# BỘ THÔNG TIN VÀ TRUYỀN THÔNG

**HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG**

---------❧•❧---------



**BÁO CÁO THỰC TẬP**

***Đề tài*:**

**Ứng dụng theo dõi Tỷ giá, Tiền điện tử**

**App Giá**

### Giảng viên hướng dẫn : ThS. NGUYỄN ANH HÀO

### Sinh viên thực hiện : NGUYỄN TẤN BẢO CƯỜNG

### Mã số sinh viên: N14DCCN278

**Lớp : D14CQCP01-N**

**Khóa : 2014 - 2019**

**Hệ : ĐẠI HỌC CHÍNH QUY**

**TP.HCM, tháng 7 /2018**

**LỜI CẢM ƠN**

Trên thực tế không có sự thành công nào mà không gắn liền với những sự hỗ trợ, giúp đỡ dù ít hay nhiều, dù trực tiếp hay gián tiếp của người khác. Trong suốt thời gian từ khi bắt đầu học tập tại trường đến nay, em đã nhận được rất nhiều sự quan tâm, giúp đỡ của quý thầy cô, gia đình và bạn bè. Với lòng biết ơn sâu sắc nhất, em xin gửi đến quý thầy cô ở khoa Công Nghệ Thông Tin 2 – Học viện Công Nghệ Bưu Chính Viễn Thông cơ sở tại TP.HCM đã tận tâm chỉ bảo, truyền đạt vốn kiến thức quý báu cho chúng em trong suốt thời gian học tập tại trường. Và đặc biệt trong học kỳ này, nếu không có những lời hướng dẫn, dạy bảo của các thầy cô thì em nghĩ bài báo cáo này của em rất khó có thể hoàn thiện được. Một lần nữa, em xin chân thành cảm ơn thầy cô. Bài báo cáo thực tập thực hiện trong khoảng thời gian 10 tuần. Bước đầu đi vào thực tế của em còn hạn chế và còn nhiều bỡ ngỡ. Do vậy, không tránh khỏi những thiếu sót là điều chắc chắn, em rất mong nhận được những ý kiến đóng góp quý báu của quý Thầy Cô và các bạn học cùng lớp để kiến thức của em trong lĩnh vực này được hoàn thiện hơn.

Em xin gởi lời cảm ơn chân thành và sự tri ân sâu sắc đối với các thầy cô ở khoa Công Nghệ Thông Tin 2 – Học viện Công Nghệ Bưu Chính Viễn Thông cơ sở tại TP.HCM đã tạo điều kiện cho em để em có thể hoàn thành tốt bài báo cáo thực tập này. Đặc biệt em xin gửi lời cảm ơn đến thầy ThS Nguyễn Anh Hào với những buổi vừa học kết hợp với thực hành trên lớp cùng những buổi nói chuyện rất hữu ích đã giúp em định hướng và hoàn thành tốt đề tài này.

Trong quá trình làm bài báo cáo, cũng như là trong quá trình làm đề tài khó tránh khỏi sai sót, rất mong các Thầy, Cô bỏ qua. Đồng thời do trình độ lý luận cũng như kinh nghiệm thực tiễn còn hạn chế nên bài báo cáo không thể tránh khỏi những thiếu sót, em rất mong nhận được ý kiến đóng góp Thầy, Cô để em học thêm được nhiều kinh nghiệm làm hành trang vững chắc để em tự tin theo đuổi sự nghiệp của mình.

Em xin chân thành cảm ơn!

Thành phố Hồ Chí Minh, tháng 7 năm 2018

Nguyễn Tấn Bảo Cường

**MỤC LỤC**

**DANH MỤC CÁC KÝ HIỆU VÀ CHỮ VIẾT TẮT**

|  |  |
| --- | --- |
| **Ký hiệu** | **Giải thích** |
| MVVM | Mô hình Model-View-ViewModel |
| CSDL | Cơ sở dữ liệu |
| Admin | Administrator |
| UI | User Interface |

**LỜI MỞ ĐẦU**

### Lý do chọn đề tài

Đi theo xu hướng hiện tại, ngoài vấn đề nhà đất, cổ phiếu, tỷ giá người ta còn bắt đầu quan tâm đến một khía cạnh đặc biệt mới nổi trong vòng 10 năm trở lại đây đó chính là Tiền điện tử. Hiện nay một cơ số người đã chuyển từ đầu tư cổ phiếu sang giao dịch thông qua Tiền điện tử, một loại tiền tệ giao dịch nặc danh, dễ sở hữu và chiếm được niềm tin của rất nhiều nhà đầu tư vì cơ chế bảo mật của nó.

*“Bitcoin và các loại tiền tệ ảo khác đang tạo nên một "cơn địa chấn" trong năm nay, với tỷ suất lợi nhuận đem lại vượt qua cổ phiếu, trái phiếu, ngoại tệ, vàng và hầu hết các kênh đầu tư truyền thống khác.*

*Giá Bitcoin đã tăng gấp vài lần kể từ đầu năm, mới chạm mốc 5.000 USD một đồng Bitcoin vào hôm qua, 3/9. Theo công ty theo dõi tín dụng CoinMarketCap, mức trần của thị trường Tiền điện tử hiện đã hơn 100 tỷ USD, tăng từ con số khoảng 20 tỷ USD vào đầu năm nay.”*

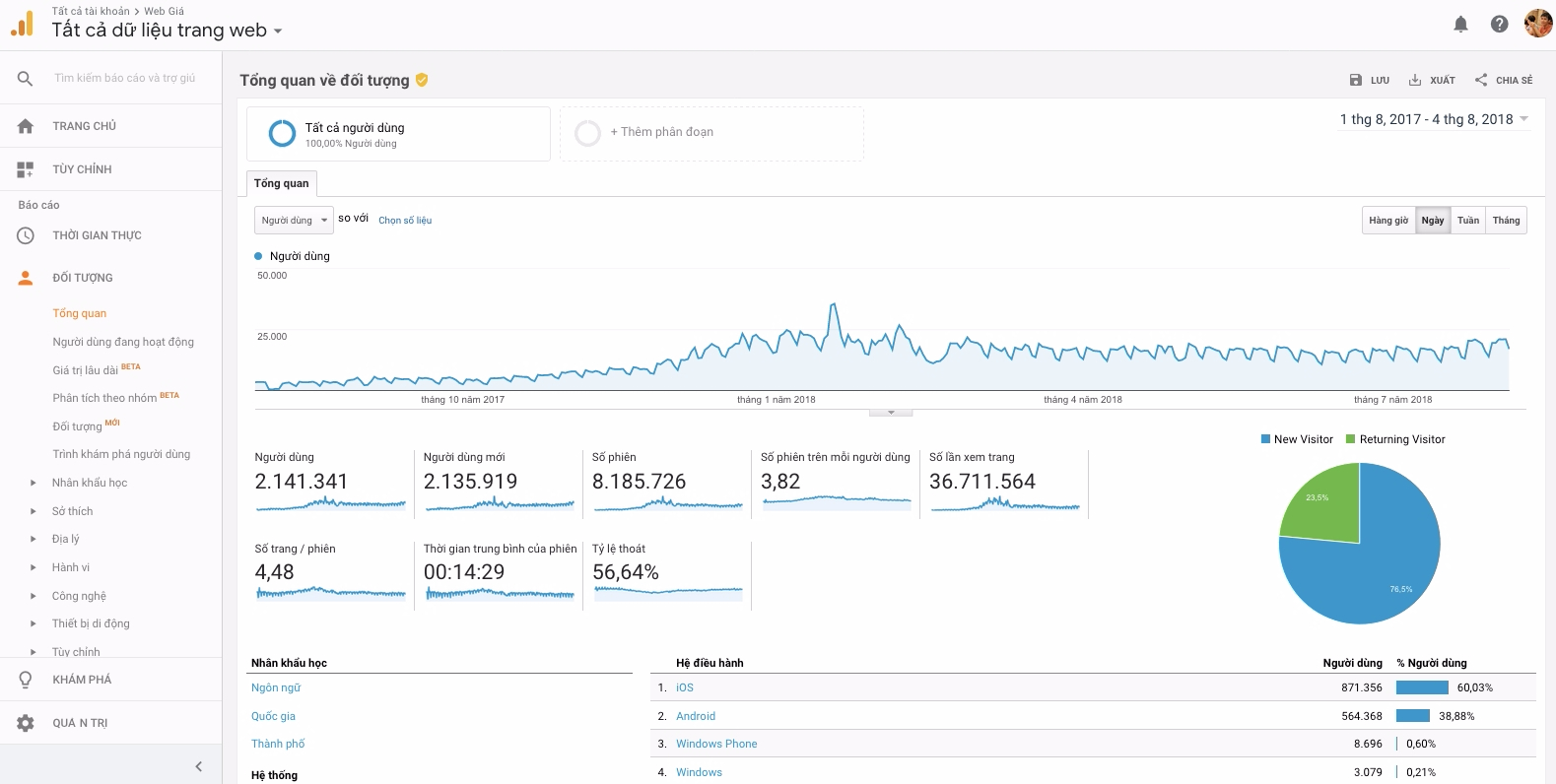
Nguồn : [1]

Người dùng sử dụng tiền điện tử được ở mọi nơi trên thế giới, thậm chí đổi thành tiền mặt khi cần thiết. Dưới hình thức đồng Bitcoin, đồng Ethereum, đồng Steem…, đây còn là nguồn đầu tư tài chính hiệu quả. Hiện tại nhiều nước trên thế giới đã cho phép giao dịch bằng tiền điện tử và thu phí.

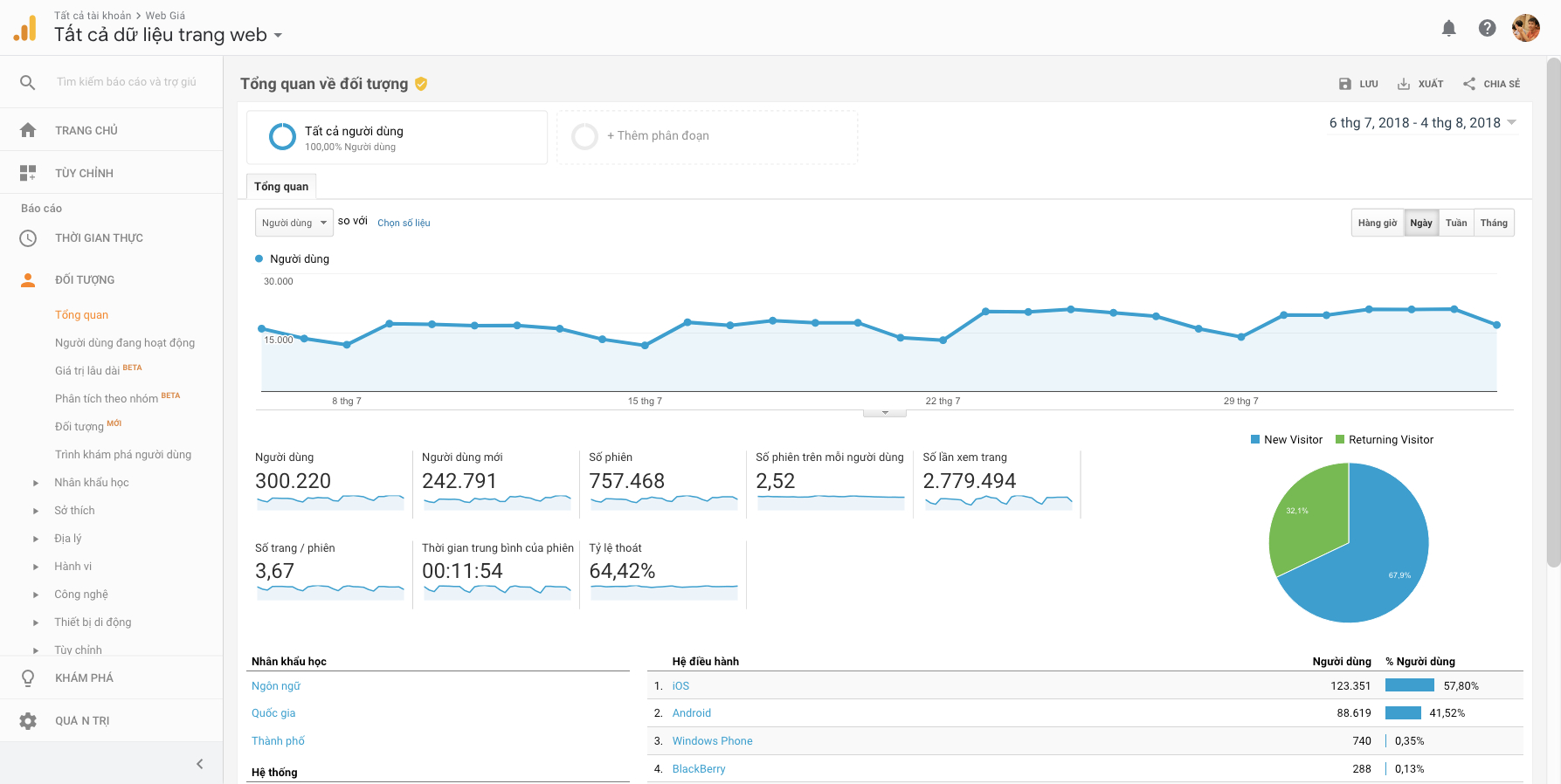
Tại Việt Nam, tiền điện tử đã xuất hiện từ lâu dưới nhiều hình thức. Phổ biến nhất là loại tiền điện tử offline (thẻ trả trước, thẻ thông minh), tiền điện tử online (ví điện tử), đồng Bitcoin… Trong đó, xuất hiện từ năm 2009, đồng Bitcoin đang thu hút sự quan tâm của giới đầu tư. Doanh nghiệp quản lý sàn giao dịch Bitcoin đầu tiên tại Việt Nam cho biết tăng trưởng đến 2 con số mỗi năm về lượng người dùng, giá trị giao dịch trung bình mỗi ngày hiện lên đến hàng nghìn USD. Vừa qua, Thủ tướng Chính phủ cũng đã phê duyệt Đề án Hoàn thiện khung pháp lý để quản lý, xử lý đối với các loại tài sản tiền điện tử. Với Đề án này, các hình thức tiền điện tử như Bitcoin sẽ được thừa nhận cũng như có hình thức quản lý phù hợp. Đồng nghĩa với việc phổ biến của đồng tiền điện tử tại Việt Nam.

Bên cạnh đó, trước khi có tiền điện tử thì tỷ giá, giá xăng dầu, giá vàng cũng thuộc lĩnh vực tài chính và được quan tâm nhiều trước khi có tiền điện tử. Cụ thể hơn, các trang báo nổi tiếng tại Việt Nam như zing.vn, vnexpress.net, vietnamnet.vn, voh.com.vn, baolaodong.vn và nhiều trang đưa tin khác đặc biệt là giavang.net liên tục đưa tin tức về giá vàng và tỷ giá mỗi giờ. Trang web giavang.net chuyên đưa tin về giá vàng, giá xăng dầu, và tỷ giá đứng thứ hạng 2400 tại Việt Nam(bởi alexa.com) tương ứng với khoảng 2 triệu lượt truy cập mỗi tháng tương đương với khoảng 60 nghìn lượt truy cập mỗi ngày.

Một báo cáo cho biết rằng trang webgia.com có thứ hạng 1900 tại Việt Nam (bởi alexa.com) chuyên thông tin về lĩnh vực tài chính, tỷ giá vàng, giá xăng, tiền điện tử… mỗi tháng có khoảng 2,8 triệu lượt truy cập trong đó ½ lượng truy cập là đến từ tiền điện tử. Điều này chứng tỏ có một số lượng người Việt đang theo dõi rất thường xuyên về giá vàng, giá xăng, tiền điện tử.



Biểu đồ lưu lượng truy cập của trang Webgia.com đến từ một nguồn đáng tin cậy cho hay, từ giai đoạn trước tháng 12 năm 2017, lượt truy cập mỗi ngày chỉ từ 2.000 đến hơn 3.000. Nhưng từ tháng 1/2018, chủ sở hữu trang web cho biết sau khi thêm bảng giá tiền điện tử vào giao diện web, lượt truy cập đã tăng đột biến lên đến 160.000 lượt mỗi ngày và sau đó bắt đầu bão hòa ở mức 80.000 cho đến nay. Báo cáo cũng cho thấy, lượng người dùng truy cập bằng trình duyệt điện thoại di động rất cao, đặc biệt là trên nền tảng IOS, từ thời điểm 1 năm trước đến nay là 870.000 lượt truy cập.



Và số liệu tháng gần nhất tức tháng 7/2018 cho thấy có khoảng 120.000 lượt truy cập bằng trình duyệt điện thoại trên nền tảng IOS tức là có khoảng 4.000 lượt truy cập mỗi ngày. Số liệu của webgia.com chỉ là 1 phần của nhiều trang website đưa tin, vậy những dẫn chứng này cho thấy số lượng người có nhu cầu sử dụng điện thoại để theo dõi tiền điện tử, giá vàng, giá xăng, tỷ giá ngoại tệ rất cao. Vậy một ứng dụng phục vụ cho những nhóm người này là rất có cơ sở để phát triển.

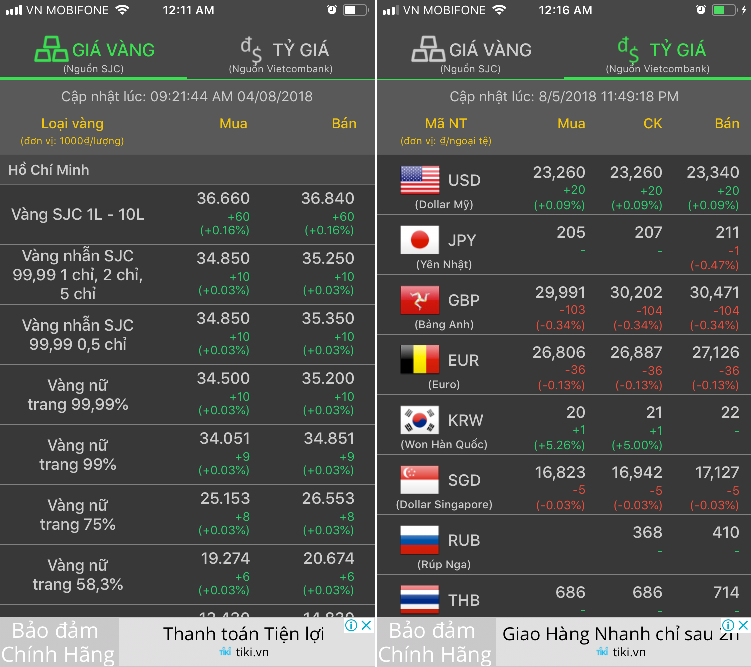
Trên thế giới có rất nhiều ứng dụng trên nền tảng IOS theo dõi tiền điện tử rất nổi tiếng như Coinmarketcap, Coin Stats. Tại Việt Nam chưa có ứng dụng nào thực sự tốt về lĩnh vực này.

Coinmarketcap có giao diện đơn giản, thân thiện với người dùng nhưng lại thiếu thông tin chi tiết của từng đồng tiền và ứng dụng chỉ có một giao diện duy nhất, muốn có thông tin chi tiết thì phải lên trang chủ của coinmarketcap gây ra sự bất tiện.

Coin Stats có thiết kế không đẹp mắt như coinmarketcap nhưng thực sự là một ứng dụng lớn mang tầm cỡ quốc tế. Ngoài việc theo dõi tiền điện tử thì còn có thể xem chi tiết của từng đồng tiền, đặt thông báo nếu giá trị của đồng tiền đạt mức, đọc tin tức quốc tế liên quan đến tiền điện tử và một số công cụ khác. Ứng dụng còn tích hợp cả ví thanh toán giao dịch bằng tiền điện tử.

Các ứng dụng trên đều hoàn thiện nhưng không hỗ trợ Tiếng Việt và việc không phân trang khi tải dữ liệu có lẽ là một điểm trừ vì sẽ là trở ngại khi mạng yếu. Đa số người dùng đều không học cách sử dụng ứng dụng, thị hiếu người Việt lại càng không. Nhất là đối với ứng dụng sử dụng hoàn toàn tiếng anh như vậy. Vì vậy cần có một ứng dụng Tiếng Việt sử dụng thiết kế của Coinmarketcap kết hợp ứng dụng công cụ của Coin Stats và sử dụng hoàn toàn Tiếng Việt đồng thời nghiên cứu thị hiếu và cải tiến trong từng phiên bản sẽ đáp ứng được nhu cầu của người dùng có liên quan đến lĩnh vực tài chính tại Việt Nam.

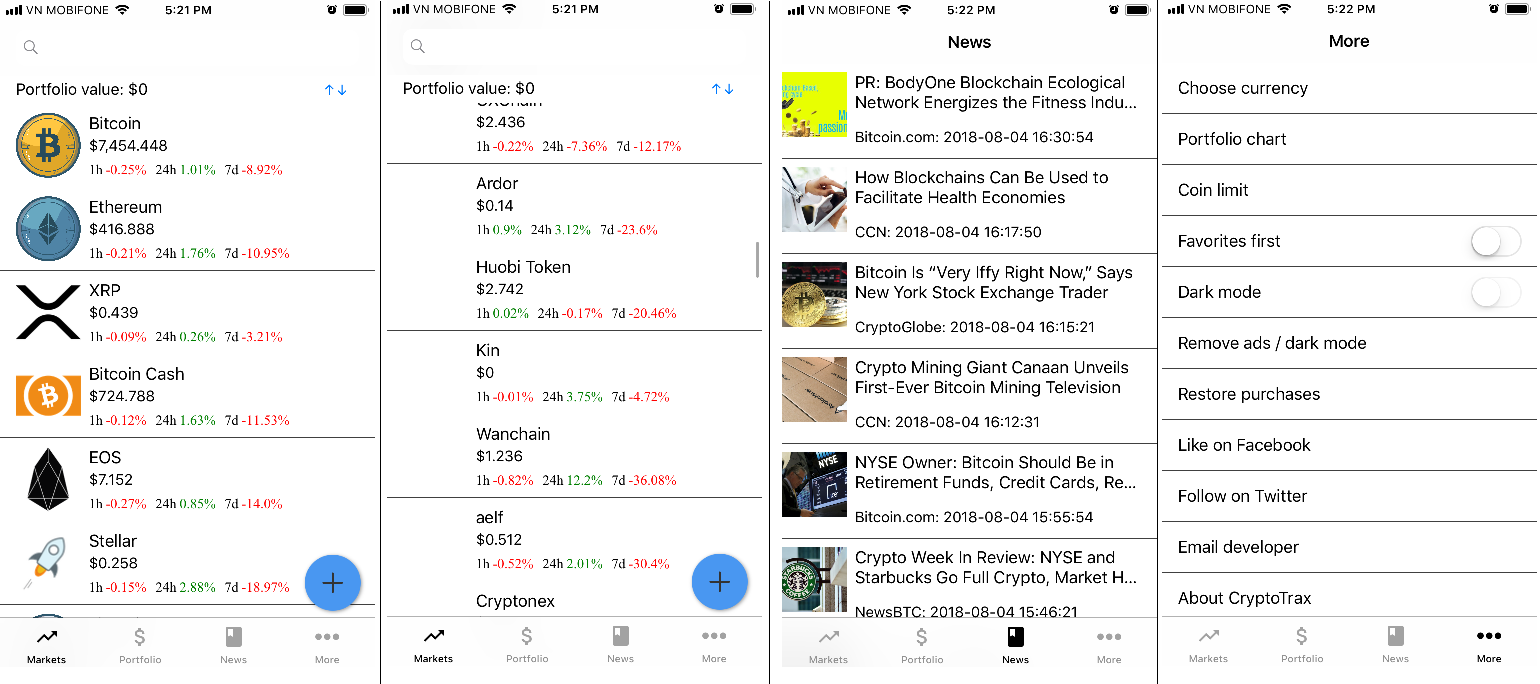
Ứng dụng Giá Vàng chuyên cung cấp thông tin giá vàng, tỷ giá ngoại tệ có giao diện thân thiện dễ sử dụng, nếu kết hợp với giá tiền điện tử, giá xăng, kèm thêm một số tính năng hỗ trợ tốt cho người dùng nữa thì sẽ được đông đảo người dùng tại Việt Nam quan tâm.



**CHƯƠNG 1: TỔNG QUAN**

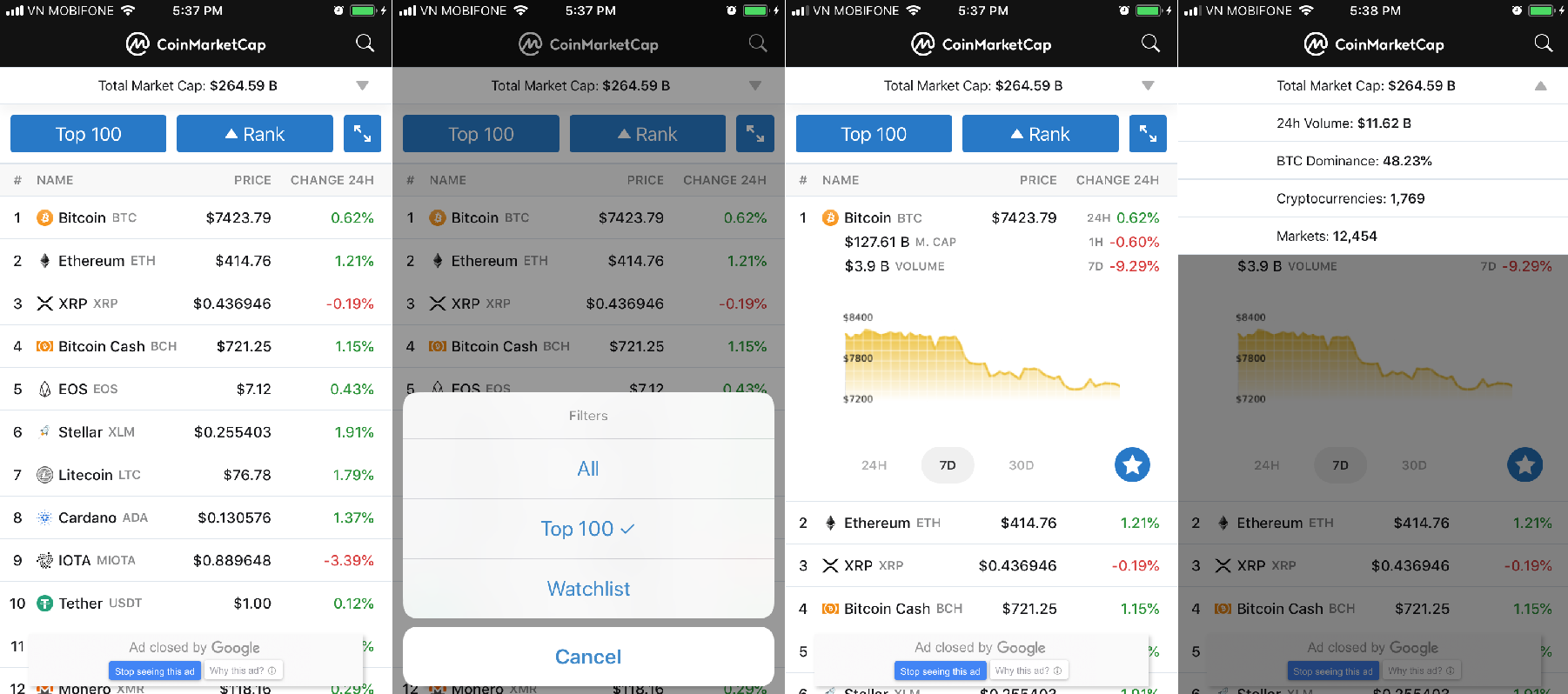
* 1. **Tình hình nghiên cứu trong nước**

Hiện chưa có đơn vị nào phát triển ứng dụng cho phép theo dõi Tiền điện tử trên hệ điều hành IOS sử dụng ngôn ngữ Tiếng Việt.

* 1. **Tình hình nghiên cứu ngoài nước**
     1. **CryptoTrax** CryptoTrax vận hành mọi thứ đơn giản. Với thiết kế gọn gàng, người dùng có thể dễ dàng xem thông tin các loại Tiền điện tử như Bitcoin, Ethereum, NEM, NEO…  
        Chức năng: 
* Theo dõi giá tiền điện tử
* Xem chi tiết tiền điện tử
* Xem tin tức thế giới liên quan đến tiền điện tử
* Thể hiện biểu đồ tròn cho nhóm tiền điện tử để phân tích

Nhận xét:

* Ứng dụng có giao diện đơn giản nhưng hiệu suất kém, dùng thử cho thấy khi lướt màn hình với tốc độ bình thường thì hình ảnh chưa kịp load, có cảm giác giật lag. Giao diện không đẹp.
* Phần mềm không hỗ trợ tiếng việt
  + 1. **CoinMarketCap** Cha đẻ của CoinMarketCap là trang web nổi tiếng <https://coinmarketcap.com>, một trang web hàng đầu về thông tin Tiền điện tử. Đây là trang không thể thiếu đối với dân đầu tư trong năm vừa qua cùng với sự tăng giá của Bitcoin và thị trường tiền điện tử nói chung. Vô cùng hữu ích để theo dõi những biến động tăng giảm của thị trường. Giao diện vô cùng đơn giản với 1 màn hình duy nhất

Chức năng:

* Theo dõi giá trị tiền điện tử.

Nhận xét:

* Ứng dụng có giao diện đẹp nhưng chức năng đơn giản. Không hỗ trợ nhiều cho người dùng.
* Tuy vậy lượng người dùng sử dụng ứng dụng coinmarketcap rất đông vì coinmarketcap có trang chủ có lượng truy cập đứng thứ 300 trên thế giới được đánh giá bởi alexa.com.
* Không hỗ trợ Tiếng Việt

**1.3 Phương pháp nghiên cứu**

**1.3.1 Khảo sát thực tế và thu thập thông tin**

Tham khảo các ứng dụng nổi tiếng thế giới về các chức năng và cách thiết kế giao diện. Các ứng dụng dùng để tham khảo là Coinmarketcap, Coin Stats, Giá Vàng, CryptoTrax, Coinmarketsum. Tham khảo bình luận điểm mạnh của từng ứng dụng. Nghiên cứu giao diện ứng dụng hàng đầu Việt Nam để tìm hiểu thị hiếu người dùng Việt.

**1.3.2 Phân tích và thiết kế**

* Từ các thông tin có được, bắt đầu thiết kế giao diện di động.
* Thiết kế cơ sở dữ liệu sử dụng mongoDB với mô hình NoSQL vì dữ liệu sử dụng thuộc dạng mô hình các đối tượng.
* Kết quả của quá trình này là vẽ mô hình usercase, mô hình API, các màn hình giao diện chức năng.

**1.3.3 Cài đặt ứng dụng**

* Bắt đầu sử dụng ngôn ngữ Swift để viết ứng dụng di động trên nền IOS bằng ứng dụng Xcode 9.0 trên nền hệ điều hành MacOS sử dụng giả lập VMWare Workstation 14.0 trên máy tính Lenovo T420 dùng hệ điều hành Window 7.

**1.3.4 Kiểm định ứng dụng**

Sau khi xây dựng hoàn thành ứng dụng, tiến thành thực hiện kiểm nghiệm ứng dụng, các bước được thực hiện như sau:

* Tạo kịch bản kiểm thử
* Nhập liệu và kiểm tra các chức năng như kịch bản
* Chỉnh sửa các lỗi phát sinh trong quá trình kiểm thử

**CHƯƠNG 2: CƠ SỞ LÝ THUYẾT**

* 1. **Tiền điện tử**
     1. **Khái niệm**

Tiền mã hóa/Tiền ảo là loại tiền kỹ thuật số được thiết lập trên nền tảng công nghệ chuỗi khối (blockchain), vận hành trên một cuốn sổ cái phân tán (distributed ledger) - một bảng ghi tất cn ả các giao dịch được cập nhật và lưu trữ bởi các thành viên tham gia hệ thống, cho phép người dùng thực hiện các giao dịch thanh toán an toàn và lưu trữ loại tiền này một cách ẩn danh và không thông qua một bên trung gian thứ ba.

Bitcoin, được tạo ra trong năm 2009 bởi lập trình viên Satoshi người vẫn chưa rõ danh tính cho đến tận nay, là tiền mã hoá đầu tiên. Kể từ đó, nhiều loại tiền mã hóa khác đã được tạo ra. Chúng thường được gọi là altcoin, viết tắt của đồng tiền thay thế. Bitcoin và các dẫn xuất của nó sử dụng kiểm soát phi tập trung đối lập với tiền điện tử tập trung và các hệ thống ngân hàng trung ương. Việc kiểm soát phi tập trung này có liên quan đến việc sử dụng cơ sở dữ liệu giao dịch blockchain của bitcoin trong vai trò như một sổ cái lưu trữ dạng phân tán.

* + 1. **Đặc điểm**

**Không thể đảo ngược:** Sau khi xác nhận, một giao dịch không thể được đảo ngược. Không phải bạn, không phải ngân hàng của bạn, không phải là tổng thống của Hoa Kỳ, không phải Satoshi, không phải thợ mỏ của bạn. Không ai có thể giúp bạn, nếu bạn gửi tiền nhằm mục đích lừa đảo hoặc nếu một hacker lấy trộm chúng từ máy tính của bạn, không có mạng lưới an toàn.

**Ẩn danh:** Cả giao dịch và tài khoản đều không được kết nối với danh tính trong thế giới thực. Bạn nhận được Bitcoin trên các địa chỉ được gọi là, những chuỗi ngẫu nhiên dường như có khoảng 30 ký tự. Mặc dù thường có thể phân tích luồng giao dịch, nhưng không nhất thiết phải có thể kết nối danh tính thực của người dùng với các địa chỉ đó.

**Nhanh chóng & toàn cầu:** Giao dịch được chuyển gần như ngay lập tức trong mạng và được xác nhận trong một vài phút. Vì chúng xảy ra trong một mạng lưới máy tính toàn cầu, nên dù bạn ở bất kỳ nơi đâu trên thế giới chỉ sau vài phút sẽ nhận được tiền.

**Bảo mật:** Các quỹ Tiền điện tử bị khóa trong một hệ thống mật mã khóa công khai, chỉ chủ sở hữu mới có thể mở và chuyển nhận. Bảo mật mạnh mẽ và sự kỳ diệu của những con số làm cho nó những tên hacker không thể làm điều xấu. Địa chỉ Bitcoin an toàn hơn Fort Knox. Không ép buộc: Không ai có thể yêu cầu hay ép buộc bạn sử dụng cryptocurrency, mọi thứ đều do bạn tự nguyện làm. Bạn có thể chuyển nhận Bitcoin hay các loại Altcoin khác sau khi cài đặt ứng dụng.

**2.2 Công nghệ BlockChain**

**2.2.1 Tổng quan**

Blockchain là công nghệ lưu trữ và truyền tải thông tin bằng các khối được liên kết với nhau và mở rộng theo thời gian. Mỗi khối chứa đựng các thông tin về thời gian khởi tạo và được liên kết với các khối trước đó.

Blockchain được thiết kế để chống lại sự thay đổi dữ liệu. Thông tin trong blockchain không thể bị thay đổi và chỉ được bổ sung thêm khi có sự đồng thuận của tất cả các nút trong hệ thống. Ngay cả khi nếu một phần của hệ thống blockchain đổ, những máy tính và nút khác sẽ tiếp tục hoạt động để bảo vệ thông tin.

Đặc biệt blockchain có khả năng truyền tải dữ liệu mà không đòi hỏi trung gian để xác nhận thông tin. Hệ thống blockchain bao gồm nhiều nút độc lập có khả năng xác thực thông tin.

Hiện nay, các doanh nghiệp lớn trên thế giới như Microsoft và sắp tới Oracle sẽ giới thiệu, tham gia chạy đua đưa các công nghệ chuỗi khối này. Giải pháp công nghệ đột phá này có thể áp dụng tối ưu cho các ngành nghề như tài chính, giao thông và nhiều lĩnh vực khác.

**2.2.2 Ưu điểm**

Blockchain được hiểu như cuốn sổ cái kế toán của một công ty, nơi mà tiền mặt được giám sát chặt chẽ. Ưu điểm lớn nhất của blockchain là tính bảo mật và minh bạch cao bởi nó ghi chép lại toàn bộ thông tin về hàng hóa, từ bước sản xuất đến tận giai đoạn tiêu dùng. Hiểu một cách đơn giản, đây là một kỹ thuật mã hóa cho phép xác minh thông tin xuyên suốt vòng đời của sản phẩm, tự động theo dõi dòng đi của sản phẩm giữa các bên. Quyền truy cập tới các thông tin này được giao cho tất cả mọi người trong hệ thống, chứ không phụ thuộc vào một bên trung gian như ngân hàng. Vì vậy, dữ liệu một khi đã được xác thực và lưu lại sẽ khó bị thay đổi hay đánh cắp.

**2.2.3 Ứng dụng**

Trong lĩnh vực vận chuyển hàng hóa, với công nghệ này, một người nhập có thể chia sẻ thông tin cho nhiều đơn vị trong cùng một mạng lưới được xây dựng. Ví dụ khi hàng hóa được chuyển từ hải quan Mỹ đến hải quan Việt Nam, khi hàng hóa chuyển đến đâu thì tất cả những thành viên tham gia mạng blockchain đều có thể theo dõi tình trạng hàng hóa và biết cụ thể thời gian đến.

Trong lĩnh vực bán lẻ hay nông nghiệp, blockchain sẽ phục vụ hiệu quả cho việc truy xuất nguồn gốc để biết rõ các sản phẩm, hàng hóa được nhập khẩu, sản xuất từ đâu.

Ví dụ, một chiếc xe ôtô từ nhà máy tới hệ thống đại lý khi bán tới tay khách hàng có thể sẽ phát sinh thêm nhiều giao dịch, trao đổi mua bán tiếp sau đó. Nếu ứng dụng công nghệ chuỗi khối hoàn toàn có thể truy xuất được nguồn gốc xe, lịch sử các giao dịch mua bán cụ thể, cũng như thông tin bảo hành, sửa chữa...

**2.3 Ngôn ngữ Swift**

**2.3.1 Tổng quan**

Swift là ngôn ngữ mới nhất hiện tại của Apple được công bố vào năm 2014. Hiện đã nâng cấp lên thành Swift 4.0 tính tới 2018. Được phát triển lên từ Objective-C và có tính hướng đối tượng. Để bắt đầu, Chúng ta cần phải sử dụng hệ điều hành Mac OS (trên máy Mac hoặc Hackintosh) và dĩ nhiên là phải cài ứng dụng Xcode. Có nhiều lí do mà Swift hiện vượt trội hơn Objective-C trên nhiều phương diện.

**2.3.2 Đặc điểm cơ bản**

**+** Swift ít lỗi hơn

Apple khẳng định Swift là ngôn ngữ lập trình có thể tránh được nhiều lỗi. Điều này được hỗ trợ bởi cú pháp và ngôn ngữ của Swift có thể loại bỏ được nhiều loại lỗi phổ biến của ngôn ngữ Objective-C. Sự vận hành ổn định này giúp tránh được những gián đoạn không mong muốn. Tuy không hoàn toàn tránh được những mã xấu nhưng lập trình viên hoàn toàn có thể được bảo vệ để tránh mắc lỗi. Kiểm soát được chất lượng được xem làm một trong những khả năng vượt trội của Swift khiến nó trở thành một ngôn ngữ lập trình an toàn.

+ Swift nhanh hơn

Mặc dù là một ngôn ngữ cấp cao tập trung vào sự phát triển nhanh chóng, hiệu suất và tốc độ của Swift không có gì để chê cả. Theo Apple, Swift nhanh gấp 2,6 lần so với Objective-C và nhanh hơn 8,4 lần so với Python 2.7. Các nhà phát triển cuối cùng muốn làm cho ngôn ngữ này nhanh hơn C + +, một trong những ngôn ngữ thực hành nhanh nhất hiện nay.

Và nó không chỉ nhanh, mà còn mạnh mẽ và đáp ứng đầy đủ với các tính năng ngôn ngữ hiện đại cho phép bạn viết mã nâng cao. Những điểm nổi bật bao gồm: generics, closures, tuples, multiple returns, iterators, các mẫu lập trình hàm có sẵn và nhiều hơn nữa.

+ Swift là mã nguồn mở

Được tạo ra để phục vụ trên nền tảng Apple, Swift được xem là một ngôn ngữ mở, dễ thích ứng với các nền tảng khi đang dần dần thích ứng để được xây dựng trên Linux. Nếu so sánh với Objective-C thì Swift có thể dễ dàng di chuyển trên nhiều nền tảng hơn

+ Swift có khả năng tương tác cao hơn

Swift là một ngôn ngữ lập trình có khả năng tương tác cao. Swift Playground hay còn gọi là môi trường tương tác riêng cho Swift đã mở ra nhiều cơ hội mới cho các developer. Công cụ này cho phép kiểm tra mã code ngay lập tức mà không cần biên tập những phần lớn hay phải tạo ra một ứng dụng hoàn chỉnh. Môi trường này trực quan hóa dữ liệu và các nhà lập trình có thể nhanh chóng kiểm tra và chỉnh sửa mọi thứ. Điều này đặc biệt được áp dụng cho những cách nhìn nhận quen thuộc và trải nghiệm với mã code.

+ Swift có cú pháp thân thiện, hiện đại

Các dòng code của Swift ngắn và dễ đọc, thứ mà rất rối rắm khi lập trình bằng Objective-C. Điều đó sẽ giúp đẩy nhanh quá trình phát triển ứng dụng và tăng hiệu quả cho quá trình bảo trì trong tương lai.

+ Swift là ngôn ngữ mới

Mặc dù là ngôn ngữ mới nhưng điều này có nghĩa là Apple sẽ chăm sóc kĩ càng hơn về ngôn ngữ này, sẽ có những cải tiến liên tục để phù hợp với hiện đại.

+ Swift có tất cả tính năng của một ngôn ngữ hiện đại

Swift có tất cả các tính năng của một ngôn ngữ lập trình hiện đại và giữ vững thông qua quá trình mã hóa. Các tính năng quan trọng bao gồm:

* Không có các biến không xác định hoặc không bắt buộc.
* Không có lỗi array-out-of-bounds.
* Không có lỗi tràn số nguyên (integer overflow).
* Xử lý rõ ràng các giá trị nil (null).
* Quản lý bộ nhớ tự động.

Vì vậy, ta có thể dành nhiều thời gian hơn để viết business logic thực tế và tốn ít thời gian lo lắng về tất cả các trường hợp có thể khiến mã của bạn thành mớ bòn bon.

+ Swift an toàn và bảo mật hơn Object-C

Nhiều tính năng trong cùng một ngôn ngữ khiến Swift nhanh hơn so với Objective-C; cũng làm cho nó an toàn và bảo mật hơn Objective-C.

Ví dụ, quản lý bộ nhớ tốt hơn nghĩa là ít cơ hội phân bổ sai dữ liệu, truy cập các bộ phận sai của bộ nhớ, thay đổi dữ liệu không cần thay đổi. Một ví dụ khác, xử lý lỗi tốt hơn nghĩa là giảm thiểu toàn bộ sự cố và khi bị treo, kịch bản thảm khốc hiếm xảy ra hơn.

**2.4 Mô hình MVVC (Model-View-ViewModel):**

**2.4.1 Tổng quan**

Từ khi iOS được tạo dựa trên mô hình kiến trúc MVC, nó có vẻ luôn là sự lựa chọn đầu tiên, và ngay cái nhìn đầu tiên nó đã không đủ sự đơn giản.

Mô hình phân chia phần mềm thành 3 phần khác nhau, trong đó Model đại diện cho data, View đại diện cho giao diện người dùng (user interface) và Controller hoạt động như chất keo kết dính giữa 2 phần còn lại. Điều này có vẻ rõ ràng, nhưng các ứng dụng ngày càng phức tạp, cách tiếp cận này cho thấy những thiếu sót của nó.

Như hướng dẫn của Apple, mọi thứ trong mô hình MVC nên được phân loại là Model, View hoặc Controller, nhưng nó sẽ tạo cho bạn câu hỏi: mọi thứ có thể thực sự được sắp xếp vào 1 trong 3, mà không bị quá tải một phần nào?

Tất cả code và logic thật khó để đặt ở nơi nào khác ngoài ViewController, điều này làm cho chúng ngày càng mở rộng với quá nhiều trách nhiệm. Đổi lại nó trở nên không thể tái sử dụng, khó để bảo trì, dễ bị lỗi và khó khăn hơi để kiểm thử.

**2.4.2 Đặc điểm của MVVM**

MVVM giải quyết các vấn đề ở trên bằng cách cung cấp một tập hợp các quy tắc rõ ràng hơn và chia nhỏ phần logic của ứng dụng, quản lý các khối nhỏ dễ dàng hơn.

Ở đây, Model chứa tất cả các *business logic*, giống như lấy data từ database hoặc API, phân tích data và xử lý một số logic tùy chỉnh. Đó là trách nhiệm của *business rule*, *data access*, *model class*…

View đại diện cho toàn bộ giao diện người dùng có thể nhìn thấy bởi người dùng và bao gồm cả UIViewController và UIView.

ViewModel chuẩn bị tất cả thông tin cho giao diện người dùng. Nó giao tiếp với ViewController thông qua một protocol và cung cấp một cách dễ dàng để kiểm thử toàn bộ ViewController.

Ở phương pháp này, ViewController hoạt động như cầu nối giữa ViewModel và View và nó không biết gì về business logic.

Kết quả là, ViewController được quy định chặt chẽ, và làm cho quá trình debug và tái sử dụng dễ dàng hơn nhiều.

**2.4.3 Cấu trúc**

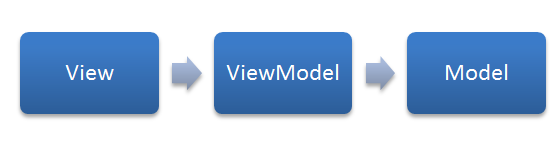
**View**: Đây là nơi quy định thiết kế giao diện cho toàn bộ giao diện mà người dùng nhìn thấy, bao gồm cả UIViewController và UIView. Nếu cần thay đổi điều gì liên quan đến thiết kế hoặc layout sau này thì đây là nơi duy nhất ta cần tìm.

**Model**: Model trong MVVM bao gồm Business Layer và API layer.

Business Layer ở đây kiểm soát Data Model và các service làm một vài logic cụ thể mà có thể extracted và encapsulated. API layer thực hiện các protocol được định nghĩa bởi Business Layer, và cũng thực hiện tất cả các yêu cầu HTTP đến máy chủ.

**ViewModel**: Lớp trung gian giữa View và Model. ViewModel có thể được xem là thành phần thay thế cho Controller trong mô hình MVC. Nó chứa các mã lệnh cần thiết để thực hiện data binding, command.

Một điểm cần lưu ý là trong mô hình MVVM, các tầng bên dưới sẽ không biết được các thông tin gì về tầng bên trên nó. Như hình minh họa dưới đây:



Để làm cho data binding dễ dàng hơn giữa ViewController và ViewModel, chúng ta thường sử dụng ReactiveCocoa/RxCocoa+RxSwift.

**2.5 Giới thiệu về RestfulAPI**

**2.5.1 RestfulAPI ( Application Programming Interface )**

- Là phần mềm trung gian cho phép 2 ứng dụng giao tiếp với nhau

- Khi sử dụng ứng dụng trên thiết bị di động hoặc trên các trang web, các ứng dụng này sẽ kết nối qua Internet và gửi dữ liệu hoặc yêu cầu truy vấn dữ liệu tới máy chủ.

- Máy chủ lấy dữ liệu, diễn giải dữ liệu, thực hiện các hành động cần thiết và gửi dữ liệu trở lại thiết bị, trang web của bạn.

- Ứng dụng nhận kết quả trả về từ máy chủ và hiển thị các thông tin đọc được cho bạn. Đây chính là API.

**2.5.2 Đặc điểm**

Có 4 phương thức được sử dụng để truy cập RESTful API:

+ GET để lấy về 1 đối tượng

+ POST để tạo ra đối tượng mới

+ PUT để sửa đổi hoặc thay thế đối tượng

+ DELETE để xoá đi đối tượng

**2.6 MongoDB**

**2.6.1 Tổng quan**

MongoDB là một chương trình cơ sở dữ liệu mã nguồn mở được thiết kế theo kiểu hướng đối tượng trong đó các bảng được cấu trúc một cách linh hoạt cho phép các dữ liệu lưu trên bảng không cần phải tuân theo một dạng cấu trúc nhất định nào. Chính do cấu trúc linh hoạt này nên MongoDB có thể được dùng để lưu trữ các dữ liệu có cấu trúc phức tạp và đa dạng và không cố định (hay còn gọi là Big Data).

MongoDB sinh ra để tăng tốc độ truy xuất dữ liệu, phù hợp cho các ứng dụng cần tốc độ phản hồi nhanh(như facebook chẳng hạn). Còn các tác nghiệp cần tính toàn vẹn dữ liệu(trong banking) thì Nosql sẽ ko bao giờ là 1 giải pháp cả mà người ta sẽ dùng Mysql.

Lí do ứng dụng sử dụng MongoDB là vì tốc độ phản hồi tạo API nhanh, ứng dụng App Giá lại cần truy xuất theo thời gian thực nên MongoDB rất phù hợp. Ngoài ra còn có nhiều database nhanh hơn như redis, memcached nhưng vì hao tốn tài nguyên ram, nên không phù hợp

**2.6.2 Lợi thế của MongoDB so với các cơ sở dữ liệu dạng quan hệ(RDBMS)**

* Ít Schema hơn: MongoDB là một cơ sở dữ liệu dựa trên Document, trong đó một Collection giữ các Document khác nhau. Số trường, nội dung và kích cỡ của Document này có thể khác với Document khác.
* Cấu trúc của một đối tượng là rõ ràng.
* Không có các Join phức tạp.
* Khả năng truy vấn sâu hơn. MongoDB hỗ trợ các truy vấn động trên các Document bởi sử dụng một ngôn ngữ truy vấn dựa trên Document mà mạnh mẽ như SQL.
* MongoDB dễ dàng để mở rộng.
* Việc chuyển đổi/ánh xạ của các đối tượng ứng dụng đến các đối tượng cơ sở dữ liệu là không cần thiết.
* Sử dụng bộ nhớ nội tại để lưu giữ phần công việc, giúp truy cập dữ liệu nhanh hơn.

**2.6.3 Một số đặc điểm của MongoDB**

* Kho lưu định hướng Document: Dữ liệu được lưu trong các tài liệu kiểu JSON.
* Lập chỉ mục trên bất kỳ thuộc tính nào.
* Các truy vấn đa dạng.
* Cập nhật nhanh hơn.

**2.7 NoSQL**

**2.7.1 Tổng quan**

NoSQL là thuật ngữ được sử dụng một loại cơ sở dữ liệu không hoàn toàn mang các đặc tính của cơ sở dữ liệu quan hệ như không đảm bảo ACID, không có đặc tính transaction v.v.. và loại cơ sở dữ liệu này được thiết kế để tăng hiệu năng xử lý dữ liệu (ghi, đọc dữ liệu) cũng như có thể dễ dàng mở rộng khả năng lưu trữ khi khối lượng dữ liệu tăng lên.

**2.7.2 Lợi thế**

* Không cần xác định trước lược đồ dữ liệu: Việc define lược đồ dữ liệu trong cơ sở dữ liệu quan hệ là bắt buộc, tuy nhiên đối với NoSQL thì lược đồ dữ liệu được define khi insert dữ liệu, và không cần phải định nghĩa trước cấu trúc đó.
* Không tồn tại quan hệ giữa các thực thể trọng NoSQL: đối với cơ sở dữ liệu quan hệ thì việc chuẩn hóa dữ liệu giữa các bảng là rất quan trọng, tuy nhiên NoSQL không đề cao việc này, các thực thể trong NoSQL được thiết kế sao cho việc truy vấn dữ liệu được thực hiện nhanh nhất và thuận tiện nhất, việc dư thừa dữ liệu không phải là điều gì to tát khi sử dụng NoSQL.
* Tính năng mở rộng: khi sử dụng NoSQL, một khối dữ liệu có thể được chia nhỏ để lưu trên nhiều máy khác nhau trong khi người dùng vẫn hình dung dữ liệu nằm trên một khối thống nhất. Tính năng mở rộng làm tăng khả năng truy vấn dữ liệu và đồng thời cũng làm giảm nguy cơ thắt cổ chai khi đồng thời có nhiều request cùng đọc hoặc ghi dữ liệu (việc đọc và ghi được chia sẻ ra nhiều máy).
* Tính phân tán: tính năng này đảm bảo dữ liệu được sao lưu ở nhiều nơi và có thể dễ dàng được backup khi cần.

Tuy nhiên khi sử dụng NoSQL chúng ta cũng phải chấp nhận:

* Không gian lưu trữ dữ liệu: do NoSQL không đề cao việc xử lý dư thừa dữ liệu nên việc áp dụng NoSQL sẽ gia tăng đáng kể dữ liệu và vì thế không gian lưu trữ dữ liệu cũng tăng lên.
* Tính đồng nhất dữ liệu: Cơ sở dữ liệu quan hệ đảm bảo rằng dữ liệu luôn đồng nhất tại mọi thời điểm, đặc tính ACID (Atomicity, Consistency, Isolation and Durability) là bắt buộc trong cơ sở dữ liệu quan hệ. Tuy nhiên NoSQL không tuân theo các quy tắc ấy, NoSQL có một quan điểm đó là, sau một khoảng thời gian, dữ liệu tại mọi điểm trên hệ thống cơ sở dữ liệu là đồng nhất với nhau (Eventually consistence), chứ không phải lúc nào dữ liệu cũng được đồng nhất, cần có một khoảng thời gian để dữ liệu ở các điểm đồng nhất với nhau.

**2.8 Reactive Programming**

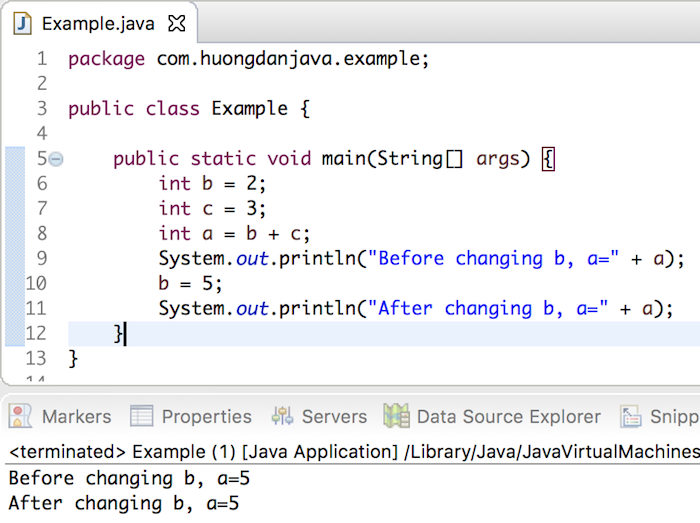
**2.8.1 Tổng quan**

Reactive Programming là một mô hình lập trình mà ở đó bất kỳ sự thay đổi nào trên đối tượng có các đối tượng khác tham chiếu tới thì các đối tượng tham chiếu tới đó đều nhận biết sự thay đổi của đối tượng đó.

Một ví dụ điển hình khi nghiên cứu Reactive Programming là:

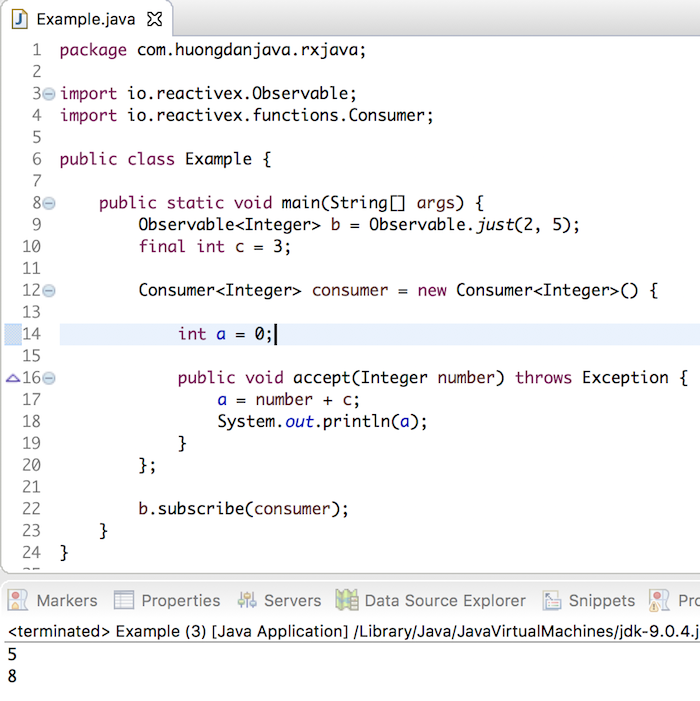
* Nếu ta có biến a là tổng của 2 số b và c, thì bất kỳ sự thay đổi nào của biến b và biến c đều làm thay đổi giá trị của biến a.

Trong những mô hình lập trình khác thì mặc dù a là tổng của b và c nhưng khi chúng ta thay đổi b hoặc c thì giá trị của a vẫn không thay đổi theo. Ví dụ như hình sau:

Sự thay đổi của b không làm thay đổi giá trị của a. Ví dụ bằng ngôn ngữ Java

Khi làm việc với Reactive Programming, khi giá trị của b thay đổi, a sẽ lập tức thay đổi mà không tốn quá nhiều mã code, tất cả sẽ gói gọn trong 1 khối code.

Điều này giúp ứng dụng của bạn hoạt động 1 cách như là real time update UI khi có thay đổi data flow, mang đến trải nghiệm tuyệt vời cho người sử dụng.

Giá trị của a sẽ tự động cập nhật mỗi khi b thay đổi vì đã được đăng kí

**2.8.2 Lợi ích**

* Giúp ứng dụng hoạt động 1 cách như là real time update ui khi có thay đổi data flow, mang đến trải nghiệm tuyệt vời cho User
* Dễ dàng lập trình bất đồng bộ (Async) thay vì phải dùng : callbacks, delegates, notifications, kvo… Thay vì phải dùng nhiều thứ làm cho ứng dụng phức tạp, khó kiểm soát.
* Chuyển đổi các thread khi thực hiện 1 cách dễ dàng.
* Viết code 1 cách ngắn ngọn với các function, operator(toán tử) hỗ trợ.
* Binding UI : đưa dữ liệu vào các UI 1 cách đơn giản, dễ sửa chữa bảo trì, mà không cần phải làm nhiều.

**2.9 Giới thiệu về ReactiveX**

**2.9.1 Tổng quan**

ReactiveX là sự kết hợp tuyệt vời giữa mô hình observable, interator và functional programing. ReactiveX là một thư viện hỗ trợ xử lý các tiến trình bất đồng bộ và xử lý các luồng sự kiện bằng cách sử dụng tuần tự các observable.

**2.9.2 Một số đặc điểm**

**Create**: Với ReactiveX ta có thể dễ dàng khởi tạo các luồng sự kiện hoặc luồng dữ liệu.

**Combine**: Dễ dàng điều chỉnh và truyển đổi luồng dữ liệu hoặc luồng sự kiện với các hàm tương tự nhau.

**Listen**: Subscribe bất cứ luồng Observable nào để thực hiện các truy vấn mong muốn.

**Functional**: đơn giản hóa code, sử dụng các function input và output rõ ràng qua các luồng có thể quan sát được.

**Async error handling**: try/catch truyền thống luôn bất lực với việc xử lý bất đồng bộ. Nhưng với ReactiveX được trang bị bộ tích hợp để xử lý lỗi.

**Concurrency made easy**: Observable và Schedulers trong ReactiveX cho phép trừu tượng thread, synchronization.

**2.9.3 Observer là gì?**

ReactiveX được kế thừa mô hình observer để hỗ trợ xử lý tuần tự data và các luồng sự kiện. Và nó trừu tượng hóa các thread để ta có thể thấy được các luồng xử lý dữ liệu hoặc các luồng sự kiện 1 cách rõ ràng hơn.

Observer parttern là một software design parttern mà trong đó mỗi object được gọi là 1 subject, cung cấp một danh sách các object phụ thuộc gọi nó là Observer. Và nó sẽ tự thông báo khi có bất cứ thay đổi nào trong trạng thái của object.

Ví dụ: Anh A là giám đốc công ty X và dưới anh A có 1 người trưởng phòng (observer) và dưới anh trưởng phòng này có 30 nhân viên. Khi anh A nói với trưởng phòng là hãy thông báo cho tất cả nhân viên được nghỉ ngày hôm nay. Thì anh trưởng phòng sẽ báo cho tất cả 30 nhân viên còn lại thông báo từ anh A.

Trong vd trên anh trưởng phòng giống như 1 Observer cho đối tượng A (anh giám đốc). Và 30 nhân viên kia là các Object khác phụ thuộc vào anh A. Khi anh A có 1 thuộc tính thay đổi thì Observer sẽ tự động thông báo cho 30 nhân viên còn lại.

**2.10 Giới thiệu về RxSwift**

**2.10.1 Tổng quan**

RxSwift  là 1 phần của ReactiveX (thường gọi là “Rx”). Rx được sử dụng ở rất nhiều ngôn ngữ và platform khác nhau. Rx là sự kết hợp của những ý tưởng hay nhất từ Observer pattern, Iterator pattern và functional programming.

Rxbắt nguồn từ .Net/C#, sau đó nó phát triển mạnh mẽ với Ruby-ists, JavaScripters, đặc biệt là Java và Android. RxSwift là framework sử dụng cho ngôn ngữ Swift theo kỹ thuật reactive.

RxSwift sẽ giúp việc lập trình trở nên đơn giản hơn. Thay cho notifications, một đối tượng khó để test, ta có thể sử dụng signals. Thay cho delegates, thứ tốn rất nhiều code, ta có thể viết blocks và bỏ đi switches/ifs lồng nhau. Ta còn có thể sử dụng KVO, IBActions, filters, MVVM và nhiều tiện ích khác được hỗ trợ mượt mà trong RxSwift.

**2.10.2 Lợi ích của RxSwift**

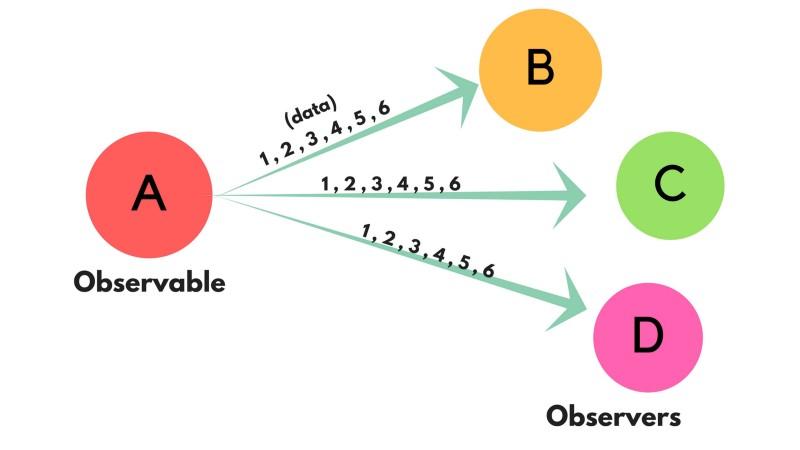
Lợi ích khi sử dụng RxSwift cũng chính là lợi ích khi lập trình Reactive Extension

+ Đối với ứng dụng sử dụng bảng để biểu diễn dữ liệu như App Giá, Reactive thể hiện ưu điểm của mình ở việc: Code ngắn; Binding dữ liệu ngay lập tức cho TableView mỗi khi request được data; Dễ quản lí code; Phù hợp với ứng dụng cần phản hồi ngay lập tức trong thao tác.

+ Code có khả năng tái sử dụng cao, Dễ hiểu và chính xác vì Rx nâng cao mức trừu tượng (abstraction) và loại bỏ các trạng thái trung gian.

+ Dễ học, RxSwift có mô hình Reactive tương đối dễ hiểu.

+ Không có memory leak. Đặc tính chủ yếu của RxSwift là sử dụng Observable và Observer. Observable có nhiệm vụ theo dõi đối tượng, và thực hiện hành động cụ thể bằng cách truyền đến các Observe con để thực thi code logic, ở đây gọi là các luồng. Các Observables này đều được gói trong một Bag gọi là disposeBag với nhiệm vụ sẽ giải phóng tài nguyên mỗi khi mà hoàn thành nhiệm vụ của mình hoặc bị lỗi trong quá trình hoạt động hoặc View bị hủy. Điều này tương tự với việc ứng dụng sẽ không tốn bộ nhớ dư thừa.



**2.11 Giới thiệu về RxCocoa**

**2.11.1 Tổng quan**

RxCocoa là framework thuộc ReactiveX, chủ yếu cho phép các thành phần giao diện như TableView, PickerView, ImageView… cũng có khả năng Reactive. Thư viện này sẽ hỗ trợ thư viện UIKit trong Xcode giúp tăng khả năng đọc hiểu và giảm xử lí rườm rà.

**2.11.2 Vì sao sử dụng RxCocoa**

Bên dưới là đoạn mã lập trình cho bảng tableView, bất cứ khi nào người dùng bấm chọn vào một cell trong bảng, sẽ chuyển đến view mới đồng thời cung cấp dữ liệu cho view mới. Ở đây có sử dụng navigationController nhằm điều hướng sang view mới nhưng không ngắt hoạt động của view cũ, bất cứ khi nào bấm nút Back để trở lại View cũ thì dữ liệu còn nguyên vẹn.



Đoạn code trên từ dòng self.tableView.rx.itemSelected sẽ thay thế cho hàm sau với nội dung tương tự

func tableView(\_ tableView: UITableView, didSelectRowAt indexPath: IndexPath) {

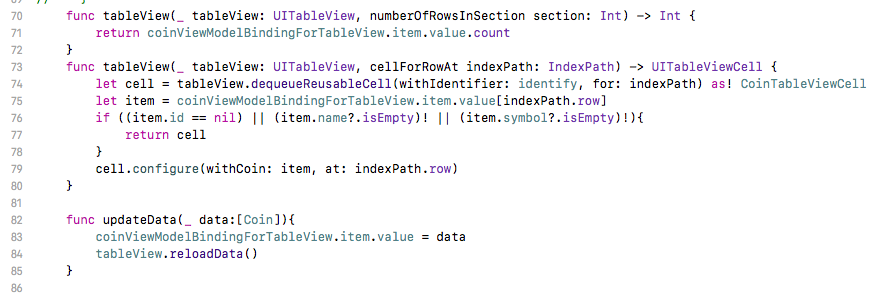
(Mã code)

}

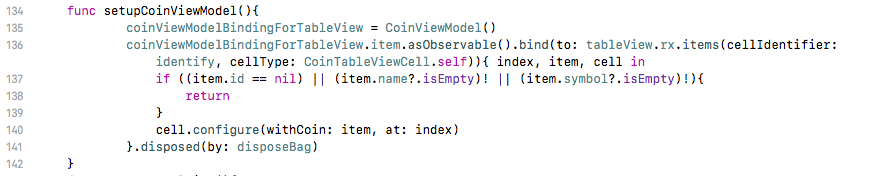
Nhìn hai đoạn mã trên thì ta sẽ không thấy sự khác biệt, tuy nhiên cứ mỗi tương tác với bảng ta sẽ phải override một hàm trong khi tất cả có thể thu về một hàm duy nhất để tiện cho việc sửa chữa bằng việc dùng RxCocoa. Ta sẽ thấy sự khác biệt hơn khi Binding dữ liệu bằng RxCocoa so với cách thông thường là kế thừa từ UITableViewDelegate, UITableViewDataSource.

UITableViewDelegate dùng để xử lí các sự kiện như didSelectRowAt ở trên, còn UITableViewDataSource dùng để cài đặt các thông số dữ liệu cho tableView.

**Khi binding dữ liệu bằng phương pháp thông thường**



**Và khi binding dữ liệu bằng RxCocoa**



Chưa tính đến việc RxCocoa sẽ tối ưu bộ nhớ để chống memory leak và tăng hiệu suất load trang bằng thuật toán thì việc tường minh bằng các đoạn mã trên dễ hiểu hơn, gom nhóm mã code lại vào một hàm và số lượng dòng code giảm đi đã là một lợi thế bảo trì, sữa chữa, học tập cho ứng dụng sử dụng RxCocoa để kết hợp với RxSwift.

RxCocoa là một phần của ReactiveX nên thường đi chung với RxSwift, RxAlamofire,...

**2.12 RxAlamofire**

**2.12.1 Tổng quan**

RxAlamofire được phát triển từ framework Alamofire chuyên xử lí request. Đặc biệt là RestfulAPI, RxAlamofire cung cấp một số hàm thực thi request, response đã được tối ưu. Cho phép lập trình theo hướng Reactive Programming, xử lí với các luồng riêng.

**2.12.2 Lợi ích**

RxAlamofire tối ưu quá trình xử lí với RestfulAPI để lập trình viên tập trung vào Business Logic Code.

**CHƯƠNG 3: GIỚI THIỆU ỨNG DỤNG APP GIÁ**

**3.1 Mục tiêu:**

Sử dụng Swift như ngôn ngữ chính để xây dựng ứng dụng App Giá trên hệ điều hành di động IOS giúp người sử dụng tại Việt Nam cập nhật nhanh chóng tình hình giá Tiền điện tử, giá vàng, giá xăng dầu. Ứng dụng còn cung cấp chức năng quy đổi tiền tệ, so sánh biểu đồ phát triển của các loại tiền. Ngôn ngữ chính của ứng dụng là Tiếng Việt.

**3.2 Lĩnh vực**

Chuyên ngành: Công nghệ phần mềm

Chuyên môn: Sử dụng trình biên dịch Xcode trên nền tảng MacOS để lập trình ứng dụng App Giá sử dụng ngôn ngữ Swift. Sử dụng nhiều Framework công nghệ mới như RxSwift, RxAlamofire, RxCocoa, Charts.

Lĩnh vực liên quan: Tài chính

**3.3 Tính năng**

* Xem giá Tiền điện tử: Cập nhật liên tục giá Tiền điện tử và thông tin chi tiết của từng đồng Tiền điện tử.
* Xem giá vàng: Cập nhật liên tục giá vàng tại từng vùng Việt Nam
* So sánh Tiền điện tử: So sánh biểu đồ phát triển của các đồng tiền theo thời gian.
* Đổi tiền tệ: Quy đổi tiền tệ từ loại tiền này sang loại tiền khác   
  (ví dụ: USD/VND)

**3.4 Yếu tố công nghệ**

* Sử dụng giả lập VMWare 14.0 để chạy hệ điều hành MacOS trên laptop Lenovo T420, Chip xử lí Intel® Core™ I5-2520M CPU @2.50GHz, RAM 8GB, 64bit, Display Intel® HD Graphics 3000.
* Phần mềm lập trình Xcode 9.0.
* Giả lập di động IOS.
* Test trên iPhone 6S.

**3.5 Phạm vi đối tượng**

* Đối tượng người Việt .
* Độ tuổi bất kì, người có nhu cầu giao dịch Tiền điện tử

**3.6 Các chức năng**

**3.6.1 Theo dõi tiền điện tử**

**3.6.1.1 Giao diện chính**

**3.6.1.2 Giao diện chi tiết tiền điện tử**

**3.6.1.3 Giao diện các chợ giao dịch tiền điện tử**

**3.6.1.4 Giao diện lịch sử tiền điện tử**

**3.6.2 Theo dõi giá vàng**

**3.6.3 Chuyển đổi tiền**

**3.6.4 So sánh biểu đồ**

**3.6.4.1 Giao diện chính**

**3.6.4.1 Giao diện so sánh biểu đồ**

**3.6.5 Menu**

**3.6.6 Giao diện thông báo**

**3.7 Mô hình hoạt động**

**3.8 Cơ sở dữ liệu**

**CHƯƠNG 4: ĐÁNH GIÁ VÀ KẾT LUẬN**

**4.1 Đánh giá**

* Ứng dụng đã hoàn thiện và bước đầu có thể đăng lên AppStore để đi vào sử dụng.
* Tuy nhiên giao diện cần có thời gian nghiên cứu thêm để hoàn thiện hơn phù hợp với người dùng.

**4.2 Kết luận**

Sau khi hoàn thành ứng dụng, nhận thấy ứng dụng còn có thể phát triển được thêm nữa:

* Thêm tính năng dự đoán khả năng tăng hay giảm của tiền điện tử dựa vào các thuật toán mô hình phát triển lịch sử dữ liệu.
* Thêm tính năng so sánh các loại tiền điện tử với nhiều loại biểu đồ khác nhau tùy vào người dùng chọn.
* Thêm tính năng đọc báo liên quan đến tiền ảo dựa vào API của một số nguồn.
* Nghiên cứu để lập trình cho phép người dùng kết nối đến ví tiền điện tử và giao dịch như một ứng dụng giao dịch thực sự.
* Lập trình cho phép chuyển đổi giao diện sang ngôn ngữ Tiếng Anh để có thể sử dụng ở mức toàn cầu.

**TÀI LIỆU THAM KHẢO**