**Hồ Việt Phú TÊN ĐỀ TÀI HỆ THỐNG HỖ TRỢ CHĂM SÓC SỨC KHỎE…………………………………………………………. 2022**

**ĐẠI HỌC ĐÀ NẴNG**

**TRƯỜNG ĐẠI HỌC BÁCH KHOA**

**KHOA CÔNG NGHỆ THÔNG TIN**

**ĐỒ ÁN TỐT NGHIỆP**

**NGÀNH: CÔNG NGHỆ THÔNG TIN**

**CHUYÊN NGÀNH: TRUYỀN THÔNG VÀ MẠNG MÁY TÍNH**

**ĐỀ TÀI:**

**HỆ THỐNG HỖ TRỢ CHĂM SÓC SỨC KHỎE**

Người hướng dẫn: **TS. PHẠM MINH TUẤN**

Sinh viên: **HỒ VIỆT PHÚ**

Số thẻ sinh viên: **102180135**

Lớp**: 18TCLC-DT1**

**Đà Nẵng, 12/2022**

**ĐẠI HỌC ĐÀ NẴNG**

**TRƯỜNG ĐẠI HỌC BÁCH KHOA**

**KHOA CÔNG NGHỆ THÔNG TIN**

**ĐỒ ÁN TỐT NGHIỆP**

**NGÀNH: CÔNG NGHỆ THÔNG TIN**

**CHUYÊN NGÀNH: TRUYỀN THÔNG VÀ MẠNG MÁY TÍNH**

**ĐỀ TÀI:**

**HỆ THỐNG HỖ TRỢ CHĂM SÓC SỨC KHỎE**

Người hướng dẫn: **TS. PHẠM MINH TUẤN**

Sinh viên: **HỒ VIỆT PHÚ**

Số thẻ sinh viên: **102180135**

Lớp**: 18TCLC-DT1**

**Đà Nẵng, 12/2022**

**TÓM TẮT**

Tên đề tài: Hệ Thống Hỗ Trợ Tư Vấn Sức Khỏe

Sinh viên thực hiện: Hồ Việt Phú

Số thẻ SV: 102180135 Lớp: 18TCLC\_DT1

Y tế và chăm sóc sức khỏe là một trong những lĩnh vực cực kỳ quan trọng nhất đối với con người, và được chú trọng đầu tư và nghiên cứ hàng trăm nay năm. Cứ mỗi năm trôi qua thì có nhiều người chết đi. Mỗi ngày trôi qua dân số thế giới ngày càng tăng và xã hội ngày càng phát triển do đó nhu cầu về y tế và chăm sóc sức khỏe ngày càng được chú trọng và đầu tư mạnh mẽ, nhưng theo nhiều báo cáo các tổ chức y tế hiện nay chỉ ra rằng trên thế giới có hơn nữa dân số là chưa tiếp cận được với các dịch vụ y tế cơ bản vì có nhiều nguyên nhân như: chi phí thăm khám, cơ sở y tế, thiết bị không đạt tiêu chuẩn … Do đó tôi muốn thực hiện dự án này để có thể kết nối con người với các nền y tế và chắm sóc sức khỏe một cách dễ dàng hơn

|  |  |
| --- | --- |
| ĐẠI HỌC ĐÀ NẴNG  **TRƯỜNG ĐẠI HỌC BÁCH KHOA**  KHOA CÔNG NGHỆ THÔNG TIN | **CỘNG HÒA XÃ HÔI CHỦ NGHĨA VIỆT NAM**  Độc lập - Tự do - Hạnh phúc |

**NHIỆM VỤ ĐỒ ÁN TỐT NGHIỆP**

Họ tên sinh viên: Hồ Việt Phú Số thẻ sinh viên: 102180135

Lớp: 18TCLCDT1 Khoa: Công Nghệ Thông Tin

Ngành: Công Nghệ Thông Tin Chất Lượng Cao

1. *Tên đề tài đồ án:*Hệ Thống Hỗ Trợ Tư Vấn Sức Khỏe

*Đề tài thuộc diện:*  *Có ký kết thỏa thuận sở hữu trí tuệ đối với kết quả thực hiện*

1. *Các số liệu và dữ liệu ban đầu:* Thu thập từ nhiều nguồn
2. *Nội dung các phần thuyết minh và tính toán:*

*- Gồm có 7 phần*

* *Cơ sở lý thuyết*
* *Phân tích hệ thống và thiết kế*
* *Blockchain, smart contract, tiền điện tử và những ứng dụng của chúng*
* *Hệ sinh thái ethereum và IPFS*
* *WebRTC và SocketIO*
* *Học máy và mô hình dự đoán bệnh*
* *Kết luận*

1. *Các bản vẽ, đồ thị ( ghi rõ các loại và kích thước bản vẽ ):*
2. *Họ tên người hướng dẫn:* T.S Phạm Minh Tuấn, Khoa Công Nghệ Thông Tin Trường Đại Học Bách Khoa – Đại Học Đà Nẵng.
3. *Ngày giao nhiệm vụ đồ án: ……../……./2022*
4. *Ngày hoàn thành đồ án: ……../……./2022.*

|  |  |
| --- | --- |
|  | *Đà Nẵng, ngày tháng năm 201* |
| **Trưởng Bộ môn** …………………….. | **Người hướng dẫn** |

**LỜI NÓI ĐẦU**

Trong quá trình thực hiện đề tài, tôi xin chân thành cảm ơn T.S Phạm Minh Tuấn , cảm ơn những ý tưởng, ý kiến cũng như những giải pháp và các kiến thức để hoàn thành được đồ án này.

Và tôi cũng xin chân thành cảm ơn các thầy cô giáo và các bạn sinh viên của khoa Công Nghệ Thông Tin – Đại Học Đà Nẵng đã giúp đỡ tôi trong nhũng năm học qua, cho tôi những kiến thức và kinh nghiệm quý báu để tôi có thể hoàn thành được đồ án này

Và cuối cùng tôi xin gửi lời cảm ơn đặc biệt tới gia đình tôi, những người đã ủng hộ, tạo động lực và giúp đỡ tôi về mặt vật chất và tinh thần cho tôi

Mặc dù em đã cố gắng hết sức để thực hiện đồ án này nhưng không thể không tránh khỏi những sai sót hay không hoàn chỉnh. Em hy vọng có thể nhận được ý kiến đóng góp từ các thầy cô để hoành tốt đồ án này

Em xin cảm ơn !

**CAM ĐOAN**

Tôi đã hiểu rõ chính sách của trường về chống đạo văn và tôi đảm đảm bảo rằng:

1. Nội dung của luận văn đồ án này do bản thân tôi thực hiện dưới sự hướng dẫn của T.S Phạm Minh Tuấn.
2. Tất cả tài liệu tham khảo trong luận văn đồ án này mà tôi sử dụng đều được trích dẫn địa chỉ và tên của tác giả.
3. Nội dung của đồ án này là do bản thân tôi tự tìm hiểu và thực hiện, không sao chép ở bất kỳ các nguồn khác hoặc những dự án đã có giải thưởng đã công bố

Sinh viên thực hiện

Hồ Việt Phú

**MỤC LỤC**

TÓM TẮT…………………………………………………………………………….i

NHIỆM VỤ ĐỒ ÁN………………………………………………………………….ii

LỜI NÓI ĐẦU VÀ CẢM ƠN…………………………………………………..........iii

LỜI CAM KẾT LIÊM CHÍNH HỌC THUẬT…………………………………........iv

MỤC LỤC………………………………………………………………………........v

DANH SÁCH CÁC HÌNH VẼ……………………………………………………….vi

DANH CÁC BẢNG ………………………………………………………………….vii

DANH SÁCH CÁC CỤM TỪ VIẾT TẮT…………………………………………...viii

[CHƯƠNG 1. CƠ SỞ LÝ THUYẾT 3](#_Toc122009699)

[1.1. Giới thiệu 4](#_Toc122009700)

[1.2. ReactJs 4](#_Toc122009701)

[1.2.1. Giới thiệu về Reactjs 4](#_Toc122009702)

[1.2.2. ReactJs hoạt động như thế nào 4](#_Toc122009703)

[1.2.3. Các tính năng nổi bật của ReactJs 5](#_Toc122009704)

[1.3. NodeJS 6](#_Toc122009705)

[1.3.1. NodeJS là gì 6](#_Toc122009706)

[1.3.2. Cách hoạt động của NodeJS 6](#_Toc122009707)

[1.3.3. Đặc điểm của Nodejs 7](#_Toc122009708)

[1.4. Firebase 8](#_Toc122009709)

[1.4.1. Firebase là gì 8](#_Toc122009710)

[1.4.2. Những tính năng chính Firebase 8](#_Toc122009711)

[1.4.2.1. Firebase RealTime Database 8](#_Toc122009712)

[1.4.2.2. Firebase Cloud Storage 9](#_Toc122009713)

[1.4.2.3. Firebase Storage 10](#_Toc122009714)

[CHƯƠNG 2. PHÂN TÍCH HỆ THỐNG VÀ THIẾT KẾ 11](#_Toc122009715)

[2.1. Phân tích yêu cầu hệ thống 11](#_Toc122009716)

[2.1.1. Phân tích yêu cầu chức năng 11](#_Toc122009717)

[2.1.1.1. Yêu cầu về chức năng 11](#_Toc122009718)

[2.1.1.2. Yêu cầu lưu trữ 11](#_Toc122009719)

[2.1.1.3. Phân tích yêu cầu chức năng 12](#_Toc122009720)

[2.1.1.4. Mô số use case của hệ thống 12](#_Toc122009721)

[2.2. Phân tích thiết kế hệ thống 18](#_Toc122009722)

[2.2.1. Thiết kế cơ sở dữ liệu 18](#_Toc122009723)

[2.2.1.1. Lược đồ cơ sở dữ liệu 18](#_Toc122009724)

[2.2.1.2. Mô tả các bảng dữ liệu 19](#_Toc122009725)

[CHƯƠNG 3. BLOCKCHAIN, SMART CONTRACT, TIỀN ĐIỆN TỬ VÀ NHỮNG ỨNG DỤNG CỦA CHÚNG 25](#_Toc122009726)

[3.1. Giới thiệu 25](#_Toc122009727)

[3.2. Blockchain 25](#_Toc122009728)

[3.2.1. Blockchain là gì 25](#_Toc122009729)

[3.2.2. Kiến trúc của blockchain 27](#_Toc122009730)

[3.2.3. Cách thức hoạt động blockchain 28](#_Toc122009731)

[3.3. Các loại Blockchain 29](#_Toc122009732)

[3.4. Cơ chết mã hóa Blockchain 30](#_Toc122009733)

[3.4.1. Cryptography 30](#_Toc122009734)

[3.5. Hàm băm và tầm quan trọng 32](#_Toc122009735)

[3.6. Smart Contract 33](#_Toc122009736)

[3.6.1. Smart Contract là gì 33](#_Toc122009737)

[3.7. Tiền điện tử 34](#_Toc122009738)

[3.7.1. Tiền điện tử là gì 34](#_Toc122009739)

[3.7.2. Ưu điểm tiền điện tử 34](#_Toc122009740)

[CHƯƠNG 4. HỆ SINH THÁI ETHEREUM VÀ IFPS 36](#_Toc122009741)

[4.1. Giới thiệu 36](#_Toc122009742)

[4.2. Hệ sinh thái Ethereum 36](#_Toc122009743)

[4.2.1. Ethereum là gì 36](#_Toc122009744)

[4.2.2. Metamask là gì 36](#_Toc122009745)

[4.2.3. Gas là gì 37](#_Toc122009746)

[4.2.4. Solidity là gì 38](#_Toc122009747)

[4.2.5. IPFS 39](#_Toc122009748)

[4.2.5.1. 39](#_Toc122009749)

[b) Cách hoạt động của IFPS 39](#_Toc122009750)

[4.3. INFURA 40](#_Toc122009751)

[4.3.1. Infura là gì 40](#_Toc122009752)

[4.4. Cấu trúc của một ứng dụng Web 3 đươc sử dụng trong luận văn 41](#_Toc122009753)

[4.5. Các tính năng trong hệ thống lưu trữ trong blockchain: 42](#_Toc122009754)

[4.5.1. Yêu cầu lưu trữ 42](#_Toc122009755)

[4.5.2. Thiết kế contract lưu dữ liệu 42](#_Toc122009756)

[CHƯƠNG 5. WebRTC VÀ SocketIO 44](#_Toc122009757)

[5.1. Giới thiêu 44](#_Toc122009758)

[5.2. WebRTC 44](#_Toc122009759)

[5.2.1. Giới thiêu 44](#_Toc122009760)

[5.2.2. WebRTC là gì 44](#_Toc122009761)

[5.2.3. Cấu trúc và cách thức hoạt động của WebRTC 44](#_Toc122009762)

[5.3. SocketIO 46](#_Toc122009763)

[5.3.1. SocketIO là gì 46](#_Toc122009764)

[5.4. Mô tả sự kết hợp giữa WebRTC và SocketIO để tạo videocall 46](#_Toc122009765)

[CHƯƠNG 6. HỌC MÁY, CÁC MÔ HÌNH DỰ ĐOÁN BỆNH 48](#_Toc122009766)

[6.1. Giới thiệu 48](#_Toc122009767)

[6.2. Học máy 48](#_Toc122009768)

[6.2.1. Học máy là gì 48](#_Toc122009769)

[6.2.2. Học có giám sát 48](#_Toc122009770)

[6.2.3. Thuật toán Naïve Bayes 49](#_Toc122009771)

[6.2.3.1. Định lý Naïve Bayes 49](#_Toc122009772)

[6.2.3.2. Thuật toán Naïve Bayes là gì 50](#_Toc122009773)

[6.3. Mô tả dữ liệu và kết quả dự đoán 51](#_Toc122009776)

[CHƯƠNG 7. KẾT LUẬN 52](#_Toc122009777)

[7.1. Kết quả đạt được 52](#_Toc122009778)

[7.2. Những vấn đề chưa giải quyết 52](#_Toc122009779)

[7.3. Hướng phát triển 52](#_Toc122009780)

[7.4. Khó khăn và thuận lợi 53](#_Toc122009781)

[7.5. Một số hình ánh của hệ thống 53](#_Toc122009782)

[53](#_Toc122009783)

**DANH SÁCH HÌNH VẼ**

[Hình 1. 1 Logo ReactJS 4](file:///C:\Users\Vietphu\Desktop\DOC_DATN.docx#_Toc122010127)

[Hình 1. 2 Logo NodeJS 6](file:///C:\Users\Vietphu\Desktop\DOC_DATN.docx#_Toc122010128)

[Hình 1. 3 Logo Firebase 9](#_Toc122010129)

[Hình 2. 1 Use case bệnh nhân 13](#_Toc122010136)

[Hình 2. 2 Use case bác sĩ 15](#_Toc122010137)

[Hình 2. 3 Use case admin 16](#_Toc122010138)

[Hình 2. 4 Lược đồ cơ sở dữ liệu 18](#_Toc122010139)

[Hình 3. 1 Minh họa các chuỗi 25](#_Toc122010173)

[Hình 3. 2 Cấu trúc của một block 26](#_Toc122010174)

[Hình 3. 3 Kiến trúc của chuỗi 28](#_Toc122010175)

[Hình 3. 4 Chuỗi liên kết dữ liệu 28](#_Toc122010176)

[Hình 3. 5 Cách hoạt động của blockchain 29](#_Toc122010177)

[Hình 3. 6 Các loại blockchain 29](#_Toc122010178)

[Hình 3. 7 Các loại cryptography 31](#_Toc122010179)

[Hình 3. 8 Hashing 32](#_Toc122010180)

[Hình 3. 9 Smart Contract 34](#_Toc122010181)

[Hình 3. 10 Metamask 37](file:///C:\Users\Vietphu\Desktop\DOC_DATN.docx#_Toc122010182)

[Hình 3. 11 Solidity 38](#_Toc122010183)

[Hình 3. 12 Mô tả IPFS 39](#_Toc122010184)

[Hình 3. 13 Cách hoạt động IPFS 40](#_Toc122010185)

[Hình 4. 1 Mô tả Infur 40](#_Toc122010186)

[Hình 4. 2 Mô tả về Web3.0 …………………………………………………………...41](#_Toc122010187)

[Hình 5. 1 Cấu trúc của WebRTC 45](#_Toc122010786)

[Hình 5. 2 Mô tả cách hoạt động của video call 47](#_Toc122010787)

[Hình 6. 1 Học có giám sát 49](file:///C:\Users\Vietphu\Desktop\DOC_DATN.docx#_Toc122010794)

[Hình 6. 2 Định lý Bayes 50](file:///C:\Users\Vietphu\Desktop\DOC_DATN.docx#_Toc122010795)

[Hình 6. 3 Định lý bayes mở rộng 50](file:///C:\Users\Vietphu\Desktop\DOC_DATN.docx#_Toc122010796)

[Hình 7. 1 Giao diện chính 53](#_Toc122010800)

[Hình 7. 2 Giao diện người dùng 54](#_Toc122010801)

[Hình 7. 3 Giao diện quản lý thông tin 54](#_Toc122010802)

[Hình 7. 4 Giao diện dự đoán bệnh 55](#_Toc122010803)

[Hình 7. 5 Giao diện thanh toán 55](#_Toc122010804)

[Hình 7. 6 Giao diện lịch đặt cuộc hẹn 56](#_Toc122010805)

[Hình 7. 7 Giao diện đơn thuốc 56](#_Toc122010806)

[Hình 7. 8 Giao diện cuộc gặp video call 57](#_Toc122010807)

[Hình 7. 9 Giao diện tìm kiếm bác sĩ 57](#_Toc122010808)

[Hình 7. 10 Giao diện xem thông tin bác sĩ 58](#_Toc122010809)

[Hình 7. 11 Giao diện rút tiền 58](#_Toc122010810)

[Hình 7. 12 Giao diện đặt lịch hẹn 59](#_Toc122010811)

**DANH SÁCH BẢNG**

[Bảng 2. 1 Danh sách các actor 12](#_Toc122010881)

[Bảng 2. 2 Use case bệnh nhân 14](#_Toc122010882)

[Bảng 2. 3 Use case bác sĩ 16](#_Toc122010883)

[Bảng 2. 4 Use case admin 17](#_Toc122010884)

[Bảng 2. 5 Bảng dữ liệu Appoitment](#_Toc122010885)

[Bảng 2. 6 Bảng dữ liệu User 21](#_Toc122010886)

[Bảng 2. 7 Bảng dữ liệu Doctor 23](#_Toc122010887)

[Bảng 2. 8 Bảng dữ liệu Meeting 23](#_Toc122010888)

[Bảng 2. 9 Bảng dữ liệu Review 24](#_Toc122010889)

[Bảng 4. 1 Lưu thông tin toa thuốc 42](#_Toc122010895)

[Bảng 4. 2 Lưu thông tin bệnh án 43](#_Toc122010896)

[Bảng 4. 3 Lưu thông tin hóa đơn 43](#_Toc122010897)

[Bảng 4. 4 Lưu thông tin thanh toán 43](#_Toc122010898)

**DANH SÁCH CÁC KÝ HIỆU, CHỮ VIẾT TẮT**

JSX JavaScript XML

ML Machine Learning

SDP Session Description Protocol

ICE **Interactive Connectivity Establishment**

STUN **Session Traversal Utilities for NAT**

TURN **Traversal Using Relays around NAT**

API Application Programming Interface

**MỞ ĐẦU**

**Lý do thực hiện luận văn:**

Mỗi ngày trôi qua dân số thế giới ngày càng tăng và xã hội ngày càng phát triển do đó nhu cầu về y tế và chăm sóc sức khỏe ngày càng được chú trọng và đầu tư mạnh mẽ, nhưng theo nhiều báo cáo các tổ chức y tế hiện nay chỉ ra rằng trên thế giới có hơn nữa dân số là chưa tiếp cận được với các dịch vụ y tế cơ bản vì có nhiều nguyên nhân như: chi phí thăm khám, cơ sở y tế, thiết bị không đạt tiêu chuẩn …

Hiện naytại những thành phố lớn như hiện nay, công việc bận rộn khiến cho việc thu xếp thời gian để đi khám và kiểm tra các bệnh thông thường rất khó khăn. Hệ thống y tế ngày càng quá tải dẫn đến việc chen chúc và chờ đợi rất mất thời gian.  
  
 Hơn nữa, bệnh viện là nơi tập trung rất nhiều bệnh nhân với nhiều căn bệnh khác nhau có cả những bệnh truyền nhiễm nguy hiểm. Đến bệnh viện thăm khám cũng đồng nghĩa với việc bạn phải chấp nhận rủi ro tiếp xúc với nhiều nguồn bệnh. Mặc dù, bệnh viện có đầy đủ trang thiết bị cũng như đội ngũ bác sĩ có trình độ cao nhưng nếu chỉ làm các dịch vụ như tiêm thuốc, thay băng, massage trị liệu hay truyền dịch thông thường thì việc đến bệnh viện là không cần thiết.

Tận dụng sự đổi mới và sự phát triển mạnh mẽ về công nghệ thông tin là một trong những giải pháp giúp giảm bớt gánh nặng tài chính và nâng cao khả năng tiếp cận các dịch vụ chăm sóc sức khỏe cho người dân, đặc biệt là nhóm dân số có thu nhập thấp và trung bình, đồng thời giúp cải thiện hệ thống và nâng cao chất lượng y tế. Vì thế

Với những lý do trên, trong luận văn này. Tôi muốn giới thiệu một hệ thống có thể giải quyết những vấn đề trên. Hệ thống này có thể dự đoán bệnh bằng các triệu chứng mắc phải và có thể tìm kiếm và kết nối đến với các bác sĩ có trình độ và chuyên môn cao. Hệ thống sử dụng công các công nghệ mới như block chain để giữ sự minh bạch trong trong quá trình chăm khám của bác sĩ đối với bệnh nhân.

**Phạm vi và mục tiêu**

* Luận văn này sử dụng các công nghệ blockchain để lưu trữ hồ sơ bệnh án,
* các toa thuốc của bác sĩ
* Sử dụng học máy để dự đoán bệnh cho bệnh nhân
* Website có các chức năng để bệnh nhân có thể tiếp cận với bác sĩ như tìm kiếm và gọi điện trực tiếp bằng video call
* Mục tiêu của đồ án này dành cho người bị bệnh và các bác sĩ

**Tổng quan**

Luận văn này được chia làm 7 phần:

* Cơ sở lý thuyết
* Phân tích thiết kế và hệ thống
* Giới thiệu, kiến trúc và cách hoạt động của blockchain, smartcontract và tiền điện tử
* Giới thiệu, kiến trúc và cách hoạt động của hệ sinh thái ethereum, metamask ,Goerli và IFPS
* Mô tả về các giao thức trong video call
* Mô tả về cách thức dự đoán bệnh
* Kết quả thực tế đã đạt được và những hạn chết của hệ thống
* Kết luận

# CƠ SỞ LÝ THUYẾT

## Giới thiệu

## ReactJs

### Giới thiệu về Reactjs

Hình 1. Logo ReactJS

Reactjs là một thư viện chứa nhiều JavaScript mã nguồn mở được Facebook xây dựng và phát triển. Thư viện này được sử dụng để tạo ra các ứng dụng trang web hấp dẫn với hiệu quả cao, tốc độ load nhanh và mã tối thiểu. Mỗi website sử dụng ReactJS phải chạy nhanh, mượt và có khả năng mở rộng cao, thao tác thực hiện đơn giản.

Có thể nói, hầu hết các tính năng hay sức mạnh của ReactJS thường bắt đầu từ việc tập trung vào các phần riêng lẻ. Do đó thay vì làm việc trên toàn bộ ứng dụng của website thì ReactJS lại cho phép các Developer phá vỡ giao diện phức tạp của người dùng trở nên đơn giản hơn nhiều. Điều này có nghĩa là các Render dữ liệu không chỉ được thực hiện ở vị trí server mà còn có thể thực hiện ở vị trí Client khi sử dụng ReactJS.

### ReactJs hoạt động như thế nào

Cú pháp trong HTML code ở trên rất giống với XML components. Tuy nhiên, thay vì sử dụng DOM class truyền thống, React sử dụng className.Thẻ JSX có tên thẻ, con và thuộc tính. Dấu ngoặc kép trong các thuộc tính JSX đại diện cho chuỗi. Yếu tố này tương tự như JavaScript.Ngoài ra, các giá trị số và biểu thức phải được viết bên trong dấu ngoặc nhọn.

### Các tính năng nổi bật của ReactJs

* Sử dụng **JSX:**Trong React, thay vì thường xuyên sử dụng JavaScript để thiết kế bố cục trang web thì sẽ dùng JSX. JSX được đánh giá là sử dụng đơn giản hơn JavaScript và cho phép trích dẫn HTML cũng như việc sử dụng các cú pháp thẻ HTML để render các subcomponent. JSX tối ưu hóa code khi biên soạn, vì vậy nó chạy nhanh hơn so với code JavaScript tương đương.
* Single-way data flow (**Luồng dữ liệu một chiều**): ReactJS không có những module chuyên dụng để xử lý data, vì vậy ReactJS chia nhỏ view thành các component nhỏ có mỗi quan hệ chặt chẽ với nhau. Luồng truyền dữ liệu trong ReactJS là luồng dữ liệu một chiều từ cha xuống con. Việc ReactJS sử dụng one-way data flow có thể gây ra một chút khó khăn cho những người muốn tìm hiểu và ứng dụng vào trong các dự án. Tuy nhiên, cơ chế này sẽ phát huy được ưu điểm của mình khi cấu trúc cũng như chức năng của view trở nên phức tạp thì ReactJS sẽ phát huy được vai trò của mình.
* **Virtual DOM:**Những Framework sử dụng **Virtual-DOM như ReactJS** khi Virtual-DOM thay đổi, chúng ta không cần thao tác trực tiếp với DOM trên View mà vẫn phản ánh được sự thay đổi đó. Do Virtual-DOM vừa đóng vai trò là Model, vừa đóng vai trò là View nên mọi sự thay đổi trên Model đã kéo theo sự thay đổi trên View và ngược lại. Có nghĩa là mặc dù chúng ta không tác động trực tiếp vào các phần tử DOM ở View nhưng vẫn thực hiện được cơ chế Data-binding. Điều này làm cho tốc độ ứng dụng tăng lên đáng kể – môt lợi thế không thể tuyệt vời hơn khi sử dụng Virtula-DOM.

##### Ưu điểm của ReactJS

* Hỗ trợ cộng đồng lớn
* Phổ biến trong giới StartUp
* Rất nhiều tiện ích nguồn mở
* API thanh lịch
* Dễ dàng sử dụng, tạo được các component nhẹ

##### Nhược điểm

Điều chỉnh cho JSX

Các giải pháp hoàn chỉnh yêu cầu thư viện của bên thứ ba

Tính khả dụng của các tùy chọn có thể gây nhầm lẫn

## NodeJS

### NodeJS là gì

Hình 1. Logo NodeJS

**Nodejs** là một nền tảng được xây dựng, vận hành tại V8 JavaScript runtime của Chrome. Với Nodejs, bạn có thể chạy Javascript trên server và thể xây dựng, phát triển các ứng dụng mạng nhanh chóng và dễ dàng.

Nền tảng này bắt đầu được xây dựng, phát triển tại California từ năm 2009 với phần Core phía dưới được lập trình bằng C++ gần như 100%. Điều này tạo nên ưu thế về tốc độ xử lý cũng như hiệu năng của nền tảng này. Đến nay, Nodejs vẫn đang ‘gây bão’ trong cộng đồng công nghệ bởi khả năng phát triển ứng dụng vượt trội.

### Cách hoạt động của NodeJS

Nodejs được sử dụng non-blocking để hướng sự ra vào đối với các dữ liệu thông qua các tác vụ về thời gian và thực hiện nó một cách nhanh chóng. Bởi Nodejs có khả năng xử lý một số lượng lớn các kết nối đồng thời bằng thông lượng cao và có khả năng mở rộng nhanh chóng.

Nếu như các ứng dụng về web truyền thống tạo ra các request và một luồng xử lý những yêu cầu mới và chiếm RAM của hệ thống sẽ không được sử dụng một cách có hiệu quả. Vì thế mà có giải pháp cho việc Nodejs sử dụng luồng đơn (Single-Threaded) và kết hợp với non-blocking I/O để thực thi request và cho phép hỗ trợ đến hàng chục ngàn các kết nối đồng thời.

### Đặc điểm của Nodejs

* NodeJS không cần đợi API trả dữ liệu về. Vì thế, mọi APIs nằm trong thư viện NodeJS đều không được đồng bộ
* Nó là một nền tảng chứ không phải một Framework như cách nhiều người nhầm tưởng. Chính điều này giúp Nodejs có thể xây dựng một website độc lập, hoàn chỉnh và nhanh chóng nhất
* Linh hoạt khi có thể chạy trên nhiều nền tảng khác nhau như: Window, MacOS, Linux
* Tính chất là một máy chủ đơn luồng
* Người dùng cần có các kiến thức cơ bản về lập trình căn bản như: các giao thức, Javascript,… Đây thật sự là các yêu cầu bắt buộc khi sử dụng Nodejs
* Việc sở hữu một cộng đồng và hệ sinh thái đa dạng, Nodejs tạo ra sự kết nối đặc biệt giữa các thành viên, đối tượng để chia sẻ với nhau về kiến thức, dự án,…hỗ trợ lẫn nhau cùng nhau phát triển
* Đặc điểm nổi bật quan trọng của Nodejs nằm ở phần core. Do được viết bằng ngôn ngữ C++ nên nó hiệu năng và tốc độ xử lý tương đối cao

##### Điểm mạnh của NodeJS

* Xử lý linh hoạt, đa dạng các đồng thời nhờ IO hướng sự kiện không đồng bộ
* Hỗ trợ và đáp ứng các yêu cầu về thời gian thực
* Tốc độ cực rất nhanh, đáp ứng được nhu cầu sử dụng của khách truy cập ‘khổng lồ’ trong thời gian ngắn
* Chia sẻ cùng một đoạn mã với cả phía máy chủ và máy khách
* Npm và các module rất mạnh mẽ và vẫn đang tiếp tục được theo dõi, nâng cấp
* Có một cộng đồng rộng khắp, có nhiều mã được chia sẻ qua github
* Khả năng tương thích với nhiều thiết bị, nhiều hệ điều hành như Mac OS, Window, Linux,…

##### Điểm yếu của Nodejs

* Không cung cấp khả năng mở rộng; và không thể tận dụng lợi thế của nhiều lõi thường có trong phần cứng cấp máy chủ ngày nay
* Việc triển khai thực hiện các quy trình thao tác dường như rất khó khăn đối với Nodejs
* Nếu không am hiểu và có kiến thức lập trình đủ tốt, đặc biệt là về Javascript, việc tiếp cận Nodejs sẽ là một thách thức lớn
* NodeJS không tương thích với các tác vụ đòi hỏi nhiều CPU mà chỉ phù hợp với những I/O như máy chủ web
* Khi thực hiện dữ liệu trên web hosting dùng chung, các trở ngại rất dễ phát sinh khi tải lên một ứng dụng NodeJS

## Firebase

### Firebase là gì

Firebase là một nền tảng sở hữu bởi google giúp chúng ta phát triển các ứng dụng di động và web. Họ cung cấp rất nhiều công cụ và dịch vụ tiện ích để phát triển ứng dụng nên một ứng dụng chất lượng. Điều đó rút ngắn thời gian phát triển và giúp ứng dụng sớm ra mắt với người dùng.

Firebase cung cấp cho người dùng các dịch vụ cơ sở dữ liệu hoạt động trên nền tảng đám mây với hệ thống máy chủ cực kỳ mạnh mẽ của Google. Chức năng chính của firebase là giúp người dùng lập trình ứng dụng, phần mềm trên các nền tảng web, di động bằng cách đơn giản hóa các thao tác với cơ sở dữ liệu.

Với firebase, bạn có thể tạo ra những ứng dụng real-time như app chat, cùng nhiều tính năng như xác thực người dùng, Cloud Messaging,... Bạn có thể dùng firebase giống như phần backend của app.

Các dịch vụ của firebase hoàn toàn miễn phí, tuy nhiên bạn cần phải trả thêm tiền nếu muốn nâng cấp lên. Điều này bạn nên cân nhắc nếu muốn xây dựng một ứng dụng lớn sử dụng phần backend là firebase, vì cái giá khi muốn nâng cấp còn khá đắt đỏ so với việc xây dựng [backend](https://itnavi.com.vn/blog/nhung-ky-nang-quan-trong-de-tro-thanh-mot-backend-developer/) truyền thống.

### Những tính năng chính Firebase

#### Firebase RealTimeDatabase

Firebase Authentication là chức năng xác thực người dùng.Hiểu một cách đơn giản, app của bạn cần phải đăng nhập/ đăng ký tài khoản để sử dụng, Firebase cung cấp cho chúng ta chức năng xác thực người dùng bằng email, số điện thoại, hay tài khoản Facebook, Google,...

Graphical user interface, application, Teams

Description automatically generated

Hình 1. Logo Firebase

Việc xác thực người dùng là một chức năng vô cùng quan trọng trong việc phát triển ứng dụng. Tuy nhiên, khi bạn muốn xác thực với nhiều phương thức khác nhau như email, số điện thoại, google, facebook sẽ tốn nhiều thời gian và công sức. Firebase Authentication giúp thực hiện việc đó một cách dễ dàng, giúp người dùng nhanh chóng tiếp cận sản phẩm hơn.

Vì thế, nó là một chức năng vô cùng hữu ích của firebase.Nếu bạn muốn xây dựng sản phẩm một cách nhanh chóng, hay chỉ đơn giản là làm bài tập, đồ án thì việc tích hợp Firebase Authentication và Firebase Realtime Database vào ứng dụng sẽ giúp bạn giảm rất nhiều thời gian so với các cách khác.

#### Firebase Cloud Storage

Firebase Cloud Storage là một không gian lưu trữ dữ liệu, nó giống như một chiếc ổ cứng. Bạn có thể upload và download các loại file bạn muốn. Đó có thể là một file ảnh, hay file văn bản, .zip, …

-Phân biệt Firebase cloud storage với Firebase realtime database.Câu trả lời rất đơn giản, Firebase là một cơ sở dữ liệu- nơi bạn có thể lưu trữ các thông tin về tài khoản người dùng, hay các thông tin về một mặt hàng nếu bạn xây dựng một app bán hàng.

Còn với Firebase cloud storage, chúng là nơi lưu trữ những file, đó có thể là những hình ảnh về một mặt hàng chẳng hạn. Bạn có thể lưu trữ link tới file hình ảnh trong database, còn file ảnh đặt trong cloud storage. Vậy là client có thể dễ dàng truy vấn và sử dụng.

#### Firebase Storage

Firebase Storage cho phép lưu trữ file được đăng tải bởi người dùng trong ứng dụng. Đây là nơi lưu trữ lý tưởng cho những ứng dụng có lưu trữ file hình ảnh, word, pdf, … từ phía người dùng. Hình ảnh trọ, dịch vụ vận chuyển hay những hình ảnh khác trong ứng dụng được lưu trữ tại Firebase Storage

# PHÂN TÍCH HỆ THỐNG VÀ THIẾT KẾ

## Phân tích yêu cầu hệ thống

### Phân tích yêu cầu chức năng

#### Yêu cầu về chức năng

Ứng dụng chạy trên nền tảng Website, yêu cầu phải có kết nối internet để có thể sử dụng được. Ứng dụng này được xây dựng gồm các chức năng để có thể hỗ trợ các bệnh nhân thăm khám sức khỏe như sau:

- Chức năng dự đóán bệnh cho bệnh nhân:

* + Cho phép người dùng dự đoán bệnh thông qua các triệu chứng
  + Cho phép lưu thông tin bệnh từ những chiệu trứng đó
  + Cho phép bệnh nhân xem lại lịch sử dự đoán

- Chức năng tìm kiếm bác sĩ:

* + Cho phép bệnh nhân tìm kiếm bác sĩ từ hệ thống
  + Có thể cho bênh nhân xem thông tin bác sĩ
  + Có thể đánh giá bác sĩ
  + Có thể bình luận về chất lượng của bác sĩ

- Chức năng đặt lịch hẹn:

* + Cho phép bệnh nhân đặt lịch để thăm khám
  + Cho phép xem thông tin về lịch đặt hẹn
  + Bệnh nhân có thể thanh toán cuộc hẹn đã đặt

- Chức năng quản lý bệnh nhân:

* + Cho phép bác sĩ xem thông tin bệnh nhân
  + Cho phép bác sĩ quản lý các cuộc hẹn
  + Cho phép bác sĩ chẩn đoán và kê toa thuốc cho bệnh nhân

- Chức năng chat: cho phép bênh nhân và bác sĩ có thể trao đổi thông tin qua video call và chat

#### Yêu cầu lưu trữ

Hệ thống cần lưu trữ các thông tin như sau:

- Lưu trữ trên nền tẳng blockchain:

* + Thông tin về toa thuốc: Thông tin bao gồm các đơn thuốc và chẩn đoán của bác sĩ
  + Thông tin dự đoán bệnh: thông tin về các triệu chứng và các mô tả khác

- Lưu trữ trên các cớ sở dữ liệu thông thường:

* + Thông tin về các cuộc đặt lịch: Thông tin về thời gian, bác sĩ, giá tiền,tình trạng của cuộc đặt lịch
  + Thông tin việc thanh toán: Thông tin về giá tiền, tình trạng của việc thanh toán,..
  + Thông tin cá nhân: Thông tin cá nhan của bác sĩ và của bệnh nhân
  + Thông tin về cuộc trò chuyện: Nội dung chat giữa bệnh nhân và bác sĩ
  + Thông tin về đánh giá bác sĩ: Số lượng thích, và bình luận của bệnh nhân

#### Phân tích yêu cầu chức năng

* Website sử dụng ngôn ngữ tiếng Anh, chạy tốt trên nền tảng Chrome, có giao diện thân thiện với người dùng
* Sử dụng các công cụ miễng phí hiện có
* Các thông tin về toa thuộc phải minh bạch với bệnh nhân

#### Mô số use case của hệ thống

##### Danh sách Actors:

|  |  |  |
| --- | --- | --- |
| STT | Tên Actor | Ý Nghĩa |
| 1 | Admin | Người có quyền quản lý toàn bộ hệ thống |
| 2 | Bệnh nhân | Người sử dụng dịch vụ |
| 3 | Bác sĩ | Người sử dụng dịch vụ |

Bảng 2. Danh sách các actor

##### Mô tả use case bệnh nhân

Diagram

Description automatically generated

Hình 2. Use case bệnh nhân

|  |  |  |
| --- | --- | --- |
| **STT** | **Tên Use case** | **Mô tả Use case** |
| 1 | Đăng nhập/Đăng xuất | Người dùng có thể đăng nhập vào hệ thống bằng tài khoản bằng Google hoặc tài khoản tạo trên hệ thống trước đó. Và có thể đăng xuất ra khỏi hệ thống |
| 2 | Quản lý thông tin | Cho phép bệnh nhân có thêm các thông tin các nhân của mình phần này có quyền không điền |
| 3 | Tìm kiếm và xem thông tin bác sĩ | Cho phép bệnh nhân tìm kiếm các thông tin về các bác sĩ hiện có trên hệ thông, có thể xem các bình luận đánh giá của người khác cho bác sĩ |
| 4 | Đặt lịch khám bệnh | Sau khi tìm được bác sĩ phù hợp bệnh nhân có thể đặt lịch hẹn cho bác sĩ, về ngày tháng và ghi chú lại cho bác sĩ |
| 5 | Video call và chat | Bệnh nhân và bác sĩ có thể gặp mặt trực tiếp thông qua cuộc gọi video hoặc có thể nhắn tin cho nhau |
| 6 | Đánh giá bác sĩ | Bệnh nhân có thể đánh giá về chất lượng phục vụ của bác sĩ thông qua phần bình luận và sao |
| 7 | Xem toa thuốc | Bệnh nhân có thể xem các đơn thuốc mà bác sĩ đã kê khai cho bệnh nhân |
| 8 | Thanh toán cho cuộc đặt lịch | Sau khi đặt lịch hẹn thì bệnh nhân phải tiến hành thanh toán để hoành tất quá trình |

Bảng 2. Use case bệnh nhân

##### Mô tả use case bác sĩ

Diagram

Description automatically generated

Hình 2. Use case bác sĩ

|  |  |  |
| --- | --- | --- |
| **STT** | **Tên Use case** | **Mô tả Use case** |
| 1 | Đăng nhập | Người dùng có thể đăng nhập vào hệ thống bằng tài khoản bằng Google hoặc tài khoản tạo trên hệ thống trước đó. Và có thể đăng xuất ra khỏi hệ thống |
| 2 | Quản lý thông tin  Cá nhân | Sau khi tạo tài khoản thì bác sĩ phải tiến hành hoàn thành tất cả các thông tin của mình để xác thực tài khoản |
| 3 | Kê toa thuốc | Sau khi thăm khám xong thì bác sĩ sẽ phải kê toa thuốc cho bệnh nhân |
| 4 | Đồng ý/ Hủy lịch khám | Sau khi bệnh nhân đặt lịch hẹn, bác sĩ có quyền không đồng ý lịch hẹn này |
| 5 | Video call và chat | Bệnh nhân có thể đánh giá về chất lượng phục vụ của bác sĩ thông qua phần bình luận và sao |
| 6 | Rút tiền | Bác sĩ có thể rut tiền của mình thông qua các cuộc hẹn đã đặt |
| 7 | Xem toa thuốc | Bác sĩ có thể xem tao thuốc đã tạo cho bệnh nhân |
| 8 | Xem lịch sử đặt lịch | Bác sĩ có thể xem lịch sử đặt lịch hẹn của mình |

Bảng 2. Use case bác sĩ

##### Mô tả use case Admin

Diagram

Description automatically generated

Hình 2. Use case admin

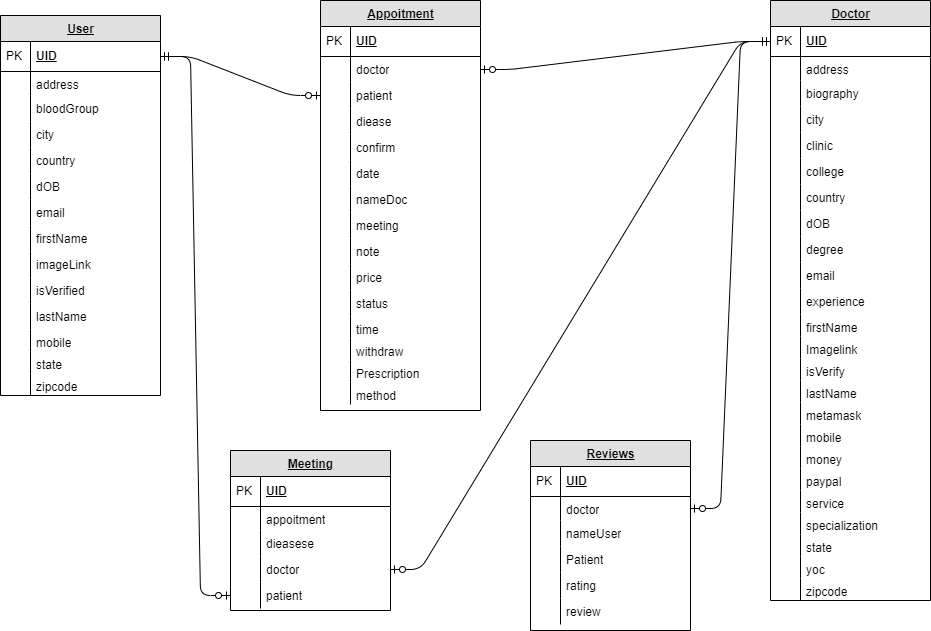
|  |  |  |
| --- | --- | --- |
| **STT** | **Tên Use case** | **Mô tả Use case** |
| 1 | Đăng nhập/Đăng xuất | Admin có thể đăng nhập vào hệ thống thông qua tài khoản đã được tạo và có thể đáng xuất tài khoản ra khỏi hệ thống |
| 2 | Tìm kiếm thông tin bác sĩ | Cho phép bệnh nhân tìm kiếm các thông tin về các bác sĩ hiện có trên hệ thông, có thể xem các bình luận đánh giá của người khác cho bác sĩ |
| 3 | Xác thực bác sĩ | Admin có thể vô hiệu hóa hoặc bỏ vô hiệu hóa tài khoản của bác sĩ |
| 4 | Xác nhận thanh toán rút tiền | Admin có thể xác thực thanh toán rút tiền của bác sĩ |

Bảng 2. Use case admin

## Phân tích thiết kế hệ thống

### Thiết kế cơ sở dữ liệu

#### Lược đồ cơ sở dữ liệu



Hình 2. Lược đồ cơ sở dữ liệu

#### Mô tả các bảng dữ liệu

**Bảng dữ liệu Appoitment**: Lưu thông tin về các cuộc đặt lịch

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **STT** | **Tên thuộc tính** | **Kiểu dữ liệu** | **Ràng buộc** | **Ý nghĩa** |
| 1 | UID | UUID | Khóa chính | Khóa chính và là duy nhất |
| 2 | doctor | string |  | Id của bác sĩ |
| 3 | patient | string |  | Id của bênh nhân |
| 4 | diease | string |  | Id của chẩn đoán bệnh |
| 5 | confirm | string |  | Bác sĩ đã xác thực cuộc hẹn này chưa |
| 6 | date | date |  | Ngày sinh của bác sĩ |
| 7 | nameDoc | string |  | Tên của hiển thị |
| 8 | meeting | string |  | Id của phỏng video call |
| 9 | note | string |  | Các thông tin của cuộc hẹn |
| 10 | price | float |  | Gía tiền của cuộc hẹn |
| 11 | status | bool |  | Tình trạng thanh toan của cuộc hẹn |
| 12 | time | date |  | Thời gian mà bệnh nhân đặt lịch |
| 13 | withdraw | string |  | Số tiền của cuộc hẹn này đã rút chưa |
| 14 | Prescription | string |  | Toa thuốc của cuộc hẹn này đã được kê chưa |
| 15 | method | string |  | Phương thức thanh toán của cuộc hẹn này là gì |

Bảng 2. Bảng dữ liệu Appoitment

**Bảng dữ liệu User**: Lưu thông tin về bệnh nhân

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| STT | Tên thuộc tính | Kiểu dữ liệu | Ràng buộc | Ý nghĩa |
| 1 | UID | UUID | Khóa chính | Khóa chính và là duy nhất |
| 2 | address | string |  | Địa chỉ của bệnh nhân |
| 3 | bloodGroup | string |  | Nhóm máu của bệnh nhân |
| 4 | city | string |  | Thành phố hiển thị |
| 5 | country | string |  | Quốc gia hiển thị |
| 6 | dOB | date |  | Ngày tháng năm sinh hiển thị |
| 7 | email | string |  | Email hiển thị |
| 8 | firstName | string |  | Tên hiển thị |
| 9 | imageLink | string |  | Địa chỉ ảnh của bệnh nhân |
| 10 | isVerified | bool |  | Tài khoản này đã xác thực chưua |
| 11 | lastName | string |  | Tên hiên thị |
| 12 | mobile | string |  | Số điện thoại của bệnh nhân |
| 13 | state | string |  | Khu vực mà bênh nhân sống |
| 14 | zipcode | string |  | Mã code thành phố của bệnh nhân |

Bảng 2. Bảng dữ liệu User

**Bảng dữ liệu Doctor**: Lưu thông tin bác sĩ

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| STT | Tên thuộc tính | Kiểu dữ liệu | Ràng buộc | Ý nghĩa |
| 1 | UID | UUID | Khóa chính | Khóa chính và là duy nhất |
| 2 | address | string |  | Địa chỉ của bác sĩ |
| 3 | biography | string |  | Thông tin sơ bộ về bác sĩ |
| 4 | city | string |  | Thành phố đang ở |
| 5 | clinic | string |  | Phòng khám đang hoạt động |
| 6 | college | string |  | Đại học bác sĩ đã học |
| 7 | country | string |  | Thành phố đang ở |
| 8 | dOB | date |  | Ngày tháng năm sinh của bác sĩ |
| 9 | degree | string |  | Bằng đại học |
| 10 | email | string |  | Email hiển thị |
| 11 | experience | string |  | Kinh nghiệm làm việc trong nghề |
| 12 | firstName | string |  | Tên hiển thị |
| 13 | Imagelink | string |  | Đia chỉ hình ảnh của bác sĩ |
| 14 | isVerify | bool |  | Tài khoản này đã xác thực chưa |
| 15 | lastName | string |  | Tên hiển thị |
| 16 | metamask | string |  | Id tài khoản của metamask đã thanh toán |
| 17 | mobile | string |  | Số điện thoại của bác sĩ |
| 18 | money | string |  | Số tiền mà bác sĩ tính cho cuộc hẹn |
| 19 | paypal | string |  | Id của tài khoản paypal đã thanh oán |
| 20 | service | string |  | Các dịch vụ bác sĩ phục vụ |
| 21 | specialization | string |  | Các chuyên ngành của bác sĩ |
| 22 | state | string |  | Khu vực đang ở của bác sĩ |
| 23 | yoc | string |  | Ngày tháng năm sinh của bác sĩ |
| 24 | zipcode | string |  | Mã zip của thành phố đang ở |

Bảng 2. Bảng dữ liệu Doctor

**Bảng dữ liệu Meeting** Lưu thông tin về các cuộc đặt lịch

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| STT | Tên thuộc tính | Kiểu dữ liệu | Ràng buộc | Ý nghĩa |
| 1 | UID | UUID | Khóa chính | Khóa chính và là duy nhất |
| 2 | appoitment | string |  | Id của cuộc lịch đã đặt |
| 3 | dieasese | string |  | Id của chẩn đoán bệnh |
| 4 | doctor | string |  | Id của bác sĩ |
| 5 | patient | string |  | Id của bệnh nhân |

Bảng 2. Bảng dữ liệu Meeting

**Bảng dữ liệu Review**: Lưu thông tin đánh giá của bác sĩ

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| STT | Tên thuộc tính | Kiểu dữ liệu | Ràng buộc | Ý nghĩa |
| 1 | UID | UUID | Khóa chính | Khóa chính và là duy nhất |
| 2 | doctor | string |  | Id của bác sĩ |
| 3 | nameUser | string |  | Tên của người đánh giá |
| 4 | Patient | string |  | Id của bệnh nhân đã đánh giá |
| 5 | rating | int |  | Số sao mà bệnh nhân đánh giá |
| 6 | review | string |  | Bình luận của bện nhân về bác sĩ |

Bảng 2. Bảng dữ liệu Review

# BLOCKCHAIN, SMART CONTRACT, TIỀN ĐIỆN TỬ VÀ NHỮNG ỨNG DỤNG CỦA CHÚNG

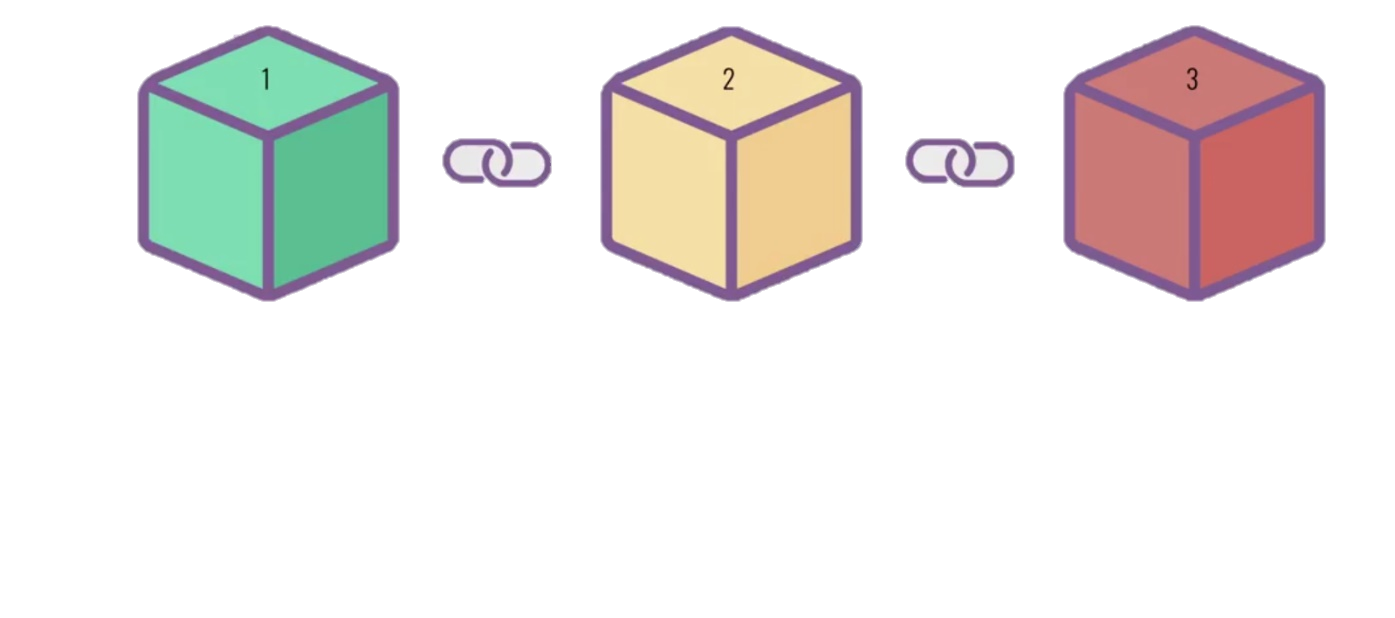
## Giới thiệu

## Blockchain

### Blockchain là gì

**Blockchain** hay còn gọi với tên chuỗi khối, block chain,… là một hệ thống cơ sở dữ liệu dạng chuỗi – khối cho phép lưu trữ và truyền tải thông tin một cách an toàn được liên kết, mở rộng với nhau nhờ các thuật toán mã hóa vô cùng phức tạp.

Nói một cách dễ hiểu, Blockchain được xem như là một cuốn sổ cái của công ty nơi mà mọi hoạt động liên quan đến tiền tệ của công ty được quản lý giám sát chặt chẽ



**CHAIN**

**CHAIN**

Hình 3. Minh họa các chuỗi

**BLOCK**

**BLOCK**

**BLOCK**

* ***Hiểu về mặt kỹ thuật thì:***
* **Block**: là một khối (có hình vuông hoặc hình chữ nhật).
* **Chain**: là một chuỗi.
* **Blockchain:** là một chuỗi các khối (có hình vuông hoặc hình chữ nhật) được liên kết với nhau tạo thành một hệ thống chuỗi lớn.
* **Cấu trúc của 1 block:**

Ví dụ ta có 1 block sau:

Graphical user interface, text, application, email

Description automatically generated

Hình 3. Cấu trúc của một block

* **BLOCK HEIGHT** 8779340 chính là số thứ tự của block  ( cho biết chiều dài của blockchain )
* TIMESTAMP: ngày và giờ mà khối này được đào, ở đây là 12 ngày 21 giờ trước vào ngày 17/6
* TRANSACTION: ta thấy là có 15 transaction (số lượng giao dịch trong khối) và 89 contract internal transaction (giao dịch là kết quả của việc thực hiện hợp đồng liên quan đến giá trị Ether)
* **MINE BY:**ở đây ta thấy 1 dãy kí tự này chính là địa chỉ ví của người đào thành công
* **BLOCK**REWARD: phần thưởng mà thợ đào sẽ nhận được ( 1 lượng ether hữu hạn trên các khoản phí được trả cho tất cả giao dịch có trong khối)
* **UNCLE**REWARD: Một unlce block có phần thưởng thấp hơn đáng kể so với một khối bình thường. Phần thưởng Uncle block hợp lệ nhưng bị từ chối vì nó không nằm trên chuỗi dài nhất, đây là cơ chế hoạt động của blockchain. Uncle block rất quan trọng trong Ethereum vì nó bảo mật cho chuỗi khối.)
* DIFFICULTY:sự nỗ lực cần thiết khi đào 1 khối mới và độ khó của thuật toán này có thể điều chỉnh theo thời gian
* **TOTAL**DIFFICULTY : tổng độ khó của chuỗi đến block này (15,035,361)
* SIZE : kích thước của khối này và sẽ được xác định bởi gas limit của khối (23,640 bytes)
* **GAS**USED:tổng lượng gas được sử dụng trong khối là 6,733,736  và tỉ lệ phần trăm nó chứa đầy trong khối (67,30%)
* **GAS**LIMIT : đây là số lượng gas tối đa mà người gửi có thể trả cho 1 transaction (10,004,811)

### Kiến trúc của blockchain

* Một khối block sẽ được lưu trữ gồm 3 phần:
* **Thành phần dữ liệu (Data)**: dữ liệu sẽ tùy thuộc vào từng loại blockchain. Chẳng hạn như Blockchain của Bitcoin sẽ chứa giao dịch.
* **Thành phần mã hóa (Hash):** Hash của khối hiện tại như một đặc điểm để nhận dạng. Nó là duy nhất và không trùng nhau giống như vân tay của chúng ta vậy.
* **Thành phần mã hóa của node trước (Hash of previous block):** các block này kết nối với nhau thông qua mã hash của khối block trước tạo thành một chuỗi hệ thống được gọi là Blockchain

Ảnh có chứa văn bản, đồng hồ

Mô tả được tạo tự động

Hình 3. Kiến trúc của chuỗi

* **Chuỗi liên kết:**
* Một chuỗi, hay một danh sách liên kết, được tạo ra khi mỗi đơn vị dữ liệu đồng thời lưu trữ một giá trị Hash của đơn vị dữ liệu trước đó. Chỉ cần có một thay đổi trong dữ liệu trước đó, thì giá trị hash sẽ bị thay đổi và mối liên kết đó sẽ bị phá vỡ. Một cấu trúc như vậy phù hợp cho việc lưu trữ và liên kết các khối dữ liệu không xuất hiện đồng thời mà xảy ra lần lượt từng thời điểm.

Text

Description automatically generated with low confidence

Hình 3. Chuỗi liên kết dữ liệu

* Trên hình 1.5, mã Hash R3 có thể dùng để truy xuất ngược lại toàn bộ dữ liệu trong chuỗi đến giá trị đầu tiên. R3 được gọi là đỉnh của chuỗi.

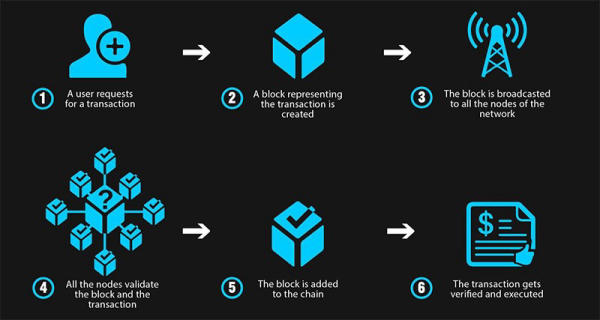
### Cách thức hoạt động blockchain

Đầu tiên, người dùng sẽ bắt đầu một giao dịch và ký kết bằng khóa riêng của nó. Khóa riêng về cơ bản sẽ tạo ra một chữ ký số duy nhất và đảm bảo rằng không ai có thể thay đổi nó. Và nếu có bất cứ ai cố gắng thay đổi thông tin giao dịch thì chữ ký số sẽ tự thay đổi và không thể xác minh nó và sau đó bị loại bỏ.

Tiếp theo, giao dịch sẽ được truyền bá rộng trên các nút xác minh. Về cơ bản, nền tảng blockchain có thể sử dụng nhiều phương pháp khác nhau để xác minh xem giao dịch có hợp lệ hay không. Các phương pháp hay thuật toán này được gọi là thuật toán đồng thuận.

Sau khi giao dịch được xác thực thì nó sẽ chiếm được một vị trí trên sổ cái và nó sẽ không bị thay đổi vì nó chứa dấu thời gian và một ID duy nhất.

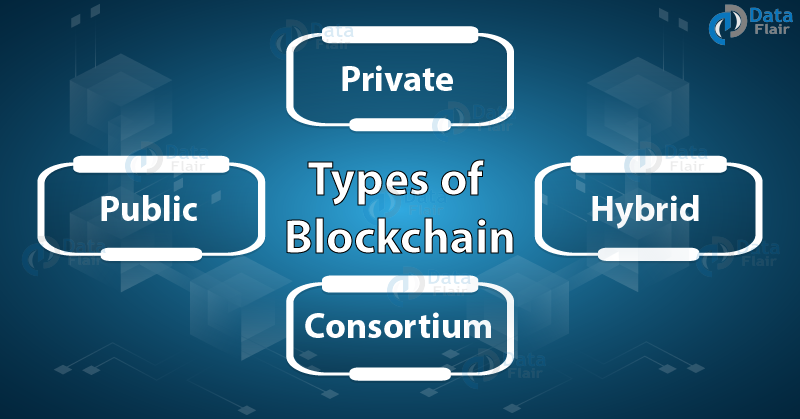
Và khối đằng sau sẽ liên kết với khối trước đó sau đó thì một khối mới sẽ tạo một mối liên kết với khối này,… cứ tiếp tục như vậy nó tạo ra một chuỗi các khối do đó nó có tên là blockchain.



Hình 3. Cách hoạt động của blockchain

## Các loại Blockchain

Trong hệ thống Blockchain chia thành 4 loại chính gồm: Public, Private, Permissioned và Hybrid Blockchain



Hình 3. Các loại blockchain

* **Public blockchain**

Public: Đây là hệ thống blockchain mà bất kỳ ai cũng có quyền đọc và ghi dữ liệu trên Blockchain được.

Quá trình xác thực giao dịch trên Blockchain này đòi hỏi phải có hàng nghìn hay thậm chí là hàng vạn nút tham gia. Do đó để tấn công vào hệ thống Blockchain này là điều bất khả thi vì chi phí rất cao. Ví dụ về public blockchain: **Bitcoin**, **Ethereum**…

* **Private blockchain**

Private: Đây là hệ thống blockchain cho phép người dùng chỉ được quyền đọc dữ liệu, không có quyền ghi vì điều này thuộc về một bên thứ ba tuyệt đối tin cậy.

Bên thứ ba này có thể hoặc không cho phép người dùng đọc dữ liệu trong một số trường hợp. Bên thứ ba toàn quyền quyết định mọi thay đổi trên Blockchain.

Vì đây là một Private Blockchain, cho nên thời gian xác nhận giao dịch khá nhanh vì chỉ cần một lượng nhỏ thiết bị tham gia xác thực giao dịch.

* **Permissioned blockchain**

Permissioned: Hay còn gọi là Consortium. Đây là một dạng của Private Blockchain nhưng bổ sung thêm một số tính năng nhất định.

Nó kết hợp giữa “niềm tin” khi tham gia vào Public và “niềm tin tuyệt đối” khi tham gia vào Private.

* **Hybrid blockchain**

Hybrid: Đây là blockchain kết hợp các đặc điểm giữa private và public blockchain.

Đảm bảo ẩn danh trong khi vẫn được liên kết với mạng công cộng.

Nó hoạt động trong một hệ sinh thái khép kín và cũng bảo vệ quyền riêng tư nhưng cho phép giao tiếp với bên thứ ba.

## Cơ chết mã hóa Blockchain

### Cryptography

Cryptography là một phương pháp phát triển các kỹ thuật và giao thức để ngăn chặn bên thứ ba truy cập và thu thập kiến ​​thức về dữ liệu từ các thông điệp riêng tư trong quá trình giao tiếp. Mật mã cũng được tạo thành từ hai thuật ngữ Hy Lạp cổ đại, Kryptos và Graphein, thuật ngữ trước đây có nghĩa là "ẩn" và sau là "viết". Có một số thuật ngữ liên quan đến mật mã, được nêu như sau:

Encryption**:**Nó là một quá trình chuyển từ bản rõ (văn bản bình thường) sang bản mã (chuỗi bit ngẫu nhiên).

Decryption**:**Quá trình ngược mã hóa, chuyển đổi bản mã sang bản rõ.

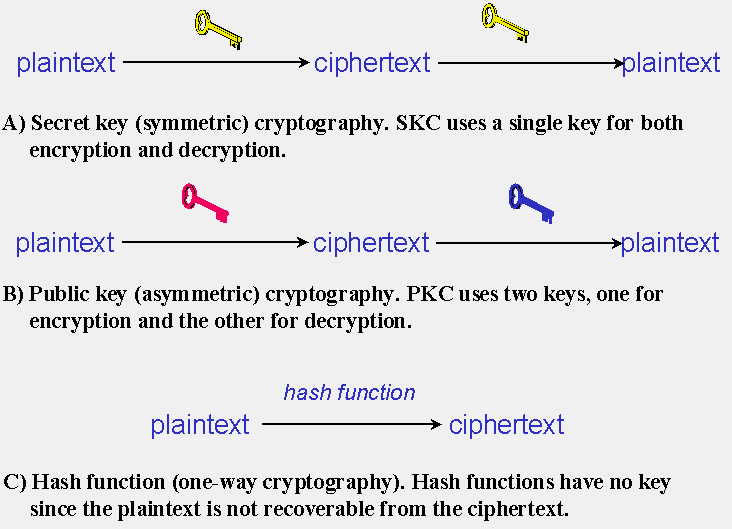
Cipher**:**Hàm toán học, tức là một thuật toán mật mã được sử dụng để chuyển bản rõ sang bản mã.

Key**:**Một lượng nhỏ thông tin được yêu cầu để tạo ra kết quả của thuật toán mật mã.

* + **Các loại Cryptography**

Để hiểu cryptography trong blockchain, người ta phải hiểu các loại mật mã. Có ba cách khác nhau mà chúng ta có thể thực hiện các thuật toán mật mã, đó là mật mã khóa đối xứng, mật mã khóa bất đối xứng và hàm băm.

* Symmetric-Key Cryptography: Trong phương pháp mã hóa này, ta đưa một khóa duy nhất ss một khóa đơn chung sẽ tạo ra vấn đề chuyển khóa an toàn giữa người gửi và người nhận. Nó còn được gọi là Mật mã khóa bí mật.
* Asymmetric-Key Cryptography**:** Phương pháp mã hóa này sử dụng một cặp khóa, khóa mã hóa và khóa giải mã, được đặt tên lần lượt là khóa công khai và khóa riêng. Cặp khóa được tạo bởi thuật toán này bao gồm một khóa riêng tư và một khóa công khai duy nhất được tạo bằng cùng một thuật toán. Nó còn được gọi là Mật mã khóa công khai.
* Hash Functions**:**Loại mã hóa này không sử dụng khóa. Nó sử dụng một mật mã để tạo ra một giá trị băm có độ dài cố định từ bản rõ. Gần như không thể khôi phục nội dung của văn bản thuần túy từ bản mã.



Hình 3. Các loại cryptography

## Hàm băm và tầm quan trọng

* + **Hàm băm**

Băm là quá trình chuyển đổi đầu vào gồm các chữ cái và ký tự có kích thước không cố định để tạo đầu ra có kích thước cố định; bằng cách sử dụng các công thức toán học như các hàm băm (được thực hiện dưới dạng các thuật toán băm) và rất cần thiết để quản lý blockchain trong tiền điện tử.

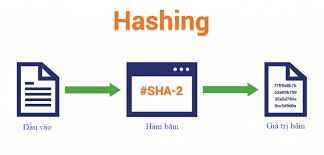
Không phải hàm băm nào cũng sử dụng mật mã hóa, tuy nhiên cái gọi là hàm băm mật mã hóa chính là cốt lõi của tiền mã hóa. Nhờ chúng mà blockchain và các hệ thống phân tán khác có thể đạt được tính toàn vẹn và bảo mật của dữ liệu ở mức độ cao. Vì tính thông dụng của bảng băm, ngày nay, đa số ngôn ngữ lập trình đều cung cấp thư viện ứng dụng bảng băm, thường gọi là thư viện collection trong đó có các vấn đề như: tập hợp (collection), danh sách (list), bảng(table), ánh xạ (mapping), từ điển (dictionary). Thông thường, các lập trình viên chỉ cần viết hàm băm cho các đối tượng nhằm tích hợp với thư viện bảng băm đã được xây dựng sẵn.

SHA 256 : kết quả đầu vào bất kỳ cho ra kết quả đầu ra là 256 bit gồm 64 ký tự. Nó được sử dụng ở mạng Bitcoin.

SHA 3 (Keccak-256): được sử dụng mạng Ethereum

Điều kiện của 1 hàm băm tốt:

* Tính toán nhanh.
* Các khoá được phân bố đều trong bảng.
* Ít xảy ra đụng độ.
* Xử lý được các loại khóa có kiểu dữ liệu khác nhau.



Hình 3. Hashing

* + **Tầm quan trọng của hàm băm**

Các hàm băm truyền thống có nhiều trường hợp sử dụng, bao gồm tra cứu cơ sở dữ liệu, phân tích tệp lớn và quản lý dữ liệu. Mặt khác, các hàm băm mật mã còn được sử dụng rộng rãi trong các ứng dụng bảo mật thông tin, chẳng hạn như xác thực thông điệp và dấu vân tay kỹ thuật số. Khi được sử dụng trong Bitcoin, các hàm băm mật mã là một phần thiết yếu của quá trình đào và cũng góp phần tạo ra các địa chỉ và khóa mới.

Băm thật sự là một công cụ mạnh mẽ khi cần xử lý một lượng thông tin lớn. Ví dụ, chúng ta có thể chạy một tệp hoặc tập dữ liệu lớn thông qua hàm băm và sau đó sử dụng kết quả đầu ra từ của nó để nhanh chóng xác minh tính chính xác và tính toàn vẹn của dữ liệu. Điều này là có thể bởi vì tính chất tất định của các hàm băm: đầu vào sẽ luôn dẫn đến một đầu ra ngắn gọn, đơn giản hóa (băm). Nhờ vào kỹ thuật này, sẽ không còn phải lưu trữ và “ghi nhớ” một lượng thông tin lớn nữa.

Băm là quy trình đặc biệt hữu ích trong công nghệ blockchain. Blockchain Bitcoin có một số hoạt động sử dụng quy trình băm, hầu hết các hoạt động đó là trong quá trình đào. Trên thực tế, gần như tất cả các giao thức tiền mã hóa đều dựa vào băm để liên kết và rút gọn các nhóm giao dịch thành các khối và cũng để tạo ra các liên kết mật mã giữa mỗi khối, tạo ra một blockchain một cách hiệu quả.

## Smart Contract

### Smart Contract là gì

Smart Contract (hay Hợp đồng thông minh) là các chương trình chạy trên blockchain. Hợp đồng thông minh cũng giống như một hợp đồng kỹ thuật số bị bắt buộc thực hiện bởi một bộ quy tắc cụ thể. Các quy tắc này do bộ mã máy tính xác định trước mà tất cả các nút (node) trong mạng đều phải sao chép và thực thi các quy tắc đó.

Về bản chất, Smart Contract chỉ là một đoạn mã chạy trên một hệ thống phân tán, cho phép tạo ra các giao thức Permissionless (tức là không cần trao quyền). Điều đó có nghĩa là:

* Hai bên trong hợp đồng có thể đưa ra các cam kết thông qua blockchain mà không cần phải biết về danh tính hay tin tưởng lẫn nhau.
* Họ có thể đảm bảo rằng nếu các điều kiện của hợp đồng không được thỏa mãn, hợp đồng sẽ không được thực thi.

Ngoài ra, việc sử dụng hợp đồng thông minh loại bỏ nhu cầu đối với các bên trung gian, giúp giảm đáng kể chi phí hoạt động.

Graphical user interface, application

Description automatically generated

Hình 3. Smart Contract

Mỗi blockchain có một phương pháp triển khai hợp đồng thông minh khác nhau, ví dụ trên Cosmos có WASM, Polkadot có ink!,... Trong đó nổi bật nhất vẫn là Smart Contract chạy trên máy ảo của Ethereum (Ethereum Virtual Machine - EVM).

## Tiền điện tử

### Tiền điện tử là gì

Tiền điện tử là loại tiền được tạo ra từ những bit số hay còn được gọi là tiền mã hóa. Loại tiền này chỉ được sử dụng duy nhất trên môi trường internet để thanh toán chi phí, đầu tư,… Khi giao dịch, người tham gia phải đảm bảo được 3 yếu tố đó là: Kết nối internet, mạng máy tính và phương tiện điện tử của tổ chức phát hành.

### Ưu điểm tiền điện tử

* Giao dịch nhanh chóng: Người sử dụng có thể nhận tiền và chuyển tiền mọi lúc mọi nơi một cách nhanh chóng.
* Phí giao dịch thấp: Chi phí giao dịch của tiền điện tử hầu hết là miễn phí hoặc phí rất thấp.
* An toàn, bảo mật: Thông tin của khách hàng sẽ được bảo mật một cách tốt nhất. Với công nghệ tiên tiến, việc gian lận sẽ được hạn chế và không phải phụ thuộc vào bên trung gian.
* Phát triển ngành thương mại điện tử: Sử dụng tiền điện tử để mua sắm trực tuyến đang rất phổ biến hiện nay. Điều này đã thúc đẩy phát triển song song giữa tiền điện tử và thương mại điện tử.
* Minh bạch: Với công nghệ blockchain, mọi thông tin giao dịch đều được lưu trữ trong chuỗi khối. Do đó, 2 bên giao dịch hoàn toàn có thể xác minh và theo dõi tiền điện tử một cách dễ dàng và nhanh chóng.

# HỆ SINH THÁI ETHEREUM VÀ IFPS

## Giới thiệu

Như những gì đã đề cập ở chương 2 trong đồ án này tôi sử dụng công nghệ blockchain và kết hợp với hệ sinh thái của ethereum. Trong chương này tôi sẽ giới thiệu và mô tả về hệ sinh thái của ethereum mà tôi đã sử dụng trong đồ án lần này bên cạnh đó

Tôi cũng sẽ giới thiệu và mô tả về infura một cơ sở hạ tầng cực kỳ quan trọng và cốt lõi của ethereum

## Hệ sinh thái Ethereum

### Ethereum là gì

Ethereum là một nền tảng blockchain phi tập trung, có khả năng thực thi các hợp đồng thông minh. Hợp đồng thông minh cho phép người tham gia giao dịch với nhau mà không cần đến sự giám sát của cơ quan nào. Các điều khoản được ghi trong hợp đồng được thực thi tự động, không ai có thể can thiệp. Hồ sơ giao dịch không thể thay đổi, có thể xác minh và phân phối an toàn trên internet giúp người tham gia toàn quyền sở hữu.

Ethereum cũng cho phép các nhà phát triển xây dựng các ứng dụng phi tập trung (DApps) và các tổ chức tự trị phi tập trung (DAOs).

Cho đến hiện nay thì tất cả các ứng dụng lớn của thị trường crypto đều bắt nguồn và phát triển trên Ethereum đặc biệt đó là Defi và NFT.

### Metamask là gì

Metamask được tích hợp như một tiện ích mở rộng (extensions) để hoạt động trên nhiều tiện ích phổ biến hiện nay như: Metamask Chrome, FireFox, Brave và Opera giống như trên các ứng dụng di động.

Ví Metamask cho phép bạn tương tác với hệ sinh thái Ethereum mà không cần phải đồng bộ toàn bộ Blockchain trên thiết bị hay máy tính. Chính bởi điều này cộng với việc giao diện thân thiện dễ thao tác đã chiếm trọn sự tin dùng của người dùng.

Graphical user interface, text, application

Description automatically generated

Hình 3. Metamask

### Gas là gì

**Gas** là một đơn vị đo lường nhằm tính toán chi phí mà nó sẽ cần để thực hiện các hoạt động nhất định, có thể là một giao dịch đơn giản, hoặc một hợp đồng thông minh, hoặc thậm chí một ICO cũng sẽ tiêu tốn một lượng gas.

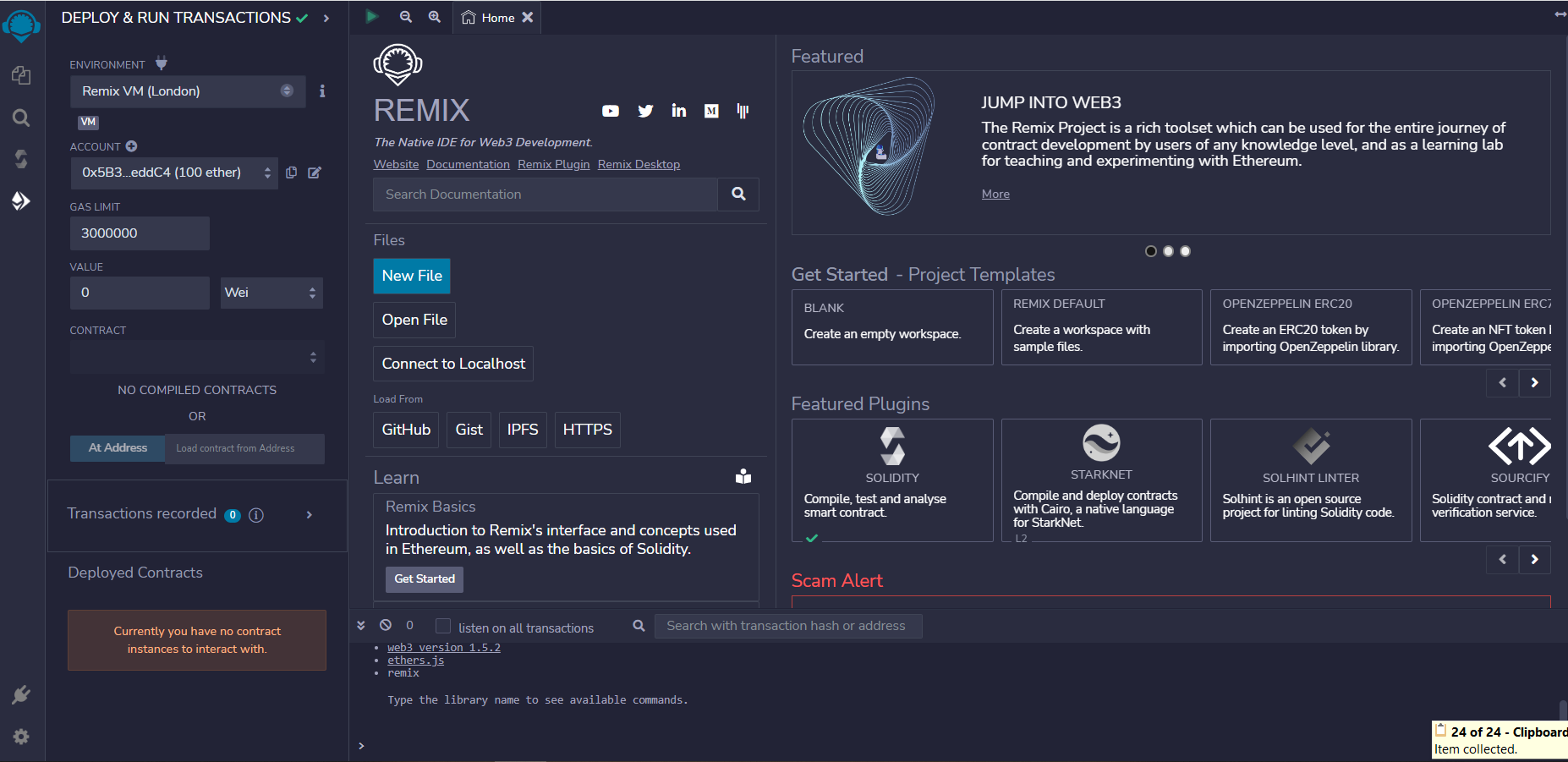
Như vậy chúng ta đã nắm được các khái niệm cơ bản về Ethereum. Tiếp theo, chúng ta sẽ đi tìm hiểu là thế nào để tạo một account và một transaction trên môi trường

### Solidity là gì

Solidity là một ngôn ngữ lập trình để tạo nên các smart contract (hợp đồng thông minh) trên nền tảng Ethereum. Đây được xem là một nền tảng hợp đồng thông minh phi tập trung hàng đầu trong lĩnh vực crypto. Cơ bản có thể xây dựng NFT marketplace (chợ nghệ thuật kỹ thuật số), Metaverse, sàn giao dịch phi tập trung (DEX), sàn tài chính phi tập trung (DeFi)... trên Ethereum.

Solidity là một ngôn ngữ lập trình cấp cao, đối tượng sử dụng của ngôn ngữ Solidity là những lập trình viên muốn phát triển các ứng dụng trên Ethereum. Bằng cách sử dụng sự kết hợp lý giữa chữ cái và số Solidity giúp các developer viết chương trình dễ dàng hơn.

Đây là ngôn ngữ lập trình hướng contract thuộc hệ sinh thái của Ethereum. Nhưng với câu lệnh và cú pháp gần giống như trên các ngôn ngữ lập trình cơ bản phổ biến (C#, Java, PHP …)



Hình 3. Solidity

### IPFS

1. *IFPS là gì:*

IPFS là viết tắt của từ Interplanetary File System, một hệ thống tập tin phân tán ngang hàng kết nối tất cả các thiết bị máy tính với nhau. Cụ thể hơn, nó sẽ phân phối dữ liệu được lưu trữ theo hình thức P2P, hay còn gọi là [mạng ngang hàng](https://coin98.net/p2p-la-gi) (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 giao thức HTTP.

A picture containing text, clock

Description automatically generated

Hình 3. Mô tả IPFS

IPFS (Interplanetary File System)

Nói cách khác, IPFS là mạng lưới chuyển phát nội dung hoàn toàn phi tập trung cho phép quản lý và lưu trữ dữ liệu một cách linh hoạt. Mỗi máy tính tham gia trong mạng lưới đảm nhận nhiệm vụ download và upload dữ liệu mà không cần sự can thiệp của máy chủ trung tâm.

#### 

### Cách hoạt động của IFPS

Đầu tiên mọi dữ liệu sẽ được mã hoá và được lưu dưới dạng [mã hash](https://coin98.net/hashrate-la-gi) (còn gọi là đối tượng IPFS). Ý tưởng chủ đạo là nếu trình duyệt của bạn muốn truy cập một trang nào đó trên IPFS thì chỉ cần đưa ra mã hash rồi mạng sẽ tìm máy có lưu trữ dữ liệu khớp với mã hash và sau đó tải dữ liệu, trang đó về từ máy tính đấy về cho bạn.

Text

Description automatically generated with medium confidence

Hình 3. Cách hoạt động IPFS

Cách hoạt động của IPFS

Cách thức hoạt động của IPFS sẽ tương tự như Bittorent 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. Tổng quan, cách hoạt động của IPFS sẽ có 2 phần chính:

Xác định tệp có địa chỉ nội dung (giá trị hash của tệp đó).

Tìm dữ liệu được lưu trữ và tải xuống: khi bạn có đoạn hash của file hay trang cần tải, mạng sẽ tìm và connect tới máy tốt nhất để tải dữ liệu xuống cho bạn.

## INFURA

### Infura là gì

Infura là một dự án cung cấp dịch vụ hỗ trợ [Web3](https://coin98.net/web3) tập trung vào Ethereum. Infura cung cấp các công cụ và cơ sở hạ tầng cho phép các nhà phát triển dễ dàng kết nối ứng dụng của họ với các blockchain nền tảng như Ethereum.

Diagram

Description automatically generated

Hình 4. Mô tả Infura

Về cơ bản, Infura cung cấp các công cụ cần thiết để bất kỳ ứng dụng nào bắt đầu phát triển trên Ethereum ngay lập tức mà không cần phải tự chạy cơ sở hạ tầng phức tạp. Sản phẩm chính của Infura cung cấp bao gồm bộ phát triển blockchain cung cấp các giao diện lập trình ứng dụng (API) và các công cụ dành cho nhà phát triển trên Ethereum.

Sử dụng API Infura Ethereum, các nhà xây dựng có thể kết nối các ứng dụng chỉ trong vài giây bằng cách sử dụng một vài dòng code. Điều này làm cho việc xây dựng [Dapp](https://coin98.net/dapp-la-gi) trên ethereum trở nên dễ dàng hơn. Hiện tại, trung bình mỗi ngày cơ sở hạ tầng của Infura xử lý 6.5 tỷ requests JSON-RPC trên mạng Ethereum, khiến Infura trở thành một trong những trụ cột thiết yếu của Ethereum.

## Cấu trúc của một ứng dụng Web 3 đươc sử dụng trong luận văn

Diagram

Description automatically generated

Hình 4. Mô tả về Web3.0

Mọi Ethereum client (tức là provider) phải triển khai JSON-RPC (là một giao thức gọi thủ tục từ xa được mã hóa bằng JSON).

Một cách hiểu đơn giản hơn, có thể hình dung nó giống như mà một gateway để giúp các nhà phát triển, các ứng dụng Web3 và các Crypto Wallet có thể giao tiếp dễ dàng hơn với các hệ thống Blockchain từ xa.

Với RPC, chúng ta có thể truy vấn thông tin liên quan đến blockchain (chẳng hạn như block number, blocks,…), cho phép chúng ta đọc dữ liệu blockchain và gửi các giao dịch đến các mạng.

Tất cả các dApp cần một cách để giao tiếp với các blockchains. Nếu không có phương tiện giao tiếp, các dApp sẽ không thể truy cập thông tin và thực hiện các giao dịch trên blockchain mà chúng hoạt động trên đó.

RPC hỗ trợ các dịch vụ như MetaMask và các ứng dụng Web 3.0 khác kết nối với thông tin trên các blockchains khác nhau. Với RPC, chúng có thể tự động tương tác với dữ liệu blockchain và thực hiện các tác vụ như giao dịch ví tiền điện tử. Nó cũng cho phép bất kỳ ai kết nối với blockchain node để tạo các ứng dụng phi tập trung (dapp).

## Các tính năng trong hệ thống lưu trữ trong blockchain:

### Yêu cầu lưu trữ

### Thiết kế contract lưu dữ liệu

##### Struct Prescripte: Lưu thông tin toa thuốc

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| STT | Tên thuộc tính | Kiểu dữ liệu | Ràng buộc | Ý nghĩa |
| 1 | id | string |  | Mã định danh của toa thuốc |
| 2 | result | string |  | Chẩn đoán cuối cùng của bác sĩ |
| 3 | Result\_detail | string |  | Chi tiết về kết quả |
| 4 | Medicne | String[][] |  | Đơn thuốc |

Bảng 4. Lưu thông tin toa thuốc

##### Struct Note: Lưu thông tin bệnh án

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| STT | Tên thuộc tính | Kiểu dữ liệu | Ràng buộc | Ý nghĩa |
| 1 | id | string |  | Mã định danh của bệnh án |
| 2 | desciption | string |  | Chẩn đoán cuối cùng của bác sĩ |
| 3 | symptoms | String[] |  | Chi tiết về kết quả |

Bảng 4. Lưu thông tin bệnh án

##### Struct Reciept: Lưu thông tin hóa đơn

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| STT | Tên thuộc tính | Kiểu dữ liệu | Ràng buộc | Ý nghĩa |
| 1 | account | string |  | Tài khoản thanh toán |
| 2 | idAppoitment | string |  | Mã cuộc hẹn |

Bảng 4. Lưu thông tin hóa đơn

##### Struct Pay: Lưu thông tin thanh toán

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| STT | Tên thuộc tính | Kiểu dữ liệu | Ràng buộc | Ý nghĩa |
| 1 | idAppoitment | string |  | Mã định danh cuộc hẹn |
| 2 | amout | string |  | Lượng tiền |
| 3 | creatTime | unit |  | Đã tạo khi nào |
| 4 | withdraw | bool |  | Đã rút tiền chưa |

Bảng 4. Lưu thông tin thanh toán

# WebRTC VÀ SocketIO

## Giới thiêu

Trong chương này to

## WebRTC

### Giới thiêu

Dựa trên đặc điểm của dữ liệu và yêu cầu của dịch vụ, các dịch vụ trên internet được chia làm hai loại chính:

* Dịch vụ thời gian thực (Real Time Service): cung cấp voice, video
* Dịch vụ phi thời gian thực(Non Real Time Service): thường dùng chỉ để truyền dẫn dữ liệu thông thường

Các ứng dung dịch vụ thời gian thực nổi tiếng có thể kể đến là Skype, Webex, Sopcast ... và đa số các ứng này hoặc là độc lập hoặc sử dụng một extension của Web, và chưa thể tích hợp hoàn toàn với Web. Tuy nhiên, điều này đã dần thay đổi khi Google giới thiệu Google Hangout, một ứng thời gian thực dựa trên nền web có thể hoạt động tốt trên một trình duyệt web (Google Chrome). Trong bài blog này, mình sẽ giới thiệu sơ lược về WebRTC từ cấu trúc, các thức hoạt động và một số API chính của WebRTC.

### WebRTC là gì

WebRTC là các API viết bằng javascript giúp giao tiếp theo thời gian thực mà không cần cài plugin hay phần mềm hỗ trợ. WebRTC có khả năng hỗ trợ trình duyệt giao tiếp thời gian thực thông qua Video Call, Voice Call hay transfer data P2P(peer-to-peer), không cần đến plugin, phần mềm khác. Vậy ý nghĩa viết tắt của chữ **WebRTC là gì?**

WebRTC là tập hợp các API và được viết bằng javascript

WebRTC là viết tắt của cụm từ **Web Real-Time Communication** rất được các lập trình viên ưa chuộng. **WebRTC** cho phép các trình duyệt giao tiếp với nhau theo thời gian thực .Ví dụ như: gọi điện, video, chơi game,… Ngoài ra, WebRTC là một sản phẩm của World Wide Web Consortium (W3C).

### Cấu trúc và cách thức hoạt động của WebRTC

Về cơ bản, dữ liệu được truyền tải trên WebRTC là dữ liệu voice và video, tuy nhiên khác với các ứng dụng video stream điển hình như Youtube hay Vimeo là các định dạng video đã trải qua chế biến để có thể giảm bớt dụng lượng, các dữ liệu này phải được nhận và gửi trực tiếp từ các Media devices (Microphone, Webcam). Để có thể thực hiện điều này, một số trình duyệt Web đã thêm vào các module hỗ trợ multimedia, điển hình là Google Chrome, Moliza Firefox, Opera và một số trình duyệt khác trên thiết bị di động, chi tiết có thể tham khảo thêm tại [CanIUse](http://caniuse.com/#feat=rtcpeerconnection) .

Diagram

Description automatically generated

Hình 5. Cấu trúc của WebRTC

Để có thể tích hợp các ứng dụng thời gian thực vào web, các browser phải thêm vào các khối chức năng hỗ trợ dành cho WebRTC:

* Voice/video engine: Thu nhận âm thanh và hình ảnh từ thiết bị ngoại vi (microphone, camera), điều chế mã hoá âm thanh và hình ảnh dựa trên các chuẩn mã hoá cơ bản trước khi truyền. Các chuẩn mã hoá cơ bản dành cho voice và video bao gồm: Opus, iSac, iLBC <voice>; VP8, H263, H264 (video).
* Transport: cũng cấp chức năng kết nối với các thành phần khác cùng tham gia trong WebRTC (STUN, TURN, ICE ...)
* Session management: đóng vai trò điều khiển hoạt động của ứng dụng
* Application Programmable Interface - API: cung cấp các hàm cơ bản để các nhà phát triển có thể sử dụng. API ở đây có thể xét nằm trên 2 mức cơ bản: API dành cho Browser developer và API dành cho Front end developer.

Để có thể trải nghiêm tốt nhất WebRTC, chúng ta nên sử dụng trình duyệt Google Chrome khi mà trình duyệt này cung cấp gần như đầy đủ các API hỗ trợ dành cho WebRTC

* microphone của người dùng
* peerConnection: gửi và nhận dữ liệu hình ảnh, giọng nói
* dataChannels: gửi và nhận dữ liệu không phải là hình ảnh, giọng nói giữa ứng dụng/trình duyệt

Nói tóm lại, **WebRTC** có thể được sử dụng cho nhiều mục đích, từ việc truyền tải video, âm thanh cho đến gửi dữ liệu theo thời gian thực giữa hai hoặc nhiều thiết bị với nhau mà không nhất thiết phải đi qua server trung gian. Điều này giúp giảm độ trễ trong việc truyền tải, giảm độ phức tạp khi phát triển ứng dụng cũng như giảm chi phí vận hành (vì không phải trả tiền thuê server, tiền điện, tiền bảo dưỡng...), kéo theo đó giá bán dịch vụ nếu có thì cũng sẽ thấp hơn.

## SocketIO

### SocketIO là gì

Việc giao tiếp 2 chiều giữa máy khách và máy chủ được thực hiện bởi socket io khi và chỉ khi máy khách có module này trong trình duyệt và máy chủ cũng đã tích hợp sẵn gói socket io. Các ứng dụng sử dụng socket io thường đòi hỏi tốc độ phản hồi ngay lập tức. Một số ví dụ điển hình như xổ số, trực tiếp bóng đá, chat…

## Mô tả sự kết hợp giữa WebRTC và SocketIO để tạo videocall

##### SDP là gì

SDP, từ viết tắt của[Session Description Protocol](https://teleworking.vn/san-pham/g-on-truy-cap-may-tinh-tu-xa/) (Giao thức Mô tả), một giao thức xác định định dạng dựa trên văn bản để mô tả các phiên phương tiện truyền trực tuyến và truyền phát đa hướng. SDP không phải là một giao thức truyền tải mà là một phương pháp mô tả các chi tiết của quá trình truyền.

Ví dụ: tệp SDP chứa thông tin về định dạng, thời gian và quyền tác giả của việc truyền, tên và mục đích của phiên, mọi phương tiện, giao thức hoặc định dạng codec, số phiên bản, thông tin liên hệ và thời gian phát sóng.

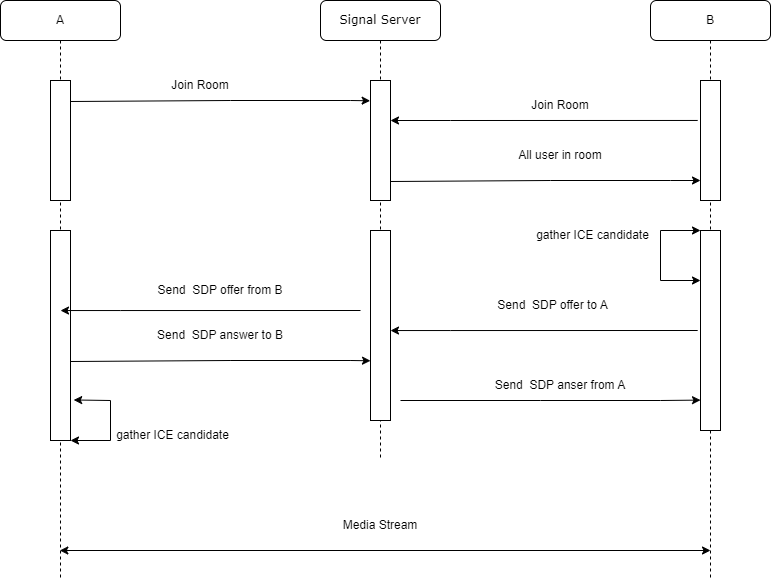
##### ICE là gì

ICE nôm na dễ hiểu là một giao thức được cùng để thiết lập phiên media dựa trên UDP đi qua NAT một cách nhanh nhất.

ICE sẽ tìm đường tốt nhất để kết nối giữa các peer, nó thử tất cả khả năng có thể kết nối một cách song song và lựa chọn con đường hiệu quả nhất (cướp ngân hàng làm giàu).

Đầu tiên nó sẽ cố gắng tạo ra một kết nối bằng cách sử dụng địa chỉ thu được từ hệ điều hành và card mạng của thiết bị, nếu không thành công (có thể thiết bị đằng sau NAT) thì ICE sẽ lấy địa chỉ bên ngoài của thiết bị bằng cách sử dụng máy chủ STUN (nhưng đời có lúc không gặp may), nếu không thành công nữa thì nó sẽ chuyển lưu lượng mạng qua một máy chủ chuyển tiếp là TURN.

##### Mô tả cách hoạt động



Hình 5. Mô tả cách hoạt động của video call

Từ sơ đồ tuần tự trên ta thấy được khái quát cách hoạt dộng của việc tạo videocall diễn như thế nào dưới đây tôi sẽ trình bày một cách chi tết về cách hoạt động và sử dụng các công nghệ đã nêu ở trên:

* User muốn tham gia vào phòng chat thì sẽ gửi thông tin của mình qua socketIO với những thông tin như thông tin user A, mã phòng, UID của SocketIO. Với mỗi uid của socketIO tương ứng là địa chỉ của một user… Nếu chưa có user tham gia thì user A phải đợi, và ngược lại đối với user B
* Nếu trong phòng đã có người thì server sẽ gửi tất cả các thông tin của tất cả user trong phòng về cho user uid socket của user đó.
* Sau khi có tất cả các uid trong phòng thì user cuối sẽ gửi SDP với thể loại là “offer” cho tất cả các user trong phòng thông qua SocketIO và sau đó sẽ thu thập các ICE của user khác
* Sau khi nhận được SDP của người thì user nhận sẽ trả lời lại bằng SDP với thể loại “answer” và sau đó thu thập các ICE của các user khác.
* Nếu như các ICE có thể tìm thấy được trong thì quá trình streaming video sẽ được bắt đầu

# HỌC MÁY, CÁC MÔ HÌNH DỰ ĐOÁN BỆNH

## Giới thiệu

Mục đích chương này là đưa ra dự đoán về căn bệnh chung và phổ biến . Điều này cho phép người dùng chia sẻ các triệu chứng liên quan đến sức khỏe của họ. Ở đây chúng tôi sử dụng một số kỹ thuật phân tích dữ liệu để dự đoán liệu một người có mắc bệnh hay không.

## Học máy

### Học máy là gì

Trong những năm gần đây, có rất nhiều thành tựu trong Machine Learning. Học máy đã trở thành một trong những ngành quan trọng nhất và góp phần vào sự phát triển của con người trong cuộc Cách mạng công nghiệp lần thứ tư.

Học máy (ML) là nghiên cứu về các thuật toán máy tính tự động cải thiện thông qua kinh nghiệm. Nó được coi là một tập hợp con của trí tuệ nhân tạo. Các thuật toán học máy xây dựng một mô hình dựa trên dữ liệu mẫu, được gọi là "dữ liệu huấn luyện", để đưa ra dự đoán hoặc quyết định mà không được lập trình rõ ràng để làm như vậy. Có ba loại kỹ thuật Machine Learning:

* Học có giám sát
* Học không giám sát
* Học tăng cường

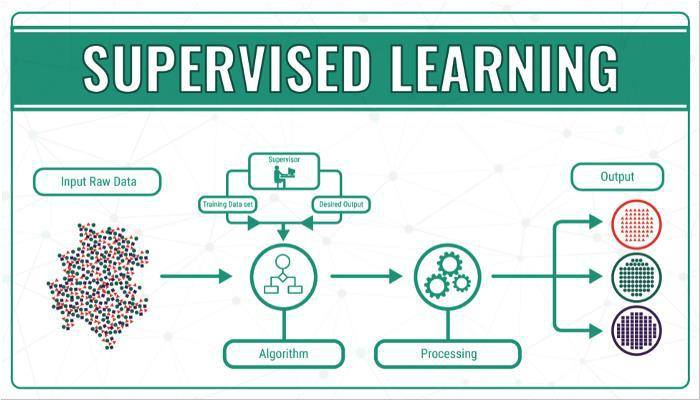
### Học có giám sát

Hầu hết Machine Learning thực tế đều sử dụng phương pháp học có giám sát. Nó được xác định bằng cách sử dụng các bộ dữ liệu được gắn nhãn để huấn luyện các thuật toán phân loại dữ liệu hoặc dự đoán kết quả một cách chính xác. Khi dữ liệu đầu vào được đưa vào mô hình, nó sẽ điều chỉnh trọng số của nó thông qua quá trình học tăng cường, đảm bảo rằng mô hình đã được trang bị phù hợp.

Học có giám sát có thể được tách thành hai loại vấn đề khi khai thác dữ liệu-phân loại và hồi quy:

• Phân loại sử dụng một thuật toán để gán chính xác dữ liệu thử nghiệm vào các danh mục cụ thể. Nó nhận ra các thực thể cụ thể trong tập dữ liệu và cố gắng kết luận cách các thực thể đó nên được gắn nhãn hoặc xác định. Các thuật toán phân loại thường là phân loại tuyến tính, máy vectơ hỗ trợ (SVM),cây quyết định, k-hàng xóm gần nhất và rừng ngẫu nhiên ...

• Hồi quy được sử dụng để hiểu mối quan hệ giữa các biến phụ thuộc và biến độc lập. Nó thường được sử dụng để đưa ra các dự đoán, chẳng hạn như doanh thu bán hàng cho một doanh nghiệp nhất định. Hồi quy tuyến tính, hồi quy logistic và hồi quy đa thức là các thuật toán hồi quy phổ biến.



Bảng 6. Học có giám sát

Hình 6. Học có giám sát

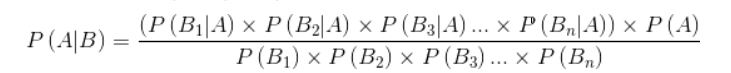
### Thuật toán Naïve Bayes

#### Định lý Naïve Bayes

Hình 6. Định lý Bayes

Bảng 6. Định lý Bayes

Nó cho chúng ta biết tần suất của A xảy ra dựa trên điều kiện là B đã xảy ra, viết tắt là P(A|B), khi mà chúng ta biết tần suất B xảy ra với điều kiện A đã xảy ra P(B|A), xác suất A xảy ra P(A), B xảy ra P(B).

– P(A|B) là xác suất xảy ra của một sự kiện ngẫu nhiên A khi biết sự kiện liên quan B đã xảy ra.  
– P(B|A) là xác suất xảy ra B khi biết A xảy ra  
– P(A) là xác suất sảy ra của riêng A mà không quan tâm đến B.  
– P(B) là xác suất xảy ra của riêng B mà không quan tâm đến A.

Bảng 6. Định lý bayes mở rộng

Hình 6. Định lý bayes mở rộng

#### Thuật toán Naïve Bayes là gì

Đây là một kỹ thuật phân loại dựa trên Định lý Bayes với giả định về sự độc lập giữa các yếu tố dự đoán. Nói một cách dễ hiểu, bộ phân loại Naive Bayes giả định rằng sự hiện diện của một đối tượng cụ thể trong một lớp không liên quan đến sự hiện diện của bất kỳ đối tượng địa lý nào khác.

#### Ứng dụng của thuật toán Naïve Bayes

Thuật toán Naive Bayes Classification được áp dụng vào các loại ứng dụng sau

- Real time Prediction: NBC chạy khá nhanh nên nó thích hợp áp dụng ứng dụng nhiều vào các ứng dụng chạy thời gian thực, như hệ thống cảnh báo phát hiện sự cố...

- Multi class Prediction: Nhờ vào định lý Bayes mở rộng ta có thể ứng dụng vào các loại ứng dụng đa dự đoán, tức là ứng dụng có thể dự đoán nhiều giả thuyết mục tiêu.

- Text classification/ Spam Filtering/ Sentiment Analysis: NBC cũng rất thích hợp cho các hệ thống phân loại văn bản hay ngôn ngữ tự nhiên vì tính chính xác của nó lớn hơn các thuật toán khác. Ngoài ra các hệ thống chống thư rác cũng rất ưu chuộng thuật toán này. Và các hệ thống phân tích tâm lý thị trường cũng áp dụng NBC để tiến hành phân tích tâm lý người dùng ưu chuộng hay không ưu chuộng các loại sản phẩm nào từ việc phân tích các thói quen và hành động của khách hàng.

- Recommendation System: Naive Bayes Classifier được sử dụng rất nhiều để xây dựng hệ thống gợi ý.

## Mô tả dữ liệu và kết quả dự đoán

- Dữ liệu sau khi làm sạch

* Nguồn dữ liệu: Kaggle
* Tổng số triệu chứng: 132
* Tổng số bệnh: 40
* Tổng số lượng bệnh từ triệu chứng:304
* Nhãn của triệu chứng là 0 và 1
* Dữ liệu đã được làm sạch
* Đầu vào: Triệu chứng bệnh
* Đầu ra: Triệu chứng bênh
* Mô hình học máy dự đoán :Naïve Bayes
* Độ chính xác sau khi đánh giá : 0.9286%

# KẾT LUẬN

## Kết quả đạt được

Trong thời gian tìm hiểu, nghiên cứu các cơ sở lý thuyết và triển khai các công nghệ em đã đạt được những kết quả sau:

- Về mặt lý thuyết:

Nắm rõ được các công nghệ đã sử dụng trong luận văn này và hiểu rõ hơn về các Frameword đã sử dụng trong luận văn này, tiếp cận và học hỏi được nhiều những công nghệ mới như blockchain,…

- Về mặt thực tiễn:

* + Về cơ bản đã xây dựng và triển khai đầy đủ các chức năng ban đầu đã đề ra, cơ bản đã hoàn thành đúng như tiến độ ban đầu đã đề ra
  + Tạo được cuộc trò chuyện video call cho bác sĩ và bệnh nhân tương đối trơn chu
  + Giao diện đơn giản
  + Kết nối được, sử dụng được công nghệ blockchain , lưu trữ và thanh toán được thông qua công nghệ blockchain

## Những vấn đề chưa giải quyết

Bên cạnh những mặt đã đạt được hệ thống vẫn còn một số nhược điểm cần phải được cải thiện như sau:

* + Giao diện hệ thống chưa đẹp mắt
  + Tốc độ truy cập cũng như xử lý còn chậm
  + Một số chức năng vẫn chưa hoàn thiện
  + Cách tạo phòng cho bác sĩ và bệnh nhân để video call vẫn còn chưa ổn định
  + Cần có nhiều phương thức thanh toán khác
  + Chưa đáp ứng yêu cầu trong thực tế

## Hướng phát triển

Với những mặt còn hạn chế đã nêu trên, hệ thống cần có những cãi thiện trong thời gian tới như sau:

* + Cần có một giao diện đẹp hơn bố trí các chức năng hợp lý
  + Tăng tốc độ truy cập của hệ thống
  + Hoàn thành hết các chức năng còn thiếu trong hệ thống
  + Cải thiện và xử dụng nhiều thuật toán hơn để so sánh tốc độ dự đoán bệnh

## Khó khăn và thuận lợi

* Thuận lợi:
  + Có các công cụ miễn phí để xây dựng hệ thống
  + Nhờ các kiến thức đã học xây dựng lên một cách dễ dàng
* Khó khăn:
* Chưa có nhiều kinh nghiệm làm việc theo mô hình phát triển phần mềm nên còn gặp khá nhiều khó khăn và trở ngại
* Chưa tiếp xúc với ứng dụng và môi trường thực tế nên hệ thống chưa đáp ứng yêu cầu trong thực tế

## Một số hình ánh của hệ thống

# 

Hình 7. Giao diện chính

Graphical user interface, application, Teams

Description automatically generated

Hình 7. Giao diện người dùng

Graphical user interface, application

Description automatically generated

Hình 7. Giao diện quản lý thông tin

Graphical user interface, application, email

Description automatically generated

Hình 7. Giao diện dự đoán bệnh

Graphical user interface, website

Description automatically generated

Hình 7. Giao diện thanh toán

Graphical user interface, text, application

Description automatically generated

Hình 7. Giao diện lịch đặt cuộc hẹn

Text

Description automatically generated with medium confidence

Hình 7. Giao diện đơn thuốc



Hình 7. Giao diện cuộc gặp video call

Graphical user interface, text, application, email

Description automatically generated

Hình 7. Giao diện tìm kiếm bác sĩ

Graphical user interface, text, application, email

Description automatically generated

Hình 7. Giao diện xem thông tin bác sĩ

Graphical user interface, application

Description automatically generated

Hình 7. Giao diện rút tiền

Graphical user interface, text, application

Description automatically generated

Hình 7. Giao diện đặt lịch hẹn

**DANH MỤC THAM KHẢO**

[1] <https://wiki.tino.org/blockchain-la-gi/>

[2] <https://coin98.net/ethereum-la-gi>

[3] <https://webrtc.org/getting-started/overview>

[4] [https://bizflycloud.vn/tin-tuc/web-30-la-gi-tim-hieu-chi-tiet-ve-web-30-ky- nguyen-moi-cua-internet-phan-1-20220316164228356.htm](https://bizflycloud.vn/tin-tuc/web-30-la-gi-tim-hieu-chi-tiet-ve-web-30-ky-%20%20nguyen-moi-cua-internet-phan-1-20220316164228356.htm)

[5] <https://soliditytips.com/articles/detect-switch-chain-metamask/>

[6] [ttps://developer.mozilla.org/en-US/docs/Web/API/MediaDevices/getUserMedia](https://developer.mozilla.org/en-US/docs/Web/API/MediaDevices/getUserMedia)

[7] <https://www.theengineeringprojects.com/2021/06/structure-of-a-block-in-blockchain.html>

[8] <https://mlsdev.com/blog/156-how-to-build-your-own-blockchain-architecture>

[9] <https://onetech.vn/blog/nhung-ung-dung-blockchain-noi-bat-nhat-trong-thuc-te-6099>