sử vay nhanh và thống kê
* Quản lý đăng nhập để truy xuất thông tin lịch sử giao dịch
* Mời bạn bè tham gia trải nghiệm ứng dụng

Diagram

Description automatically generated

Hình 2.1 Sơ đồ tổng thể hệ thống

**2.2.2. Xem thông tin thị trường tiền điện tử**

Diagram

Description automatically generated

Hình 2.2 Sơ đồ ca sử dụng xem thông tin thị trường tiền điện tử

Ở chức năng này, ứng dụng sẽ dựa trên dữ liệu cung cấp từ các sàn giao dịch nổi tiếng và tín cậy của giới tiền điện tử là TradingView và Binance để cập nhật thông tin của các đồng tiền điện tử.

**2.2.3. Tạo ví điện tử lưu trữ Ethereum**

Diagram

Description automatically generated

Hình 2.3 Sơ đồ ca sử dụng tạo ví điện tử ethereum

Ở phần này, ứng dụng có hỗ trợ tạo tài khoản ví điển tử ngay trên ứng dụng và sau khi tạo thành công bạn sẽ nhận được 3 thông tin cơ bản sau:

* Khóa công khai (public key): địa chỉ ví của bạn
* 12 cụm từ ghi nhớ để khôi phục ví điện tử của bạn
* Khóa bí mật (private key): Khóa cho phép truy cập toàn bộ chức năng của ví điện tử mà không cần các phản hồi yêu cầu xác thực giao dịch

**2.2.4. Tạo, thêm tổng cung cho token riêng của bạn**

Diagram

Description automatically generated

Hình 2.4 Sơ đồ ca sử dụng tương tác với token

Ở mục này người dùng có thể tạo riêng cho mình một token đơn giản và thêm tổng cung cho token đó. Tất cả thông tin sẽ được hiển thị khá cần thiết về một token chuẩn ERC-20. Ứng dụng chỉ cho phép người khởi tạo token mới có quyền thêm tổng cung cho token đó.

**2.2.5. Vay nhanh với tùy chỉnh tham số và lưu lịch sử giao dịch**

Diagram

Description automatically generated

Hình 2.5 Sơ đồ ca sử dụng chức năng vay flash loan

Ở phần này, người dùng sẽ trải nghiệm khả năng vay nhanh một khoản coin hay token và dùng nó tạo ra lợi nhuận nhanh chóng chỉ trong một giao dịch duy nhất.

Lịch sử giao dịch được lưu đầy đủ, thống kê chi tiết và tính toán khoản lợi nhuận trung bình, lời nhuận cao nhất thấp nhất trong toàn bộ quá trình vay trên ứng dụng.

**2.3. Sơ đồ tuần tự**

Dưới đây là sơ đồ tuần tự của những chức năng chính:

**2.3.1. Xem thông tin thị trường tiền điện tử**

A picture containing diagram

Description automatically generated

Hình 2.6 Sơ đồ tuần tự xem thông tin thị trường tiền điện tử

**2.3.2. Tạo ví điện tử lưu trữ Ethereum**

Diagram

Description automatically generated

Hình 2.7 Sơ đồ tuần tự chức năng tạo ví điện tử ethereum

**2.3.3. Tạo, thêm tổng cung cho token riêng của bạn**

Diagram

Description automatically generated

Hình 2.8 Sơ đồ vòng tuần tự tạo token

Diagram

Description automatically generated

Hình 2.9 Sơ đồ tuần tự thêm tổng cung cho token

**2.3.4. Vay nhanh với tùy chỉnh tham số và lưu lịch sử giao dịch**

Diagram

Description automatically generated

Hình 2.10 Sơ đồ tuần tự chức năng vay flash loan

**2.4. Sơ đồ lớp**

Graphical user interface, application, Teams

Description automatically generated

Hình 2.11 Sơ đồ lớp hệ thống

**2.5. Cơ sở dữ liệu**

Do ứng dụng triển khai trên blockchain nên dữ liệu đều được lưu trên các khối dữ liệu phi tập trung và ứng dụng này sử dụng cơ sử dữ liệu để lưu và quản lý thông tin lịch sử giao dịch. Ngoài ra, còn lưu trữ thêm một số thông tin cần thiết cho trãi nghiệm người dùng như về hình ảnh, thông tin cá nhân cơ bản.

Cơ sở dữ liệu gồm:

* + - * + Bảng Customer

|  |  |  |  |
| --- | --- | --- | --- |
| **No.** | **Name** | **Type** | **Allow Null** |
| 1 | \_id | int | FALSE |
| 2 | address | varchar | FALSE |
| 3 | nickname | varchar | TRUE |
| 4 | referralCode | varchar | FALSE |
| 5 | referrals | int | TRUE |
| 6 | history | int | TRUE |
| 7 | avatar | varchar | TRUE |

Bảng 2. 1 Bảng cơ sở dữ liệu Customer

* + - * + Bảng History

|  |  |  |  |
| --- | --- | --- | --- |
| **No.** | **Name** | **Type** | **Allow Null** |
| 1 | \_id | int | FALSE |
| 2 | name | varchar | FALSE |
| 3 | sender | varchar | FALSE |
| 4 | history | varchar | FALSE |
| 5 | profit | varchar | FALSE |
| 6 | txhash | varchar | FALSE |

Bảng 2. 2 Bảng cơ sở dữ liệu History

* + - * + Bảng Referral

|  |  |  |  |
| --- | --- | --- | --- |
| **No.** | **Name** | **Type** | **Allow Null** |
| 1 | \_id | int | FALSE |
| 2 | addressFrom | varchar | FALSE |
| 3 | addressTo | varchar | FALSE |
| 4 | referralCode | varchar | FALSE |
| 5 | referralAt | datetime | FALSE |

Bảng 2. 3 Bảng cơ sở dữ liệu Referral

* + - * + Bảng Image

|  |  |  |  |
| --- | --- | --- | --- |
| **No.** | **Name** | **Type** | **Allow Null** |
| 1 | \_id | int | FALSE |
| 2 | filename | varchar | FALSE |
| 3 | originalFilename | varchar | FALSE |
| 4 | publicUrl | varchar | FALSE |
| 5 | createDate | datetime | FALSE |

Bảng 2. 4 Bảng cơ sở dữ liệu Image

Graphical user interface, application

Description automatically generated

Hình 2.11 Sơ đồ cơ sở dữ liệu hệ thống

# Chương 3: TRIỂN KHAI VÀ CHẠY THỬ NGHIỆM

## Môi trường triển khai

* + - **Smart contract**

Ở phần này em đã dùng solidity để thực hiện việc viết nên các contract. Để tiện hơn cho việc sử dụng và kiểm thử chương trình thì em có tận dụng Remix để kiểm tra code trực tuyến. Các contract sẽ được triển khai trực tiếp lên sàn giao dịch của Ethereum (cụ thể hơn đó là mạng Kovan).

* + - **Ứng dụng web**

Em đã sử dụng ReactJS để viết nên ứng dụng này.

* + - **Các công cụ phát triển**

Trong quá trình thực hiện em đã dùng những công cụ sau: Visual Studio Code, KeystoneJS, Mongo DB, Metamask, GridFS và Git.

## Triển khai hệ thống

**Các giai đoạn triển khai:**

* + - Giai đoạn 1: Lựa chọn mạng triển khai thử nghiệm:

Chọn Kovan vì:

|  |  |  |  |
| --- | --- | --- | --- |
| Mạng | Id mạng | Cơ chế đồng thuận | Thời gian của khối |
| Ropsten | 3 | POW | sub-30 seconds |
| Rinkeby | 4 | POA | 15 seconds |
| Kovan | 42 | POA | 4 seconds |

Bảng 3. 1 So sánh các mạng thử nghiệm trên Ethereum

Tốc độ của mạng Kovan khá nhanh so với các mạng khác của Ethereum

* + - Giai đoạn 2: Lựa chọn giao thức vay:

Có các nền tảng vay như AAVE, Compound, Dydx, Uniswap, … Nhưng AAVE là nền tảng cho vay flash loan đầu tiên kèm với ưu điểm về chi phí vay 0.09% mỗi lần (ở testnet 0.35%), còn Uniswap 0.3%

* + - Giai đoạn 3: Triển khai smart contract vay flash loan và các thao tác giao dịch trao đổi token để tạo nên sự chênh lệch giá.

Cụ thể là tương tác hoán đổi tiền điện tử (swap token) với các sàn phi tập trung.

Để hoàn thành được giai đoạn này, kiến thức cần tìm hiểu đó là cách trao đổi (swap) coin ở các sàn phi tập trung, cụ thể trong đồ án này là uniswap V1, uniswap V2 và sushiswap V2. Bao gồm các mức kiến thức sau:

+ Triển khai cơ chế vay flash loan từ giao thức AAVE protocol tại đây: [www.github.com/aave/aave-protocol/blob/master/contracts/lendingpool/LendingPool.sol](http://www.github.com/aave/aave-protocol/blob/master/contracts/lendingpool/LendingPool.sol). Điểm đáng chú ý là hàm flashloan và executeOperation

Text

Description automatically generated

Hình 3.1 Cấu trúc hàm vay flash loan của giao thức

+ Swap bằng Uniswap V1 sử dụng hàm createExchange(), tokenToTokenSwapInput():

Text

Description automatically generated

Hình 3.2 Cấu trúc hàm swap token ở Uniswap V1

| **Parameter** | **Type** | **Description** |
| --- | --- | --- |
| tokens\_sold | uint256 | Số lượng token ERC20 đầu vào đã bán |
| min\_tokens\_bought | uint256 | Lượng tối thiếu token ERC20 đã mua |
| min\_eth\_bought | uint256 | ETH tối thiểu mua dưới dạng trung gian |
| deadline | uint256 | Thời hạn giao dịch |
| token\_addr | address | Địa chỉ token ERC20 đầu ra |

Bảng 3. 2 Tham số truyền của hàm tokenToTokenSwapInput

+ Swap bằng Uniswap V2 ủy thác đến một địa chỉ contract và gọi đến interface UniswapV2Router02 sử dụng các hàm swapExactTokenForTokens()

Text

Description automatically generated

Hình 3.3 Cấu trúc hàm swap token của Uniswap V2

| **Name** | **Type** |  |
| --- | --- | --- |
| amountIn | uint | Số lượng mã thông báo đầu vào để gửi |
| amountOutMin | uint | Số lượng mã thông báo đầu ra tối thiểu phải nhận được để giao dịch không hoàn nguyên |
| Path | address[] calldata | Một mảng địa chỉ mã thông báo path.length >= 2, Các bể chứa mỗi cặp địa chỉ liên tiếp phải tồn tại và có tính thanh khoản |
| To | address | Người nhận mã thông báo đầu ra |
| deadline | uint | Thời gian mà fsau đó giao dịch sẽ hoàn nguyên |
|  |  |  |
| amounts | uint[] memory | Só lượng mã thông báo đầu vào và tất cả số lượng mã thông báo đầu ra tiếp theo |

Bảng 3. 3 Tham số truyền của hàm swapExactTokenForTokens

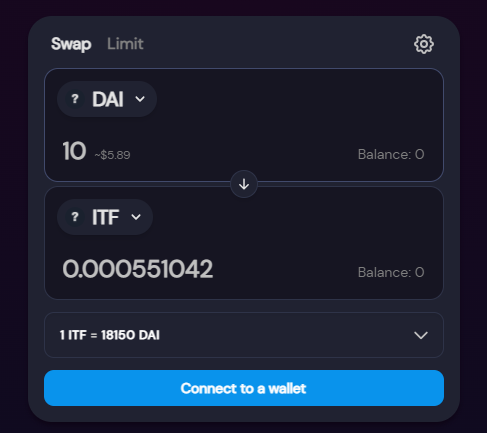
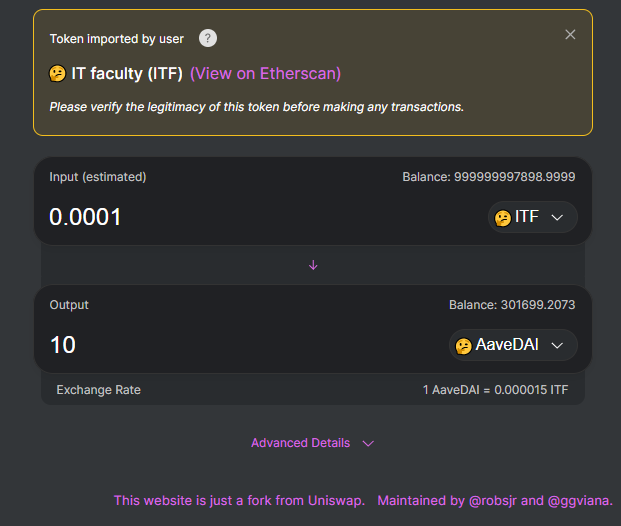
* + - Giai đoạn 4: Thiết lập môi trường thử nghiệm ứng dụng bằng việc tạo token, thiết lập giá ở các sàn theo từng cặp token hoặc neo với một phần giá của Ethereum.

Để thiết lập được môi trường thử nghiệm chức năng và trải nghiệm ứng dụng thì cần các yêu cầu sau:

+ Truy cập trang faucets.chain.link để lấy nhiều đồng ETH nhất có thể và sử dụng nó triển khai dự án.

+ Tạo các token và thêm tổng cung lưu trữ trong ví holder

+ Thêm các token vào các sàn và tạo sự khác biệt về giá để mục hiện flash loan thuận lợi.



Hình 3.4 Cặp giao dịch ITF-DAI đã được thêm ở uniswap va sushiswap

* + - Giai đoạn 5: Xây dựng giao diện tương tác thân thiện và deploy thành sản phẩm.

**3.2.2 Phân tích thuật toán**

Điểm đặc biệt của vay flash loan là toàn bộ quá trình vay và mua bán cho đến khi hoàn trả, tất cả chỉ diễn ra trong một giao dịch (transaction) duy nhất. Diều này làm nên sự khác biệt lớn.

**Bản chất của vay flash loan là một hình thức cho vay nhưng phải trả ngay sau khi giao dịch kết thúc.** Cụ thể:

Diagram

Description automatically generated

Hình 3.5 Sơ đồ hoạt động của ứng dụng flash loan

Bước 1: My Smart Contract yêu cầu vay một lượng 5 token USDT đến giao thức

Bước 2: Giao thức yêu cầu bể thanh khoản cho vay

Bước 3: Bể thanh khoản chuyển 5 USDT cho giao thức xử lý

Bước 4: Giao thức chuyển 5 USDT về My Smart Contract xử lý

Tại đây 5 USDT được dùng để thao tác 2 hành động:

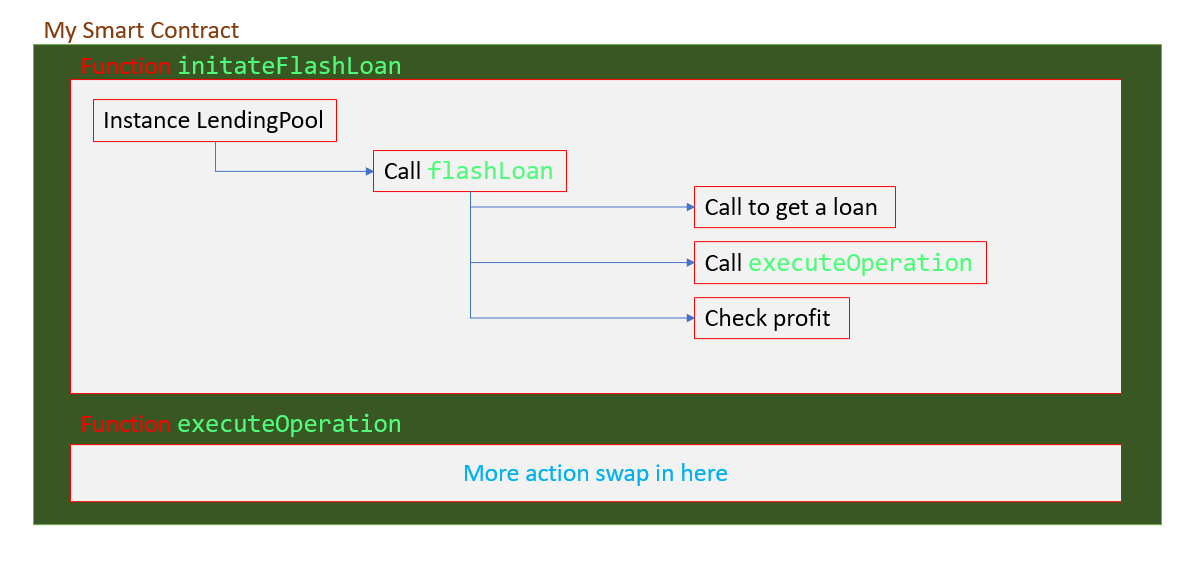
1 là Mua token A ở sàn Ex1 giá 5 USDT

2 là Bán token A ở sàn Ex2 giá 6 USDT

Sau khi kiểm tra có lợi nhuận thì làm bước 5, không thì dừng chương trình và hoàn nguyên lại giao dịch.

Bước 5: Hoàn trả 5 USDT đã vay + 0.015 USDT phí khởi tạo giao thức vay

Bước 6: Giao thức hoàn trả khoản vay về bể thanh khoản

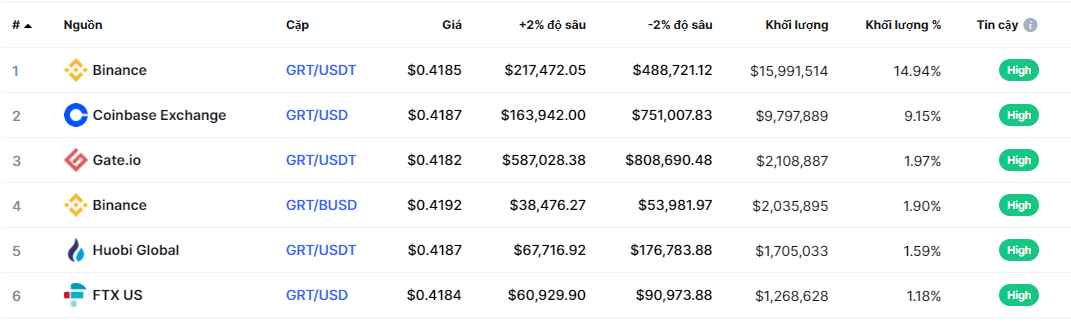


Hình 3.6 Sơ đồ luồng hoạt động của code

Ở hình trên, My Smart Contract khởi tạo một biểu hiện instance của LendingPool và gọi hàm flashloan của nó. Bên trong hàm này, chứa một lời gọi hàm đến một hàm thực thi các ACTION flash loan ở My Smart Contract tự định nghĩa là executeOperation. Đến khi kết thúc lời gọi hàm, là lúc đã thao tác các ACTION xong, lúc này LendingPool có thể kiểm tra kết quả giao dịch có đem lại lợi nhuận hay không? Và điều đó đảm bảo tính toàn vẹn của khoản vay.

**Mục đích khi sử dụng flash loan là tạo ra thu nhập bằng việc ăn chênh lệch giá. Vậy chênh lệch giá do đâu?**

Giá của một đồng coin bất kỳ trên các sàn tập trung hay phi tập trung tại một thời điểm có biến động khác nhau. Cụ thể là:



Hình 3.7 Giá chênh lệch của GRT token trên các sàn tập trung

Riêng đối với các sàn phi tập trung thì giá của một đồng coin phụ thuộc vào giao thức ORACLE mà sàn sử dụng để lấy dữ liệu giá. ORACLE ở đây chính là giao thức giúp chuyển dữ liệu off-chain thành các dữ liệu on-chain.

Như đã biết về mạng blockchain thì muốn xác thực dữ liệu thì phải xác thực với nhiều node trong mạng. Điều đó làm chậm quá trình cập nhật dữ liệu giá giữa các sàn giao dịch dẫn đến độ trễ thông tin gây ra chênh lệch giá.

Sử dụng khác giao thức ORACLE cũng dẫn đến chênh lệch vì thuật toán khác nhau.

Ngoài ra, vấn đề hiện nay vẫn còn gặp trở ngại rất lớn, đó là cơ sở hạ tầng mạng thấp, tốc độ truyền tải dữ liệu không thể ngay lập tức được mà nó luôn có độ trễ nhất định.

Những vấn đề trên dẫn đến giá tại một thời điểm ở trên các sàn sẽ khác nhau.

**Câu hỏi đặt ra liệu Flash loan ăn chênh lệch giá bằng cách nào ?**

Đó là mua ở sàn có giá thấp và bán ở sàn có giá cao. Đặc trưng của giao thức vay flash loan là thực hiện giao dịch nhanh và toàn bộ chỉ diễn ra trong một giao dịch duy nhất. Đảm bảo tính chớp nhoáng, tức thời của khoản vay và giao dịch chuyển đổi trên các sàn.

## Kết quả demo và hướng dẫn sử dụng chương trình

* + 1. **Xem thông tin thị trường tiền điện tử**

Graphical user interface, chart, line chart

Description automatically generated

Hình 3.8 Giao diện xem thông tin thị trường tiền điện tử

Ở màn hình này, phần chart hiển thị biểu đồ của loại tiền điện tử người dùng đang quan tâm. Ngoài ra có thể tìm và xem giá của các loại tiền điện tử khác theo từng khung giờ và loại biểu đồ tùy chỉnh khác.

Đến biểu đồ khung thời gian từng phút thì dữ liệu sẽ được cập nhật realtime giá của từng loại.

* + 1. **Tạo ví điện tử lưu trữ Ethereum**

Graphical user interface

Description automatically generated with medium confidence

Hình 3.9 Giao diện tạo ví điện tử ethereum

Chỉ cần có nhu cầu tạo ví thì ứng dụng sẽ ngay lập tức tạo một ví điện tử với 3 tham số cơ bản:

* Địa chỉ ví (public key)
* Cụm từ khôi phục ví 12 tử
* Khóa bí mật nhằm cho phép tương tác trực tiếp với các giao dịch mà không cần yêu cầu xác thực thanh toán trước mỗi giao dịch.
  + 1. **Tạo, thêm tổng cung cho token riêng của bạn**

Graphical user interface, text, application

Description automatically generated

Hình 3.10 Giao diện mint token

Ở phần này, người dùng có thể tạo một token với tham số tên và kí hiệu là đã tạo ngay một token với chủ sở hữu là ví vừa khởi tạo thao tác tạo đó.

Để sử dụng được token đó thì phải thêm bước mint (tạo tổng cung) cho token để lưu hành.

Sau đó, cho phép nó lưu hành trên sàn và xác thực nó là một token đúng chuẩn ERC-20.

* + 1. **Vay nhanh với tùy chỉnh tham số**

B1: Lựa chọn loại coin muốn vay

Graphical user interface, text, application, chat or text message

Description automatically generated

Hình 3.11 Giao diện khởi tạo khoản vay flash loan

Graphical user interface, text, application

Description automatically generated

Hình 3.12 Giao diện chọn loại coin muốn vay

B2: Nhập số lượng coin muốn vay và click nút Add để thêm các hoạt động sau khi vay thành công

Graphical user interface, application

Description automatically generated

Hình 3.13 Giao diện chọn số lượng coin muốn vay

Graphical user interface, application

Description automatically generated

Hình 3.14 Giao diện thêm và tùy chỉnh các hoán đổi coin

B3: Lựa chọn các sàn, các loại coin muốn trao đổi (swap) giữa các sàn đó với nhau sao cho có thể tạo ra được lợi nhuận tốt nhất có thể dựa trên chiến lược của người dùng.

Graphical user interface, text, application, chat or text message

Description automatically generated

Hình 3.15 Giao diện lựa chọn sàn giao dịch

Graphical user interface, application

Description automatically generated

Hình 3.16 Giao diện xác nhận các thiết lập

B4: Xác thực giao dịch và chờ đợi kết quả

Graphical user interface, text, application, chat or text message

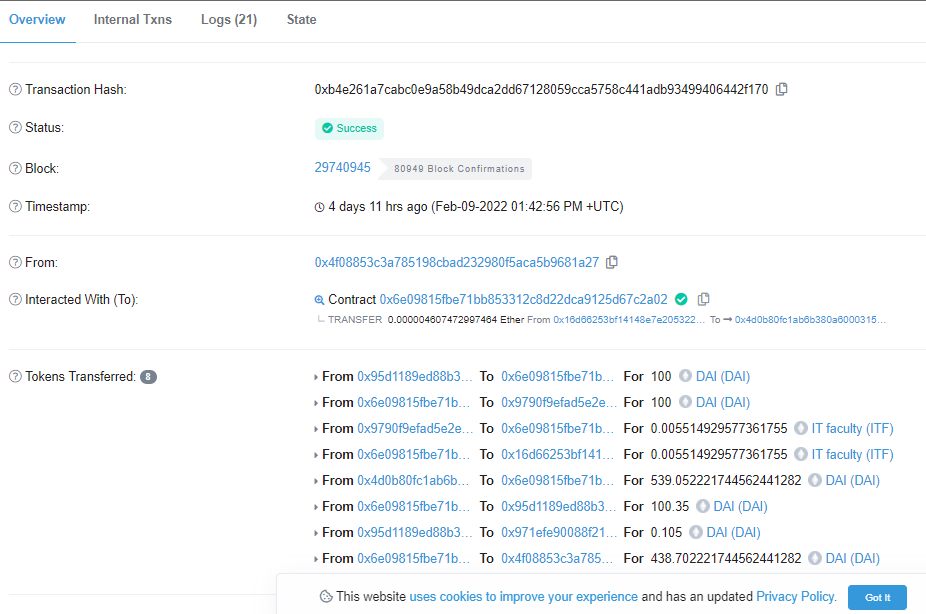
Description automatically generated

Hình 3.17 Giao diện chờ kết quả

Graphical user interface, application

Description automatically generated

Hình 3.18 Xác thực giao dịch trên metamask



Hình 3.19 Kết quả giao dịch thành công trên kovan.etheescan.io

Graphical user interface, text, application, email

Description automatically generated

Hình 3.20 Kết quả giao dịch thất bại trên kovan.etheescan.io

* + 1. **Lưu lịch sử vay nhanh và thống kê**

Graphical user interface, application

Description automatically generated

Hình 3.21 Giao diện lịch sử giao dịch

Graphical user interface, application

Description automatically generated

Hình 3.22 Giao diện thống kê dữ liệu lịch sử các giao dịch

Lịch sử giao dịch sẽ được lưu lại trên một cơ sở dữ liệu tập trung, cụ thể trong ứng dụng này là mongoDB cloud và quản lý dữ liệu đó bằng thư viện KeystoneJS sử dụng graphQL để truy vấn và các tương tác liên quan.

Graphical user interface, application

Description automatically generated

Hình 3.23 Giao diện lịch sử của giao dịch thành công

Graphical user interface, application

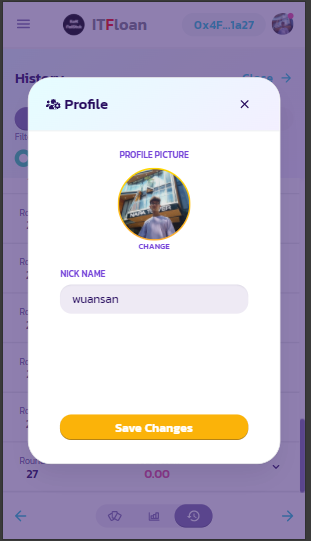
Description automatically generated

Hình 3.24 Giao diện lịch sử giao dịch thất bại

Bên cạnh đó quản lý thông tin này còn dễ dàng truy xuất lịch sử giao dịch được lưu trên cơ sở dữ liệu tập trung.

Trong trường hợp này, hệ thống không có khả năng lưu lịch sử trên cơ sở dữ liệu phi tập trung vì trong trường hợp nếu giao dịch vay flash loan thất bại thì tất cả dữ liệu của nó không được lưu trên blockchain.

* + 1. **Quản lý đăng nhập để truy xuất thông tin lịch sử giao dịch**



Hình 3.25 Giao diện thông tin cá nhân

Với mục tiêu tạo sự tương tác về mặt hình ảnh, nickname thay vì sử dụng địa chỉ ví.

* + 1. **Mời bạn bè tham gia trải nghiệm ứng dụng**

Graphical user interface, text, application

Description automatically generated

Hình 3.26 Giao diện mời thêm bạn bè

Graphical user interface, application, Teams

Description automatically generated

Hình 3.27 Giao diện tổng hợp kết quả mời bạn bè

Ứng dụng có chức năng mời bạn bè cũng tham gia sử dụng và trải nghiệm ứng dụng flash loan này và số lượng mời càng lớn sẽ có khả năng nhận được một lượng token hay một phần thưởng nào đó giá trị.

Với mục đích thu hút người dùng càng lớn đến trải nghiệm và đánh giá các tính năng cũng như hiệu suất của hệ thống.

* + 1. **Đánh giá**

- Ưu điểm

* Về cơ bản, ứng dụng đã có thể thực hiện vay một khoản tiền điện tử đơn giản và nhanh chóng.
* Tạo ra được một môi trường khiến cho các nhà giao dịch tiền điện tử có thể theo dõi mọi thông tin về giá ở từng thời điểm mà họ mong muốn. Ngoài ra, còn có một trang liên tục cập nhật các thông tin và thắc mắc cơ bản về ứng dụng flash loan
* Minh bạch các khoản vay vì chỉ diễn ra trong một giao dịch duy nhất nên khả năng sử dụng tiền đó chuyển sang một ví điện tử là không thể.
* Tận dụng hạn chế của cơ sở hạ tầng blockchain để ăn chênh lêch giá giữa các sàn giao dịch tiền điện tử
* Giao dịch được xử lý nhanh hơn và tiết kiệm hơn so với các thao tác trao đổi (swap) coin hoạt động riêng lẻ.

- Nhược điểm

* Muốn tạo ra được lợi nhuận phải có chiến lược và tìm hiểu nghiên cứu rất kĩ giá của các tiền điện tử, cũng như tốc độ mạng truyền tải cực nhanh.
* Trên blockchain hiện tại vẫn còn rất nhiều giao thức chưa đủ khả năng chống lại các cuộc tấn công và thao túng sử dụng Flash loan nên tiềm ẩn rất nhiều rủi ro trên thị trường.
* Ứng dụng chỉ được xây dựng mục đích thử nghiệm và nghiên cứu, chi phí xây dựng trên mainnet khá đắt đỏ nên chỉ triển khai được trên mạng thử nghiệm.

# KẾT LUẬN

* + - 1. **Thành tựu**

Sau khi xây dựng hế thống này, có thể hiểu cách xây dựng một ứng dụng web tương tác với mạng blockchain. Từ quá trình xây dựng giao diện, xử lý dữ liệu cho đến triển khai hệ thống thành một sản phẩm đóng gói trên một hosting và liên kết nó đến một tên miền tùy chỉnh.

Em cũng đã cải thiện được kỹ năng của em rất nhiều, bao gồm kỹ năng nghiên cứu, kỹ năng kỹ thuật, tiếng Anh, thuyết trình cũng như khá nhiều kỹ nằng mềm cần có khác..

Về dự án, em đã thực hiện theo đúng kế hoạch được tạo ra trong tuần đầu tiên của thời gian làm đồ án tốt nghiệp và thực hiện tất cả các chức năng trong kế hoạch đó. Em cũng cố gắng làm cho ứng dụng trở nên linh hoạt nhất và gần gũi nhất nhằm mục đích nâng cao trải nghiệm người dùng.

* + - 1. **Hướng phát triển**

Với một số nhược điểm em đã đề cập trong phần đánh giá, công việc trong tương lai của dự án này sẽ như sau:

* Cố gắng hoàn thiện và triển khai nó trên mạng chính thức của Ethereum
* Mở rộng thêm các hướng dẫn chi tiết hơn, giải thích cụ thể hơn về ứng dụng và khả năng thực sự của nó.
* Thêm các dịch vụ và chức năng thú vị để thu hút sự tương tác của người dùng.
* Xa hơn, ứng dụng có thể triển khai không chỉ là mạng Ethereum mà có thể là các mạng khác như Near, Polkadot, Avalanche, …

# TÀI LIỆU THAM KHẢO

[1] <https://www.v5.keystonejs.com/> , truy cập vào 01/03/2022

[2] <https://www.graphql.org/> , truy cập vào 01/03/2022

[3] https://www.restapitutorial.com/ , truy cập vào 01/03/2022

[4] https://www.reactjs.org/, truy cập vào 01/03/2022

[5] [https://github.com/ethereum/EIPs/blob/master/EIPS/eip-20.md/](https://github.com/ethereum/EIPs/blob/master/EIPS/eip-20.md) , truy cập vào 14/02/2022

[6] <https://solidity.readthedocs.io/en/v0.5.13/>, truy cập vào 01/03/2022

[7] https://iztuts.com/bai-1-gioi-thieu-keystonejs/, truy cập vào 14/02/2022

[8] <https://piyopiyo.medium.com/how-to-generate-ethereum-wallet-with-ethers-js-d0ef74eadfd8>, truy cập vào 01/03/2022

[9] https:// coin568.com/poa-la-gi/, truy cập vào 01/03/2022

[10] https://www.youtube.com/watch?v=vcvPeqhrbzk, truy cập vào 01/03/2022