LỜI CẢM ƠN

Em xin bày tỏ lời cảm ơn sâu sắc nhất tới PGS.TS. Đỗ Năng Toàn, thầy đã tận tình hướng dẫn và giúp đỡ em rất nhiều trong quá trình làm tốt nghiệp để tìm hiểu, nghiên cứu đề tài "Tìm hiểu bài toán khai phá dữ liệu văn bản" được giao để em có thể hoàn thành tốt đề tài tốt nghiệp của mình.

Em xin chân thành cảm ơn sự dạy bảo của các thầy cô giáo khoa CNTT – Trường ĐHDLHP đã trang bị cho em những kiến thức cơ bản để em có thể hoàn thành tốt đề tài tốt nghiệp.

Tuy có nhiều cố gắng trong quá trình làm đề tài nhưng em không tránh khỏi sai sót. Em rất mong thầy cô giáo chỉ dẫn, đóng góp cho em những ý kiến quý báu để giúp em hoàn thiện hơn đề tài của mình cũng như là để phát triển mở rộng đề tài sau này.

Em xin chân thành cảm ơn!

Hải Phòng, ngày tháng năm

Sinh viên

Bùi Thị Mây.

MỤC LỤC

LỜI NÓI ĐẦU

Trong thời đại ngày nay, thông tin là nhu cầu thiết yếu đối với mọi người trên mọi lĩnh vực. Mỗi phút trôi qua hàng triệu triệu trang web được đẩy lên nhằm làm giàu nguồn tài nguyên vô tận này. Vấn đề đặt ra là làm sao ta có thể nắm bắt, cập nhật, chia sẻ thông tin một cách tổng quát, nhanh chóng và dễ dàng trong một khối lượng thông tin khổng lồ như vậy. Do đó đòi hỏi phải khai phá nguồn dữ liệu đó để lấy được những thông tin có ích một cách tự động. Trên thế giới hiện nay, rất nhiều website **cung cấp tập tin RSS** để chia sẻ và cập nhật thông tin một cách dễ dàng và nhanh chóng. Một số website **hỗ trợ đọc tin RSS** như: Google Reader, Yahoo,…và một số phần mềm như: RSSReader, FeedDemon. Còn hiện tại ở Việt Nam, có một số phần mềm hỗ trợ

đọc tin như: Vietspider, iCA và website hỗ trợ đọc tin RSS trực tuyến thì chưa nhiều.

Chính vì vậy đề tài **"Tìm hiểu bài toán khai phá dữ liệu văn bản"** được đưa ra nhằm ứng dụng khai phá dữ liệu vào việc xây dựng hệ thống thu thập tin tức từ nhiều nguồn website khác giúp cho người dùng có thể nắm bắt thông tin một cách dễ dàng và tiết kiệm thời gian.

Nội dung đề tài gồm các phần chính sau:

Chương 1 – Khái quát về khai phá dữ liệu

Tìm hiểu khái niệm, quá trình và các bài toán trong khai phá dữ liệu.

Chương 2 - Khai phá dữ liệu trong lấy tin tự động

Nội dung của chương 2 là tìm hiểu về lấy tin tự động và ứng dụng khai phá dữ liệu trong lấy tin tự động (tìm hiểu ngôn ngữ XML và công nghệ RSS)

Chương 3 – Phân tích thiết kế chương trình

Nội dung của chương 3 là quá trình khảo sát, phân tích và thiết kế chi tiết cho chương trình hỗ trợ đọc tin RSS.

Chương 4 – Xây dựng chương trình

Nêu ra các lớp, phương thức cơ bản để xây dựng website hỗ trợ đọc tin RSS. Và cuối cùng là đưa một số màn hình giao diện đạt được.

Kết luận và phương hướng phát triển

Phần cuối cùng này sẽ là những kết luận về kết quả đạt được và các ưu nhược điểm của đề tài. Bên cạnh đó, phần cùng này cũng nêu ra các phương hướng để có thể tiếp tục phát triển đề tài trong tương lai nhằm ngày một hoàn thiện và đáp ứng được nhu cầu ngày một cao của người sử dụng.

CHƯƠNG 1 – KHÁI QUÁT VỀ KHAI PHÁ DỮ LIỆU

Nội dung chương 1 gồm :

- Phần 1: Khái niệm khai phá dữ liệu
- Phần 2: Quá trình khai phá dữ liệu
- Phần 3: Các bài toán thông dụng trong khai phá dữ liệu.

1.1 Khái niệm khai phá dữ liệu

- Khai phá dữ liệu – Data mining: Là một bước của tiến trình khai phá tri thức (KDD)
- KDD - Knowledge discovery in database: Thuật ngữ tổng quát gồm các bước như tiền xử lý, KPDL, hậu xử lý.

1.2. Quá trình khai phá dữ liệu

a. Tìm hiểu nghiệp vụ và dữ liệu

- Nhà tư vấn nghiên cứu kiến thức về lĩnh vực áp dụng, bao gồm các tri thức cấu trúc về hệ thống, các nguồn dữ liệu hiện hữu, ý nghĩa, vai trò và tầm quan trọng của các thực thể dữ liệu.

b. Chuẩn bị dữ liệu

- Giai đoạn này sử dụng các kỹ thuật tiền xử lý để biến đổi và cải thiện chất lượng dữ lượng dữ liệu để thích hợp với những yêu cầu của các giải thuật học:
- Các giải thuật tiền xử lý bao gồm:
  - ✓ Xử lý dữ liệu bị thiếu / mất: Các dữ liệu bị thiếu sẽ được thay thế bởi các giá trị thích hợp.
  - ✓ Khử sự trùng lặp: Các đối tượng dữ liệu trùng lặp sẽ bị loại bỏ đi. Kỹ thuật này không được sử dụng cho các tác vụ có quan tâm đến phân bổ dữ liệu.
  - ✓ Giảm nhiễu: Nhiễu và các đối tượng tách rời khỏi phân bố chung sẽ bị loại đi khỏi dữ liệu.
  - ✓ Chuẩn hóa: Miền giá trị của dữ liệu sẽ được chuẩn hóa.
  - ✓ Rời rạc hóa: Các dữ liệu số sẽ được biến đổi ra các giá trị rời rạc.
  - ✓ Rút trích và xây dựng đặc trưng mới từ các thuộc tính đã có.
  - ✓ Giảm chiều: Các thuộc tính chứa ít thông tin sẽ được loại bỏ bớt.

c. Mô hình hóa dữ liệu

- Các giải thuật học sử dụng các dữ liệu đã được tiền xử lý trong giai đoạn hai để tìm kiếm các quy tắc ẩn và chưa biết.

d. Hậu xử lý và đánh giá mô hình

- Dự trên đánh giá của người dùng sau khi kiểm tra trên các tập thử, các mô hình sẽ được tinh chỉnh và kết hợp lại nếu cần. Chỉ các mô hình đạt được mức yêu cầu cơ bản của người dùng mới đưa ra triển khai trong thực tế.
- Trong giai đoạn này, các kết quả được biến đổi từ dạng học thuật sang dạng phù hợp với nghiệp vụ và dễ hiểu hơn cho người dùng.

e. Triển khai tri thức

- Các mô hình được đưa vào hệ thống thông tin thực tế dưới dạng các môđun hỗ trợ việc đưa ra quyết định.
- Mối quan hệ chặt chẽ giữa các giai đoạn trong quá trình KPDL là rất quan trọng cho việc nghiên cứu trong KPDL. Một giải thuật trong KPDL không thể được phát triển độc lập, không quan tâm đến bối cảnh áp dụng mà thường được xây dựng để giải quyết một mục tiêu cụ thể.
- Quá trình này có thể được lặp lại nhiều lần một hay nhiều giai đoạn dựa trên phản hồi từ kết quả của các giai đoạn sau.

1.3. Các bài toán thông dụng trong KPDL

- Phân lớp (Classification): Với một tập các dữ liệu huấn luyện cho trước và sự huấn luyện của con người,các giải thuật phân loại sẽ học ra bộ phân loại (classifier) dùng để phân các dữ liệu mới vào trong những *lớp* (còn gọi là *loại*) đã được xác định trước.
- Dự đoán (Prediction) sẽ học ra các bộ dự đoán. Khi có dữ liệu mới đến, bộ dự đoán sẽ dựa trên thông tin đang có để đưa ra một giá trị số học cho hàm cần dự đoán.
- Tìm luật liên kết (Association Rule) tìm kiếm các mối liên kết giữa các thành phần từ dữ liệu.

- Phân cụm (Clustering) sẽ nhóm các đối tượng dữ liệu có tính chất giống nhau vào cùng một nhóm.

CHƯƠNG 2

KHAI PHÁ DỮ LIỆU TRONG LẤY TIN TỰ ĐỘNG

Nội dung chương 2 gồm:

- Phần 1: Lấy tin tự động (Định nghĩa lấy tin tự động và quy trình lấy tin tự động).
- Phần 2: Khai phá dữ liệu trong lấy tin tự động (Tìm hiểu về XML và RSS).

PHẦN I: LẤY TIN TỰ ĐỘNG

1. Định nghĩa

- Lấy tin tự động là quá trình tìm kiếm các thông tin có giá trị trong các khối dữ liệu lớn.
- Là việc trích lấy các thông tin từ các trang Web có nội dung cần quan tâm tới.

2. Quy trình lấy tin tự động

Với các loại dữ liệu khác nhau, quá trình lấy tin tự động thông thường đều được thực hiện qua các bước sau:

- **Bước 1**: Tìm hiểu về lĩnh vực và xác định các vấn đề có liên quan.
- **Bước 2**: Thu thập và tiền xử lý dữ liệu. Đây là bước rất quan trọng, chiếm phần lớn thời gian và sức lực (70 ÷ 80%) trong cả tiến trình.
- **Bước 3**: **Lấy tin tự động** trích chọn ra các mẫu, các thông tin có ý nghĩa. Bước này gồm các phương thức để tạo ra các thông tin hữu ích từ dữ liệu.
- **Bước 4**: Đưa các thông tin ra hiển thị.

PHẦN II: KHAI PHÁ DỮ LIỆU TRONG LẤY TIN TỰ ĐỘNG

Đặt vấn đề:

Sự phát triển nhanh chóng của mạng Internet và Intranet đã sinh ra một khối lượng khổng lồ các dữ liệu dạng siêu văn bản (dữ liệu Web). Cùng với sự thay đổi và phát triển hàng ngày hàng giờ về nội dung cũng như số lượng các trang Web trên Internet thì vấn đề tìm kiếm thông tin đối với người sử dụng lại ngày càng khó khăn.

Có thể nói trang Web như là cuốn từ điển bách khoa toàn thư. Thông tin trên các trang Web đa dạng về mặt nội dung cũng như hình thức, có thể nói Internet như một xã hội ảo, nó bao gồm các thông tin về mọi mặt của đời sống kinh tế, xã hội được trình bày dưới dạng văn bản, hình ảnh , âm thanh,….Tuy nhiên cùng với sự đa dạng và số lượng lớn thông tin như vậy đã nảy sinh vấn đề quá tải thông tin. Người ta không thể tự tìm kiếm địa chỉ trang Web chứa thông tin mà mình cần do vậy yêu cầu đặt ra là làm thế nào để lấy được thông tin mà mình cần trong khối lượng thông tin khổng lồ đó. Do vậy người ta đã **ứng dụng khai phá dữ liệu để lấy tin tự động.**

1. Tìm hiểu XML

1.1. Nguồn gốc và mục đích

XML (Extensible Markup Language) tức là ngôn ngữ đánh dấu mở rộng ra đời vào tháng 2/1998, do W3C đề xuất. XML là tập con của SGML (Standardized Generalized Makup Language). XML được thiết kế để chuyển tải và lưu trữ dữ liệu.

Mục đích chính của XML là đơn giản hoá việc chia sẻ dữ liệu giữa các hệ thống khác nhau, đặc biệt là các hệ thống được kết nối Internet.

1.2. Đặc điểm

- XML dùng văn bản (text) để mô tả thông tin. XML không phụ thuộc vào ứng dụng, phần mềm và phần cứng.

- XML có khả năng mô tả nhiều loại dữ liệu khác nhau. XML không định nghĩa trước thẻ (tag). Thẻ (tag) do người dùng tự định nghĩa.

1.3. Cấu trúc của XML

Một tài liệu XML được lưu trữ và tổ chức như một cây với một phần tử gốc(root) và các phần tử con (như là nhánh cây, lá cây).

Ví dụ:

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<note>
<to>Nam</to>
<from>Ba</from>
<heading>Nhac nho</heading>
<body>Dung quen buoi hen vao cuoi tuan!</body>
</note>
```

Dòng đầu tiên là khai báo XML, đây là dòng không bắt buộc. Dòng này với nhiệm vụ khai báo phiên bản XML đang sử dụng và còn có thể chứa thêm thông tin về mã hoá ký tự và các phụ thuộc ngoài.

Dòng tiếp theo là đặc tả phần tử gốc (root element) của tài liệu.

Bốn dòng tiếp theo là các phần tử con (child element) của tài liệu (to, from, heading, body).

Và dòng cuối cùng là kết thúc của phần tử gốc.

Tóm lại, ta có thể khái quát như sau: Mỗi tài liệu XML đều xuất phát từ phần tử gốc, và mỗi phần tử phải có hai thẻ: mở "<..>" và đóng "</…>" . Các phần tử có thể có nội dung và thuộc tính, giống như trong HTML. Giữa thẻ mở và thẻ đóng là nội dung của phần tử. Các phần tử có thể lồng nhau. Trong thẻ mở có thể chứa hoặc không chứa thuộc tính của phần tử

```
<root>
  <child>
    <subchild>.....</subchild>
  </child>
</root>
```

Ví dụ:

```
<bookstore>
<book category="COOKING">
```

```
  <title lang="vi">Sach nau an kieu Chau A</title>
  <author>Bui Thi May</author>
  <year>2009</year>
  <price>30.00</price>
</book>
<book category="CHILDREN">
  <title lang="en">Harry Potter</title>
  <author>J K. Rowling</author>
  <year>2009</year>
  <price>29.99</price>
</book>
<book category="WEB">
  <title lang="vi">Tim hieu ve XML</title>
  <author>Duong Quang Thien</author>
  <year>2009</year>
  <price>39.95</price>
</book>
</bookstore>
```

Tất cả các phần tử <book> đều được chứa trong <bookstore>. Mỗi phần tử <book> lại có bốn phần tử con <title>, <author>, <year>, <price>.

1.4. Ứng dụng của XML

Do XML dễ hiểu, mang tính không phụ thuộc vào ứng dụng, phần mềm và phần cứng, dễ dàng chia sẻ,… nên nó ngày càng được ứng dụng rộng rãi.

Thứ nhất, nó được ứng dụng trong Web Services với vai trò là cầu nối trung gian cho việc trao đổi dữ liệu giữa những Web Services.

Thứ hai, nó được ứng dụng trong mô hình ADO.NET của Microsoft với vai trò lưu trữ và chuyển dữ liệu.

Thứ ba, nó được ứng dụng trong công nghệ OpenSearch. Khi người dùng nhập từ khoá tìm kiếm thì kết quả trả về cho người dùng dưới dạng RSS hoặc Atom (là một định dạng tập tin dựa trên chuẩn XML).

Và một ứng dụng mang tầm nhìn tương lai hơn nữa đó là Semantic Web.

“The Semantic Web = a Web with a meaning”

Semantic Web được hiểu và dịch ra tiếng việt là web ngữ nghĩa. Semantic Web là web dữ liệu. (***web of data***). Có rất nhiều dữ liệu mà chúng ta sử dụng hàng ngày, nhưng nó không là một phần, một bộ phận của web. Chúng ta có

thể xem thông tin tài khoản ngân hàng, xem ảnh, xem e-mail, nghe nhạc trên web. Nhưng chúng ta có thể vừa xem ảnh vừa xem lịch vừa xem thông tin về tài khoản ngân hàng trên cùng một website được không? Tại sao không? Bởi vì chúng ta không có web dữ liệu. Bởi vì dữ liệu do những ứng dụng lưu trữ và mỗi ứng dụng giữ nó cho riêng chúng. Nói đến Semantic Web là nói đến hai vấn đề sau:

- Là về những khuôn thức phổ biến(common formats) cho việc ***tích hợp, kết hợp cơ sở dữ liệu từ nhiều nguồn khác nhau. Dữ liệu được chia sẻ, được sử dụng lại ở những ứng dụng khác nhau.***
- Dữ liệu quan hệ với đối tượng thực (real world objects) như thế nào. Semantic Web là web mà có thể ***đặc tả các thông tin theo cách mà máy tính có thể hiểu được.***

Semantic Web ***không phải là links giữa những trang web.*** Mà nó đặc tả những ***mối quan hệ*** giữa các sự vật, sự việc(như A là bộ phận của B, Y là thành viên của Z), ***thuộc tính*** của các sự vật, sự việc(như cân nặng, chiều cao).

"If HTML and web made all the online document look like one huge book, RDF, schema, and inference languages make all data in the world look like one huge database".

Tim Berners-Lee, Weaving the Web, 1999.

"Nếu HTML và web làm cho tài liệu giống như một cuốn sách, thì RDF, lược đồ, và những ngôn ngữ suy luận làm cho tất cả dữ liệu trên toàn thế giới như một cơ sở dữ liệu khổng lồ".

Một ứng dụng phổ biến về Semantic Web hiện nay là ngôn ngữ RSS (một định dạng tập tin dựa trên chuẩn XML). Sau đây ta sẽ tìm hiểu RSS là gì?

2. Tìm hiểu về RSS

2.1. Tổng quan về RSS

RSS là tên viết tắt của từ Really Simple Syndication(tức là nguồn cung cấp thông tin cực kỳ đơn giản).

RSS là công nghệ khai thác và cung cấp thông tin tuỳ theo thị hiếu và mức độ quan tâm của người dùng. Thay vì phải dành thời gian để tìm đọc các tin mới trên những website, thì với phần mềm đọc tin RSS, bạn chỉ cần lựa chọn tin cần đọc trong danh sách tin mới nhất được cập nhật liên tục từ nhiều website có tích hợp RSS. Sử dụng RSS, các nhà cung cấp nội dung web có thể dễ dàng tạo và phổ biến các thông tin như tiêu đề tin, tóm tắt, hình ảnh và link liên kết tới trang web chứa nội dung đầy đủ.

Hiện nay, công nghệ RSS đang ngày dần phổ biến. Đứng riêng một mình thì RSS gần như vô dụng, mà nó phải cần một trình duyệt có hỗ trợ RSS, hoặc một chương trình chuyên nghiệp để đọc tin RSS từ các trang web có RSS. Hiện nay chỉ có một vài trình duyệt đời mới như Firefox, Opera,… có hỗ trợ đọc tin RSS, còn Internet Explorer 6 của Microsoft hoàn toàn không có chức năng này, chỉ có Internet Explorer 7 mới được tích hợp.

Khi truy cập vào các trang web có hỗ trợ RSS thì tất cả những trình duyệt có công nghệ RSS đều tự động đưa ra thông báo rằng trang web đang truy cập là dạng trang có RSS bằng một biểu tượng màu vàng cam có 3 chấm ở giữa . Khi nhấn chuột vào biểu tượng này thì trình duyệt sẽ tự động ghi vào nội dung tin tức vừa cập nhật mới nhất.

Phiên bản RSS đầu tiên ra đời vào năm 1997 do Dave Winer ở UserLand thiết kế với tên gọi là scriptingNews. Và cho đến bây giờ phiên bản đang được dùng phổ biến đó là RSS 2.0. Sau đây là cấu trúc cú pháp chuẩn của RSS 2.0.

2.2. Lịch sử ra đời RSS

- RDF (*Resource Description Framework*) Site Summary, phiên bản đầu tiên của RSS được tạo bởi Dan Libby của Netscape vào *tháng 3 năm 1999* dùng cho cổng điện tử My Netscape. Phiên bản này trở thành *RSS 0.9.*

- *Tháng 7 năm 1999*, Libby đưa ra bản phác thảo đầu tiên đặt tên là *RSS 0.91* (RSS viết tắt của *Rich Site Summary*). Từ đó, Libby đề xuất ra định dạng tương tự *RSS 1.0.*

- Cùng thời điểm đó Winer đưa ra phiên bản sửa đổi của RSS 0.91 cho website Userland.
- *Tháng 12 năm 2000,* nhóm RSS-DEV tiếp tục đưa ra *RSS 1.0* dựa trên bản phác thảo góp ý sửa đổi cho bản đặc tả kỹ thuật đưa ra bởi Tristan Louis. Giống với RSS 0.9 bản này dựa vào đặc tả kỹ thuật của RDF, nhưng có tính khả mở hơn, với nhiều mục bắt nguồn từ các từ vựng metadata chuẩn như Dublin Core.
- Mười chín ngày sau, Winer cho ra phiên bản *RSS 0.92*, và một vài chỉnh sửa có tính tương thích với các thay đổi của RSS 0.91 dựa trên cùng bản góp ý.
- *Tháng 4 năm 2001,* ông đưa ra bản phác thảo của *RSS 0.93* mà hầu hết là giống với bản 0.92. Bản thảo *RSS 0.94 ra đời vào tháng 8*, phục hồi lại những thay đổi trong bản 0.93, và thêm vào thuộc tính (attribute) type cho thành phần (element) description .
- *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.

2.3. Quy định của RSS

RSS được viết trong XML. Vì RSS là một định dạng tập tin dựa trên chuẩn XML nên nó cũng tuân theo những qui định của XML:

- Tất cả các phần tử phải có thẻ đóng.
- RSS phân biệt chữ hoa, chữ thường.
- Các phần tử phải được lồng đúng cách.
- Các thuộc tính phải thường được đặt trong dấu “” .
- Chú thích trong RSS:

```
<!-- This is an RSS comment -->
```

2.4. Cú pháp của RSS

Cấu trúc cú pháp của RSS rất đơn giản. Hãy xem ví dụ dưới đây:

```
<?xml version="1.0" encoding="ISO-8859-1" ?>
<rss version="2.0">
<channel>
  <title>W3Schools Home Page</title>
  <link>http://www.w3schools.com</link>
  <description>Free web building tutorials</description>
  <item>
    <title>RSS Tutorial</title>
    <link>http://www.w3schools.com/rss</link>
    <description>New RSS tutorial on
W3Schools</description>
  </item>
  <item>
    <title>XML Tutorial</title>
    <link>http://www.w3schools.com/xml</link>
    <description>New XML tutorial on
W3Schools</description>
  </item>
</channel>
</rss>
```

Dòng đầu tiên trong tài liệu RSS là dòng khai báo XML, khai báo phiên bản của XML và mã hoá ký tự được sử dụng trong tài liệu RSS.

Dòng thứ hai là khai báo RSS và phiên bản RSS là 2.0.

Dòng tiếp theo chứa phần tử **<channel>**. Phần tử này được dùng để miêu tả RSS feed. Ba dòng tiếp theo tức là: Phần tử <channel> gồm có ba phần tử con (child element), ba phần tử này là bắt buộc phải có:

- **<title>**: đặc tả tiêu đề của channel(ví dụ: W3schools Home Page).
- **<link>**: đặc tả liên kết của channel(ví dụ: www.w3school.com/rss).
- **<description>**: đặc tả miêu tả của channel(ví dụ: Free web building tutorials).

Mỗi phần tử <channel> có một hoặc nhiều phần tử <item>.

Mỗi phần tử **<item>** định nghĩa một mục(an article) trong RSS feed. Ví dụ ở đây ta có 2 mục đó là : RSS Tutorial và XML Tutorial

Phần tử <item> có ba phần tử con(child element): <title>, <link>, <description>.

Và hai dòng cuối cùng là các thẻ đóng phần tử <channel> và <rss>.

2.5. Các phần tử trong RSS <channel>

Như đã nói trong ví dụ trên, phần tử **<channel>** miêu tả RSS feed. Và phần tử này gồm ba phần tử con bắt buộc phải có là **<title>**, **<link>**, và **<description>**. Ngoài ra, còn có nhiều phần tử con khác nữa để cho chúng ta lựa chọn. Chẳng hạn như sau:

Phần tử	Đặc tả
<category>	Các danh mục trong feed của bạn
<copyright>	Tài liệu có bản quyền
<image>	Ảnh
<language>	Ngôn ngữ dùng trong tài liệu RSS
<lastBuidDate>	Ngày cuối cùng sửa tin
Phần tử	**Đặc tả**
<managingEditor>	Địa chỉ email của tác giả
<pubDate>	Ngày đăng tin
<ttl>	Thời gian (tính theo phút) mà tin tức có thể lưu giữ trước khi nó được cập nhật, làm mới từ nguồn cung cấp
<webMaster>	Địa chỉ email của người quản trị web RSS

Ví du:

```
<image>
  <url>http://www.w3schools.com/images/logo.gif</url>
  <title>W3Schools.com</title>
  <link>http://www.w3schools.com</link>
</image>
```

Phần tử **<image>** cũng yêu cầu cần phải có ba phần tử con là:

- **<url>**: đặc tả link liên kết tới ảnh.
- **<title>**: đặc tả dòng văn bản khi ảnh không thể hiển thị được.

- **<link>**: đặc tả link liên kết tới website trong <channel>.

2.6. Các phần tử trong RSS <item>

Phần tử **<item>** đặc tả danh mục của RSS feed. Cũng giống như phần tử <channel> phần tử <item> cũng gồm ba phần tử con bắt buộc phải có đó là: **<title>**, **<link>**, và **<description>**. Ngoài ra còn có thêm một số phần tử con khác nữa để chúng ta lựa chọn. Chẳng hạn như sau:

Phần tử	Đặc tả
<author>	Địa chỉ email của tác giả
<category>	Item này thuộc một hay nhiều danh mục
<comments>	Đường dẫn link tới phần nhận xét về item
<guid>	Là một chuỗi duy nhất để nhận dạng item
<enclosure>	Audio, media
<pubDate>	Ngày đăng tin
<source>	Item này thuộc channel nào

Ví du:

```
<enclosure url = “http://www.w3chool.com/rss/rss.mp3”
 length = “5000”  type = audio/mpeg />
```

Phần tử <enclosure> phải gồm ba thuộc tính bắt buộc đó là:

- url: đặc tả url của file media
- length: đặc tả dung lượng của file media
- type: đặc tả định dạng của file media

Vậy là ta đã đi tìm hiểu đầy đủ những kiến thức cơ bản về công nghệ và ứng dụng của XML, trong đó có RSS. Sau đây là phần phân tích và thiết kế chương trình hỗ trợ đọc tin RSS.

CHƯƠNG 3 – PHÂN TÍCH THIẾT KẾ CHƯƠNG TRÌNH

Nội dung chương 3 bao gồm:

- Phần 1: Tổng quan về chương trình. Phần này nêu ra mục đích và đối tượng sử dụng hệ thống
- Phần 2: Khảo sát, phân tích và đánh giá yêu cầu. Phần này bao gồm khảo sát một số website, phần mềm trong và ngoài nước. Rồi từ đó đưa ra đánh giá và lựa chọn giải pháp.
- Phần 3: Phân tích chức năng của hệ thống.Phần này đưa ra biểu đồ Use-case, đặc tả chi tiết các Use-case và vẽ biểu đồ tuần tự (sequence diagram).
- Phần 4: Thiết kế cơ sở dữ liệu. Trong phần này đưa ra đặc tả chi tiết các bảng và mô hình mối quan hệ giữa các bảng

3.1. Tổng quan về chương trình

Trong thời đại bùng nổ thông tin như hiện nay thì việc khai thác, thu thập và chia sẻ thông tin đóng một vai trò quan trọng. Với một dữ liệu khổng lồ trên mạng, làm sao ta có thể nắm bắt được thông tin mới nhất, nhanh chóng nhất mà không phải tốn thời gian lướt từng website để đọc và tìm kiếm thông tin.

Trên cơ sở này, hệ thống bóc tách thông tin được xây dựng nhằm phục vụ cho việc trích xuất thông tin từ các website, rồi tất cả thông tin được hiển thị trên một website, giúp cho người đọc có thể nắm bắt được thông tin một cách súc tích, nhanh chóng và tiết kiệm thời gian.

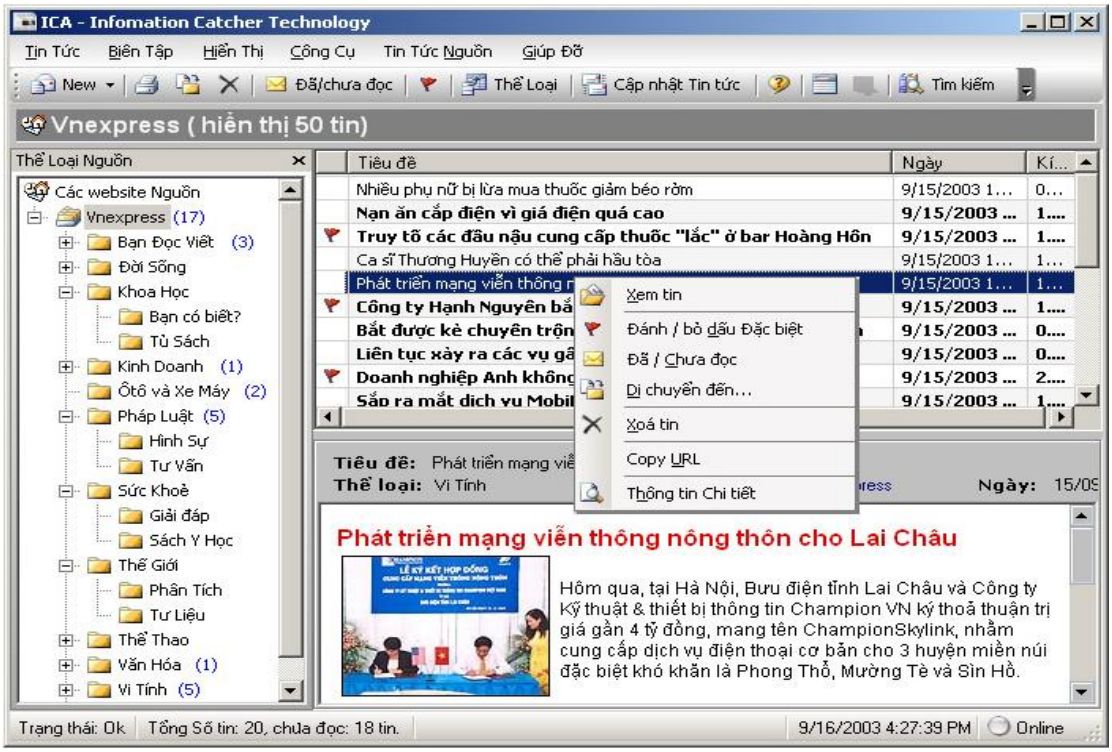
Đối tượng sử dụng hệ thống là tất cả cộng đồng người sử dụng mạng. Quản trị viên có thể quản lý tài khoản người dùng, quản lý các đường dẫn(link).

3.2. Khảo sát, phân tích và đánh giá yêu cầu

3.2.1. Khảo sát một số chương trình hỗ trợ đọc tin tức RSS

- **iCA:**

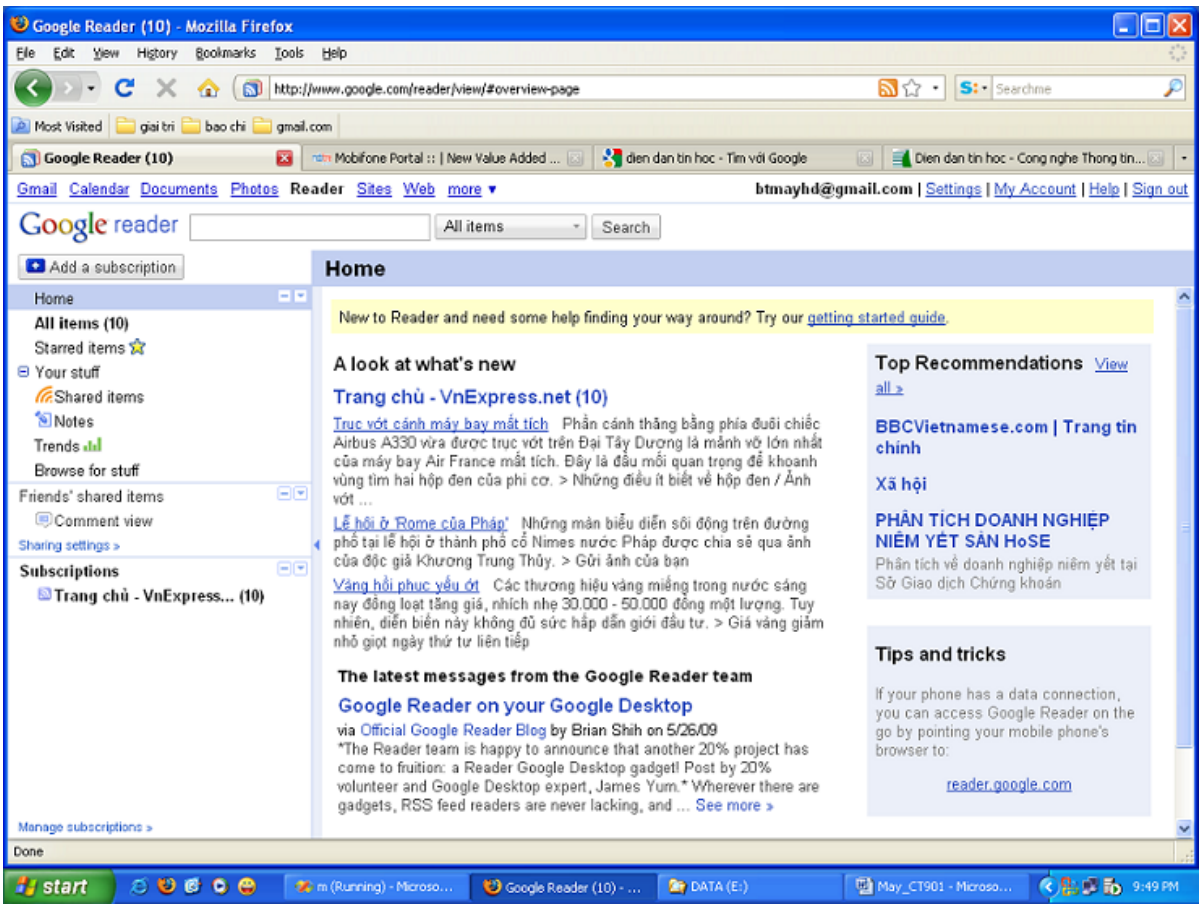
iCA là tên gọi tắt của "Information Catcher", là phần mềm được xây dựng dựa trên nền tảng và công nghệ dot NET của Microsoft. Phần mềm iCA hoạt động với tính năng nhận các thông tin từ Website tổng hợp sau đó hiển thị đầy đủ .



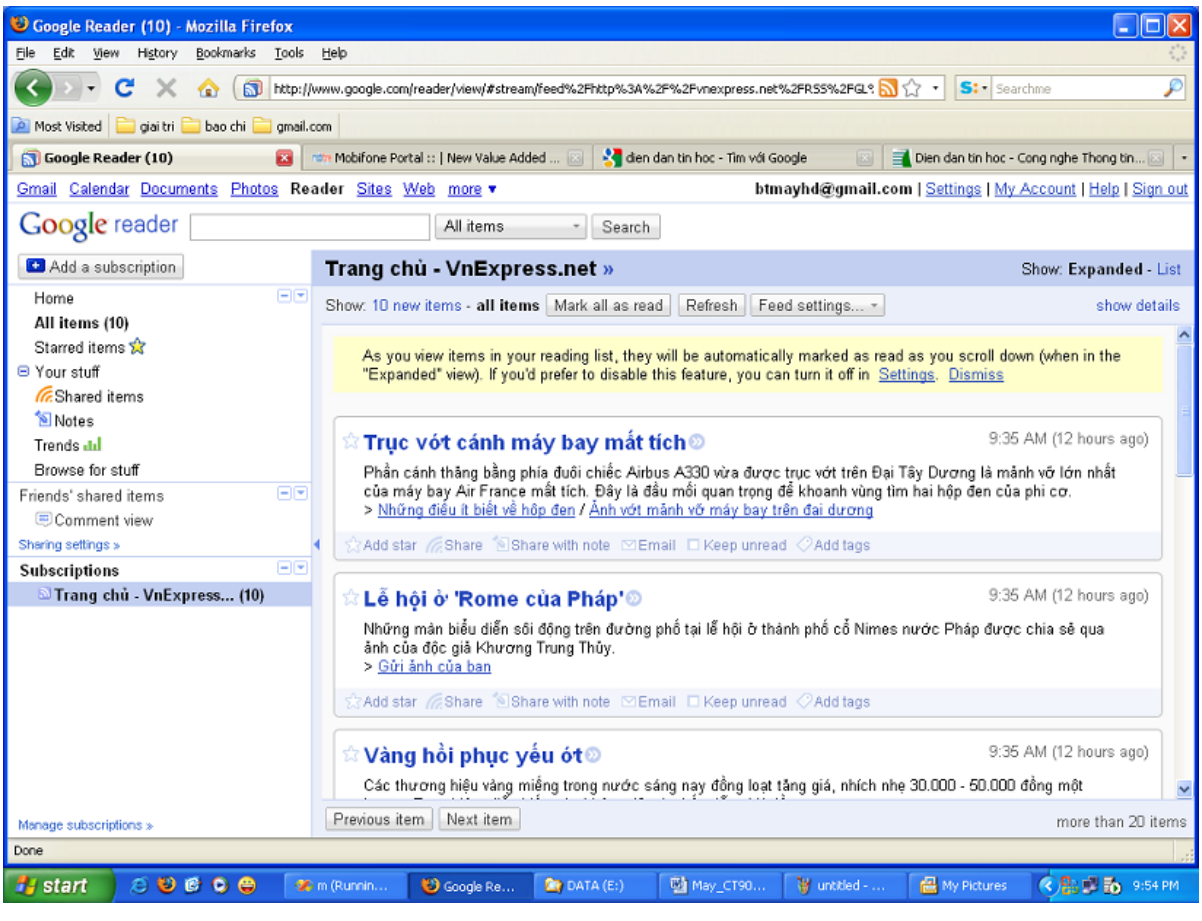
Hình 2 – Giao diện của iCA

- **Google Reader:**

Google Reader là một sản phẩm của Google dựa trên nền WebForm, có rất nhiều tính năng nổi trội: lựa chọn số tin tức được hiển thị, chia sẻ tin với bạn bè, phân nhóm tin tức, tìm kiếm tin tức…..Dưới đây là trang chủ Goolge Reader với giao diện tổng quan những kênh tin người dùng thêm vào.



Hình 3 – Giao diện trang chủ Google Reader



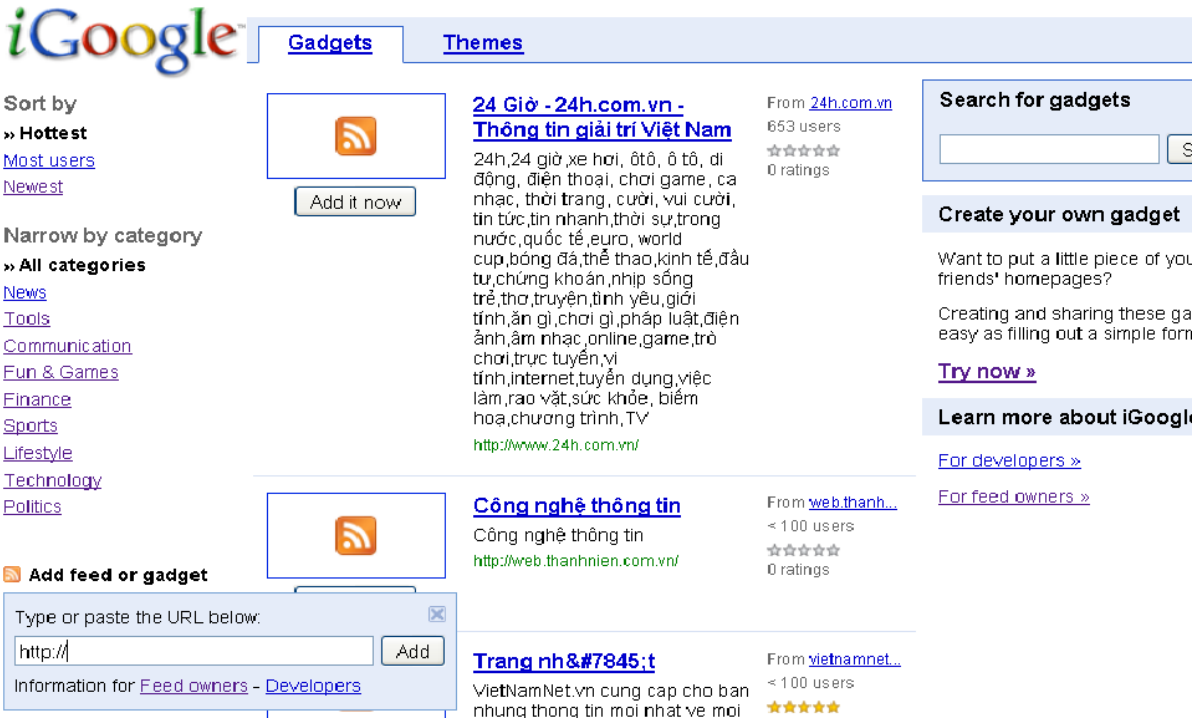
Hình 4 – Giao diện trang chi tiết của Google Reader

- **iGoogle:**

iGoogle là một cổng cá nhân (Personal Portal), sử dụng công nghệ AJAX và .NET Framework 3.5. Khi người dùng thêm kênh tin từ trang Google Reader, thì nó sẽ được tự động cập nhật vào trang iGoogle.

Hình 3 – Giao diện trang chủ của iGoogle

iGoogle còn cung cấp sẵn một directory RSS (là do những người dùng chia sẻ).

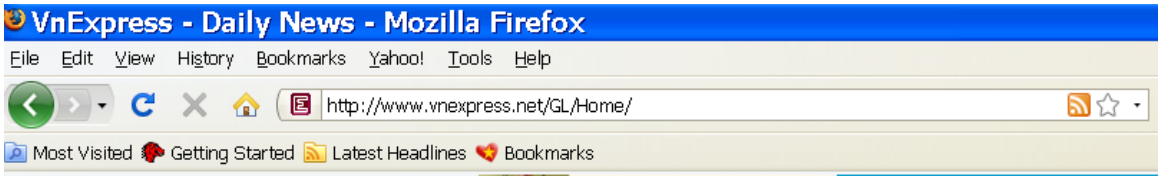


Hình 5 – Giao diện trang Gagdet của iGoogle

- **Trình duyệt FireFox**

Hiện nay các trình duyệt phiên bản mới nhất cũng hỗ trợ công nghệ RSS. Ví dụ như: Internet Explore 7.0 của Microsoft, Opera, FireFox,……..

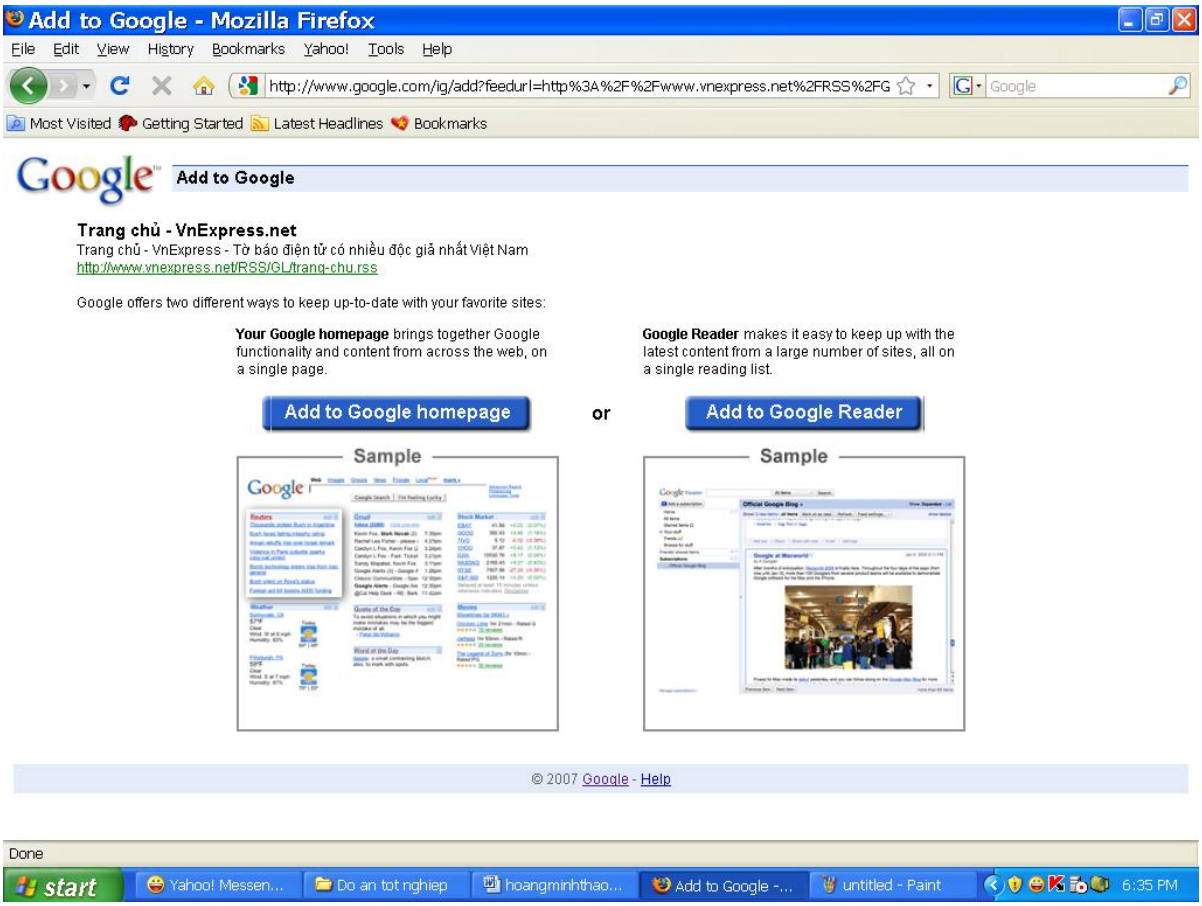
Khi bạn vào một website nào đó mà sử dụng công nghệ RSS thì trên trình duyệt của FireFox có xuất hiện biểu tượng màu da cam, ở giữa có ba chấm trắng.



Hình 6 – Giao diện trình duyệt FireFox

Nếu bạn muốn lấy tin từ trang tin đó, bạn chỉ cần kích vào biểu tượng đó và nó sẽ tự động chuyển tới trang lấy tin của Google Reader và iGoogle.

Hoặc bạn có thể sử dụng Live Bookmark được tích hợp trong trình duyệt FireFox để lấy tin.



Hình 7 – Giao diện trang lấy tin RSS

3.2.2. Tổng hợp yêu cầu của người dùng

Mục tiêu của đề tài là xây dựng nên một hệ thống hỗ trợ người dùng chọn kênh tin tức, thu thập tin tức, quản lý các kênh tin, tạo ra một website tin tức cho chính người dùng mà không phải lướt từng website để đọc tin tức. Thông qua việc khảo sát một số phần mềm đọc tin tức trong và ngoài nước, và yêu cầu từ phía người dùng, có thể tóm tắt yêu cầu của người dùng đối với hệ thống bóc tách thông tin như sau:

- Người dùng có thể tạo ra kênh tin tức cho riêng mình bằng cách chỉ cần đăng ký một tài khoản và đăng nhập vào nhập đường dẫn link tới địa chị trang website cần lấy tin.
- Người dùng có thể tổ chức, quản lý kênh tin tức của mình với các chức năng:
- Tạo nhóm tin tức(như: tin giáo dục, xã hội, tin chứng khoán,…), sửa nhóm tin và xoá nhóm tin.
- Lựa chọn số tin tức được hiển thị.
- Người dùng còn có thể tìm kiếm thông tin.

3.2.3. Đánh giá và lựa chọn giải pháp

Thông qua việc khảo sát một số website, phần mềm hỗ trợ đọc tin tức RSS ở trên, ta thấy có giải pháp để xây dựng hệ thống đó là: WinForm và WebForm. Sau đây, em sẽ đi phân tích những thuân lợi hay khó khăn của hai giải pháp trên. Và cuối cùng sẽ lựa chọn giải pháp cho chương trình của mình.

- **Sử dụng WinForm:**

  + Ưu điểm:

    Hỗ trợ nhiều tính năng

    Khả năng chạy không cần mạng(offline)

  + Nhược điểm:

    Người dùng phải mất thời gian cài đặt

Khó khăn trong việc nâng cấp: mỗi khi hệ thống nâng cấp, cập nhật thêm chức năng mới thì người dùng phải cài lại chương trình.

- **Sử dụng WebForm:**

+ Ưu điểm:

- Tính cơ động: Không cần cài đặt, không cần cấu hình, với ứng dụng sử dụng Web Forms, người dùng chỉ cần dùng một trình duyệt web kết nối với mạng Internet là có thể truy cập ở bất cứ chỗ nào. Đây có thể nói là ưu điểm lớn nhất của các ứng dụng Web Forms.

- Dế thay đổi: Sử dụng Web Forms đồng nghĩa với tất cả dữ liệu và chương trình đã nằm trên máy chủ. Chính vì vậy khi muốn sửa đổi, nâng cấp hệ thống, việc nâng cấp trên Web Forms có thể diễn ra rất dễ dàng. Người cung cấp dịch vụ chỉ cần cập nhật trực tiếp lên máy chủ, còn phía người dùng, các công việc này hoàn toàn trong suốt.

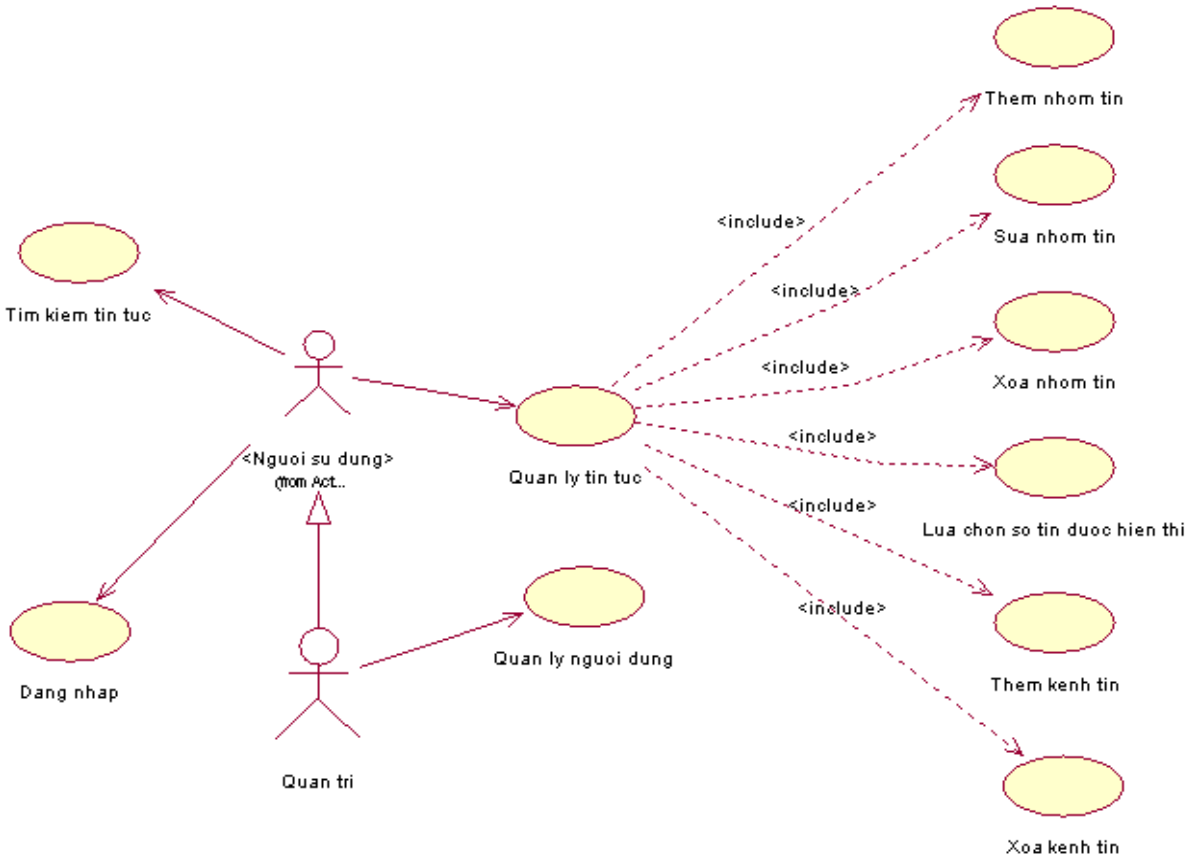
- Tính chia sẻ: có thể chia sẻ tin tức.

Sau khi xem xét các khía cạnh, ưu và nhược điểm của các công nghệ, em quyết định chọn WebForm để phát triển hệ thống. Cụ thể ở đây là công nghệ .NET của Microsoft, sử dụng ngôn ngữ lập trình C# và hệ quản trị CSDL Microsoft SQL Server 2000.

3.3. Phân tích chức năng hệ thống

3.3.1. Biểu đồ Use Case

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



Hình 8 - Biểu đồ User – case

3.3.2. Đặc tả các Use - case

- **Đặc tả Use – case đăng nhập**

  + **Tóm tắt**

  Use case này môt tả cách đăng nhập vào hệ thống bóc tách thông tin.

  + **Dòng sự kiện chính**

  - Use case này bắt đầu khi một actor (người dùng, quản trị viên) muốn đăng nhập vào hệ thống.
  - Hệ thống yêu cầu các actor nhập tên và mật khẩu.
  - Hệ thống kiểm tra tên và mật khẩu mà actor đã nhập. Nếu đúng hệ thống cho phép actor đăng nhập vào hệ thống.

  + **Dòng sự kiện khác**

  - Tên / mật khẩu sai:

Nếu trong dòng sự kiện chính các actor nhập tên, mật khẩu sai thì hệ thống sẽ báo lỗi. Actor có thể quay trở về đầu dòng sự kiện hoặc huỷ bỏ việc đăng nhập, lúc này use case kết thúc.

+ **Các yêu cầu đặc biệt**

Không có.

+ **Điều kiện tiên quyết**

Không có.

+ **Post condition**

Nếu use case thành công thì người đăng nhập sẽ có các quyền sử dụng hệ thống tương ứng. Ngược lại trạng thái của hệ thống không đổi.

+ **Điểm mở rộng**

Không có.

- **Đặc tả Use-case quản lý tin tức**

+ **Tóm tắt**

Use case này cho phép người sử dụng(đã là đăng nhập thành công) quản lý tin tức: thêm, sửa, xoá nhóm tin, lựa chọn số tin tức được hiển thị và thêm, xoá kênh tin.

+ **Dòng sự kiện**

Use case này bắt đầu khi người dùng đăng nhập vào hệ thống và thêm kênh tin và nhóm tin.

+ **Dòng sự kiện chính**

- Hệ thông sẽ liệt kê các nhóm tin, kênh tin của riêng thành viên đó.
- Thêm, sửa, xoá nhóm tin và kênh tin, lựa chọn số tin tức hiển thị.

+ **Các yêu cầu đặc biệt**

Không có

+ **Điều kiện tiên quyết**

Không có

+ **Post conditions**

Nếu use case thành công, thông tin về nhóm tin, kênh tin sẽ được cập nhật vào cơ sở dữ liệu.

+ **Điểm mở rộng**

Không có

- **Đặc tả Use- case quản lý người dùng**

+ **Tóm tắt**

Use case này cho phép quản trị viên thêm, sửa, xoá, tìm kiếm thông tin về thành viên sử dụng hệ thống. Quản lý trang tin của các thành viên (thêm, sửa, xoá trang tin của người sử dụng).

+ **Dòng sự kiện chính**

- Quản trị viên lựa chọn chức năng quản lý người dùng
- Hệ thống nhận thông tin từ quản trị viên.
- Hệ thống kiểm tra thông tin nhập vào
- Hệ thống truy xuất cơ sở dữ liệu.
- Theo từng yêu cầu của quản trị viên hệ thống sẽ thực hiện như sau:
  - ✓ Nếu quản trị viên yêu cầu thêm , sửa, xoá hoặc cập nhật lại thông tin về trang tin riêng của người sử dụng thì hệ thống sẽ cập nhật lại cơ sở dữ liệu tương ứng với các yêu cầu.
  - ✓ Nếu quản trị viên yêu cầu tìm kiếm thông tin người sử dụng thì hệ thống đưa ra những người sử dụng thoả yêu cầu của quản trị viên.
- Hệ thống thông báo thực hiện thành công.
- Use – case kết thúc.

+ **Dòng sự kiện phụ**

Nếu hệ thống không truy xuất được cơ sở dữ liêu thì sẽ báo lỗi, use – case kết thúc.

+ **Các yêu cầu đặc biệt**

Không có.

+ **Điều kiện tiên quyết**

Người quản lý đăng nhập vào hệ thống với quyền quản trị viên trước khi use – case bắt đầu.

+ **Post conditions**

Nếu use – case thành công thì thông tin của người sử dụng sẽ được cập nhật vào hệ thống. Ngược lại trạng thái của hệ thống không thay đổi.

+ **Điểm mở rộng**

Không có.

- **Đặc tả Use-case tìm kiếm tin tức**

+ **Tóm tắt**

Use case này cho phép người sử dụng tìm kiếm thông tin mà mình muốn tìm.

+ **Dòng sự kiện**

Use case này bắt đầu khi người dùng chọn chức năng tìm kiếm tin tức.

+ **Dòng sự kiện chính**

- Người dùng nhập thông tin muốn tìm.
- Công cụ google sẽ tìm kiếm.
- Liệt kê tất cả thông tin thoả yêu cầu.

+ **Dòng sự kiện phụ**

Nếu không tìm thấy thì thông báo cho người dùng biết là không tìm thấy.

+ **Các yêu cầu đặc biệt**

Không có.

+ **Điều kiện tiên quyết**

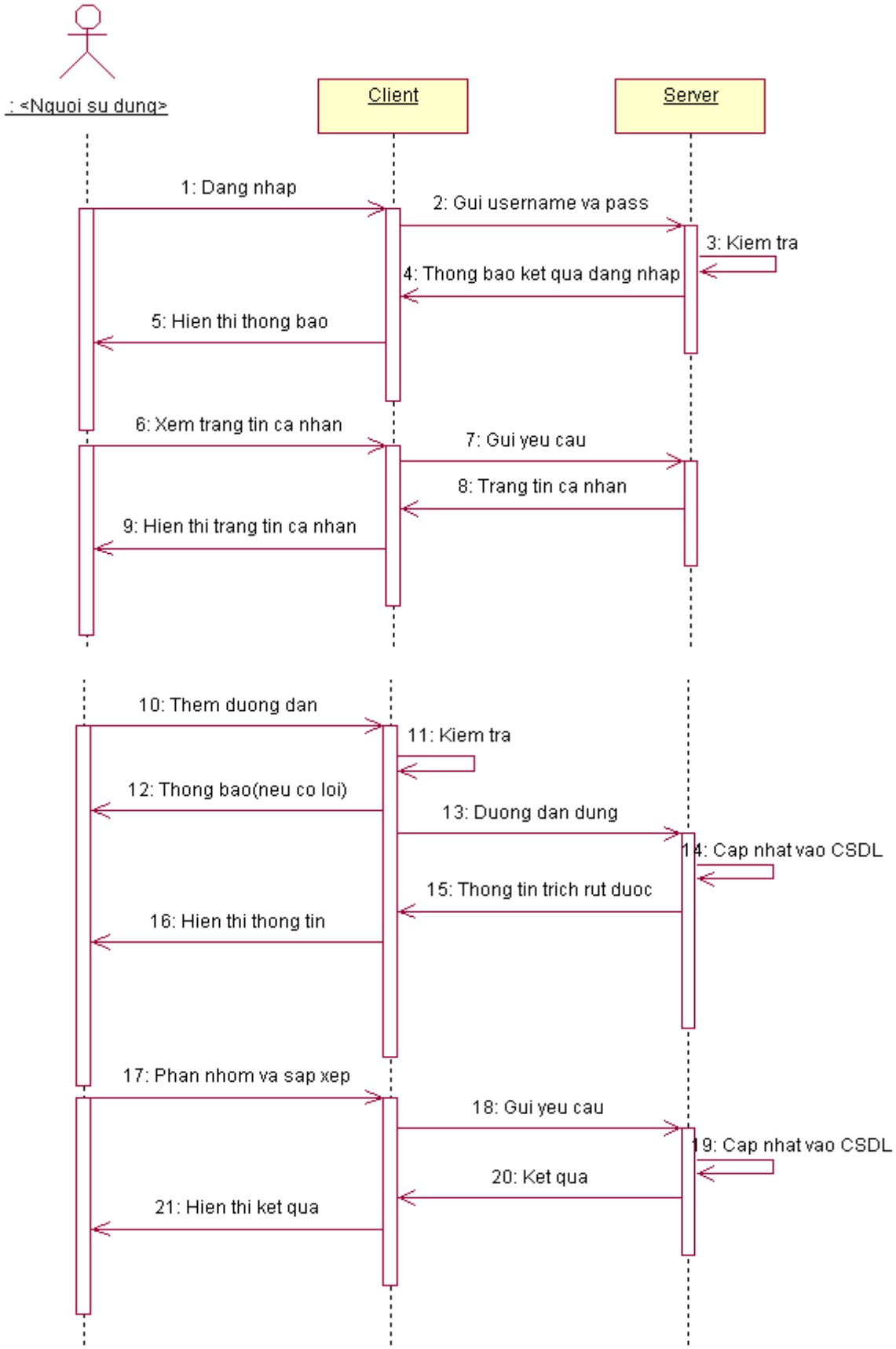
Không có.

+ **Post conditions**

Không có.

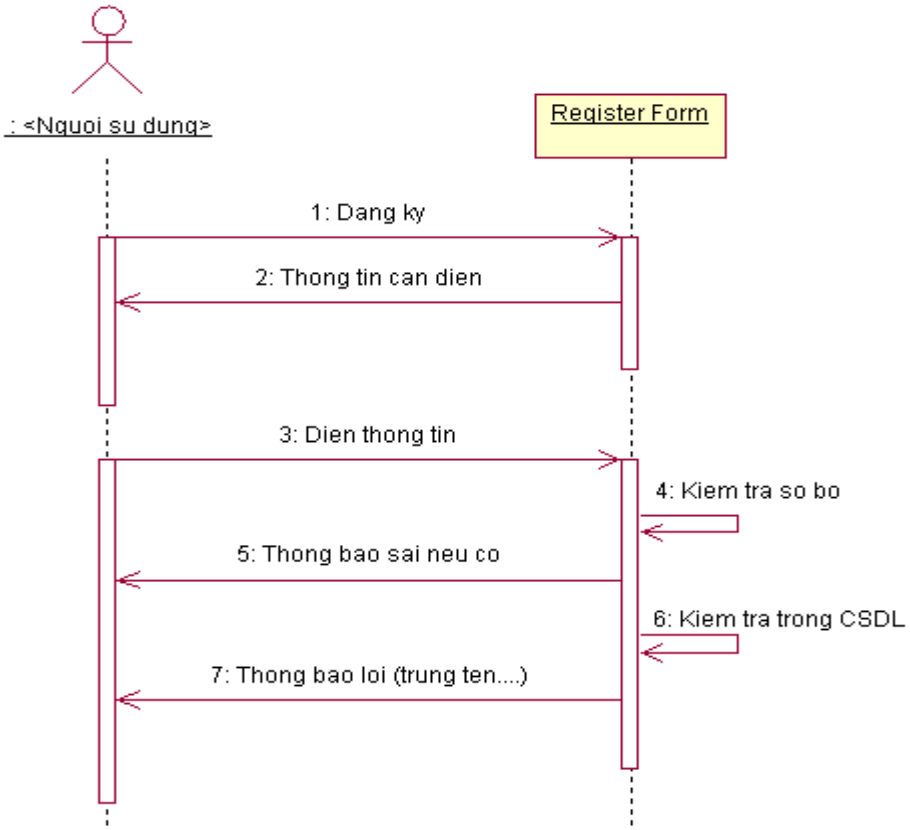
3.3.3. Biểu đồ tuần tự (Sequence Diagram)

Hoạt động của hệ thống: Nhìn một cách bao quát, hệ thống gồm những thao tác cơ bản sau:



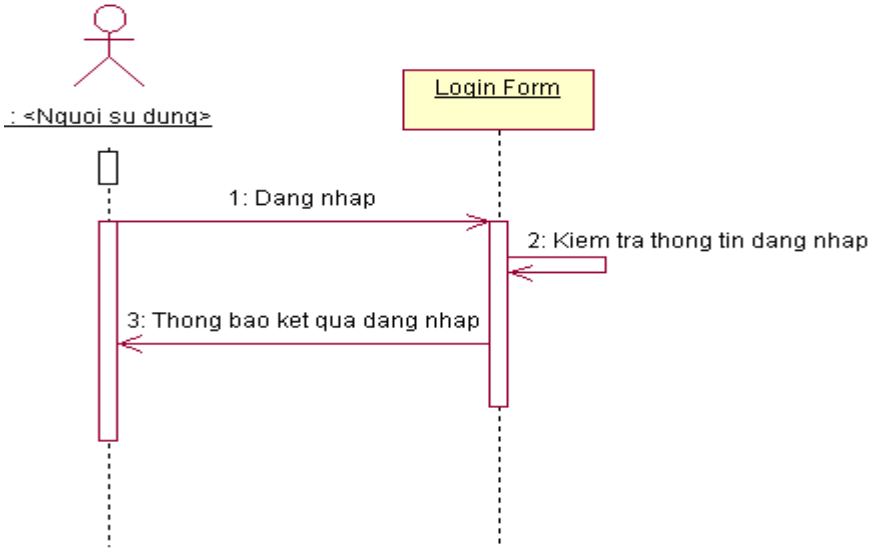
Hình 9 - Biểu đồ tuần tự - Toàn cảnh hệ thống

Đăng ký tài khoản: Để có thể tạo trang tin cá nhân người sử dụng cần phải đăng ký một tài khoản. Người dùng chỉ cần điền đúng và đầy đủ các thông tin mà chương trình đưa ra. Server có trách nhiệm cung cấp tài khoản mới cho người dùng.



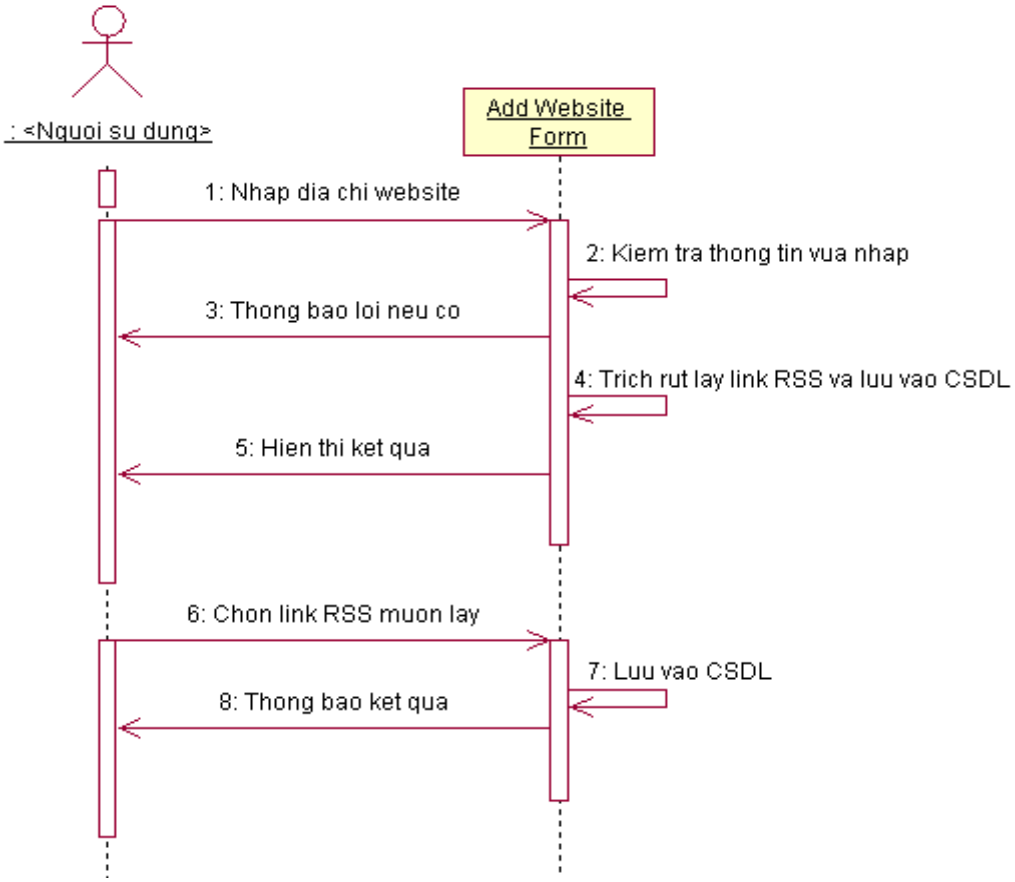
Hình 10 - Biểu đồ tuần tự - Đăng ký tài khoản

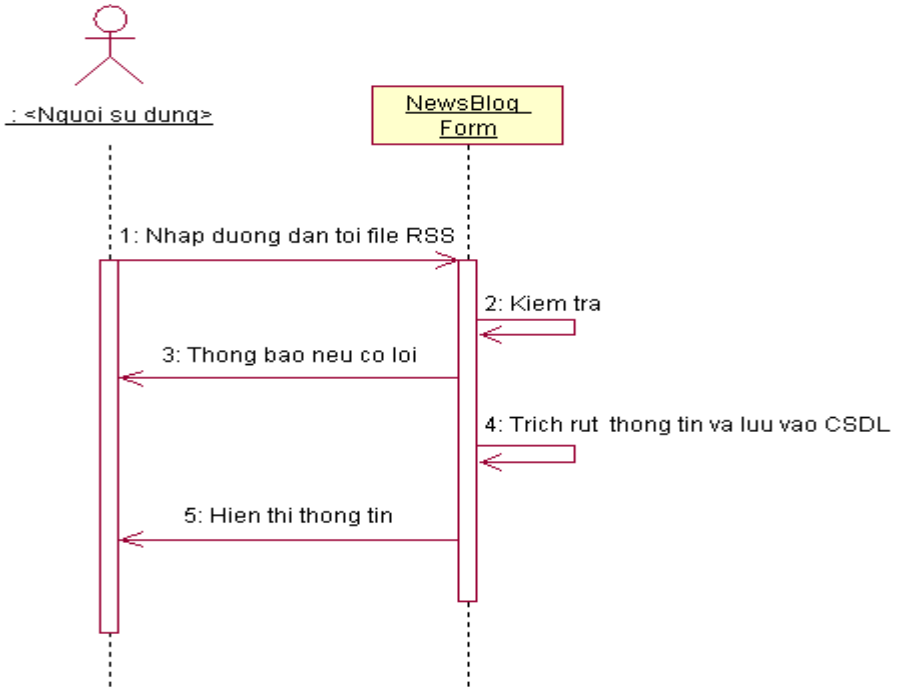
Đăng nhập hệ thống: Là hành động người dùng sử dụng tài khoản được cấp để vào hệ thống. Sau khi nhập các thông tin cần thiết, chương trình sẽ kết nối và kiểm tra tính hợp lệ. Người dùng sẽ được phản hồi kết quả.



Hình 11 - Biểu đồ tuần tự - Đăng nhập hệ thống

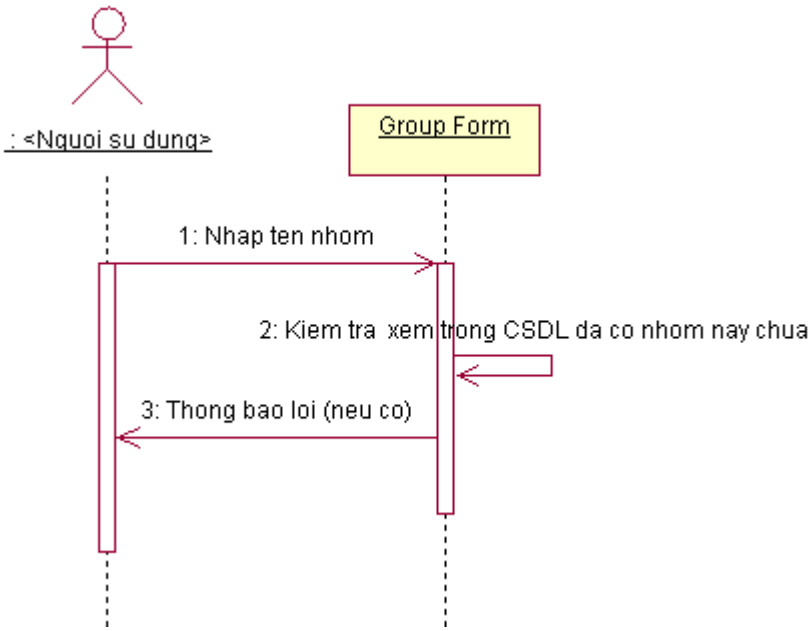
Thêm đường dẫn: Để lấy thông tin từ website khác, người dùng có thể nhập trực tiếp đường dẫn tới tập tin RSS, chương trình sẽ tự động trích rút tin tức và hiện thị lên cho người dùng. Hoặc người dùng có thể nhập đường dẫn tới website cung cấp RSS, chương trình sẽ trích rút các đường dẫn tới các tập tin RSS cho người dùng lựa chọn.





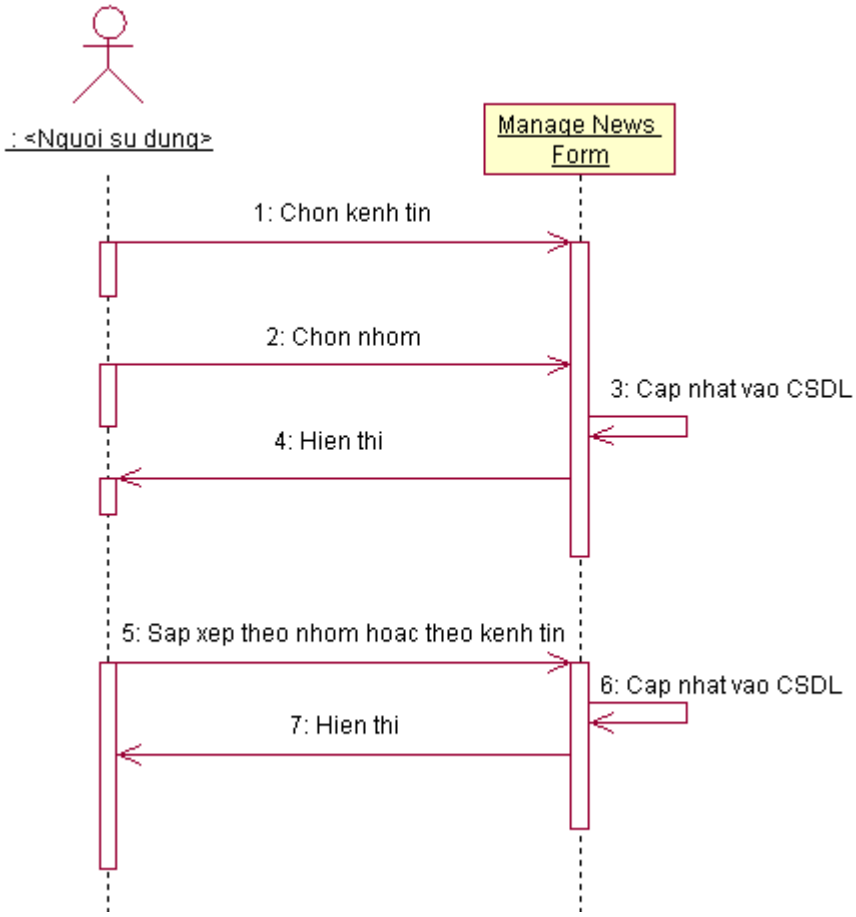
Hình 12 - Biểu đồ tuần tự - Thêm đường dẫn link

Thêm nhóm tin: Là thao tác mà người dùng thêm mới nhóm để phân loại tin tức.



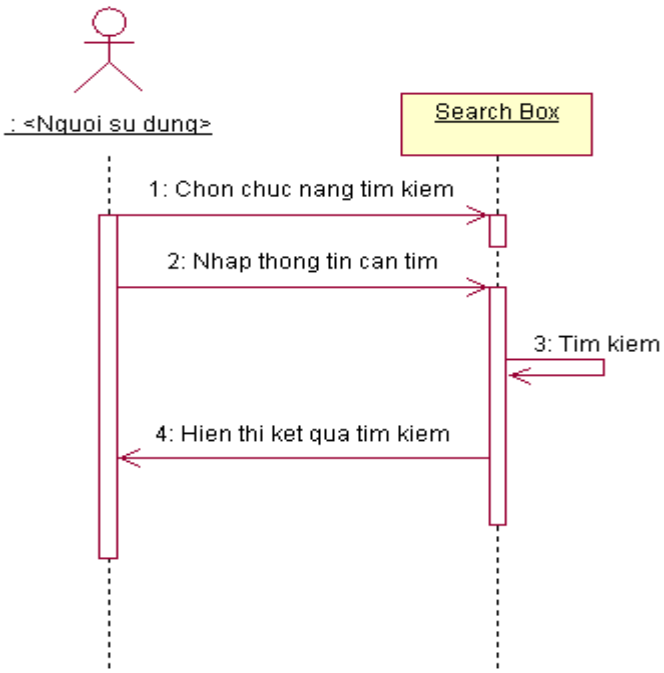
Hình 13 - Biểu đồ tuần tự - Thêm nhóm tin

Sắp xếp, phân loại nhóm tin:



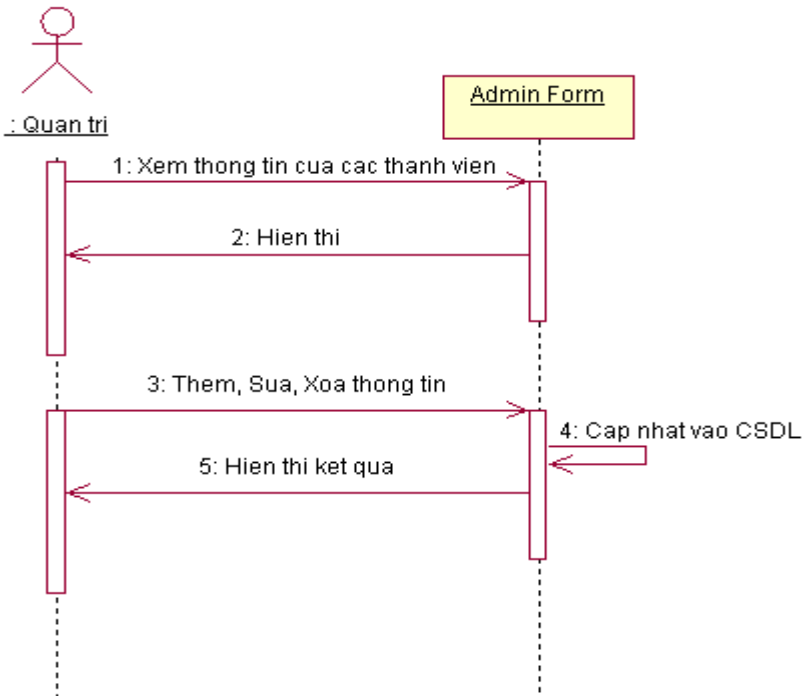
Hình 14 - Biểu đồ tuần tự - Sắp xếp nhóm tin

Tìm kiếm tin tức: Trước hết người dùng chọn chế độ tìm kiếm, đó là tìm kiếm tin tức trong hệ thống hay tìm kiếm trên google search.



Hình 15 - Biểu đồ tuần tự - Tìm kiếm thông tin

Quản lý người dùng: Đây là thao tác chỉ dành cho người dùng có quyền là quản trị. Quản trị viên có thể cung cấp tài khoản mới cho người dùng, có thể xoá tài khoản người dùng, quản lý trang tin cá nhân của người dùng.



Hình 16 - Biểu đồ tuần tự - Quản lý người dùng

3.4. Thiết kế cơ sở dữ liệu

Dữ liệu của chương trình ở mức vừa phải, chưa quá lớn nên em lựa chọn hệ quản trị cơ sở dữ liệu Microsoft SQL Server 2000.

3.4.1. Đặc tả chi tiết các bảng dữ liệu

Bảng Urls: chứa thông tin về địa chỉ website chứa các kênh tin.

tblUrls					
STT	**Tên trường**	**Kiểu dữ liệu**	**Độ dài**	**Ghi chú**	**Diễn giải**
1	UrlID	int	4	Khoá chính	Mã địa chỉ
2	uLink	nvachar	50	Khác rỗng	Đường dẫn tới website
3	uTitle	nvarchar	50		Tiêu đề của website
4	uDescription	nvarchar	50		Đặc tả về website

Bảng 1 - Bảng Urls (địa chỉ website)

Bảng Channels: chứa thông tin về các kênh tin tức

tblChannels					
STT	**Tên trường**	**Kiểu dữ liệu**	**Độ dài**	**Ghi chú**	**Diễn giải**
1	ChannelID	int	4	Khoá chính	Mã kênh tin
2	cLink	nvachar	50	Khác rỗng	Đường dẫn tới file RSS
3	cTitle	nvarchar	50	Khác rỗng	Tiêu đề của kênh tin
4	cDescription	navarchar	MAX	Khác rỗng	Đặc tả chi tiết về kênh tin
5	LastUpdated	dateTime		Khác rỗng	Thời gian cập nhật kênh tin
6	ItemCount	int	4	Khác rỗng	Số lượng tin tức có trong kênh tin

Bảng 2 - Bảng Channels (kênh tin)

Bảng Items: chứa thông tin về những tin tức mà hệ thống bóc tách lấy về.

tblItems					
STT	**Tên trường**	**Kiểu dữ liệu**	**Độ dài**	**Ghi chú**	**Diễn giải**
1	ItemID	int	4	Khoá chính	Mã tin tức
2	ChannelID	int	4	Khác rỗng	Mã kênh tin
3	iLink	nvarchar	50	Khác rỗng	Đường dẫn tới chi tiết của tin tức
4	iTitle	nvarchar	50	Khác rỗng	Tiêu đề của tin tức
5	iDescription	nvarchar	MAX	Khác rỗng	Nội dung chi tiết của tin tức
6	iPubDate	datetime			Ngày xuất bản tin
7	iAuthor	nvarchar	50		Tác giả viết tin

Bảng 3 - Bảng Items (tin tức)

Bảng Group: chứa thông tin về nhóm tin của mỗi người sử dụng

tblGroup					
STT	**Tên trường**	**Kiểu dữ liệu**	**Độ dài**	**Ghi chú**	**Diễn giải**
1	GroupID	int	4	Khoá chính	Mã nhóm
2	GroupName	nvachar	50	Khác rỗng	Tên nhóm
3	UserName	nvarchar	50	Khác rỗng	Tên đăng nhập của người sử dụng

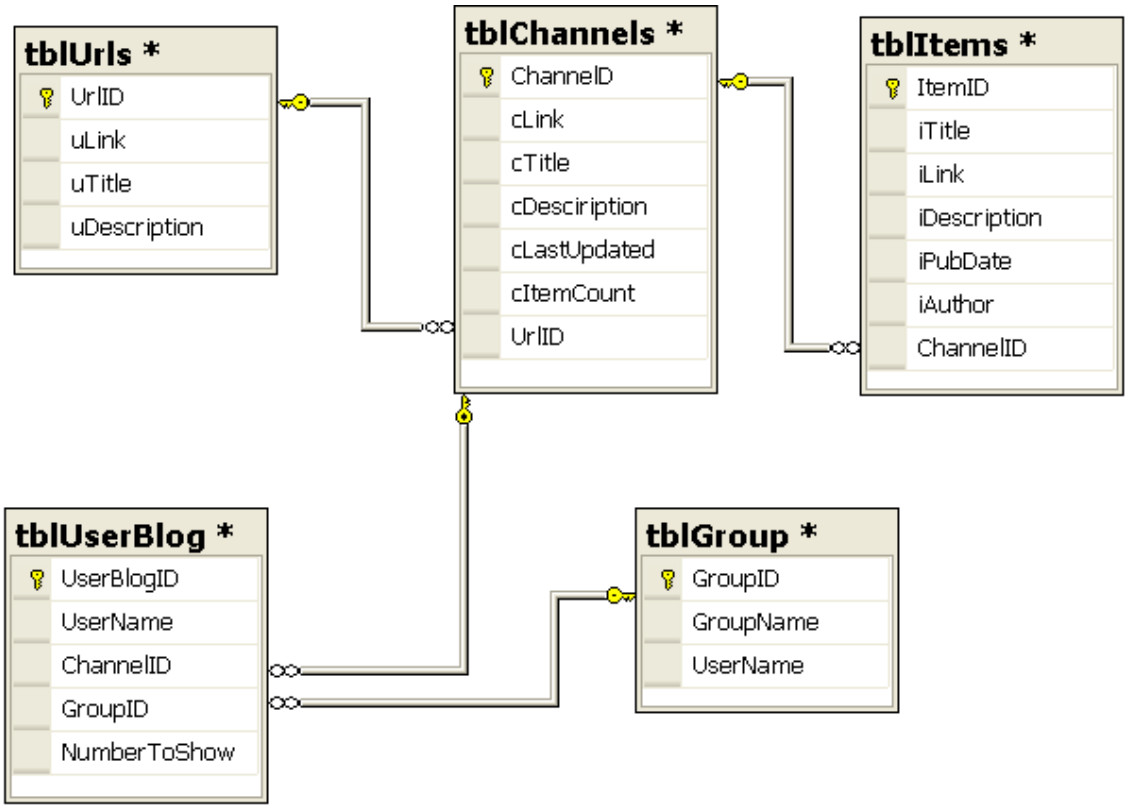
Bảng 4 - Bảng Group (nhóm tin tức)

Bảng UserBlog: chứa thông tin về blog tin tức của mỗi người dùng.

tblUserBlog					
STT	**Tên trường**	**Kiểu dữ liệu**	**Độ dài**	**Ghi chú**	**Diễn giải**
1	UserBlogID	int	4	Khoá chính	Mã trang blog tin tức của mỗi người dùng
2	UserName	nvarchar	50	Khác rỗng	Tên đăng nhập của người sử dụng
3	ChannelID	int	4	Khác rỗng	Mã kênh tin
4	GroupID	int	4	Khác rỗng	Mã nhóm
5	NumberToShow	int	4		Số lượng tin người dùng chọn trên mỗi kênh tin

Bảng 5 - Bảng UserBlog (kho tin tức của mỗi người dùng)

3.4.2. Mô hình quan hệ



Hình 17 – Mô hình quan hệ dữ liệu giữa các bảng

CHƯƠNG 4 – XÂY DỰNG CHƯƠNG TRÌNH

Nội dung chương 4 bao gồm:

- Phần 1: Mô tả qui trình lấy link RSS tự động, và đọc tập tin RSS.
- Phần 2: Một số màn hình giao diện đạt được

Phần trên ta đã xác định được yêu cầu chức năng của chương trình. Để đáp ứng được những chức năng đó ta cần làm như sau.

4.1. Qui trình tự động lấy đường dẫn tới tập tin RSS

Khi người dùng nhập đường dẫn tới webisite (chẳng hạn: http://vietnamnet.vn), thì nhiệm vụ của hệ thống là lấy tất cả những file RSS mà webisite đó cung cấp.

Bước 1: Ta phải tải nội dung trang HTML của website đó về.

Bước 2: Ta sử dụng đến biểu thức chính qui (Regular Expression) để lọc ra những thẻ <link> chứa đường dẫn tới file RSS.

Bước 3: Lọc ra đường dẫn tới file RSS, ta cũng dùng biểu thức chính qui để match() được href chứa đường dẫn tới file RSS.

Bước 4: Sau khi đã lấy được đường dẫn tới file RSS, lưu vào CSDL .Tiếp theo, đọc file RSS đó.

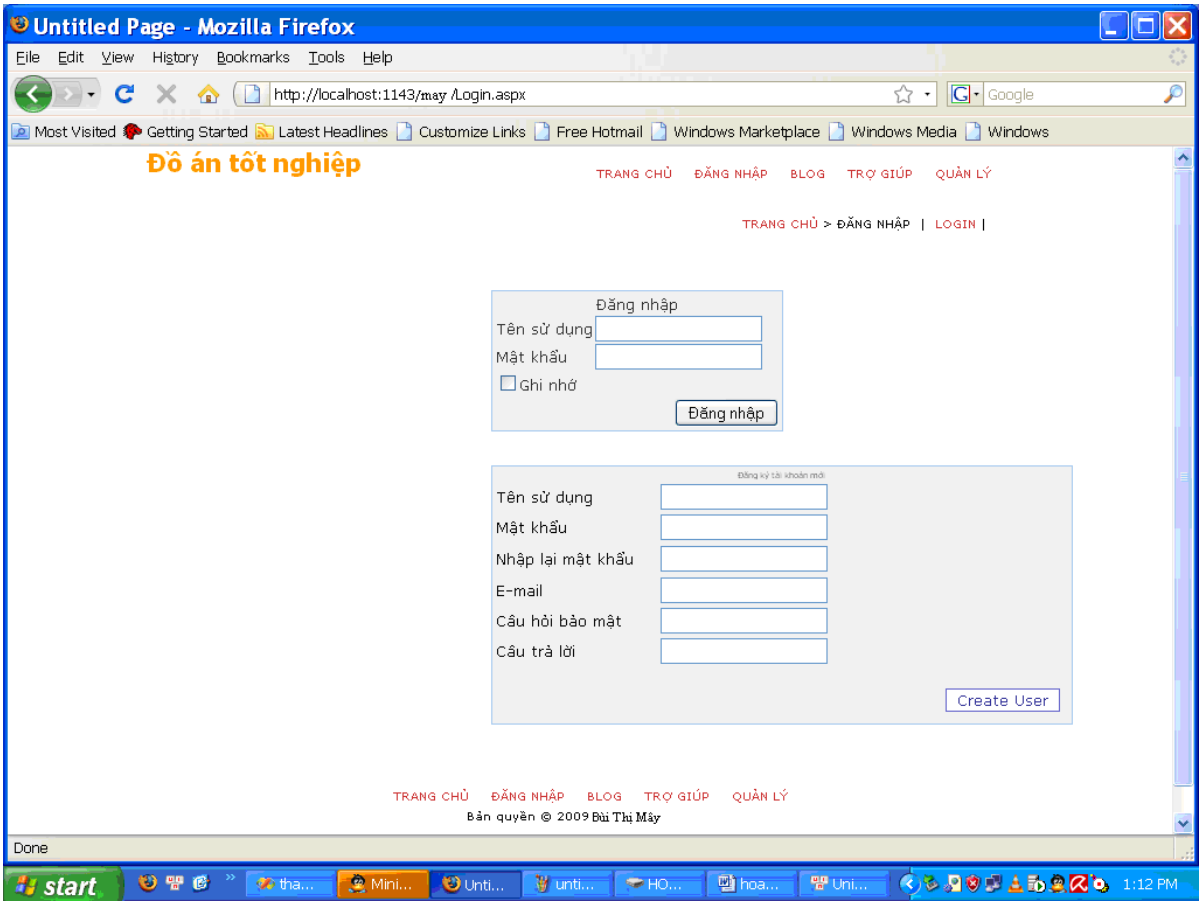
4.2. Qui trình đọc tập tin RSS

Người dùng có thể nhập trực tiếp đường dẫn tới file RSS. Nhiệm vụ của hệ thống là trích rút dữ liệu từ file RSS. Để trích rút dữ liệu ta làm như sau:

Bước 1: Trước tiên là thiết kế lớp RSSItem để chứa các dữ liệu mà ta trích rút từ file RSS.

Bước 2: Đọc file RSS

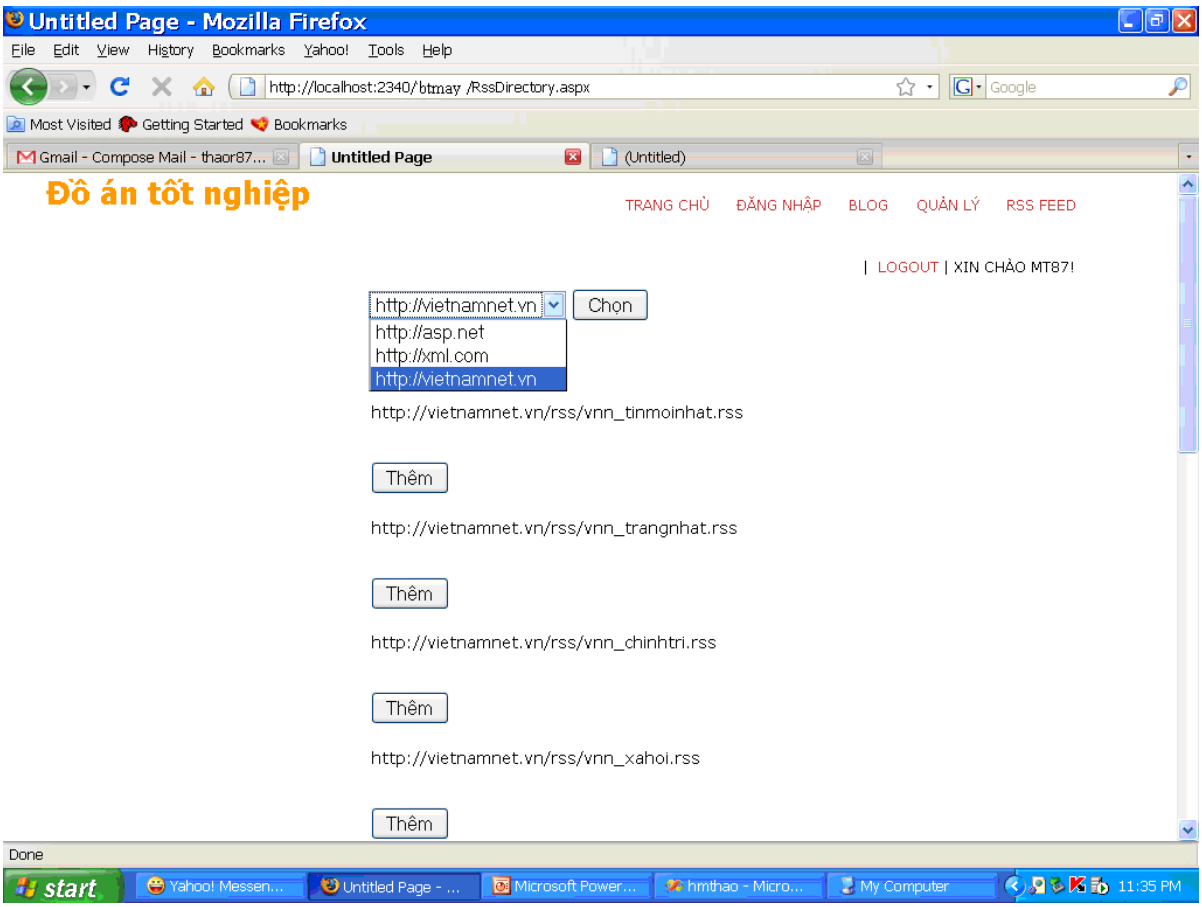
4.3. Một số màn hình giao diện đạt được



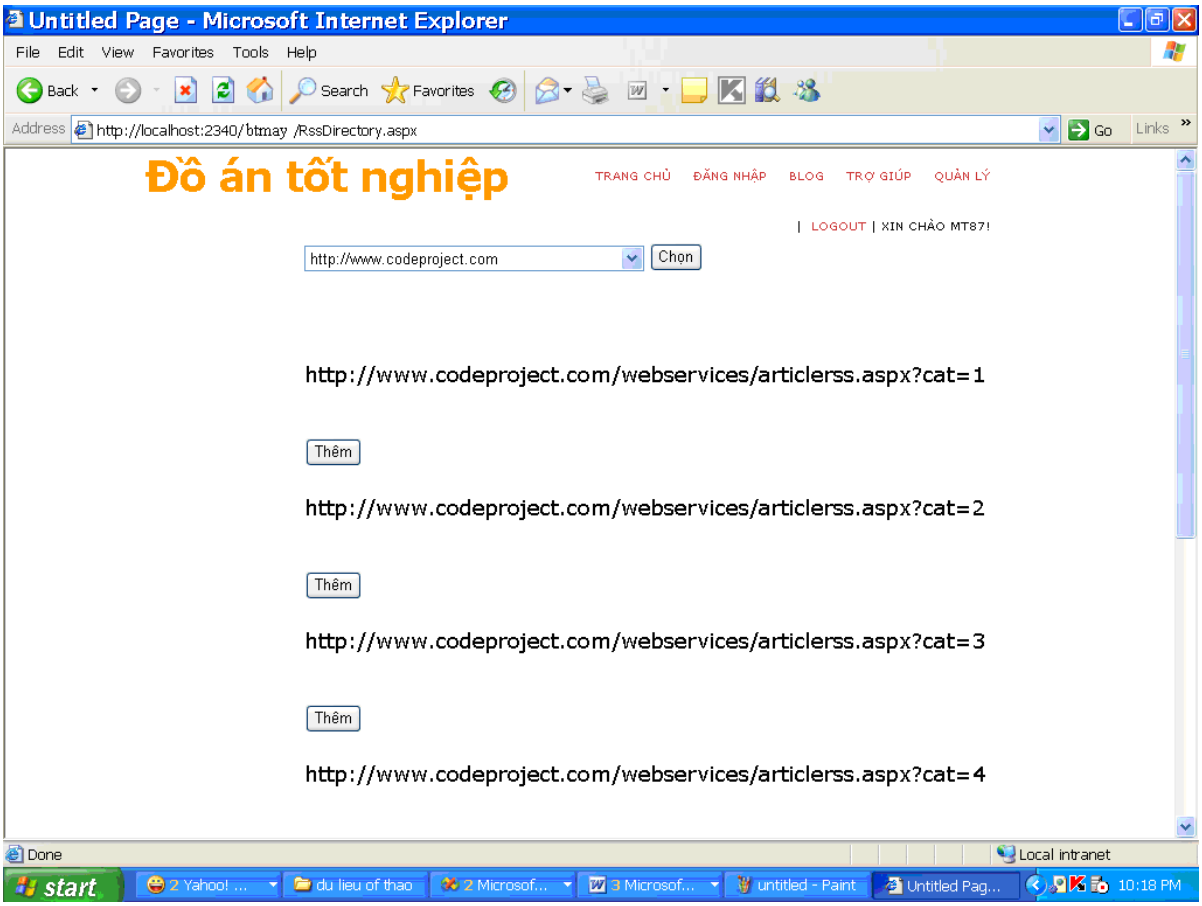
Hình 18 – Giao diện trang đăng nhập

Hình 19 – Giao diện trang quản lý người dùng

Hình 20 – Giao diện blog

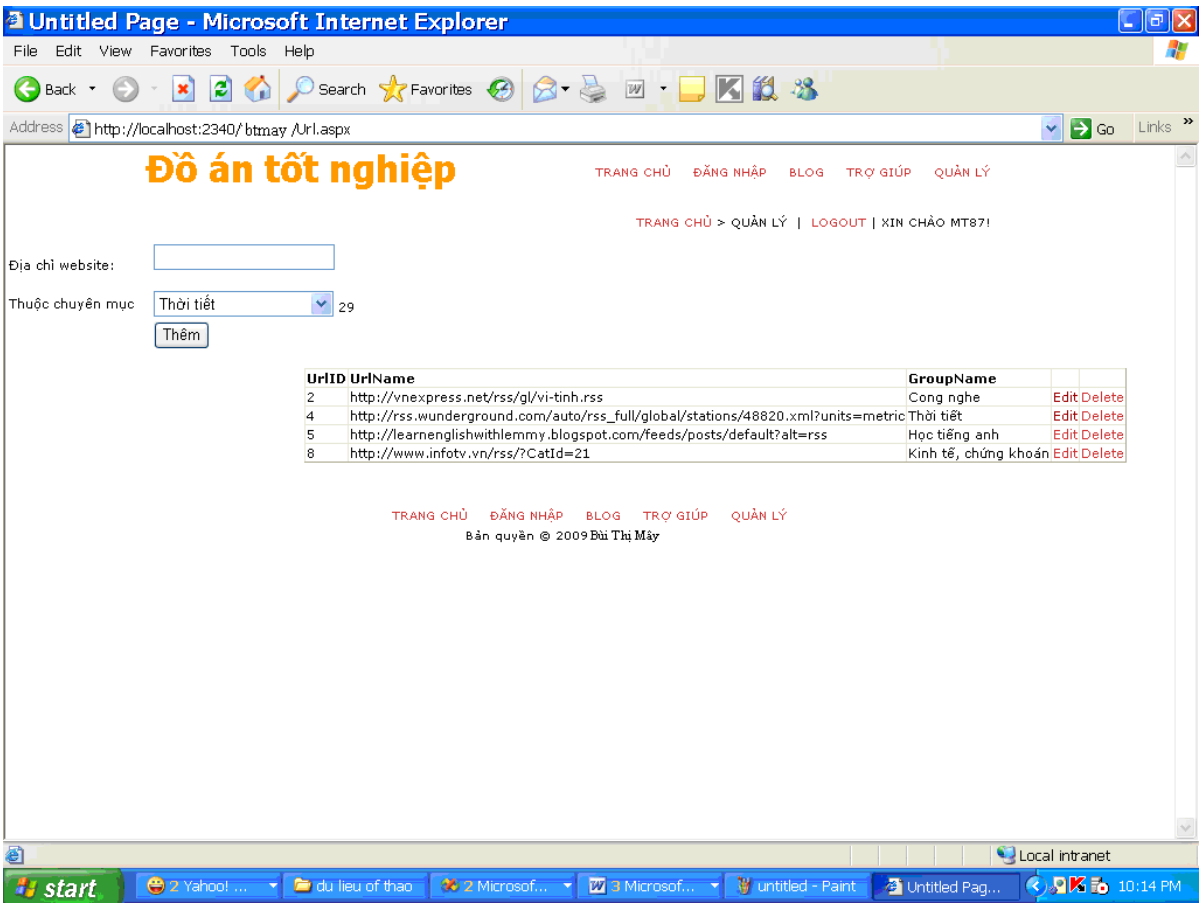


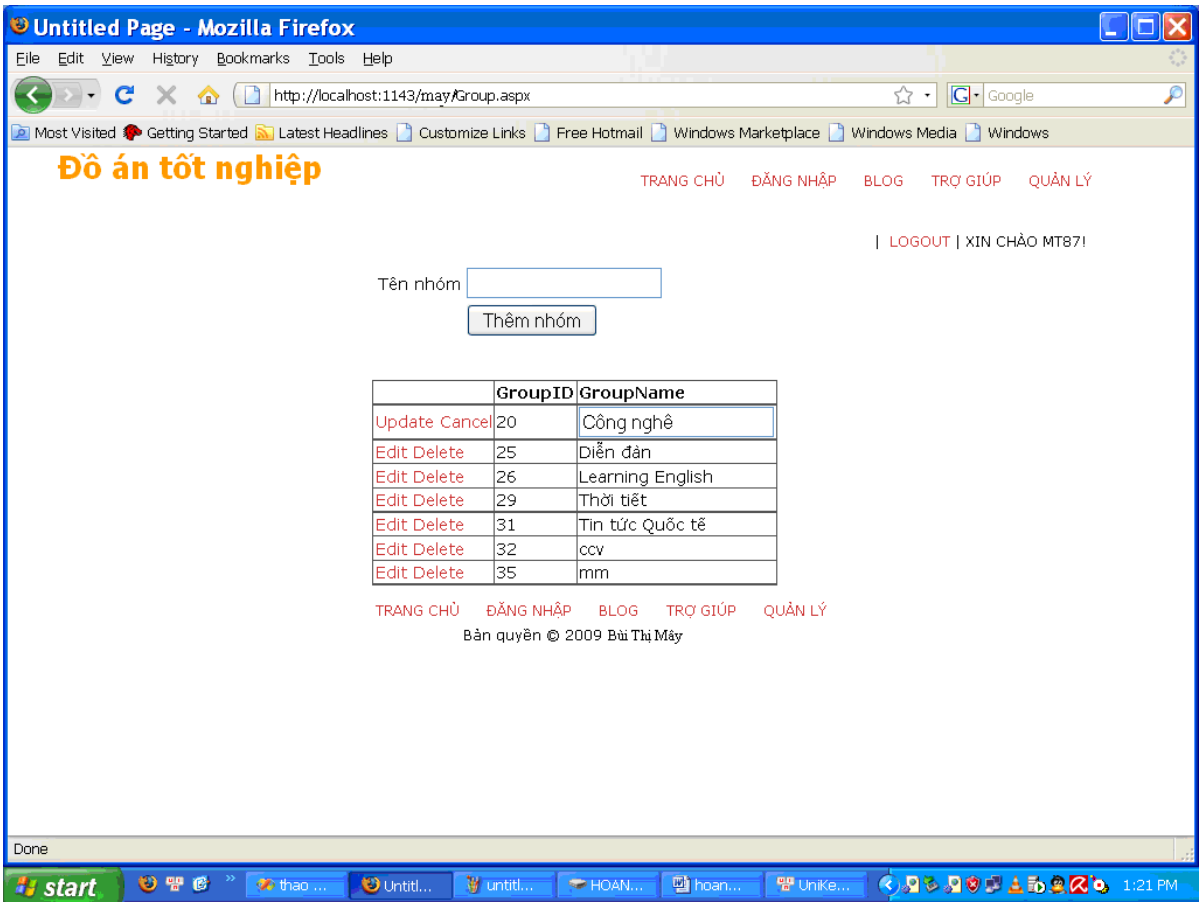
Hình 21 – Giao diện thư mục RSS cung cấp sẵn



Hình 22 – Giao diện trang lấy link RSS tự động

Hình 23 – Giao diện trang tin tức lấy về





Hình 24 – Giao diện trang quản lý nhóm tin

KẾT LUẬN VÀ PHƯƠNG HƯỚNG PHÁT TRIỂN

Trong quá trình tìm hiểu một số công nghệ XML, em thấy rằng XML là một chuẩn khá thân thiện, dễ đọc hiểu, là nền tảng để phát triển nhiều ngôn ngữ khác có ứng dụng cao trong đó có RSS. Hiện nay rất nhiều website chia sẻ tin tức đều dùng chuẩn RSS, nên việc xây dựng một chương trình hỗ trợ đọc tin RSS là điều hết sức cần thiết. Với sự nỗ lực và cố gắng, đồ án đã đạt được những kết quả sau:

Kết quả đã đạt được

- Hiểu và biết cách sử dụng một tài liệu XML trong vấn đề chia sẻ dữ liệu
- Biết cách đọc và ghi một tài liệu XML.
- Hiểu và biết cách sử dụng biểu thức chính qui trong việc tìm kiếm chuỗi.
- Xây dựng được website hỗ trợ đọc tin RSS với những chức năng cơ bản.
- Hiểu và nắm được kiến thức cơ bản XML trong .NET.
- Biết cách lập trình với ngôn ngữ C# .

Bên cạnh đó, chương trình còn có hạn chế

- Cách sắp xếp, và tổ chức tin tức chưa được tốt
- Chương trình chưa trau truốt vào giao diện.
- Chức năng của chương trình còn hạn chế.

Hướng phát triển của đề tài

- Khắc phục những hạn chế của chương trình.
- Tìm hiểu thêm công nghệ mới đó là AJAX, .NET Framework 3.5, tìm hiểu cách xây dựng một PORTAL , trọng tâm là PERSONAL PORTAL. Đây là một xu hướng khá phổ dụng trên thế giới hiện nay. Một số ví dụ điển hình như : iGoogle, My Yahoo,….. Với những kiến thức nền tảng công nghệ đó, em sẽ phát triển chương trình thành một PERSONAL PORTAL. Một PERSONAL PORTAL là nơi để mọi người có thể chia sẻ được thông tin,dữ liệu với nhau: như tin tức, tranh ảnh, video, thời tiết, chỉ số chứng khoán,……

TÀI LIỆU THAM KHẢO

[1] **Dương Quang Thiện**. .NET toàn tập - Tập 5: Lập trình Web dùng ASP.NET và C# - Nhà xuất bản Tổng hợp TP.HCM

[2] **Nguyễn Ngọc Bình Phương – Thái Thanh Phong**. Ebook: Các giải pháp lập trình C#. Nhà sách Đất Việt

Website

[3] http://www.w3schools.com/xml/

[4] http://msdn.microsoft.com/

[5] http://www.xml.com/

[6] http://www.codeproject.com

[7] http://www.asp.net

PHỤ LỤC

Phụ lục A - PHẦN CODE CHÍNH CỦA CHƯƠNG TRÌNH

Lớp Channel

```
namespace mayRSS
{
    public class Channel
    {
        // khai bao truong, thuoc tinh, ham tao
        private String _Link;
        public String Link
        {
            get { return _Link; }
            set { Link = value; }
        }
        public Channel(String link)
        {
            _Link = link;
        }
    }
  }
```

Lớp RSSItem

```
using System;
using System.Collections.Generic;
using System.Text.RegularExpressions;

namespace mayRSS
{
    public class RSSItem
    {

        // khai bao fields
        private string _Title;
        private string _Link;
        private string _Description;
        private string _Image;
        private string _Enclosure; // mp3, audio
        private DateTime? _PubDate;
        private string _AuthorName;


        // ham tao constructor
        public RSSItem(string title, string link, string desc,
  string img, string enclosure,
            DateTime? pubDate, string authorName)
        {
            _Title = title;
            _Link = link;
            _Description = desc;
            _Image = img;
            _Enclosure = enclosure;
            _PubDate = pubDate;
            _AuthorName = authorName;
        }

        // khai bao thuoc tinh properties

        public string Title
        {
            get{ return _Title; }
            set{ _Title = value;}
        }

        public string Link
        {
            get{ return _Link; }
            set{ _Link = value;}
        }

        public string Description
        {
            ge{ return _Description;}
```

```
            set{ _Description = value;}
        }
        public string Image
        {
            get{ return _Image;}
            set{ _Image = value;}

        }

        public string Enclosure
        {
            get { return _Enclosure; }
            set { _Enclosure = value; }
        }


        public DateTime? PubDate
        {
            get { return _PubDate; }
            set { _PubDate = value; }
        }

        public string AuthorName
        {
            get { return _AuthorName; }
            set { _AuthorName = value; }
        }

    }
}
```

Lớp LoadURL: lấy link RSS tự động

```
using System;
using System.Text.RegularExpressions;
using System.Collections.Generic;
using System.Net;
using System.IO;
using System.Data;

namespace mayRSS
{
  public class LoadURL
  {
    // khai bao field
    private bool _success = false;

    // khai bao thuoc tinh
    public bool Success
    {
      get { return _success; }
```

```
    }
    public Channel[] listChannel;

    // ham tao contructor
   public LoadURL(string linkWebsite)
    {
      Regex RegX = new
               Regex("<link.*type=\"application/rss\\+xml\".*"
               ,          RegexOptions.IgnoreCase);

      MatchCollection ChannelsOnWebsite =
               RegX.Matches(LoadHTML(linkWebsite));

      if (ChannelsOnWebsite.Count > 0)
            {
                listChannel  = new
Channel[ChannelsOnWebsite.Count];
                int fdi = 0;
                foreach (Match channel in ChannelsOnWebsite)
             {

listChannel[fdi]=FindLinkRSS(channel.Value.ToString ());
                  fdi++;
                }
                _success = true;

            }

        }
        // load noi dung trang web ve
  public static string LoadHTML(string linkHTML)
  {
    try
    {
      // Tạo yêu cầu.
      WebRequest obj = WebRequest.Create(linkHTML);
      // Lấy đáp ứng. công việc này sẽ lấy nội dung trang web
về
      WebResponse webRespone = obj.GetResponse();
      // Đọc đáp ứng (dạng stream).
      StreamReader sr = new
                    StreamReader(webRespone.GetResponseStream(
                    ));
      string result = sr.ReadToEnd();
      return result;
    }
    catch(Exception ex)
    {
       return null;

    }
  }
```

```
  public static Channel FindLinkRSS(string link)
  {
    string url = "";
    try
    {
      Match Url = Regex.Match(link , "(?<=href=).*");
      if (Url.Success)
      {
        int EndOfHref = Url.Value.ToString().IndexOf(" ", 1);
        if (EndOfHref == -1) { EndOfHref =
                        Url.Value.ToString().IndexOf("\"", 1);
                        }

        if (EndOfHref == -1) { EndOfHref =
                        Url.Value.ToString().IndexOf("'", 1);
                        }
        url = Url.Value.ToString().Substring(0, EndOfHref);
        url = url.Replace("\"", "").Replace("'", "");
       }
     }
     catch { url = ""; }
     return new Channel(url);
     }
  }
}
```

Phương thức đọc tập tin RSS

```
public static RSSItem ReadRSSItem(XmlReader subReader)
 {
   string title = null;
   string link = null;
   string desc = null;
   string img = null;
   string enclosure = null;
   DateTime? pubDate = null; //Nullable<DateTime>
   string authorName = null;
   subReader.MoveToContent();
   while (subReader.Read())
    {
      switch (subReader.Name.ToLower())
       {
         case "title":
           title = subReader.ReadString();
         break;
         case "link":
           link = subReader.ReadString();
         break;
         case "description":
           desc = subReader.ReadString();
           /* if (desc.Length > 250)
```

```
                { desc = desc.Substring(0, 250).ToString();
}*/
          break;
          case "image":
            subReader.MoveToContent();
            subReader.ReadStartElement("url");
          img = subReader.Value .ToString ();
            subReader.ReadEndElement();
          break;
          case "enclosure":
            subReader.MoveToContent();
            subReader.MoveToAttribute("url");
            enclosure = subReader.Value ;
            break;
          case "pubdate":
            string convert =
                     ConvertDateTime
.MDYtoDMY(subReader.ReadString());
            DateTime d;

            if (DateTime.TryParse(convert  ,out d))
             {
                pubDate = Convert.ToDateTime(convert );
             }
             break;
           case "author":
             authorName = subReader.ReadString();
             break;
          }
        }
             if (!String.IsNullOrEmpty(title)
                 && !String.IsNullOrEmpty(link)
                 && !String.IsNullOrEmpty(desc))
             {
                 RSSItem item = new RSSItem(title, link,
                     desc, img,enclosure, pubDate, authorName);
                 return item;
             }
             return null;
       }
```

Phụ lục B - TÌM HIỂU BIỂU THỨC CHÍNH QUI (REGULAR EXPRESSION)

Sử dụng biểu thức chính quy để bảo đảm dữ liệu nhập đúng cấu trúc và chỉ chứa các ký tự được quy định trước đối với từng dạng thông tin.

Biểu thức chính quy được xây dựng trên hai yếu tố: trực kiện (*literal*) và siêu ký tự (*metacharacter*). Trực kiện mô tả các ký tự có thể xuất hiện trong mẫu mà bạn muốn so trùng; siêu ký tự hỗ trợ việc so trùng các ký tự đại diện (wildcard), tầm trị, nhóm, lặp, điều kiện, và các cơ chế điều khiển khác. Dưới đây sẽ giới thiệu một số siêu ký tự thường dùng.

Siêu ký tự	**Mô tả**
.	Mọi ký tự trừ ký tự xuống dòng (`\n`).
\d	Ký tự chữ số thập phân (digit).
\D	Ký tự không phải chữ số (non-digit).
\s	Ký tự whitespace (như khoảng trắng, tab...).
\S	Ký tự non-whitespace.
\w	Ký tự word (gồm mẫu tự, chữ số, và dấu gạch dưới).
\W	Ký tự non-word.
^	Bắt đầu một chuỗi hoặc dòng.
\A	Bắt đầu một chuỗi.
$	Kết thúc một chuỗi hoặc dòng.
\z	Kết thúc một chuỗi.
|	Ngăn cách các biểu thức có thể so trùng, ví dụ `AAA|ABA|ABB` sẽ so trùng với `AAA`, `ABA`, hoặc `ABB` (các biểu thức được so trùng từ trái sang).
[abc]	So trùng với một trong các ký tự trong nhóm, ví dụ `[AbC]` sẽ

	so trùng với A, b, hoặc C.
[^abc]	So trùng với bất cứ ký tự nào không thuộc các ký tự trong nhóm, ví dụ [^AbC] sẽ không so trùng với A, b, or C nhưng so trùng với B, F,…
[a-z]	So trùng với bất kỳ ký tự nào thuộc khoảng này, ví dụ [A-C] sẽ so trùng với A, B, hoặc C.
( )	Xác định một biểu thức con sao cho nó được xem như một yếu tố đơn lẻ đối với các yếu tố được trình bày trong bảng này.
?	Xác định có một hoặc không có ký tự hoặc biểu thức con đứng trước nó, ví dụ A?B so trùng với B, AB, nhưng không so trùng với AAB.
*****	Xác định không có hoặc có nhiều ký tự hoặc biểu thức con đứng trước nó, ví dụ A*B so trùng với B, AB, AAB, AAAB,…
+	Xác định có một hoặc có nhiều ký tự hoặc biểu thức con đứng trước nó, ví dụ A+B so trùng với AB, AAB, AAAB,… nhưng không so trùng với B.
{n}	Xác định có đúng n ký tự hoặc biểu thức con đứng trước nó, ví dụ A{2} chỉ so trùng với AA.
{n,}	Xác định có ít nhất n ký tự hoặc biểu thức con đứng trước nó, ví dụ A{2,} so trùng với AA, AAA, AAAA,… nhưng không so trùng với A.
{n, m}	Xác định có từ n đến m ký tự đứng trước nó, ví dụ A{2,4} so trùng với AA, AAA, và AAAA nhưng không so trùng với A hoặc AAAAA.