**ĐẠI HỌC QUỐC GIA HÀ NỘI**

**TRƯỜNG ĐẠI HỌC CÔNG NGHỆ**



**Nguyễn Văn Khỏe**

XÂY DỰNG HỆ THỐNG THƯƠNG MẠI ĐIỆN TỬ DỰA TRÊN NỀN TẢNG BLOCKCHAIN

**ĐỒ ÁN TỐT NGHIỆP ĐẠI HỌC HỆ CHÍNH QUY**

**Ngành: Truyền thông và mạng máy tính**

* + 1. **HÀ NỘI - 2018**

**ĐẠI HỌC QUỐC GIA HÀ NỘI**

**TRƯỜNG ĐẠI HỌC CÔNG NGHỆ**

**Nguyễn Văn Khỏe**

**XÂY DỰNG HỆ THỐNG THƯƠNG MẠI ĐIỆN TỬ DỰA TRÊN NỀN TẢNG BLOCKCHAIN**

**ĐỒ ÁN TỐT NGHIỆP ĐẠI HỌC HỆ CHÍNH QUY**

**Ngành: Truyền thông và mạng máy tính**

**Cán bộ hướng dẫn: TS. Hoàng Xuân Tùng, TS. Nguyễn Đình Hóa**

**HÀ NỘI - 2018**

**LỜI CAM ĐOAN**

Tôi xin cam đoan những nội dung trong đồ án dưới đây là do chính em thực hiện dưới sự hướng dẫn trực tiếp của TS. Hoàng Xuân Tùng và T.S Nguyễn Đình Hóa. Tất cả tài liệu tham khảo nghiên cứu liên quan đều được ghi nguồn trích dẫn rõ ràng. Nếu có gì sai em xin hoàn toàn chịu trách nhiệm.

Hà Nội, ngày 15 tháng 10 năm 2018

Sinh viên

NGUYỄN VĂN KHỎE

**LỜI CẢM ƠN**

Em xin gửi lời cảm ơn sâu sắc của mình tới thầy **Hoàng Xuân Tùng** và thầy **Nguyễn Đình Hóa** thuộc **bộ môn Truyền thông và Mạng máy tính, trường Đại Học Công Nghệ, Đại Học Quốc Gia Hà Nội**, những người đã trực tiếp hướng dẫn em trong suốt quá trình nghiên cứu, hoàn thành đồ án.

Em cũng xin chân thành cảm ơn các Thầy Cô trong khoa Công Nghệ Thông Tin nói riêng và các thầy cô trong trường Đại Học Công Nghệ - Đại Học Quốc Gia Hà Nội nói chung đã dạy dỗ và tận tình chỉ bảo em trong suốt quá trình học tập tại trường.

Cuối cùng em xin gửi lời cảm ơn tới gia đình, bạn bè đã luôn động viên ủng hộ giúp đỡ em để có thể hoàn thành đồ án này.

Một lần nữa, em xin chân thành cảm ơn.

Hà Nội, ngày 15 tháng 10 năm 2018

Sinh viên

NGUYỄN VĂN KHỎE

Mục lục

[Chương 1. Đặt vấn đề 7](#_Toc528196314)

[1.1. Thực tiễn thương mại điện tử, bối cảnh liên quan 7](#_Toc528196315)

[1.2. Tầm quan trọng 7](#_Toc528196316)

[1.3. Mục tiêu nghiên cứu 8](#_Toc528196317)

[Chương 2. Cơ sở lý thuyết (30%) 8](#_Toc528196318)

[2.1. Blockchain 8](#_Toc528196319)

[2.1.1. Định nghĩa 8](#_Toc528196320)

[2.1.2. Đặc tính 9](#_Toc528196321)

[2.1.3. Ứng dụng Blockchain 9](#_Toc528196322)

[2.1.4. Ưu, nhược điểm blockchain với các công nghệ trước đó 9](#_Toc528196323)

[2.2. Hyperledger 9](#_Toc528196324)

[2.2.1. Hyperledger Fabric 9](#_Toc528196325)

[2.2.2. Hyperledger Composer 9](#_Toc528196326)

[2.3. Một số công nghệ ứng dụng 10](#_Toc528196327)

[2.3.1. OAUTH 10](#_Toc528196328)

[2.3.2. Restful API 10](#_Toc528196329)

[2.3.3. AngularJs 10](#_Toc528196330)

[2.3.4. Angular typescript 10](#_Toc528196331)

[2.3.5. Docker 10](#_Toc528196332)

[2.4. Tổng kết chương 10](#_Toc528196333)

[Chương 3. Phân tích thiết kế bài toán xây dựng hệ thống thương mại điện tử dựa trên nền tảng blockchain 10](#_Toc528196334)

[3.1. Tại sao nên xây dựng hệ thống TMĐT ứng dụng công nghệ Blockchain? 10](#_Toc528196335)

[3.1.1. TMĐT truyền thống 10](#_Toc528196336)

[3.1.2. TMĐT ứng dụng blockchain 12](#_Toc528196337)

[3.1.3. So sánh chi tiết 12](#_Toc528196338)

[3.2. Mô tả, xây dựng hệ thống TMĐT ứng dụng Blockchain 14](#_Toc528196339)

[3.2.1. Mô tả chương trình. 14](#_Toc528196340)

[3.2.2. Cấu trúc hệ thống 14](#_Toc528196341)

[3.2.3. Phân tích thiết kế 15](#_Toc528196342)

[Chương 4. Demo (20%) 18](#_Toc528196343)

[4.1. Trình bày thuật toán 18](#_Toc528196344)

[4.2. Quy trình chạy, khởi tạo server 18](#_Toc528196345)

[4.3. Hiển thị kết quả 18](#_Toc528196346)

[Chương 5. Kết luận (10%) 18](#_Toc528196347)

[5.1. Khái quát toàn bộ nội dung đồ án 18](#_Toc528196348)

[5.2. Nhận xét, đánh giá 18](#_Toc528196349)

[5.2.1. Những vấn đề còn tồn tại 18](#_Toc528196350)

[5.2.2. Định hướng phát triển 18](#_Toc528196351)

[Tài liệu tham khảo 18](#_Toc528196352)

**BẢNG CÁC TỪ KHÓA VIẾT TẮT**

|  |  |
| --- | --- |
| **Từ viết tắt** | **Thuật ngữ đầy đủ** |
| TMĐT | Thương mại điện tử |
| HC | Hyperledger Composer |
| HF | Hyperledger Fabric |
| BNA | Mạng lưu trữ thông tin doanh nghiệp |
| Assets | Tài sản, hàng hóa |
| Participants | Các bên liên quan |
| Transactions | Các giao dịchs |
| CSDL | Cơ sở dữ liệu |
| JSON | JavaScript Object Notation |

**DANH MỤC HÌNH ẢNH**

[Figure 1: Cấu trúc business network archive 10](#_Toc528196353)

[Figure 2: Mô hình B2C trong TMĐT truyền thống 11](#_Toc528196354)

[Figure 3: Mô hình TMĐT ứng dụng công nghệ Blockchain 12](#_Toc528196355)

[Figure 4: Cấu trúc hệ thống của ứng dụng 14](#_Toc528196356)

[Figure 5: Sơ đồ mô tả cơ chế xác thực người dùng qua github 16](#_Toc528196357)

[Figure 6: Sơ đồ mô tả cơ sở dữ liệu hệ thống 16](#_Toc528196358)

[Figure 7: Hình vẽ minh họa world-state database 17](#_Toc528196359)

[Figure 8: Hình ảnh minh họa Blockchain database 17](#_Toc528196360)

# Đặt vấn đề

## Thực tiễn thương mại điện tử, bối cảnh liên quan

Thương mại điện tử (E-commerce) là một hình thức kinh doanh thương mại trên cơ sở mạng máy tính toàn cầu. Nó được dự báo là phương thức hoạt động chủ yếu trong nền kinh tế số. Thương mại điện tử trong thời gian gần đây đã được các nước quan tâm và phát triển [3].

Thương mại điện tử chính là nguồn tài nguyên khổng lồ, không những nó không thể bị cạn kiệt như các loại tài nguyên thiên nhiên mà trong kỷ nguyên công nghệ thông tin và viễn thông phát triển như vũ bão, nó ngày càng phong phú và phát triển hơn.

Theo thống kê của Cục viễn thông (Bộ thông tin và truyền thông), tính tới thời điểm tháng 10/2017, cả nước có gần 118 triệu thuê bao di động, trong đó số người sử dụng smartphone chiếm tới 84%. TMĐT tại khu vực ASEAN có tốc độ tăng trưởng internet khoảng 14%/năm. Thống kê của Nielsen cũng cho thấy trung bình mỗi người sử dụng Internet tại Việt Nam chi 160 USD/năm cho TMĐT. Nhờ đó, tốc độc tăng trưởng thương mại điện tử tại Việt Nam tăng khoảng 22%/năm và triển vọng có thể lên tới 30-50%/năm trong giai đoạn 2020-2025 [9].

Với thương mại điện tử mọi thông tin về sản phẩm, khách hàng, doanh nghiệp đều được thường xuyên cập nhập, tạo điều kiện cho khách hàng nắm được những thông tin mới nhất. Việc trao đổi, liên lạc giữa các bên trở lên nhanh chóng, tiết kiệm thời gian, tiền bạc.

## Tầm quan trọng

Trong sự phát triển kinh tế, thương mại điện tử Việt Nam chiếm vai trò vô cùng quan trọng.

TMĐT tạo ra sự thay đổi trong hệ thống sản xuất, như từ mô hình sản xuất hàng loạt sang mô hình sản xuất theo thị trường. Dây chuyền sản xuất còn hợp nhất với các bộ phận tài chính, tiếp thị và bộ phận kinh doanh khác. Ví dụ, trường hợp của công ty Intel, việc sử dụng trang web ERP, các đơn đặt hàng từ phía khách hàng được chuyển tiếp đến người sử dụng hệ thống thiết kế được hỗ trợ bởi máy tính và được chuyển tới dây chuyền sản xuất chỉ trong vài giây [4]. Còn rất nhiều các trường hợp tương tự khác, việc trao đổi thông tin kịp thời trở thành yếu tố quan trọng trong hệ thống liên kết. Giúp những hệ thống này trở lên linh hoạt và dễ thích ứng với sự thay đổi của nhu cầu thị trường. Tối thiểu hóa những dư thừa, tồn đọng trong hệ thống sản xuất.

Thương mại điện tử giúp người tiêu dùng dễ dàng hơn trong việc chọn lựa sản phẩm, dịch vụ. Đây là một kênh phân phối lớn, kênh thông tin bổ sung quan trọng giúp người tiêu dùng dễ tiếp cận sản phẩm, dịch vụ. Chỉ với một chiếc máy tính hay một chiếc smart-phone có kết nối Internet, truy cập vào các sàn giao dịch TMĐT trực tuyến, khách hàng có thể nhận được rất nhiều lời mời chào lựa chọn sản phẩm. Các hê thống TMĐT hiện nay còn tích hợp nhiều công nghệ mới như AI, Machine Learning… sử dụng những thuật toán gợi ý sản phẩm mỗi khi người tiêu dùng sử dụng ứng dụng. Giúp người dùng tiết kiệm thời gian tìm kiếm mặt hàng mình quan tâm. Một điểm ưu không thể không nhắc tới trong TMĐT là sự cản trở về không gian địa lý, thời gian, thiếu thông tin đã được giải quyết một cách khá tối ưu thông qua các dịch vụ TMĐT đi kèm hiện nay có thể kể đến như Grab hay Giaohangtietkiem… Chỉ cần đăng ký tài khoản trên hệ thống và tiến hành đặt hàng, đặt lịch và ngồi nhà chờ đợt, sẽ có nhân viên tới giao mặt hàng tới tận tay bạn. Tiết kiệm rất nhiều thời gian và công sức đi lại của người tiêu dùng.

## Mục tiêu nghiên cứu

Với nguồn nhân lực và tiếp thu nhanh công nghệ mới, Việt Nam được coi là một quốc gia năng động với tỷ lệ người dùng Internet luôn luôn tăng và xếp top đầu các nước sử dụng Internet. Tuy vậy trong TMĐT vấn đề thông tin cá nhân còn được xem nhẹ trong ý thức người Việt. Nhiều công ty doanh nghiệp đang thu thập số lượng lớn dữ liệu người dùng cá nhân. Các dữ liệu này thường là các dữ liệu rất nhạy cảm, liên quan tới quyền riêng tư của người tiêu dùng. Các mối lo ngại về bảo mật thông tin cho người dùng trở thành vấn đề lớn cho các doanh nghiệp đó, không ai có thể đảm bảo hệ thống của mình bảo mật hoàn hảo.

Ngoài ra, tại thị trường TMĐT ở Việt Nam, quyền lợi của người tiêu dùng, các đại lý phân phối vẫn xếp bên dưới quyền lợi của doanh nghiệp. Ban đầu, doanh nghiệp tiến hành xây dựng sàn giao dịch của họ, thiết lập các quy định, điều lệ phù hợp với họ nhất. Cũng không thể phủ nhận họ rất coi trọng ý kiến, quyền lợi bên tiêu dùng và bên đại lý phân phối vì đây chính là nguồn lợi nhuận của họ. Tuy nhiên xét về phương diện đại lý phân phối, họ không trực tiếp được phân phối sản phẩm mà phải trung chuyển qua bên thứ ba là doanh nghiệp, kèm theo đó chịu những quy định điều lệ mà bên doanh nghiệp đề ra. Phương diện người tiêu dùng thì không được mua với đúng mức giá chuẩn mà bên đại lý phân phối ban đầu đưa ra…

Trong khuôn khổ đồ án này, em xin đề xuất xây dựng một hệ thống TMĐT mới do em phát triển dựa trên việc sử dụng các ưu điểm của công nghệ Blockchain. Cho phép áp dụng những thuật toán mới giúp cải thiện tính bảo mật thông tin, quyền lợi giữa các bên tiêu dùng, phân phối sản phẩm được đảm bảo một cách tối ưu hơn so với TMĐT hiện nay.

# Cơ sở lý thuyết (30%)

## Blockchain

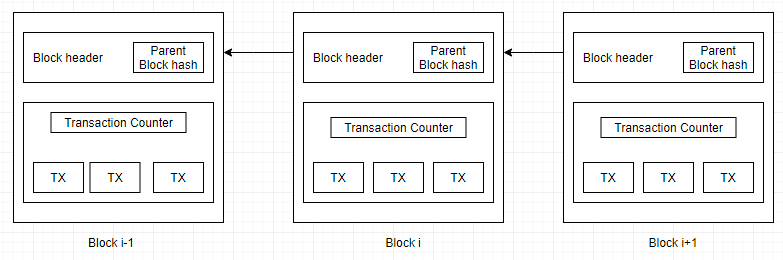
Một mạng blockchain được định nghĩa như một sổ kế toán bất biến (immutable ledger) để ghi lại tất cả các giao dịch, được duy trì trong hệ thống phân phối giữa các bên không tin cậy [1]. Blockchain dựa trên nền tảng công nghệ số kế toán phân phối (DLT). Đưa ra sự đồng thuận bằng cơ chế xác thực thông qua mạng máy tính tạo điều kiện các giao dịch ngang hàng mà không phải thông qua một cơ quan trung gian.

Mỗi bên tham gia mạng blockchain sẽ lưu trữ một bản sao sổ kế toán toàn mạng. Các bên thực hiện giao thức đồng thuận để xác thực giao dịch, nhóm chúng thành một khối và xây dựng các chuỗi băm trên mỗi khối.

### Cấu trúc

#### Khối

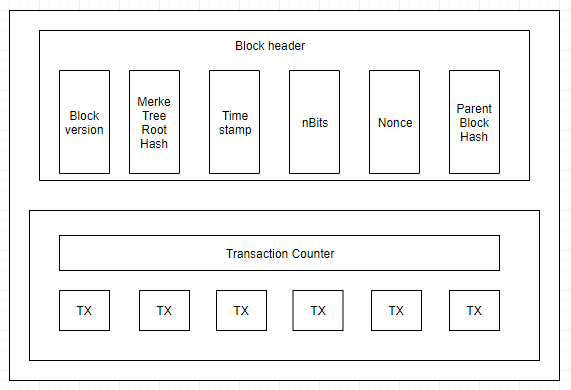
Tên của blockchain xuất phát từ cấu trúc kỹ thuật của nó – một chuỗi các khối.



Hình 1: Một blockchain bao gồm một chuỗi nối tiếp các khối

Trong bài báo “Do you need a Blockchain?”, hai tác giả Karl Wust, Authur Gervak đã viết rằng: Blockchain là một bản ghi giữ một danh sách các phiên giao dịch có vai trò như một sổ kế toán công khai. Mỗi khối được liên kết với khối trước đó bằng mật mã băm [<wust2018you>]. Với mỗi khối băm trước đó trong tiêu đề khối, một khối chỉ có một khối cha. Khối đầu tiên được khởi tạo được gọi là khối gốc hay còn gọi là genesis-block đây là khối duy nhất không có khối cha [<rosenfeld2014analysis>].

Đi sâu chi tiết bên trong một khối:



Hình 2: Cấu trúc của một khối

Một khối thì gồm:

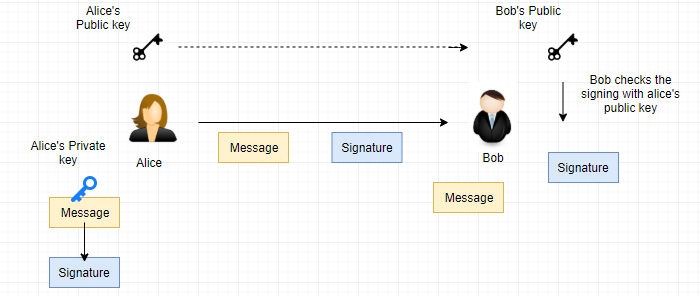
* Tiêu đề khối
* Phiên bản khối: được chia theo quy tắc xác thực riêng.
* Merke tree root hash: Giá trị băm của tất cả giao dịch trong khối.
* Time stamp: thời gian khối được khởi tạo.
* nBits: ngưỡng mục tiêu của một khối băm hợp lệ.
* Nonce: trường 4-byte, thường bắt đầu bằng 0 và tăng cho phép tính băm.
* Mã băm khối cha: giá trị băm 256-bit của khối cha.
* Thân khối
  + Transaction counter: bộ đếm số các giao dịch trong khối. Số lượng các giao dịch phụ thuộc vào kích thước khối và kích thước các giao dịch.
  + Trasactions: các phiên giao dịch xảy ra.

Blockchain sử dụng kỹ thuật mật mã không đối xứng và cơ chế xác thực giao dịch [<prpic2017mechanismcrypto>].

#### Chữ ký số

Mỗi một người dùng sở hữu một cặp khóa riêng và khóa công khai. Khóa riêng để ký các giao dịch, được giữ bí mật.

Ví dụ: Người dùng Alice muốn chuyển một số bitcoin cho người dùng Bob. Alice có một cặp khóa: một khóa riêng và một khóa công khai. Trong giai đoạn ký kết, Alice ký giao dịch này với khóa riêng của mình (trong ví của cô ấy) và sau đó thêm khóa công khai của mình vào blockchain. Ở giai đoạn xác minh, Bob sử dụng khóa công khai của Bob. Theo cách này Bob có thể dễ dàng kiểm tra rằng dữ liệu có bị giả mạo hay không.



Hình 3: Sơ đồ minh họa quá trình xác nhận chữ ký số giữa Alice và Bob

### Đặc điểm

#### Cơ sở dữ liệu phân tán

Theo như Peters, Gareth W and Panayi, Efstathios viết trong cuốn “Banking Beyond Banks and Money” đã chia ra năm lại cơ sở dữ liệu chính trong cơ sở dữ liệu truyền thống:

* Cơ sở dữ liệu quan hệ, như là SQL hay các biến thể, các bảng dữ liệu được triển khai theo mối quan hệ một chiều hay hai chiều chiều.
* Cơ sở dữ liệu khóa-giá trị (key-value), lưu trữ các cặp khóa và giá trị. Khi cần truy xuất dữ liệu thì chỉ cần thông qua khóa.
* Cơ sở dữ liệu cột: dữ liệu được lưu trong các cột, cũng gần giống cơ sở dữ liệu quan hệ tuy nhiên dùng biểu diễn những bảng thưa thớt không có quan hệ với bảng nào.
* Cơ sở dữ liệu tài liệu: Lưu trữ tài liệu.
* Cơ sở dữ liệu bảng biểu, đồ thị: mô hình biểu diễn qua các nút và các mối quan hệ.

Tất cả các loại cơ sở dữ liệu đều thuộc kiểu tập trung trong một server và phân phối dữ liệu trên nhiều trang web được kết nối thông qua mạng máy tính. Các toán tử liên quan tới dữ liệu trong cơ sở dữ liệu truyền thống là tạo mới, thêm, sửa, xóa ( hay còn được gọi là toán tử C.R.U.D). Kiếm soát dữ liệu thuộc về quyền của các quản trị viên hệ thống, cho phép truy cập và thay đổi dữ liệu, duy trì máy chủ tập trung.

Một blockchain có kiến trúc tương tự như một linked-list. Bao gồm các nút phân cấp và mỗi nút đều tham gia vào vai trò quản trị.

Với blockchain, hệ thống cơ sở dữ liệu phân tán cho phép phân vùng truy xuất thông tin lớn hơn, chia nhỏ dữ liệu để dễ dàng xử lý. Các nút kết nối không cần đồng nhất về dữ liệu mà chúng lưu trữ. Do thiết kế của các cơ sở dữ liệu này và việc sao chép dữ liệu trên các nút khác nhau đem lại một số các ưu điểm:

* Độ tin cậy và tính khả dụng cao hơn.
* Cải thiện hiệu suất và thông lượng.
* Dễ dàng mở rộng và phát triển lên.

#### Hợp đồng thông minh

Khái niệm hợp đồng thông minh được Nick Szabo giới thiệu vào năm 1994 và được định nghĩa là một giao thức giao dịch trên máy tính mà thực hiện các điều khoản của hợp đồng [<christidis2016blockchains>].

Szabo đã đưa ra ví dụ về một hợp đồng thông minh thực thi cho hình thức thanh toán vay mua xe. Nếu chủ xe không thanh toán kịp thời hạn, hợp đồng thông minh sẽ thu hồi quyền truy cập vật lý và kiểm soát trở lại của chiếc xe cho ngân hàng [<zhang2016town>]. Về quy trình, ông thiết lập các điều khoản hợp đồng và nhúng chúng vào các thuộc tính của hệ thống để nó tự thực thi chúng. Việc làm như vậy nhằm giảm thiểu bên trung gian đáng tin cậy giữa các bên tham gia giao dịch và sự xuất hiện của các ngoại lệ về tính xác thực của bên trung gian. Tuy nhiên tại thời điểm đó ông vẫn chưa có đủ phương tiện và môi trường tích hợp cần thiết thể thực hiện hóa tất cả. Chỉ đến khi công nghệ Blockchain ra đời, hợp đồng thông minh mới được ứng dụng rộng rãi.

Trong bối cảnh blockchain, các hợp đồng thông minh là các lệnh viết theo ngôn ngữ kịch bản được lưu trữ trong hệ thống. Chúng nằm trong chuỗi nên chúng có một địa chỉ duy nhất. Kích hoạt hợp đồng thông minh bằng cách giải quyết một giao dịch mà nó liên quan. Hợp đồng thông minh được thực hiện một cách tự động và độc lập trong mạng. Hợp đồng thông minh được giao nhiệm vụ quản lý tương tác dựa trên dữ liệu giữa các thực thể mạng.

Những yếu tố cần thiết để lập nên một smart-contract:

* Chủ thể hợp đồng: Chương trình được cấp khả năng truy cập đến sản phẩm, dịch vụ liệt kê trong hợp đồng để có thể mở khóa hay khóa dữ liệu.
* Chữ ký điện tử số: Tất cả các bên tham gia hợp đồng đều thỏa thuận thông qua các private-key của họ.
* Điều khoản hợp đồng: Có dạng chuỗi các hoạt động. Yêu cầu các bên tham gia phải chấp nhận và ký.
* Nền tảng phi tập trung: phân bổ trên các nút của mạng blockchain.

Theo như [Andrew Tar](https://cointelegraph.com/explained/smart-contracts-explained) viết trong bài báo “Smart Contracts, Explain” <<TODO>> thì lợi ích của hợp đồng thông minh đem lại là không hề nhỏ:

* Tính bảo mật: Các smart-contract được phân bố và mã hóa trên các node. Việc này đảm bảo làm nó khó có thể bị thất lạc và thay đổi. Vì một khi dữ liệu lưu vào một khối thì gần như không thể thay đổi.
* Tiết kiệm thời gian, tiền bạc: Các công đoạn được tự động hóa và gần như loại bỏ hoàn toàn bên trung gian.
* Tính tiêu chuẩn hóa: hiện nay xuất hiện nhiều loại hợp đồng thông minh phù hợp để đáp ứng cho nhu cầu nghiệp vụ khác nhau.

#### Cơ chế đồng thuận

Hệ thống máy tính đáng tin cậy là hệ thống xử lý tất cả các thành phần hỏng hóc và đồng nhất hoạt động giữa các thành phần khác nhau bên trong. Tình huống này còn được thể hiện một cách trừu tượng qua bài toán một nhóm tướng lĩnh Byzatine cắm trại với quân đội của họ xung quanh một tòa thành kẻ thù [<lamport1982byzantine>]. Do khoảng cách địa lý, các tướng lĩnh giao tiếp với nhau qua sứ giả. Họ chỉ có thể giành chiến thắng khi đồng ý một chiến lược chung. Vấn đề đặt ra làm sao có thể đảm bảo các tướng lĩnh trung thành sẽ đạt được đúng chiến lược. Thời buổi liên lạc khó khăn, chưa có bộ đàm hay điện thoại, những tin nhắn trực tiếp liên lạc thông qua người đưa tin bằng ngựa. Người này sẽ đưa tin từ một vị tướng chỉ huy tới các vị tướng đạo quân khác. Một thông điệp bằng miệng là một thông điệp có nội dung tùy thuộc vào sứ giả truyền tin, nếu có tướng lĩnh phản bội thì kế hoạch được gửi đi sẽ bị thay đổi. Đây là một bài toán không có lời giải.

Để khắc phục những vấn đề tương tự như trong bài toán trên thì từ trước tới nay người ta sử dụng một hệ thống dịch vụ bên thứ ba tin cậy (TTP – trusted third party). Như là sử dụng một dịch vụ đưa tin của một nhà cung cấp đảm bảo: như dịch vụ ngân hàng hay nhà mạng… Bên thứ ba tin cậy được chứng minh qua số năm kinh nghiệm, chất lượng dịch vụ… Trong bài báo “A Proposed Architecture for Trusted Third Party Services” của ba tác giả Nigel Jerreries và Chris Mitchell, Michael Walker đã định nghĩa vai trò của dịch vụ bên thứ ba tin cậy trong lĩnh vực bảo mật: Để tạo một nền tảng có thể sử dụng để cung cấp dịch vụ người dùng, giải pháp dịch vụ bên thứ ba tin cậy mà từ đó người dùng có thể có được các mã hóa mật mã cần thiết để mã hóa dữ liệu của chúng hoặc sử dụng bảo mật các dịch vụ, cung cấp chính xác những thông tin cho bên thực thi pháp luật [<jefferies1996proposed>]. Việc sử dụng dịch vụ bên thứ ba đem lại không ít lợi ích tuy nhiên cũng đi kèm một số bất lợi cho các bên tham gia. Khi mà các bên tham gia muốn sử dụng dịch vụ phải chấp nhận các điều khoản mà bên thứ ba yêu cầu, bị bên thứ ba thu thập thông tin cá nhân – điều này rất nguy hiểm khi mà hệ thống của bên thứ ba bị hacker xâm nhập, thông tin người dùng sẽ bị lợi dụng vào các hành động bất hợp pháp, một số quyền lợi chính đáng không được tận dụng triệt để… Có thể lấy ngay một ví dụ đơn giản hiện nay về quy trình bán hàng trên các sàn giao dịch thương mại điện tử như Lazada, Shopee hay lớn hơn như Alibaba ... Ở đây các sàn giao dịch đóng vai trò là bên thứ ba tin cậy, khi người dùng muốn đăng sản phẩm lên ứng dụng để bán, họ phải kê khai thông tin cá nhân, thông tin mặt hàng, chấp nhận tất cả các điều khoản do bên thứ ba yêu cầu, phải chịu những khoản phí hoa hồng cho bên thứ ba…

Vấn đề đặt ra là để đảm bảo quyền lợi cao nhất, làm thế nào để các cơ quan tổ chức có thể trực tiếp tin tưởng tuyệt đối vào nhau mà không phải lệ thuộc bên thứ ba tin cậy đồng thời vẫn giải quyết được các vấn đề tướng Byzantine. Công nghệ Blockchain ra đời và có thể giải quyết vấn đề này bằng cách cung cấp các giao thức đồng thuận trên mạng của mình. Như đã biết, blockchain là một sổ cái peer-to-peer phi tập trung mà không có một cơ quan trung gian quản lý, kiểm soát nó. Các nút độc lập trong mạng phải đi đến sự đồng thuận về trạng thái sổ kế toán. Đồng thuận có thể định nghĩa đơn giản như một giải pháp được chấp nhận, hỗ trợ từ tất cả các thành viên trong mạng lưới. Một khối chỉ được tạo ra khi có sự đồng thuận từ tất cả các nút trong mạng. Dưới đây là một số cơ chế đồng thuận mà blockchain sử dụng:

* Proof-of-work

Bài báo “Blockchain Contract: A Complete Consensus using Blockchain” [<watanabe2015blockchain>] đã viết: Để đảm bảo độ tin cậy, để có thể thêm một khối mới vào khối trước đó thì công sức, thời gian để tạo ra khối đó cần phải được chứng minh được gọi là một Proof-of-work. Trong PoW mỗi nút trong một mạng tính toán giá trị băm của tiêu đề khối kế tiếp. Sự đồng thuận đòi hỏi giá trị băm được tính toán phải bằng hoặc nhỏ hơn giá trị ngưỡng [<zheng2017pow>]. Khi tìm được giá trị băm thỏa mãn thì nút sẽ phát sóng ra toàn mạng, thông báo tới các nút khác để các nút khác cùng xác nhận tính hợp lệ của giá trị băm. Nếu khối được xác nhận, các nút sẽ thêm khối mới này vào blockchain. Các nút còn được gọi là các miner (thợ đào Bitcoin). Phần thưởng sẽ là coin tương ứng với công sức mà họ đầu tư. Kỹ thuật này duy trì cân bằng mạng lưới Blockchain: khi Bitcoin tăng giá việc giải quyết bài toán PoW để tạo một khối vào mạng Blockchain trở lên khó khăn hơn và ngược lại khi giá trị bitcoin việc giải quyết bài toán PoW trở lên dễ dàng hơn.

* Proof of Stake

Không giống như PoW, giao thức PoS không tốn nhiều năng lượng cho các phép toán. PoW là một hình thức chứng minh quyền sở hữu tiền tệ [<king2012ppcoin>]. Theo thuật toán PoS, mỗi nút trong mạng được liên kết tới một địa chỉ. Nếu nút càng nắm giữ nhiều tiền khả năng đào coin của nó càng lớn. Những người có nhiều tiền tệ hơn sẽ ít có khả năng tấn công mạng vì nếu tấn công khả năng rủi ro là rất lớn. Như xổ số vậy, bạn càng có nhiều tờ xổ số, tỷ lệ trúng sẽ càng cao. Tuy nhiên, nếu không trúng thì thiệt hại của người đầu tư càng nhiều càng lớn. Tương tự như PoW thì PoS không dễ dàng giả mạo. Đây là một yêu cầu quan trọng trong hệ thống tiền tệ.

* PBFT (Practical byzantine fault tolerance):

Là một bản sao thuật toán để khắc phục vấn đề byzantine. Mỗi khối được xác định một vòng, một vòng được chọn theo một số quy tắc. Trách nhiệm của nó là đặt hàng giao dịch. Toàn bộ quá trình được chia làm ba giai đoạn: chuẩn bị trước, chuẩn bị và cam kết. Trong mỗi giai đoạn, một nút sẽ triển khai giai đoạn tiếp theo nếu nhận được số phiếu bầu trên 2/3 tổng số nút [<zheng2017pbft>].

#### Toàn vẹn dữ liệu

Khi một khối được thêm vào mạng blockchain thì gần như dữ liệu của nó trở nên gần như bất biến. Dữ liệu này không thể xóa hay sửa được vì chi phí và công sức để làm việc này là vô cùng lớn và mất thời gian, tiền bạc.

#### Bảo mật, tin cậy

Ngày nay, sự gia tăng các hành vi vi phạm quyền riêng tư người dùng càng trở lên nghiêm trọng. Trong đó bên thứ ba thu thập và kiểm soát một lượng lớn dữ liệu cá nhân.

Trong bài báo “Decentralizing Privacy: Using Blockchain to Protect Personal Data” có trích dẫn ước tính 20% dữ liệu của thế giới đã được thu thập trong vài năm qua. Facebook, mạng xã hội trực tuyến lớn nhất, đã tiến hành thu thập 300 petabyte dữ liệu cá nhân kể từ khi được thành lập, tương đương gấp 100 lần số lần Thư viện Quốc hội Hoa Kỳ thu thập trong 200 năm [<zyskind2015decentralizing>].

Blockchain đã chứng minh, trong không gian tài chính, việc phân bổ dữ liệu cá nhân phi tập trung trong hệ thống quản lý đảm bảo cho người dùng toàn quyền kiểm soát dữ liệu của mình. Blockchain là một nền tảng tập trung vào sự riêng tư. Blockchain được ví như một sổ kế toán công khai, nơi đó các đồng tiền kỹ thuật số được giám sát chặt chẽ, cuốn sổ này được chia sẻ cho mọi người tham gia mạng blockchain, mọi người đều có thể xem thông tin các giao dịch và không ai có thể sửa chữa nó. Mỗi người tham gia hệ thống đều có một bản copy hoàn chỉnh của cuốn sổ này, và khi dữ liệu trên cuốn sổ này có thay đổi thì bản copy kia cũng được thay đổi theo.

#### Minh bạch

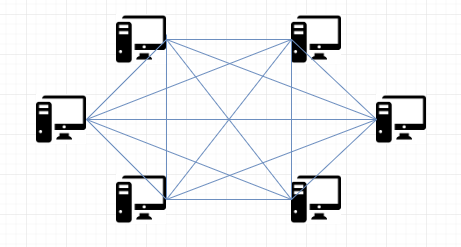
Tất cả các giao dịch trên mạng blockchain đều được kê khai rõ ràng, có nghĩa là tổng số bản ghi có thể kiểm chứng và không có một hành động thay đổi dữ liệu nào tồn tại [<sharma2017block>].

### Phân loại hệ thống

Hệ thống blockchain được phân ra 3 loại chính: public blockchain, private blockchain và consortium blockchain [<lin2017surveysystem>].

#### Public blockchain

Tất cả mọi nút đều có quyền đọc ghi dữ liệu trên mạng Blockchain. Quá trình xác thực giao dịch đòi hỏi sự đồng thuận từ tất cả các nút tham gia.



Hình 4: Mô hình minh họa mạng public blockchain

Hầu hết các thương mại điện tử được xây dựng trên public blockchain, chỉ cần kết nối Internet ai cũng có thể truy cập và khai thác dữ liệu trong mạng blockchain [<xu2016blockchain>]. Sử dụng public blockchain đem lại sự minh bạch trong thông tin trong kiểm toán, nhưng chưa đảm bảo tính riêng tư của thông tin.

##### Bitcoin

Bitcoin là đồng tiền kỹ thuật số phi tập trung đầu tiên [<nakamoto2008bitcoin>]. Không được phát hành bởi bất kỳ chính phủ hay ngân hàng, tổ chức nào và dựa vào các giao thức mã hóa và mạng người dung phi tập trung để tạo, lưu trữ và chuyển giao. Được phát triển bởi một người hoặc một nhóm người với bút danh Satoshi Nakamato và được đề xuất vào năm 2008. Sử dụng hệ thống thanh toán trực tuyến, trong đó kỹ thuật mã hóa được sử dụng để điều chỉnh việc tạo ra các đơn vị tiền tệ và xác minh việc chuyển tiền, hoạt động độc lập với ngân hàng trung ương [<swan2015blockchain>]. Không chịu sự quản lý từ bất kỳ tổ chức nào.

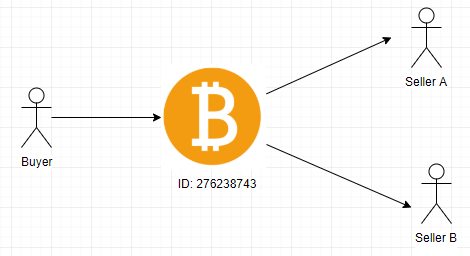
* **Transactions**

Giao dịch ngang hàng peer-o-peer trên bitcoin được diễn ra trực tiếp giữa bên gửi và bên nhận. Cho phép người dùng cuối tạo ra các giao dịch vô danh thay vì tiết lộ thông tin cá nhân, các giao dịch diễn ra công khai về thông tin về giao dịch: số lượng bitcoin chuyển giao, thời gian... [<koshy2014analysis>] Các nút thu thập thông tin giao dịch trong một khoảng thời gian vào một khối. Lịch sử giao dịch được lưu trữ tất cả trong sổ cái. Mỗi nút tham gia mạng sẽ có một bản sao thông tin mạng Blockchain.

Hệ thống BTC đã tạo ra một bài toán để duy trì tính ổn định hệ thống. Một khối giao dịch mới được tạo ra bằng cách thực hiện PoW dựa trên kỹ thuật băm. Hàm băm là bất kỳ thuật toán ánh xạ dữ liệu. Thông thường hàm băm không thể đảo ngược, Bitcoin sử dụng thuật toán SHA-256 trong thuật toán băm của mình [<taylor2013bitcoin>]. Mỗi khối mang theo một PoW-chứng minh sự đầu tư công sức, thời gian giải quyết các thông tin cần thiết để tạo một khối chứa thông tin các giao dịch trong một khoảng thời gian. Khi một block được một nút đưa ra, nó cần được chứng minh tính hợp hệ của mình bằng cách phát ra toàn mạng, đợi sự đồng thuận từ toàn bộ các nút. Mỗi nút kiểm tra tính hợp lệ của khối đề xuất qua thông tin các phiên giao dịch, thời gian, số bitcoin.... Các nút có thể tham gia và rời mạng theo ý muốn, chấp nhận chuỗi PoW làm bằng chứng cho sự diễn ra các giao dịch.

Tính bảo mật trong các giao dịch: Chỉ có thể thay đổi thông tin lịch sử giao dịch khi có thể làm lại tất cả các PoW của tất các các khối trong chuỗi. Việc này là không thể bởi vì việc đó gây tốn quá nhiều thời gian, công sức tính toán và tiền bạc.

Bài toán Double-spending: hình thức gian lận sử dụng hai hay nhiều phiên giao dịch khác nhau trên cùng một số lượng tiền nhất định trên một tài khoản. [<karame2012two>].



Hình 5: Sơ đồ minh họa về bài toán Double-Spending

Lấy một ví dụ đơn giản như khi gần đây, báo chí Việt Nam có đăng tin các vụ lừa đảo mua bán nhà như anh A đem một căn hộ A mang bán cho nhiều người trong cùng một khoảng thời gian. Lợi dụng lòng tin của những người mua nhà, anh A tiến hành những hành động phi pháp để chiếm lợi cho bản thân. Hình thức Double-spending trong blockchain cũng diễn ra tương tự như trong trường hợp khi A là một hacker dùng cùng một đơn vị bitcoin đem ra giao dịch cho nhiều người khác trong cùng một khoảng thời gian. Bài toán được giải quyết bằng cách sử dụng tính linh hoạt của mạng lưới máy chủ phi tập trung để tạo bằng chứng tính toán của thứ tự thời gian các phiên giao dịch.

* **Mining**

Không có một thực thể tập trung nào quyết định việc tạo bitcoin [<nakamoto2008mining>]. Quy trình khai thác bitcoin là việc giải quyết bài toán PoW từ một khối các giao dịch. Những người thực hiện việc đào bitcoin được gọi là các thợ mở (miner). Khi một khối được xác minh là hợp lệ từ tất các các nút trong mạng, thợ mỏ tạo khối sẽ nhận được một khoản bitcoin tương ứng.

Việc khai thác bitcoin càng trở lên cạnh tranh hơn khi giá bitcoin tăng. Càng nhiều thợ mở việc giải bài toán PoW càng khó và mạng trở lên an toàn hơn.

Bài báo “Technical Background and Data Analysis” đã nói rằng: trung bình mất 10 phút thì mới có một khối được tạo ra [<badev2014bitcoin>]. Bitcoin được tạo ra trong các khối, luôn được tạo theo tỷ lệ ổn định. ViếtSố lượng bitcoin bị giới hạn ở mức 21 triệu và giới hạn này dự kiến sẽ đạt được vào năm 2140.

##### Ethereum

Ethereum là một framework cho tiền điện tử, nó sử dụng công nghệ Blockchain để cung cấp một nền tảng tính toán toàn cầu mở, được gọi với tên là máy ảo Ethereum (Ethereum Virtual Machine – EVM) [<bhargavan2016ethereum>]. Tương tự Bitcoin, Ethereum cũng có một đơn vị tiền ảo, được gọi là Ethe, trên một blockchain dựa trên PoW. Ethereum là một sổ cái toàn diện hơn Bitcoin: lưu trữ toàn bộ các chương trình hoàn chỉnh dưới dạng EVM đồng thời cho phép các giao dịch truy cập vào các hàm logic, với dữ liệu bổ sung dưới dạng các đối số.

Ethereum còn được định nghĩa đơn giản là một hệ thống có thể phát hiện các thay đổi với thông tin và ghi nhớ chúng theo thời gian [<dannen2017introducing>]. Dễ dàng cho phép lập trình viên xây dựng và phát triển các ứng dụng phi tập trung.

Trong khi blockchain bitcoin chủ yếu là được sử dụng để theo dõi quyền sở hữu tiền tệ kỹ thuật số, khối Ethereum tập trung vào việc thực thi mã lập trình của bất kỳ ứng dụng phân cấp nào [<zhang2018review>].

* **Hợp đồng thông minh**

Ethereum là một nền tảng blockchain mã nguồn mở kết hợp hợp đồng thông minh, cung cấp máy ảo phân cấp đề xử lý hợp đồng. Thông qua sử dụng đồng tiền Ethe, người dùng có thể tạo ra nhiều dịch vụ khác nhau, các ứng dụng hoặc các hợp đồng trên nền tảng này [<lin2017ethesmart>].

Các hợp đồng thông minh trong các chương trình của Ethereum được viết bằng Turning-complete theo ngôn ngữ bytecode, được gọi là bytecode EVM [<atzei2017ethsmatcontract>]. Một hợp đồng thông minh là một tập hợp các hàm, mỗi hàm được định nghĩa bởi một chuỗi các bytecode.

Hàm khởi tạo được thực hiện bởi các giao dịch xác định hợp đồng thông minh. Người gửi giao dịch trở thành chủ sở hữu của hợp đồng thông minh [<bogner2016decentralised>]. Hợp đồng lưu trữ một giá trị khóa các bên liên quan giao dịch. Cả hai bên đồng ý với các điều kiện ký kết của hợp đồng. Mạng Ethereum đảm bảo rằng hợp đồng được thực hiện tương ứng.

* **Token**

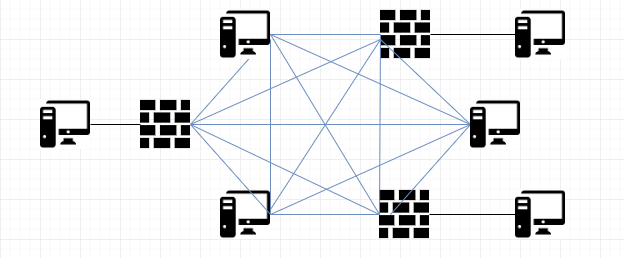
Được sử dụng để giữ các giao dịch tập tin ngang hàng trong kiểm tra, đảm bảo người tiêu dùng có thể thực hiện các thanh toán từ nhỏ tới lớn cho nhà cung cấp dịch vụ [<wood2014ethereum>].

* **Work to earn**

Thợ mở làm việc để kiếm Ethe token. Hơn cả một loại tiền tệ mã hóa để trao đổi, Ethe cũng được sử dụng bởi các nhà phát triển ứng dụng để thanh toán các chi phí giao dịch và dịch vụ trong mạng lưới Ethereum.

* **PoW, PoS**

#### Consortium blockchain



Hình 6: Mô hình consortium blockchain

##### Hyperledger

Dự án Hyperledger ([www.hyperledger.org](http://www.hyperledger.org)) là kết quả của sự nỗ lực cộng tác để tạo ra một khung phân phối sổ kế toán mã nguồn mở, thúc đẩy công nghệ blockchain. Nó nhằm mục đích nâng cao công nghệ blockchain bằng cách xác định và thực hiện một nền tảng tiêu chuẩn mở ngành công nghiệp để phân phối sổ cái (distributed ledgers), có thể biến đổi cách thức giao dịch kinh doanh được tiến hành trên toàn cầu. Dự án được tổ chức bởi Linux Foundation, bao gồm các nhà lãnh đạo về tài chính, ngân hàng, Internet of Thing (IoT), chuỗi cung ứng, sản xuất và công nghệ [].

Có thể coi Hyperledger là một hệ sinh thái được phát triển để minh bạch, công khai khai sáng tiềm năng của công nghệ Blockchain ứng dụng vào các lĩnh vực trong doanh nghiệp.

Hyperledger không hỗ trợ Bitcoin hoặc bất kỳ cryptocurrency khác.

Hyperledger phát triển một chiến lược gọi là Hyperledger Umbrella với nhiều dự án khác nhau.

* **Hyperledger Fabric**

Hyperledger Fabric là một mô đun và hệ thống phân tán có thể mở rộng để chạy mạng blockchain cho phép [2]. Hỗ trợ các giao thức đồng thuận blockchain có thể cắm vào để đảm bảo rằng các giao dịch được xác thực theo chính sách bởi những người tham gia mạng lưới kinh doanh được chỉ định.

Hyperledger Fabric cũng là hệ thống blockchain đầu tiên chạy trên tiêu chuẩn ứng dụng phân tán, ngôn ngữ lập trình phục vụ mục đích chung, không phụ thuộc vào một ngôn ngữ tiền điện tử cụ thể nào.

* + Ledger

Trên trang chủ <https://hyperledger-fabric.readthedocs.io> có viết: một ledger, hay còn được gọi là sổ cái, là một sổ kế toán chứa toàn bộ nhật ký giao dịch trên toàn hệ thống Blockchain. Cũng như khi chúng ta kiểm tra tài khoản ngân hàng của mình. Mỗi tháng tạo một số các phiên giao dịch như: rút tiền, chuyển tiền hay mở tài khoản tiết kiệm… Khi chúng ta muốn kiểm tra lại tất cả các giao dịch trước đây phục vụ cho việc đánh giá kế hoạch chi tiêu thì danh sách các phiên giao dịch trước đó mà hệ thống trả lại cho chúng ta cũng là một sổ cái.

Chúng ta sẽ đi sâu vào kiến trúc cụ thể của một ledger trong hyperledger fabric.

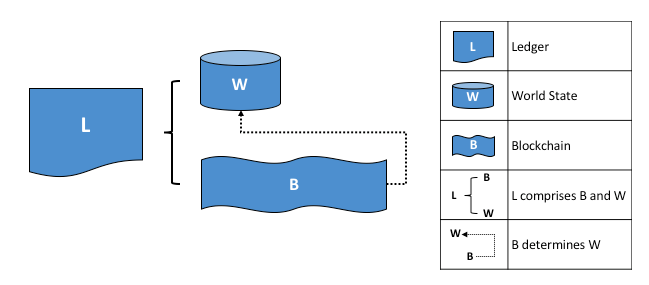


Figure 2: Sơ đồ mô tả cơ sở dữ liệu hệ thống

Cơ sở dữ liệu tích hợp blockchain này bao gồm hai phần chính: world-state database and blockchain database.

Đầu tiên, world-state database chứa giá trị hiện tại của một tập hợp trạng thái của sổ kế toán. Mặc định, trạng thái của sổ cái được biểu thị dưới cặp key-value. Cũng như cái tên world state có nghĩa là trạng thái trái đất mà trái đất là luôn luôn vận động, các cặp key-value này có thể được tạo mới, cập nhật và xóa.

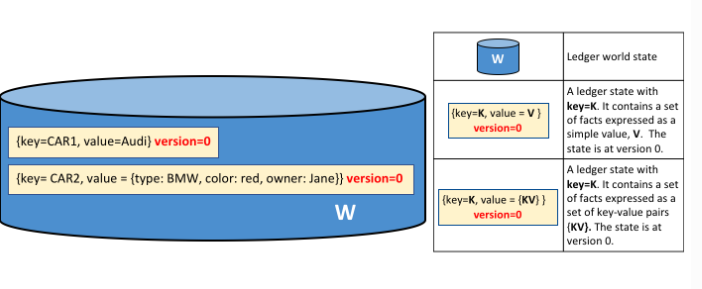


Figure 3: Hình vẽ minh họa world-state database

Thứ hai, blockchain database – nhật ký giao dịch ghi lại tất cả các trạng thái thay đổi của world-state database. Các giao dịch được thu thập sẽ được lưu vào block nối tiếp vào chuỗi blockchain trong mạng. Cấu trúc này khác hoàn toàn với cấu trúc bên trong world-state database vì một khi dữ liệu được thêm vào thì nó gần như không thể thay đổi được. Nó là một chuỗi bất biến của các khối, các khối chứa một tập hợp các giao dịch có thứ tự.

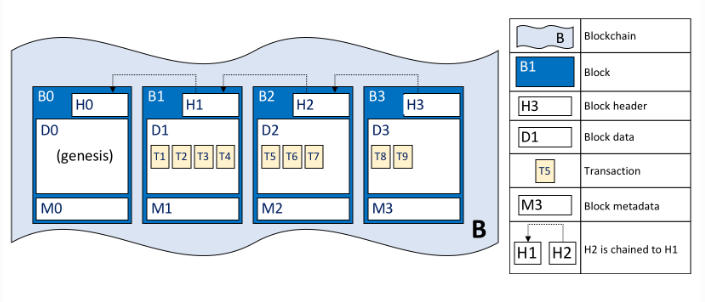


Figure 4: Hình ảnh minh họa Blockchain database

Sự kết hợp hai database này lại thu được một sổ cái hay còn gọi với thuật ngữ Ledger.

* **Hyperledger Composer**

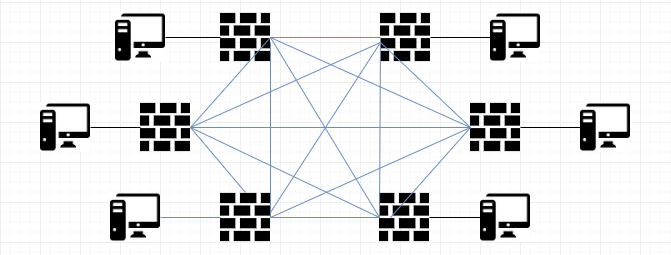
Hyperledger Fabric được sử dụng để chạy mạng lưới kinh doanh trong khi đó Hyperledger Composer được sử dụng để triển khai các hợp đồng thông minh hoặc các chức năng xử lý giao dịch chạy trên mạng blockchain [10].

Mục tiêu chính mà Hyperledger Composer được tạo ra là tăng tốc thời gian xử lý. Nhanh chóng mô hình hóa mạng lưới kinh doanh hiện tại, chứa các tài sản, các giao dịch liên quan tới chúng, tài sản là hàng hóa, dịch vụ hoặc tài sản vô hình hoặc hữu hình. Là một phần trong mạng doanh nghiệp.

Hyperledger composer là tập hợp các công cụ cho phép xây dựng một mạng blockchain dễ dàng hơn và bao gồm các thành phần sau:

* Ngôn ngữ mô hình gọi là CTO (tên dự án ban đầu là Concerto).
* Giao diện người dùng Hyperledger Composer Playground qua trình duyệt cho phép cấu hình nhanh chóng, triển khai và thử nghiệm mạng doanh nghiệp. Sử dụng bộ nhớ cục bộ của trình duyệt để mô phỏng trạng thái lưu trữ của mạng blockchain.
* Công cụ giao diện dòng lệnh (CLI) để thực thi mạng lưới blockchain thông qua tích hợp chạy một thực thể Hyperledger Fabric blockchain.

#### Private blockchain



* **Ripple**

Một nền tảng dựa trên Blockchain nhắm tới mục tiêu liên quan tới các lĩnh vực tài chính, thanh toán đồng thời thúc đẩy thời gian các giao dịch trong vài giây

Giao thức đồng thuận

Sự đồng thuận trong toàn bộ mạng lưới đạt được khi mỗi mạng con riêng lẻ thực hiện sự đồng thuận

### Tổng kết

Blockchain cơ bản là một cơ sở dữ liệu phân tán, các bản ghi hay sổ cái công khai của tất cả các giao dịch hoặc sự kiện kỹ thuật số được thực hiện giữa các bên tham gia hệ thống. Mỗi giao dịch trong sổ cái được xác minh theo sự đồng thuận nhất trí giữa các thêm tham gia. Thông tin một khi được lưu vào mạng blockchain thì gần như không thể nào xóa và sửa vì chi phí cho việc xóa, sửa đổi dữ liệu cực kỳ lớn và mất thời gian.

Không thể tranh cãi ích lợi của việc sử dụng công nghệ Blockchain khi nó hoạt động hoàn hảo trong những năm qua và đang được áp dụng trên nhiều hệ thống như tài chính, ngân hàng, y tế...

Công nghệ Blockchain đem lại tiềm năng cách mạng kỹ thuật số to lớn bằng cách cho phép một sự đồng thuận phân tán, nơi mà mỗi người trong hệ thống, không quan trọng ở đâu chỉ cần internet và sử dụng hệ thống, tiến hành các giao dịch liên quan tới tài khoản kỹ thuật số. Không kể thời gian hiện tại, quá khứ hay tương lai công nghệ này xác minh các giao dịch mà không gây ảnh hưởng tới quyền riêng tư, tài sản của mỗi cá nhân trong hệ thống.

## TMĐT

TMĐT hiện nay được tổ chức theo mô hình business-to-consumer (B2C). Mô hình này mô tả các hoạt động kinh doanh phục vụ người tiêu dùng cuối (end-users) với các sản phẩm, dịch vụ. Một cá nhân truy cập trang web của tổ chức trước khi họ mua sản phẩm. Thông qua các thao tác click chuột, nhấp vào các liên kết mua sắm thì hệ thống trang web sẽ gửi cho họ hàng trăm sản phẩm, dịch vụ cung cấp bởi các thương hiệu hàng đầu. Giao dịch mua được thực hiện như thể họ giao dịch trực tiếp với các người bán, các thương gia. Các giao dịch mua được gửi tới người mua và tổ chức giới thiệu mặt hàng sẽ nhận được hoa hồng trên mỗi giao dịch thành công [7].

Để có thể đưa các sản phẩm, dịch vụ của mình đến người tiêu dùng thông qua website thì các thương gia cần phải chi trả một khoản chi phí cho tổ chức quản lý website.

Ta có thể khái quát mô hình B2C như sau:

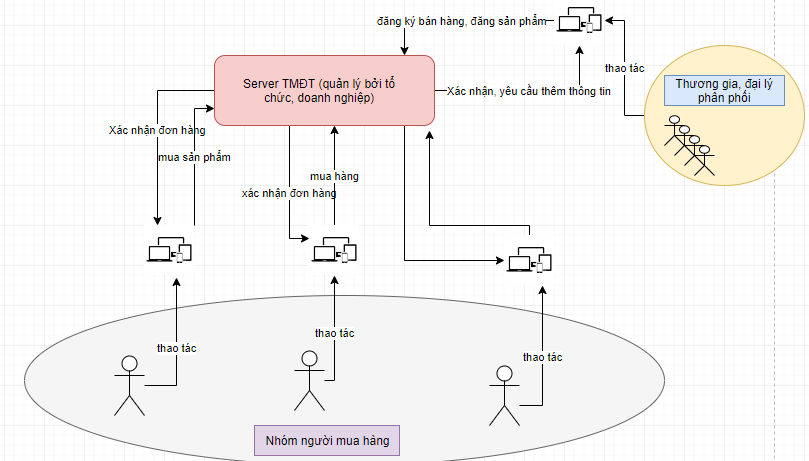


Figure 1: Mô hình B2C trong TMĐT truyền thống

Ta thấy rõ được ở TMĐT truyền thống chia làm ba nhóm đối tượng chính: Người tiêu dùng (bên mua), tổ chức kinh doanh quản lý website, thương gia (người bán).

Bên nhóm người tiêu dùng, họ tiến hành thao tác với ứng dụng trên website tiến hành chọn lựa đặt hàng những mặt hàng mình ưa thích. Những đơn hàng, thông tin thay đổi liên quan tới người dùng sẽ được đóng gói lại và tạo lên một request chứa gói thông tin này cho bên server, nhóm tổ chức quản lý server này họ sẽ tiến hành lưu trữ, xác nhận các đơn hàng và thay đổi tùy theo những yêu cầu hợp lệ từ bên khách hàng.

Bên nhóm thương gia, đại lý phân phối họ tiến hành đăng ký thương hiệu trên website, liệt kê những mặt hàng mình muốn bán, đăng sản phẩm của mình lên website. Tương tự như bên nhóm người tiêu dùng, tất cả các thông tin này sẽ được đóng gói lại được gửi qua một request cho bên server. Bên server sẽ tiến hành tạo quy định hợp đồng, tiến hành lưu trữ đăng sản phẩm của bên đại lý muốn bán…

Tổ chức quản lý hệ thống website quyết định tất cả các giao dịch có được phép thành công hay không. Họ có thể truy cập vào thông tin tài khoản cả bên mua và bên bán, quy định tất cả các điều lệ mà bên bán và bên mua phải tuân theo. Quyền lợi mà người bán và người mua không được đẩy lên cao nhất khi mà giá trị sản phẩm họ mua hay bán không chính xác bằng giá trị thực của sản phẩm.

## Đề xuất ứng dụng Blockchain vào TMĐT

Mô hình TMĐT này em thiết kế theo mô hình peer-to-peer (P2P). P2P nhấn mạnh vào các hoạt động mua bán được thực hiện trực tiếp giữa người mua, người bán thông qua website. Không như hệ thống TMĐT truyền thống thì bên tổ chức quản lý server chiếm mọi quyền trong tất cả các giao dịch. Ở đây vai trò của người tiêu dùng (bên mua) và người bán hay các thương gia có quyền lớn nhất (bên bán). Bên quản lý server không có hoặc có rất ít quyền trong việc thiết lập các giao dịch.

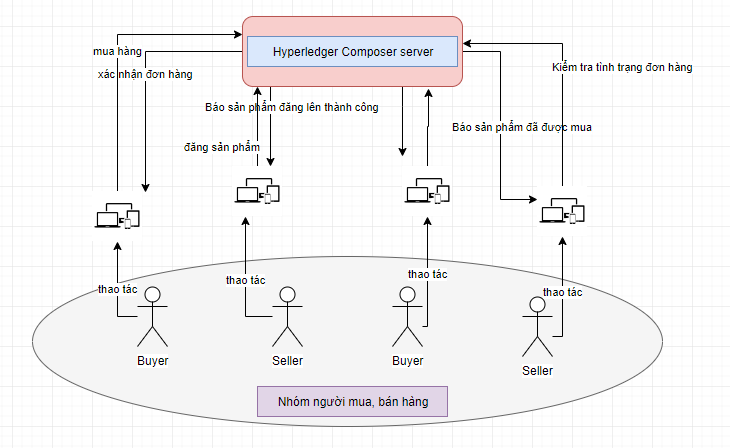


Figure 6: Mô hình TMĐT ứng dụng công nghệ Blockchain

Ở đây nhóm bán và nhóm tiêu dùng được gộp chung lại một nhóm. Họ có quyền cao nhất trong các giao dịch. Gần như peer-to-peer, giao dịch trực tiếp với nhau. Khác với server truyền thống thì server trong ứng dụng TMĐT áp dụng blockchain này sử dụng Hyperledger composer server. Như được trình bày ở phần <<TODO>> thì server này lưu trữ thông tin các giao dịch trên mạng blockchain và state DB <<TODO>>.

Tiếp sau đây là phần so sánh chi tiết giữa TMĐT truyền thống và TMĐT tích hợp công nghệ Blockchain mà em đề xuất.

* Tổ chức quản lý hệ thống

*Điểm chung*: Doanh nghiệp, tổ chức hay một cá nhân đứng ra quản lý hệ thống. Hệ thống website lưu trữ tất cả các giao dịch, thông tin khách hàng, đối tác bán hàng…

*Khác biệt*: So với TMĐT truyền thống thì TMĐT ứng dụng blockchain có điểm khác biệt cơ bản ở vai trò của con người đứng ra tổ chức hệ thống được giảm bớt đi.

Ví dụ để quản lý một hệ thống TMĐT thì doanh nghiệp, tổ chức phân ra rất nhiều các ban, các phòng… Mỗi ban, mỗi phòng chiếm một vai trò nhất định như: duy trì, quản lý máy chủ, tìm kiếm đối tác, mặt hàng, quản lý thu chi, hành pháp…

Để quản lý một hệ thống TMĐT ứng dụng Blockchain thì rất nhiều ban, phòng… có thể được bỏ qua. Có thể chỉ cần ít nhất bộ phận duy trì, quản lý máy chủ là đủ để duy trì cả hệ thống. Nhu cầu nhân lực được giảm đi rất nhiều.

* Minh bạch, bất biến
* Bảo mật, tin cậy

Sàn giao dịch TMĐT truyền thống dễ bị khai thác dữ liệu. Các lỗ hổng chính là mục tiêu tấn công của các hacker. Họ có thể lợi dụng chúng để tiến hành làm giả, thay đổi thông tin các giao dịch. Tiến hành mua nhiều sản phẩm mà không mất bất kỳ chi phí nào. Hay ăn cắp thông tin của người dùng khác…

TMĐT ứng dụng công nghệ Blockchain là một sàn giao dịch ứng dụng công nghệ mới, tính bảo mật cao hơn. Thông tin các giao dịch, định danh người dùng một khi được lưu vào mạng blockchain thì cực kỳ khó để có thể thay đổi. Từ đó duy trì tính xác thực của thông tin.

* Hợp đồng thông minh

Trong sàn giao dịch TMĐT truyền thống: Khi bên doanh nghiệp, tổ chức quản lý hệ thống tiến hành ký kết hợp đồng kinh doanh với các thương gia, các đại lý cung cấp mặt hàng. Thì để tạo ra hợp đồng điện tử này thì phải mất chi phí cho bên pháp lý biên soạn và có bên thứ ba đứng ra thực thi. Ở đây phải mất thời gian và tốn kém, nhiều khi còn thiếu tính minh bạch trong các khâu để hoàn thành hợp đồng. Những hợp đồng xảy ra sự cố thì lại phải dựa vào hệ thống pháp lý để giải quyết.

Với sàn giao dịch TMĐT ứng dụng công nghệ Blockchain thì hợp đồng giữa các bên được gọi với thuật ngữ smart-contract (hợp đồng thông minh), ở đó các bên tham gia phải chấp thuận các điều khoản trong hệ thống được lập trình viên thiết lập từ trước. Toàn bộ đoạn mã được thực thi trên hệ thống sổ cái Blockchain. Trong đó, quy định tất cả các điều khoản và hình phạt tương đương. Sau khi hệ thống khởi chạy thì ký kết việc hợp đồng giữa các bên không cần sự can thiệp của bên tạo hợp đồng nữa. Quyền lợi giữa bên mua và bên bán được đảm bảo, độ tin cậy cao, ít xuất hiện lỗi ngoài ý muốn. Mục đích chính ở đây là cho phép bên mua và bên bán có thể giao dịch trực tiếp với nhau thông qua internet mà không phải lệ thuộc quá nhiều vào bên trung gian.

* Đồng thuận

Với TMĐT truyền thống: thì có một hệ thống phân cấp rõ ràng. Doanh nghiệp, tổ chức quản lý sàn giao dịch (gọi tắt bên A) là bên có quyền cao nhất. Các giao dịch, thông tin người dùng hay người bán (các thương gia, các đại lý) (gọi tắt bên B) có thể bị bên A thay đổi. Bên A có thể hủy hay áp đặt các điều lệ mới để đặt lợi nhuận kinh doanh của mình. Để đăng một sản phẩm lên sàn thì người bán phải chịu một khoản phí hoa hồng cho bên A. Mức giá mà người mua mua trên sàn giao dịch này lớn hơn mức giá chuẩn mà người bán đặt ra. Từ đây bên A có thể thu lợi nhuận từ cả bên bán và bên mua. Từ đó thấy được bên chịu thiệt luôn là bên B.

Ngược lại với TMĐT ứng dụng Blockchain thì đề cao tính ngang hàng (peer-to-peer). Trong mọi giao dịch bên A không có vai trò gì. Giao dịch chỉ được phép thực hiện khi bên mua và bên bán chấp thuận, đảm bảo thuận mua vừa bán. Mức giá sản phẩm của bên mua phải chi trả luôn bằng đúng mức giá mà bên bán đề ra. Ở đây quyền lợi người tiêu dùng và người bán hàng được đảm bảo một cách tối ưu.

*Tóm lại*, khi ứng dụng công nghệ Blockchain vào TMĐT lợi ích đem lại vô cùng to lớn. Giảm bớt chi phí, nhân lực, thời gian, tiền bạc… Thông tin được bảo mật tốt hơn.

## Tổng kết chương

# Giải pháp bài toán xây dựng hệ thống thương mại điện tử dựa trên nền tảng blockchain

Ứng dụng mang đầy đủ các chức năng cơ bản mà một sàn giao dịch thương mại điện tử hiện nay đang có. Có thể kể đến như: đăng nhập vào ứng dụng, mua sản phẩm, bán sản phẩm, giỏ hàng, thanh toán, nạp tiền, lịch sử giao dịch, kiểm soát thông tin khách hàng và các phiên giao dịch liên quan.

* Đăng nhập hệ thống: Người dùng đăng nhập vào hệ thống thông qua tài khoản mạng xã hội như facebook, github, gmail.
* Quản trị danh mục sản phẩm: Cho phép hiển thị danh mục các sản phẩm đang được bày bán, cập nhật giá…
* Giỏ hàng điện tử: Cho phép người mua hàng có thể cùng lúc đặt nhiều món hàng với các số lượng khác nhau. Quá trình đặt hàng thông qua tính năng này cho phép thu thập đầy đủ các thông tin của người dùng kèm theo các thông tin thanh toán.
* Quản trị thông tin thành viên/khách hàng: Người quản trị có thể xem tất cả thông tin khách hàng, các phiên giao dịch liên quan tới khách hàng. Từ đó thu thập và đề xuất tái kiến trúc dựng nâng cấp hệ thống. Còn khách hàng thì có thể thay đổi thông tin của mình và xem xét lịch sử giao dịch.
* Mua hàng: Chuyển trạng thái hàng hóa đang được bày bán sang đã được đặt mua khi một giao dịch mua bán thành công.
* Bán hàng: Người dùng cung cấp thông tin sản phẩm và bày bán lên sàn giao dịch.
* Thanh toán trực tuyến: Người tiêu dùng nạp tiền vào ứng dụng và được quy đổi sang một số tiền tương ứng. Với số tiền điện tử này họ có thể tiến hành thanh toán các giao dịch mua bán.
* Kiểm soát thông tin khách hàng và các giao dịch liên quan: chức năng dành cho quản trị viên hệ thống, giám sát hệ thống để thực hiện những cải tiến, điều chỉnh hệ thống để phù hợp hơn với người tiêu dùng.

Ứng dụng được phát triển theo mô hình client-server. Giao diện client trên nền tảng website. Ứng dụng được đặt tên là Exchange App.

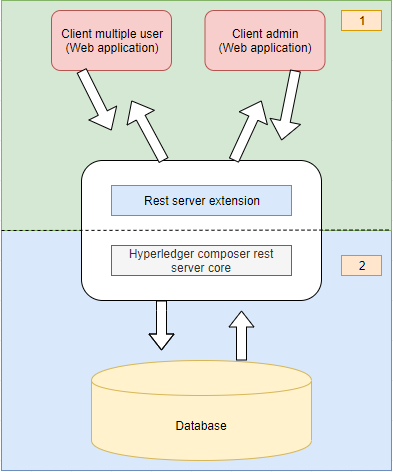


Figure 8: Cấu trúc hệ thống của ứng dụng

Hệ thống được chia ra hai phần chính:

* Phần phát triển: client và rest server extension (phần 1 trong hình).
* Phần công cụ sử dụng: Hyperledger composer rest server core và cơ sở dữ liệu tích hợp blockchain (hyperledger fabric) (phần 2 trong hình).

Về công cụ sử dụng: Hyperledger composer server core được miêu tả chi tiết ở phần <<TODO bên trên>>. Cơ sở dữ liệu tích hợp công nghệ blockchain được miêu tả rõ ở phần ledger fabric <<TODO>>. Client được phát triển với hai ứng dụng: một chạy ở chế độ đa người dùng và một chạy ở chế độ admin.

Về phần phát triển, bên client được phát triển trên nền tảng web application cung cấp giao diện web cho người dung tương tác với ứng dụng. Rest server extension là rest service được phát triển khả rộng trên công cụ Hyperledger Composer server.

## Hyperledger composer rest server extension

### Model

* **Enumerated type**

Một tập các giá trị, thuộc tính cụ thể đã được biết, mô hình kiểu liệt kê. Giúp hạn chế các giá trị lặp trở nên dễ dàng hơn, xác thực đơn giản hơn.

Ví dụ như danh mục sản phẩm bao gồm: Điện thoại thông minh, quần áo, xe cộ, rau, củ, quả… hay tình trạng của sản phẩm đang bày bán hay đã được mua.

* **Concept**

Thực thể tồn tại trong mô hình kinh doanh, nhưng không phải tài sản, các bên tham gia. Là một đại diện của một tập hợp. Ví dụ như địa chỉ nhà bao gồm tên nước, thành phố, quận.

* **Assets**

Assets ở đây có thể là bất cứ tài sản nào có giá trị như: một chiếc điện thoại nokia, một chiếc áo, một kilogam xoài... Tất cả chúng là những thứ có thể được đem lên trao đổi.

* **Participants**

Một participant là một thành viên tham gia mô hình mạng.

Tất cả các bên được xác thực bởi oauth <<TODO: 3.2.2.1.1>> trước khi họ có quyền truy cập vào ứng dụng. Khi người dùng này được xác thực, họ được chỉ định một thẻ ID blockchain bởi Membership Services =[]=[]. Ở trong, hệ thống này bên tham gia là người mua và người bán. Người dùng vừa có thể là người mua vừa có thể là người bán. Một cá nhân có thể đăng ký một, hai hoặc ba tài khoản. Mỗi tài khoản được gán cho một ID duy nhất. Mỗi người tham gia là chủ sở hữu dữ liệu và yêu cầu dữ liệu. Người yêu cầu dữ liệu là người gửi được xác thực yêu cầu truy cập dữ liệu cụ thể của chủ sở hữu.

* **Transactions**

Khi một asset được động tới như mua hay thay đổi trạng thái, thì tất cả những thay đổi đó sẽ làm thay đổi trạng thái của sổ cái blockchain. Sự tương tác trong model của Hyperledger Composer đều được gọi là transactions, dịch nôm na trong tiếng việt là các phiên giao dịch.

### Script file

Triển khai chi tiết các nghiệp vụ logic của các transaction để tạo nên một smart-contract như được mô tả ở <<TODO>>. Nêu rõ các điều kiện, điều khoản cụ thể, các giá trị được thiết lập. Ví dụ: chuyển đổi thông tin mặt hàng giữa hai khách hàng A và B, A phải là chủ sở hữu của mặt hàng Z, mặt hàng Z còn trong tình trạng còn hàng, B thì phải đủ số tiền để thanh toán mặt hàng…

Trong Exchange App, mô tả chi tiết các logic nghiệp vụ của các transaction.

Transaction Trade:

* + Đổi quyền sở hữu sản phẩm cho người mua.
  + Chuyển tình trạng sản phẩm từ đang bán sang đã được mua.
  + Chuyển trạng thái giỏ hàng của sản phẩm sang tình trạng phi giỏ hàng.
  + Tính tổng số tiền của sản phẩm (số lượng x giá thành trên từng đơn vị sản phẩm), và trừ số tiền đó vào tài khoản của người mua đồng thời cộng vào số dư tài khoản của người bán.
  + Cập nhật trạng thái sổ kế toán blockchain. Bước này được coi là quan trọng nhất.

Transaction MoveProToCart:

* Đổi trạng thái giỏ hàng mặc định của sản phẩm khi được bày bán bằng email người mua tạm thời (chưa mua, mới chỉ tích sản phẩm vào giỏ hàng).
* Cập nhật trạng thái sổ kế toán blockchain.

Transaction IncreaseBalance:

* Cộng vào số dư tài khoản của người dùng với số tiền được nạp.
* Cập nhật trạng thái sổ kế toán blockchain.

### Access control

Trong một mạng lưới kinh doanh không phải ai cũng có quyền truy cập mọi thứ. Ví dụ, người mua chỉ xem được thông tin của mình và những món hàng mình đã mua, không xem được thông tin của tất cả người dùng, hay được phép sửa thông tin món hàng người khác đăng bán.

### Query file

File chứa các câu lệnh truy vấn dữ liệu sql, có thể kèm các điều kiện hay hoàn cảnh cụ thể.

### Business network archive (BNA)

Business network archive là một file nén lưu trữ thông tin về các tài sản, hàng hóa, và các phiên giao dịch liên quan tới chúng. Được đóng gói lại và xuất ra dưới dạng file nén để triển khai trên ứng dụng web.

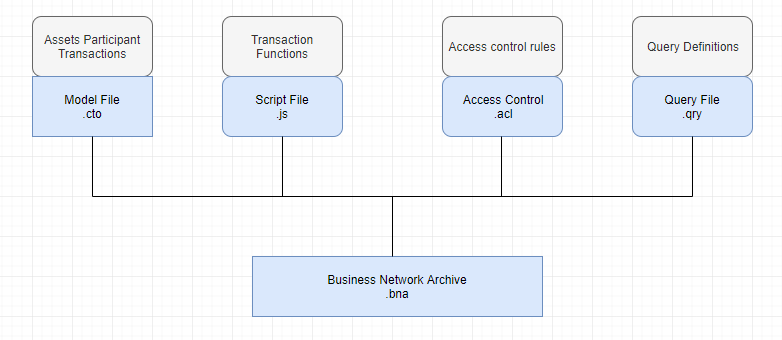


Figure 9: Mô hình Business network archive

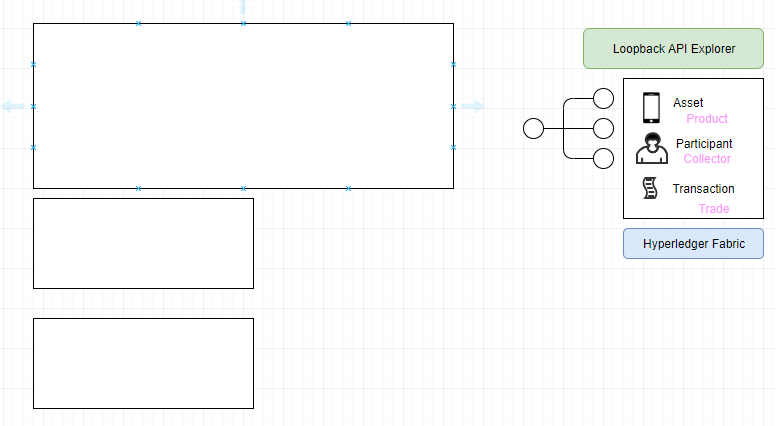
Hyperledger Composer đóng gói dữ liệu từ bốn file trên và một số file cấu hình cần thiết thành một file (định dạng .bna). Sử dụng file này kết hợp với ID Cards (bao gồm thông tin người đăng nhập và xác thực) ta sẽ kết nối tới một sổ kế toán phân phối (distributed ledger) trên nền Fabric. Nơi đó dữ liệu được lưu trên nền tảng Blockchain.

### Rest API support

REST là từ viết tắt của REpresentational State Transfer là một tập hợp các nguyên tắc kiến trúc cho các hệ thống hypermedia phân tán và lần đầu tiên được Roy Feilding vào năm 2000 trình bày trong luận án nổi tiếng của ông [<rodriguez2008restful>].

REST đã nổi lên trong những năm qua như một mô hình thiết kế dịch vụ Web chiếm ưu thế.

Khi hệ thống được chạy, để truy cập và cập nhật dữ liệu bên trong sổ kế toán, HC hỗ trợ phương thức REST. Ví dụ: getAvailbleProducts (thông tin tất cả các món hàng đang được rao bán) hay getAllTransactions (thông tin giao dịch toàn hệ thống) … Dữ liệu trả về dưới dạng JSON object.



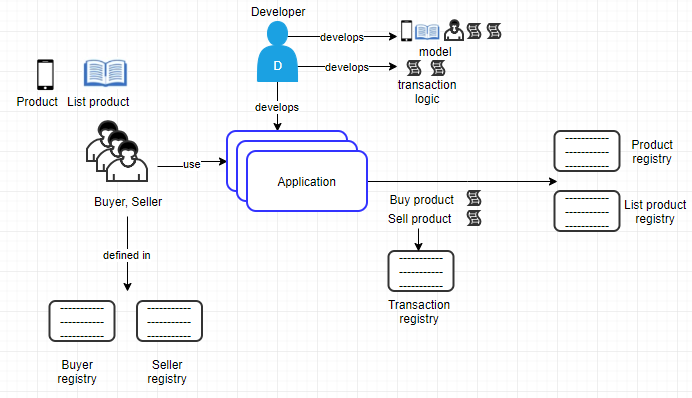


Figure 10: Mô hình tổng quan xây dựng ứng dụng

### Lưu trữ dữ liệu

Khi hệ thống được khởi chạy, tất cả dữ liệu sẽ được lưu trên Hyperledger Composer Fabric. Các dữ liệu bất biến như id các giao dịch thì được lưu trên blockchain database, còn các dữ liệu có thể sửa đổi như thông tin mặt hàng, người dùng thì được lưu trên world state database. Chi tiết về hai cơ sở dữ liệu này đã được nêu rõ ở phần 2.1.3.2 chương 2.

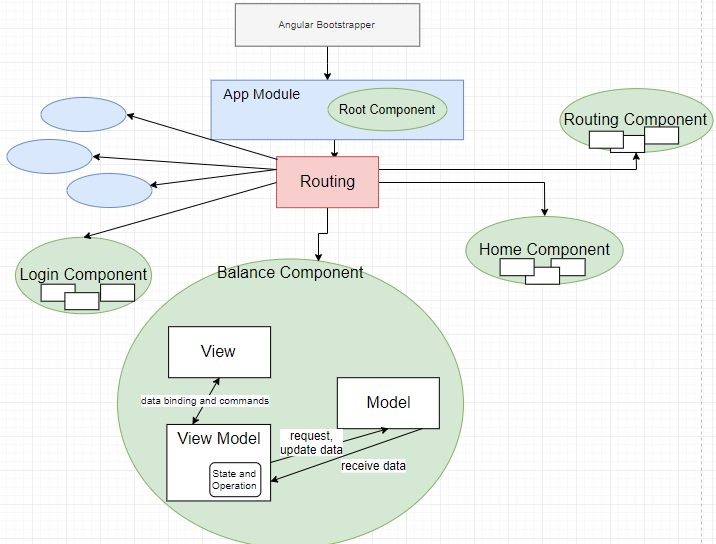
### Giao thức bảo mật

TLS – Transport layer security hay còn gọi là bộ giao thức tầng giao vận, phục vụ cơ bản cho bảo mật của World Wide Web. Cho phép các ứng dụng giao tiếp trên một mạng, phân phối trong các xác thực người dùng cuối và quyền truyền dẫn được đảm bảo [<gajek2008universally>].

## Client đa người dùng

Exchange App được phát triển trên framework Angular typescript.

### Cấu trúc

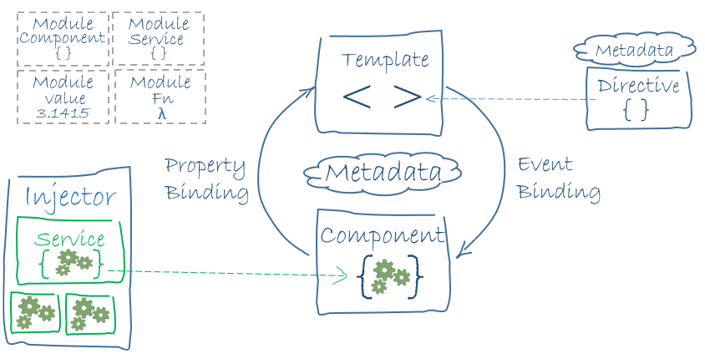


#### Component

Component là một thành phần không thể thiếu đối với mỗi một ứng dụng được xây dựng trên nền tảng Angular. Một ứng dụng Angular có thể chứa rất nhiều component khác nhau. Mỗi component bao gồm các chức năng như được liệt kê như phần <<TODO>> đính kèm với các mẫu HTML, các kiểu CSS được sử dụng bên trong. Các component được nạp từ một module gốc là module của toàn ứng dụng.

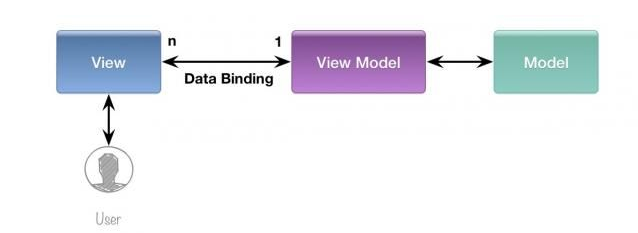
#### Dependency Injection

Trong Exchange App, mỗi một component thực hiện một chức năng riêng. Các chức năng đó được liệt kê ở phần <<TODO>>



#### Mô hình MVVM

Để có thể có dễ dàng thiết kế một cấu trúc code sạch và dễ tái sử dụng, em sử dụng mô hình MVVM.



View: Phần giao diện của ứng dụng để hiển thị dữ liệu và tương tác với người dùng.

Model: Các đối tượng truy xuất dữ liệu và thao tác dữ liệu.

View Model: Chứa các mã lệnh cần thiết thêm thực hiện điền dữ liệu vào view.

Mỗi một component trong ứng dụng có chứa một mô hình MVVM. Trong đó

Ràng buộc dữ liệu

Ràng buộc thuộc tính: Thuộc tính HTML có thể ràng buộc với giá trị của một trường trong component. Bất kỳ khi nào giá trị của thuộc tính thay đổi, nó sẽ tự động cập nhật giá trị của thuộc tính HTML mà nó bị ràng buộc. Ví dụ như khi người dùng click vào link đăng nhập trên màn hình

*<a href="https://localhost:3000/auth/facebook">Đăng nhập thông qua facebook</a>*

Từ khóa href trên tạo liên kết trong HTML.

Ràng buộc sự kiện: Sự kiện xảy ra trên bất kỳ phần tử HTML nào có thể ràng buộc. Phương thức này được gọi khi có bất kỳ sự kiện nào được kích hoạt bởi người dùng trên trang web. Như click mua các một sản phẩm trên hệ thống. Sau sản phẩm chuyển trang trạng thái giỏ hàng.

*<button mat-button (click)="addToCart(product.productId)">Thêm vào giỏ hàng</button>*

Ràng buộc hai chiều: <<TODO>>

#### Định tuyến

### Thao tác dữ liệu

Các thao tác liên quan tới dữ liệu server được ứng dụng client sử dụng công cụ HttpClient – cung cấp một API HTTP được đơn giản hóa, xây dựng một interface XMLHttpRequest giao tiếp với trình duyệt. Lợi ích khi sử dụng HttpClient là yêu cầu hay phản hồi dữ liệu tới server và cả xử lý lỗi dữ liệu.

### Giao thức bảo mật

Để tăng cường bảo mật về dữ liệu truyền tải thì em sử dụng giao thức HTTPS sử dụng thêm chứng chỉ SSL giúp mã hóa dữ liệu truyền tải nhằm bảo mật giữa Web server đến các trình duyệt web.

SSL là viết tắt của giao thức Secure Sockets Layer được Netscape phát triển và là giao thức Internet chuẩn để đảm bảo dữ liệu an toàn trên Internet [<bhiogade2002secure>].

Giao thức truyền tải HTTPS là giao thức truyền thông được thiết kế để truyền thông tin được mã hóa giữa các máy tính trên World Wide Web. HTTPS là một phiên bản HTTP nhưng an toàn hơn, bảo mật hơn, sử dụng giao thức bảo mật SSL.

### Hợp thức hóa dữ liệu

Cải thiện chất lượng dữ liệu tổng thể bằng cách hợp thức hóa các dữ liệu mà người dùng nhập vào, đảm bảo tính chính xác và đầy đủ.

Ví dụ với trường hợp điền thêm thông tin người dùng về số điện thoại, quê quán… sau khi đăng nhập thành công qua facebook (hay google, github). Để thêm các xác nhận hợp lệ vào biểu mẫu, em thêm các thuộc tính xác thực khớp với với thực tế như: số điện thoại không được để trống hay độ dài ngắn nhất là 10 số, quê quán có thể bắt buộc nhập hoặc không…

### Hiệu ứng, màu sắc

Ứng dụng sử dụng Javascript để tạo các hiệu ứng như trên thanh điều hướng, các item di chuyển.

Sử dụng framework bootstrap 4 để phát triển một responsive web. Hiệu chỉnh màu sắc, căn lề một component hay một nút bấm, hiệu ứng trỏ chuột thay đổi khi lướt qua các item…

## Client quản trị

### Tính năng

#### Danh sách mặt hàng trên hệ thống

#### Lịch sử giao dịch hệ thống

#### Danh sách khách hàng

#### Thông tin khách hàng và thông tin giao dịch liên quan

#### Sửa, cập nhật, xóa thông tin

### Giao thức bảo mật

### Hiệu ứng, màu sắc

## Công cụ phụ trợ

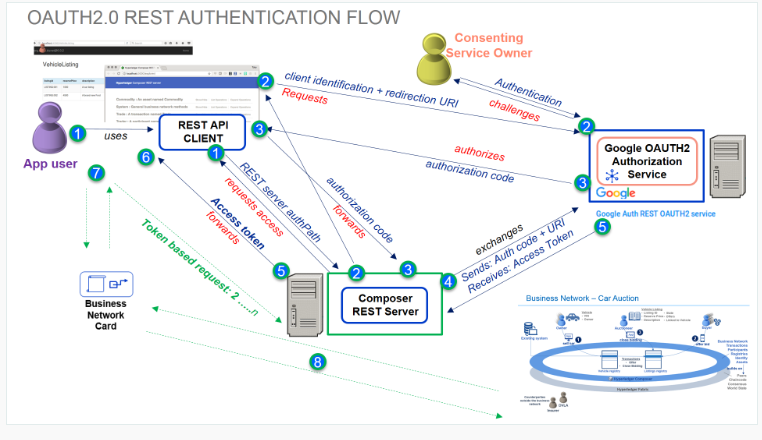
### **Cơ chế xác thực**

Máy chủ Hyperledger Composer REST tạo ra một tập các API chứa thông tin bên trong mạng. Các API này được các ứng dụng khách (client) gọi để tương tác với mạng Blockchain. Khi các máy khách gọi một trong số chúng, máy chủ REST sẽ gửi giao dịch tới mạng Blockchain. Giao dịch này được ký xác thực bằng một chứng chỉ để biết danh tính client nào đang phát sinh giao dịch. Máy chủ REST có thể được cấu hình để sử dụng xác thực. Điều này cho phép ứng dụng khách xác thực với máy chủ REST và sau đó máy chủ REST có thể phân biệt giữa mỗi máy khách. Tuy vậy, điều này vẫn không cho phép Blockchain có thể phân biệt các máy khách với nhau. Để cho phép máy chủ REST ký vào mỗi giao dịch một danh tính khác nhau cho mỗi máy khách được xác thực thì máy chủ cần phải chạy ở chế độ đa người dùng. Mỗi máy khách sau khi được chứng thực sẽ được cung cấp một tài khoản ví (wallet) trên máy chủ có chứa chi tiết danh tính và các cấu hình cần thiết để kết nối tới mạng Blockchain. Tài khoản ví này được sử dụng để ký các giao dịch.

Để có thể cài đặt máy chủ REST ở chế độ đa người dùng thì em sử dụng cơ chế xác thực OAuth - phương thức chứng thực, ủy quyền cho phép bên thứ ba ứng dụng truy cập vào tài nguyên người dùng nằm trên một dịch vụ nào đó, hoặc thay mặt chủ sở hữu tài nguyên bằng cách dàn xếp tương tác phê duyệt giữa chủ sở hữu tài nguyên và dịch vụ HTTP [<hardt2012oauth>].

Điều này cho phép từng người trong mạng, với ủy quyền của mình được phép truy cập những tài nguyên được phép và tương tác với triển khai những hợp đồng thông minh trong hệ thống mạng. Ứng dụng khách viết bằng Angular và sẽ sử dụng ba API của ba ứng dụng cung cấp ứng thực đó là Google+, Github và Facebook. Ba kiểu xác thực này thực chất tương đồng nhau nên em sẽ nêu cụ thể và rõ nhất về một hình thức, ở đây là xác thực qua Facebook.

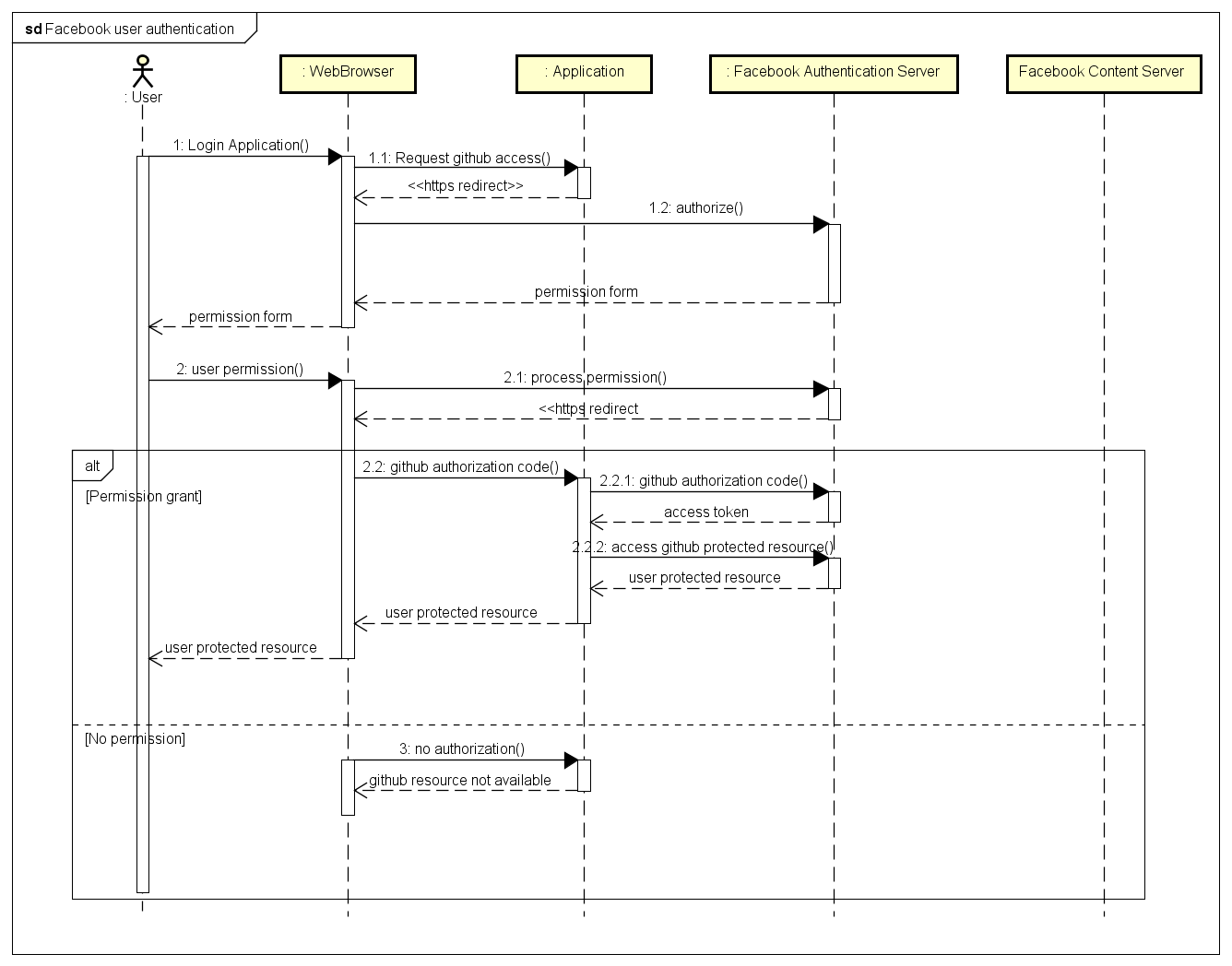
Máy chủ REST sử dụng một thư viện mã nguồn mở có tên là passport-facebook.



Đầu tiên ứng dụng khách sử dụng API chứa đường dẫn xác thực trên hệ thống máy chủ REST (bước 1). Hệ thống sẽ gửi định danh của ứng dụng khách và đường dẫn điều hướng (được thiết lập từ trước) dẫn tới máy chủ xác thực của Facebook. Ở đây nó sẽ yêu cầu chúng ta vào facebook để xác thực. Máy chủ tính toán nếu đúng là người dùng facebook đó xác thực thì sẽ gửi lại cho bên máy chủ REST mã xác thực đính kèm thông qua cookie trình duyệt. Bên máy chủ sau khi nhận mã xác thực thì gửi mã xác thực kèm URI (đường dẫn sau khi xác thực thành công) đến máy chủ facebook. Đến đây quá trình xác thực thành công, máy chủ Facebook gửi cho máy chủ REST một access-token (đoạn mã sinh ra ngẫu nhiên được sử dụng bí mật cho mỗi người dùng). Access-token được coi như một chiếc chìa khóa để mở kho báu bên trong. Khi người dùng chưa được xác thực thì các api của hệ thống máy chủ REST không cho phép truy cập đối với từng người dùng cụ thể. Khi người dùng đã được xác thực, nhận được access-token gửi từ máy chủ Facebook thì các API này sử dụng access-token mới có thể truy cập dữ liệu. Việc này đảm bảo quyền riêng tư dữ liệu, bảo mật thông tin.

Lưu ý, với lần đăng nhập đầu tiên với hệ thống thì sau khi đã xác thực tài khoản với máy chủ Facebook thì hệ thống REST server lấy thông tin người dùng tạo ra một thẻ định danh và xuất file này vào hệ thống để tạo một tài khoản ví người dùng trong mạng blockchain. Với những lần đăng nhập sau thì chỉ cần xác thực qua facebook thì bước tạo file và xuất vào hệ thống được bỏ qua.

Chi tiết hơn về luồng thực thi thì ta có thể xem biểu đồ sau:



# Triển khai

## Chuẩn bị

(Chú ý: Các bước cài đặt được viết bên dưới có tham khảo từ trang <https://hyperledger.github.io>)

### Cài đặt

#### Công cụ tiên quyết

* Hệ điều hành: Ubuntu Linux 18.04.1 LTS.
* Docker Engine: phiên bản …
* Docker-Compose: phiên bản …
* Nodejs phiên bản …
* Npm …
* Git: …
* Python: …
* IDE: Visual Studio Code

#### **Cài đặt Hyperledger Composer**

Cài đặt công cụ CLI thao tác trên terminal của ubuntu:

* composer-cli: công cụ cần thiết nhất bao gồm tất cả các thao tác cần thiết trên terminal. Cài qua dòng lệnh

*sudo npm install -g* [*composer-cli@0.20*](mailto:composer-cli@0.20)

* composer-rest-server: tiện ích để máy chủ REST hiển thị thông tin của ứng dụng dưới dạng API RESTful. Dùng lệnh:

*sudo npm install -g composer-rest-server@0.20*

* generator-hyperledger-composer: tiện ích giúp tự động. Cài đặt qua lệnh:

*sudo npm install -g* [*generator-hyperledger-composer@0.20*](mailto:generator-hyperledger-composer@0.20)

* yeoman: Công cụ tạo tạo khung ứng dụng đơn giản, từ đó em có thể sử dụng để phát triển ứng dụng riêng của mình. Công cụ này sử dụng tiện ích của generator-hyperledger-composer. Có thể coi như yeoman như là một bản thiết kế sơ bộ của một ngôi nhà, mà trong đó từng chi tiết nhỏ như mái hay cửa sổ, cửa chính... được tạo nên từ generator-hyperledger-composer. Yeoman được cài đặc thông qua lệnh:

*sudo npm install -g yo*

#### Cài đặt Hyperledger Fabric

Sử dụng các lệnh sau để cài đặt một Hyperledger Fabric chạy trên nền local máy.

* Tạo thư mục lưu (giả sử: ~/fabric-dev-servers), lấy tập tin nén chứa các công cụ cần thiết và giải nén vào thư mục lưu.

*mkdir ~/fabric-dev-servers && cd ~/fabric-dev-servers*

*curl -O* [*https://raw.githubusercontent.com/hyperledger/composer-tools/master/packages/fabric-dev-servers/fabric-dev-servers.tar.gz*](https://raw.githubusercontent.com/hyperledger/composer-tools/master/packages/fabric-dev-servers/fabric-dev-servers.tar.gz)

*tar -xvf fabric-dev-servers.tar.gz*

* Sử dụng những file scripts bên trong thư mục ~/fabric-dev-servers để tiến hành download phiên bản Hyperledger Fabric v1.2

*cd ~/fabric-dev-servers*

*export FABRIC\_VERSION=hlfv11*

*./downloadFabric.sh*

#### Cài đặt Angular

* Angular CLI: công cụ để tạo một project đơn giản tạo mã ứng dụng và thư viện và thực hiện một loạt các nhiệm vụ phát triển liên tục như kiểm tra, đóng gói và triển khai.

*npm install -g @angular/cli*

### Phát triển ứng dụng

Thương mại điện tử là một mạng lưới kinh doanh liên quan tới:

* Hàng hóa, sản phẩm như điện thoại, quần áo, rau, củ, quả...
* Bên kinh doanh: người mua, người bán.
* Thỏa thuận giữa các bên kinh doanh.

Dựa vào mạng lưới này, em sẽ xây dựng một sàn giao dịch.

#### Server extenstion

Đầu tiên, sử dụng Yeoman để tạo sơ lược cấu trúc ứng dụng.

*yo hyperledger-composer:businessnetwork*

Tiếp đến trong giao diện terminal nhập tên ứng dụng là **exchange** và điền các thông tin cần thiết như description, author name, email…

Khi khung project sau khi tạo bao gồm tệp mô hình (file có đuôi .cto) và chứa tài liệu truy cập quyền truy cập (file premissions.acl), tệp tập lệnh logic (logic.js) chứa các chức năng xử lý giao dịch và tệp các lệnh truy vấn cơ sở dữ liệu (queries.qry) và tệp package.json chứa meta data của ứng dụng.

##### Model

Mô hình mạng lưới ứng dụng được mô tả dưới file với định dạng .cto, được tập chung chặt chẽ với một số từ khóa: namespace, resouces, import dữ liệu từ namespace khác.

**Namespace**: không gian phạm vi mà mọi thứ khai báo bên trong file model này được phép sử dụng và thay đổi tùy logic nghiệp vụ. Cũng như namespace trong lập trình môn C++ thì namespace trong hyperledger composer là duy nhất. Ứng dụng **exchange** của em có namespace ở đây là uet.khoenguyen.exchange.

**Resource**: Chứa tất cả thông tin, các đối tượng trong hệ thống: assets, participants, transaction, enum type, concept.

**Enum**:

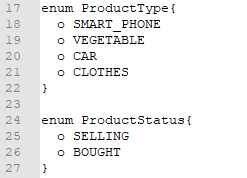
* ProductType: các loại sản phẩm

|  |  |
| --- | --- |
| ProductType | Mô tả |
| SMART\_PHONE | Điện thoại thông minh |
| VEGETABLE | Rau, củ, quả |
| CAR | Xe ô tô |
| CLOTHES | Quần áo |

* ProductStatus: tình trạng sản phẩm

|  |  |
| --- | --- |
| ProductStatus | Mô tả |
| SELLING | Đang được bày bán |
| BOUGHT | Đã được mua |

Triển khai:

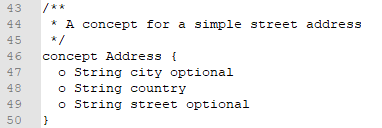


Hình 7: Hình mô tả các enum

**Concept**

|  |  |  |
| --- | --- | --- |
| Address | Kiểu dữ liệu | Mô tả |
| country | String | Tên quốc gia |
| city | String | Tên thành phố, tỉnh |
| street | String | Tên phường, huyện |

Triển khai:

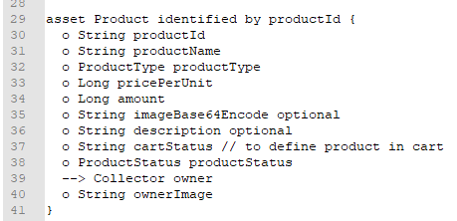


**Asset**

Trong ứng dụng, asset ở đây là những món hàng mà người dùng muốn trao đổi mua hoặc bán được trừu tượng hóa với tên Product.

|  |  |  |
| --- | --- | --- |
| Thuộc tính của Product | Kiểu dữ liệu | Mô tả |
| productName | String | Tên mặt hàng |
| productType | String | Loại sản phẩm |
| pricePerUnit | Long | Giá trên từng đơn vị sản phẩm (đơn vị VNĐ) |
| amount | Long | Số lượng |
| imageBase64Encode | String | Mã hóa base64 của hình ảnh mặt hàng |
| description | String | Mô tả mặt hàng |
| cartStatus | String | Tình trạng mặt hàng trong giỏ hàng |
| productStatus | ProductStatus | Tình trạng mua, bán mặt hàng |
| owner | Collector | Chủ mặt hàng |
| ownerImage | String | Ảnh chủ mặt hàng |

Triển khai:



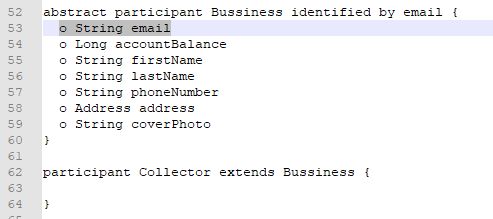
Asset Product này được định danh bởi productId, productId này là duy nhất có chức năng định danh cho mỗi sản phẩm được đăng lên hệ thống.

**Participant**

Participant bao gồm người mua hàng, người bán hàng, được trừu tượng hóa với tên là Collector.

|  |  |  |
| --- | --- | --- |
| Thuộc tính của Collector | Kiểu dữ liệu | Mô tả |
| email | String | Email người dùng login qua github |
| accountBalance | Long | Số dư tài khoản |
| firstName | String | Tên người dùng |
| lastName | String | Họ người dùng |
| phoneNumber | String | Số điện thoại người dùng |
| address | Address | Địa chỉ người dùng |
| coverPhoto | String | Mã hóa base64 của ảnh đại diện người dùng |

Triển khai:



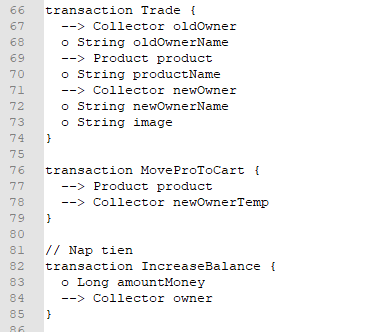
Ở đây em ứng dụng đặc tính kế thừa trong lập trình hướng đối tượng. Khai báo một participant trừu tượng Bussiness được định danh qua email. Participant kế thừa kế thừa những đặc tính từ Bussiness. Khai báo như này để đảm bảo sau này phát triển ứng dụng dễ khả rộng hơn và tối ưu code.

**Transactions**

Có một số transaction như sau: chuyển nhượng mặt hàng khi giao dịch mua bán thành công (Trade), nạp tiền (IncreaseBalance), chuyển sản phẩm sang giỏ hàng (MoveProToCart).

|  |  |  |
| --- | --- | --- |
| Thuộc tính của Trade | Kiểu dữ liệu | Mô tả |
| oldOwner | Collector | Chủ sở hữu cũ của sản phẩm |
| oldOwnerName | String | Tên chủ sở hữu cũ của sản phẩm |
| product | Product | Sản phẩm được trao đổi |
| productName | String | Tên sản phẩm được trao đổi |
| newOwner | Collector | Chủ sở hữu mới của sản phẩm |
| newOwnerName | String | Tên chủ sở hữu mới của sản phẩm |
| image | String | Mã hóa base64 của ảnh sản phẩm |

Triển khai:

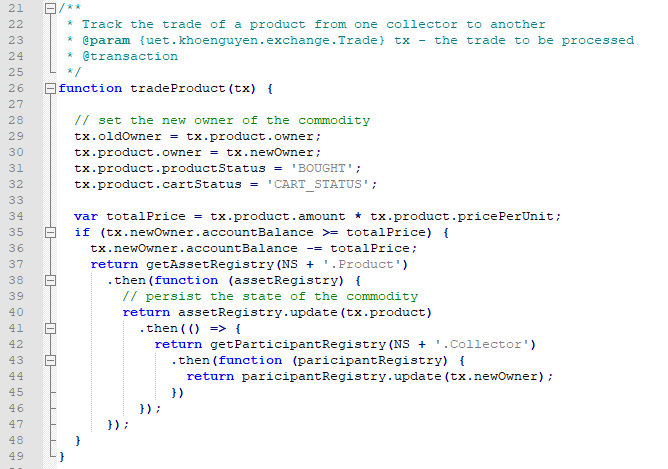


##### Script file

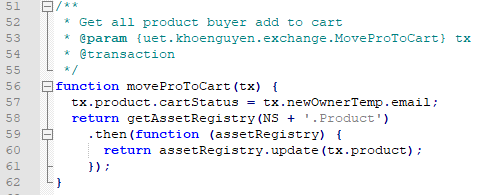
Tiến hành cập nhật trên file logic.js.

Ứng với mỗi transaction, em định nghĩa một function tương ứng. Trong mỗi function xử lý các câu lệnh logic liên quan tới nghiệp vụ liên quan.

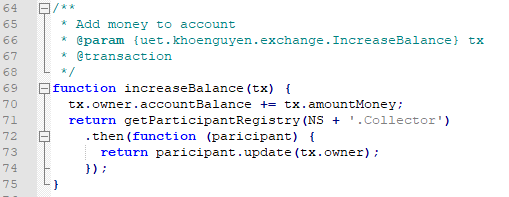
Với transaction Trade, người đứng tên sản phẩm mới sẽ được cập nhật trên dữ liệu sản phẩm, tình trạng sản phẩm chuyển từ đang được bày bán sang đã được mua.



Với transaction MoveToCart, thì trong function tương ứng thì trạng thái của sản phẩm sẽ được cập nhật.



Với transaction IncreaseBalance, thì trong function tương ứng tài khoản của người nạp sẽ tang lên tương ứng với giá trị thẻ nạp vào.

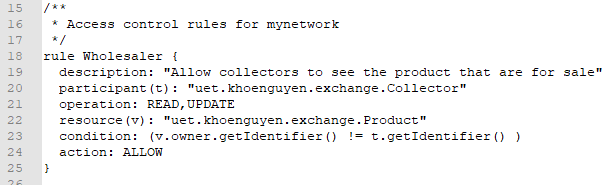


##### Phân quyền

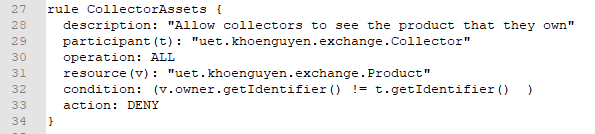
Tiến hành cập nhật file permissions.acl.

Khai báo một số quyền truy cập dữ liệu đối với mỗi người dùng khác nhau.

* Wholesaler: Cho phép mọi người dùng được phép xem những sản phẩm đang được bày bán, trừ những sản phẩm mình đăng lên sàn giao dịch. Người dùng chỉ có quyền đọc dữ liệu (READ), với điều kiện là chủ sở hữu sản phẩm khác với người dùng đang sử dụng.



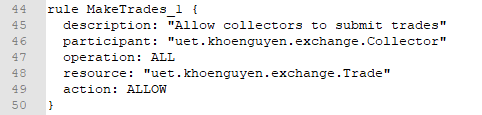
* CollectorAssets: Chặn không cho phép người dùng khác có thể sửa thông những sản phẩm mà họ không sở hữu. Chặn tất cả các quyền CREAT, READ, UPDATE, DELETE.



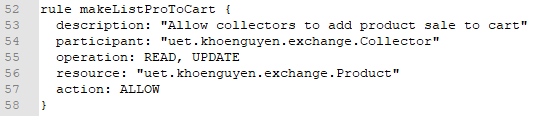
* ProductViews: Cho phép người dùng được tạo, xem và sửa thông tin sản phẩm mà họ sở hữu. Người dùng có tất cả các quyền truy cập dữ liệu: CREAT, READ, UPDATE, DELETE.



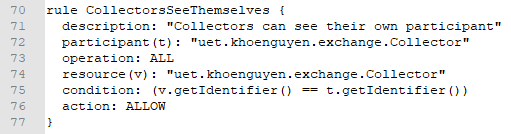
* MakeTrade\_1: Cho phép người dùng tạo, xem phiên giao dịch mua sản phẩm.



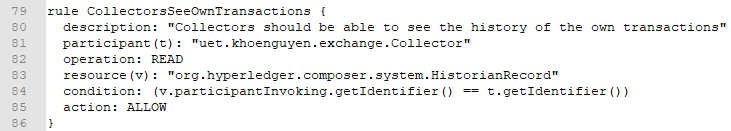
* makeListProToCart: Cho phép người dùng xem và thêm sản phẩm đang được bày bán vào giỏ hàng. Người dùng truy cập dữ liệu với quyền READ, UPDATE.



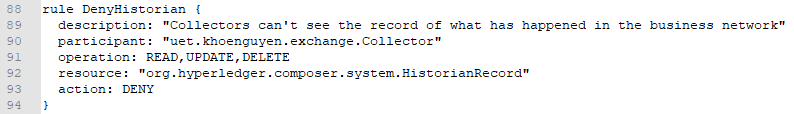
* CollectorsSeeThemeselves: Cho phép người dùng tạo, xem, cập nhật hay xóa thông tin của mình trên ứng dụng. Truy cập dữ liệu với quyền CREATE, READ, UPDATE, DELETE.



* CollectorsSeeOwnTransactions: Mọi người dùng đều có quyền xem tất cả thông tin các phiên giao dịch.



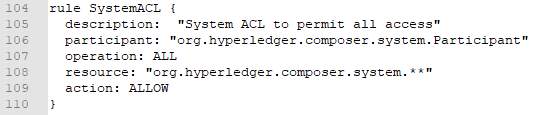
* DenyHistorian: Người dùng (không phải quản trị viên) sẽ không thể xem các bản ghi lịch sử của hệ thống.



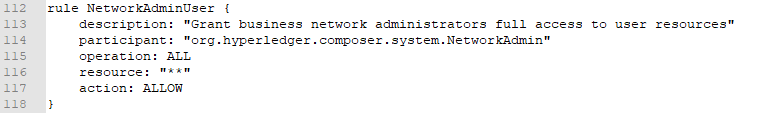
* NetworkUpdate: Cho phép hệ thống cập nhật tất cả các quyền truy cập.



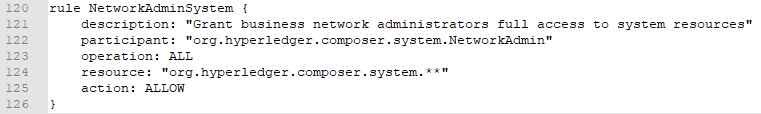
* SystemACL: Cho phép hệ thống cập nhật tất cả các quyền truy cập.



* NetworkAdminUser: Người dùng admin có quyền truy cập tất cả dữ liệu người dùng.



* NetworkAdminSystem: Người dùng admin có quyền truy cập tất cả dữ liệu hệ thống.



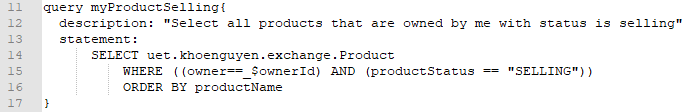
##### Truy vấn dữ liệu

Khai báo các lệnh sql query phục vụ việc tìm kiếm những sản phẩm đang trong tình trạng còn bán, đã được mua, những sản phẩm mà người dùng đã mua hoặc đang bán, danh sách các mặt hàng trong giỏ hàng...

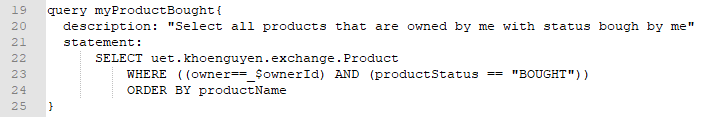
* availableProducts: Lấy tất cả các sản phẩm đang được bày bán trừ sản phẩm mà người dùng đang bán.



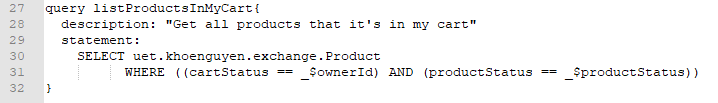
* myProductSelling: Lấy tất cả các sản phẩm mà người dùng đang bày bán.



* myProductBought: Lấy tất cả các sản phẩm mà người dùng đã mua.



* listProductsInMyCart: Lấy tất các các sản phẩm trong giỏ hàng người mua.



|  |  |  |  |
| --- | --- | --- | --- |
| Participant | Participant accessed | Assets accessed | Transactions accessed |
| Người mua | Người mua ® | Product (RU) | Wholesaler, CollectorAssets,  ProductViews,  MakeTrade\_1,  makeListProToCart,  CollectorsSeeThemeselves,  CollectorsSeeOwnTransactions |
| Người bán | Người bán ® | Product © |  |
| Admin | ALL (CRUD) | ALL (CRUD) | ALL (CRUD) |

#### Client đa người dùng

Tạo workspace vào khởi tạo ứng dụng

*ng new exchange-client-multi*

Sau khi hệ thống khởi tạo xong, truy cập vào thư mục workspace (exchange-client-multi).

Việc khởi tạo khung xương project đã xong. Tiếp theo, tiến hành chỉnh sửa phát triển ứng dụng.

Sử dụng thư viện @angular material, cdk, flex-layout… để tạo các pop-up đăng nhập, các card hiển thị thông tin, form điền thông tin…

##### Component

Ứng dụng chia làm nhiều component khác nhau, mỗi component là một phần của màn hình, đảm nhiệm thể hiện các chức năng tương ứng:

* App component – view tổng thể của cả ứng dụng
* Login component: pop-up hiển thị khi người dùng click button đăng nhập.
* Routing component: Thanh điều hướng giữa các component.
* Home component: Trang chủ hiện danh sách các mặt hàng đang được bày bán.
* Sell component: Màn hình đăng bán sản phẩm.
* History component: Màn hình lịch sử giao dịch.
* Cart component: Giao diện giỏ hàng, chứa những mặt hàng người dùng chọn và sắp mua.
* Balance component: Màn hình chức năng nạp tiền.
* Account component: Giao diện hiển thị thông tin người dùng, các mặt hàng đã mua, đã bán.

Tạo các component qua lệnh

*ng generate component <<tên component>>*

Một component đầy đủ các file cầu hình cần thiết được tạo: một file module (import thư viện, thiết lập logic), một file css (thêm các đặc tính màu sắc, animation chuột, chỉ mục…), một file html.

##### Services

Là các giá trị hay thực hiện các chức năng mà ứng dụng cần. Là cầu nối trung gian giữa kết nối ứng dụng exchange server extension (back-end) với ứng dụng exchange-client-multi (font-end). Lấy dữ liệu từ server nạp vào ứng dụng và thiết lập những request tạo, xóa, cập nhật dữ liệu trong server.

Hỗ trợ tiêm phụ thuộc (dependency injection)

##### Routing

##### Data binding

##### HTTPS

* **Giao diện khách**

Giao diện khi người dùng lần đầu truy cập ứng dụng. Giao diện yêu cầu có một button “Đăng nhập”. Khi click button này thì một pop up hiển thị lên cho phép người dùng đăng nhập thông qua các tài khoản khác nhau như: facebook, google, github.

Bắt đầu từ component app của ứng dụng. Tạo background và chèn các thuộc tính màu sắc, animation trong app.component.css.

Giao diện pop up và xử lý sự kiện người dùng click vào các chỉ mục tương ứng được xử lý trong một component mới - component login. Tạo component login qua câu lệnh

Khai báo luồng logic trong app.component.ts cho trường hợp giao diện khách: tạo một hàm openLoginPopup() có chức năng gọi tới component login (sử dụng thư viện MatDialog trong thư viện @angular/material).

Với mỗi mục lựa chọn đăng nhập qua facebook, google, github có một đường URL xác thực tương ứng được gọi khi người dùng click vào

/auth/facebook, /auth/google, /auth/facebook.

* **Giao diện khi đã đăng nhập**

Với trường hợp người dùng truy cập ứng dụng lần đầu thì sẽ hiện ra một form điền thêm thông tin người dùng như: họ tên, số điện thoại, địa chỉ, ảnh đại diện… được khai báo trong một section của app.component.html.

Việc người dùng thêm các thông tin vào và bấm button “Submit” thì tất cả thông tin

Trường hợp người dùng đã đăng nhập ứng dụng vào thời gian trước đây thì tiến hành vào thẳng trang chủ dành cho người dùng bình thường.

#### Client quản trị

## Quy trình khởi tạo, chạy ứng dụng

### Khởi chạy Server

#### Khởi chạy Hyperledger Fabric

cd $COMPOSER\_ROOT/fabric-dev-servers

sudo ./startFabric.sh

./downloadFabric.sh

./createPeerAdminCard.sh

Lệnh đầu tiên sẽ mất vài phút, cung cấp tất cả các thứ cần thiết vào môi trường Docker. Lệnh thứ hai để khởi chạy một Hyperledger Fabric cục bộ. Lệnh cuối cùng tạo một ID Cards cho Fabric admin – PeerAdminCard.

#### Đóng gói Hyperledger Composer server extension và kết nối tới Hyperledger Fabric

Triển khai exchange-app vào Hyperledger Fabric cục bộ, sử dụng Composer CLI và thi hành câu lệnh sau:

*cd $COMPOSER\_ROOT/Exchange/client/exchange-app*

*composer archive create -t dir*

*composer network install -a exchange@0.0.2.bna -c PeerAdmin@hlfv1 -n .*

*composer network start -n exchange -V 0.0.2 -A admin -S adminpw -c PeerAdmin@hlfv1*

*composer card import -f* [*admin@exchange.card*](mailto:admin@exchange.card)

Nén project thành một file BNA.

Lệnh thứ hai được sử dụng để cài đặt file nén BNA từ lệnh đầu tiên vào mạng Hyperledger Fabric cục bộ.

Lệnh tiếp theo, tạo một phiên bản cụ thể của mạng đã được cài đặt ở lệnh trước đó vào mạng Hyperledger Fabric. Đồng thời câu lệnh này tạo ra một thẻ admin mới. Thẻ này nén hai file connection.json và metadata.json, chứa danh tính admin và tất cả các thông tin cần thiết để kế nối tới mạng blockchain.

Ở lệnh cuối cùng, lệnh này tiến hành import thẻ admin vào ví blockchain local.

Chúng ta cũng có thể tiến hành xem các card đã được import thành công hay chưa qua lệnh:

*composer card list*

#### Khởi chạy Hyperledger composer rest server

* Server dành cho admin

*cd HyperledgerComposers/ ExchangeApp/server/exchange/*

*composer-rest-server -c admin@exchange -p 3001 -n never*

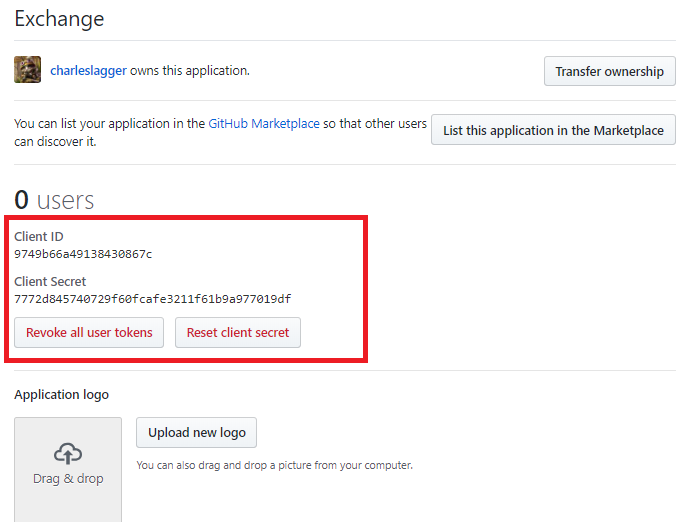
Server này được mở ở cổng 3001, sử dụng thẻ admin để kết nối tới ví và api được generate ra sẽ không bao gồm namespace như được khai báo trong model.

* Server đa người dùng

Trước tiên đăng nhập vào github, sau đó truy cập vào link <https://github.com/settings/developers>, tiến hành tạo một ứng dụng OAuth trên github điền các thông tin như hình sau



Bấm Register application và màn hình chuyển sang giao diện sau



Lưu ý vùng khoanh đỏ, copy client id và client secret để tiến hành các câu lệnh tiếp theo.

*cd HyperledgerComposers /Exchange/server/exchange/*

*export COMPOSER\_PROVIDERS='{"github":{"provider":"github","module":"passport-github","clientID":"9749b66a49138430867c","clientSecret":" 7772d845740729f60fcafe3211f61b9a977019df","authPath":"/auth/github","callbackURL":"/auth/github/callback","successRedirect":"http://localhost:4200?loggedIn=true","failureRedirect":"/"}}'*

*echo "Starting composer rest server for multiple clients at port 3000"*

*composer-rest-server -c admin@exchange -m true -n never*

Câu lệnh đầu tiên chúng ta export COMPOSER\_PROVIDERS vào Docker image, lệnh này chứa clientID và clientSecret chúng ta lấy từ github OAuth, đồng thời cung cấp các url chuyển tiếp khi xác thực github thành công hoặc thất bại.

Câu lệnh tiếp theo khởi chạy server ở chế độ đa người dùng, đầu tiên cũng sử dụng thẻ admin để kết nối tới ví và không sử dụng namespace trong các api được generate ra. Sau này khi người dùng nào đó đăng nhập vào Exchange App thì hệ thống cũng sẽ tự động generate một card tương ứng và duy nhất chứa thông tin của người dùng đó.

### Khởi chạy client app đa người dùng

*cd HyperledgerComposers/Exchange/client/exchange-app*

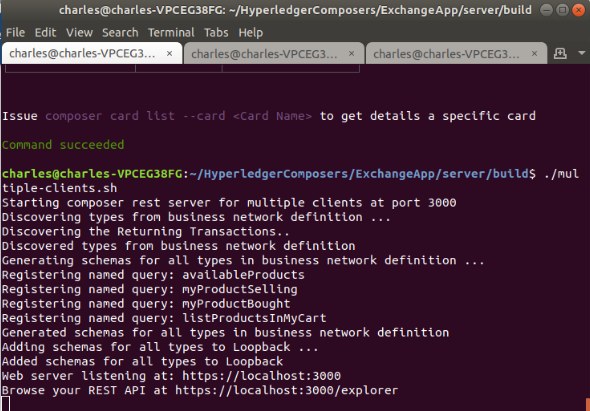
*ng serve –ssl true*

### Khởi chạy client app admin

## Hiển thị kết quả

### Server extension

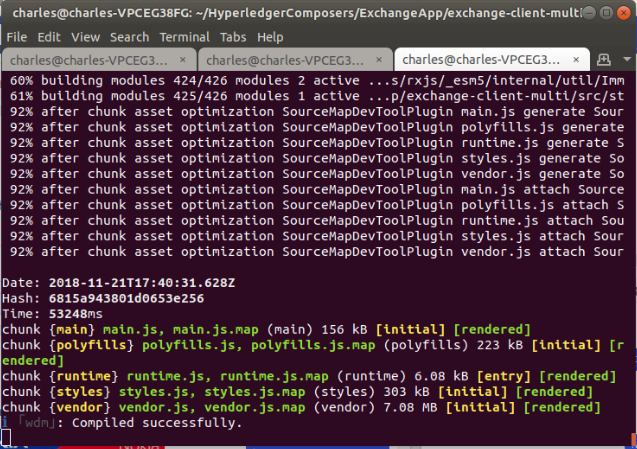
Khởi chạy server thành công

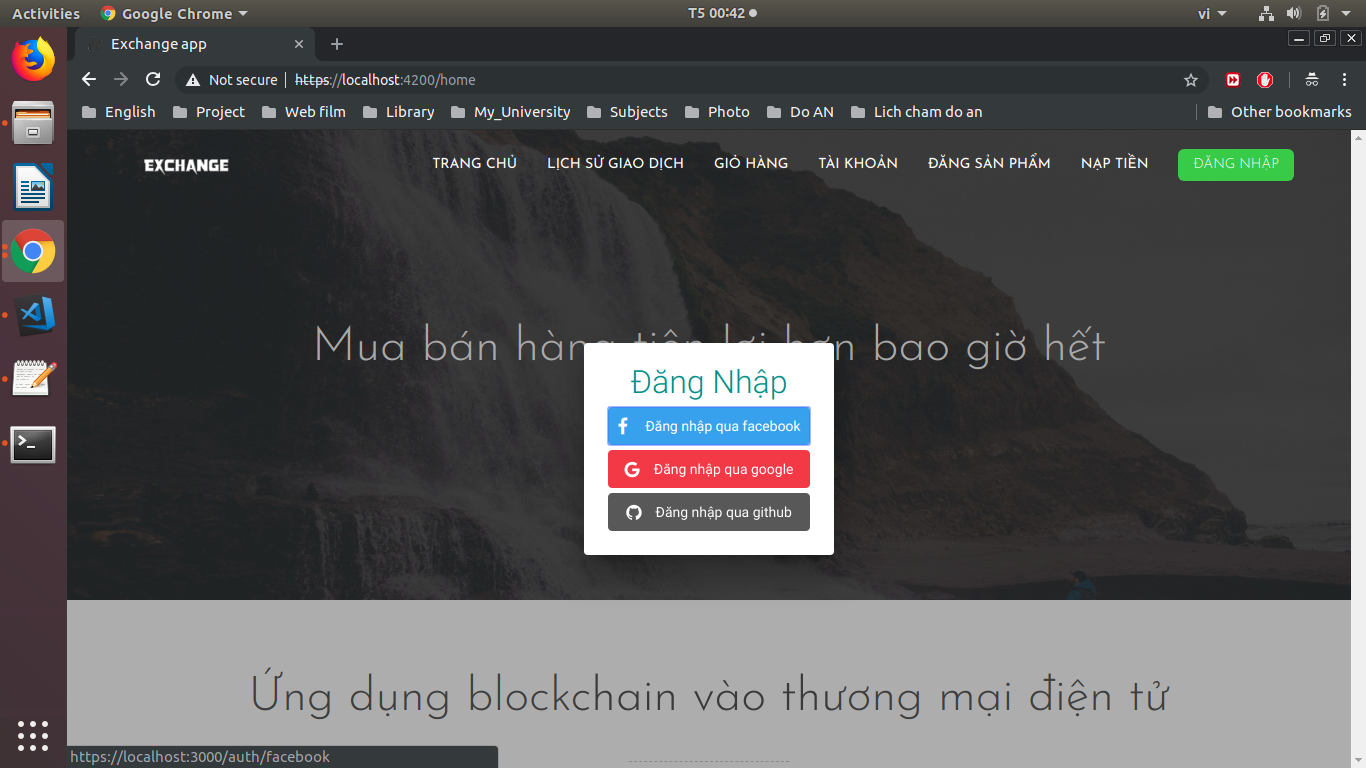


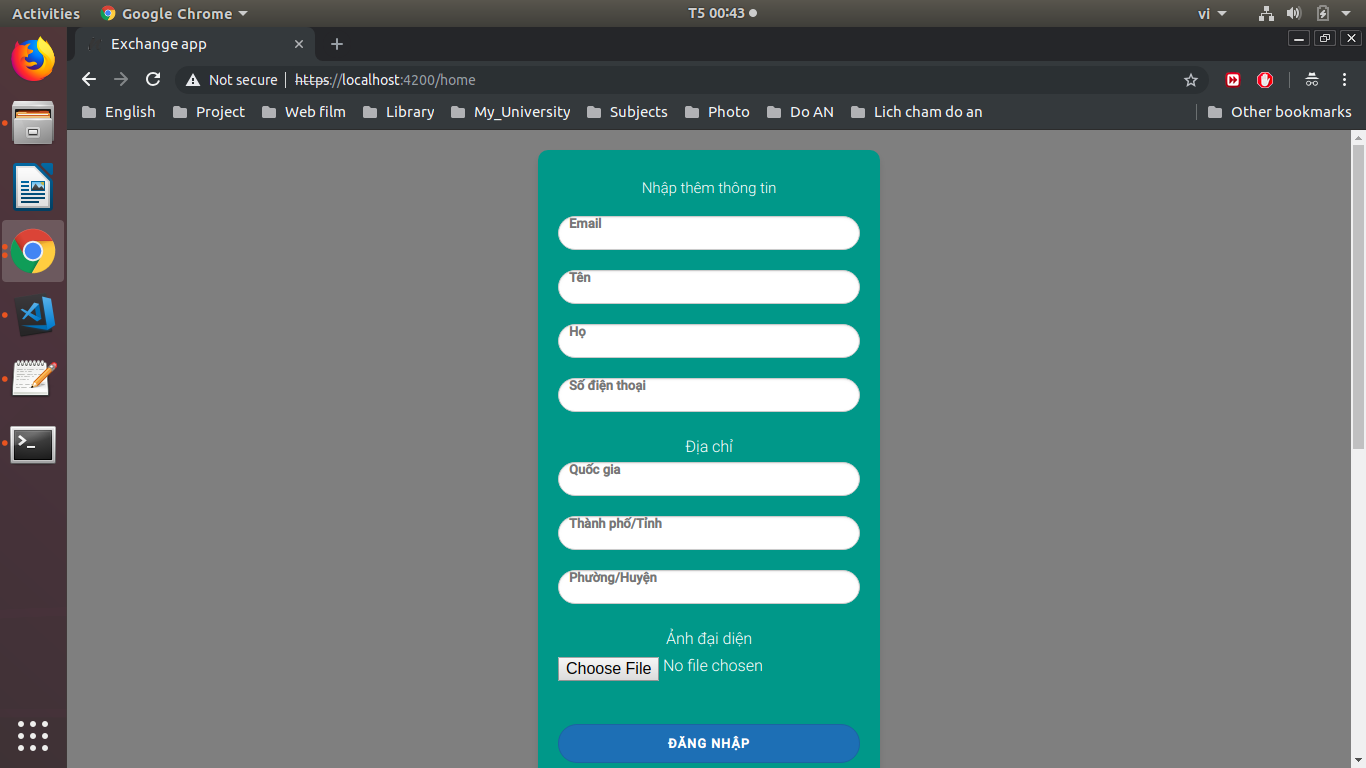


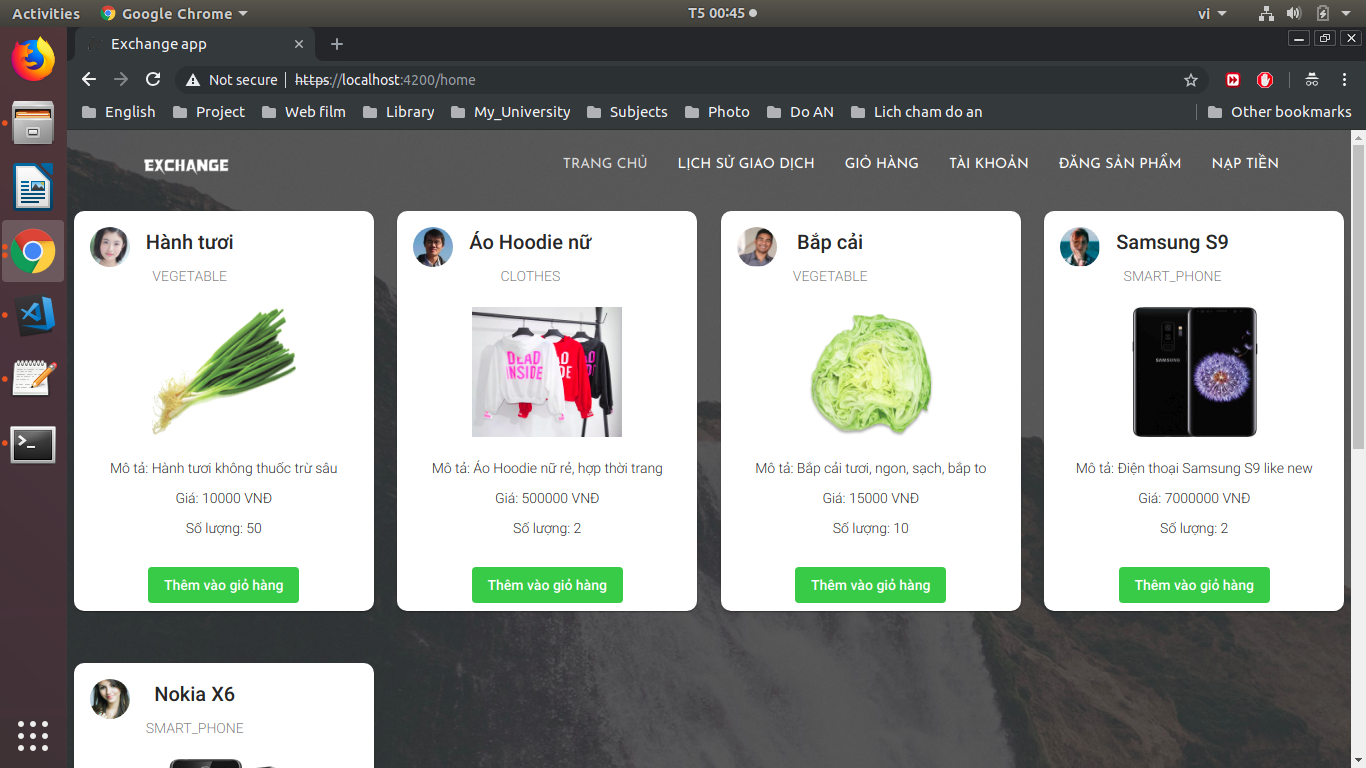
### Client đa người dùng

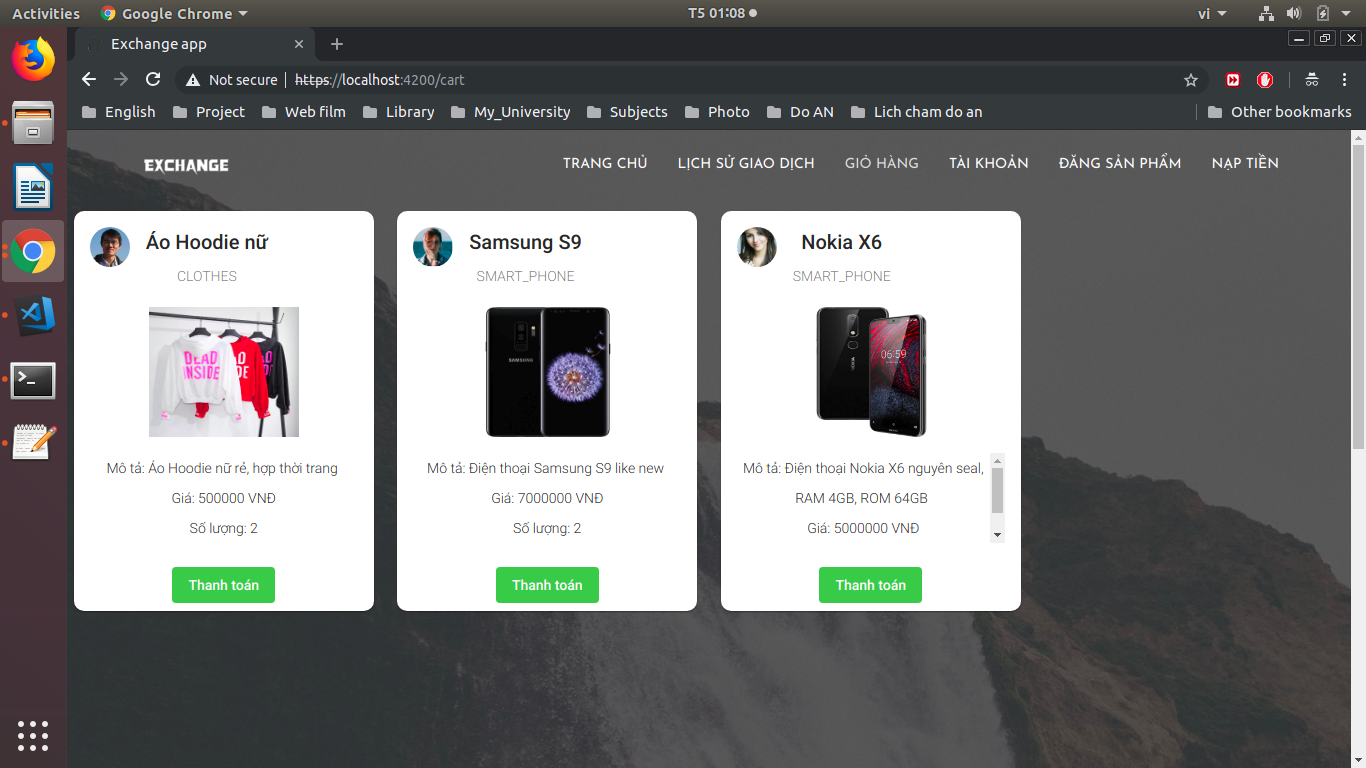
Khởi chạy ứng dụng thành công

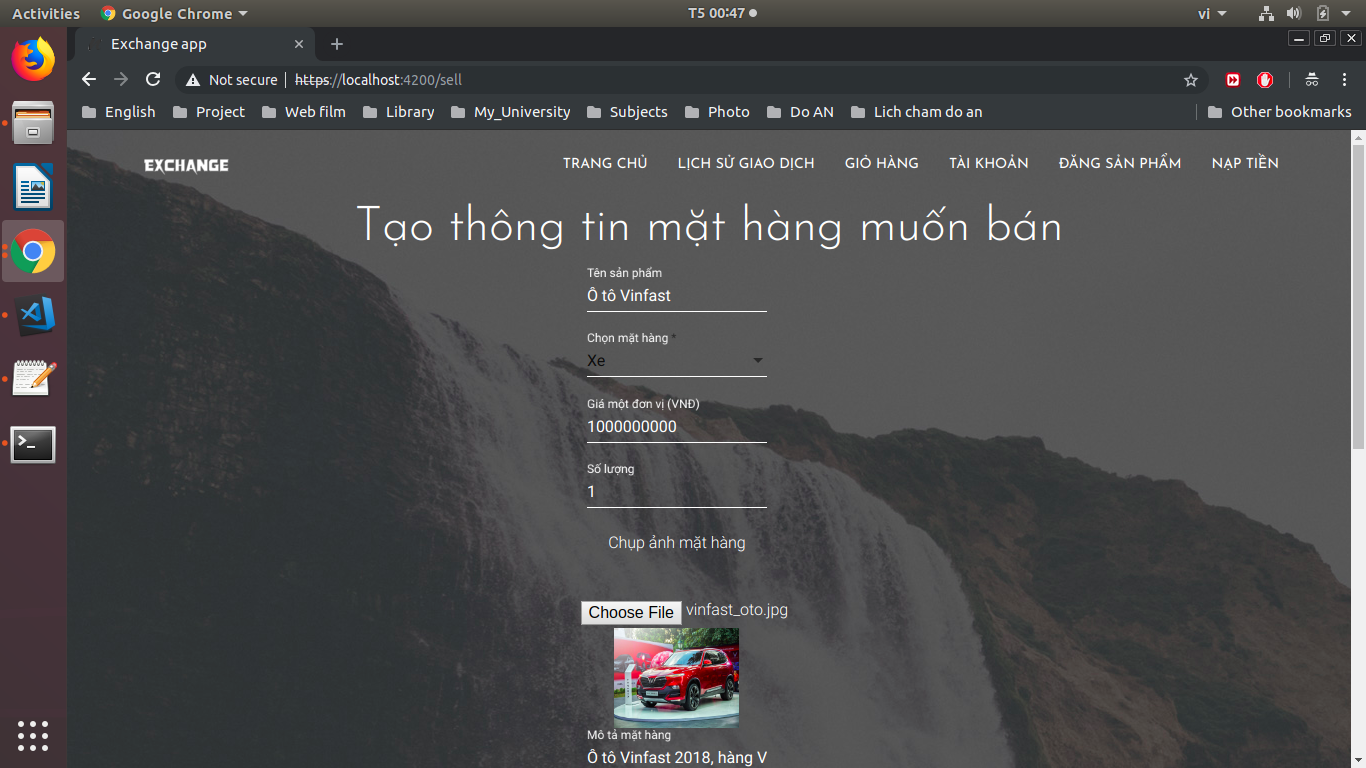


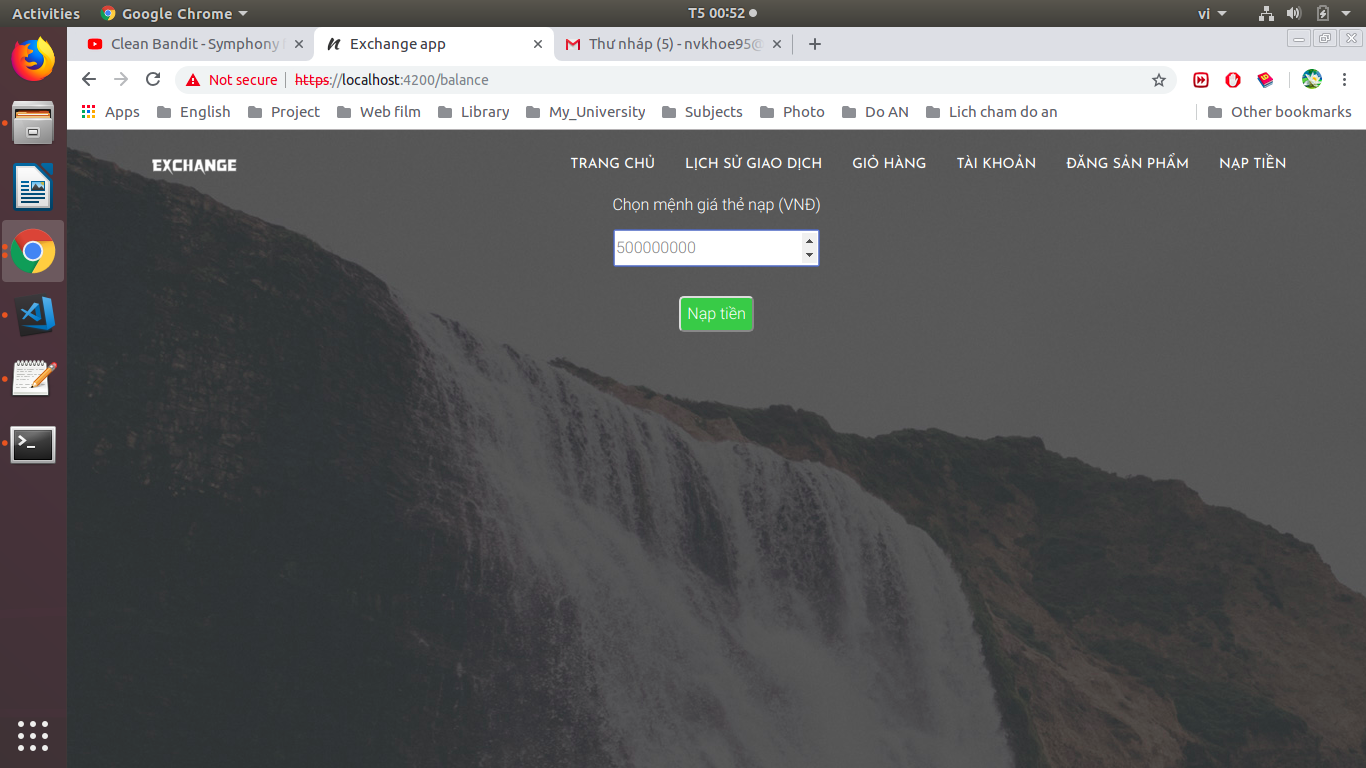


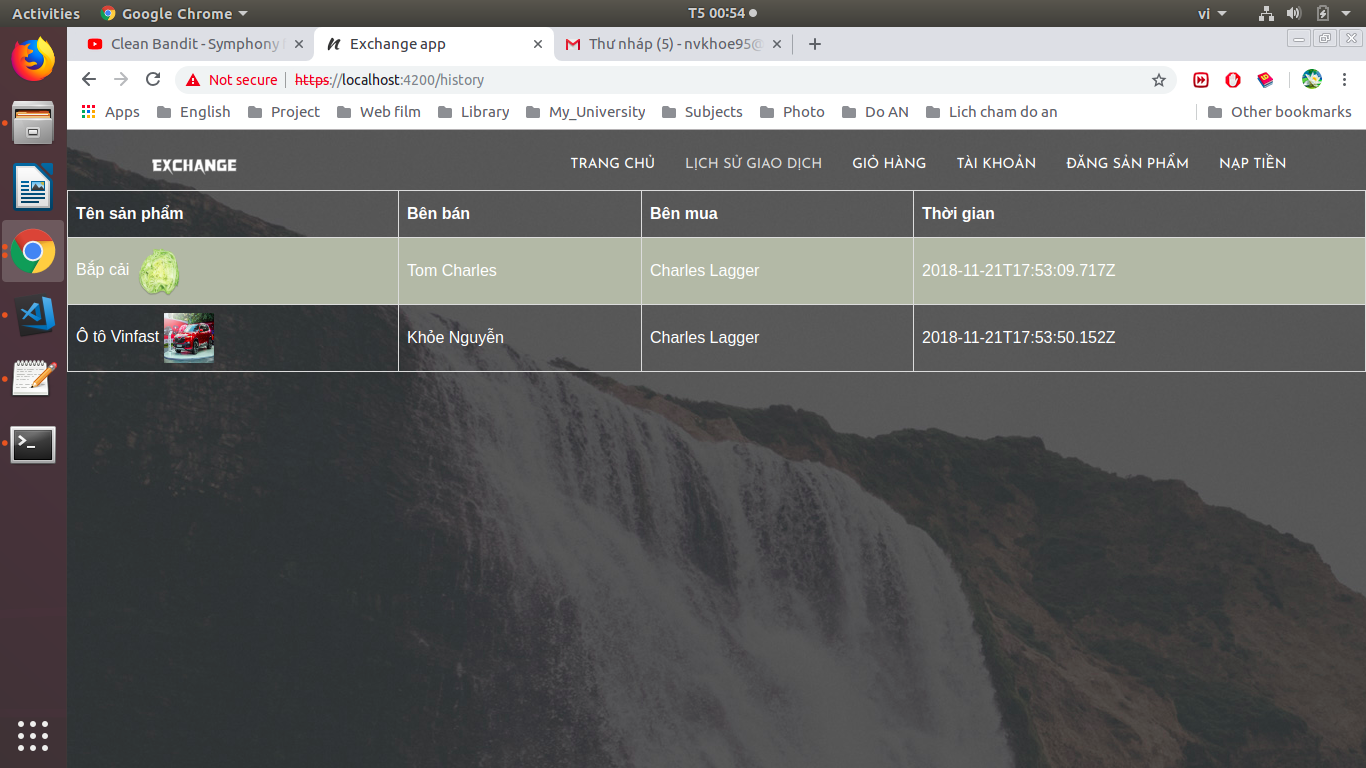












### Client admin

# Kết luận (10%)

## Khái quát toàn bộ nội dung đồ án

## Nhận xét, đánh giá

### Những vấn đề còn tồn tại

### Định hướng phát triển

Tài liệu tham khảo

[1] Elli Androulaki, Artem Barger, Vita Bortnikov, Christian Cachin, Konstantinos Christidis, Angelo De Caro, David Enyeart, Christopher Ferris, Gennady Laventman, Yacov Manevich, et al. Hyperledger fabric: a distributed operating system for permissioned blockchains. In *Proceedings of the Thirteenth EuroSys Conference*, page 1. ACM, 2018.

[2] Elli Androulaki, Artem Barger, Vita Bortnikov, Christian Cachin, Konstantinos Christidis, Angelo De Caro, David Enyeart, Christopher Ferris, Gennady Laventman, Yacov Manevich, et al. Hyperledger fabric: a distributed operating system for permissioned blockchains. In *Proceedings of the Thirteenth EuroSys Conference*, page 14. ACM, 2018.

[3] Nguyen Hoai Anh and Ao Thu Hoai. Thuong mai dien tu. page 1, 2011.

[4] Nguyen Hoai Anh and Ao Thu Hoai. Thuong mai dien tu. page 45, 2011.

[5] Christian Cachin. Architecture of the hyperledger blockchain fabric. In *Workshop on Distributed Cryptocurrencies and Consensus Ledgers*, volume 310, 2016.

[7] Vinod Kumar and Er Gagandeep Raheja. Business to business (b2b) and business to consumer (b2c) management. *International Journal of Computers & Technology*, 3(3b):447–451, 2012.

[8] Sara Rouhani, Darryl G Humphery, Luke Butterworth, Ralph Deters, and Adam D Simmons. Medichainâ€‹ tmâ€‹: A secure decentralized medical data asset management system. page 3.

[9] Nguyen Thi Dieu Chi. Hoat dong thuong mai dien tu tai viet nam trong ky nguyen ky thuat so. *Tap chi Khoa hoc va Cong nghe Viet Nam*, (6A):22, 2018.

[10] Uchi Ugobame Uchibeke, Sara Hosseinzadeh Kassani, Kevin A Schneider, and Ralph Deters. Blockchain access control ecosystem for big data security. *arXiv preprint arXiv:1810.04607*, page 1, 2018.