ĐẠI HỌC QUỐC GIA HÀ NỘI

TRƯỜNG ĐẠI HỌC CÔNG NGHỆ

TRẦN HỮU DỰ

CÁC KỸ THUẬT PHÂN TÍCH VÀ LẤY TIN TỰ ĐỘNG TỪ WEBSITE

LUẬN VĂN THẠC SĨ CÔNG NGHỆ THÔNG TIN

Hà Nội -2016

ĐẠI HỌC QUỐC GIA HÀ NỘI

TRƯỜNG ĐẠI HỌC CÔNG NGHỆ

TRẦN HỮU DỰ

CÁC KỸ THUẬT PHÂN TÍCH VÀ LẤY TIN TỰ ĐỘNG TỪ WEBSITE

Ngành: Công nghệ Thông tin

Chuyên ngành: Kỹ Thuật Phần Mềm

Mã số: 60 48 0103

LUẬN VĂN THẠC SĨ CÔNG NGHỆ THÔNG TIN

NGƯỜI HƯỚNG DẪN KHOA HỌC: TS. TÔ VĂN KHÁNH

Hà Nội -2016

LỜI CAM ĐOAN

Tôi xin cam đoan, đây là công trình nghiên cứu của bản thân, các số liệu các đoạn mã chương trình của ứng dụng, các kết quả trình bày trong luận văn là trung thực và chưa từng được ai công bố trong bất kỳ công trình luận văn nào trước đây.

Tác giả luận văn

Trần Hữu Dự

LỜI CẢM ƠN

Trước tiên tôi xin chân thành cảm ơn đến thầy giáo TS. Tô Văn Khánh - người đã tận tình chỉ bảo và giúp đỡ tôi trong suốt quá trình thực hiện đề tài luận văn thạc sĩ cho đến khi hoàn thành đề tài.

Tôi xin bày tỏ lòng biết ơn chân thành tới các thầy cô giáo khoa Công nghệ thông tin, trường Đại học Công nghệ, Đại học Quốc Gia Hà Nội - nơi tôi đã theo học trong những năm qua. Các thầy cô đã dạy và cung cấp những kiến thức quý báu, tạo điều kiện tốt nhất cho tôi trong suốt quá trình học tập và nghiên cứu tại trường.

Sau cùng tôi xin chân thành cảm ơn những người thân trong gia đình, cảm ơn bạn bè cùng khóa, đồng nghiệp trong cơ quan đã giúp đỡ tôi trong quá trình học tập và nghiên cứu thực hiện luận văn này.

Tuy nhiên, trong quá trình làm luận văn tôi cũng đã rất cố gắng nghiên cứu, tìm hiểu các vấn đề liên quan song luận văn vẫn chưa thực sự được hoàn chỉnh, vẫn còn những thiếu sót nhất định. Tôi rất mong nhận được những ý kiến đánh giá, góp ý của các thầy cô giáo, các bạn để luận văn được hoàn thiện hơn.

Hà nội, tháng 11 năm 2016

Học viên

Trần Hữu Dự

MỤC LỤC

BẢNG CÁC KÝ HIỆU VÀ CHỮ VIẾT TẮT

STT	Thuật ngữ viết tắt	Thuật ngữ đầy đủ
1	API	Application Programming Interface
2	CSS	Cascading Style Sheets
3	DOM	Document Object Model
4	HTML	HyperText Markup Language
5	JAXP	Java API for XML Processing
6	JDK	Java Deverlopment Kit
7	LISP	LISt Processing
8	MathML	Math Markup Language
9	RDF	Resource Description Framework
10	RSS	Really Simple Syndication
11	SAX	Simple API for XML
12	SGML	Standard Generalized Markup Language
13	StAX	Streaming API cho XML
14	SVG	Scalable Vector Graphics
15	URL	Uniform Resource Locator
16	WHATWG	Web Hypertext Application Technology Working Group
17	XHTML	Extensible HyperText Markup Language
18	XML	Extensible Markup Language
19	CSS	Cross Site Scripting

DANH MỤC HÌNH VẼ

DANH MỤC HÌNH BẢNG BIỂU

MỞ ĐẦU

Trong thực tế, tất cả mọi người, tất cả các ngành nghề, mọi lĩnh vực đều có nhu cầu thu thập và cập nhật thông tin, tin tức nhanh nhất có thể. Nguồn cung cấp các thông tin đó đến từ nhiều nguồn các nhau, tại nhiều địa điểm khác nhau... trong đó việc cập nhật tin tức từ các website cũng là một trong số những nguồn cung cấp thông tin cần thiết và quan trọng. Việc cập nhật thông tin này sẽ rất khó khăn và tốn kém về mặt thời gian, công sức... nếu chúng ta phải thường trực thường xuyên trên máy tính hoặc điện thoại liên tục truy cập các trang mạng để đọc tin mới từ các website. Do đó đã thúc đẩy nghiên cứu các kỹ thuật phân tích và lấy tin tự động từ các website. Đây là một ứng dụng được xây dựng để chạy trên thiết bị điện thoại di động, yêu cầu thiết bị di động này có kết nối với mạng Internet thông qua wifi hoặc 3G và các website phải hỗ trợ RSS. Tuy nhiên trong thực tế không phải tất cả người dùng đều có thiết bị di động thông minh, các địa điểm đều có Internet Wifi hoặc được phủ sóng 3G, và không phải tất các website đều có hỗ trợ RSS nên việc cập nhật tin tức mới ngay lập tức cũng có những hạn chế nhất định.

Với những thuận lợi và khó khăn trên khi thiết kế, triển khai hệ thống phân tích và lấy tin tự động từ các website đã đặt ra nhiều hướng nghiên cứu để hoàn chỉnh ứng dụng. Trong đó các nhà khoa học đang quan tâm mạnh mẽ hướng nghiên cứu chính là phân tích được cấu trúc của một website, nhận biết được website có hỗ trợ RSS và tiến hành lấy dữ liệu cơ bản về tin tức mới được cập nhật thông qua file RSS mà website cung cấp. Vấn đề đọc và lấy tin tức mới trên website là vấn đề quyết định xem ứng dụng có phát hiện ra tin tức mới và thông báo Notification cho người dùng một cách kịp thời nhất. Chính vì vậy nghiên cứu các kỹ thuật phân tích và lấy tin tự động từ các website tới người dùng một cách kịp thời có ý nghĩa lý luận và thực tiễn.

Mục tiêu chính của luận văn là nghiên cứu tổng thể các kỹ thuật phân tích và đọc tin tự động từ website. Mục đích nhằm nắm bắt được các công nghệ và các kỹ thuật phân tích và lấy tin tự động từ file RSS của website để xây dựng ứng dụng chạy trên thiết bị di động thông báo cho người dùng biết có tin mới một cách tự động và trong thời gian nhanh nhất có thể.

Bố cục của luận văn tuân theo mẫu của trường Đại Học Công Nghệ-ĐHQGHN. Luận văn gồm có 4 chương chính ngoài ra còn có phần mở đầu, kết luận và tài liệu tham khảo. Trong đó chương 1 sẽ nêu lên nhu cầu cập nhật tin tức của người dùng và ứng dụng của việc đọc tin tự động.

Chương 2 sẽ giới thiệu về cấu trúc và kỹ thuật làm việc với XML, RSS 2.0, các phương pháp kỹ thuật Interface DOM, Interface SAXđể phân tích RSS 2.0.

Chương 3 sẽ trình bày chi tiết hơn về các hàm được cung cấp dùng trong kỹ thuật phân tích website với Tool Jsoup.

Chương 4 là phần thực nghiệm xây dựng ứng dụng tự động cập nhật thông báo về nội dung mới từ các website.

Kết luận tóm lược lại các kết quả đã nghiên cứu được về các kỹ thuật phân tích XML đọc tin tự đông.

Tài liệu tham khảo: các nguồn tài liệu đã sử dụng thực hiện luận văn.

Xin trân trọng cảm ơn

Tác giả: Trần Hữu Dự

CHƯƠNG 1 GIỚI THIỆU

1.1. Nhu cầu cập nhật tin tức của người dùng

Hầu hết mọi người quan tâm đến nhiều trang website có thay đổi nào trên một lịch trình không thể đoán trước nội dung ví dụ về các trang web như là các trang web tin tức, các trang cộng đồng và thông tin tổ chức tôn giáo, các trang thông tin sản phẩm, các trang web y tế, và weblog...Liên tiếp kiểm tra từng trang web để xem nếu có bất kỳ nội dung mới có thể rất tẻ nhạt.

Trước đây, Email thông báo về những thay đổi là một giải pháp sớm để vấn đề này. Thật không may, khi chúng ta nhận được thông báo email từ nhiều trang một cách vô tổ chức, với số lượng lớn, và thường nhầm lẫn với thư rác.

Sau đó RSS ra đời là một cách tốt hơn để cung cấp, thông báo về nội dung mới và thay đổi.Thông báo thay đổi đến từ nhiều trang web khác nhau được xử lý một cách dễ dàng, kết quả được trình bày một cách có tổ chức, có cấu trúc tốt và khác biệt từ email.

1.2. Ứng dụng của đọc tin tự động

Ứng dụng trong thương mại: Ứng dụng đọc tin tự động là một ứng dụng trực tuyến tiết kiệm thời gian tốt nhất và rất dễ sử dụng. Với công nghệ đọc và phân tích cú pháp của RSS là một cách để cho người dùng lướt web có thể có được những tin tức mới nhất hoặc "tóm" được những bản hợp đồng du lịch trọn gói giá rẻ nhưng chất lượng, mua được những tấm vé xem hòa nhạc mong muốn hoặc mua được những mặt hàng thiết yếu tại các siêu thị lớn cùng với chương trình khuyến mãi hấp dẫn và gần như là tất cả mọi thứ mà mọi người thường mua thông qua mạng Internet.

Thay vì phải ngồi gõ địa chỉ của các trang website mỗi khi muốn xem có tin tức gì mới hoặc muốn xem những bài mới trên blog của những người thân chúng ta chỉ cần cài đặt ứng dụng đọc tin tự động một lần.

Một blogger truyền thông nổi tiếng Jeff Jarvis [9] đã từng phát biểu: *"Tôi không sử dụng bookmark. Nếu một trang web không có RSS, tôi luôn có cảm giác khó chịu"*.

Charlene Li, chuyên gia phân tích của Forrester, nhận xét rằng RSS được ứng dụng rất nhiều trong các tình huống khác nhau của đời sống. Charlene phát biểu [9]: *"Hiện tại, tôi đang tìm kiếm những tấm vé xem chương trình The Jersey*

Boys. Nó đã được bán hết.Tuy nhiên, nếu may mắn, thỉnh thoảng bạn cũng sẽ kiếm được một cái trên Craigslist".

Ứng dụng trong y tế: Y học thế giới trong những thập niên trở lại đây phát triển nhanh chóng với tốc độ bằng cả hàng thế kỷ trước, rất nhiều căn bệnh trước đây được coi là bệnh nan y không có phương thức nào cứu chữa thì ngày nay đã tìm được phương pháp điều trị triệt để. Với biết bao công trình y học được công bố, bao nhiêu giải thưởng Nobel đã được trao tặng cho các nhà khoa học.Tuy nhiên song song với sự tiến bộ của ý học thì ngày càng có nhiều loại bệnh phức tạp hơn, số lượng bệnh nhân mắc bệnh cũng ngày một gia tăng.Vì thế mà nhu cầu cập nhật thông tin y học về các loại bệnh mới, các phương pháp điều trị mới trở thành mối quan tâm hàng đầu của nhiều bệnh nhân. Ví dụ bệnh nhân bị mắc tiểu đường hàng ngày họ cần phải luyện tập, ăn kiêng, thường xuyên kiểm tra lượng đường trong máu của mình và luôn tuân thủ lời khuyên của bác sĩ. Nếu như họ sử dụng ứng dụng để thường xuyên cập nhật tin tức từ 1 website về các biểu hiện tiến triển của bệnh, lời khuyên từ bác sĩ, thông tin về các loại thuốc mới điều trị tiểu đường, hay các loại thực phẩm tốt đối với họ thì rất là hữu ích.

Ứng dụng trong trường học: Các trường đại học, cao đẳngthường duy trì một kênh liên lạc với sinh viên và giảng viên bởi 1 website. Trên website thường xuyên cập nhật các thông tin liên quan tới sinh viên ví dụ như thông tin môn học, thông tin cảnh báo học vụ, thông tin thi cử, thông tin điểm số.... Khi sinh viên có mối quan tâm tới một trong các thông tin trên thì hoàn toàn có thể sử dụng ứng dụng như công cụ theo dõi cập nhật thông tin cho mình một cách hiệu quả và kịp thời nhất.

Có thể thấy rằng ý nghĩa trong thực tiễn của RSS nói chung và Ứng dụng đọc tin tự động nói riêng là rất có ý nghĩa, quan trọng và cần thiết.

CHƯƠNG 2
CÁC KỸ THUẬT PHÂN TÍCH VÀ LẤY TIN TỰ ĐỘNG

2.1. Giới thiệungôn ngữ mở rộng đánh dấu XML

Khái niệm XML: XML (viết tắt của từ tiếng Anh Extensible Markup Language, "ngôn ngữ đánh dấu mở rộng") [10] là ngôn ngữ đánh dấu với mục đích chung do W3C đề nghị, để tạo ra các ngôn ngữ đánh dấu khác nhau. Đây là một tập con đơn giản của SGML, có khả năng mô tả nhiều loại dữ liệu khác nhau, đặc biệt là các hệ thống được kết nối với Internet. Các ngôn ngữ dựa trên XML (thí dụ như RDF, RSS, MathML, XHTML, SVG và cXML) được định nghĩa theo cách thông thường, cho phép các chương trình sửa đổi và kiểm tra hợp lệ bằng ngôn ngữ này mà không cần có hiểu biết trước về hình thức của chúng [10].

Đặc điểm của XML: XML cung cấp một phương tiện dùng văn bản (text) để mô tả thông tin, áp dụng một cấu trúc kiểu cây cho thông tin. Xét về mặt hình thức văn bản, mọi thông tin đều thể hiện dưới dạng text, chen giữa là các thẻ đánh dấu (markup) với nhiệm vụ ký hiệu sự phân chia thông tin thành một cấu trúc có thứ bậc của các dữ liệu ký tự, các phần tử dùng để chứa dữ liệu, và các thuộc tính của các phần tử. Về mặt logic XML tương tự với các biểu thức S (S-expression) của ngôn ngữ lập trình LISP ở chỗ chúng đều mô tả các cấu trúc dạng cây trong đó mỗi nút có thể có một danh sách tính chất của riêng mình.

Đơn vị cơ sở của XML là các ký tự theo định nghĩa của Universal Character Set (bộký tự toàn cầu). Các ký tự được kết hợp theo các tổ hợp chuỗi hợp lệ để tạo thành tài liệu XML. Tài liệu này gồm một hoặc nhiều thực thể, mỗi thực thể thường là một phần nào đó của các ký tự thuộc tài liệu được mã hóa dưới dạng một chuỗi các bit và lưu trữ trong một tệp văn bản (text file).

Tài liệu XML được soạn thảo tạo, bảo trì môt cách đơn giản, thuận tiện bởi sự phổ biến của các phần mềm soạn thảo văn. Trước khi xuất hiện XML có rất ít các ngôn ngữ mô tả dữ liệu với các đặc điểm đa năng, thân thiện với giao thức Internet, dễ đọc và dễ tạo. Thực tế, đa số các định dạng trao đổi dữ liệu thời đó đều có tính chuyên dụng, có tính độc quyền, và có định dạng nhị phân (chuỗi bit thay vì chuỗi ký tự) khó dùng chung giữa các ứng dụng phần mềm khác nhau hay giữa các hệ nền (platform) khác nhau .Việc tạo và bảo trì trên các trình soạn thảo thông dụng lại càng khó khăn.

Bằng cách cho phép định tên dữ liệu, cấu trúc thứ bậc, ý nghĩa của các phần tử,các thuộc tính có tính chất mở, có thể được định nghĩa bởi một giản đồ tùy biến được, XML cung cấp một cơ sở cú pháp cho việc tạo lập các ngôn ngữ đánh dấu dựa trên XML theo yêu cầu. Cú pháp chung của các ngôn ngữ đó là cố định - các tài liệu phải tuân theo quy tắc chung của XML, bảo đảm rằng tất cả các phần mềm hiểu XML ít ra cũng phải có khả năng đọc (Phân tích cú pháp - parse) và hiểu bố cục tương đối của thông tin trong các tài liệu đó. Giản đồ chỉ bổ sung một tập các ràng buộc cho các quy tắc cú pháp. Các giản đồ thường hạn chế tên, thuộc tính, các cấu trúc thứ bậc được phép của phần tử, ví dụ chỉ cho phép một phần tử tên 'ngày sinh' chứa một phần tử tên 'ngày' và một phần tử có tên 'tháng', nỗi phần tử phải chứa đúng một ký tự. Đây là điểm khác biệt giữa XML và HTML. HTML có một bộ các phần thử và thuộc tính không mềm dẻo, chỉ có một tác dụng và nói chung là không thể dùng cho mục đích khác.

XML không hạn chế về việc nó được sử dụng như thế nào. Mặc dù XML về cơ bản là dạng text, các phần mềm với chức năng trừu tượng hóa nó thành các định dạng khác giàu thông tin hơn đã nhanh chóng xuất hiện, quá trình trình trừu tượng hóa này được thực hiện chủ yếu qua việc sử dụng các giản đồ định hướng kiểu dữ liệu (datatype-oriented schema) và khuôn mẫu lập trình hướng đối tượng (mà trong đó, mỗi tài liệu XML được thao tác như là một đối tượng). Những phần mềm như vậy có thể coi XML như là dạng text đã được tuần tự hóa chỉ khi nó cần truyền dữ liệu qua mạng.

Khai báo định dạng file: Dòng đầu tiên trong file XML phải là dòng khai báo XML phần tùy chọn này dùng để nhận dạng đây là một file tài liệu XML giúp các công cụ nhận ra chúng là file XML, SGML hay một vài loại ngôn ngữ đánh dấu khác. Khai báo có thể được viết như sau:

- Khai báo dạng đơn giản như sau *<?xml?>* .
- Khai báo bao gồm phiên bản của XML *(<?xml version="1.0"?>).*
- Khai báo bao gồm cả việc mã hóa ký tự *<?xml version="1.0" encoding="utf-8"?>* cho bộ mã tiêu chuẩn quốc tế đa ngôn ngữ Unicode.
- Khai báo không gian tên (NameSpaces): Để sử dụng một không gian tên, cần xác định một tiền tố *xmlns* và đặt chúng trong một chuỗi riêng biệt. Đây là cách có thể xác định tiền tố xmlns cho ba phần tử <title>:

  <?xml version="1.0"?>

  <customer_summary

xmlns:addr="http://www.xyz.com/addresses/"
xmlns:books="http://www.zyx.com/books/"
xmlns:mortgage="http://www.yyz.com/title/">
... <addr:name><title>Mrs.</title> ... </addr:name> ...
... <books:title>Lord of the Rings</books:title> ...
... <mortgage:title>NC2948-388-1983</mortgage:title>

Vì khai báo này phải được ưu tiên thiết lập ở phần đầu tiên trong tệp dữ liệu, cho nên nếu chúng ta có kế hoạch kết hợp các tệp tài liệu XML nhỏ thành một tệp tài liệu XML lớn hơn, có lẽ nên loại bỏ thông tin tùy chọn này.

Tạo phần tử gốc trong tài liệu: Thẻ bắt đầu và thẻ kết thúc của phần tử gốc bao quanh toàn bộ nội dung của file tài liệu XML. Và chỉ có duy nhất một phần tử gốc trong một file dữ liệu, và thẻ "đóng" này để chứa đựng tất cả nội dung của file tài liệu XML.Thí dụ sử dụng phần tử gốc có tên <recipe>.

<?xml version="1.0" encoding="UTF-8"?>
<recipe>
</recipe>

Khi tạo tài liệu, nội dung và các thẻ thêm vào sẽ đặt ở giữa <recipe> và </recipe>.

Đặt tên các phần tử tùy biến: Với ngôn ngữ XML cho phép đặt tên các phần tử một cách tùy biến, sau đó định nghĩa theo Document Type Definition hoặc theo lược đồ XML tương ứng với các tên đó. Những tên phần tử tạo ra có thể bao gồm các ký tự theo hệ thống chữ cái, các chứ số và các ký tự đặc biệt. Với quy tắc đặt tên các phần tử như sau:

- Dấu cách không được phép dùng để đặt tên.
- Tên phải được bắt đầu bằng một ký tự theo hệ thống bảng chữ cái, không phải là số hay ký tự đặc biệt

Ví dụ: Trường hợp có nhiều phần tử, nếu muốn thêm phần tử <recipename>, phần tử này sẽ bắt đầu bằng thẻ <recipename> và thẻ kết thúc tương ứng </recipename>.

<?xml version="1.0" encoding="UTF-8"?>
<recipe>
<recipename>Ice Cream Sundae</recipename>
<preptime>5 minutes</preptime>

</recipe>

Trong file XML có thể tồn tại các thẻ trống, chúng không có bất kỳ nội dung gì bên trong và được biểu thị như một thẻ đơn bao gồm một cặp thẻ mở đầu và kết thúc

Ví dụ: *có <img src="mylogo.gif">*, nó là một phần tử đơn lẻ và đứng một mình. Nó không chứa đựng bất kỳ phần tử con cũng như văn bản nào, chính vì vậy nó là một phần tử trống và có thể biểu thị như sau ** (được kết thúc bởi một dấu cách và gạch chéo kết thúc như thông thường)

Lồng ghép các phần tử: Lồng ghép là đặt một phần tử này nằm bên trong một phần tử khác. Mà trong đó các phần tử mới nằm bên trong được gọi là các phần tử con, các phần tử này được bao bên ngoài bởi một phần tử gọi là cha. Như ở ví dụ bên trên chúng ta có phần tử gốc, phần tử cha là <recipe>, các thông tin của các phần tử con được lồng ghép bên trong bao gồm các phần tử*<recipename>, <ingredlist>* và *<preptime>*.

Bên trong phần tử con <ingredlist> lại có rất nhiều các thông tin của các phần tử con ở trong nó *<listitem>*như ví dụ sau đây:

```
<?xml version="1.0" encoding="UTF-8"?>
<recipe>
    <recipename>Ice Cream Sundae</recipename>
    <ingredlist>
        <listitem>
            <quantity>3</quantity>
            <itemdescription>chocolate syrup</itemdescription>
        </listitem>
    </ingredlist>
    <preptime>5 minutes</preptime>
</recipe>
```

Trong một văn bản XML, các phần tử có thể được lồng ghép với nhau ở nhiều cấp độ. Nghĩa là vẫn có thể lồng ghép các phần tử cấp thấp hơn ở bên trong phần tử con và nhiều cấp nhỏ hơn nữa ở trong nó. Một lỗi cú pháp thông thường là khi lồng ghép không chính xác các phần tử cha và phần tử con. Đó là do bất cứ một phần tử con nào cũng phải được bao quanh một cách trọn vẹn giữa thẻ bắt đầu

và thẻ kết thúc của phần tử cha. Một điều chú ý là các phần tử con phải được đóng trước khi các phần tử con khác bắt đầu.

Thêm các thuộc tính: Các thuộc tính hoàn toàn có thể được thêm vào các phần tử, bao gồm cặp tên-giá trị, với giá trị được đặt bên trong hai dấu ngoặc kép ("), ví dụ: type="dessert". Các thuộc tính này cung cấp cách thức để lưu trữ thêm các thông tin mỗi khi sử dụng phần tử, sự thay đổi giá trị của thuộc tính là cần thiết khi sử dụng cùng một phần tử ở các vị trí khác khác nhau trong cùng một tài liệu.
Chúng được thêm vào bên trong thẻ bắt đầu của một phần tử: <recipe type="dessert">.

Nếu thêm vào nhiều thuộc tính, hãy tách chúng riêng rẽ bằng dấu cách: <recipename cuisine="american" servings="1">.
Ví dụ: Tệp tài liệu XML với các phần tử và các thuộc tính

```
<?xml version="1.0" encoding="UTF-8"?>
    <recipe type="dessert">
        <recipename cuisine="american" servings="1">
            Ice Cream Sundae
        </recipename>
        <preptime>5 minutes</preptime>
    </recipe>
```

Có thể sử dụng một vài hoặc vô số các thuộc tính khi cần thiết. Các tên thuộc tính có thể bao gồm các ký tự giống như quy ước đặt tên phần tử, tên thuộc tính không bao gồm dấu cách và tên phải bắt đầu bằng các ký tự trong bảng chữ cái[10, 11].

2.2. Giới thiệu Kỹ thuật RSS

RSS (Really Simple Syndication) [6] có nghĩa là "dịch vụ cung cấp thông tin thực sự đơn giản", dùng trong việc chia sẻ tin tức Web (Web syndication) được dùng bởi nhiều website tin tức và weblog với công nghệ của RSS cho phép người dùng Internet có thể đặt mua thông tin từ các websites có cung cấp khả năng RSS (*RSS feeds*); chúng thường là các site có nội dung thay đổi và được thêm vào thường xuyên. Để có thể dùng công nghệ này, người quản trị site đó tạo ra hay quản lí một phần mềm chuyên dụng (như là một hệt thống quản lý nội dung - *content management system-CMS*) mà 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. Khác với việc mua nhiều ấn

bản của các tờ báo hay tạp chí in giấy, hầu hết việc mua RSS là miễn phí. 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, và các siêu dữ liệu (meta-data) khác. Thông tin này được cung cấp dưới dạng một tập tin XML được gọi là một RSS feed, *webfeed*, *RSS stream*, hay *RSS channel*.

Định nghĩa RSS: RSS là viết tắt của một trong những thuật ngữ sau:

- Really Simple Syndication: "Dịch vụ cung cấp thông tin thực sự đơn giản".
- Rich Site Summary, RDF Site Summary[13]: Tóm lược thông tin phong phú, tóm lược thông tin theo định dạng RDF.
- RDF - Resource Discovery Framework:(tạm dịch là "cơ cấu khám phá tài nguyên"): là định dạng của phiên bản RSS 1.0. Ở một số trang web biểu tượng RDF sử dụng thay cho RSS.

RSS được viết bằng XML. Một file RSS là một danh sách các đối tượng - chính là các mẫu tin - được miêu tả gồm có: tiêu đề, nội dung tóm lược, một liên kết đến trang chính của tin đó, ngày tháng, tác giả...cũng có thể thêm vào.Một file rss không có phần mở rộng thống nhất. Trong đó phổ biến nhất là: .xml, .rss, .rdf (có thể có nhiều phần mở rộng khác ngoài 3 định dạng này).

Lịch sử phát triển RSS: Trước RSS, có nhiều định dạng khác cũng từng được dùng cho vấn đề chia sẻ thông tin, nhưng không có định dạng nào được dùng rộng rãi cho đến ngày nay, vì hầu hết chủ yếu dùng cho từng dịch vụ đơn. Lịch sử phát triển của RSS trải qua các giai đoạn như sau:

- RDF (*Resource Description Framework*) Site Summary, phiên bản đầu tiên của RSS, được tạo ra bởi Dan Libby củaNetscape vào tháng Ba 1999 dùng cho cổng điện tử My Netscape. Phiên bản này trở thành RSS 0.9. Vào tháng Bảy 1999, đáp trả lại các đề nghị và góp ý, Libby đưa ra bản phác thảo ban đầu đặt tên là RSS 0.91 (RSS viết tắt của Rich Site Summary), nhằm đơn giản hóa định dạng và tích hợp một số phần trong định dạng scriptingNews của Winer.
- Nhóm RSS-DEV tiếp tục đưa ra RSS 1.0vào tháng 12 năm 2000 dựa trên bản phác thảo góp ý sửa đổi cho bản đặc tả kĩ thuật đưa ra bởi Tristan Louis. Giống với RSS 0.9 (không phải 0.91) bản này dựa vào đặc tả kĩ thuật của RDF, nhưng có tính khả mở hơn, với nhiều mục bắt nguồn từ các từ vựng metadata chuẩn như Dublin Core.

- Vào tháng 9 năm 2002, Winer cho ra bản cuối cùng của RSS 0.92, bây giờ gọi là *RSS 2.0* và nhấn mạnh "Really Simple Syndication" là nghĩa của ba kí tự viết tắt RSS. Đặc tả kĩ thuật của RSS 2.0 loại bỏ thuộc tính *type* từng được thêm vào trong RSS 0.94 và cho phép người dùng có thể thêm thành phần mở rộng nhờ dùng XML Namespaces. Nhiều phiên bản của RSS 2.0 đã được ra đời, nhưng chỉ số của phiên bản thì vẫn không thay đổi.
- Vào tháng 11, 2002, Thời báo New York đã bắt đầu cung cấp cho người đọc khả năng mục các tin có hỗ trợ *RSS feeds* liên quan đến nhiều chủ đề khác nhau. Vào tháng Giêng, 2003, David Winer đã gọi việc dùng RSS của thời báo New York Time là một "điểm nhấn" (*tipping point*) trong việc đưa định dạng RSS trở thành một chuẩn[6].

Cấu trúc định dạng tập tin RSS 2.0 [6]: Một file RSS là do một phần tử (element) ***<channel> </channel>*** và các phần tử con của nó tạo nên. Ngoài nội dung <Item> được liệt kê trong phần thân <item> </item> thì phần tử <channel> còn chứa các element đại diện cho siêu dữ liệu của kênh RSS - chẳng hạn như <title>, <link> và <description>. Những mục này là các phần chính của kênh RSS và chứa nội dung thường hay thay đổi. Dưới đây là chi tiết các thành phần:

- **Thành phần <channel> của RSS**

Một kênh **<channel>** thường có ba phần tử để biết thông tin của kênh bao gồm: <title> (tiêu đề): Tên của kênh hoặc nguồn cấp tin <link> (liên kết): URL của trang web hoặc vùng trang web có liên kết với kênh này <description> (mô tả): Mô tả ngắn gọn về kênh đó.

Ngoài các thành phần trên thì **<channel>** còn có nhiều phần tử con là tùy chọn và không bắt buộc.

Ví dụ như phần tử <image> (hình ảnh) có ba phần tử con bắt buộc là:

- <url>: URL của một hình ảnh GIF, JPEG hoặc PNG đại diện cho kênh.
- <title>: Mô tả hình ảnh. Nó được dùng cho thuộc tính ALT của thẻ <image> trong HTML khi kênh được hiển thị bằng HTML.
- <link>: URL của trang web. Khi kênh được hiển thị dưới dạng HTML, hình ảnh có thể hoạt động như một liên kết đến trang web.

Phần tử <image> cũng có ba phần tử con không bắt buộc sau:

- <width> (chiều rộng): Quy định chiều rộng hình ảnh tính bằng pixel. Giá trị tối đa là 144 và giá trị mặc định là 88.
- <height> (chiều cao): Quy định chiều cao hình ảnh tính bằng pixel. Giá trị tối đa là 400 và giá trị mặc định là 31.
- <description>: Chứa đoạn văn bản có trong thuộc tính title của liên kết tạo thành hình ảnh khi được hiển thị.

Ngoài ra, có thể sử dụng các phần tử tùy chọn khác:

- <language> (ngôn ngữ): en-us (anh-mỹ)
- <copyright> (bản quyền): Copyright 2003, James Lewin
- <managingEditor> (người biên tập quản lý): dan@spam_me.com (Dan Deletekey)
- <webMaster>: dan@spam_me.com (Dan Deletekey)
- <pubDate> (ngày xuất bản): Sat, 15 Nov 2003 0:00:01 GMT (Thứ Bảy, 15.11.2003, giờ chuẩn theo kinh tuyến Greenwich (GMT) là 0:00:01)
- <lastBuildDate> (ngày dựng cuối cùng): Sat, 15 Nov 2003 0:00:01 GMT
- <category> (thể loại): ebusiness (thương mại điện tử)
- <generator> (trình tạo): CMS 2.0
- <docs> (các tài liệu): http://blogs.law.harvard.edu/tech/rss
- <cloud> (đám mây): Cho phép đăng ký với một "đám mây" để được thông báo về các bản cập nhật cho kênh, thực hiện một giao thức đăng ký - xuất bản nhẹ cho các nguồn cấp tin RSS.
- <ttl>: Thời gian sống (Time to live) một con số đại diện cho số phút mà một nguồn cấp tin có thể được lưu trữ trong bộ nhớ đệm trước khi nó cần được làm mới.
- <rating> (đánh giá): Đánh giá PICS cho kênh.
- <textInput> (nhập văn bản): Định nghĩa hộp nhập có thể được hiển thị với kênh.
- <skipHours>: Thông tin này để báo cho các chương trình tổng hợp <Item> biết rằng có thể bỏ qua không cập nhật nội dung trong bao nhiêu giờ.
- <skipDays>: Thông tin này để báo cho các chương trình tổng hợp <Item> biết rằng có thể bỏ qua không cập nhật nội dung trong bao nhiêu ngày.

▪ **Thành phần <Item> bản tin**

Các **<Item>** là phần quan trọng nhất của một nguồn cấp tin. Mỗi <Item> có thể là một bài weblog, một bài viết hoàn chỉnh, một bài phê bình phim, một mục

rao vặt trên báo, hoặc bất cứ thứ gì muốn cung cấp cho kênh. Các <Item>có thể thường xuyên thay đổi nội dung, có thể chứa bao nhiêu <Item>cũng được. Đối với phiên bản đặc tả trước đó thì có một giới hạn là 15 <Item>và nếu muốn bảo đảm tính tương thích ngược thì có thể tuân theo giới hạn này.

Một <Item>thường chứa ba phần tử như sau:

- <title>: Phần tử này là tên của <Item>. Theo tiêu chuẩn sử dụng, phần tử này sẽ được dịch thành một đầu đề trong mã HTML.
- <link>: Phần tử này là URL của <Item>. Thường thì tiêu đề được sử dụng như là một liên kết trỏ đến URL chứa bên trong phần tử <link>này.
- <description>: Phần tử này thường là một bản tóm tắt hoặc lời nhận xét về <Item>.

Tất cả các phần tử đều là tùy chọn, nhưng một <Item> phải có ít nhất một phần tử <title> hoặc một phần tử <description>.

Một số phần tử tùy chọn khác trong <Item>có thể được sử dụng:

- <author> (tác giả): Địa chỉ Thư điện tử của tác giả.
- <category> (thể loại): Hỗ trợ phân loại <Item>.
- <comments> (các nhận xét): URL của một trang để viết các nhận xét về <Item>.
- <enclosure> (đính kèm): Hỗ trợ các đối tượng đa phương tiện liên quan đến <Item>.
- <guid>: Một liên kết cố định được gắn đồng nhất với <Item>.
- <pubDate> (ngày xuất bản): Ngày xuất bản của <Item>.
- <source> (nguồn): Kênh RSS là nguồn của các bản tin. Phần tử này có thể có ích khi các <Item> được gộp chung với nhau.

Ở đây <chanel> được chứa trong <rss version="2.0">. Ví dụ rất cơ bản này cho thấy các mục và hình ảnh được chứa như thế nào trong <chanel>. Các phần tử được hiển thị là các phần tử con của <chanel> được dùng phổ biến nhất.

Ví dụ File RSS 2.0 đơn giản

```
<?xml version="1.0"?>
<rss version="2.0">
<channel>
<title>tiêu đề của channel</title>
<link>http://www.urlofthechannel.com/</link>
<description>Nội dung mô tả của channel.
```

</description>
<language>Ngôn ngữ thể hiện</language>
<image>
<title>Tiêu đề của hình ảnh</title>
<url>http://www.urlofthechannel.com/images/logo.gif</url>
<link>http://www.urlofthechannel.com/</link>
</image>
<item>
<title>Tiêu đề của các nội dung thành phần</title>
<link>http://www.itworld.com/nl/ecom_in_act/11122003/</link>
<description>Mô tả nội dung của các thành phần .</description>
</item>
</channel>
</rss>[6, 11, 15].

2.3. Phương pháp Interface DOM phân tích RSS 2.0

Khái niệm: Document Object Model [1] là một phương pháp tiếp cận dữ liệu từ XML, trong đó tất cả các thành phần của XML như thẻ, thuộc tính, text… đều được xem như là các đối tượng. Dựa vào các đối tượng này mà chúng ta có thể trích xuất thông tin, thay đổi thông tin hay truy vấn thông tin dựa vào các hàm mà đối tượng cung cấp.

W3C DOM là tiêu chuẩn mà W3School giới thiệu. Tiêu chuẩn này không phụ thuộc vào hệ điều hành hay ngôn ngữ lập trình . W3C DOM được chia làm 3 phần: Core DOM là tập hợp các đối tượng dùng cho cấu trúc 1 văn bản; XML DOM là tập hợp các đối tượng dùng cho cấu trúc 1 văn bản XML; HTML DOM là tập hợp các đối tượng dùng cho cấu trúc 1 văn bản HTML.

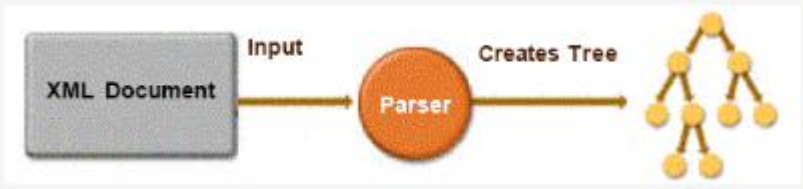
Các đối tượng trong DOM: Tất cả các thành phần trong XML đều được xem như là 1 "node". Trong đó root là document node, các thẻ là element node, thuộc tính là attribute node, text là text node, ghi chú là comment node….
Khi parser đọc XML thành đối tượng DOM trên bộ nhớ chúng ta sẽ có 1 cấu trúc cây với các đối tượng là các nodevà trên cấu trúc này chúng ta sẽ thấy mối quan hệ phân tầng giữa các node. Trên cùng là thẻ root (documentnode) từ node này chúng ta có thẻ rẽ nhánh đến các tầng thấp hơn của cấu trúc cây cho đến khi chúng ta chạm tới text node là tầng thấp nhất của cấu trúc này.

Mối quan hệ giữa các node được mô tả bằng thuật ngữ “ parent” và “ child” (cha và con). Thuật ngữ “leaf node” dùng để chỉ các node không có child node

Các đặc điểm của DOM: Truy cập tài liệu XML như là một cấu trúc cây, việc truy cập đượctới hầu hết các nút element và các nút text. Có thể "rà xoát" (Traversing) cây từ sau ra trước.Với việc tải và lưu trữ toàn bộ cây DOM đòi hỏi máy tính phải cung cấp bộ nhớ lớn. Mô hình DOM có thể gây ra nặng nề trong việc tải và lưu dữ liệu. DOM được sử dụng khi rà xoát và hiệu chỉnh cây

Với một tài liệu XML mô hình DOM sẽ duyệt và chuyển nó thành một mô hình cây của các Object .Ví dụ một element tương ứng với một Object element, một thuộc tính tương ứng với một Object thuộc tính các Object này được tạo ra trong bộ nhớ (memory) và có cấu trúc cây. Chính vì vậy mà có thể rà xoát tài liệu XML này bằng cách rà xoát trên các Object trong bộ nhớ và có thể thêm bớt các thông tin, object để tạo ra một tài liệu XML mới hoặc một sản phẩm mới.

Hoạt động của DOM: Hình 2.3.1 cho thấy đầu vào là một tài liệu XML được bộ phân tích bởi mô hình DOM và một sẽ cây được tạo ra trong bộ nhớ mangthông tin của tài liệu đó. Việc phân tích tài liệu XML bây giờ đưa về phân tích, xử lý các nút của cây.



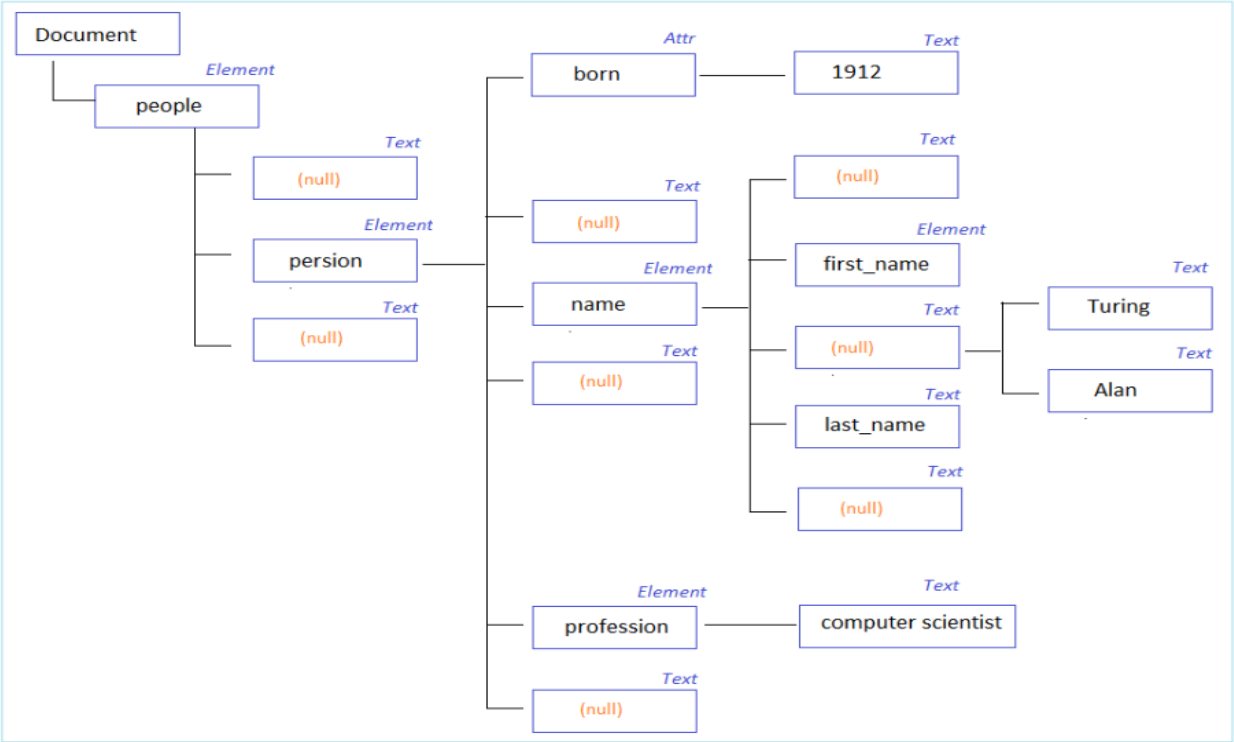
Hình 2.1 - Mô hình hoạt động DOM[3]

Cấu tạo cây DOM và các kiểu nút(Node): Tài liệu XML được hình dung như là một cây. Một cây được làm từ các nút cây (nodes), có 12 kiểu nút cây khác nhau.
Các nút cây có thể chứa các nút cây khác (phụ thuộc vào kiểu nút cây là gì). Các nút cha gồm có các nút con, các nút con lại có thể gồm các nút con khác.
Tài liệu XML được cấu tạo bởi 12 kiểu Node khác nhau như liệu kê dưới đây:

- Document (Mô tả một nút lớn nhất đó là toàn bộ tài liệu XML)
- DocumentFragment (Một đoạn tài liệu XML)
- Element
- Attr (Nút thuộc tính)
- Text (Nút chứa text)

- Comment (Ghi chú trong tài liệu XML)
- ProcessingInstruction (Tương ứng với chỉ lệnh trong XML)
- DocumentType (Định nghĩa XML)
- Entity (Tương ứng với thực thể trong XML)
- EntityReference (Tương ứng với các thực thể tham chiếu trong XML)
- CDATASection(Tương ứng với các phân đoạn trong XML)
- Notation(Tương ứng với các chú thích NOTATION trong XML)

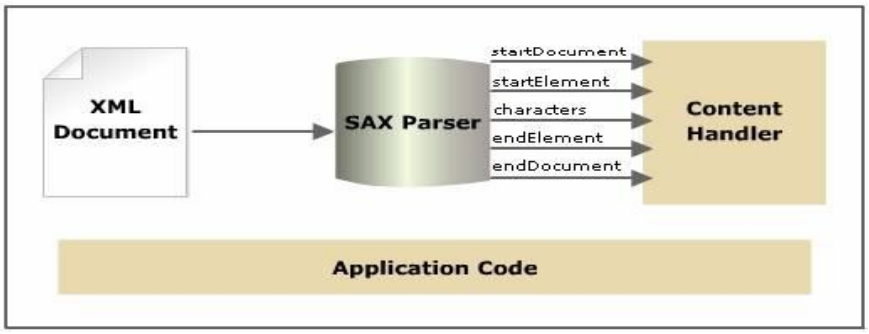
Mô hình cây DOM được thể hiện dưới các dạng node (xem Hình 2.2):



Hình 2.2 - Mô hình tổng thể cây DOM[3]

2.4. Phương pháp Interface SAX phân tích RSS 2.0

Simple API for XML (SAX)[4] là một phương pháp phân tích dữ liệu XML dựa vào sự kiện trong quá trình đọc file XML từ trên xuống dưới. Như vậy để dùng phương pháp này các ngôn ngữ lập trì nh cần định nghĩa ra các sự kiện và các hàm tương ứng với từng sự kiện . Khi đó người lập trình sẽ viết class dựa trên các hàm mà ngôn ngữ lập trình cung cấp và dùng class này để phân tích dữ liệu XML . Trong quá trình phân tích khi gặp sự kiện nào parser sẽ gọi các đoạn mã trong hàm tương ứng sự kiện để thự c thi ví dụ như trong Java khi parser gặp sự kiện mở thẻ, parser sẽ gọi lại các đoạn code trong hàm "startElement" để thi hành



Hình 2.3 - Mô hình SAX xử lý parser XML[4]

Một điểm cần lưu ý rằng SAX không ghi nhớ XML như một cấu trúc cây trên bộ nhớ. SAX chỉ xây dựng cấu trúc của nhiệm vụ đang thi hành trên bộ nhớ do đó SAX sẽ thực hiện nhanh hơn và ít tốn tài nguyên hơn.

Bộ phân tích SAX được gọi là SAXParser và được tạo bằng javax.xml.parsers.SAXParserFactory. Khác với bộ phân tích DOM, bộ SAXParser không tạo ra một hình thức đại diện của tài liệu XML trong bộ nhớ và vì thế nó hoạt động nhanh hơn ít tốn bộ nhớ hơn. Thay vào đó, bộ phân tích SAXParser thông báo cho các trình khách cấu trúc của tài liệu XML bằng cách gọi các hàm callbacks, nghĩa là, bằng cách gọi các phương thức của trường hợp bản mẫu: Javadoc: SE đã được cung cấp cho bộ phân tích.

SAX gồm có 2 interface chính: XMLReader là interface dùng để đọc XML và ContentHandler dùng để nhận dữ liệu từ XML . Hai interface này giải quyết 90% nhu cầu của người dùng với SAX.

Lớp DefaultHandler nằm trong gói org.xml.sax.helpers, lớp này thực hiện các giao diện ContentHandler, ErrorHandler, DTDHandler và EntityResolver. Đại bộ phận các trình khách chỉ quan tâm đến những phương pháp được định nghĩa trong giao diện ContentHandler.

Các phương pháp của giao diện ContentHandler, được DefaultHandler thực hiện, được gọi đến khi bộ phân tích SAX bắt gặp những phần tử tương ứng trong bản tài liệu XML. Những phương pháp chủ yếu trong giao diện này bao gồm:

```
public interface ContentHandler {
public void setDocumentLocator(Locator locator);
public void startDocument() throws SAXException;
public void endDocument() throws SAXException;...
}
```

Các trình khách cung cấp một phân tích lớp của DefaultHandler.Phân lớp này được sử dụng để lấn quyền những phương pháp trên và xử lý dữ liệu.Quá trình này có thể bao gồm việc lưu trữ dữ liệu vào trong cơ sở dữ liệu, hoặc viết chúng ra một luồng dữ liệu.

2.5. Phân tích nội dung XML trong Android sử dụng XmlPullParser

Sơ lược về XmlPullParser[12, 16]: Android không cung cấp hỗ trợ cho StAX API của Java. Tuy nhiên Android lại đi kèm với một trình phân tích XmlPullParser làm việc tương tự như StAX.Nó cho phép mã ứng dụng "kéo" hoặc tìm kiếm các sự kiện từ trình phân tích, trái ngược với trình phân tích SAX tự động đẩy các sự kiện cho trình xử lý. XMLPullParser là một trình giao diện được định nghĩa để cung cấp các tính năng phân tích cú pháp XML trong gói XMLPULL V1 API.

Theo đó các loại phân tích cú pháp khác nhau phụ thuộc vào các tính năng được thiết lập:

- *non-validating*: loại phân tích cú pháp không kiểm duyệt khi *FEATURE_PROCESS_DOCDECL* được thiết lập giá trị True
- validating parser:Phân tích cú pháp có kiểm duyệt khi *FEATURE_VALIDATION* được thiết lập giá trị True (mặc định là FEATURE_PROCESS_DOCDECL được thiết lập giá trị True)
- Khi FEATURE_PROCESS_DOCDECL có giá trị FALSE (giá trị này là mặc đinh và nếu có giá trị khác được yêu cầu thì phải thiết lập trước khi phân tích cú pháp bắt đầu). Khi đó trình phân tích cú pháp sẽ hoạt động như non-validating với điều kiện trong tài liệu XML không có DOCDECL khai báo kiểu tài liệu trong file XML (Các thực thể nội bộ bên trong vẫn được định nghĩa với phương thức defineEntityReplacementText()). Chế độ này được thiết kế để hoạt động trong các môi trường hạn chế nhất định như J2ME.

Ưu điểm của XmlPullParser: Phương pháp này với khá nhiều ưu điểm đạt được như đơn giản trong xử lý mã nguồn, sử dụng ít tài nguyên bộ nhớ thiết bị, tốc độ xử lý nhanh hơn phương pháp DOM và SAX do tập trung vào dữ liệu hiện hành để xử lý, tính năng lọc dữ liệu tương đối đơn giản dễ sử dụng.

Đặc điểm của XmlPullParser[12, 16]: Bộ phân tích của XmlPullParser xem tài liệu XML chỉ có các phần tử và các nút văn bản tương ứng với phần tử để

phân tích cú pháp. Các thuộc tính không tạo ra một sự kiệncó thể lấy các thuộc tính ra từ một phần tử. Chúng ta chỉ tìm thấy các kiểu sự kiện START_TAG và TEXT tương ứng với các thẻ bắt đầu phần tử và các nút văn bản của phần tử. Đầu tiên, hãy xác định thẻ phần tử rồi lấy giá trị nút văn bản cho thẻ phần tử đó. Sử dụng biến kiểu int là iter để chỉ ra các phần tử "article" khác nhau trong tài liệu XML và biến kiểu String là elemtext để chỉ rõ tên thẻ phần tử. Để chỉ rõ một biến kiểu int cho một trình lặp lại (iterator) và một biến kiểu String cho tên phần tử, hãy sử dụng đoạn mã trong ví dụ 1.

Ví dụ 1 Chỉ rõ các biến

```
int iter = 0;
String elemtext = null;
```

Trước khi kết thúc tài liệu XML cần hoàn thành:

- Xác định các kiểu sự kiện.
- Lấy các tên thẻ phần tử và các giá trị văn bản thẻ.
- Thiết lập các giá trị nút văn bản trên các widget TextView tương ứng.

Lấy tên thẻ phần tử như trong ví dụ 2.

Ví dụ 2 Lấy các tên thẻ phần tử cho các thẻ bắt đầu

```
while (eventType != XmlPullParser.END_DOCUMENT) {
if (eventType == XmlPullParser.START_TAG) {
        String elemName = xpp.getName();
...
                                        }
                                    }
```

Nếu tên thẻ phần tử là "catalog", hãy lấy các giá trị của các thuộc tính "journal" và "publisher" " và thiết lập các giá trị thuộc tính trên các widget TextView "journal" và "publisher". Sử dụng phương thức getAttributeValue() của XmlResourceParser để nhận được các giá trị thuộc tính, như trong ví dụ 3.

Ví dụ 3 Lấy và thiết lập các giá trị thuộc tính

```
if (elemName.equals("catalog")) {
String journalAttr = xpp.getAttributeValue(null,"journal");
String publisherAttr = xpp.getAttributeValue(null,"publisher");
journal.setText(journalAttr);
publisher.setText(publisherAttr);
...
```

```
}
```

Tăng thêm biến trình lặp iter cho mỗi phần tử "article", như trong ví dụ 4.

Ví dụ 4 Tăng thêm biến

```
if (elemName.equals("article")) {
iter = iter + 1;
                    }
```

Lấy các giá trị nút văn bản nếu kiểu sự kiện là TEXT và thiết lập các giá trị nút văn bản trên các widget TextView tương ứng. Sử dụng biến elemtext kiểu String, đã được thiết lập cho kiểu sự kiện START_TAG, để lấy tên thẻ phần tử. Sử dụng phương thức getText() củaXmlResourceParser để lấy giá trị nút văn bản. Thiết lập các giá trị nút văn bản trên các widget TextView bằng cách sử dụng phương thức setText, như trong ví dụ 5.

Ví dụ 5 Lấy các giá trị nút văn bản

```
else if (eventType == XmlPullParser.TEXT) {
//Obtain the element name and element text node values and
//set the text node values on the corresponding TextView
//widgets
}
```

Với trình phân tích cú pháp kéo, một tài liệu XML chỉ là một dãy các sự kiện phân tích cú pháp. Hãy lấy sự kiện phân tích cú pháp tiếp theo bằng cách sử dụng phương thức next ( ) (tiếp theo), như sau: xpp.next();.

- *START_TAG:* An bắt đầu từ khóa XML đã được đọc.
- *TEXT*: nội dung văn bản được đọc; các nội dung văn bản có thể được lấy ra bằng cách sử dụng gettext () phương pháp.
- *END_TAG*: Một kết thúc từ khóa đã được đọc.
- *END_DOCUMENT*: Không có thêm các sự kiện có sẵn.

CHƯƠNG 3
KỸ THUẬT PHÂN TÍCH WEBSITE VỚI JSOUP

3.1. Giới thiệu

Ngày nay khối lượng thông tin lưu trữ trên Internet ngày càng tăng chóng mặt, việc trích rút được các thông tin chính xác, gọn gẽ từ kho dữ liệu trên là vô cùng có ý nghĩa. Tuy nhiên đây là một bài toán tương đối phức tạp, để có thể làm được điều này trước tiên ta cần xác định được phần nội dung chính của một trang Web bất kỳ. Trang web bất kỳ ở đây có nghĩa là cấu trúc của trang Web không hề được biết trước. Hơn nữa trang Web phải thực sự có nội dung chính nghĩa là nếu chúng ta nhìn vào đó thì có thể biết được đâu là chủ đề của trang Web thì việc xác định này mới có ý nghĩa. Vấn đề đặt ra đã làm nảy sinh các nhu cầu nghiên cứu, xử lý trên khối lượng dữ liệu thông tin này sao cho hiệu quả và nhanh chóng nhất. Đã có rất nhiều ứng dụng và nhiều Tools ra đời như: Search Engine[18], RSS[15], Feedback[17], jSon[14], Jsoup[5]...

Trong đó Tool Jsoup là một thư viện mã nguồn mở của Java. Jsoup là một công cụ phân tích cú pháp mã HTML, cung cấp nhiều Packages và nhiều API thuận tiện, an toàn để thao tác với HTML.

3.2. Định nghĩa Jsoup

Jsoup là Java HTML Parser[5]. Nói cách khác Jsoup là một thư viện được sử dụng để phân tích tài liệu HTML. Jsoup cung cấp các API dùng để lấy dữ liệu và thao tác dữ liệu từ URL hoặc từ file HTML. Nó sử dụng các phương thức giống với DOM, CSS, JQuery để lấy dữ liệu và thao tác với dữ liệu.

Jsoup thực hiện các đặc điểm kỹ thuật HTML WHATWG, và phân tích cú pháp HTML tương tự như DOM mà các trình duyệt hiện đại thực hiện như:

- Phân tích cú pháp HTML từ một URL, file hoặc chuỗi.
- Tìm và trích xuất dữ liệu sử dụng DOM hoặc CSS Selector.
- Xử lý các phần tử, thuộc tính, text của HTML.
- Dữ liệu người dùng gửi được bảo vệ an toàn, chặn lỗ hổng bảo mật (XSS) cho phép Hacker chèn mã độc vào Web.
- Xuất dữ liệu đầu ra HTML gọn gàng.

Jsoup được thiết kế để làm việc với tất cả các phiên bản HTML trên thực tế, từcơ bản và xác nhận, tới không hợp lệ tag-soup, Jsoup sẽ tạo ra một cây phân tích phù hợp.

3.3. Thành phần của Jsoup API

Jsoup có 6 Packages và nhiều Class khác nhau. Trong khung khổ của luận văn này chỉ xin phép giới thiệu 3 lớp rất quan trọng đó là các lớp:

- org.jsoup.Jsoup;
- org.jsoup.nodes.Document ;
- org.jsoup.nodes.Element;

Sau đây là các mô tả kỹ thuật về các phương thức được cung cấp bởi các lớp:

Jsoup.java

Phương thức	Mô tả
static Connection connect(String url)	Tạo và trả về đối tượng Connection kết nối của URL
static Document parse(File in, String charsetName)	Phân tích 1 file tài liệu html với chỉ định mã hóa.
static Document parse(File in, String charsetName, String baseUri)	Phân tích file tài liệu html với chỉ định mã hóa, và baseUri
static Document parse(String html)	Phân tích mã html chuyển thành Document
static Document parse(String html, String baseUri)	Phân tích mã html với basiUri thành đối tượng Document.
static Document parse(URL url, int timeoutMillis)	Phân tích một URL thành document.
static String clean(String bodyHtml, Whitelist whitelist)	Trả về HTML an toàn từ HTML đầu vào, bằng cách phân tích HTML đầu vào và lọc nó qua một danh sách trắng của các thẻ và các thuộc tính được phép.

Bảng 3.1 - Các phương thức của lớp Jsoup.java

Document.java

Phương thức	Mô tả

Element body()	Truy nhập vào phần tử body
Charset charset()	Trả về charset được sử dụng trong tài liệu này
void charset(Charset charset)	Sét charset sử dụng cho tài liệu này.
Document clone()	Tạo một phiên bản copy của tài liệu này, bao gồm copy cả các node con.
Element createElement(String tagName)	Tạo mới một phần tử
static Document createShell(String baseUri)	Tạo một tài liệu rỗng, thích hợp cho việc thêm các phần tử vào nó.
Element head()	Truy cập vào phần tử head.
String location()	Trả về URL của tài liệu này.
String nodeName()	Trả về node name của node này.
Document normalise()	Normalise the document.
String outerHtml()	Trả về outer HTML của node này.
Document.OutputSettings outputSettings()	Trả về các sét đặt đầu ra hiện tại của tài liệu.
Document outputSettings(Document.OutputSettings outputSettings)	Sét đặt đầu ra cho tài liệu.
Element text(String text)	Sét đặt text của body của tài liệu này.
String title()	Trả về nội dung tiêu đề của tài liệu.
void title(String title)	Sét đặt nội dung tiêu đề cho tài liệu.
boolean updateMetaCharsetElement()	Trả về true nếu phần tử với thông tin charset trong tài liệu này đã bị cập nhập thay đổi thông qua Document.charset(Charset).
void updateMetaCharsetElement(boolean update)	Sét đặt phần tử với thông tin charset trong tài liệu này đã bị cập nhập thay đổi thông qua Document.charset(Charset) hay không.

Bảng 3.2- Các phương thức của lớp Document.java

3.4. Các phương thức DOM

Jsoup có một vài phương thức gần giống với các phương thức trong mô hình DOM (Phân tích tài liệu XML)

Phương thức	Mô tả
Element getElementById(String id)	Tìm một phần tử cho bởi ID, bao gồm hoặc bên dưới phần tử này.
Elements getElementsByTag(String tag)	Tìm các phần tử, bao gồm và cả đệ quy dưới phần tử này, với tên thẻ chỉ định.
Elements getElementsByClass(String className)	Tìm phần tử có classNam cho bởi tham số, bao gồm hoặc dưới phần tử này.
Elements getElementsByAttribute(String key)	Tìm kiếm các phần tử có thuộc tính cho bởi tham số, không phân biệt chữ hoa chữ thường.
Elements siblingElements()	Trả về các phần tử anh em với phần tử hiện tại.
Element firstElementSibling()	Trả về phần tử anh em đầu tiên của phần tử hiện tại.
Element lastElementSibling()	Trả về phần tử anh em cuối cùng của phần tử hiện tại.

Bảng 3.2- Các phương thức của DOM

Các phương thức lấy dữ liệu trên Element

Phương thức	Mô tả
String attr(String key)	Trả về giá trị thuộc tính cho bởi key của phần tử này.
void attr(String key, String value)	Sét giá trị thuộc tính. Nếu thuộc tính đã tồn tại, nó sẽ bị thay thế.
String id()	Trả về thuộc tính ID, nếu có, hoặc trả về string rỗng nếu không có.
String className()	Trả về chuỗi chữ giá trị của thuộc tính

	"class", nó có thể chứa nhiều class name, ngăn cách bởi khoảng trắng. (Ví dụ trả về " header gray")
Set<String> classNames()	Trả về tất cả các class names. Ví dụ , trả về tập hợp 2 phần tử "header" và "gray".Chú ý, sửa đổi trên tập hợp này không làm thay đổi thuộc tính của phần tử. Muốn thay đổi sử dụng phương thức*classNames(java.util.Set)*.
String text()	Trả về một văn bản kết hợp text của nó và tất cả các text của tất cả các phần tử con.
void text(String value)	Gán text cho phần tử này.
String html()	Trả về String các HTML bên trong thẻ này. Ví dụ * a * trả về * a *. (Node.outerHtml() sẽ trả về * a *.)
void html(String value)	Sét Html bên trong phần tử này. Xóa hết các HTML sẵn có bên trong.
Tag tag()	Trả về Tag cho phần tử này.
String tagName()	Trả về tên thẻ của phần tử này. Ví dụ div.

Bảng 3.3- Các phương thức lấy dữ liệu của Element

Các phương thức vận dụng HTML

Methods	Description
Element append(String html)	Nối thêm HTML vào trong phần tử này. Html được cung cấp sẽ được phân tích, và các node sẽ được nối vào phía cuối tập các node con của phần tử này.
Element prepend(String html)	Nối thêm HTML vào phần tử này. Html được cung cấp sẽ được phân tích, và các node sẽ được nối vào phía trước tập các node con của phần tử này.

Element appendText(String text)	Tạo và nối một TextNode mới vào phần tử này.
Element prependText(String text)	Tạo và nối một TextNode mới vào phía trước tập các node con của phần tử này.
Element appendElement(String tagName)	Tạo một phần tử mới cho bởi tag name. Và nối nó vào như phần tử con ở cuối cùng.
Element prependElement(String tagName)	Tạo mới một phần tử bởi tag name, và nối nó vào như phần tử con đầu tiên.
Element html(String value)	Sét đặt html bên trong phần tử này. Xóa hết Html bên trong sẵn có.

Bảng 3.3 - Các phương thức vận dụng của HTML

3.5. Các phương thức giống Css, jQuery

Các phần tử JSoup hỗ trợ cú pháp giống với CSS (hoặc JQuery) giúp tìm kiếm các phần tử phù hợp, những hỗ trợ như vậy là rất mạnh mẽ. Các phương thức lựa chọn có sẵn trong class Document, Element hoặc Elements.

Selector Unit (Bộ lựa chọn)

Các Selector	Mô tả
tagname	Tìm kiếm các phần tử theo tên thẻ. Ví dụ: a
ns|tag	Tìm kiếm các phần tử theo tên thẻ trong một không gian tên (namespace), ví dụ fb|name nghĩa là tìm các phần tử <fb:name>
#id	Tìm kiếm phần tử theo ID, ví dụ #logo
.class:	Tìm kiếm các phần tử theo tên class, ví dụ .masthead
[attribute]	Các phần tử với thuộc tính, ví dụ [href]
[^attr]	Các phần tử với thuộc tính bắt đầu bởi, ví dụ [^data-] tìm kiếm các phần tử với thuộc tính bắt đầu bởi data-

[attr=value]	Các phần tử với giá trị thuộc tính, ví dụ [width=500] (Cũng có thể sử dụng dấu nháy kép)
[attr^=value], [attr$=value], [attr*=value]	Các phần tử với giá trị thuộc tính bắt đầu, kết thúc bởi, hoặc chứa giá trị, ví dụ [href*=/path/]
[attr~=regex]	Các phần tử với giá trị khớp với biểu thức chính quy, ví dụ img[src~=(?i)\.(png|jpe?g)]
*	Tất cả các phần tử, ví dụ *

Bảng 3.4- CácSelector Unit

Selector kết hợp

Selector	Mô tả
el#id	Phần tử với ID, ví dụ div#logo
el.class	Các phần tử với class, ví dụ div.masthead
el[attr]	Các phần tử với thuộc tính, ví dụ a[href]
Kết hợp bất kỳ	ví dụ a[href].highlight
ancestor child	(Phần tử tổ tiên- và hậu duệ) Các phần tử hậu duệ của một phần tử, ví dụ. .body p tìm kiếm các phần tử p bất kỳ làfinds p elements anywhere under a block with class "body"
parent > child	Các phần tử con trực tiếp của phần tử cha, ví dụ div.content > p tìm kiếm các phần tử p là con trực tiếp của div có class ='content'; và body > * tìm kiếm các phần tử con trực tiếp của thẻ body
siblingA + siblingB	Tìm kiếm phần tử anh em B ngay phía trước của phần tử A, ví dụ div.head + div
siblingA ~ siblingX	Tìm kiếm các phần tử anh em X trước phần tử A, ví dụ h1 ~ p
el, el, el	Nhóm nhiều Selector, tìm kiếm các phần tử khớp với một trong những Selector; ví dụ div.masthead, div.logo

Bảng 3.5- CácSelector kết hợp

Pseudo selectors

Selector	Mô tả
:lt(n)	Tìm kiếm các phần tử có chỉ số anh em (vị trí trong cây DOM quan hệ với phần tử cha của nó) nhỏ hơn n; ví dụ td:lt(3)
:gt(n)	Tìm kiếm các phần tử có chỉ số anh em lớn hơn n, ví dụ div p:gt(2)
:eq(n)	find elements whose sibling index is equal to n; e.g. form input:eq(1)
:has(seletor)	Tìm kiếm các phần tử chứa các phần tử khớp với selector; ví dụ div:has(p)
:not(selector)	Tìm kiếm các phần tử không khớp với selector; ví dụ div:not(.logo)
:contains(text)	Tìm kiếm các phần tử chứa đoạn text đã cho. Tìm kiếm không phân biệt chữ hoa chữ thường; ví dụ p:contains(jsoup)
:containsOwn(text)	Tìm kiếm các phần tử trực tiếp chứa đoạn text đã cho
:matches(regex)	Tìm kiếm các phần tử mà text khớp với biểu thức chính quy chỉ định; ví dụ div:matches((?i)login)
:matchesOwn(regex)	Tìm kiếm các phần tử mà text của nó khớp với biểu thức chính quy chỉ định.
Chú ý: Cách đánh chỉ số pseudo bắt đầu từ 0, phần tử đầu tiên có chỉ số 0, phần tử thứ 2 có chỉ số 1...	

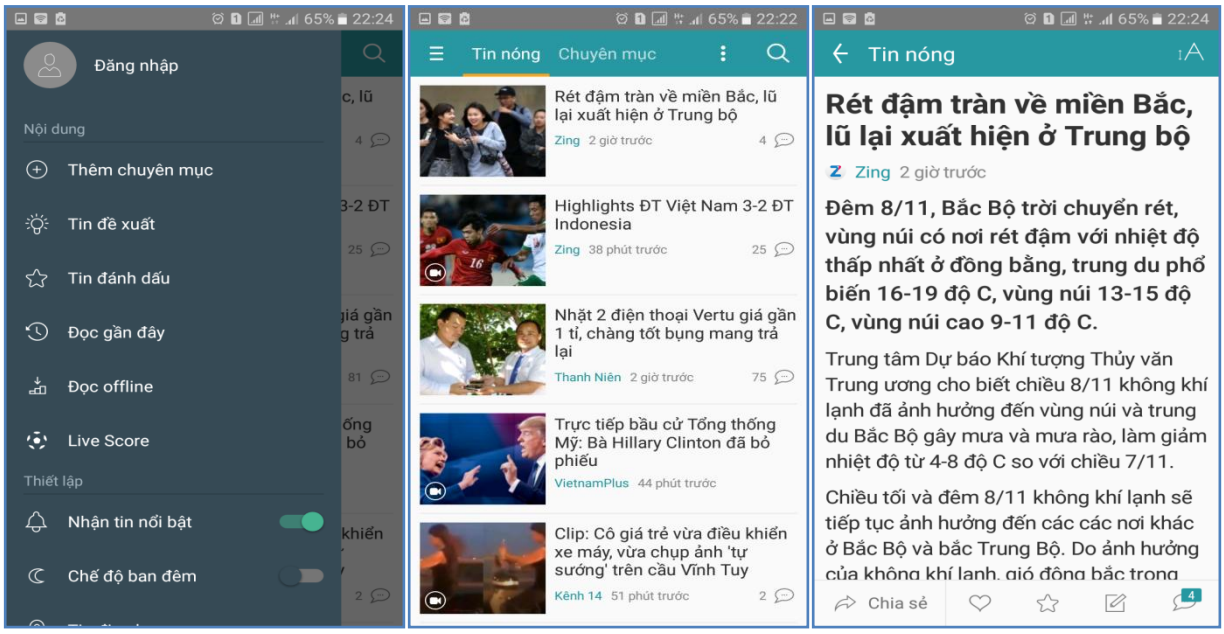
Bảng 3.6- CácPseudo Selector

CHƯƠNG 4
XÂY DỰNG VÀ PHÁT TRIỂN ỨNG DỤNG TỰ ĐỘNG CẬP NHẬT TIN TỨC

4.1. Một số ứng dụng hỗ trợ đọc tin tức từ file RSS hiện có

Báo mới[7] là một ứng dụng đọc tin tức online và offline từ trang báo điện tử http://www.baomoi.com. Ứng dụng được xây dựng trên cơ sở sử dụng công nghệ Android. Một số chức năng chính nổi trội bao gồm: Thêm chuyên mục nhân tin mới, Tin đề xuất, Tin đánh dấu, Tin đọc gần đây, đọc offline, LiveScore, cùng một số chức năng cài đặt như Nhận tin nổi bật, Chế độ bao đêm, Tin địa phương....

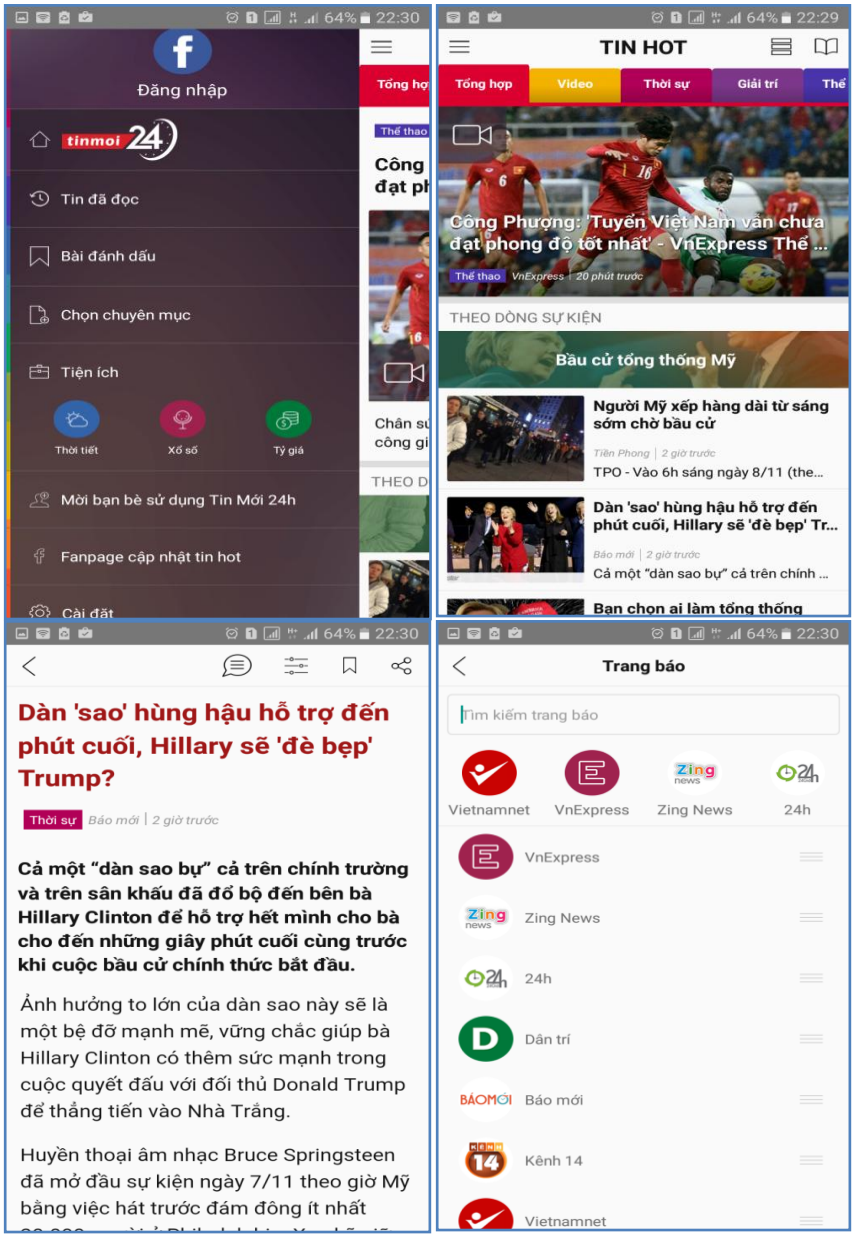
Với các màn hình giao diện của ứng dụng (xem Hình 4.1 từ trái qua phải) màn hình menu, màn hình home hiển thị tóm tắt tất cả các tin tức mới, màn hình hiển thị chi tiết cho một tin tức.



Hình 4.1 - Giao diện của BaoMoi

TinMoi24h[8] cũng là một trong số các ứng dụng được xây dựng công phu lấy nguồn tin từ trang http://tinmoi24.com/**TinMoi24** giúp chúng ta không tốn quá nhiều thời gian và dung lượng vào những trình duyệt thông thường trên điện thoại. Giao diện thân thiện giúp cho người dùng có thể thoải mái đọc tin tức và sử dụng một số chức năng cùng với tiện ích cung cấp kèm của ứng dụng như: đọc các tin đã

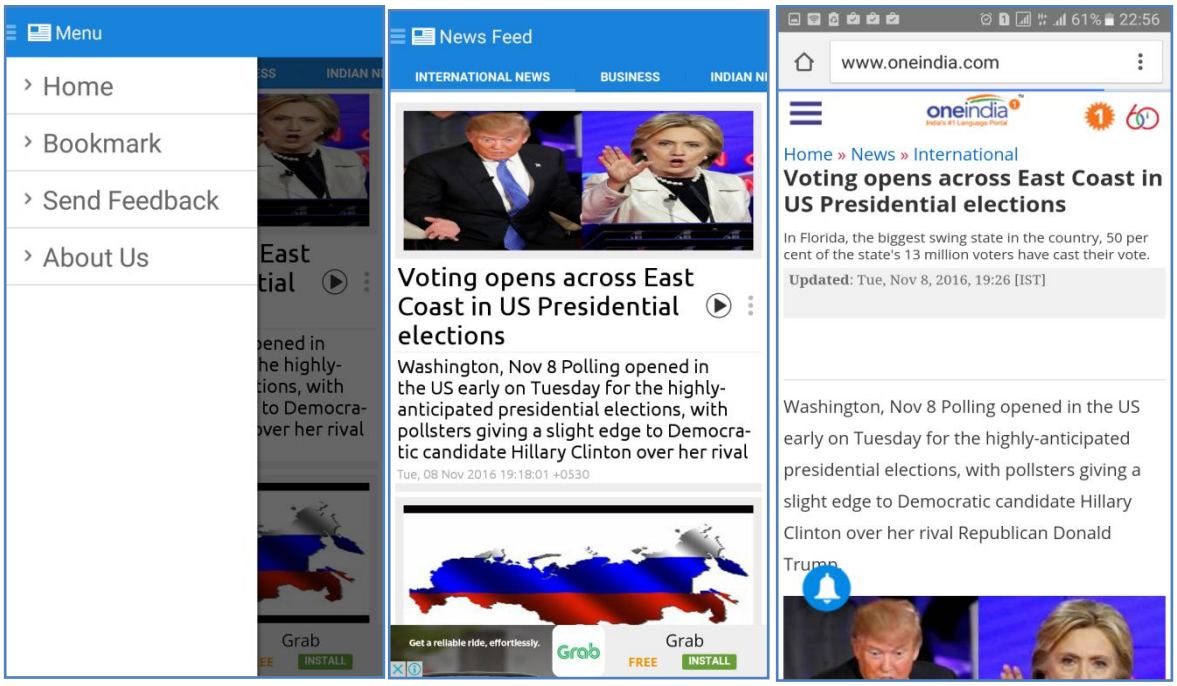
đọc, đánh dấu các bài tin, Chọn chuyên mục cần đọc tin mới. Các tiện ích đi kèm như: xem dự báo thời tiết, xem kết quả xổ số, tỷ giá, mời bạn bè sử dụng Tin Mới 24h... Một số giao diện của ứng dụng (xem Hình 4.2 từ trái qua phải từ trên xuống dưới) đầu tiên là màn hình menu, màn hình trang chủ, màn hình chi tiết nội dung tin tức và sau cùng là màn hình tìm kiếm.



Hình 4.2- Giao diện của TinMoi24h

News Feed[2] là một ứng dụng khá phổ biến trên thị trường hiện nay News Feed lấy nguồn tin từ http://www.oneindia.com.cho phép chúng ta đọc tin tức

nhanh nhất và hiệu quả nhất. Ứng dụng giúp chúng ta xem tin tức nhanh nhất, đơn giản, thuận tiện và đầy thú vị với giao diện slide show trình chiếu thú vị.Với các chứ năng đơn giản gọn nhẹ tiết kiệm được dung lượng của máy từ trái qua phải của Hình 4.3 là các màn hình giao diện Menu với các chức đơn giản như Home trang chủ load tất cả các tin nổi bật một cách tóm tắt, tiếp nữa là chức năng đánh dấu bài tin, và chức năng phản hồi thông tin bên phải cùng là giao diện chi tiết cho nội dung của một tin tức.



Hình 4.3- Giao diện của News Feed

4.2. Phân tích thiết kế hệ thống đọc tin tự động từ website

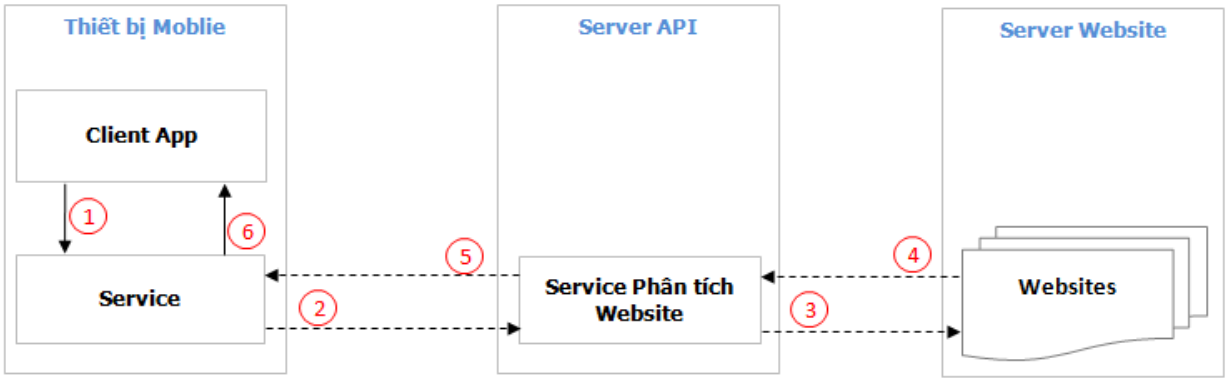
Việc khai thác, thu thập và chia sẻ thông tin trong tất cả các ngành nghề, lĩnh vực đóng một vai trò vô cùng quan trọng. Mạng Internet là một kho dữ liệu khổng lồ, vậy làm sao ta có thể nắm bắt thông tin nhanh chóng nhất, cập nhật mới nhất mà không phải tốn nhiều thời gian tìm kiếm thông tin, lướt từng website, từng mục tin để đọc và tìm kiếm.

- **Thiết kế tổng quan hệ thống**

Trên cơ sở nắm bắt nhu cầu người dùng một bài toán được phát biểu kèm theo giải pháp công nghệ xây ứng dụng bóc tách thông tin, trích xuất thông tin một cách tự động từ các website, sau đó thông báo nhắc cho người dùng biết đã có tin

tức mới giúp người sử dụng nắm bắt thông tin một cách súc tích, nhanh chóng, tiết kiệm thời gian nhất.Ở đây có hai giải pháp được đưa ra cho thiết kế tổng quan hệ thống như sau:

Giải pháp 1: Mô hình Client - Server - Website



Hình 4.4- Mô hình Client-Server-Website

Thiết kế mô hình tổng thể như Hình 4.4 luồng dữ liệu sẽ hoạt động như sau:

1: Client sẽ gọi 1 Service yêu cầu quét thu thập dữ liệu mới tới Server API,

2: Service gửi đi yêu cầu quét thu thập dữ liệu mới tới Server API

3: Server API nhận yêu cầu tiến hành phân tích RSS và lưu dữ liệu tin tức mới của Website.

4: Website cung cấp file RSS cho phép Server đọc dữ liệu từ RSS.

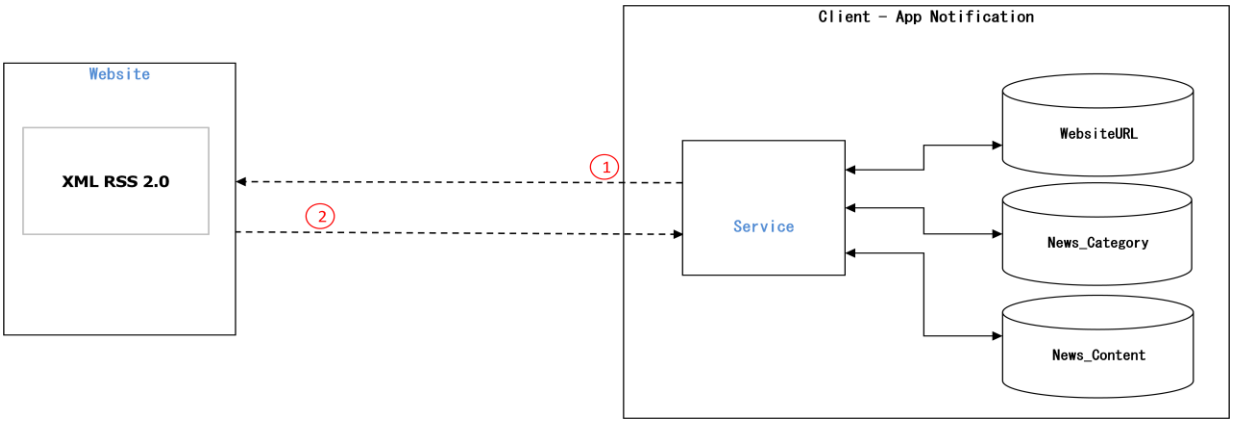
5: Server API trả về dữ liệu mới cho Client.

6: Client tiến hành hiển thị thông tin, và đẩy thông báo tự động tới giao diện người dùng.

Với giải pháp 1 thể hiện được các ưu điểm như Client và Server phân tích website hoạt động độc lập nhau từ đó sẽ giảm tải cho Client. Khi Client Offline thì Server vẫn lấy được tin mới và lưu tại server.Mô hình này có thể nâng cấp mở rộng thêm các API mới, các module mới.Số lượng Client nhỏ thì mô hình sẽ xử lý rất

nhanh. Bên cạnh những ưu điểm thì mô hình này cũng có các nhược điểm cần phải chú ý như: Với số lượng Client lớn, nhiều request có thể dẫn tới quá tải, sập hệ thống Server API.Khi sử dụng mô hình này ta phải tính tới chuyện nâng cấp Server, đầu tư thiết bị phần cứng tốn kém.Phải có phương án cân bằng tải, phân tán dữ liệu.

Giải pháp 2: Mô hình Client - Website



Hình 4.5-Mô hình Client-Website

Với thiết kế mô hình tổng thể như trên luồng dữ liệu sẽ hoạt động như sau:

1: Client sẽ gọi 1 Service yêu cầu quét thu thập dữ liệu mới tới Website, tiến hành phân tích RSS, đọc dữ liệu và lưu xuống DB Client.

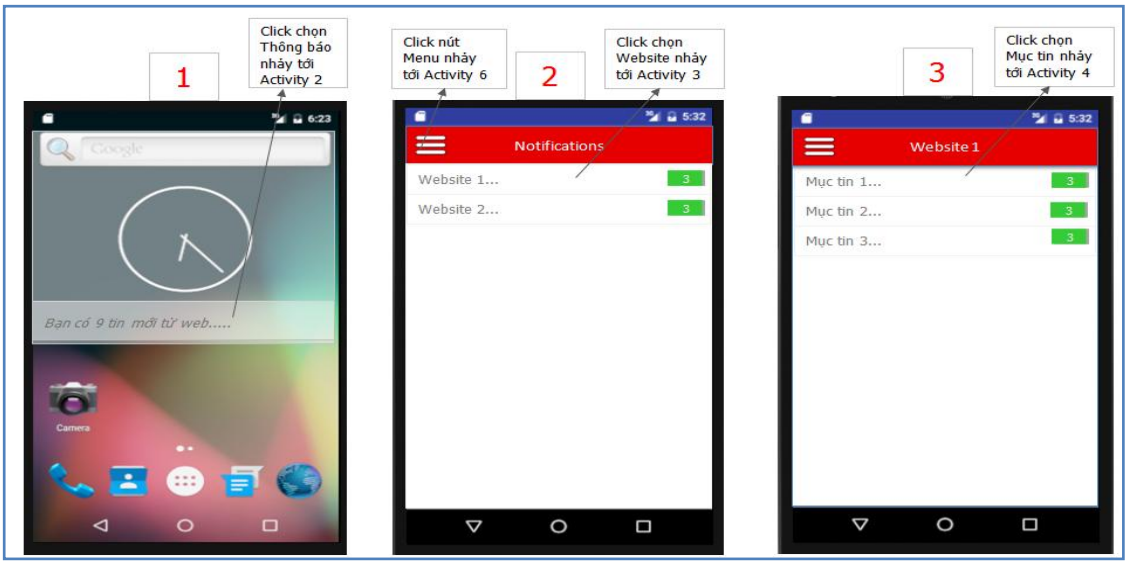
2: Website cung cấp file RSS và dữ liệu cho Client.

Trong giải pháp thứ 2 này thì ưu điểm thể hiện mạnh ở chỗ không tập trung xử lý tại một nơi mà xử lý riêng lẻ đối với từng Client nên tránh được hiện tượng quá tải.Việc nâng cấp ứng dụng có thể được thực hiện một cách thuận lợi.Tuy nhiên có nhược điểm là phụ thuộc vào Internet, 3G, offline không tự động lấy được tin mới, phụ thuộc vào phần cứng của Client.

Căn cứ vào ưu điểm và nhược điểm của 2 giải pháp trên. Thông qua việc khảo sát một số ứng dụng viết để đọc báo online, phần mềm hỗ trợ đọc tin RSS ở trên thì ứng dụng được xây dựng dựa trên giải pháp 2 là phù hợp hơn để đơn giản, dễ dùng hướng đến cộng đồng người dùng mạng cùng với thiết bị di động thông minh của mình. Ngôn ngữ lập trình được lựa chọn công nghệ Android để phát triển

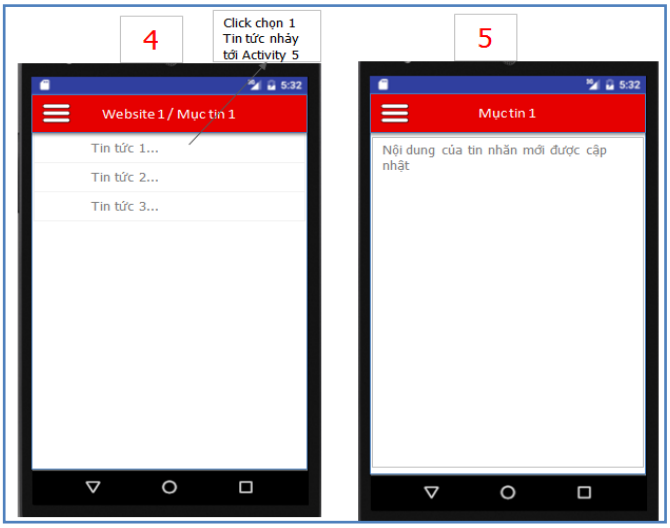
hệ thống. Cụ thể ở đây sẽ dùng Android 5.1, thư viện phân tích cú pháp website Jsoup 1.9.2, package phân tích cú pháp Xml đọc RSS là XmlPullParser để tối ưu ứng dụng

- **Thiết kế giao diện**



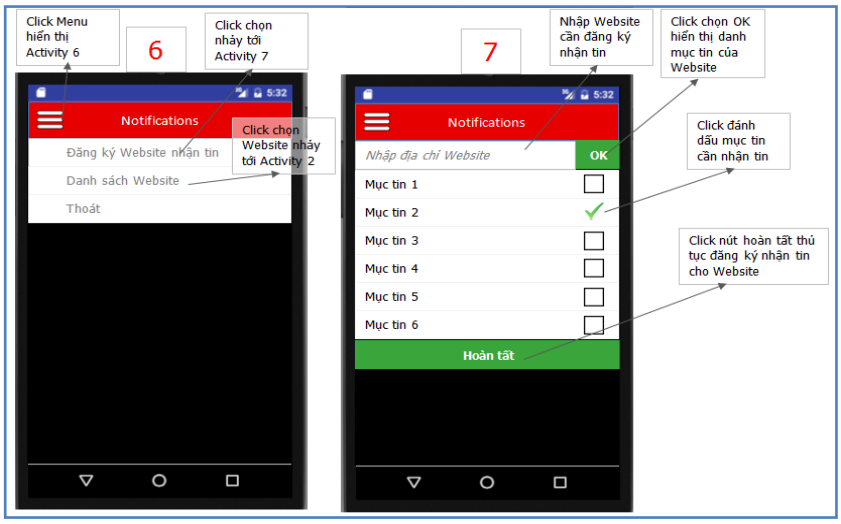
Hình 4.6-Thiết kế giao diện hiển thị Danh sách website

- (Xem Hình 4.6) là 3 màn hình giao diện được thiết kế đánh số thứ tự 1, 2, 3, trong đó thứ tự 1 là giao diện thiết kế thể hiện ứng dụng nhận được Notifi từ service hiển thị ở dạng thông báo tổng quát, thứ tự 2 là khi kích hoạt ứng dụng ban đầu load dữ liệu từ database và hiển thị danh sách website đã đăng ký nhận tin, và số lượng tin tức mới của các website được báo notification



Hình 4.7- Thiết kế giao diện hiển thị nội dung của tin tức

Hình 4.7 là 2 màn thiết kế giao diện hiển thị danh sách các tin tức và nội dung chi tiết của tin tức

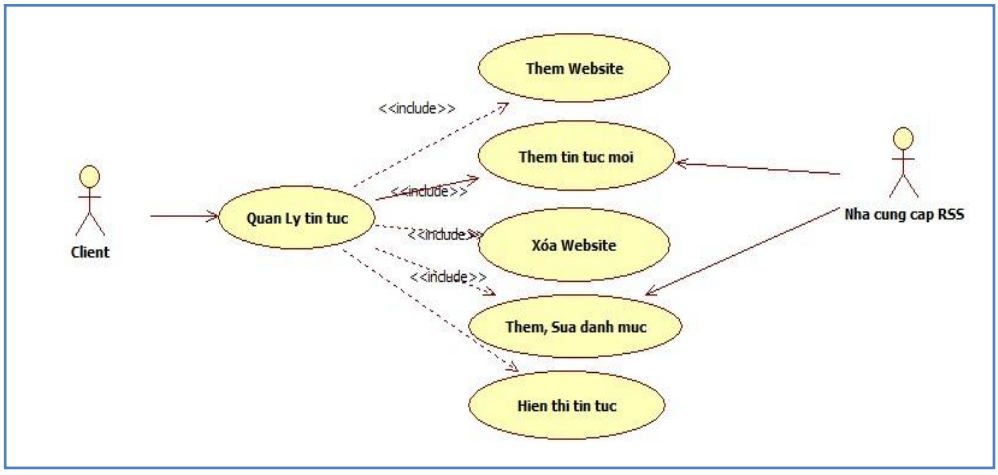


Hình 4.8- Thiết kế giao diện đăng ký website nhân tin

Hình 4.8 là 2 màn hình thiết kế giao diện cho menu truy cập các chức năng và màn hình giao diện đăng ký website nhận tin mới

- **Phân tích chức năng hệ thống**

- **Biểu đồ Use Case:** Biểu đồ User-Case thể hiện sự tương tác giữa người dùng và hệ thống. Từ đó xác định được hệ thống cần phải làm gì.



Hình 4.9 - Biểu đồ User-Case

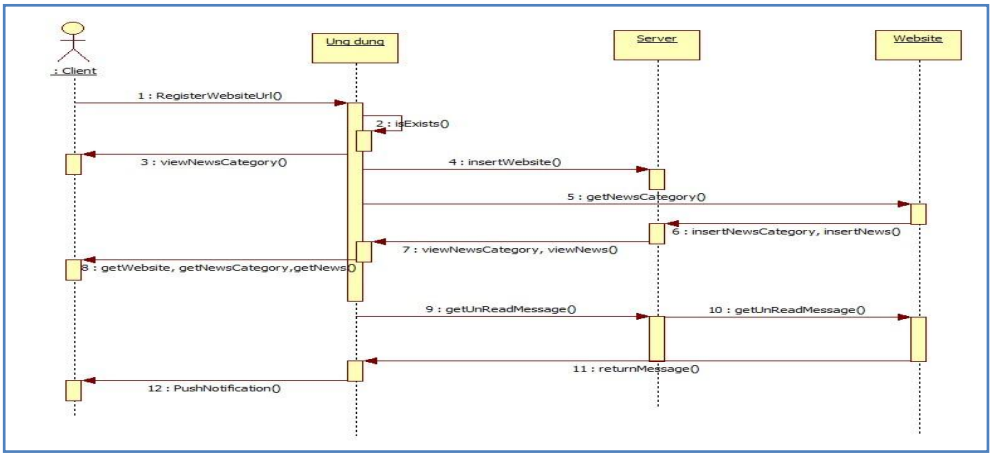
- **Đặc tả các Use-case:** Hệ gồm 1 Use - case quản lý tin tức cho phép người sử dụng khi mở ứng dụng lên quản lý tin tức bao gồm các việc

như: thêm mới dịa chỉ website muốn nhận tin tức mới, cập nhật trạng thái nhận tin cho từng danh mục tin của từng website, có thể xóa địa chỉ website không muốn nhận tin mới. Hiển thị danh sách các website đã đăng ký nhận tin, hiển thị danh sách các danh mục tin của từng website, hiển thị danh sách các tin tức tương ứng với từng danh mục tin và đọc tin tức mới nhất.

Dòng sự kiện củaUse-case này được bắt đầu thực hiện khi người dùng bật ứng dụng lên vào hệ thống và thêm địa chỉ website (kênh tin) muốn nhận tin tức, đánh dấu các danh mục nhận tin tức mới. Dòng sự kiện chính: Ban đầu hệ thống sẽ kiểm tra trong cơ sở dữ liệu xem đã tồn tại địa chỉ website nhận tin(kênh tin) nào chưa. Nếu chưa có thì hệ thống sẽ chuyển tới trang đăng ký kênh tin. Nếu có đã tồn tại trong hệ thống thì liệt kê danh sách các kênh tin đã có trong hệ thống cùng với số lượng các tin mới chưa đọc của kênh tin.Sau đó thì người dùng có thể: thêm kênh tin, xóa kênh tin, cập nhật trạng thái nhận tin mới của các danh mục tin, đọc tin...

Việc Nhập địa chỉ website nhất thiết phải có thành phần Protocol (http, https...). Nhập địa chỉ tới trang mà website cung cấp danh sách các mục được cung cấp RSS. Điều kiện tiên quyết cần thiết để ứng dụng có thể chạy tốt đó là chỉ làm việc với các trang website có cung cấp RSS. Kết quả thu được nếu Use-case thành công thì thông tin về website, danh mục tin tức, tin tức sẽ được lưu vào cơ sở dữ liệu.

- **Biểu đồ tuần tự (Sequence Diagram):** Hoạt động của hệ thống nhìn một cách tổng thể (xem Hình 4.10) gồm các hành động cơ bản như sau.



Hình 4.10 -Biểu đồ tuần tự toàn hệ thống

Đăng ký một website nhận tin: Để có thể nhận tin tức mới từ một website thì người dùng cần phải đăng ký nhận tin cho 1website đó. Người dùng chỉ cần nhập đúng đường dẫn tới trang có chứa các danh mục mà website cung cấp RSS.Ứng dụng sẽ có nhiệm vụ kiểm tra xem kênh tin vừa nhập đã có trong hệ thống ứng dụng chưa, nếu đã có thì hiển thị danh sách các danh mục tin của kênh tin cho người dùng cập nhật. Nếu chưa có sẽ tiến hành thêm mới kênh tin và danh mục tin vào trong cơ sở dữ liệu.

Hiển thị danh sách kênh tin, danh mục tin, tin tức: Ứng dụng sẽ tiến hành đọc danh sách kênh tin, danh sách danh mục tin, tin tức từ cơ sở dữ liệu đã được lưu trong hệ thống một cách thông thường, đối với việc hiển thị chi tiết một tin tức thì ứng dụng sẽ dùng URL của tin tức đó và tải về để hiện thị.

Đẩy thông điệp có tin mới: Ứng dụng có một service chạy ngầm được đặt theo thời gian chạy, cứ sau một khoảng thời gian được đặt trước service tiến hành chạy 1 lần để kiểm tra xem có tin tức nào mới được đưa lên website. Nếu phát hiện được một nội dung tin tức mới thì sẽ đẩy Notification thông báo.

- **Thiết kế cơ sở dữ liệu**

Cơ sở dữ liệu của chương trình không quá lớn, hơn nữa do ứng dụng chạy trên thiết bị di động nên hệ quản trị cơ sở dữ liệu được cung cấp bởi Android SqlLite. Sau đây là thiết kế cấu trúc các bảng dữ liệu và mô hình quan hệ giữa các bảng:

- **Thiết kế cấu trúc các bảng dữ liệu**

Bảng website_url: Lưu thông tin về địa chỉ các Website để chứa các kênh tin.

website_url				
STT	Tên trường	Kiểu dữ liệu	Ghi chú	Diễn giải
1	website_id	INTEGER	Primary Key AutoIncrement	Mã địa chỉ website
2	website_title	TEXT		Tiêu đề của website
3	website_url	TEXT	Khác rỗng	Đường dẫn chính của web

4	website_url_rss	TEXT		Đường dẫn tới trang gồm RSS
5	website_count_ news	INTEGER		Tổng số tin của web chưa đọc

Bảng 4.1: Bảng website_url (địa chỉ website)

Bảng news_category: Lưu thông tin về các danh mục được chọn để nhận tin mới.

news_category				
STT	Tên trường	Kiểu dữ liệu	Ghi chú	Diễn giải
1	Id	INTEGER	Primary Key AutoIncrement	Mã địa chỉ của category
2	website_id	INTEGER	Khác rỗng	Mã địa chỉ website
3	website_url	TEXT	Khác rỗng	Đường dẫn chính của web
4	news_category _title	TEXT		Tiêu đề của danh mục nhận tin mới
5	news_category _url	TEXT	Khác rỗng	Đường dẫn của danh mục nhận tin mới
6	news_category _rss_status	INTEGER	0: không nhận 1: nhận	Trạng thái nhận tin của danh mục
7	news_category _count_news	INTEGER	Mặc định 0	Tổng số tin chưa đọc của mục

Bảng 4.2 - Bảng news_category(Mục nhận tin mới)

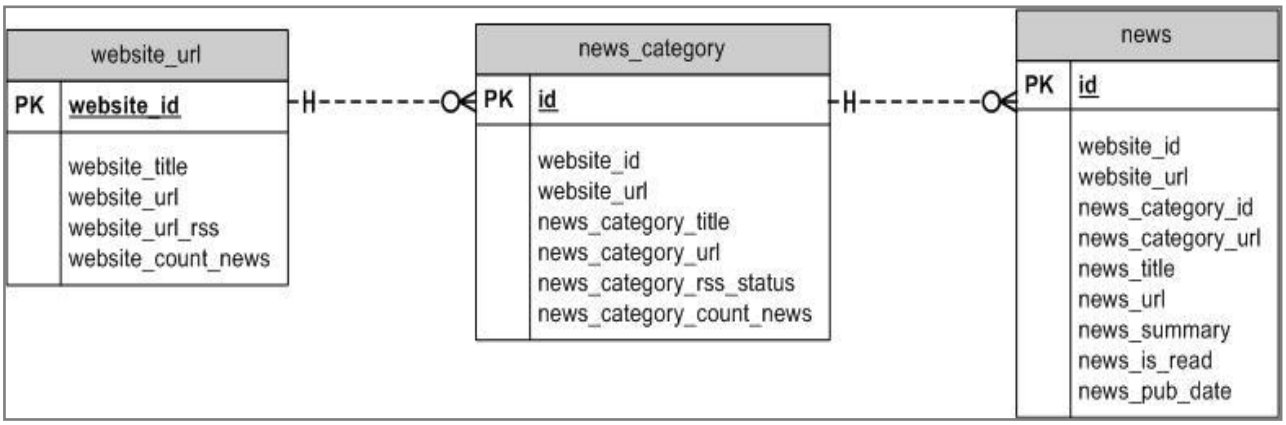
Bảng news: Lưu thông tin các tin tức của một danh mục nào đó.

News				
STT	Tên trường	Kiểu dữ liệu	Ghi chú	Diễn giải
1	Id	INTEGER	Primary Key AutoIncrement	Mã địa chỉ của tin tức
2	website_id	INTEGER	Khác rỗng	Mã địa chỉ website
3	website_url	TEXT	Khác rỗng	Đường dẫn chính của web
	news_category	INTEGER	Khác rỗng	Mã địa chỉ của danh mục

	_id			nhận tin mới
4	news_category _url	TEXT	Khác rỗng	Đường dẫn của danh mục nhận tin mới
5	news_title	TEXT		Tiêu đề ngắn gọn của tin tức
6	news_url	TEXT		Đường dẫn của danh mục nhận tin mới
7	news_summary	TEXT		Tóm tắc tin nhắn
8	news_is_read	INTEGER	Mặc định 0 0: chưa đọc 1: đã được đọc	Trạng thái của tin tức xem đã được đọc chưa
9	news_pub_date	TEXT		Ngày giờ của tin tức

Bảng 4.3 - Bảng news (Các tin tức mới nhận được)

- **Mô hình quan hệ (ER)**



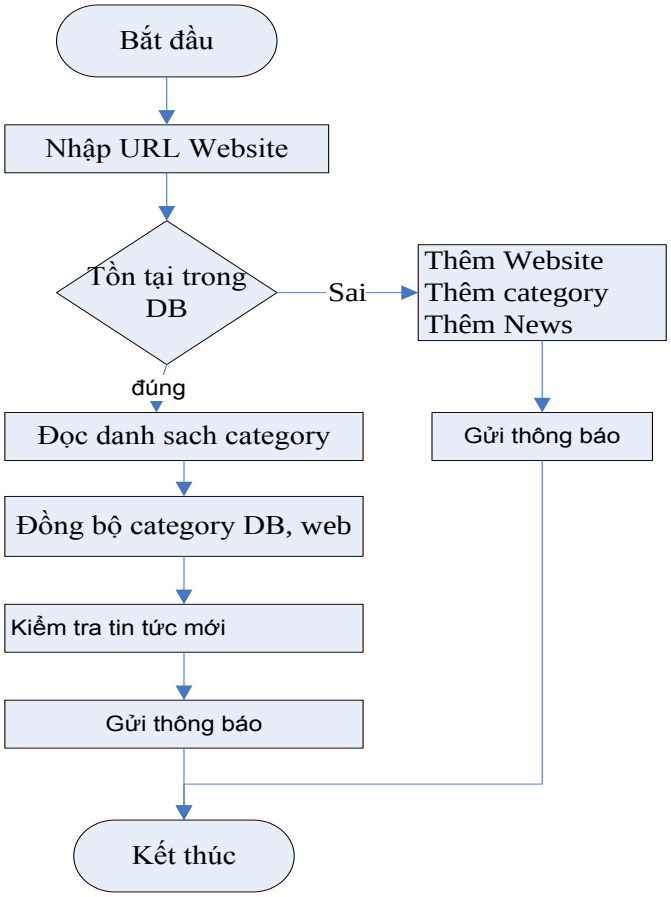
Hình 4.11 - Mô hình quan hệ thực thể(ER)

- **Một số thuật toán được cài đặt**

Thuật toán 1: kiểm tra xem một địa chỉ website đã tồn tại trong cơ sở dữ liệu, nếu đã tồn tại thì lấy danh sách các danh mục tin tức có hỗ trợ RSS của web đó lên cho người dùng cập nhật. Trong quá trình đó cũng thực hiển kiểm tra trên thực tế các danh mục đó còn tồn tại trên Website thực tế không. Nếu có mục nào mới được thêm trên website thì bổ xung vào cơ sở dữ liệu, sau đó tiến hành cập nhật trạng thái đọc tin và tổng số tin của các danh mục.

Nếu kiểm tra website nhập vào chưa có trong hệ thống thì sẽ phân tích để lấy danh sách các danh mục nhận tin trên mã HTML của website. Sau khi lấy được danh sách các danh mục thì thêm mới địa chỉ website và các danh mục này vào cơ sở dữ liệu. Đồng thời tiến hành đọc các tin tức và đưa vào cơ sở dữ liệu với các danh mục được người đánh dấu nhận tin mới.

Sơ đồ khối giải thuật



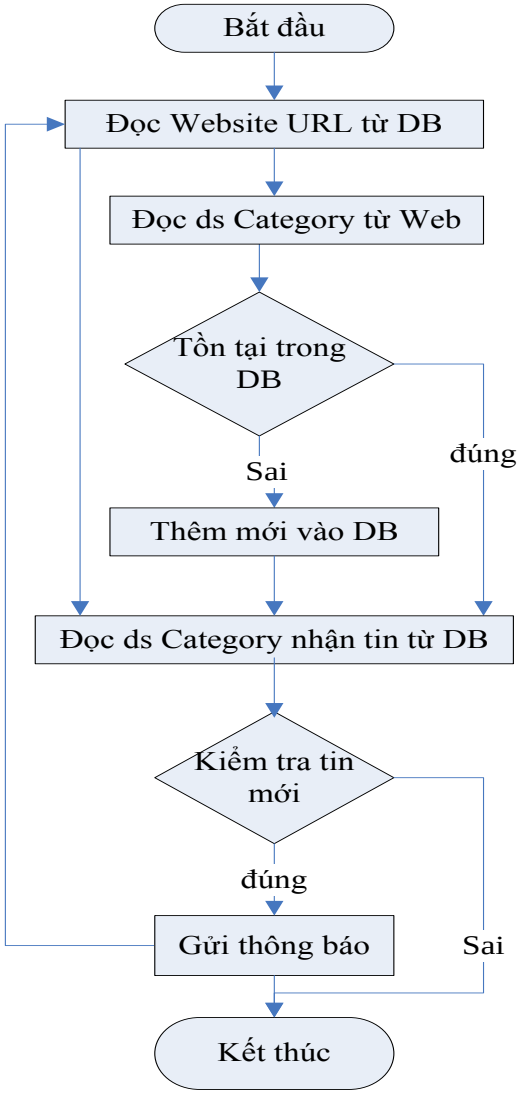
Hình 4.12 - Sơ đồ giải thuật kiểm tra sự tồn tại website và danh mục tin tức

Thuật toán 2: Kiểm tra mỗi khi website cập nhật tin tức mới, báo có tin mới.

Mỗi khi trang website được đang ký nhận tin mới mà cập nhật đưa một nội dung tin tức mới lên website thì có một service chạy ngầm dưới máy điện thoại của Client sẽ thực hiện nhiệm vụ đọc và phát hiện tin tức mới. Nếu có nội dung mới được đưa lên thì sẽ tự động báo Notification cho người dùng.

Service sẽ thực hiện đọc lần lượt từng website đã được đăng ký nhận tin mới trong hệ thống máy Client và đọc danh sách các mục tin được đánh dấu nhận tin mới. Đồng thời tiến hành đọc các danh mục tin từ website, trong số các mục tin tức này có cái nào thuộc tập các danh mục được đánh dấu nhận tin lấy lên từ cơ sở dữ liệu thì tiến hành đọc các tin tức mới của mục này đem so sánh với các tin tức của mục tương ứng đã có trong cơ sở dữ liệu. Nếu là tin tức mới thì ghi nhận vào cơ sở dữ liệu và gửi một thông điệp báo có tin tức mới nhận được.

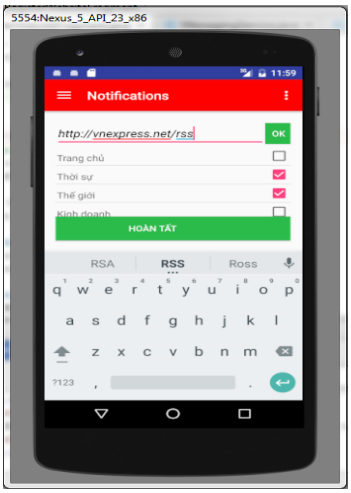
Sơ đồ giải thuật như sau



Hình 4.13 - Sơ đồ giải thuật kiểm tra sự tồn tại tin tức

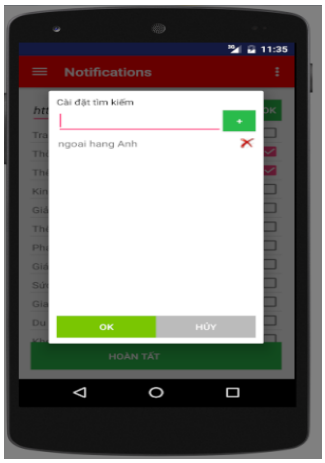
4.3. Các chức năng cơ bản của ứng dụng

Chức năng đăng ký: Xem Hình 4.14 Chức năng cho phép người dùng nhập vào địa chỉ của website hoặc mục muốn nhận tin mới. Người dùng nhập địa chỉ website muốn nhận tin rồi nhấn nút OK.Hệ thống sẽ tự động kiểm tra tính hợp lệ của đường link địa chỉ vừa được nhập vào, kiểm tra sự tồn tại của đường link trong database. Sau đó sẽ load toàn bộ dữ liệu về danh mục các mục tin sẽ nhận lên một listview với tùy biến cho phép người dùng tích chọn sẽ nhận tin hoặc không tích chọn nhận tin.



Hình 4.14 -Chức năng đăng ký website nhận tin mới

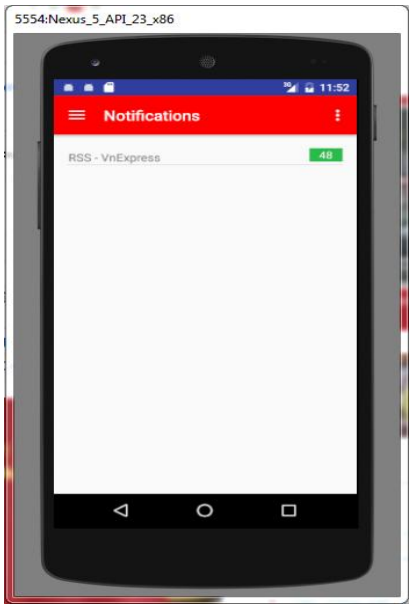
Để tùy biến cài đặt điều kiện nhận tin mới tại các danh mục tin người dùng kích vào dòng danh mục tin tức, một cửa sổ Popup sẽ được hiển thị cho phép nhận điều kiện tùy biến nhận tin xem Hình 4.15



Hình 4.15 -Chức năngtùy biến cài đặt nhận tin

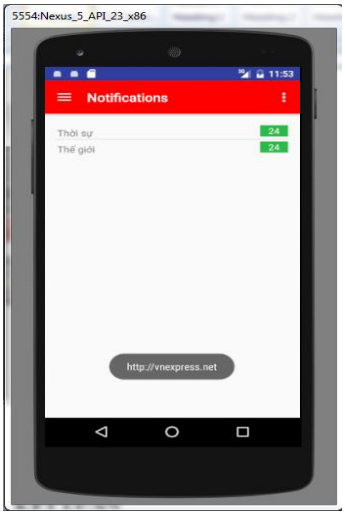
Sau cùng người dùng nhấn nút hoàn tất để kết thúc chức năng đăng ký nhận tin, lúc này ứng sẽ tiến hành lưu các thông tin về địa chỉ website về mục đánh dấu nhận, về các tin tức liên quan ... xuống database.

Chức năng hiển thị danh sách website đã đăng ký: Hình 4.16 là giao diện chức năng hiển thị danh sách các website đã đăng ký nhận tin được load lên từ database khi có lời gọi hiển thị, kèm theo đó là tổng số tin tức mới chưa đọc. Khi kích chọn vào một website hệ thống sẽ chuyển hướng sang màn hình chức năng tiếp theo là hiển thị danh sách các mục tin tức của 1 website.



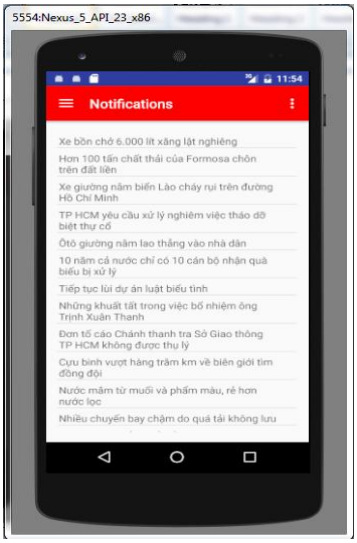
Hình 4.16 -Chức năng danh sách website nhận tin mới

Chức năng hiển thị các danh mục tin tức của 1 website: Tại chức năng này (Xem Hình 4.17) hệ thống sẽ hiển thị tất cả các danh mục tin trên một lưới hiển thị mà người dùng đã tích chọn nhận tin mới khi ở chức năng đăng ký, kèm bên phải là tổng số tin tức chưa đọc của danh mục tin này. Khi kích chọn vào 1 mục tin nào đó thì hệ thống cũng chuyển hướng sang màn hình chức năng hiển thị tóm tắt các tin tức mới trong một danh mục tin.



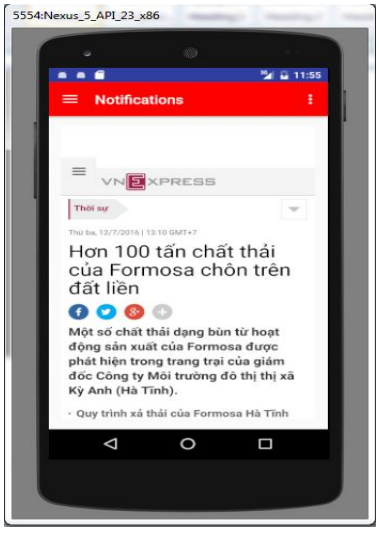
Hình 4.17 -Chức năng danh mục của website nhận tin mới

Chức năng hiển thị tóm tắt nội dung các tin tức: (Xem Hình 4.18) Chức năng này thực hiện việc hiển thị tóm tắt tin tức bao gồm tiêu đề, tóm lược chính... Khi người dùng kích chọn vào 1 tin tức thì hệ thống sẽ chuyển hướng sang màn hình hiển thị chi tiết của tin tức giúp người dùng đọc được dễ dàng.



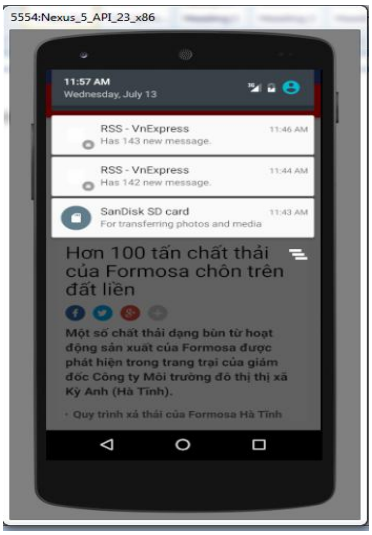
Hình 4.18 -Chức năng danh sách tin mới

Chức năng hiển thị chi tiết tin tức: (Xem Hình 4.19) Chức năng này hiển thị chi tiết nội dung của tin tức thông qua web Browers giúp người dùng có thể đọc toàn bộ tin tức.



Hình 4.19 -Chức năng danh nội dung tin tức mới

Chức năng tự động thông báo có tin tức mới: Hệ thống sử dụng một services chạy ngầm phía dưới tự động quét file RSS của các website từ đó lọc, phát hiện ra các tin tức mới được cập nhật trên website, đến số lượng các tin tức mới của từng website. Sự dụng công nghệ push notification của Android để thực hiện báo cho người dùng biết. (Xem Hình 4.20) là giao diện đẩy tin thông báo, người dùng muốn đọc các tin tức mới thì kích chọn vào message một danh sách các website có tin mới sẽ được hiển thị cùng với các tin tức mới.



Hình 4.20-Chức năng tự động thông báo tin mới

4.4. Kết quả thực nghiệm và nhận xét

- **Kết quả thực nghiệm:** Ứng dụng phân tích và lấy tin tự động từ website đã áp dụng thành công các kỹ thuật phân tích DOM, SAX cấu trúc file XML, RSS, HTML. Cụ thể như sau:
  - Áp dụng thành công công cụ Jsoup với kỹ thuật phân tích DOM để phân tích mã nguồn HTML của website để lấy được danh mục các mục tin có cung cấp file RSS.
  - Áp dụng thành công kỹ thuật phân tích SAX với XMLPullParser trong Android để phân tích cú pháp và đọc dữ liệu của RSS 2.0.
  - Phát triển được tính năng tự động quét phát hiện các tin tức mới sau đó tự động gửi thông báo cho người dùng.
  - Ứng dụng có thể phân tích và lấy tin tốt với hầu hết các website có cung cấp RSS
- **Nhận xét:** Đối với mỗi bài toán phân tích lấy tin tự động từ website cụ thể cần phải cân nhắc xem xét thực trạng bài toán, đánh giá được qui mô, độ phức tạp, các yêu cầu về chức năng hệ thống để có thể đưa ra lựa chọn giải pháp, lựa chọn công nghệ cho phù hợp.

Trong trường hợp cụ thể thực nghiệm của luận văn này với các website có file nguồn mã Html, Xml vừa và nhỏ sử dụng kỹ thuật phân tích DOM là rất tốt vì toàn bộ mã nguồn sẽ được tải vào bộ nhớ để bộ phân tích DOM thực hiện một lần duy nhất tạo lên cây DOM, việc truy cập ngẫu nhiên tới các các phần tử của cây DOM là thuận tiện, nhanh chóng, hiệu suất xử lý dữ liệu cao.

Ngược lại các website thường có tuần suất cung cấp tin mới ở file RSS cao, số lượng tin mới nhiều nên việc áp dụng kỹ thuật phân tích SAX để xử lý một nhiệm vụ hiện hành cần thực thi tại một thời điểm trên bộ nhớ là rất tốt và cho hiệu quả cao, giảm việc yêu cầu tài nguyên của thiết bị phần cứng.

Việc áp dụng các công cụ Jsoup và thư viện XmlPullParser trong Android với trường hợp cụ thể này để phù hợp là thực sự cần thiết.

KẾT LUẬN

Luận văn tìm hiểu và trình bàycác kỹ thuật phân tích và lấy tin tự động từ website với các kết quả chính đạt được là:

- ***Hệ thống lại các kỹ thuật phân tích cấu trúc của website*** để có thể lấy được các tin tức một cách tự động từ các website đó.
- ***Phát triển và xây dựng một ứng dụng di động*** trên nền Android có khả năng tự động cập nhật và thông báo đến cho người dùng các tin tức mới tùy theo cài đặt.

Các kỹ thuật áp dụng để phân tích và để lấy tin tự động từ website được trình bày trong luận văn gồm có XML, RSS 2.0, DOM, SAX, XmlPullParser, Jsoup:

- ***XML*** là ngôn ngữ đánh dấu với mục đích dùng chung, có khả năng mô tả nhiều loại dữ liệu khác nhau với đặc điểm đa năng thân thiện với các giao thức Internet, là nền tảng để phát triển nhiều ngôn ngữ khác có ứng dụng cao đặc biệt là RSS.
- ***RSS 2.0*** là sản phẩm được tạo thành từ cấu trúc XML. RSS là dịch vụ cung cấp thông tin thực sự đơn giản dùng trong việc chia sẻ tin tức web. RSS là một danh sách các đối tượng chính là các mẩu tin được miêu tả gồm có: tiêu đề, nội dung tóm lược, một liên kết đến trang chính, ngày tháng, tác giả...Để đọc được các dữ liệu cung cấp từ file RSS của các website thì Java API có cung cấp Các kỹ thuật phân tích file XML (RSS) DOM, SAX và kỹ thuật phân tích RSS trong Android với package XmlPullParser để đọc file Xml RSS.
- ***DOM*** là kỹ thuật tiếp cận dữ liệu XML mà trong đó tất cả các thành phần tạo nên XML được xem là đối tượng, được xem là các "node". Khi parser đọc XML thành đối tượng DOM trên bộ nhớ chúng ta sẽ có một cấu trúc cây với các đối tượng là các node, giữa các node này có mối quan hệ phân tầng cha-con.Dựa vào các đối tượng này mà chúng ta có thể trích xuất thông tin, thay đổi thông tin thông qua các hàm mà đối tượng cung cấp.

- ***SAX*** là một phương pháp kỹ thuật phân tích dữ liệu XML dựa vào sự kiện trong quá trình đọc file XML từ trên xuống dưới. Vì thế để sử dụng phương pháp này các ngôn ngữ cần định nghĩa ra các sự kiện và các hàm tương ứng với sự kiện. SAX không ghi nhớ XML như một cấu trúc cây trên bộ nhớ mà chỉ xây dựng cấu trúc của nhiệm vụ đang thi hành trên bộ nhớ dó đó SAX sẽ thực hiện nhanh hơn và ít tốn tài nguyên hơn so với DOM. Bên cạnh các kỹ thuật này thì Android còn cung cấp một kỹ thuật XML Pull Parser. và 1 tool của java cũng khá mạnh đó là Jsoup.
- ***XmlPullParser*** làm việc tương tự như StAX nó cho phép mã ứng dụng "kéo" hoặc tìm kiếm các sự kiện từ trình phân tích, trái ngược với SAX tự động đẩy các sự kiện cho trình xử lý. XmlPullParser cũng thể hiện được ưu điểm đơn giản trong sử dụng, ít tốn bộ nhớ và xử lý nhanh hơn DOM và SAX.
- ***Jsoup*** là một thư viện được sử dụng để phân tích tài liệu HTML, được cung cấp các API dùng để lấy dữ liệu và thao tác dữ liệu từ URL hoặc từ file HTML. Sử dụng các phương pháp DOM, CSS, JQuery để lấy dữ liệu và thao thác với dữ liệu. Jsoup được thiết kế để có thể làm việc được với tất cả các phiên bản HTML trên thực tế, từ cơ bản và xác nhận tới không hợp lệ tag-soup, Jsoup tạo thành một cây phân tích phù hợp.

Các ứng dụng di động ngày nay được xây dựng rộng khắp bởi sự phổ biến của các thiết bị di động cầm tay và Internet ngày nay. Áp dụng các kỹ thuật phân tích cấu trúc của website để xây dựng các ứng dụng cập nhật tin tức một cách tùy biến theo yêu cầu của người dùng sẽ được phát triển rộng trong thời gian tới. Luận văn đã hệ thống lại phần lý thuyết của chủ đề này và xây dựng một ứng dụng cập nhật tin tức có thể áp dụng trong thực tế và vẫn còn có thể cải tiến để trở thành một ứng dụng cập nhật tin tức hữu dụng cho người dùng.

TÀI LIỆU THAM KHẢO

Tiếng Việt

[1]. *DOM và SAX*, http://bodua.vn/knowledge/La%CC%80m-vie%CC%A3c-vo%CC%81i-XML/DOM-va%CC%80-SAX_72.html.

[2]. *NewsFeed*, https://play.google.com/store/apps/details?id=com.aaravmedia.newsfeed.

[3]. *Phương pháp Interface DOM phân tích RSS 2.0*,http://o7planning.org/vi/10109/phan-tich-xml-bang-cach-su-dung-mo-hinh-dom-trong-java#a6435.

[4]. *Phương pháp Interface SAX phân tích RSS2.0*, http://www.bodua.vn/knowledge/Java-Server-Pages/Du%CC%80ng-SAX-do%CC%A3c-XML_54.html.

[5]. *Phương pháp phân tích HTML Jsoup*, http://o7planning.org/vi/10399/huong-dan-su-dung-java-jsoup-phan-tich-html.

[6]. *RSS 2.0*, https://www.ibm.com/developerworks/vn/library/webservices/201301/x-rss20/.

[7]. *Ứng dụng đọc báo mới*, https://play.google.com/store/apps/details?id=com.epi&hl=vi.

[8]. *Ứng dụng đọc báo Tinmoi24h*, https://play.google.com/store/apps/details?id=mobi.fiveplay.tinmoi24h&hl=vi.

[9]. *Ứng dụng đọc tin tự động Việt Báo*, http://vietbao.vn/Vi-tinh-Vien-thong/RSS-Ung-dung-tuyet-voi-nhat/55135693/217/.

[10]. *XML*,https://www.ibm.com/developerworks/vn/edu/xmlintro/.

[11]. *XML structure*, https://www.ibm.com/developerworks/vn/library/12/x-androidxml/#listing13.

Tiếng Anh

[12]. *Development Org.XmlPull.V1.XmlPullParser NameSpace*, https://developer.xamarin.com/api/type/Org.XmlPull.V1.XmlPullParser/.

[13]. Gabe Beged-Dov, *JFinity Systems LLC... RDF Site Summary (RSS 1.0)*, http://web.resource.org/rss/1.0/spec#s9.

[14]. *Json*, http://www.w3schools.com/js/js_json_intro.asp.

[15]. *RSS 2.0 Specification*, http://cyber.law.harvard.edu/rss/rss.html#ltcategorygtSubelementOfLtitemgt.

[16]. *RSS2.0 XML Pull Parsing*,http://www.extreme.indiana.edu/xmlpull-website/index.shtml.

[17]. *Techniques for feedback*, http://www2.le.ac.uk/offices/red/rd/career-development/research-staff/mentoring/feedback.

[18]. *Techniques for Search engine*, http://www.htmlgoodies.com/beyond/seo/article.php/3837311/Search-Engine-Optimization-SEO-Advanced-Techniques.htm.