
Responsive website design for higher education utilizing mobile centric features

Deanna Klein, Minot State University, deanna.klein@minotstateu.edu

Aleksandar Gubic, aleksandar_gubic@yahoo.co.uk

Abstract

Technology utilized by students in Higher Education is quickly moving from a desktop or laptop to other various sized devices and varying specifications. This trend is causing those in higher education to consider how to provide a pleasing experience when viewing their university website on various sized screens. The recent response to the trend is through the development of Responsive Website Design. This paper will present a process taken to optimize a College of Business website for a mobile environment, a review of the current state of mobile web implementation in higher education and thirdly, recommendations for mobile centric features that would further enhance the student experience.

Keywords: Responsive Web Design, Mobile Centric, Technological Forecasting, Higher Education

Introduction

The popularization of technologies such as 3G and Wi-Fi, as well as the increase in performance and size of mobile devices, enabled users to browse websites much like they would do on a desktop or laptop. Modern mobile web browsers function just as well as their desktop counterparts. They are able to display websites written in languages such as HTML, CSS, PHP, ASP.NET, JavaScript, etc., and are supported by the ever-growing use of high-speed mobile networks (Seymour, Hussain & Reynolds, 2014). This ensures that users have a suitable replacement for their desktop web browsers or, in some cases, can solely rely on mobile web browsers.

However, one major difference in using a mobile web browser compared to a desktop browser is screen size and resolution. Today there is a plethora of mobile devices with different specifications on the market. They differ in screen sizes, resolutions, processing power, and other features. When compared to desktop-based computers, it is clear that mobile devices can be different from one another in a wide array of features. Where desktop-based computers differ in, for example, processing power, monitors have been produced in roughly the same screen sizes and have been supporting high resolutions for years. In fact, as early as January 2009, at least 57% of users were viewing websites with resolutions higher than 1024 x 768, which today is the resolution of the iPad Mini. Furthermore, that number has increased to 90% in January 2013 (W3Schools.com, n.d.). High resolutions of desktop-based monitors are not a concern when building a responsive website, as websites with a maximum width of, for example, 1080px are viewed with ease on a monitor with a resolution of 2560 x 1600. Therefore, the focus is on smaller resolutions, as they go hand in hand with screen sizes on mobile devices.

The main goal of this project is to modify the existing website so that it adapts to the screen size and features of the device it is viewed on. As the current layout is best viewed at higher resolutions and with a desktop-based browser, this adaptation to smaller devices ensures that all users have a pleasant experience while viewing the website no matter which resolution they view it on. This project will be carried out in three stages.

The first stage is designing multiple resolution versions of the website. Based on the principle of “mobile first” design, the design process will revolve around the smallest determined resolution and, as the resolution increases, will adapt the content accordingly. The second stage provides a look into the current trends in higher education mobile website development. These trends can be used to inspire the implementation of certain features and design principles. The third stage provides a set of recommendations in regards to implementing mobile centric features to the website. Upon successful improvements to the website, by way of making it respond to different screen sizes and resolutions, the focus shifts on features. These features are mobile centric, which means that they will be best used on mobile devices.

Background

The existing website of the Minot State University College of Business serves to provide an array of information to students, faculty and staff. It has been built primarily with desktop-based web browsers as the preferred applications for viewing. It is content centric, with limited end user interaction. With the increasing use of mobile devices, there is a genuine need to optimize the website for a mobile environment. This optimization comes in the form of making the website responsive, which means to make it adapt to various screen sizes and features of mobile devices. Making the website responsive also opens up new possibilities of implementing mobile centric features, thus enabling the website to function like a mobile application. What this could bring to end users is a truly pleasant web browsing experience that can also serve to host future implementation of various types of features. The purpose of this paper is to document the development process of the Minot University College of Business responsive website. It presents design choices for specific screen sizes, discusses the technology used, outlines implemented features, and provides insight into possibilities for further development.

Current trends in higher education mobile website development

Published reports indicate that at least 65% of people aged 18-29 are accessing the Internet using their mobile device which is causing immediate attention from university marketing departments and enrollment services (Smith, 2010). In recent years, the number of institutions of higher education implementing a mobile website has increased dramatically with many more institutions expected to get on board in the near future. Having said that, a recent study indicates that only 37 of the 345 Division I colleges and universities were reported to have a mobile version of their website (Darvick, 2010). Many institutions contemplate moving to a mobile application but struggled with the expense for the development costs.

However, many universities have responded to the student demand for mobile centric web access through the development of a responsive website (Asher, 2014). A responsive website will avoid the less desirable experience of students browsing their traditional website on a mobile device.

By choosing a responsive website solution, institutions of higher ed. are addressing the behaviors of their current and perspective students while accommodating for the resolution required for the various popular mobile devices.

While it is apparent that implementing a responsive website is a solution to meeting the needs of students utilizing a variety of mobile physical devices as well as a variety of device platforms, the specific features students desire to access through their mobile device remains uncertain. The research provides a plethora of features desired by a consumer or general user of mobile devices; however, there is still a lack of literature to address the features most desired by the student population in higher education.

Design

The first step in making the existing website responsive is designing the layout. As stated earlier, the process of designing a responsive website has to be based on the principle of “mobile first” in order to achieve the best results. This principle ensures that all of the original content is properly laid out on the smallest determined resolution and that, as the resolution increases, the layout design evolves to accommodate the current resolution using CSS media queries.

When designing a responsive layout it is important to consider which elements need to be included in order to maintain the familiar feel when switching devices. For this project, there are two determined resolutions based on the extent of use. Even though these resolutions represent “breaking points” at which the design is subject to a wider array of changes, users will be able to have a pleasant browsing experience even when viewing the website on a non-listed resolution. This is due to the implementation of a fluid layout. “Using media queries alone to change a design is fine if we have a specific known target device. But this strategy isn’t really future proof. At present, when we resize our viewport, the design snaps at the points that the media queries intervene and the shape of our layout changes. However, it then remains static until the next viewport “break point” is reached” (Frain, 2012).

The listed resolutions that represent “breaking points” are expressed in pixels and are accompanied by examples of devices that use them. The determined resolutions are:

320 x 480 (ACER E310, Apple iPhone, Apple iPod Touch, HTC Legend, Motorola Devour, Samsung Galaxy Ace),

1024 x 768 (Apple iPad, small desktop monitors, etc).

When designing a responsive website it is important to first consider the lowest resolution it will be viewed in. The next step is to design the elements of the website and to accommodate the content, and to ensure that, as the resolution increases, so does the size of the elements. This enables a smooth transition.

Working within a small space necessitates assigning priorities to elements that are displayed at the top of the page.

In this case, these elements are, as displayed in Fig 1.:

- The MSU logo

- Tagline
- Menu icon
- The title “College of Business”
- Main images and/or video
- Page title and/or first paragraph



Fig 1. The top of the website viewed in the smallest determined resolution

The four elements that are constantly displayed throughout the website are: the logo, tagline, menu icon and the title “College of Business”. The decision to retain these four elements throughout the website has been made in order to ensure the user has a sense of familiarity while browsing. Other elements listed above are subject to a change in order depending on the page. Some pages display images or video followed by text, while others display text first, followed by an image or video.

This prioritization only serves to help with the decision of which elements will be initially displayed and does not force any subtraction of content. Instead, the content has been placed vertically and is accessible by scrolling. Vertical scrolling has been enabled to ensure that the rest of the content is easily accessible. On touch-based interfaces scrolling is carried out by simply dragging ones finger vertically.

Much like the displayed elements at the top of the page, there is a set of elements constantly present at the bottom of the page. They are placed with the intent of providing options for further browsing. These elements are, as displayed in Fig 2.:

- Menu set 1 (containing links to about us, faculty & staff, academics, colleges & departments, student services, athletics, videos and contact us pages)
- Menu set 2 (containing links to centennial website, future students, current students, distance education, alumni & donors, campus connection and calendar of events pages)

- CoB calendar
- Facebook page link
- Site map
- Privacy policy
- Additional information

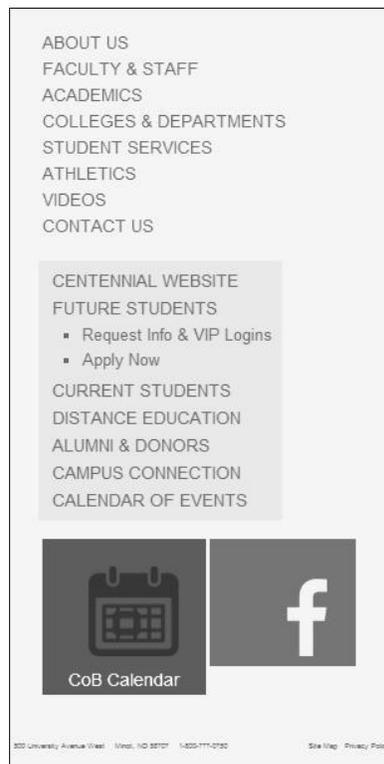


Fig 2. The bottom of the website viewed in the smallest determined resolution

The decision to place these elements at the bottom of the page is due to the very nature of their function. As the user scrolls down while viewing the content, in most cases, they require additional information. On desktop-based devices this issue is usually resolved by positioning these elements on the side or on the top of the page to minimize scrolling. By positioning them at the bottom, the user is offered additional information after viewing the content. The two significantly larger links are the “CoB Calendar” and “Facebook like page”. They are viewed as independent from both menu sets 1 and 2, and are emphasized by their size due to the potential frequency of their use.

Viewport of 320 x 480 pixels

For this project the lowest determined resolution is 320 x 480 pixels. This is the lowest most commonly used resolution on mobile devices such as the Apple iPhone, Samsung Galaxy Ace, HTC Legend, etc. (This is Chemistry, n.d.).

Portrait orientation (320 x 480)

The landscape orientation of devices that use the resolution of 320 x 480 pixels limits the width to 320 pixels. While images span through the width of the page, to make use of the space, the text is formatted in such a way to properly break into multiple rows, ensuring the user has a comfortable viewing experience.



Fig 3. Portrait orientation (320 x 480)

Landscape orientation (320 x 480)

When the device is rotated to the landscape orientation, the viewport, in this instance, switches to 480 x 320 pixels. That means that the width is slightly increased. While there are no major changes to the top of the page or the content, these elements are stretched to accommodate the wider space available. The only major change occurs in the bottom of the page. Because of the new space available, the two menu sets can now be positioned horizontally instead of vertically as seen in Fig 4.



Fig 4. Landscape orientation (320 x 480)

Viewport of 1024 x 768 pixels

The resolution of 1024 x 768 pixels represents the second major breaking point. As this resolution is big enough to make viewing of the original version of the website comfortable (Fig 5.), no major changes were applied to the design. The choice to make this particular resolution a breaking point rests on the fact that the specified width and height is sufficient enough to accommodate the elements of the website in their original form. Thus, there was no need to adapt them for this resolution. If used on devices such as tablets, switching orientations results in the full usage of set widths as the website scales to accommodate the space.

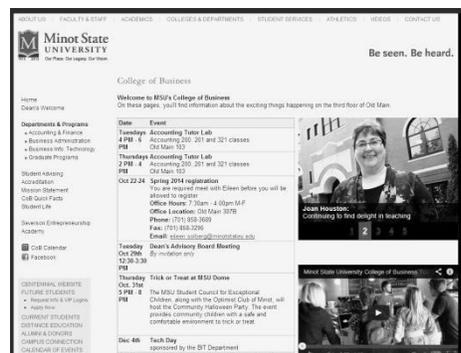


Fig 5. Original version (1024 x 768)

Viewports in between

For the purpose of this project, as mentioned above, there are two major breaking points. While there is an array of mobile devices supporting different resolution, the decision to design the website around two breaking points rests on the ability to support any resolution in between with a fluid layout. This type of layout has a major advantage over the fixed layout due to the ability

to adapt to any screen size. That means that if a user was to manipulate the size of the browser and shrink it down below the second breaking point, the content would adapt accordingly. Likewise, if a user was to enlarge the browser beyond the second breaking point, not only would the original design be preserved, but the advantage of the fluid layout would once again come into the spotlight as the website would easily adapt.

Source code analysis

The design process, as previously described, involves the placement of elements depending on the resolution the website is viewed in. Upon determining which elements will be included, as well as their appearance, the next step consists of coding the website. In the case of this project, the original design was made previously available; therefore, a re-design was not necessary. Instead, the main focus laid exclusively on lower resolution versions.

During the process of designing the responsive website it was essential to make some modifications to the original source code. These modifications affected both the HTML and CSS code. This was done to ensure that the elements and content are displayed correctly.

The first major modification comes in the form of converting the layout dimensions from pixels to percentage. Converting these values enables a transition from a fixed to a fluid layout. In order to successfully create a responsive website it is essential to initially carry out the aforementioned conversion due to the constant change in viewport size. For example, the width of the page element “wrapper” in a fixed layout is written in the form of

```
#wrap {width: 956px; }      (1)
```

This indicates that there the wrapper (with the id “#wrap”) is set to a fixed width of 956 pixels (1). Such a setting is acceptable if the website is not intended to be viewed on a device with a screen width of less than 956 pixels, however, attempting to view the website on a device with a smaller screen width would result in a scaled version of the website. The user would then have to constantly zoom-in, zoom-out, and scroll in order to successfully navigate the website. While websites with a fixed width, viewed on most modern mobile devices, are typically identical as if viewed on a desktop, the need to constantly manipulate the zoom and scrolling causes inconvenience for the user.

Fixing this issue, with regards to the example noted above, involves converting the width of the page “wrapper” to the form of

```
#wrap {width: 96%; }      (2)
```

With the width now converted into percentages, the element will cover 96% of the entire width of the page regardless of the screen size and viewport (2). The value of 96 has been set in order to ensure that the element does not cover the entire page. This was the same reason the fixed width was set to 956 pixels, due to the fact it was designed around the determined resolution with a 1024 pixel width.

Fluid layouts, however, have their limitations. If the page is scaled down to a certain point, the appearance of elements and content may be deemed as inadequate. In order to make the website usable at lower resolutions as well, it is necessary to make it respond to this type of change. As

described earlier, there are two major breaking points described with media queries. The 320 x 480 pixels media query is written as

```
@media screen and (max-width: 480px){} (3)
```

The maximum allowed width for the first breaking point is set to 480 pixels (3). Under this CSS rule, elements will appear as described when viewed on screens that have a maximum width of 480 pixels. After the first breaking point has been reached, the maximum width is no longer considered to be 480, but 1023 pixels, written as

```
@media screen and (min-width: 481px) and (max-width: 1023px){} (4)
```

In a similar, yet slightly different scenario, the set maximum width of 1023 pixels. However, due to the fact that the previous breaking point had a maximum width of 480 pixels, in order to avoid conflict between the two breaking points, there has to be a defined minimum width as well (4).

Defining breaking points for smaller resolutions implies a change in the element structure in comparison to the original version of the website. The most notable change, in this instance, is the introduction of a new, mobile friendly, navigation system. This navigation system is called upon to be displayed depending on which resolution the website is currently being viewed in. The first two breaking points describe the navigation system as

```
#nav-top-mobile {padding: 5px 1.04%; margin: 0 5% 10px; font-size: 1em; display: block;}
```

```
#nav-mobile {display: block; float: left; padding: 5px 3.68421%; margin: 0 5% 10px; list-style: none; background: #e5e5e5; line-height: 1.666666em; font-size: 1em; width: auto;}
```

```
#social-mobile {display:inline-block;float:left;padding: 5px 0; margin: 0 5% 10px; list-style: none; line-height: 1.666666em; font-size: 1em; width:90%;}
```

```
#main-mobile-nav {display:none; width: 100%;background: #c03;list-style: none;float:left;padding:5px 0 5px;}(5)
```

The navigation system consists of four menu sets (#nav-top-mobile, #nav-mobile, #social-mobile, and #main-mobile-nav). Each of these menu sets has properties such as margin, padding, float, line-height, font size, background, and width (5). However, one key property that determines whether or not the menu set will be visible is “display”. For this, mobile friendly, navigation system is only described under the breaking points that target lower resolutions, the “display” property has to be set to “block”.

Once the second breaking point has been reached, the website reverts to its original version. This means that elements such as the mobile friendly navigation are no longer relevant and therefore must be replaced with the original navigation, written as

```
#nav-mobile {display: none;}
```

```
#nav-top-mobile {display: none;}
```

```
#social-mobile {display: none;}
```

```
#main-mobile-nav {display: none;}
```

```
#nav-top {display: inline;}
```

.sub-site-nav {display: block} (6)

Setting the “display” property to “none” removes the mobile friendly navigation and replaces it with the original navigation (#nav-top and .sub-site-nav) (6).

Recommendations for future integration of mobile centric features

Upon the analysis of current trends in higher education mobile website development, it is apparent that developing mobile centric features for this purpose represents a step forward in harnessing the power of mobile devices. This section describes the third stage of this project. It provides a set of recommendations of mobile centric features. These features would be implemented within the responsive website. As described earlier, converting the existing website into a responsive website ensures a pleasant viewing experience on a wide array of mobile devices. It eliminates drawbacks such as the need to constantly zoom in and out in order to view the content. Stage one described the theory behind this process and how it delivers a better overall browsing experience for the end-user.

The new responsive design can also serve to host a set of mobile centric features that can be used to enhance productivity on campus. This section also provides a detailed description of the proposed features.

Online chat support

The feature of online chat support is present in a plethora of commercial websites. It is primarily used to ensure a quick response to user queries. This method is a step forward in the field of support. While the use of email still remains a reliable method of responding to user queries, its disadvantages, such as relatively slow response time, are even more evident when used on mobile devices.

Online chat represents a quick and easy way to seek support when using a mobile device due to the fact it is based on instant messaging. The use of this feature is highly intuitive due to the very nature of messaging on mobile devices.

The implementation of this feature is demonstrated in the following example. The user logs in with the intent of submitting a query to student service support. They select the option from the main menu and are put on hold until one of the student support service staff members responds to the request. After a text confirming the response pops up on the screen of the user’s device, the user and a student support service staff member are connected within a private chat session environment. Here the user can submit any relevant query and get an instant answer.

News stream

The ability to provide a constant flow of relevant information is essential in order for the user to keep up to date with current events. However, since users typically do not spend as much time reading the content on a mobile device as they do on a desktop based device, it is important that only the most relevant information is delivered.

To demonstrate this streamlining of information, the proposed solution is to deliver content based on different pre-determined parameters. In this case, the pre-determined parameters could include, but not be limited to, department, interests, age, and others.

An example of how this feature could be put to use is shown in the following example. The user logs in and selects the option to view the latest news, after which, a number of articles populate the screen. The user notices that all of these articles have one or more parameter, such as department, related to their profile settings and does not receive any unwanted information. By setting up different parameters within their profiles, users could receive only relevant information.

Comments

Comment features offer the ability of users to post their opinions related to a certain piece of content, whether it is an article, notification, multimedia, etc. With the growing popularity of social media also comes the ability to integrate their application programming interfaces (API) alongside the existing comment system.

For the purpose of this project, implementation of a comment feature could come in the form of allowing users to express their opinions on a certain news article.

Remote printing

Remote printing is the ability of a user to submit one or more documents to be printed at a convenient location, online, without having to physically transfer the documents to the printing device. This feature requires active staff members who would monitor the operation of the server and printing device itself.

An example of implantation of this feature is described in the following scenario. The user logs in and selects the “remote printing” option, where upon they are prompted to submit one or more files (with a reasonable file size limit). After submission, the documents are uploaded to the main server and are queued for printing. Sorting of documents is done by date and time, and priority is given to the earliest submissions. Once the printing is complete, the user receives a notification stating that their documents are ready for pickup.

Technology

The responsive website is based on HTML, CSS as well as JavaScript. HTML is used for markup and, for the purpose of this project, contains the new semantic tags introduced in version 5. CSS implements styles to the pages and designates the breaking points described above. As the responsive website contains animated elements, such as the slide show on the index page, it was necessary to write their source code in the appropriate language. These elements are written in JavaScript.

The proposed mobile-centric features require additional technology in order to achieve the desired functionality. As the main idea behind these features, in other words, this web app, is OS independence as well as the focus on open source, the language chosen to handle functionality is

PHP. The method of coding chosen is object oriented programming because of its ease of implementation. For the purpose of storing data, a MySQL database was selected for implementation. The front end of the web app is best developed in jQuery Mobile for its focus on mobile devices. These technologies are fairly standard and common when it comes to web development. However, the Online chat and support feature requires the use of even more advanced technology. Because it is necessary to have a quick and responsive messaging service, particularly on mobile devices, the simple approach of constant requests for updates (usually handled by JavaScript) has been deemed as inadequate due to concerns regarding bandwidth. Instead, the use of Node.js, and its extension Socket.io, would greatly improve overall performance and responsiveness. Their implementation is somewhat more difficult; however, the benefits of their use outweigh the initial cost.

Conclusions

According to the current trends in higher education mobile website development, there is an increase in the use of mobile devices which introduces new possibilities of how students can connect with their universities. While creating a responsive website is only the first step in bringing the university experience to mobile devices, development has to go further beyond into an exploration of various types of features in order to truly harness the power of these new technologies.

The features proposed in this paper showcase the potential of increasing productivity on campus. However, they are by no means the limit of what can be achieved. Their implementation, testing, and use can help to introduce new ideas which in turn can lead to even bigger advancements in higher education mobile website development. With additional expectations from university students for performance utilizing a mobile device, it is recommended that further research includes a survey of students to identify and prioritize features to include in the design of the mobile centric responsive website environment, specifically to meet the interests of students in higher education.

References

- Asher, J. (2014). Moovweb and Forrester Research Release Enterprise Mobility Report Survey of Responsive Web Design Project Outcomes. Retrieved from <http://finance.yahoo.com/news/moovweb-forrester-research-release-enterprise-140000738.html>. March 29, 2014.
- Darvick, E., (2011). *Mobile Websites in Higher Education*. Earth Media Group. Retrieved from <http://www.slideshare.net/EarthboundMediaGroup/mobile-websites-in-higher-education>
- Frain B, (2012). *Responsive Web Design with HTML5 and CSS3*.
- Seymour, Hussain & Reynolds. (2014). How to Create an App. *International Journal of Computer and Information Systems*. Retrieved from <http://journals.cluteonline.com/index.php/IJMIS/issue/current>. March 29, 2014.
- Smith, A., (2010). Mobile Access 2010. Pew Internet & American Life Project, July 7, 2010. Retrieved from

http://www.pewinternet.org/~media/Files/Reports/2010/PIP_Mobile_Access_2010.pdf
on November 26, 2013.

This is Chemistry, (n.d.) *Browser and resolution statistics for web designers*. Retrieved from <http://visual.ly/browser-and-resolution-statistics-web-designers>.

W3Schools.com, (n.d.) *Browser display statistics*, Retrieved from http://www.w3schools.com/browsers/browsers_display.asp

Biographies

Deanna Klein is an associate professor in the College of Business, Department of Business Information Technology. Deanna teaches undergraduate and graduate classes in the area of systems analysis and design, project management and Projects in MIS. She is a professional member of the International Association for Computer Information Systems (IACIS), and Informing Science Institute (ISI), is currently International Assembly for Collegiate Business Education (IACBE) Region 5 Past President, as well as a member and Past President of Delta Kappa Gamma – Gamma Chapter.

Aleksandar Gubic is a teacher of Technical Education at the Ivan Goran Kovacic Junior High in Subotica, Serbia. His field of work includes IT, teaching and research and his recent work has been centered around the implementation of various aspects of technology in education.