

PHƯƠNG PHÁP THIẾT KẾ WEB ĐÁP ỨNG VÀ ỨNG DỤNG VÀO THIẾT KẾ WEBSITE KHOA CÔNG NGHỆ THÔNG TIN

Phạm Quang Dũng, Phan Thị Thu Hồng, Trần Thị Thu Huyền, Vũ Thị Lưu

Khoa Công nghệ thông tin, Học viện Nông nghiệp Việt Nam

Email: pqdung@vnua.edu.vn

Ngày gửi bài: 11.11.2014

Ngày chấp nhận: 17.05.2015

TÓM TẮT

Những năm gần đây cùng với sự phát triển bùng nổ của các thiết bị cầm tay như điện thoại thông minh, máy tính bảng và các thiết bị khác là xu hướng thiết kế web hỗ trợ cho thiết bị di động như: Tối ưu hóa chạm màn hình, tối giản các nút điều khiển, không gian trống, tăng cường video và hình ảnh, thiết kế học tập từ các ứng dụng (Apps), tương tác với mạng xã hội. Một trong những xu hướng thiết kế web mới nhất mang tính cách mạng và phát triển đáp ứng được những yêu cầu trên tại thời điểm này là thiết kế web đáp ứng (Responsive web design). Bài báo này trình bày các kết quả đạt được trong việc phân tích, thiết kế, cài đặt và triển khai thử nghiệm website phiên bản cho mobile của khoa Công nghệ thông tin sau khi đã tiến hành tìm hiểu về thực trạng sử dụng phương pháp thiết kế web đáp ứng tại một số đơn vị trong lĩnh vực Công nghệ thông tin ở Việt Nam cũng như trên thế giới.

Từ khóa: Mobile web, thiết kế web, web đáp ứng, thiết bị di động.

Responsive Design and Its Application to Designing Website of the Faculty of Information Technology

ABSTRACT

In recent years, there is an explosive growth of portable devices such as smart phones, tablets and other devices, the trend of web design that supports for mobile devices such as optimizing touch screens, minimizing control buttons and empty space, enhancing videos and graphics, designing learning applications, interacting with social networks. One of the newest web design trends and most revolutionary development to meet above-mentioned requirements at the moment is responsive web design. This paper presents the results obtained in the analysis, design, installation and deployment of mobile version of the Faculty of Information Technology's website after having carried out an investigation in the actual usage of responsive web design method at some organizations in the field of Information Technology in Viet Nam and around the world.

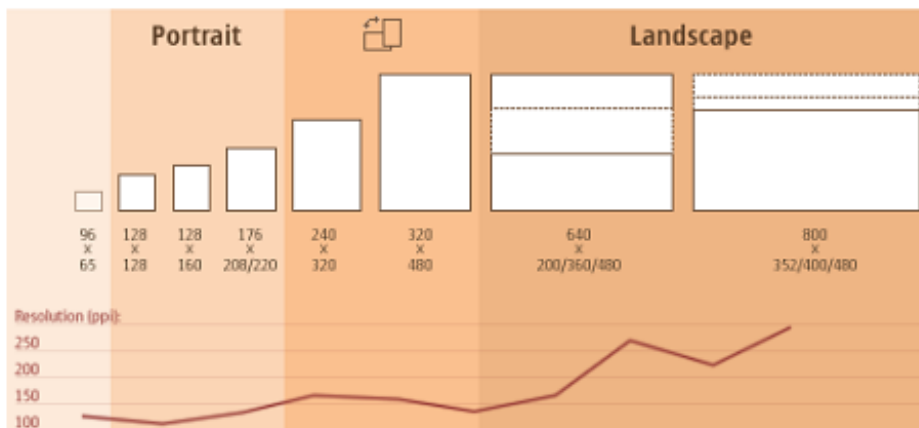
Keywords: Fluid layout, media queries, mobile web, portable devices, responsive web, web design.

1. ĐẶT VẤN ĐỀ

Ngày nay chúng ta sống trong một thời đại với một số lượng lớn các thiết bị khác nhau có thể kết nối với Internet. Không chỉ là máy tính để bàn nữa mà còn có điện thoại di động, iPhone, iPad, BlackBerry, máy tính xách tay, game platform, tivi, feed readers,... Dự kiến, trong những năm tiếp theo, số lượng thiết bị ra mắt trên thị trường với kích thước màn hình khác nhau có thể kết nối với Internet sẽ tăng

lên nhanh chóng. Khi thiết kế, các nhà thiết kế phải tính đến hàng trăm kích thước màn hình khác nhau cho cả màn hình nằm ngang và màn hình dọc. Morten Hjerde (2008) đã tiến hành xem xét một cơ sở dữ liệu thiết bị gồm khoảng 400 mẫu khác nhau được thu thập từ năm 2005 đến năm 2008 và thấy rằng các kích cỡ màn hình và định dạng tương ứng như hình 1.

Năm 2007, một cuộc khảo sát khẳng định, chỉ có khoảng 50% người sử dụng mở rộng tối đa cửa sổ trình duyệt web của họ (Johansson,

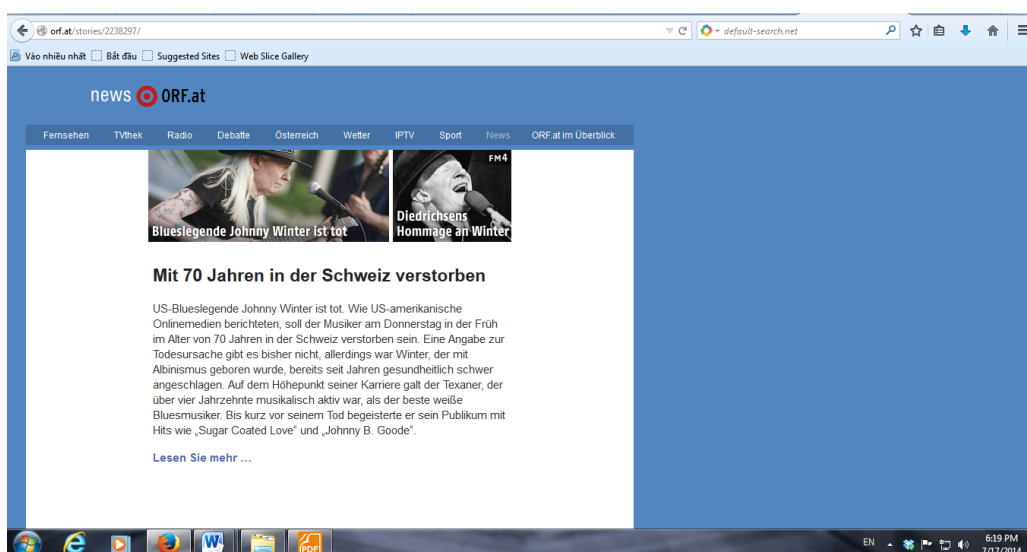


Hình 1. Kích thước màn hình và định dạng tương ứng sau khi khảo sát cơ sở dữ liệu của 400 mẫu thiết bị khác nhau

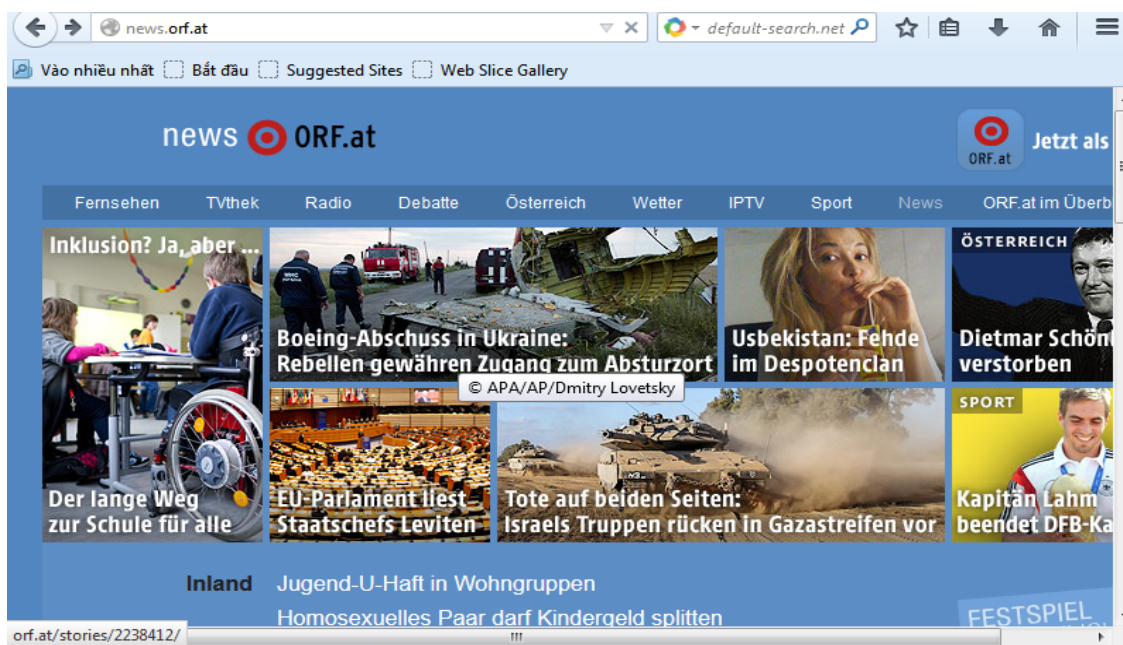
2007). Từ đó chúng ta có thể thấy rằng có một số lượng lớn các kích cỡ màn hình khác nhau được sử dụng nhưng có thể người dùng sẽ không thực sự tận dụng lợi thế của tất cả các không gian màn hình. Tuy nhiên nhiều nhà phát triển trang web không quan tâm đến vấn đề kích thước màn hình hiển thị. Họ không sử dụng các thiết bị với kích thước màn hình nhỏ hơn hoặc lớn hơn và không nghĩ đến người dùng của họ. Điều gì sẽ xảy ra khi một trang web được thiết kế cố định mà không có sự thay đổi linh hoạt trong không gian màn hình. Ví dụ xem trang web của ORF (trang web của đài phát thanh nước Áo) - một trang web được thiết kế với kích

thước cố định. Vì chiều ngang của website được thiết lập bởi một con số cố định nên các trang web này không thay đổi kích thước theo độ phân giải màn hình mà vẫn giữ nguyên một kích thước. Tuy nhiên, khi máy tính có độ phân giải cao hơn như 1024 hoặc 1280 điểm ảnh (chiều ngang) thì website không còn hiển thị toàn màn hình nữa mà sẽ dư ra một khoảng trống bên phải của website (Hình 2).

Hình 3 cũng chỉ ra website của ORF được thiết kế kích thước cố định nên khi kích thước này cao hơn độ phân giải màn hình có thể làm xuất hiện thanh cuộn ngang (horizontal scrollbar) gây khó khăn cho việc lướt web. Câu hỏi



Hình 2. Website được thiết kế với kích thước cố định, còn nhiều phần trống ở bên phải



Hình 3. Website được thiết kế với kích thước cố định nên khi kích thước này cao hơn độ phân giải màn hình có thể làm xuất hiện thanh cuộn ngang (horizontal scrollbar)

đặt ra có nên phát triển trang web với nhiều phiên bản khác nhau cho từng thiết bị? Nhiều nhà phát triển web thực hiện thiết kế hai phiên bản khác nhau cho trang web của họ: một cho máy tính cá nhân và một cho thiết bị di động, phiên bản dành cho thiết bị di động sẽ được truy cập qua tên miền khác (hoặc miền con). Những yêu cầu được gửi tới website sẽ được kiểm tra xem là từ thiết bị nào, nếu là thiết bị di động thì sẽ chuyển hướng sang tên miền được tạo riêng như trên. Ngoài việc phải tạo ra script để xác nhận thiết bị truy cập thì cách giải quyết ở trên còn gặp phải một nhược điểm khác nữa đó là ta luôn phải tạo ra hai khung nhìn khác nhau cho cùng một trang, và mỗi khi cần thay đổi một nội dung gì đó thì ta cần phải thay đổi cả hai. Rõ ràng đây không phải là một lựa chọn tốt, bởi số lượng các thiết bị hiện tại là rất lớn vậy nên giải pháp này đã phải giảm bớt một lượng người dùng trên một vài loại thiết bị nào đó. Hơn nữa, trên màn hình lớn, có không gian rộng hơn cho nhiều thông tin hơn trên màn hình nhỏ và một người sử dụng điện thoại di động có nhu cầu khác so với một người dùng máy tính. Do đó, hoàn toàn không thể thiết kế một trang web cho tất cả các thiết bị. Vậy còn một giải pháp nào

khác tốt hơn giải pháp phát triển trang web với nhiều phiên bản khác nhau? Các nhà thiết kế web phải tìm ra câu trả lời làm thế nào để trình bày các trang web trên màn hình với kích thước khác nhau và có thể chuyển đổi màn hình hiển thị ngang hoặc dọc tùy ý? Câu hỏi cho vấn đề này đã được bởi Ethan Marcotte một nhà phát triển và cũng là nhà thiết kế web trả lời vào năm 2011: Thiết kế Web đáp ứng [Marcotte, 2011]. Thiết kế Web đáp ứng là một kỹ thuật để phát triển một trang web duy nhất mà định dạng khác nhau cho các kích cỡ màn hình khác nhau để có thể sử dụng trên các thiết bị như máy tính để bàn, máy tính xách tay, máy tính bảng, điện thoại thông minh và các thiết bị khác. Điều này sẽ giúp các website chuyên nghiệp hơn, giảm thiểu được rắc rối khi cần cập nhật hay nâng cấp website khi dùng 2 phiên bản web riêng cho máy tính và cho thiết bị di động. Đây là một phương pháp thiết kế mới, mà việc làm chủ phương pháp cùng những kỹ thuật liên quan là động lực để các giảng viên ngành Công nghệ thông tin hướng đến, nhằm nâng cao trình độ chuyên môn, phong phú bài giảng và hướng dẫn sinh viên, cũng như ứng dụng vào việc phát triển các website theo hướng tiên tiến.

Chính vì lý do trên trong phạm vi nghiên cứu của bài báo này, chúng tôi đã lựa chọn phương pháp thiết kế web đáp ứng để triển khai website phiên bản cho mobile của khoa Công nghệ thông tin. Phần còn lại của bài báo chúng tôi giới thiệu phương pháp thiết kế và phát triển web cho mobile. Phần tiếp theo là bản phân tích thiết kế hệ thống cho website khoa Công nghệ thông tin và cuối cùng triển khai thực tế.

2. THIẾT KẾ WEB ĐÁP ỨNG

Trong phần này chúng tôi sẽ trình bày những kỹ thuật chính của phương pháp thiết kế web đáp ứng gồm: Fluid Grid, Fluid Image và Media Queries.

Fluid Grid

Một trong những kỹ thuật quan trọng nhất để đạt được thiết kế web đáp ứng là phải tạo ra một lưới linh hoạt (Fluid grid [Marcotte, 2009]). Trình duyệt web sẽ thay đổi theo kích thước và độ phân giải màn hình, nội dung tự động căn chỉnh phù hợp giao diện. Trước khi lưới linh hoạt trở nên phổ biến, hầu hết các trang web được thiết kế với chiều rộng cố định và lấy nội dung làm trung tâm, vì hầu hết các máy tính có độ phân giải màn hình tương tự. Nhưng bây giờ số lượng thiết bị truy cập Internet với độ phân giải màn hình khác nhau quá đa dạng, thiết kế chiều rộng cố định không phải là một giải pháp tốt cho một trang web (WebDesignShock, 2011),

vì thế “lưới linh hoạt” là một câu trả lời tốt nhất cho web hiện đại. Thay vì thiết kế một giao diện (layout) dựa trên số điểm ảnh (pixel) cố định hay giá trị phần trăm tùy ý, lưới linh hoạt (fluid grid) được thiết kế kỹ càng hơn theo tỷ lệ. Theo cách này, khi một giao diện được thu gọn trên màn hình điện thoại hay trải rộng trên một màn hình lớn, tất cả mọi thành phần trong giao diện đều tự thay đổi kích thước để tương thích với nhau. Để tính được tỷ lệ cho thành phần của từng trang, cần phải chia các nhân tố mục tiêu theo từng bối cảnh. Công thức tính tỷ lệ như sau: $\text{target/context} = \text{result}$. Ví dụ, nếu giao diện có kích thước bề ngang là 960 pixel, và bề rộng của thành phần mục tiêu là 200 pixel thì giá trị phần trăm sẽ là 20,83%.

Fluid Image

Thành phần thứ hai cần tối ưu trong thiết kế đáp ứng là hình ảnh. Với những màn hình nhỏ đơn giản là thay đổi kích thước hình ảnh thông qua việc điều chỉnh chiều dài chiều rộng ảnh. Điều này giúp nội dung và hình ảnh vẫn giữ được sự tổ chức hợp lý như ban đầu nhưng thời gian tải trang vẫn không được tối ưu hóa. Thời gian tối ưu hóa là rất quan trọng bởi vì trên các thiết bị di động kết nối Internet không nhanh như kết nối Internet ở nhà hoặc nơi làm việc (WebDesignShock, 2011). Có rất nhiều kỹ thuật dùng để tạo ra fluid image đáp ứng được với mọi kích cỡ màn hình khác nhau. Sau đây chúng tôi xin trình bày một số kỹ thuật thông dụng:



Hình 4. Tỷ lệ phần trăm của 200 pixel [WebDesignShock, 2011]

Sử dụng thuộc tính max-width được đề xuất bởi Richard Rutter

Một phương pháp để cung cấp một thay đổi kích thước hình ảnh linh hoạt là sử dụng các thuộc tính chiều rộng tối đa mà được đề xuất bởi Richard Rutter. Rutter thực hiện liên kết hình ảnh đến một phần tử cha với một kích thước xác định và thiết lập giá trị tối đa chiều rộng đến 100%. Nếu bây giờ tải trang web trên một màn hình với độ phân giải nhỏ, hình ảnh tải về lúc đó sẽ có kích thước ban đầu. Nếu độ phân giải màn hình lớn hơn so với hình ảnh gốc, chiều rộng tối đa của hình ảnh sẽ điều chỉnh để phù hợp kích thước màn hình (WebDesignShock, 2011).

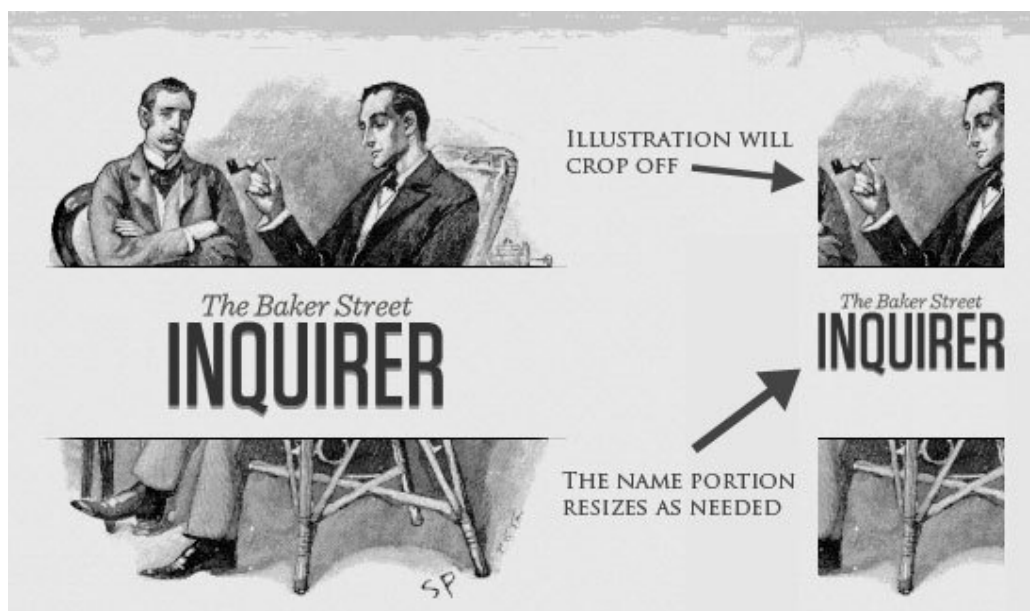
Sử dụng phương pháp cắt ảnh CSS

Nếu thay đổi kích thước ảnh quá nhỏ, hình ảnh sẽ xuất hiện với chất lượng thấp tuy nhiên chúng ta vẫn muốn hình ảnh đó sẽ xuất hiện trên web và không muốn bỏ ảnh đó đi vì ảnh đó quan trọng. Vì vậy, hình ảnh được chia thành 2 nhóm: một nhóm có thể cắt bỏ đi những phần không quan trọng và nhóm khác sẽ giữ nguyên ảnh nhưng sẽ phóng to lên hoặc thu nhỏ lại. Đó chính là phương pháp cắt ảnh CSS - một trong

những phương pháp để đạt được thay đổi kích thước động bằng cách sử dụng thuộc tính chiều rộng và chiều cao trong CSS. Trong kỹ thuật này một trong hai giá trị, chiều rộng hoặc chiều cao, được giữ giá trị cố định và yếu tố còn lại sẽ thay đổi linh hoạt. Nếu độ phân giải màn hình bây giờ thay đổi, phần linh hoạt của hình ảnh sẽ được cắt bỏ. Để khai báo bên nào sẽ được cắt bỏ phải hoặc trái, cần xác định giá trị của thuộc tính thay đổi của hình ảnh, (Gillenwater, 2009).

Sử dụng phân tử HTML5 <picture>, CSS3

Để đảm bảo thay đổi kích thước hình ảnh năng động và tối ưu hóa thời gian tải, nhà thiết kế phải sử dụng nhiều hình ảnh ở độ phân giải khác nhau. Phương pháp này sử dụng HTML5 và cs.S3 để xác định hai nguồn dữ liệu cho tất cả các hình ảnh. Một nguồn dữ liệu cho độ phân giải màn hình cao và một cho những màn hình có độ phân giải thấp hơn. Cùng với việc sử dụng kỹ thuật truy vấn phương tiện (media queries) có thể kiểm tra độ phân giải màn hình người sử dụng và lựa chọn nguồn dữ liệu phù hợp với người sử dụng.



Hình 5. Ví dụ về phương pháp cắt ảnh CSS. "The Baker Street Enquirer" của Ethan Marcotte. Khi cần hiển thị trên thiết bị màn hình nhỏ, ảnh phía bên trái sẽ được cắt gọn lại như phần bên phải

Nguồn: Kayla Knight (2011)

Media Queries

Lưới linh hoạt - fluid grid là một phần rất quan trọng khi tạo ra web đáp ứng. Tuy nhiên khi bề rộng của trình duyệt quá hẹp thì giao diện lúc đó có thể sẽ bị phá vỡ. Ví dụ, giao diện (layout) với nhiều cột không thích hợp cho màn hình nhỏ của điện thoại. Media queries - yếu tố thứ ba của thiết kế web đáp ứng có thể xử lý việc này. Media queries là một trong những module mới được thêm vào trong CSS3. Đây là một sự cải thiện của Media Type đã có từ CSS2, bằng việc thêm vào những cú pháp truy vấn để ta có thể đáp ứng được cho nhiều thiết bị với nhiều kích cỡ màn hình khác nhau. Module Media Query hiện đã được bổ sung đầy đủ trong các trình duyệt hiện đại như Webkit, Firefox, Opera hay IE (kể từ version 9). Media Queries có thể giúp nhận biết được thiết bị truy cập thông qua những thuộc tính của nó. Media Queries giúp chúng ta áp dụng những luật CSS khác nhau cho những thiết bị có kích cỡ màn hình khác nhau. Từ đó ta có thể đạt được mục đích của mình đó là hiển thị những nội dung khác nhau (cả hình ảnh hay văn bản) với những thiết bị khác nhau. Các nhà thiết kế web có thể tạo ra web với nhiều giao diện (layout) sử dụng các tài liệu HTML đơn và cung cấp chọn lọc stylesheets dựa trên các tính năng khác như kích thước trình duyệt, định hướng, độ phân giải và màu sắc. Mặc dù media queries cung cấp một giải pháp tuyệt vời cho việc phát triển các trang web đáp ứng cao, vẫn còn rất nhiều việc để làm để cung cấp một giải pháp hoàn chỉnh cho các thiết bị di động. [WebDesignShock, 2011].

Ví dụ sau cho chúng ta thấy một kiểu stylesheet khi màn hình thiết bị phù hợp với truy vấn.

```
<link rel="stylesheet" type="text/css"
media="screen and (max-device-width: 480
px)"
href="style.css" />
```

Ví dụ trên gồm 2 thành phần

1: Một kiểu màn hình,

2: Truy vấn thực trong ngoặc đơn, có chứa một thuộc tính media đặc biệt (giá trị chiều rộng lớn nhất của thiết bị) để kiểm tra, sau đó là giá trị mục tiêu (480px).

Trong ví dụ này chúng ta yêu cầu kích thước màn hình của thiết bị có giá trị nhỏ hơn hoặc bằng 480px. Nếu vượt quá giá trị này, ví dụ chúng ta sẽ kiểm tra hình ảnh của trang web trên Iphone sử dụng CSS style, ngược lại liên kết này sẽ bị bỏ qua. Ngoài ra chúng ta còn có thể sử dụng hai thuộc tính "min-width" và "max-width". "Chiều rộng tối thiểu" và "chiều rộng tối đa" là các thuộc tính thể hiện kích thước giới hạn thực của màn hình. Thuộc tính min-width thiết lập trình duyệt hoặc màn hình rộng tối thiểu là một tập hợp các phong cách (hoặc phong cách riêng biệt) sẽ áp dụng những loại màn hình đó. Nếu bất kì màn hình nào có kích thước nhỏ hơn mức giới hạn này, liên kết phong cách sẽ bị bỏ qua. Thuộc tính chiều rộng tối đa thì ngược lại, khi bất cứ kích thước nào lớn hơn giá trị kích thước tối đa thì không áp dụng cho các truy vấn media tương ứng (Knight, 2011).

Cú pháp của Media Query được mở rộng ra từ cú pháp của Media Type như sau:

```
@media media_type (feature:value) { rules }
```

Các thuộc tính feature và value sẽ giúp chúng ta xác định chính xác từng loại màn hình mà ta mong muốn. Đôi khi có những thuộc tính media feature đứng một mình không cũng có ý nghĩa, mà không cần đến value, thì cú pháp của Media Query sẽ như sau:

```
@media media_type (feature) { rules }
```

Để sử dụng Media Query hiệu quả, chúng ta có thể thực hiện theo 3 cách sau:

Sử dụng từ khoá @media để tính toán các luật bên trong một stylesheet:

```
@media (expression) {body {font-size: 15px}}
```

Khai báo trong tag link với thuộc tính là media. Như vậy ta có thể gọi đến một stylesheet bên ngoài nếu thỏa mãn điều kiện của media feature:

```
<link href="external files link"
rel="stylesheet" media="all and (expression)">
```

Sử dụng từ khoá @import để tải stylesheet từ một stylesheet bên ngoài:

```
@import url('file') all and (expression);
```

3. PHÂN TÍCH THIẾT KẾ HỆ THỐNG

3.1. Phân tích chức năng

Website Khoa Công nghệ thông tin được thiết kế theo hướng tiên tiến, hỗ trợ phân quyền cập nhật cho một số tài khoản chuyên trách để dễ có được lượng thông tin cập nhật và phong phú. Tuy nhiên trước khi chính thức đăng tải thì các bài được duyệt bởi tài khoản có quyền cao hơn. Một số chức năng chính của website như sau:

- Website có giao diện song ngữ: tiếng Việt và tiếng Anh
- Phân quyền duyệt bài viết và cập nhật nội dung cho một số tài khoản, ví dụ của Ban chủ nhiệm Khoa.
- Phân quyền cập nhật nội dung các trang web tương ứng cho các bộ môn, các giáo viên, các trợ lý khoa...; phân quyền đăng tin tương ứng cho các trợ lý, các tổ chức đoàn thể. Khi các tài khoản viết tin mới sẽ ở chế độ Chờ duyệt cho đến khi một tài khoản duyệt, chỉnh sửa rồi chính thức đăng bài lên.
- Tìm kiếm tin tức, nội dung theo từ khóa trong các trang thuộc website.

- Thống kê lượt truy cập, số người trực tuyến

- Mỗi trang đều có Tiêu đề hoặc Mô tả (có nội dung trong thẻ <Title>, <Description>)

Biểu đồ phân cấp chức năng dưới đây sẽ cho thấy chi tiết hơn các chức năng của website Khoa Công nghệ thông tin.

3.2. Phân tích ngữ cảnh

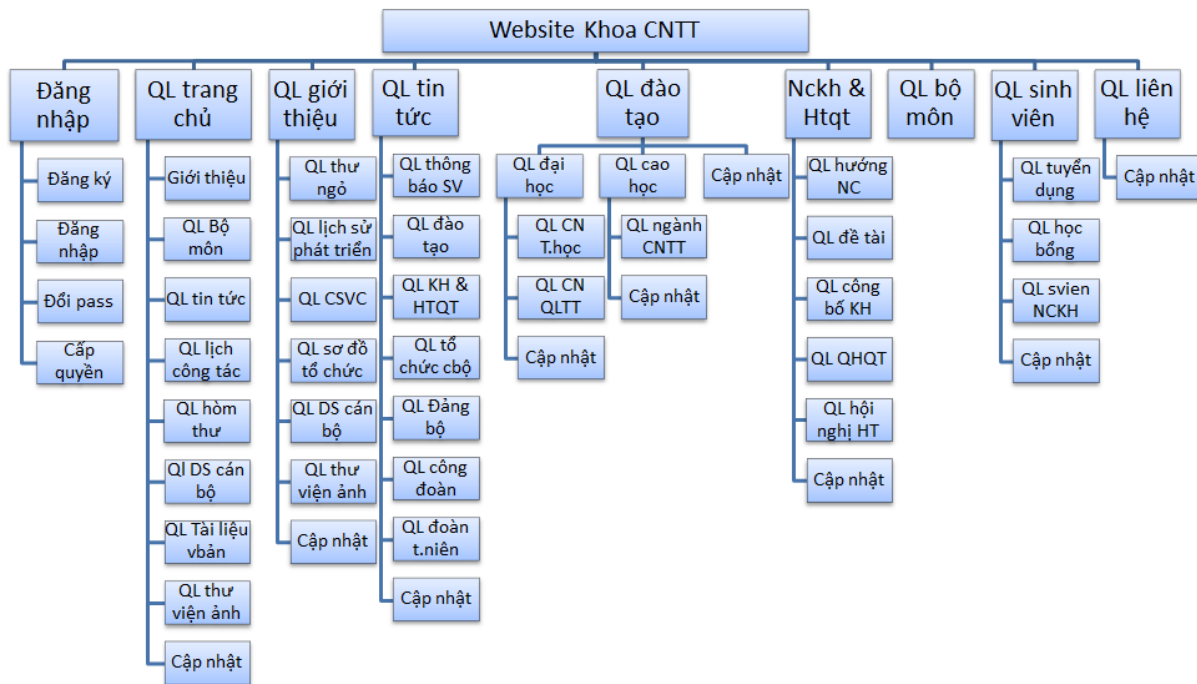
Có 4 nhóm tác nhân ngoài tham gia vào quá trình hoạt động của hệ thống: Admin; BCN Khoa; Trợ lý/giảng viên; User. Mỗi tác nhân có phạm vi hoạt động và quyền riêng của mình.

- Admin với quyền quản trị cao nhất, quản lý và điều phối hoạt động của các tác nhân trong hệ thống.

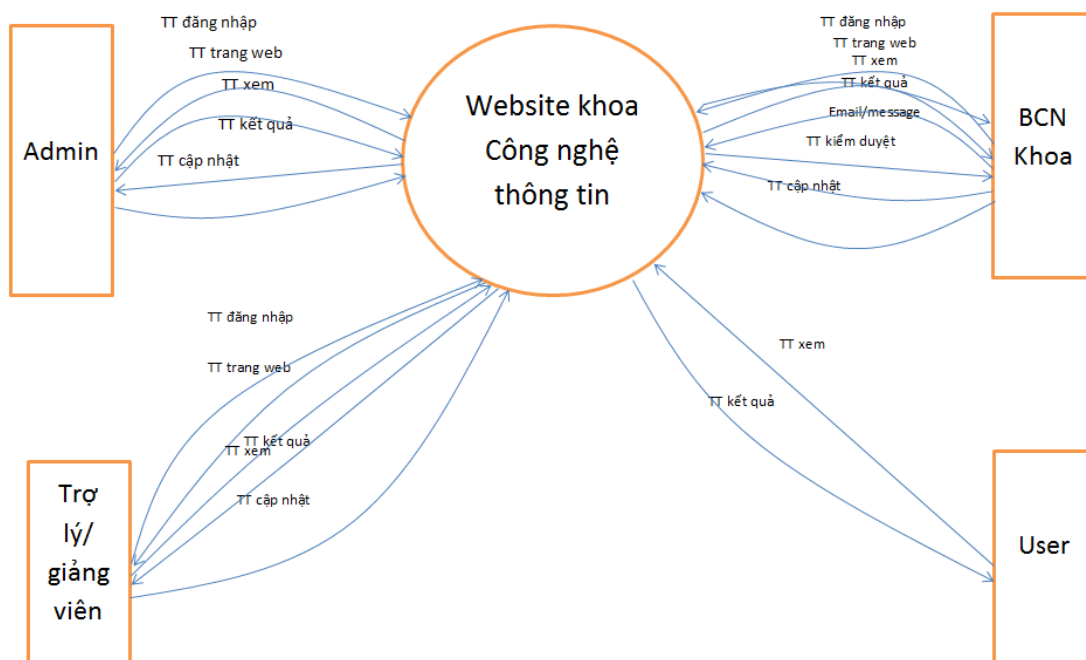
- BCN Khoa có quyền quản trị cấp 2, ngoài những hoạt động cơ bản còn có nhiệm vụ quản lý các hoạt động của tác nhân bên dưới.

- Trợ lý/giảng viên có quyền quản trị bên dưới BCN Khoa, và được cấp quyền hoạt động theo chức năng nhiệm vụ của từng đối tượng.

- User là tác nhân không cần tên truy cập và mật khẩu, có quyền thấp nhất là xem và tìm kiếm.



Hình 6. Biểu đồ phân cấp chức năng



Hình 7. Biểu đồ luồng dữ liệu mức ngữ cảnh

4. KẾT QUẢ VÀ THẢO LUẬN

Website Khoa Công nghệ thông tin được thiết kế và xây dựng theo phân tích thiết kế nêu ở phần trên, có giao diện đáp ứng với 4 khoảng độ phân giải chiều ngang màn hình:

- Từ 960 điểm ảnh trở lên, ví dụ máy tính cá nhân, máy tính xách tay
- Trong khoảng 768 - 960 điểm ảnh, ví dụ máy tính bảng
- Trong khoảng 480 - 768 điểm ảnh, ví dụ máy tính bảng có độ phân giải màn hình thấp hơn
- Dưới 480 điểm ảnh, ví dụ điện thoại di động

Chúng tôi đã áp dụng nhiều kỹ thuật thiết kế web đáp ứng để xây dựng website Khoa Công nghệ thông tin. Cụ thể là:

4.1. Hình ảnh

Khi load giao diện sẽ đặt thuộc tính `img max-width: 100%` (độ rộng lớn nhất = 100% tỉ lệ màn hình hiện tại cho phép hiển thị, trường hợp một số ảnh bị tràn là do thiết lập riêng của nội dung bài viết đó => cần chỉnh lại).

```
.service img { max-width: 100%; height: auto; }
```

4.2. Media queries

Có 4 loại giao diện theo khoảng độ phân giải màn hình. Với độ phân giải chiều ngang lớn hơn 960 thì mặc định sử dụng giao diện PC

4.2.1. @media only screen and (min-width: 768px) and (max-width: 960px)

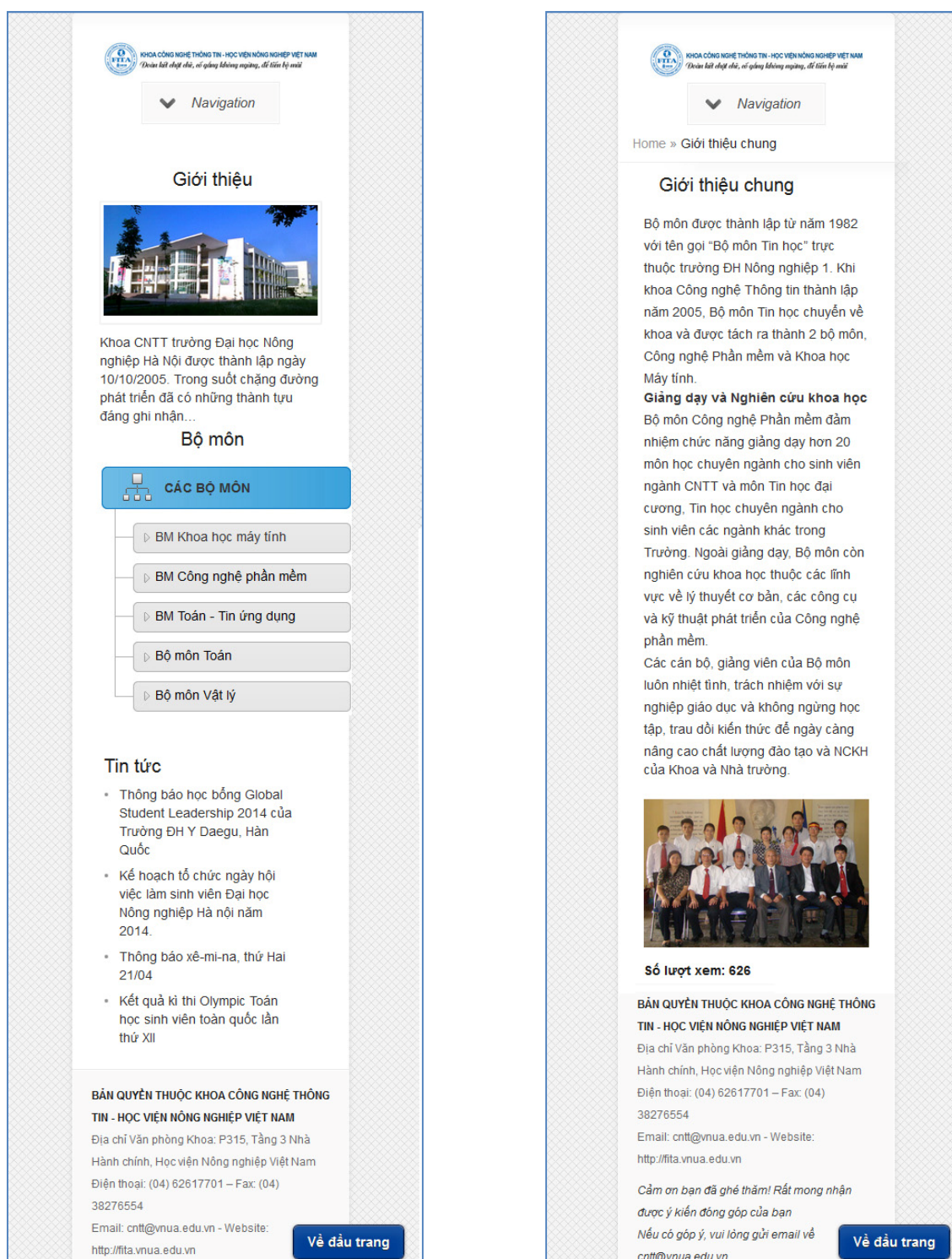
```
#left-area { width: 487px; }
#sidebar { width: 155px; padding-left: 37px; }
#content { padding: 0px 36px 60px 31px; }
background-position: 518px 0px; }
#footer-content { width: 748px; }
```

4.2.2. @media only screen and (max-width: 768px)

```
#top-menu, #sidebar { display: none; }
#content-area { padding: 35px 36px 10px; }
#services { margin-bottom: 10px; }
.service { float: none; width: 386px; }
.service h3.title { text-align: center; }
#footer-content { width: 458px; }
```

4.2.3. @media only screen and (max-width: 480px)

```
#top-menu, .post-thumbnail { display: none; }
#mobile_menu { width: 240px; }
```



Hình 8. Thiết kế giao diện web Khoa CNTT đáp ứng với màn hình có độ phân giải chiều ngang nhỏ hơn 480 điểm ảnh - ví dụ: điện thoại di động

Ghi chú: Hình trái: trang chủ; Hình phải: một trang tin



Hình 9. Thiết kế giao diện web Khoa CNTT đáp ứng với màn hình có độ phân giải chiều ngang trong khoảng 480-768 điểm ảnh - ví dụ: máy tính bảng

Ghi chú: Hình trái: trang chủ; Hình phải: một trang tin



Hình 10. Thiết kế giao diện trang chủ web Khoa CNTT đáp ứng với màn hình có độ phân giải chiều ngang lớn hơn 960 điểm ảnh - ví dụ các máy tính cá nhân, laptop

```
#container { width: 300px; }
#content-area { padding:35px 30px 10px; }
    #services { margin-bottom: 10px; }
        .service { float: none; width: 240px; }
#footer-content { width:298px; }
}
```

4.2.4. Lớn hơn 960px

Mặc định sử dụng giao diện PC.

4.3. Fluid layout, fluid grid

Các kỹ thuật fluid layout, fluid grid được áp dụng vào chính 4 khoảng giao diện nêu trên

- Dành cho điện thoại: max 480px,
- Dành cho Tablet có độ phân giải thấp: 481 - 768px (hiển thị dạng xoắn dọc),
- Dành cho Tablet có độ phân giải cao: 769 - 960px (hiển thị dạng nằm ngang),
- Dành cho PC: lớn hơn 960px

Dưới đây là một số giao diện website Khoa Công nghệ thông tin đã được thiết kế đáp ứng tự động với độ phân giải của màn hình thiết bị truy cập.

5. KẾT LUẬN

Phương pháp thiết kế web đáp ứng là phương pháp mới, có ưu điểm linh hoạt đáp ứng theo độ phân giải màn hình của thiết bị truy cập web. Nhà thiết kế không phải làm nhiều phiên bản web tương ứng với nhiều loại thiết bị mà chỉ cần thiết kế một số giao diện cho một phiên bản web duy nhất. Điều này rất có ý nghĩa khi chủ website có nhu cầu thay đổi nội dung trang web, nhà thiết kế sẽ chỉ phải chỉnh sửa một lần thay vì nhiều lần như phương pháp truyền thống. Hơn nữa, tốc độ đáp ứng web của máy chủ nhanh hơn, tiết kiệm dung lượng lưu trữ của máy chủ.

Trong bài báo này, chúng tôi đã tìm hiểu thực trạng thiết kế web, những ưu nhược điểm của phương pháp thiết kế web truyền thống và phương pháp web đáp ứng. Những kỹ thuật áp dụng trong thiết kế web đáp ứng đã được nghiên cứu cẩn thận và áp dụng tối đa vào xây dựng website Khoa Công nghệ thông tin. Website đã được phân tích thiết kế và xây dựng chức năng phù hợp với một đơn vị giáo dục, khoa học. Bốn khoảng giao diện ứng với các thiết bị truy cập đã được tham khảo cho phù hợp với thiết kế của nhiều website khác. Website mới của Khoa Công nghệ thông tin đã được vận hành từ tháng 5/2013, tại địa chỉ <http://www.vnua.edu.vn/khoa/fit/>, ngày càng được hoàn thiện. Chúng tôi cũng đã triển khai thử nghiệm web đáp ứng của Khoa tại cùng địa chỉ trên, và đã nhận được những phản hồi tích cực từ phía những người sử dụng.

Với sự nắm bắt tốt những kỹ thuật của phương pháp thiết kế mới và sự thành công khi triển khai thử nghiệm sản phẩm của đề tài, chúng tôi sẽ giới thiệu phương pháp này tới nhiều đồng nghiệp và sinh viên ngành Công nghệ thông tin qua các buổi seminar, trao đổi chuyên môn, các bài giảng, cũng như hướng dẫn nghiên cứu. Chúng tôi cũng đề xuất lãnh đạo Học viện và Ban Khoa học Công nghệ tiếp tục ủng hộ trong việc ứng dụng phương pháp thiết kế web này với những website của Học viện và các sản phẩm thực tiễn khác.

TÀI LIỆU THAM KHẢO

- Hjerde, Morten (2008). Mobile screen size trends. <http://sender11.typepad.com/sender11/2008/04/mobile-screen-s.html>
- Jeffrey L. Whitten, Lonnie D. Bentley (2007). Systems Analysis and Design Methods. McGraw-Hill.
- Joe Casabona (2013). Responsive Design with WordPress: How to make great responsive themes and plugins. New Riders.
- Johansson, Roger (2007). Poll results: 50.4% of respondents maximise windows. http://www.456bereastreet.com/archive/200704/poll_results_504_of_respondents_maximise_windows/
- Jonathan Fielding (2014). Beginning Responsive Web Design with HTML5 and CSS3. Apress. <http://coding.smashingmagazine.com/2011/01/12/guidelines-for-responsive-web-design/> (truy cập ngày 25/11/2013)
- Knight, Kayla (2011). Responsive Web Design: What It Is and How To Use It. <http://www.smashingmagazine.com/2011/01/12/guidelines-for-responsive-web-design/>.
- Marcotte, Ethan (2011). Responsive Web Design. A Book Apart Edition. Jeffrey Zeldman. ISBN 978-0-9844425-7-7. <http://www.abookapart.com/products/responsive-web-design>.
- Luke Wroblewski (2011). Mobile First. Ingram.
- Marcotte, Ethan (2009). "Fluid Grids". A List Apart. <http://alistapart.com/article/fluidgrids>
- Nguyễn Văn Ba (2002). Phân tích và thiết kế hệ thống thông tin. Nhà xuất bản Đại học Quốc Gia.
- TAKA (2014). Thiết kế website responsive. <http://www.lamweb.edu.vn/thiet-ke-website-responsive/>
- Thoriq Firdaus (2013). Responsive Web Design by Example. Packt Publishing.
- WebDesignShock (2011). Responsive Web Design, most complete guide. <http://www.webdesignshock.com/responsive-web-design/>.