

ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH
TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN
KHOA CÔNG NGHỆ PHẦN MỀM

-----***-----



BÁO CÁO CHUYÊN ĐỀ TỐT NGHIỆP

**CHUYÊN ĐỀ MOBILE & PERVASIVE
COMPUTING (UBIQUITOUS COMPUTING)**

**ĐỀ TÀI: XÂY DỰNG ỨNG DỤNG
ĐỌC TIN TỨC**

Giảng viên hướng dẫn: Ths. Nguyễn Trác Thúc

Sinh viên thực hiện: Lê Minh Trạng - 12520452

Phạm Anh Đại – 12520048

Nguyễn Thanh Tân – 12520376

TP. Hồ Chí Minh, tháng 12 năm 2016

[illegible]

LỜI CẢM ƠN

Đầu tiên, chúng em xin gửi lời cảm ơn chân thành đến tập thể quý Thầy Cô Trường Đại học Công nghệ thông tin – Đại học Quốc gia TP. HCM và quý Thầy Cô khoa Kỹ thuật phần mềm đã giúp cho nhóm chúng em có những kiến thức cơ bản làm nền tảng để thực hiện đề tài này.

Đặc biệt, chúng em xin gửi lời cảm ơn và lòng biết ơn sâu sắc nhất tới Thầy Nguyễn Trác Thức (Giảng viên hướng dẫn môn học chuyên đề tốt nghiệp Mobile & Pervasive Computing). Thầy đã trực tiếp hướng dẫn tận tình, sửa chữa và đóng góp nhiều ý kiến quý báu giúp nhóm chúng em hoàn thành tốt báo cáo môn học chuyên đề tốt nghiệp của mình.

Trong thời gian một học kỳ thực hiện chuyên đề tốt nghiệp, chúng em đã vận dụng những kiến thức nền tảng đã tích lũy đồng thời kết hợp với việc học hỏi và nghiên cứu những kiến thức mới. Từ đó, chúng em vận dụng tối đa những gì đã thu thập được để hoàn thành một báo cáo chuyên đề tốt nghiệp tốt nhất. Tuy nhiên, trong quá trình thực hiện, chúng em không tránh khỏi những thiếu sót. Chính vì vậy, chúng em rất mong nhận được những sự góp ý từ phía các Thầy Cô nhằm hoàn thiện những kiến thức mà nhóm chúng em đã học tập và là hành trang để nhóm chúng em thực hiện tiếp các đề tài khác trong tương lai.

Xin chân thành cảm ơn các quý Thầy Cô!

MỤC LỤC

LỜI CẢM ƠN	2
MỤC LỤC	3
MỞ ĐẦU.....	5
DẪN NHẬP	5
MỤC TIÊU VÀ NỘI DUNG THỰC HIỆN ĐỀ TÀI	5
ĐỐI TƯỢNG VÀ PHẠM VI NGHIÊN CỨU ĐỀ TÀI.....	6
BỐ CỤC ĐỀ TÀI.....	6
KẾT CHƯƠng	7
CHƯƠNG 1: TỔNG QUAN VỀ MOBILE & PERVASIVE	
COMPUTING	9
1. 1 THỰC TRẠNG VỀ MOBILE & PERVASIVE COMPUTING TRONG	
XU HƯỚNG PHÁT TRIỂN NGÀY NAY	9
1. 2 GIỚI THIỆU VỀ MOBILE & PERVASIVE COMPUTING	12
1. 3 ỨNG DỤNG THỰC TẾ CỦA MOBILE & PERVASIVE COMPUTING	
.....	14
1. 3. 1 Mobile	14
1. 3. 2 Pervasive Computing	19
1. 4 KẾT CHƯƠng	24
CHƯƠNG 2: CÔNG NGHỆ SỬ DỤNG	25
2. 1 GIỚI THIỆU VỀ UNIVERSAL WINDOWS PLATFORMS (UWP) ...	25
2. 1. 1 UWP là gì?.....	25
2.2.2 Đặc điểm của UWP.....	26
2.2.3 Kết luận.....	27
2. 2 TÌM HIỂU VỀ XAML, CẤU TRÚC PROJECT UWP VÀ CÁC	
CONTROL CƠ BẢN TRONG UWP	28
2. 2. 1 Tìm hiểu về XAML.....	28

2. 2. 2 Cấu trúc project UWP	30
2. 2. 3 Các control cơ bản.....	34
2. 3 TÌM HIỂU VỀ MÔ HÌNH MVVM	38
2. 3. 1 Khái niệm về mô hình MVVM	38
2.3.2 Thiết kế UI và khai báo các đối tượng Logic, MVVM trong ứng dụng đọc tin tức.....	39
2. 4 RSS.....	41
2. 4. 1 Giới thiệu cấu trúc và định dạng tệp tin RSS (Really Simple Syndication).....	41
2. 4. 2 Những điểm mạnh của RSS.....	42
CHƯƠNG 3: PHÂN TÍCH HỆ THỐNG	45
3. 1 PHÂN TÍCH HỆ THỐNG VỀ CHỨC NĂNG	45
3. 1. 1 Lập biểu đồ luồng dữ liệu mức khung cảnh.....	45
3. 1. 2 Lập biểu đồ luồng dữ liệu mức đỉnh.....	45
3. 1. 3 Lập biểu đồ luồng dữ liệu mức dưới đỉnh.....	46
CHƯƠNG 4: THIẾT KẾ HỆ THỐNG VÀ GIAO DIỆN	47
4. 1 THIẾT KẾ SƠ ĐỒ LỚP	47
4. 1. 1 Use Case.....	48
4. 1. 2 Sequence diagram.....	49
4. 2 GIAO DIỆN ỨNG DỤNG	52
CHƯƠNG 5: TỔNG KẾT	54
5. 1 CÀI ĐẶT ỨNG DỤNG	54
5. 2 CÔNG CỤ PHÁT TRIỂN ỨNG DỤNG	55
5. 3 KẾT QUẢ ĐẠT ĐƯỢC VÀ HẠN CHẾ	55
5. 4 HƯỚNG PHÁT TRIỂN.....	55
5. 5 PHÂN CÔNG CÔNG VIỆC	56
TÀI LIỆU THAM KHẢO	57

MỞ ĐẦU

DẪN NHẬP

Ngày nay cùng với sự phát triển như vũ bão của công nghệ thông tin, các hệ điều hành trên thiết bị di động đang trở nên rất được quan tâm. Các hệ điều hành tiêu biểu như, Symbian của Motorola, iOS của Apple, Android của Google, ... và đặc biệt hệ điều hành em muốn nói tới là Windows Phone của Microsoft – một trong những hệ điều hành phổ biến nhất trên thế giới. Windows Phone đang là hệ điều hành di động của tương lai, được nhiều người ưa chuộng nhất do tính năng dễ sử dụng, mã nguồn mở để phát triển, ...

Kết hợp với đó là sự phát triển nhanh chóng của xã hội, nhu cầu cập nhật tin tức xã hội mọi lúc mọi nơi là rất cần thiết, vì vậy nhóm chúng em đã chọn đề tài *“Tìm hiểu về nền tảng Window hợp nhất UNIVERSAL WINDOWS PLATFORMS và xây dựng ứng dụng đọc tin tức”* với mục tiêu nghiên cứu, tìm hiểu hệ điều hành Window Phone và xây dựng ứng dụng để có thể đáp ứng nhu cầu cập nhật tin tức xã hội hiện nay một cách nhanh chóng và tiện lợi nhất.

MỤC TIÊU VÀ NỘI DUNG THỰC HIỆN ĐỀ TÀI

Với mục tiêu tập trung nghiên cứu các phương pháp, kế hoạch xây dựng cũng như công nghệ mới, các kỹ thuật xây dựng, đề tài đề ra các nội dung cụ thể sau:

- ❖ Tìm hiểu về nền tảng Window hợp nhất UNIVERSAL WINDOWS PLATFORMS và các kỹ thuật, control như XAML (làm giao diện ứng dụng)
- ❖ Tìm hiểu và áp dụng mô hình MVVM
- ❖ Tìm hiểu về RSS.
- ❖ Sau khi nắm bắt được cách thức xây dựng ứng dụng đọc tin tức trên hệ điều hành Window Phone tiến hành phân tích, thiết kế và xây dựng ứng dụng đọc tin tức.

- ❖ Xây dựng hoàn chỉnh ứng dụng đọc tin tức theo nhu cầu sở thích của từng người, có chức năng share tin tức bạn yêu thích qua mạng xã hội, tin nhắn, mail.

ĐỐI TƯỢNG VÀ PHẠM VI NGHIÊN CỨU ĐỀ TÀI

- ❖ Đối tượng: Tất cả những người có thói quen đọc báo, tin tức.
- ❖ Phạm vi không gian: xây dựng ứng dụng đọc tin tức với tên ứng dụng UIT READER.
- ❖ Thị trường hướng đến đầu tiên là Việt Nam.
- ❖ Phạm vi thời gian: Dự định 3 tháng.
- ❖ Thời gian thực hiện đề tài: Từ 01/10/2016 đến 31/12/2016

Giai đoạn 1: Từ ngày 1/10 - 25/10:

- a. Tìm hiểu công nghệ, sau đó chọn kỹ thuật (UWP).
- b. Phân tích yêu cầu của đề tài (phân tích, thiết kế hệ thống, thiết kế giao diện), sau đó lên kế hoạch xây dựng đề tài.

Giai đoạn 2: Từ ngày 26/10 – 30/11:

- a. Xây dựng ứng dụng theo kế hoạch, yêu cầu đã thực hiện.
- b. Hoàn thành prototype của đề tài.

Giai đoạn 3: Tháng 1/12 - 23/12:

- a. Review 3 lần trước khi đóng đề tài.
- b. Viết báo cáo.

BỐ CỤC ĐỀ TÀI

Đề tài được bố cục gồm các chương mục như sau:

- ❖ **Mở đầu:** Giới thiệu tóm tắt về động cơ, mục tiêu, nội dung nghiên cứu, bố cục chung của đề tài và kết luận.
- ❖ **Chương 1:** Trình bày tổng quan về Mobile & Pervasive Computing và tầm ảnh hưởng đến thế giới hiện tại.

- ❖ **Chương 2:** Trình bày tổng quan về UWP và các kỹ thuật sử dụng để xây dựng đề tài.
- ❖ **Chương 3:** Trình bày chi tiết về các yêu cầu của hệ thống.
- ❖ **Chương 4:** Trình bày chi tiết về thiết kế của hệ thống và giao diện của ứng dụng.
- ❖ **Chương 5:** Trình bày về cách cài đặt, sử dụng demo, nêu lên các kết quả đạt được, hạn chế và hướng phát triển của chuyên đề tốt nghiệp.
- ❖ **Phân chia công việc**
- ❖ **Tài liệu tham khảo**

KẾT CHƯƠng

Di động đang và sẽ trở thành xu thế của tương lai. Cho nên các hãng thiết bị di động nổi tiếng trên thế giới vô cùng khuyến khích các nhà lập trình viên lên ý tưởng và làm ra một ứng dụng trên ý tưởng của mình. Google cũng không phải là ngoại lệ họ cũng đang khuyến khích các nhà lập trình viên xây dựng nhiều ứng dụng trên nền tảng Window Phone để phục vụ nhu cầu của người sử dụng.

Cùng với đó, vai trò của các ứng dụng đọc tin tức trên thiết bị di động ngày càng trở nên quan trọng. Khác với báo chí truyền thông có giới hạn cập nhật tin tức, các tờ báo trực tuyến đã cung cấp được sự tiện lợi trong việc cập nhật và phát hành thông tin. Về phía người dùng, họ có thể xem thông tin mọi lúc mọi nơi với thiết bị di động của mình. Về phía những người cung cấp thông tin, họ có thể dễ dàng cập nhật những tin tức mới nhất, thời sự nhất. Do đó việc sử dụng các ứng dụng đọc tin tức luôn là điều cần thiết hiện nay nhằm đáp ứng nhu cầu cập nhật thông tin của mỗi người.

Vì vậy nhằm đáp ứng yêu cầu cập nhật tin tức của người dùng ở mọi nơi, mọi lúc với thiết bị di động của mình, chủ yếu là với những người sử dụng hệ điều hành Window Phone nên em đã chọn và thực hiện đề tài “Tìm hiểu về nền tảng

Window hợp nhất UNIVERSAL WINDOWS PLATFORMS và xây dựng ứng dụng độc tin tức”.

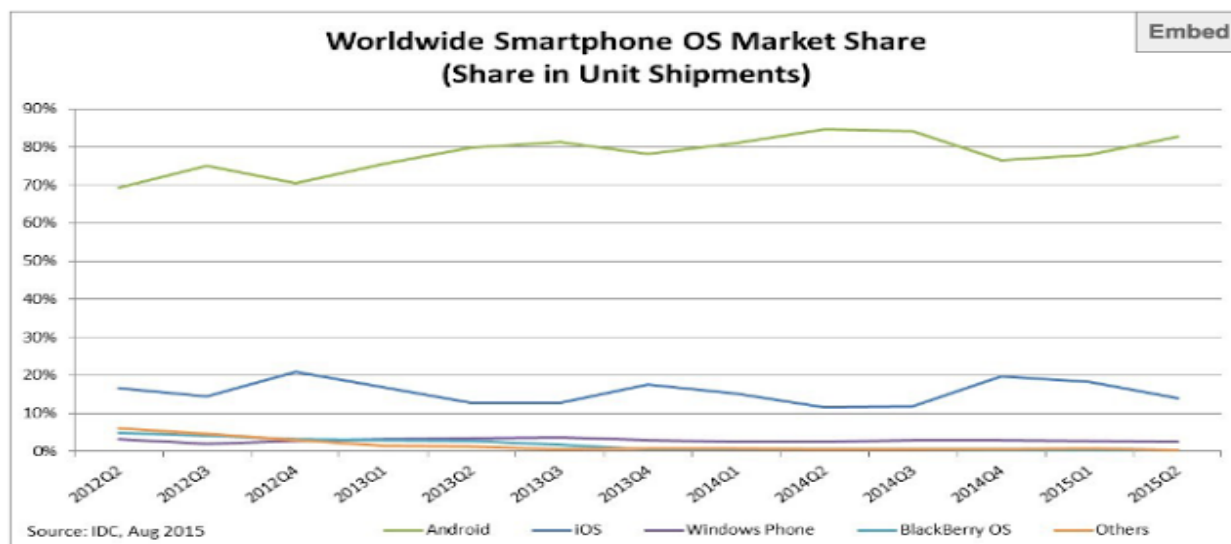
CHƯƠNG 1: TỔNG QUAN VỀ MOBILE & PERVASIVE COMPUTING

Chúng ta đang sống trong thời kỳ mà mọi thứ gần như chuyển lên điện thoại di động thông minh (Smartphone), các thiết bị IOT (nhà thông minh, xe hơi, đồng hồ...). Chúng ta dễ dàng bắt gặp hình ảnh con người cầm cuối nhìn vào chiếc smartphone của mình trên đường phố, trên phương tiện giao thông, hay trong quán cafe, những nơi công cộng. Và gần hơn nữa là trong chính ngôi nhà của chúng ta, bạn có thể tương tác với các thiết bị điện tử trong nhà như bóng đèn, tủ lạnh, cửa chính... từ xa mà không cần phải tiếp cận chúng. Chiếc điện thoại, máy tính phổ biến (Pervasive computing) đã trở thành một phương tiện không thể thiếu trong đời sống cá nhân của con người hiện đại.

1. 1 THỰC TRẠNG VỀ MOBILE & PERVASIVE COMPUTING TRONG XU HƯỚNG PHÁT TRIỂN NGÀY NAY

Những con số thống kê nói lên nhiều điều!

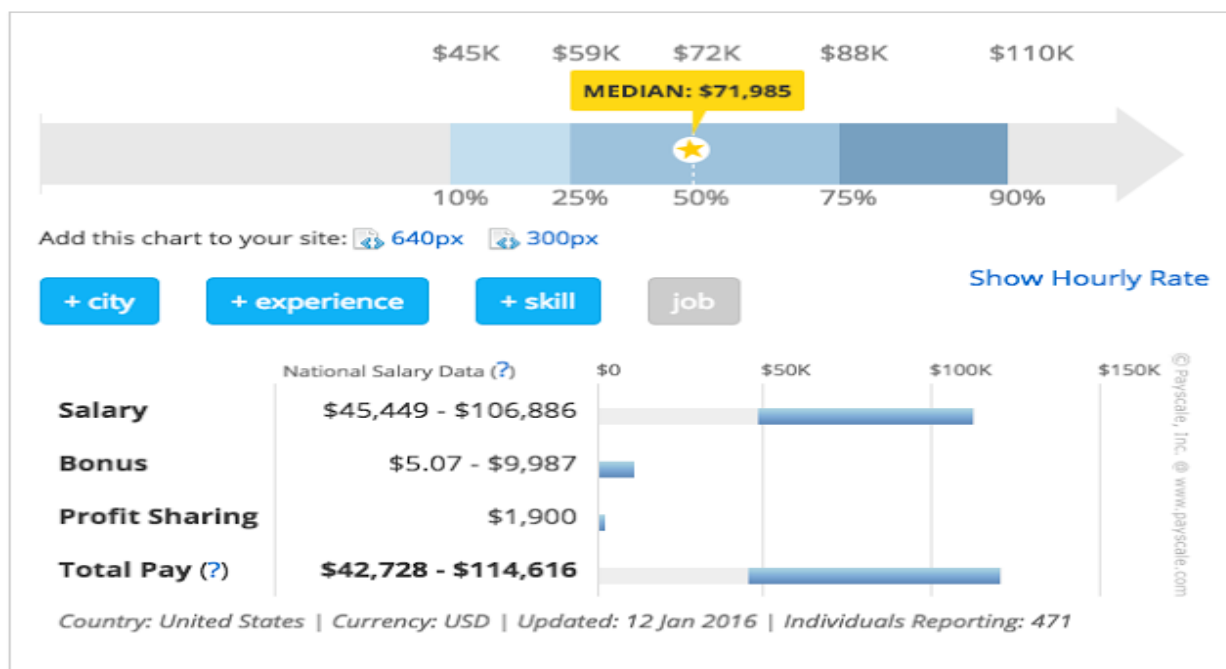
Theo IDG, thị trường smartphone toàn cầu tăng 13% sau một năm (2015), với 341. 5 triệu chiếc được bán ra. Trong đó, Android chiếm 82. 8% thị phần. Samsung vẫn là nhà sản xuất smartphone dẫn đầu thế giới ở phân khúc tầm trung và cao cấp. Theo sau đó là iOS chiếm 13. 9%, Windows Phone là 2, 6%.



Period	Android	iOS	Windows Phone	BlackBerry OS	Others
2015Q2	82.8%	13.9%	2.6%	0.3%	0.4%
2014Q2	84.8%	11.6%	2.5%	0.5%	0.7%
2013Q2	79.8%	12.9%	3.4%	2.8%	1.2%
2012Q2	69.3%	16.6%	3.1%	4.9%	6.1%

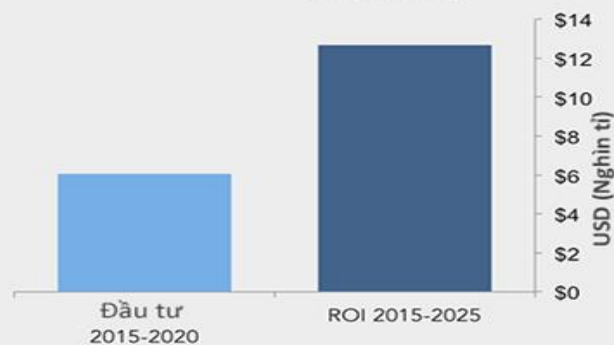
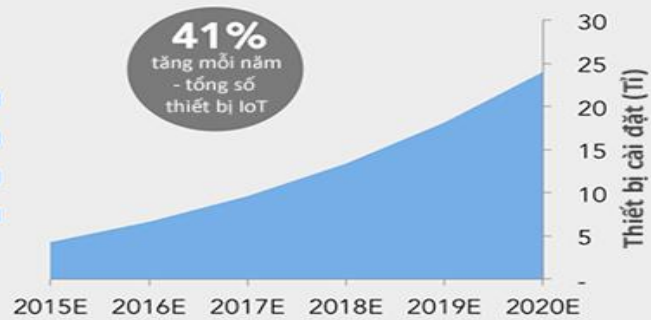
Source: IDC, Aug 2015

Thị phần của hệ điều hành cho smartphone



Mức lương trung bình của lập trình viên di động theo thống kê của Payscale [1]

QUY MÔ THỊ TRƯỜNG



Đối tượng hệ sinh thái IoT phục vụ:
Doanh nghiệp, Chính phủ và Người dùng



Quy mô thị trường IoT tới năm 2020 (nguồn của Business Insider)

1. 2 GIỚI THIỆU VỀ MOBILE & PERVASIVE COMPUTING

Điện thoại của thế kỷ 21

Điểm khác biệt lớn nhất của chiếc smartphone so với các thiết bị di động khác đó là nó "smart"-thông minh. Đó là nó có một hệ điều hành mạnh mẽ, trên đó, người dùng có thể cài vô số ứng dụng tùy thích cho nó. Các ứng dụng có thể chỉ là các trò chơi đơn giản, đến các ứng dụng phục vụ công việc hằng ngày, hay những ứng dụng chuyên dụng phục vụ cho nghề nghiệp chuyên môn. Với cấu hình phần cứng mạnh mẽ, chiếc điện thoại di động có năng lực mạnh hơn những máy tính để bàn hay laptop cách đây vài năm. Nói một cách ngắn gọn, dễ hiểu là người dùng có trong tay một chiếc máy tính mạnh mẽ trên tay, có thể mang đi khắp mọi nơi.

Điểm đặc biệt thứ 2 của smartphone đó là nó có thể kết nối mạng không dây bất cứ nơi đâu qua các công nghệ Wifi, 3G, hay thậm chí 4G. Chính điều này đã làm cho người dùng sử dụng smartphone có thể kết nối vào Internet mọi lúc, mọi nơi. Họ có thể cập nhật tin tức cũng như đóng góp chia sẻ thông tin với bạn bè ở khắp mọi nơi.

Điểm đặc biệt thứ 3 đó là chiếc điện thoại bây giờ nó là "cửa ngõ" của chúng ta bước vào không gian số! Mỗi người chúng ta ngoài 1 thực thể bằng xương bằng thịt tồn tại ngoài đời thường, còn tồn tại nhiều "thực thể số" tồn tại trên không gian mạng: một hình ảnh bóng bẩy trên Facebook, một tài khoản Gmail/Yahoo/MSN... dùng để liên lạc, hay tài khoản Viber, Skype, WhatsUp dùng để gọi điện thoại miễn phí, hay thậm chí là 1 bang chủ võ lâm lãnh đạo hàng nghìn quân tướng trong một game online nào đó. Tất cả đều hội tụ trên chiếc điện thoại của chúng ta. Do đó, chiếc điện thoại trở nên vật bất ly thân của chúng ta bởi vì những tiện ích do chúng mang lại.

Ngoài ra, chiếc điện thoại di động ngày nay, nó được trang bị vô số cảm biến như GPS, cảm biến rung/gia tốc, cảm biến điện dung, Camera, cảm biến vân tay, cảm biến nhiệt độ, v. v. Đã làm cho các ứng dụng trên di động trở nên thông minh hơn, phù hợp với người dùng hơn là những ứng dụng trên máy tính để bàn. Điều

này làm cho ứng dụng trên di động trở nên đầy màu sắc, thú vị hơn, tự nhiên hơn là trên máy để bàn.

Máy tính phổ biến (Pervasive computing)

Máy tính phổ biến chính là tương lai của mạng máy tính cho phép truyền thông không dây giữa các cá nhân và các thiết bị máy tính và giữa các thiết bị chính nó.

Các thiết bị máy tính phổ biến là rất nhỏ, với các vi xử lý ẩn bên trong và được nhúng vào trong bất kỳ loại đối tượng nào như quần áo, nhà, xe hơi, đồng hồ,... (còn được gọi là Ubiquitous Computing).

Mục đích của máy tính phổ biến (Pervasive Computing) là dành cho máy tính có sẵn bất cứ nơi nào cần thiết. Nó trải rộng tính thông minh và kết nối tất cả mọi thứ.

- Tàu thủy, máy bay, ô tô, cầu, đường hầm, máy, tủ lạnh, cánh cửa, thiết bị chiếu sáng, giày dép, nón, công cụ, nhà và thậm chí cả những thứ như cốc cà phê và những con chip gắn trên cơ thể con người.
- Truy cập internet di động, truyền thông không dây, các thiết bị cầm tay, và bluetooth đã làm cho pervasive computing trở thành hiện thực.

Các ví dụ về máy tính phổ biến (Pervasive Computing)

- Dịch vụ mạng băng thông cao tiên tiến
- Thiết bị mạng, cảm biến, dụng cụ, giao diện
- Thông tin và tính toán các nguồn lực
- Viễn thông bao gồm cả thoại không dây toàn cầu và các mạng dữ liệu băng rộng
- Công nghệ phần mềm về tương tác người – máy.

1. 3 ỨNG DỤNG THỰC TẾ CỦA MOBILE & PERVASIVE COMPUTING

1. 3. 1 Mobile

Ngày nay, hầu như tất cả mọi người, từ già trẻ gái trai, đều có một chiếc điện thoại thông minh bên mình. Smartphone giúp chúng ta kết nối nhanh chóng với Internet – một kho lưu trữ thông tin khổng lồ. Thực tế là, các nhà khoa học cũng bắt đầu cảm thấy hứng thú với chúng, và họ bắt đầu thiết kế các phần mềm hỗ trợ mọi người và góp phần thay đổi thế giới.

Máy đo mức độ ô nhiễm



Các nhà khoa học từ Đại học California đã phát triển một add-on nhỏ cho smartphone để đo mức độ ô nhiễm không khí. Phần mềm có tên CitiSense này thu thập các dữ liệu từ các cảm biến và hiển thị một bức tranh về mức độ ô nhiễm không khí. Ngoài ra, người dùng có thể xem được dữ liệu về mức độ ô nhiễm của một vùng nào đó qua dữ liệu thu thập được chia sẻ từ smartphone của người khác. Phần mềm này thực sự rất hữu dụng, nhất là với những người bị bệnh hen – họ sẽ biết cần phải tránh xa những vùng nào. Phần mềm còn giúp các nhà khoa học theo dõi chi tiết hơn về mức độ ô nhiễm của vùng nào đó. Tại San Diego, nơi mà phần mềm này được triển khai lần đầu tiên, chỉ có 10 trạm theo dõi mức độ ô nhiễm.

Các nhà nghiên cứu nói rằng trong một vùng có hơn 3 triệu người dân mà chỉ cần 100 người sử dụng phần mềm này là các nhà khoa học đã có thể xây dựng được một cơ sở dữ liệu phong phú.

Kính hiển vi cầm tay



Một chiếc kính hiển vi cầm tay có khả năng soi được virus đã được thiết kế bởi các nhà khoa học của UCLA. Thiết bị này lắp vừa mặt sau của smartphone, và nó được thiết kế để sử dụng trong trường hợp phòng thí nghiệm của bạn không có đầy đủ dụng cụ. Một trong những tính năng của thiết bị này là đếm số lượng virus, điều này giúp ích cho các bác sĩ để theo dõi hiệu quả điều trị cho bệnh nhân.

Một chiếc kính hiển vi nhỏ hơn đã được phát triển bởi các kỹ sư đến từ Berkely. Cũng như để phục vụ cho nghiên cứu, các kỹ sư tin rằng loại kính này hoàn toàn thích hợp cho tất cả mọi người, nhất là nó giúp trẻ em có thể tiếp cận được với khoa học một cách dễ dàng hơn: chúng sẽ chụp các bức ảnh về mọi thứ xung quanh, và ghi nhớ dễ dàng hơn nhiều so với những bài giảng lý thuyết trên lớp. Các

kỹ sư mong rằng thiết bị này sẽ sớm được sử dụng rộng rãi trong các tiết học khoa học.

Cảnh báo động đất

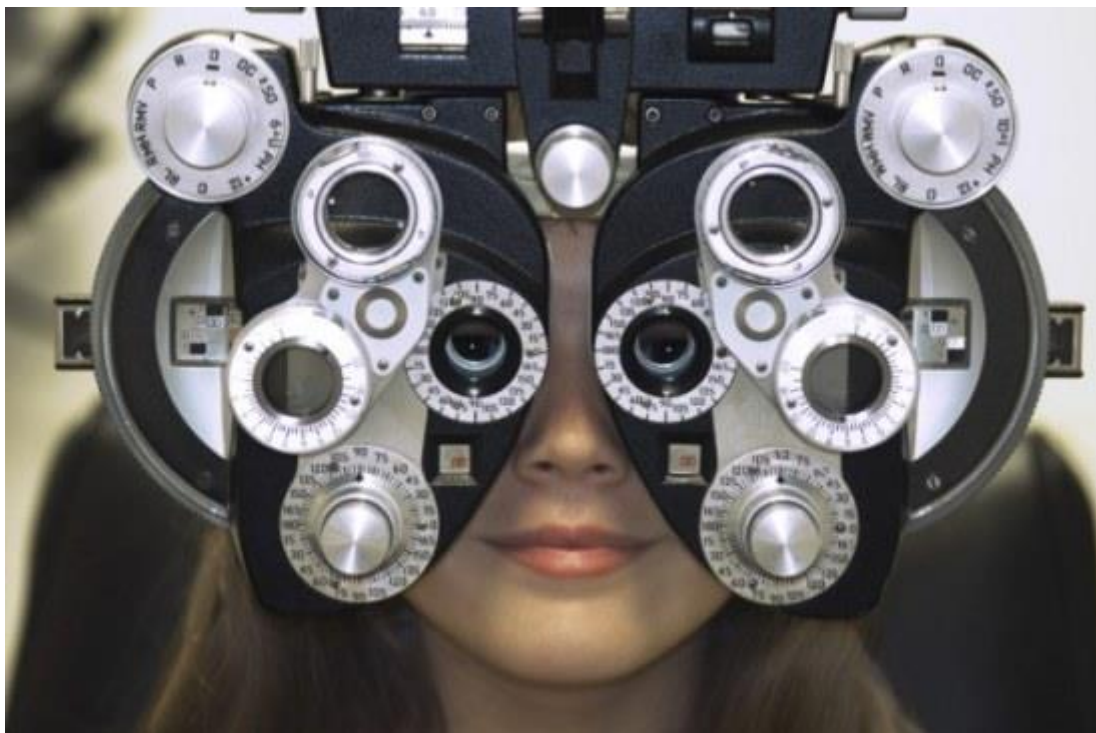


Hầu hết tất cả các smartphone hiện nay đều được trang bị cảm biến gia tốc kế - MEMS accelerometer (microelectromechanical system), nó giúp màn hình điện thoại có thể xoay theo đúng chiều bạn đang đứng. Nhưng nó còn có tác dụng hơn thế nữa. Các nhà địa chấn học từ Viện Địa lý Quốc gia Italy đã sử dụng chip MEMS trong chiếc iphone 4 và 5 để đo thang điểm động đất. Ý tưởng biến chiếc smartphone trở thành một máy đo địa chấn sẽ giúp cho đội cứu hộ xác định được chính xác tâm chấn và ứng cứu kịp thời. Bạn sẽ không phải lo về việc nhiễu tín hiệu, vì các gia tốc kế có khả năng phân biệt được sự khác nhau giữa rung động của một trận động đất với các động tác di chuyển hàng ngày như đi bộ hay chạy nhảy...

Các nhà khoa học hy vọng sẽ sử dụng công nghệ này để tạo nên một mạng lưới cảnh báo sớm các trận động đất. Bằng cách thu thập một số lượng lớn thông tin từ

các điện thoại, xử lý chúng, chúng ta sẽ dự đoán được nơi trận động đất sẽ diễn ra tiếp theo và gửi cảnh báo tới tất cả những người sử dụng phần mềm mà không phải sử dụng một thiết bị phần cứng nào khác.

Phụ kiện Y khoa



Sức mạnh và sự tiện dụng của các thiết bị điện tử đã trợ giúp cho các bác sĩ rất nhiều, nhất là ở những nơi không đầy đủ phương tiện. Một trong những vấn đề sức khỏe lớn nhất ở các nước đang phát triển, chính là đôi mắt. Tật khúc xạ ở mắt chiếm một tỉ lệ lớn trong các bệnh về mắt, hoàn toàn có thể khắc phục bằng cách đeo kính. Tuy nhiên, nếu không phát hiện và điều trị, nó có thể dẫn đến giảm sút khả năng đọc và viết, và từ đó dẫn đến đói nghèo. Khó khăn ở đây, không phải là về kính, giá thành một chiếc kính cận khá rẻ, chỉ khoảng 3\$, tuy nhiên vấn đề chính lại là các test để kiểm tra thị lực. Các nhà khoa học của MIT đã thiết kế một thiết bị để gắn vào smartphone, chỉ cần để chúng sát mắt, tia laser sẽ quét qua và đưa ra kết quả về các vấn đề ở mắt, từ đó sẽ có cách xử trí thích hợp. Giá của thiết bị này chỉ khoảng 2\$, nhưng nó sẽ giúp nâng cao chất lượng sống và cải thiện nhiều về kinh tế - văn hoá đất nước.

Mắt chỉ là một trong số rất nhiều vấn đề sức khỏe có thể được kiểm tra nhờ các phụ kiện cho điện thoại. Các nhiễm khuẩn ở tai, đo chức năng thận và sự xuất hiện của các dị nguyên trong thức ăn hoàn toàn có thể được kiểm tra bằng các phụ kiện cho điện thoại. Đặc biệt, hiện đã có những phụ kiện siêu âm, giúp hỗ trợ đắc lực cho các bác sĩ và các nữ hộ sinh ở những nơi không có đủ thiết bị.

Điện toán đám mây



Trong nghiên cứu, người ta thường phải xử lý một lượng lớn dữ liệu, và kéo theo đó là một lượng lớn máy tính để vận hành. Một trong những cách thực hiện, đó là chia sẻ công việc cho các máy tính khác nhau, vì vậy các nhà khoa học đã tạo một phần mềm để tận dụng khả năng tính toán của các thiết bị Android khắp thế giới. Có tên BOINC, phần mềm này sử dụng các thiết bị trong trạng thái rỗi và đang sạc – ví dụ như cắm sạc qua đêm chẳng hạn – để giúp xử lý một phần dữ liệu.

Một trong những dự án sử dụng BOINC, đó là FightAIDS@Home – dự án phát triển các loại thuốc mới giúp chống lại virus HIV. Đây là một phần của dự án World Commynity Grid được thực hiện bởi IBM, sử dụng các máy tính để bàn và laptop rảnh rỗi. Việc thuê một siêu máy tính để xử lý dữ liệu rất tốn kém – khoảng 1.000\$/giờ, do vậy cách này giúp hoàn thành công việc nhanh hơn mà ít tốn kém hơn. Một dự án nghiên cứu về bệnh nhiệt đới, khi được thực hiện qua phương pháp này sẽ giúp tiết kiệm thời gian thực hiện từ 30 năm xuống còn chỉ một năm.

1.3.2 Pervasive Computing

Dầu khí là một trong những ngành công nghiệp chủ chốt ứng dụng IoT trên diện rộng tới năm 2020.

Dự kiến sẽ có 5,4 triệu thiết bị IoT được triển khai tại các cơ sở khai thác tới năm 2020. Chủ yếu sẽ là các bộ cảm biến kết nối Internet giúp cung cấp thông tin về môi trường.



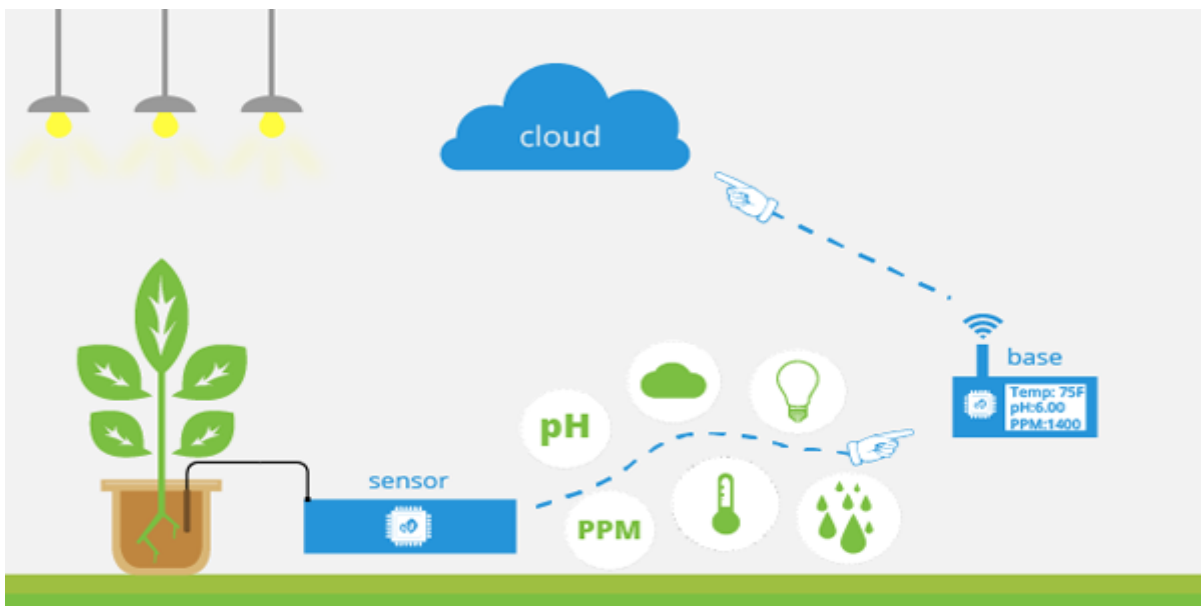
Trong khi đó, xe kết nối đang là xu hướng nổi bật của thiết bị IoT hiện nay. Dự tính tới năm 2020, sẽ có hơn 220 triệu xe kết nối lưu thông trên đường.



Về bảo hiểm, 74% lãnh đạo trong ngành bảo hiểm tin rằng IoT sẽ thay đổi cơ bản chính sách bảo hiểm trong 5 năm tới, 74% có kế hoạch đầu tư phát triển và thực hiện các chiến lược về IoT – theo một nghiên cứu của SMA Research.

Còn với quốc phòng, chi tiêu cho các thiết bị bay không người lái dự kiến sẽ đạt 8,7 tỉ USD vào năm 2020. Ngoài ra, theo dự báo của Frost & Sullivan, sẽ có khoảng 126.000 robot quân sự sẽ được triển khai vào năm 2020.

Về nhà kết nối, tới năm 2030, phần lớn các thiết bị trong gia đình sẽ được kết nối Internet. Các nhà sản xuất cũng sẽ trang bị tính năng kết nối cho bất cứ vật dụng nào mà họ sản xuất ra.

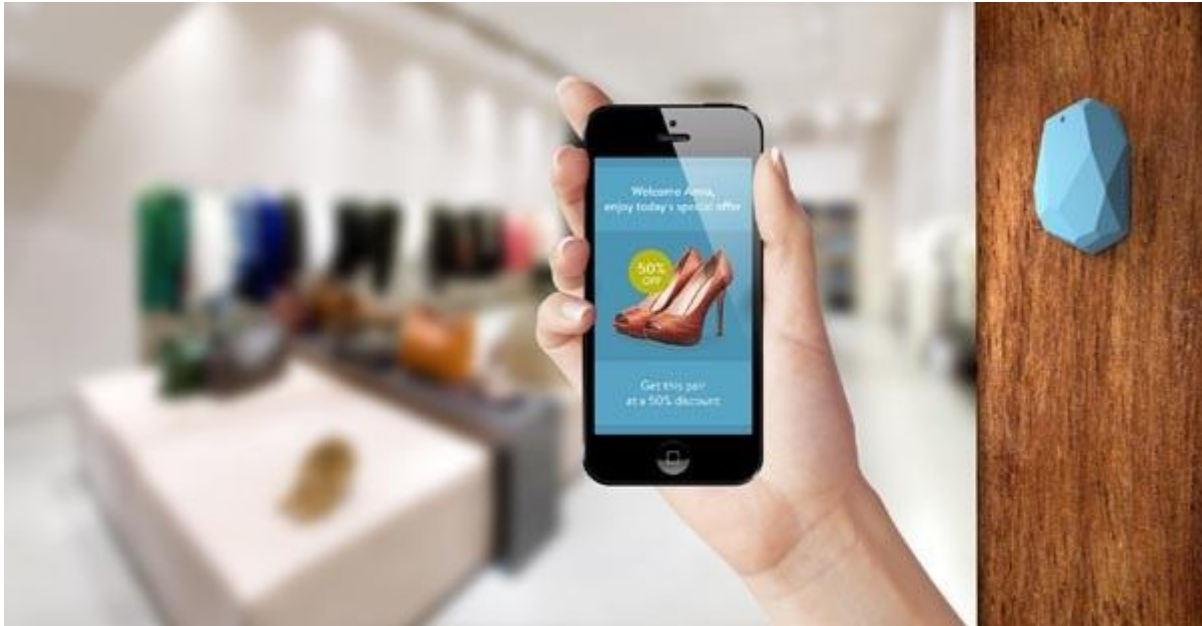


Lĩnh vực nông nghiệp cũng không nằm ngoài vòng xoáy IoT. Dự kiến sẽ có 75 triệu thiết bị IoT được triển khai trong lĩnh vực này vào năm 2020, với tỉ lệ tăng hàng năm đạt 20%. Chủ yếu đó sẽ là những bộ cảm biến đặt trong lòng đất để theo dõi độ axit, nhiệt độ và các thông số giúp canh tác vụ mùa hiệu quả hơn.

Trong **lĩnh vực dịch vụ ăn uống**, sẽ có 310 triệu thiết bị IoT được các công ty cung cấp dịch vụ ăn uống sử dụng vào năm 2020. Phần lớn các thiết bị này sẽ kết nối thông tin tới quầy bán thực phẩm và các công ty sản xuất đồ ăn nhanh. Danh sách các ngành ứng dụng IoT còn rất dài. Chẳng hạn về **hạ tầng, các đô thị** trên thế giới sẽ tăng đầu tư cho hệ thống IoT với mức tăng 30% mỗi năm, từ 36 tỉ USD năm 2014 lên 133 tỉ USD năm 2019. Khoản đầu tư này sẽ tạo ra 421 tỉ USD giá trị kinh tế cho các thành phố toàn cầu năm 2019.

Trong khi đó, các **công ty năng lượng** khắp thế giới đang cố đáp ứng nhu cầu ngày càng cao về năng lượng. Để làm được điều này, họ sẽ lắp đặt gần 1 tỉ thiết bị đo tích hợp chức năng IoT vào năm 2020.

Về **dịch vụ bán lẻ**, thiết bị Beacon kết nối qua bluetooth với các ứng dụng di động đang được sử dụng trong các cửa hàng để theo dõi hành vi mua sắm của người dùng, đồng thời giúp đẩy trực tiếp quảng cáo theo ngữ cảnh vào thiết bị người dùng. Chỉ tính riêng tại Mỹ, doanh thu tạo ra từ Beacon dự đoán sẽ lên tới 44, 4 tỉ USD.



Việc ứng dụng thành tựu công nghệ trong **lĩnh vực dịch vụ khách sạn** cũng được đánh giá khá tốt. Khảo sát của tạp chí Hospitality Technology cho thấy, 31% khách sạn đang sử dụng thẻ hệ khóa cửa mới, 33% có thiết bị điều khiển phòng hiện đại, 16% có TV kết nối mạng Internet và 15% sử dụng thiết bị Beacon trong khách sạn.

Công nghệ cũng giúp giảm chi phí đáng kể cho ngành hậu cần. Các cảm biến gắn trên xe vận chuyển sẽ giúp hạn chế mất mát hoặc hư hỏng hàng hóa. Ở quy mô lớn hơn, những robot kiểu như Amazon Kiva sẽ giúp giảm đáng kể chi phí nhân công trong nhà kho.



Y tế đã và đang là lĩnh vực điển hình của ứng dụng CNTT. Ước tính sẽ có tới 646 triệu thiết bị IoT được ngành y tế sử dụng vào năm 2020. Các thiết bị y tế kết nối có thể thu thập dữ liệu, tự động quy trình và thực hiện nhiều công việc khác. Tuy nhiên, nó cũng tiềm ẩn nguy cơ tin tặc hack vào các thiết bị này khiến cho bệnh nhân gặp nguy hiểm.

Theo **Ngân hàng thế giới WB**, chỉ tính riêng trong năm 2015 đã có gần 3 triệu cây ATM mới được lắp đặt trên toàn thế giới. Một số cây ATM còn hỗ trợ tính năng video trực tiếp, giúp cung cấp sự hỗ trợ tức thời và an toàn cho người sử dụng.

Và cuối cùng, ngôi nhà bạn đang sống sẽ biến thành **nhà thông minh** nhờ các tiến bộ trong ngành xây dựng. Khảo sát của Daintree Networks cho thấy 43% quản lý xây dựng Mỹ tin rằng IoT sẽ tác động cơ bản tới cách thức xây nhà trong 2 hoặc 3 năm tới.

1. 4 KẾT CHƯƠng

Thế kỷ 21 là thế kỷ của di động và máy tính phổ biến (ở đây là các thiết bị IOT). Ở các nước phát triển, mỗi người có thể có nhiều hơn 1 chiếc điện thoại di động bên mình và vô số thiết bị IOT gắn liền với đời sống hằng ngày. Nó mở ra một cơ hội phát triển ứng dụng cho thị trường này là vô cùng to lớn. Thị trường đó còn mở ra một mô hình kinh doanh mới đó là mảnh đất màu mỡ cho các nhà phát triển ứng dụng độc lập hay khởi nghiệp.

Để có thể thành công được trong một môi trường cạnh tranh mang tính toàn cầu này, đòi hỏi các bạn phải có một sự chuẩn bị kỹ lưỡng về thời gian, tiền bạc và quan trọng nhất là sự quyết tâm học hỏi. Tất cả những công cụ, tài liệu đều sẵn có và gần như miễn phí trên mạng. Điều cần nhất của các bạn là một nguồn động lực và môi trường để các bạn học tập, trao đổi và nhận được sự hỗ trợ khi cần thiết.

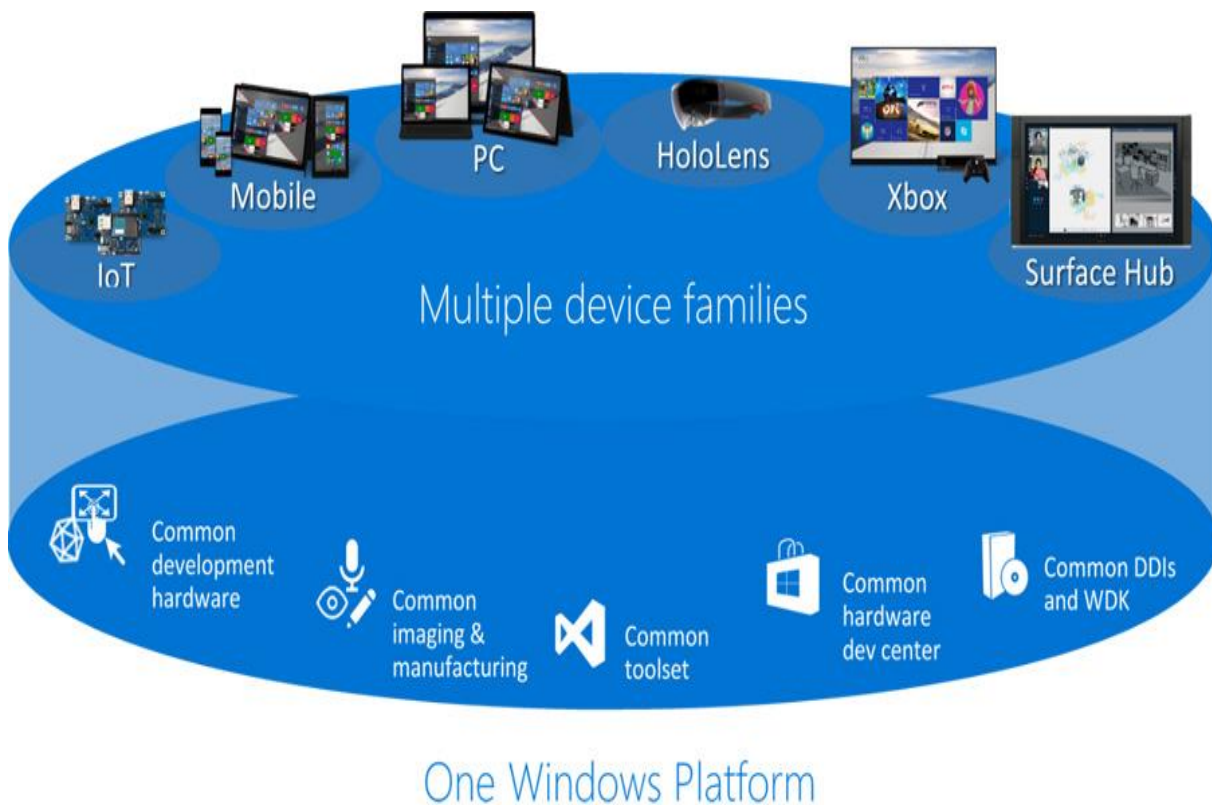
Việc phát triển tư duy lập trình có thể bắt đầu từ lứa tuổi cấp 2 trở đi. Bạn là học sinh phổ thông, hay công chức đi làm, hoặc một người đã về hưu muốn học lập trình đều có thể học được. Hoặc nếu các bạn là sinh viên cần củng cố thêm kiến thức về lập trình, hay cần nắm lại căn bản để tiến xa hơn thì phát triển tư duy lập trình là điểm mấu chốt để các bạn tìm lại đam mê trên con đường nghề nghiệp của mình.

CHƯƠNG 2: CÔNG NGHỆ SỬ DỤNG

2. 1 GIỚI THIỆU VỀ UNIVERSAL WINDOWS PLATFORMS (UWP)

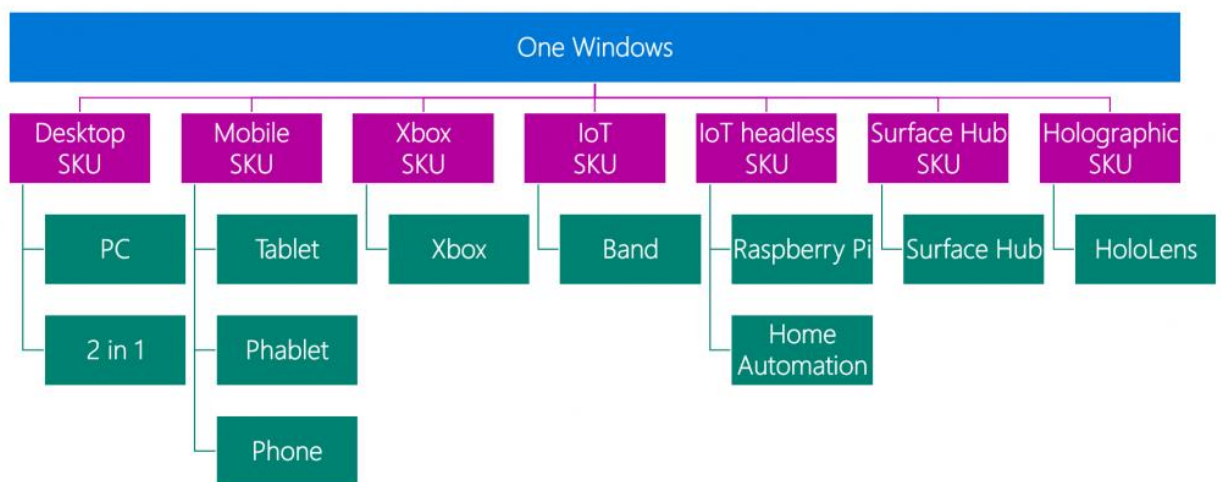
2. 1. 1UWP là gì?

- UWP là một cấu trúc ứng dụng thống nhất giữa các nền tảng được tạo bởi Microsoft và được giới thiệu lần đầu trong Windows 10. Mục đích của nền tảng phần mềm này là giúp phát triển các ứng dụng kiểu Metro chạy trên cả Windows 10 và Windows 10 Mobile mà không cần phải viết lại ứng dụng cho mỗi nền tảng. Nó hỗ trợ cho việc phát triển các ứng dụng Windows bằng C++, C#, VB. NET, hoặc XAML. API được thực hiện bằng C++, và được hỗ trợ trong C++, VB. NET, C#, và JavaScript.
- Được thiết kế như một phần mở rộng của nền tảng WindowsRuntime được giới thiệu lần đầu trong Windows Server 2012 và Windows 8, UWP cho phép nhà phát triển tạo các ứng dụng có thể chạy trên nhiều loại thiết bị.



2.2.2 Đặc điểm của UWP

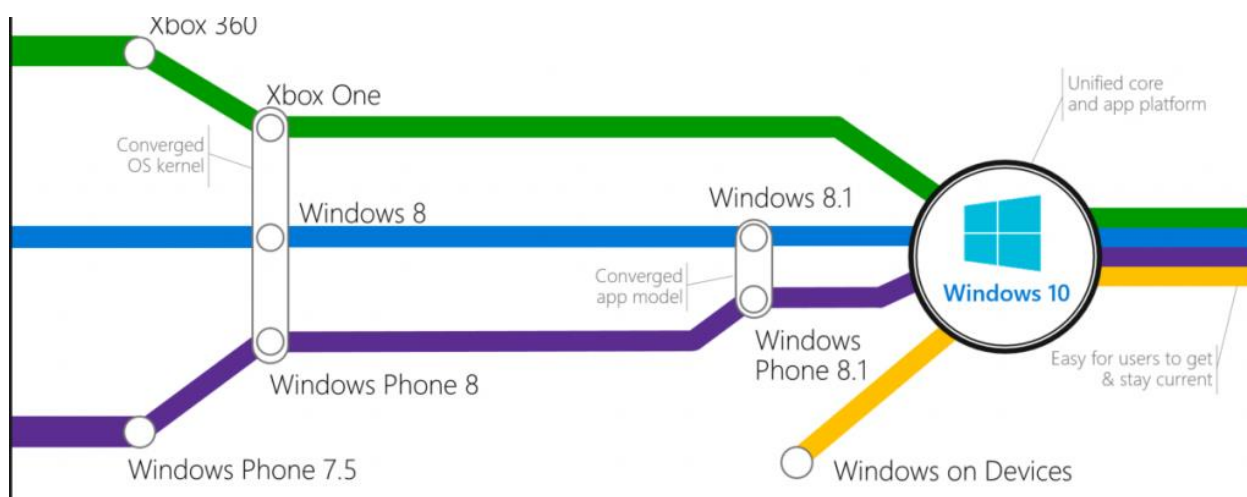
- Điểm tuyệt vời nhất của UWP đó là các bộ tính năng mà nó cung cấp cho app sẽ giống nhau ở tất cả mọi thiết bị chạy Windows 10 và cả các phiên bản Windows tiếp theo trong tương lai. Sự giống nhau này bao gồm việc chỉ có một bộ hàm API giống nhau, một định dạng app như nhau (đuôi *.appx), và ở cấp cao hơn là chỉ có 1 cửa hàng duy nhất để phân phối các app universal. Tất nhiên là ta sẽ cần chỉnh lại giao diện cho phù hợp và dễ dùng, nhưng về cơ bản chỉ cần bật cho máy nào thì máy đó sẽ chạy được app.



- UWP cũng có những tính năng mở rộng riêng cho từng loại thiết bị và Microsoft gọi nó là các Extension SDK và hàm API chuyên biệt. Ví dụ, Extension SDK cho HoloLens sẽ bổ sung thêm tính năng điều khiển bằng cử chỉ cho app, thứ mà máy tính hay điện thoại không có. Extension SDK cho Xbox sẽ có thêm khả năng truy cập và sử dụng cảm biến Kinect để nhận biết chuyển động của người dùng. Bằng cách này, các nhà phát triển có thể dễ dàng tùy biến riêng tính năng cho từng loại máy mà không "đụng chạm" tới phần chung của app.
- UWP là một phần của Windows 10 và Windows 10 Mobile. Các ứng dụng UWP không chạy trên các phiên bản Windows cũ hơn.
- Các ứng dụng sử dụng nền tảng này được phát triển bằng Visual Studio 2015. Các ứng dụng kiểu Metro cũ dành cho Windows 8. 1, Windows Phone

8. 1 hoặc cho cả hai (universal 8. 1) cần được điều chỉnh để chuyển sang UWP.

- Universal App không phải là một khái niệm mới, nó đã được Microsoft giới thiệu trên Windows và Windows Phone 8. 1. Tuy vậy UWP vẫn có nhiều điểm khác biệt so với các ứng dụng Universal trên các phiên bản HĐH trước. Nếu như trên Windows & Windows Phone 8. 1, ứng dụng Universal cần nhiều giao diện được thiết kế riêng để phù hợp với màn hình của từng thiết bị thì với UWP, giao diện được tự động điều chỉnh theo độ phân giải của màn hình. Do đó, công việc của các nhà phát triển ứng dụng được giảm bớt khối lượng, đồng thời giao diện app cũng trở nên đẹp mắt và thân thiện hơn.



2.2.3 Kết luận

- Universal Windows Platform kế thừa nền tảng WinRT của Windows 8/8. 1, đồng thời bổ sung những API mới của Windows 10. Vì vậy ứng dụng UWP sẽ có hiệu năng cao đáng kể và mang lại những trải nghiệm tốt nhất cho người dùng.
- Microsoft đang muốn thống nhất mọi thứ để đơn giản hóa việc phát triển và bảo dưỡng cho cả mình lẫn các nhà phát triển, như vậy thì khi có nâng cấp hay thay đổi gì thì mọi người sẽ được hưởng lợi ngay lập tức, không cần

phải chờ đợi hay viết lại phần mềm từ đầu. Việc bảo trì, sửa lỗi cũng dễ hơn, thống nhất hơn, nhanh chóng hơn. Hãng cũng tốn ít chi phí hơn cho những việc này, tiết kiệm tiền, tức là lợi nhuận cao hơn.

- UWP là một thành phần cực kì quan trọng cho tương lai của Microsoft. UWP sẽ giúp ích cho cả Microsoft, các lập trình viên viết app Windows, và cho cả người dùng chúng ta. Nhờ có UWP, chúng ta sẽ có các app tốt để xài nhanh hơn, được update dễ dàng hơn và thống nhất trên nhiều thiết bị khác nhau.

2. 2 TÌM HIỂU VỀ XAML, CẤU TRÚC PROJECT UWP VÀ CÁC CONTROL CƠ BẢN TRONG UWP

2. 2. 1 Tìm hiểu về XAML

- XAML là gì?
 - ❖ XAML – Extensible Application Markup Language – phát âm zammel
 - ❖ Là một ngôn ngữ đánh dấu dùng để thể hiện các đối tượng trong .Net.
- Vì sao sử dụng XAML?
 - ❖ Tách rời nội dung về đồ họa ra khỏi code
 - ❖ Nâng cao khả năng tùy biến các control trong .Net

- Các cú pháp cơ bản trong XAML

- ❖ Khai báo 1 thành phần:

<[Tên thành phần] [Tên thuộc tính] = [giá trị] >< /[Tên thành phần]>

hoặc <[Tên thành phần] [Tên thuộc tính] = [giá trị]/>

- ❖ Khai báo namespace cho trang XAML

```
<Page x:Name="Root"
      x:Class="RssReader.AppShell"
      xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
      xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
      xmlns:local="using:RssReader"
      xmlns:controls="using:RssReader.Controls"
      xmlns:common="using:RssReader.Common"
      xmlns:viewmodels="using:RssReader.ViewModels"
      xmlns:d="http://schemas.microsoft.com/expression/blend/2008"
      xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"
```

- ❖ Thành phần thuộc tính (Property Element) dùng để khai báo các thuộc tính phức tạp.

```
<controls:PageHeader>
  <controls:PageHeader.HeaderContent>
    <TextBlock Text="Edit your feeds"
      Foreground="{ThemeResource Foreground100CrimsonBrush}"
      Style="{ThemeResource SubtitleTextBlockStyle}" />
  </controls:PageHeader.HeaderContent>
</controls:PageHeader>
```

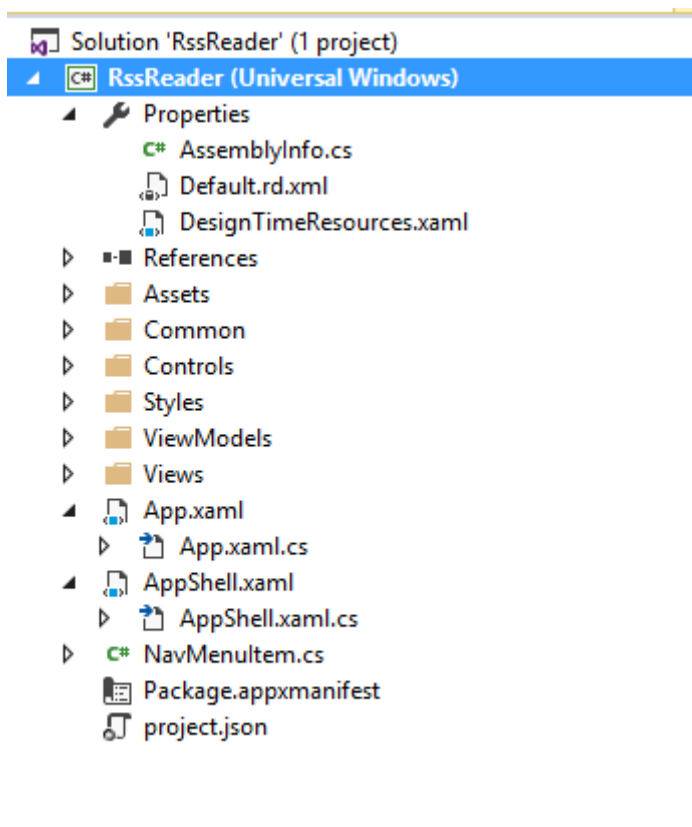
- ❖ Collection Syntax: thuộc tính của một element có thể là tập hợp của nhiều element.

```
<Grid.RowDefinitions>
  <RowDefinition Height="Auto" />
  <RowDefinition Height="Auto" />
  <RowDefinition Height="Auto" />
  <RowDefinition Height="*" />
  <RowDefinition Height="Auto" />
</Grid.RowDefinitions>
```

- ❖ Attached Property Syntax (thuộc tính đính kèm): một số thuộc tính ở element cha được đính kèm vào element con.

```
<Grid>
  <Button Grid.Row="4"
    Margin="24,0" Width="120"
    Content="Add a feed"
    Click="{x:Bind AddFeed}"
    Style="{StaticResource ButtonStyle2}"/>
</Grid>
```

2. 2. 2 Cấu trúc project UWP



AssemblyInfo.cs

Đây là file chứa các thông tin về assembly của ứng dụng như tên assemble, miêu tả assembly, phiên bản, ... Lưu ý là các thông tin tên, miêu tả, phiên bản, ... ở đây là các thông tin dành cho assemble, không phải dành cho ứng dụng. Những thông tin như tên, miêu tả, phiên bản, ... cho ứng dụng được thiết lập ở tập tin **Package.appxmanifest** sẽ được giới thiệu bên dưới.

Default.rd.xml

Đây là file chứa thông tin về các Runtime Directives được sử dụng trong .NET Native.

Thư mục Assets

Đây là thư mục chứa các hình ảnh của ứng dụng mà Visual Studio mặc định tạo ra.

LockScreenLogo. scale-200.png

Đây là hình ảnh logo của ứng dụng sẽ được sử dụng để hiển thị lên màn hình chờ của Windows 10. Bạn chỉ quan tâm đến hình ảnh này khi mà ứng dụng của bạn có sử dụng API liên quan đến màn hình chờ. Kích thước chuẩn của tập tin hình ảnh này là 24×24 pixels.

SplashScreen. scale-200.png

Đây là hình ảnh sẽ được xuất hiện đầu tiên khi ứng dụng của bạn được mở. Có thể tạm dịch hình ảnh này là hình ảnh màn chờ. Hình ảnh này sẽ được đặt giữa màn hình và có kích thước chuẩn là 620×300 pixels.

Square150x150Logo. scale-200.png

Đây là hình ảnh logo vuông nhỏ của ứng dụng. Windows 10 hỗ trợ 4 kích thước logo, logo vuông lớn có kích thước chuẩn là 310×310 px, logo vuông nhỏ có kích thước 150×150 px, logo vuông nhỏ có kích thước 71×71 px và logo chữ nhật ngang có kích thước 310×150 px.

Wide310x150Logo. scale-200.png

Đây là hình ảnh logo chữ nhật ngang của ứng dụng. Kích thước chuẩn của logo này là 310×150 px.

Square44x44Logo. scale-200.png

Đây là hình ảnh logo của ứng dụng được Windows 10 sử dụng để hiển thị ở danh sách ứng dụng. Kích thước chuẩn của logo này là 44×44 px.

Square44x44Logo. targetsizes-24_altform-unplated.png

Tương tự như Square44x44Logo. scale-200. png, đây cũng là hình ảnh logo của ứng dụng được Windows 10 sử dụng để hiển thị ở danh sách ứng dụng. Tuy nhiên đây là tập tin hình ảnh đã được thu nhỏ lại với kích thước 24×24 px.

StoreLogo.png

Đây là hình ảnh logo của ứng dụng sẽ được sử dụng để hiển thị trên Store. Một lời khuyên dành cho các bạn đó là hãy thiết kế logo này thật ấn tượng để thu hút được người dùng khi họ tìm kiếm ứng dụng trên Store. Việc để lại ấn tượng đầu tiên bao giờ cũng tốt hơn cả. Logo này có kích thước chuẩn là 50×50 px.

App.xaml

Đây là một trong những file sẽ được tải đầu tiên khi ứng dụng của bạn được chạy. File này chứa các cấu hình liên quan đến resources cho ứng dụng.

App.xaml.cs

Đây cũng là file sẽ được tải đầu tiên khi ứng dụng của bạn được chạy. Ở đây chứa hàm khởi động **OnLaunched** để bạn có thể viết code xử lý khi ứng dụng của bạn được chạy. Bạn cũng có thể viết những đoạn code xử lý khi ứng dụng của bạn bị ngừng hoạt động (bị suspend) trong hàm **OnSuspending**.

```
private void OnSuspending(object sender, SuspendingEventArgs e)
{
    var deferral = e.SuspendingOperation.GetDeferral();
    deferral.Complete();
}
```

Đoạn code trên có nhiệm vụ lưu trạng thái ứng dụng và ngừng mọi hoạt động nền khi bị ngừng hoạt động

AppShell.xaml

Như có nói ở trên, đây là trang được Visual Studio mặc định tạo ra, và nó có thể được coi là trang chủ của ứng dụng của bạn mặc định. Đây là trang đầu tiên mà

người dùng sẽ nhìn thấy sau hình ảnh màn chờ. Mặc định thì trang này là một trang trống và công việc của bạn là thêm các control để thiết kế giao diện cho trang này.

AppShell.xaml.cs

Đây là file xử lý code logic cho trang AppShell. Tương tự như AppShell. xaml, bên trong file này không có xử lý gì ngoài một hàm khởi tạo (constructor). Bạn có thể viết các đoạn mã xử lý khi mà trang này được tải lên bằng các override hàm

```
private void OnNavigatedToPage(object sender, NavigationEventArgs e)
{
    if (e.Content is Page && e.Content != null)
    {
        var control = (Page)e.Content;
        control.Loaded += Page_Loaded;
    }
}
```

OnNavigatedTo.

Đoạn code trên có nhiệm vụ sau khi chuyển hướng thành công, nó sẽ cài đặt bàn phím tập trung vào trang tải.

project.json

Một điểm mới trong project Windows 10 đó là các Nuget package được sử dụng trong project sẽ được quản lý ở một file với định dạng JSON và file đó có tên là project. json. project. json sẽ chứa khai báo về các package được sử dụng trong project như tên package, phiên bản, ...

Package.appmanifest

Đây là một file vô cùng quan trọng với ứng dụng của bạn. Nó chứa tất cả các thông tin cấu hình cho ứng dụng như tên ứng dụng, mô tả ứng dụng, phiên bản, thông tin package để đẩy ứng dụng lên Store, các quyền mà ứng dụng cần, màu sắc logo, các hình ảnh logo, các hướng màn hình được hỗ trợ ...

2. 2. 3 Các control cơ bản

1/ SplitView:

Là control cho phép bạn tổ chức layout cho ứng dụng của bạn theo dạng side menu hay còn được gọi là hamburger menu, một dạng layout vô cùng phổ biến cho các ứng dụng trên di động. Mặc dù thiết kế giống hamburger menu tuy nhiên SplitView không xây dựng sẵn tính năng hamburger menu. SplitView thực tế cung cấp cho bạn một menu trống cho phép bạn tùy biến theo ý thích của bạn.

Sử dụng SplitView khá đơn giản, nó bao gồm 2 phần chính, một là **Pane** và hai là **Content**.

A/ Pane là thuộc tính cho phép bạn chứa các đoạn mã của menu

```
<SplitView x:Name="RootSplitView" DisplayMode="CompactInline" OpenPaneLength="280" IsTabStop="False">
    <SplitView.Pane>
        <Grid Background="{ThemeResource Background100OffWhiteBrush}">
            <Grid.RowDefinitions>
                <RowDefinition />
                <RowDefinition Height="Auto" />
                <RowDefinition Height="Auto" />
            </Grid.RowDefinitions>
        </Grid>
    </SplitView.Pane>
```

B/ Content là thuộc tính cho phép bạn chứa các đoạn mã hiển thị nội dung của trang.

```
<SplitView.Content>
    <Frame x:Name="AppShellFrame"
        <Frame.ContentTransitions>
            <TransitionCollection>
                <NavigationThemeTransition>
                    <NavigationThemeTransition.DefaultNavigationTransitionInfo>
                        <EntranceNavigationTransitionInfo />
                    </NavigationThemeTransition.DefaultNavigationTransitionInfo>
                </NavigationThemeTransition>
            </TransitionCollection>
        </Frame.ContentTransitions>
    </Frame>
</SplitView.Content>
```

Ngoài hai thuộc tính quan trọng Pane và Content, SplitView còn có một số thuộc tính cũng quan trọng không kém khác:

- **OpenPaneLength**: Thuộc tính này cho phép bạn chỉnh độ rộng của menu khi nó được mở ra.
- **PanePlacement**: Thuộc tính này cho phép bạn tùy biến vị trí đặt menu nằm bên trái hay bên phải màn hình.
- **IsPaneOpen**: Đây là thuộc tính lưu trữ trạng thái đóng/mở của menu. Khi menu được mở, thuộc tính IsPaneOpen sẽ nhận giá trị true và khi đóng, IsPaneOpen nhận giá trị false.
- **DisplayMode**: Thuộc tính này cho phép bạn tùy biến cách mà menu hiển thị. Hiện tại SplitView hỗ trợ 4 kiểu hiển thị menu khác nhau:
 - **Inline**: Khi menu được mở, nó sẽ đẩy nội dung của trang sang một bên. Khi menu được đóng lại, nội dung trang sẽ trở về vị trí cũ.
 - **Overlay**: Khi menu được mở, nó sẽ hiển thị đè lên nội dung của trang. Khi menu được đóng lại, menu sẽ ẩn đi.
 - **Compact Overlay**: Tương tự như Overlay, khi menu được mở, nó sẽ hiển thị đè lên nội dung của trang. Tuy nhiên khi menu được đóng lại, menu vẫn được hiển thị nhưng ở dưới chế độ thu gọn (Compact Mode).
 - **Compact Inline**: Tương tự như Overlay, khi menu được mở, nó sẽ đẩy nội dung của trang sang một bên. Tuy nhiên khi menu được đóng lại, giống với Compact Overlay, menu vẫn được hiển thị nhưng ở dưới chế độ thu gọn (Compact Mode).

2/WebView

Là control kiểm soát để lưu trữ nội dung web trong ứng dụng của bạn.

WebView cung cấp một số hàm API cho chuyển hướng cơ bản: [GoBack](#) , [GoForward](#) , [Stop](#) , [Refresh](#) , [CanGoBack](#) , và [CanGoForward](#). Bạn có thể sử dụng chúng để thêm khả năng duyệt web điển hình cho ứng dụng của bạn.

Để thiết lập các nội dung ban đầu của WebView, thiết lập các Source property trong XAML. Các phân tích cú pháp XAML tự động chuyển đổi chuỗi thành một [Uri](#).

Các Source property có thể được thiết lập trong mã, nhưng thay vì làm như vậy, bạn thường sử dụng một trong những Navigate methods để tải nội dung trong code. Để tải nội dung web, sử dụng Navigate methods với một [Uri](#) có sử dụng http hoặc https scheme.

```
if (ViewModel.CurrentArticle != null)
{
    ArticleWebView.Navigate(ViewModel.CurrentArticle.Link);
}
else
{
    ArticleWebView.NavigateToString(string.Empty);
}
```

3/ Layout controls

Là những controls kế thừa từ Panel, dùng để chứa và sắp xếp các đối tượng con nằm trong nó.

Grid:

- ❖ Sắp xếp các thành phần con theo dòng và cột.
- ❖ Dễ dàng mở rộng nếu cần
- ❖ Phải định nghĩa các dòng và cột trước khi dùng

Khai báo dòng cột cho Grid

```
<Grid.ColumnDefinitions>
    <ColumnDefinition Width="*" />
    <ColumnDefinition Width="Auto" />
</Grid.ColumnDefinitions>
<Grid.RowDefinitions>
    <RowDefinition Height="Auto" />
    <RowDefinition Height="*" />
</Grid.RowDefinitions>
```

ScrollView: Dùng để chứa nội dung

```
<ScrollView x:Name="ContentElement"
    Grid.Row="1"
    Margin="{TemplateBinding BorderThickness}"
    Padding="{TemplateBinding Padding}"
    HorizontalScrollMode="{TemplateBinding ScrollView.HorizontalScrollMode}"
    HorizontalScrollBarVisibility="{TemplateBinding ScrollView.HorizontalScrollBarVisibility}"
    IsTabStop="False"
    IsHorizontalRailEnabled="{TemplateBinding ScrollView.IsHorizontalRailEnabled}"
    IsVerticalRailEnabled="{TemplateBinding ScrollView.IsVerticalRailEnabled}"
    IsDeferredScrollingEnabled="{TemplateBinding ScrollView.IsDeferredScrollingEnabled}"
    VerticalScrollBarVisibility="{TemplateBinding ScrollView.VerticalScrollBarVisibility}"
    VerticalScrollMode="{TemplateBinding ScrollView.VerticalScrollMode}"
    AutomationProperties.AccessibilityView="Raw"
    ZoomMode="Disabled" />
```

4/Text controls

TextBlock: dùng để hiển thị chuỗi

```
<TextBlock x:Name="AddRSSHelp"
    Grid.Row="2"
    Margin="24,24,24,0"
    Text="Nhập URL RSS FEED vào danh sách của bạn."
    TextWrapping="Wrap" VerticalAlignment="Top"
    Foreground="{ThemeResource Foreground100DimGrayBrush}"
    Style="{StaticResource CaptionTextBlockStyle}"/>
```

TextBox: dùng để nhập liệu

```
<TextBox x:Name="LinkTextBox"
    Grid.Row="0" Margin="24,24,24,12"
    Text="{x:Bind ViewModel.CurrentFeed.LinkAsString, Mode=TwoWay}"
    PlaceholderText="RSS feed URL" Header="RSS feed URL"
    KeyDown="LinkTextBox_KeyDown"
    Style="{StaticResource TextBoxStyle1}"/>
```

5/Button: là nút để thực hiện 1 chức năng nào đó

```
<Button x:Name="RemoveFeed"
    Grid.Column="1"
    Content="Remove"
    Click="RemoveFeed_Click"
    Style="{StaticResource ButtonStyle2}" HorizontalAlignment="Right" Width="120" Margin="20,0,0,0"/>
```

6/Label: là nhãn để thể hiện 1 tính chất nào đó

```
<AppBarButton Icon="Add"
    Label="Add feed"
    Click="{x:Bind AddFeed}"
    ToolTipService.ToolTip="Add feed" />
<AppBarButton Icon="Edit"
    Label="Rename"
    ToolTipService.ToolTip="Edit feed"
    Click="{x:Bind EditFeed}"
    IsEnabled="{x:Bind CanEdit, Mode=OneWay}" />
<AppBarButton Icon="Delete"
    Label="Delete"
    Click="{x:Bind DeleteFeed}"
    ToolTipService.ToolTip="Delete feed" />
```

2. 3 TÌM HIỂU VỀ MÔ HÌNH MVVM

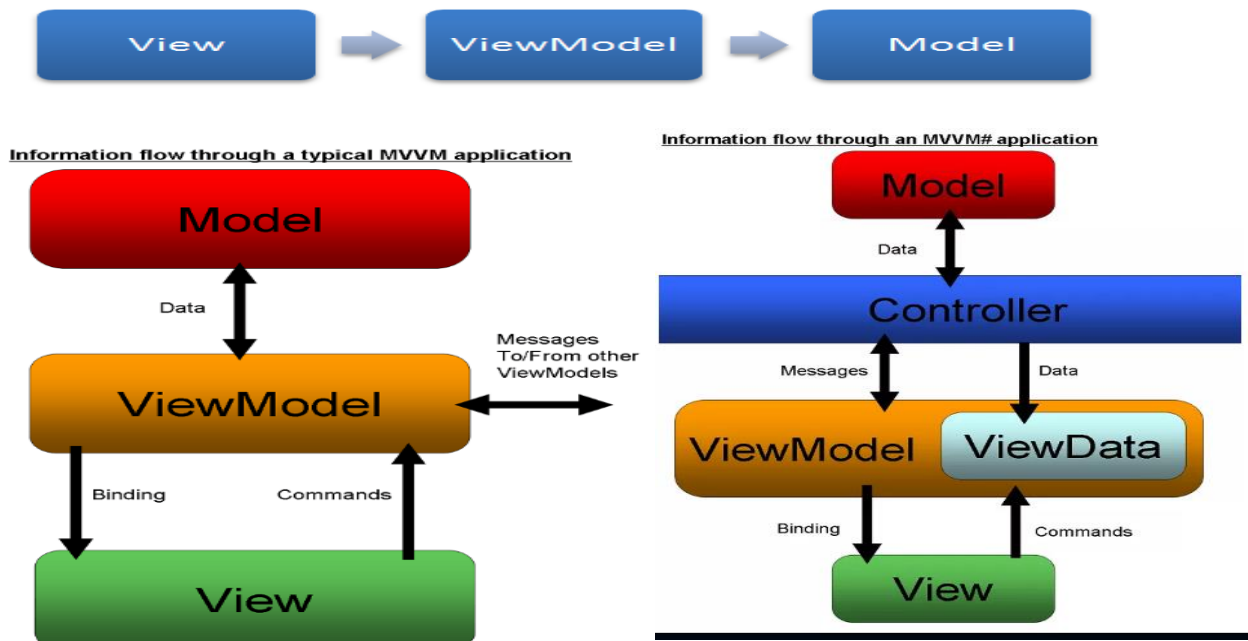
2. 3. 1 Khái niệm về mô hình MVVM

MVVM không phải là framework hay thư viện, api... nó chỉ đơn thuần là hướng dẫn ta định nghĩa cấu trúc của ứng dụng.

- **View:** Tương tự như trong mô hình MVC, View là phần giao diện của ứng dụng để hiển thị dữ liệu và nhận tương tác của người dùng. Một điểm khác biệt so với các ứng dụng truyền thống là View trong mô hình này tích cực hơn. Nó có khả năng thực hiện các hành vi và phản hồi lại người dùng thông qua tính năng binding, command.
- **Model:** Cũng tương tự như trong mô hình MVC. Model là các đối tượng giúp truy xuất và thao tác trên dữ liệu thực sự.

- **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:



2.3.2 Thiết kế UI và khai báo các đối tượng Logic, MVVM trong ứng dụng đọc tin tức.

MVVM

Khai báo, sử dụng các class hỗ trợ MVVM trong ứng dụng đọc tin tức

- ICommand: Giúp gửi-nhận lệnh sự kiện, phương thức giữa View-ViewModel.
- INotifyPropertyChanged: Giúp đồng bộ dữ liệu View-ViewModel.

Trong thư mục Mvvm, ta tạo 03 class sau:

- BindableBase: Implement interface INotifyPropertyChanged.
- Command và Command<T>: Implement interface ICommand.
- BooleanToVisibilityConverter: chuyển đổi các trạng thái UI.
- Extensions: hỗ trợ định dạng, lấy các thuộc tính của 1 đường link.

Models

Model đại diện cho một tập hợp các class, những thứ miêu tả business logic và data. Có cũng định nghĩa business rules cho data những thứ định nghĩa cách data được thay đổi và sử dụng như thế nào.

Ví dụ: FeedDataSource.cs (Ở đây nó xử lý dữ liệu về các link RSS đã lưu trữ, các article đã đưa vào danh sách favorite)

- FeedDataSource.cs // xử lý dữ liệu của bài báo
 - Loading dữ liệu Feed mặc định từ các trang báo
 - using **StorageFile.GetFileFromApplicationUriAsync.**
 - Loading Feed và dữ liệu bài viết từ RSS server
 - using **SyndicationClient.RetrieveFeedAsync.**
 - Loading và lưu Feed, các bài báo yêu thích đến local storage
 - using **StorageFolder.TryGetItemAsync, StorageFolder.CreateFileAsync, DataContractSerializer.**

ViewModels

Tạo một class VM dành cho một trang view cụ thể.

Ví dụ:

- FeedViewModel.cs
- MainViewModel.cs
- ArticleViewModel.cs

Views

- Thiết kế UI theo nhu cầu
- Gán các giá trị Binding
- Khai báo để gọi Command hoặc Event (nút button share)

Ví dụ: Ở đây, xây dựng UI hợp lý cho ứng dụng và resource XAML để tùy chỉnh các thuộc tính như màu sắc, mẫu và các hiệu ứng.

- AddFeedView.xaml //nơi thêm bài báo
- DetailPage.xaml //nơi hiện nội dung chi tiết bài báo
- EditFeedsView.xaml //nơi chỉnh sửa bài báo
- FeedView.xaml //nơi hiện danh mục bài báo
- MasterDetailPage.xaml //nơi hiện danh mục tóm tắt về bài báo (tiêu đề, tóm lược, thời gian, nút yêu thích)

MasterDetailPage.xaml, MasterDetailPage.xaml.cs,

DetailPage.xaml and DetailPage.xaml.cs

- Ở đây code điều chỉnh màn hình hiển thị danh sách các bài báo, các WebView thấy nội dung bài viết, và nút hamberger để các tiêu đề trở về độ rộng của cửa sổ hiện hành.

2. 4 RSS

2. 4. 1 Giới thiệu cấu trúc và định dạng tệp tin RSS (Really Simple Syndication)

- RSS (Really Simple Syndication) Dịch vụ cung cấp thông tin cực kì đơn giản. Dành cho việc phân tán và khai thác nội dung thông tin Web từ xa (ví dụ như các tiêu đề, tin tức). Sử dụng RSS, các nhà cung cấp nội dung Web có thể dễ dàng tạo và phổ biến các nguồn dữ liệu ví dụ như các link tin tức, tiêu đề, và tóm tắt.

Một cách sử dụng nguồn kênh tin RSS được nhiều người ưa thích là công nghệ của RSS cho phép người dùng Internet có thể đăng ký nhận thông tin từ các trang web có cung cấp tính năng RSS (RSS feeds), kết hợp nội dung vào các nhật trình Web (weblogs, hay "blogs"). Blogs là những trang web

mang tính cá nhân và bao gồm các mẫu tin và liên kết ngắn, thường xuyên cập nhật.

- Để có thể dùng công nghệ này, người quản trị trang đó tạo ra hay quản lí một phần mềm chuyên dụng (như một hệ thống quản lí nội dung - content management system-CMS), với định dạng XML mà máy có thể đọc được, có thể biểu diễn các bài tin mới thành một danh sách, với một hoặc hai dòng cho mỗi bài tin và một liên kết đến bài tin đầy đủ đó.
- Định dạng RSS cung cấp nội dung web và tóm lược nội dung web cùng với các liên kết đến phiên bản đầy đủ của nội dung tin đó cùng các siêu-dữ-liệu (meta-data) khác. Thông tin này được cung cấp dưới dạng một tập tin XML. RSS hoạt động theo mô hình "xuất bản và đăng ký" (publish and subscribe). Trong đó, những nội dung được xuất bản từ một nguồn có thể được cung cấp qua nhiều kênh theo phương thức đồng bộ hoá (syndicated).
- Lợi thế nổi bật của hình thức này là thông tin được đẩy đi liên tục từ nguồn tới các đối tượng người đọc. Bản thân nhà cung cấp thông tin cũng không sợ mất bản quyền thông tin vì các tiêu đề trong RSS feed đều liên kết ngược trở lại website xuất phát.
- Trong khi đó, độc giả cũng thực sự hưởng lợi. Họ có thể tiếp cận thông tin theo ý thích xét cả ở ba tiêu chí: ở đâu- khi nào- và như thế nào. Độc giả tiết kiệm được không ít thời gian vì RSS feed lưu trữ những đầu mục thông tin ngắn gọn, liên tục được cập nhật.

2. 4. 2 Những điểm mạnh của RSS

- *RSS cập nhật thông tin:* Ưu việt hơn email, đăng ký một RSS feed, thông tin từ nhà cung cấp sẽ liên tục được cập nhật; bạn không phải vào lại website hoặc đăng ký một RSS feed khác.
- *RSS tiết kiệm về chi phí:* Phí chuyên tải và phân phối giảm đáng kể khi dùng RSS. Băng thông được giải phóng đáng kể với lượng thông tin truyền tải lớn

và không cần nhiều email hay các mã lệnh định dạng cầu kỳ. Bản thân trên một website, bạn chỉ cần một khoảng không khiêm tốn để đặt biểu tượng cho RSS feed.

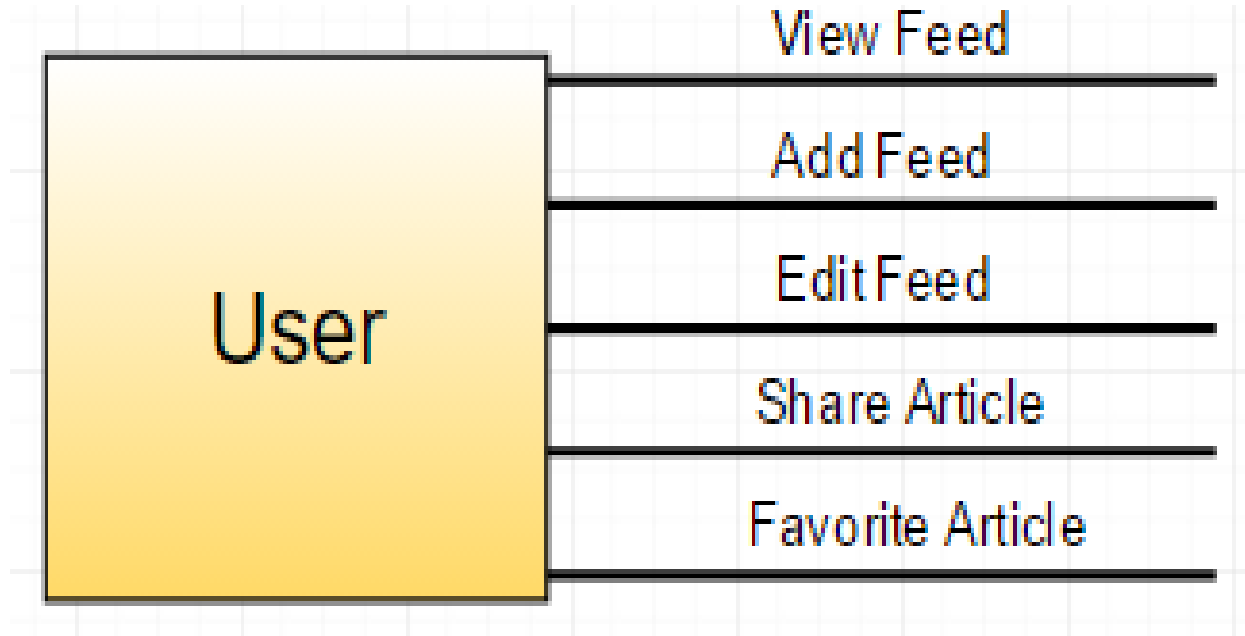
- *RSS tương hỗ với nhiều định dạng:* Ngoài những định dạng thông thường như HTML, ASP... RSS còn có tính mở, hỗ trợ nhiều định dạng ngôn ngữ, nhiều thiết bị đầu cuối (máy tính, điện thoại di động, PDA...).
- *RSS độc lập với email:* Tuy nhiên, các RSS feed vẫn có thể được "thiết kế" để nằm ngay trong email của bạn, bổ sung thêm những tính năng mới cho công cụ nhận thông tin điện tử truyền thống.
- *RSS cho phép tổ chức nội dung:* Thông tin có thể dễ dàng lưu trữ, phân loại và sắp xếp theo từng chủ đề một cách hoàn toàn tự động.
- *Người nhận RSS hoàn toàn chủ động:* Đăng ký và chối bỏ một news feed hoàn toàn phụ thuộc vào bản thân người sử dụng. Điểm này xác lập sự thuận tiện của RSS khi thực tế, không ít người đành phải chấp nhận "chung thân" một khi đăng ký một bản tin thư.
- *RSS đảm bảo quyền cá nhân:* Người đăng ký RSS feed không phải cung cấp email cho nhà cung cấp thông tin mà nhiều khả năng điểm tới cuối cùng là cơ sở dữ liệu của các công ty chuyên về quảng cáo.
- *RSS có thể tái sử dụng:* Thông tin trong RSS hoàn toàn có thể tái sử dụng cho nhiều mục đích: cung cấp cho các kênh thông tin khác (nằm trong feed khác), tích hợp vào các thư viện động hoặc nguyên liệu phục vụ quá trình học tập v. v...
- *RSS an toàn:* Xuất phát từ phương thức truyền tải, RSS không thể mang theo virus hoặc trojan như newsletter hoặc các tệp tin đính kèm. Trong trường hợp các feed có mang các đoạn mã độc hại, người sử dụng có thể dễ dàng "phân vùng" và chủ động loại bỏ thông tin.
- *RSS linh hoạt trong cách cung cấp thông tin:* Người sử dụng tiết kiệm được thời gian, thông tin được cung cấp ở dạng tóm tắt (thường chỉ là các tiêu đề-

title). Chính vì vậy, hàm lượng thông tin truyền tải lớn trong mỗi feed (loại bỏ các loại hình quảng cáo pop-up, định dạng mang tính trang trí...). Người sử dụng chỉ click vào những siêu liên kết để đến với nội dung chi tiết mà mình thực sự quan tâm. Ngay khi đã được gửi đi (đồng bộ hoá dữ liệu), các nhà cung cấp vẫn có thể chỉnh sửa, cập nhật thông tin trong các RSS feed.

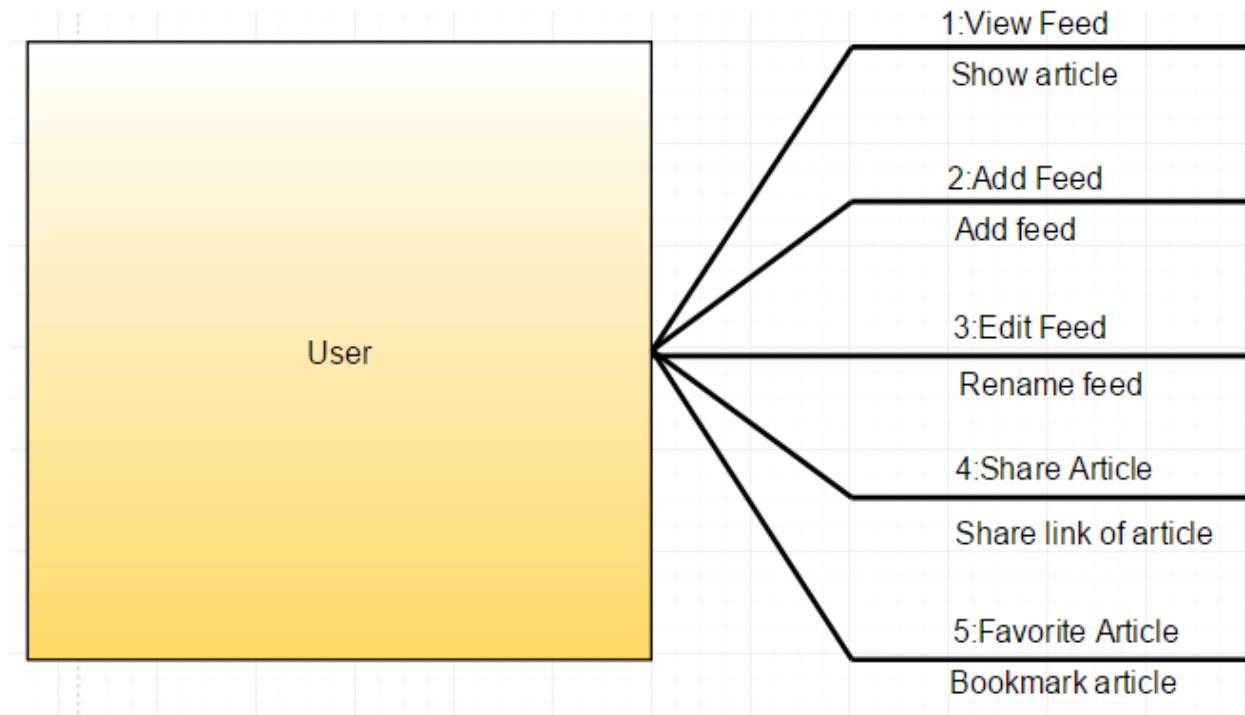
CHƯƠNG 3: PHÂN TÍCH HỆ THỐNG

3. 1 PHÂN TÍCH HỆ THỐNG VỀ CHỨC NĂNG

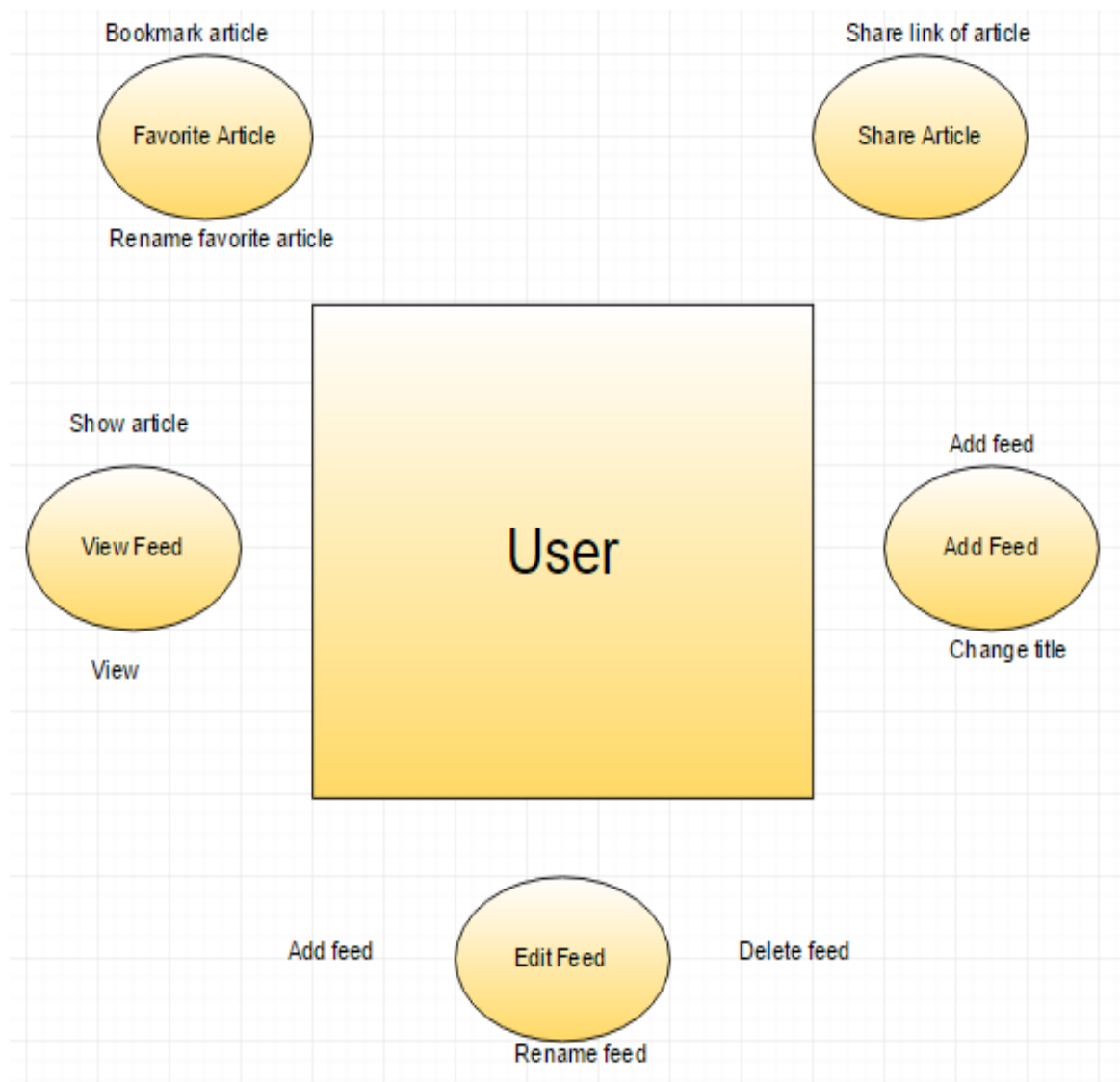
3. 1. 1 Lập biểu đồ luồng dữ liệu mức khung cảnh



3. 1. 2 Lập biểu đồ luồng dữ liệu mức đỉnh



3. 1. 3 Lập biểu đồ luồng dữ liệu mức dưới đỉnh



CHƯƠNG 4: THIẾT KẾ HỆ THỐNG VÀ GIAO DIỆN

4.1 THIẾT KẾ SƠ ĐỒ LỚP

Các tác nhân:

- Người sử dụng.
- Trang web cung cấp RSS.

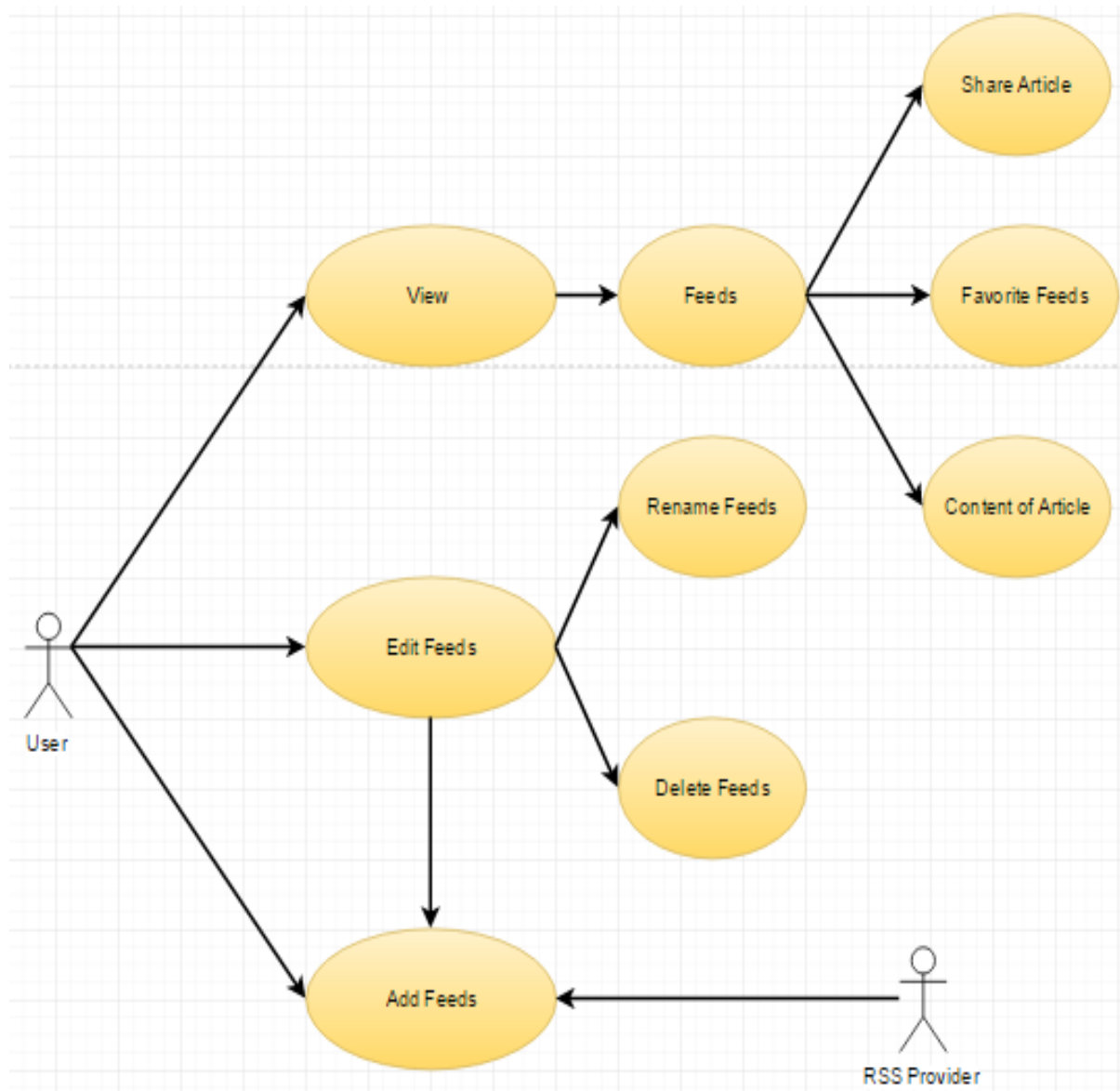
Các Use Case:

- View Feed & Articles: Đọc các mục RSS và các bài viết trong mục đó.
- Edit Feed: Đổi tên danh mục RSS, xóa tên danh mục RSS.
- Add Feed: Thêm mới 1 mục RSS.

Đặc tả hệ thống

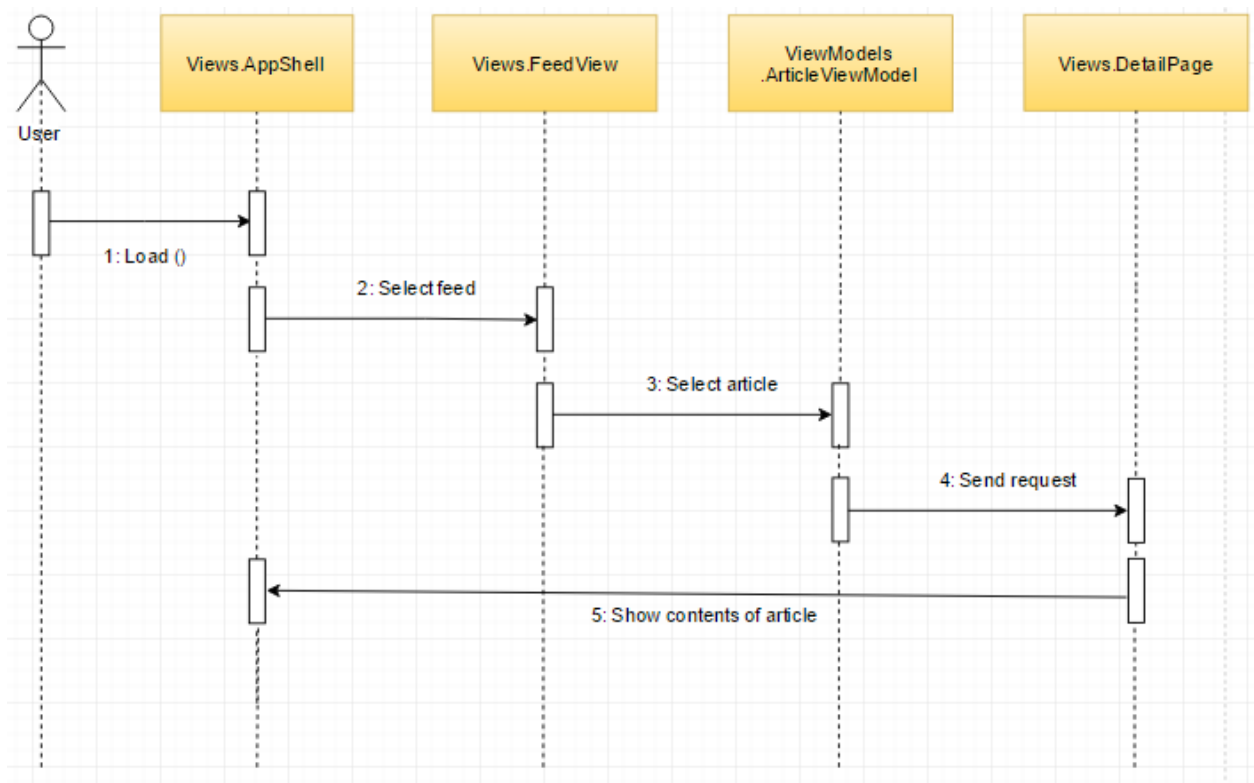
- Người dùng chọn các tác vụ của chương trình để thực hiện.
- Nếu trong cơ sở dữ liệu đã có các trang tin chương trình sẽ load ra các trang để lấy tin từ các trang đó, người dùng chọn một trong số các trang tin có sẵn sau khi chọn chương trình gọi tới browser để thực hiện view trang cần xem.
- Nếu trong cơ sở dữ liệu không có tin, chọn Add rồi nhập đường dẫn đến trang cần đọc dạng RSS, add xong trang tin sẽ có trong cơ sở dữ liệu mỗi khi người dùng chương trình.
- Để xóa trang tin trong cơ sở dữ liệu người dùng chọn Edit sau đó chọn vào trang tin cần xóa, sau đó nhấn vào Delete để xóa.

4. 1. 1 Use Case

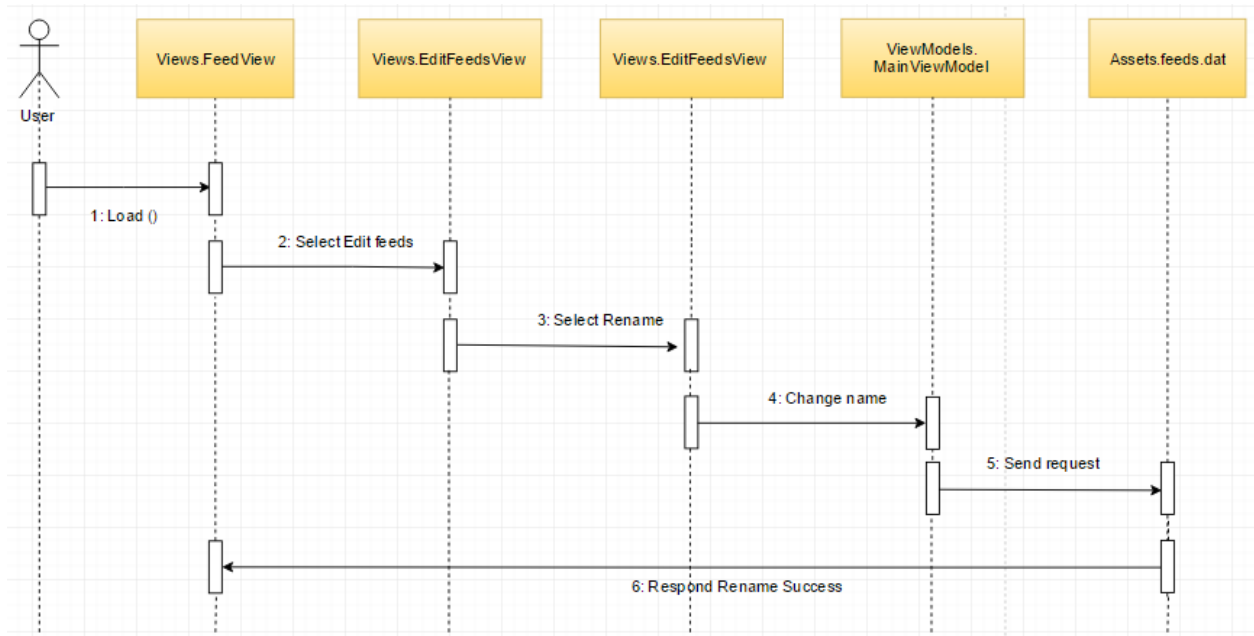


4. 1. 2 Sequence diagram

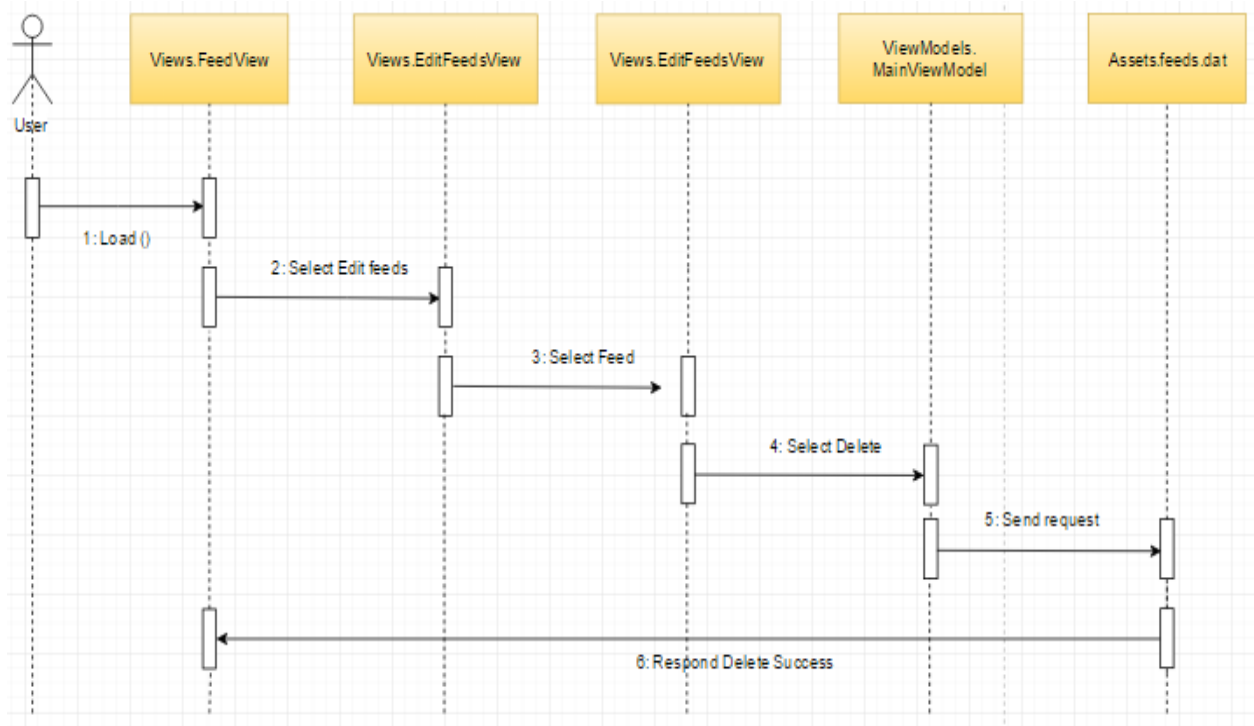
4. 1. 2. 1 Xem tin tức



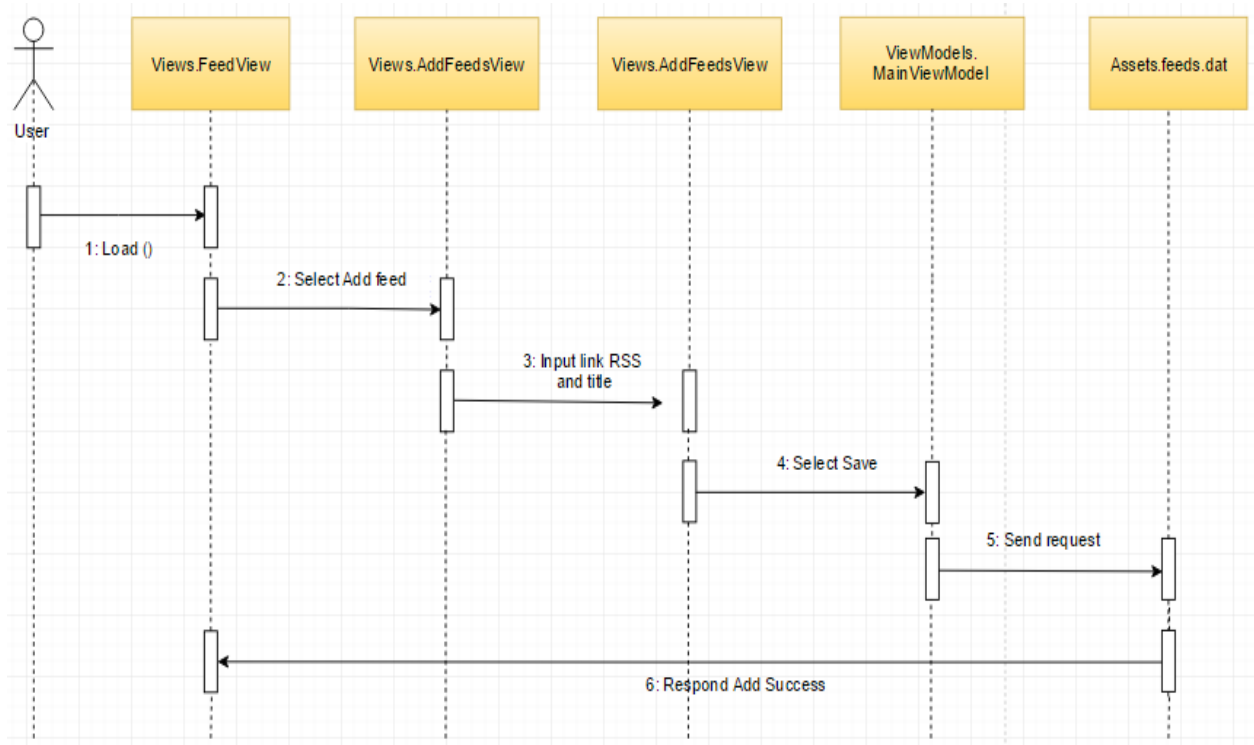
4. 1. 2. 2 Đổi tên trang tin



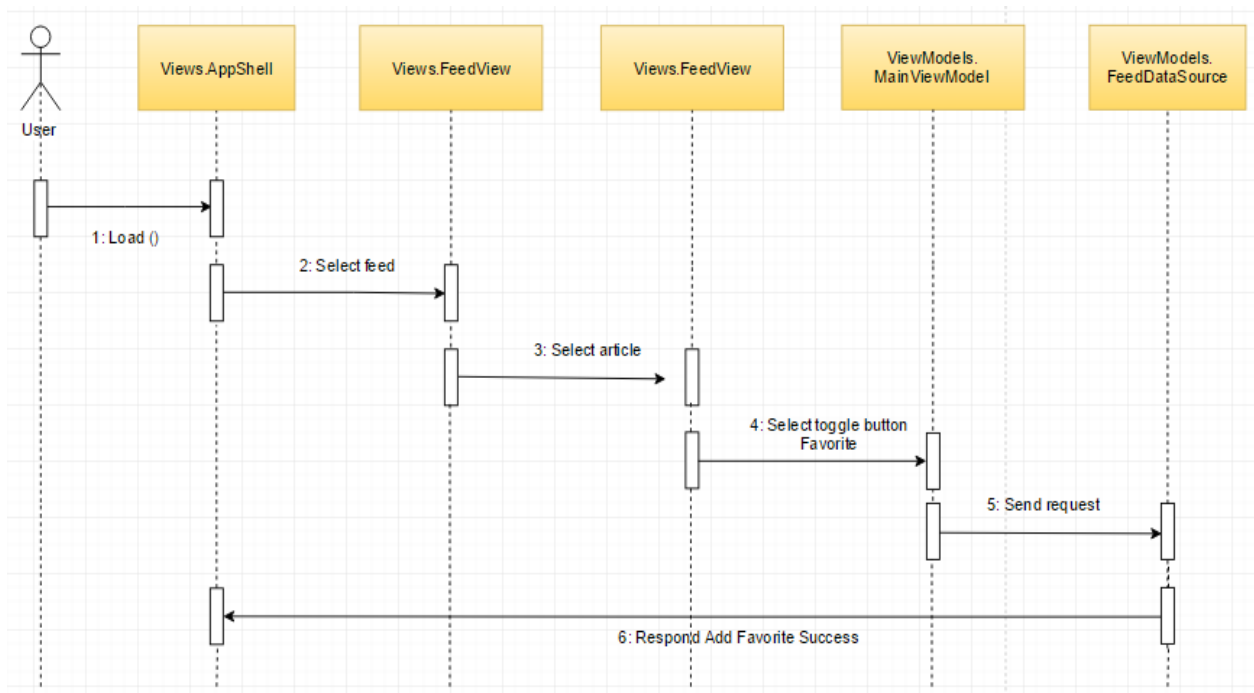
4. 1. 2. 3 Xóa trang tin



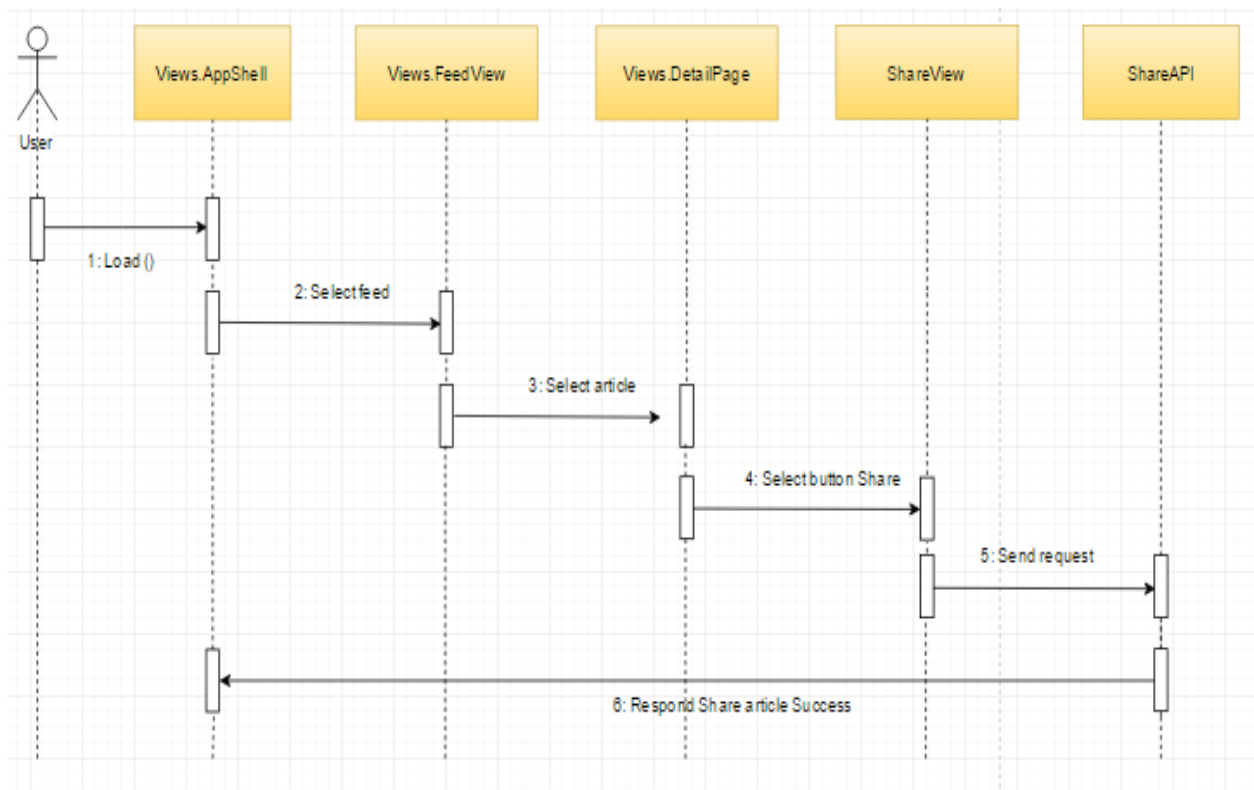
4. 1. 2. 4 Thêm trang tin



4. 1. 2. 5 Thêm tin tức yêu thích

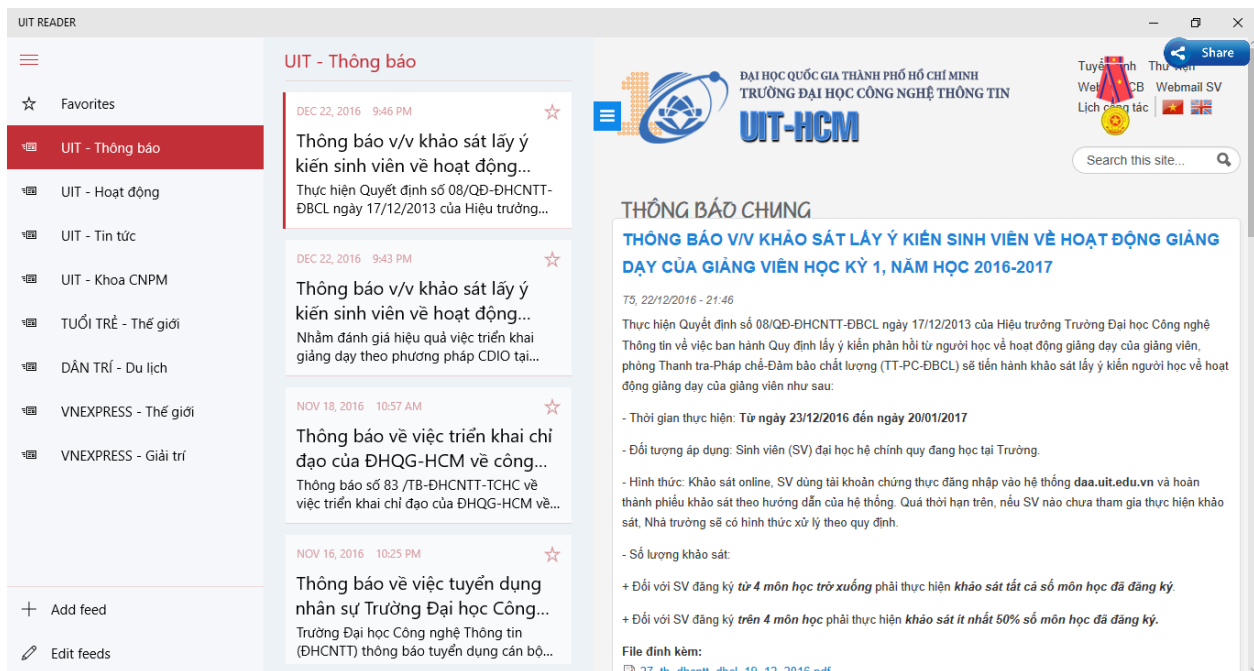


4. 1. 2. 5 Share tin tức yêu thích

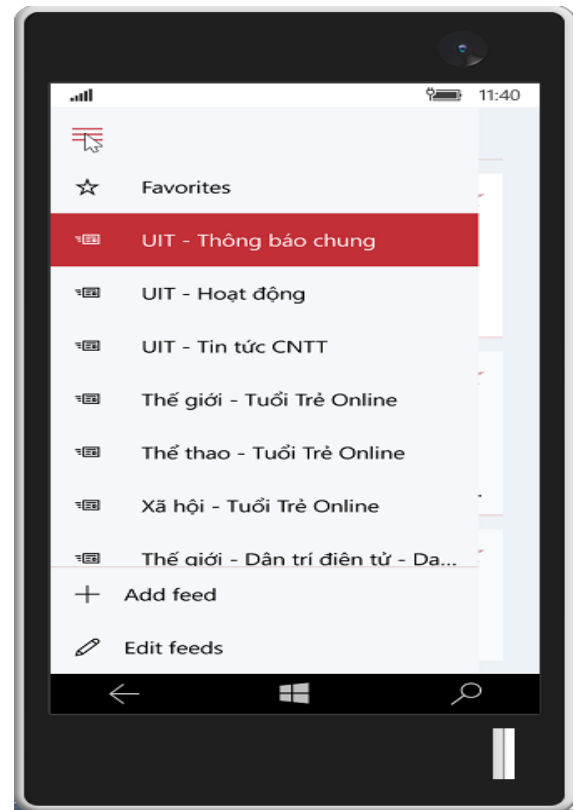
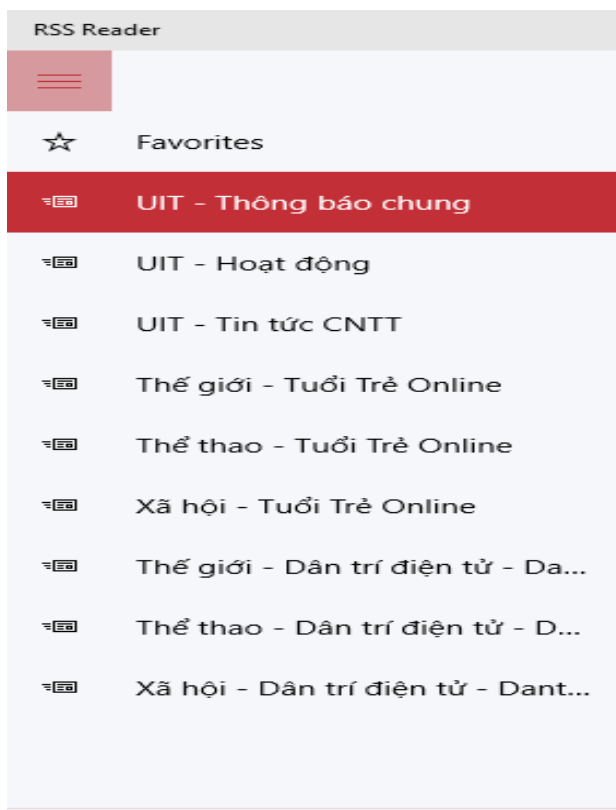


4. 2 GIAO DIỆN ỨNG DỤNG

❖ Giao diện màn hình chính



❖ Giao diện danh mục tin tức



❖ Giao diện list hạng mục trong từng danh mục bài báo



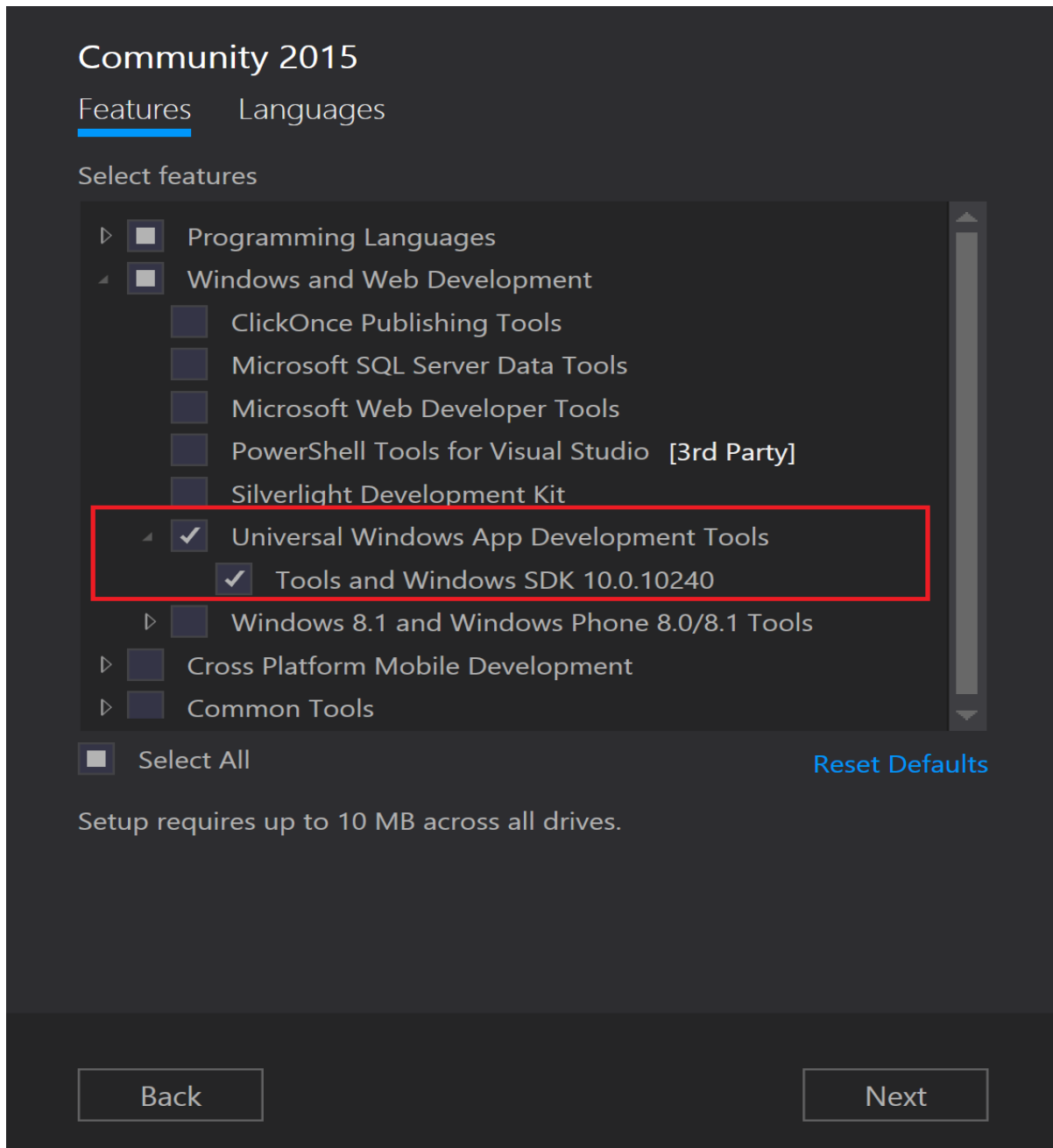
❖ Giao diện nội dung bài báo



CHƯƠNG 5: TỔNG KẾT

5.1 CÀI ĐẶT ỨNG DỤNG

- Bước 1: Cài đặt Visual Studio 2015, Chọn Universal Windows App



- Bước 2: Update Visual và download máy ảo
- Bước 3: Open project và build project.

5.2 CÔNG CỤ PHÁT TRIỂN ỨNG DỤNG

- Visual Studio 2015
 - UNIVERSAL WINDOWS PLATFORMS
 - C#
 - XAML

5.3 KẾT QUẢ ĐẠT ĐƯỢC VÀ HẠN CHẾ

Kết quả đạt được của đề tài

Sau quá trình nghiên cứu, tìm hiểu, báo cáo chuyên đề tốt nghiệp mang lại một số kết quả có ý nghĩa như sau:

- Khái quát được mục đích của đề tài: xây dựng ứng dụng đọc tin tức
 - ❖ Đã tìm hiểu được về công nghệ cũng như các kỹ thuật làm ứng dụng đọc tin tức của một UWP Project.
 - ❖ Đã xây dựng được một chương trình lấy tin tự động có định dạng RSS, giúp người dùng có thể tự lựa chọn, tổng hợp các trang báo yêu thích, có thể chia sẻ bài báo hay qua mạng xã hội, tin nhắn hoặc mail...

Một số hạn chế của đề tài

- ❖ Chưa xây dựng được một số chức năng của chương trình lấy tin tức như: lưu tin khi người dùng muốn đọc ở chế độ offline.
- ❖ Chưa xây dựng được giao diện thân thiện với người dùng.

5.4 HƯỚNG PHÁT TRIỂN

- Xây dựng server cho phép đồng bộ tài khoản giữa desktop và điện thoại, có kho RSS lưu trữ sẵn cho người dùng chọn...

- Xây dựng thêm các chức năng khác như: xem giá vàng, thông tin chứng khoán, dự báo thời tiết cho các thiết bị di động trên hệ điều hành Window Phone.
- Nghiên cứu và tìm hiểu sâu các kiến thức liên quan tới nền tảng Window hợp nhất UNIVERSAL WINDOWS PLATFORMS để có thể xây dựng thêm 1 số ứng dụng khác như là: chat, gửi tin nhắn, chia sẻ video, file, nghe nhạc, chơi game, mail.

5. 5 PHÂN CÔNG CÔNG VIỆC

Tên	Lê Minh Trọng	Phạm Anh Đại	Nguyễn Thanh Tân
Công việc	<ul style="list-style-type: none"> • Tìm hiểu về các control cơ bản UWP • Tìm hiểu về RSS • Phân tích và thiết kế hệ thống • Code phần ViewModel • Tổng hợp và viết báo cáo hoàn chỉnh 	<ul style="list-style-type: none"> • Tìm hiểu về XAML • Phân tích và thiết kế hệ thống • Code phần ViewModel, giao diện ứng dụng • Làm slide báo cáo • Quay video và chỉnh sửa video 	<ul style="list-style-type: none"> • Khảo sát bài toán • Tìm hiểu về UWP, cấu trúc UWP • Phân tích và thiết kế hệ thống • Code phần giao diện ứng dụng • Làm slide giới thiệu về UWP • Lên sườn và viết báo cáo

TÀI LIỆU THAM KHẢO

[1] Lương của lập trình viên di động http://www.payscale.com/research/US/Job=Mobile_Applications_Developer/Salary
<https://developer.microsoft.com/en-us/windows/develop>
<https://channel9.msdn.com/Series/Windows-10-development-for-absolute-beginners>
http://www.tutorialspoint.com/windows10_development/windows10_development_uwp.htm
<https://comentsys.wordpress.com/2015/05/31/windows-10-universal-windows-platform-tutorials/>
<http://www.bravohe.com/2016/02/mvvm-so-luoc-mo-hinh-lap-trinh-model-view-viewmodel.html>
<https://yinyangit.wordpress.com/2011/12/02/co-ban-ve-mvvm-model-view-viewmodel-pattern/>
[https://vi.wikipedia.org/wiki/RSS_\(%C4%91%E1%BB%8Bnh_d%E1%BA%A1ng_t%E1%BA%ADp_tin\)](https://vi.wikipedia.org/wiki/RSS_(%C4%91%E1%BB%8Bnh_d%E1%BA%A1ng_t%E1%BA%ADp_tin))