

A MOBILE WEBSITE PROTOTYPE

A Paper
Submitted to the Graduate Faculty
of the
North Dakota State University
of Agriculture and Applied Science

By

Ashish Teotia

In Partial Fulfillment of the Requirements
for the Degree of
MASTER OF SCIENCE

Major Department:
Computer Science

November 2012

Fargo, North Dakota

North Dakota State University
Graduate School

Title

A MOBILE WEBSITE PROTOTYPE

By

ASHISH TEOTIA

The Supervisory Committee certifies that this *disquisition* complies with
North Dakota State University's regulations and meets the accepted standards
for the degree of

MASTER OF SCIENCE

SUPERVISORY COMMITTEE:

DR. KENDALL NYGARD

Chair

DR. DEAN KNUDSON

DR. RICHARD RATHGE

DR. TARIQ KING

Approved:

11/29/2012

Date

DR. KENDALL NYGARD

Department Chair

ABSTRACT

More people have started using smart phones, and in the near future, cell phones/smart phones will take the place of computers. I developed a mobile website prototype based on standards, such as keeping it task oriented, by filtering and showing the menu items for North Dakota State University Computer Science students (prospective, current, or alumni); shrinking the data by using smaller images; making the data fit the screen, eliminating the need to scroll the page; and many more. I followed a test-and-learn approach to develop this prototype, utilizing a design that works well on mobile phones and utilizing the same tools used in traditional web design and development.

ACKNOWLEDGEMENTS

I would like to express my gratitude by thanking everyone who has supported me during the completion of this research paper. I would like to thank Dr. Kendall Nygard for guiding me all these semesters. His support and vital encouragement contributed a lot to my research. My sincere thanks also go to the other three members of my graduate committee for their patience and understanding with the production of this paper. I would like to thank Dr. Dean Knudson who gave me an insight about the “lifecycle of a project.” I would like to thank Stephanie Sculthorp (Administrative Secretary) of the Computer Science Department for constant reminders to complete my paper on time. Last but not least, I would like to thank my family, especially my parents, Nisha Teotia and Narendra Teotia, and brother, Anuj Teotia, for guidance, love, much needed support, and making it possible to study at North Dakota State University.

TABLE OF CONTENTS

ABSTRACT.....	iii
ACKNOWLEDGEMENTS.....	iv
LIST OF FIGURES	viii
LIST OF ABBREVIATIONS.....	ix
CHAPTER 1. PROBLEM STATEMENT AND INTRODUCTION.....	1
Problem Statement.....	1
Introduction.....	1
CHAPTER 2. LITERATURE REVIEW	3
Goal 1.....	3
Goal 2.....	3
Goal 3.....	4
CHAPTER 3. OBJECTIVES AND TASKS	5
Objective 1.....	5
Objective 2.....	5
Objective 3.....	5
Objective 4.....	5
Objective 5.....	6
Objective 6.....	6
Task 1.....	6

Task 2.....	6
Task 3.....	6
Task 4.....	7
Task 5.....	7
CHAPTER 4. WIREFRAMES	8
Wireframe 1.....	8
Wireframe 2.....	9
Wireframe 3.....	10
CHAPTER 5. USE CASES	12
CHAPTER 6. ARCHITECTURE AND TECHNOLOGIES.....	13
Technologies Used	14
CHAPTER 7. WORK.....	15
Menu Simplification.....	15
CHAPTER 8. TESTING.....	17
Testing Task 1.....	17
Testing Task 2.....	17
Testing Task 3.....	17
CHAPTER 9. SCREENSHOTS	21
CHAPTER 10. QUESTIONS LEFT UNANSWERED.....	25
CHAPTER 11. CONCLUSION AND FUTURE WORK	26

Conclusion.....	26
Future Work.	27
REFERENCES	29
APPENDIX. CODE.....	31

LIST OF FIGURES

<u>Table</u>	<u>Page</u>
1. Fundamental Wireframe for Home Page.	8
2. Wireframe for Showing News or Other Information as a Slider.	9
3. Wireframe for Showing Checklist to Graduate Students.	10
4. Wireframe With Various Filters for Showing Specific Menu According to the Choice Made.	11
5. Use Case for Mobile Website Prototype.	12
6. Two-Tier Client Server Architecture.	13
7. Header for Secondary Pages for NDSU.	15
8. Current Menu on NDSU CS Department website.	16
9. MoileOK Test Results.	18
10. MobiReady Test Results.	19
11. Mobile Website Readiness Test Results.	20
12. Home Page of the Prototype Website.	21
13. Expanded Menu from More Text Button.	22
14. Slider Page for News or Other Information.	23
15. Mobile Website Page for Showing Menu According to Various Filters.	24

LIST OF ABBREVIATIONS

CEO.....	Chief Executive Officer
CS.....	Computer Science
CSS.....	Cascading Style Sheets
GIF.....	Graphic Interchange Format
GUI.....	Graphical User Interface
HTML.....	HyperText Markup Language
IE.....	Internet Explorer
JPEG.....	Joint Photographic Experts Group
KB.....	Kilobyte
NDSU.....	North Dakota State University

CHAPTER 1. PROBLEM STATEMENT AND INTRODUCTION

Problem Statement. Develop a prototype for a student-oriented, mobile website for the North Dakota State University Computer Science Department.

Introduction. This paper provides the explanation and the steps I followed to develop a prototype mobile version of the current desktop website for the NDSU Computer Science (CS) Department. The paper also describes the challenges faced while designing the Graphical User Interface (GUI).

I developed a prototype website for the Computer Science Department. One good feature of mobile website development is that the site is fundamentally built with the same tools used for traditional web design and development. The use of same tools makes mobile website development more approachable and easier to maintain than application (app) development [13]. Many users want to visit a mobile website on the go, without downloading a mobile app. Also, a mobile website can be viewed on any mobile phone with a browser (with some cascading style sheet [CSS] changes for different browsers).

This paper explains about how I designed the prototype, keeping in mind various users, such as current students, prospective students, and graduate students. All these users need not necessarily see the same menu. For example, prospective students will want to know more about the programs offered by the department than the current students and graduate students.

In order to make the mobile website prototype, I followed three goals (The details are given in Chapter 2.): (1) keep the navigation simple, (2) make the website task-oriented, and (3) limit the content. In order to achieve these goals, I set several specific objectives. (The details are given in Chapter 3.) I used objectives such as (1) making sure that the prototype looks similar on devices with various screen sizes, (2) maintaining compatibility on the major mobile web

browsers, (3) making the website well structured so that it is easy for the user to view or browse, (4) organizing the content to be easily accessible on a small screen, (5) providing checkboxes or radio buttons for easy data input by the user, and (6) keeping the size of the webpages small in order for them to load quickly.

In order to make sure that I met my goals and objectives, I followed certain tasks. (The details are given in Chapter 3.) I classified my tasks as follows: (1) not using Flash because it is no longer supported by Adobe on mobile phones [14], (2) not using cookies, (3) avoiding the use of a table layout that sometimes makes content difficult to read, (4) making the size of the page relative as a percentage rather than fixed in pixels because we can easily have layout issues on a bigger screen, and (5) using fonts such as Arial or San Serif because they are supported by all browsers.

This paper is organized into 11 chapters, starting with the Problem Statement and Introduction. Chapter 2 is the Literature Review, followed by Chapter 3 which explains the Objectives and Tasks. Chapter 4 shows the Wireframes that I used in designing the website. Chapter 5 gives Use Cases and explains the actors involved. Chapter 6 gives the Architecture and Technologies, with Chapter 7 illustrating the design and Chapter 8 outlining the Testing. The code is in Appendix A. Screenshots are provided in Chapter 9, unanswered questions in Chapter 10, and the Conclusion and Future Work in Chapter 11.

CHAPTER 2. LITERATURE REVIEW

After years of anticipation, smart phones users, devices, networks, and brands have all caught up with each other. Now, there is an emphasis on 128 x 160 pixels, and mobile use is growing at an ever-faster pace [1]. After reading the literature on the mobile web and app development over the internet, I came to a conclusion that building websites optimized for mobile devices is very similar to standard site development, but it is also different in terms of certain factors that take on a much more significant role with a mobile website. For example, screen-size variations, user attention spans, and usability issues are more critical. (Limited space is a huge factor.) Mobithink.com did a survey [7] and asked various mobile-marketing experts (Senior Director, Global Marketing Advertising Yahoo, CEO of Novarra, etc.) about the things they think are essential for good mobile websites and ways to improve the sites.

These same issues are also present for desktop-based websites, but the issues are sometimes easier to overlook on a desktop. I will include some lessons learned about the optimization that is happening on the mobile web as well as how we design and think about traditional web design and website architecture. Goals for the project are given in the following sections.

Goal 1. Simplify the navigation on the mobile website [15] in comparison to the full website. The need for extreme simplification of the navigation is to one to the first things that becomes evident when considering mobile websites. Navigation not only becomes very prominent and important on a mobile website, but it is also often trimmed down substantially to focus on only the most critical elements.

Goal 2. Make the website task oriented. People visit mobile websites with a task-oriented mindset, so the most visited features should be shown first. .

Goal 3. Shrink the content in comparison with a full site. It is obvious that mobile websites must shrink their content. Not only are the number of options reduced to the core functionally and purpose of the website, but the copy is also vastly reduced. The whole content should still be accessible by providing an option to see the entire website or to provide/classify data in some way. Content is vital, but it needs to be in small bites.

We assert that we have many lessons to learn from mobile-web development issues, particularly the need to focus on the necessary elements. To design a mobile website, certain design principles, as explained in the Objectives and Tasks chapter, should be kept in mind.

CHAPTER 3. OBJECTIVES AND TASKS

This chapter provides the specific objectives for the project. The chapter also includes the tasks necessary to meet the objectives.

Objective 1. Make the mobile website look similar on devices with various screen sizes. To make sure I achieve that goal, I avoid using tables as the main layout because a user needs to scroll horizontally and vertically to read the content (because the available white space available is smaller) [2].

Objective 2. The mobile website must be compatible with the standard browsers (Firefox, Chrome, Safari, and Internet Explorer). To ensure maximum compatibility, I used Cascading Style Sheets (CSS) [8] because I could easily modify the property of a HyperText Markup Language (HTML) [9] element by changing the structure of the entire page. This compatibility is possible because almost all browsers have CSS capabilities.

Objective 3. In order to ensure a well-structured mobile website, I had to consider that different mobile phones have different screen sizes. This limitation can be a major issue if you want your mobile website design to run on all of them. I structured the mobile website within 128 x 160 pixels [1].

Objective 4. The information on the website must be organized properly in order make it accessible and easy to read because of the smaller screen. In order to ensure readability, the most important information for mobile users to see must be at the top of the page, and if there is more information than you can fit easily on one screen, then a summary and a link to read more must be given [3]. It can be time consuming to browse through a mobile site, not to mention tedious to read through it. This design organization makes the mobile site convenient for users.

Objective 5. Accepting users' input on a mobile website could become difficult.

Therefore, I used radio buttons and lists for the users' convenience [4]; users can choose what they need. I also used a JPEG or GIF format for images on the mobile website because the popular mobile browsers support these formats.

Objective 6. When accessing a mobile website, users need to pay data charges. To minimize the data usage I tried to keep the page size less than 20 kilobytes [16]. Lastly, I had to make sure that users are getting their money's worth by providing what they need in less time. (See Figure 4 for details.)

The first thing we should remember when writing pages for the smart-phone market is that we do not have to make any changes. The great thing about most available smart phones is that they use Webkit browsers (Safari on iOS and Chrome on Android) to display web pages, so if a page looks okay in Safari or Chrome, it will look fine on most smart phones (just a lot smaller). There are things we can do to make the browsing experience more pleasant (dividing pages into smaller chunks, etc). There is a list of things that we should avoid when designing a mobile website. The tasks include the following items:

Task 1. Make use of javascript as it is supported by all the standard browsers and avoid using Flash as all of the mobile phones do not support Flash (Even Adobe is not working on mobile Flash anymore [14].), so it is not a good idea to include it on your wireless pages.

Task 2. Make the website store no data on users device. In order to achieve that I avoided creating cookies on the device because many cell phones have limited-to-no cookie support.

Task 3. Make the website easy to read and requires less scrolling. For this I did not use tables for the layout of a mobile page. I tried to avoid tables in general as they are not supported on every cell phone (although iPhones and other smart phones support them), and you can end

up with strange results. (See Objective 1 for more details.) If we have to use a table layout, we need to make sure not to nest it in another table. These table layout are difficult for mobile browsers to support and, at best, make the page load more slowly.

Task 4. Absolute measures should be avoided. In other words, do not define the dimensions of objects in absolute sizes (such as pixels, millimeters, or inches). If you define something as 100px wide, it might be half the screen on one mobile device, and on another, it might be two times the width. Relative sizes (like em's [10] and percentages) work best.

Task 5. Make sure to use a font which is available on most mobile browsers (such as Arial or San Serif). Do not assume that any fonts you are used to having will be available on cell phones.

Based on the above objectives and tasks, I designed a prototype mobile website for the North Dakota State University Computer Science Department. I also evaluated some existing websites based on the above objectives and tried to optimize my website design based on my findings or tasks.

CHAPTER 4. WIREFRAMES

In this chapter, I will discuss various website Wireframes.

Wireframe 1. Figure 1 illustrates a wireframe that can be used as a home page. Menu items could display the most frequently visited items. (See Figure 12 for a screenshot of how the menu would look.) A button which when clicked would give the user other navigation items has be given. (See Figure 13 for a screenshot of the expanded menu.)

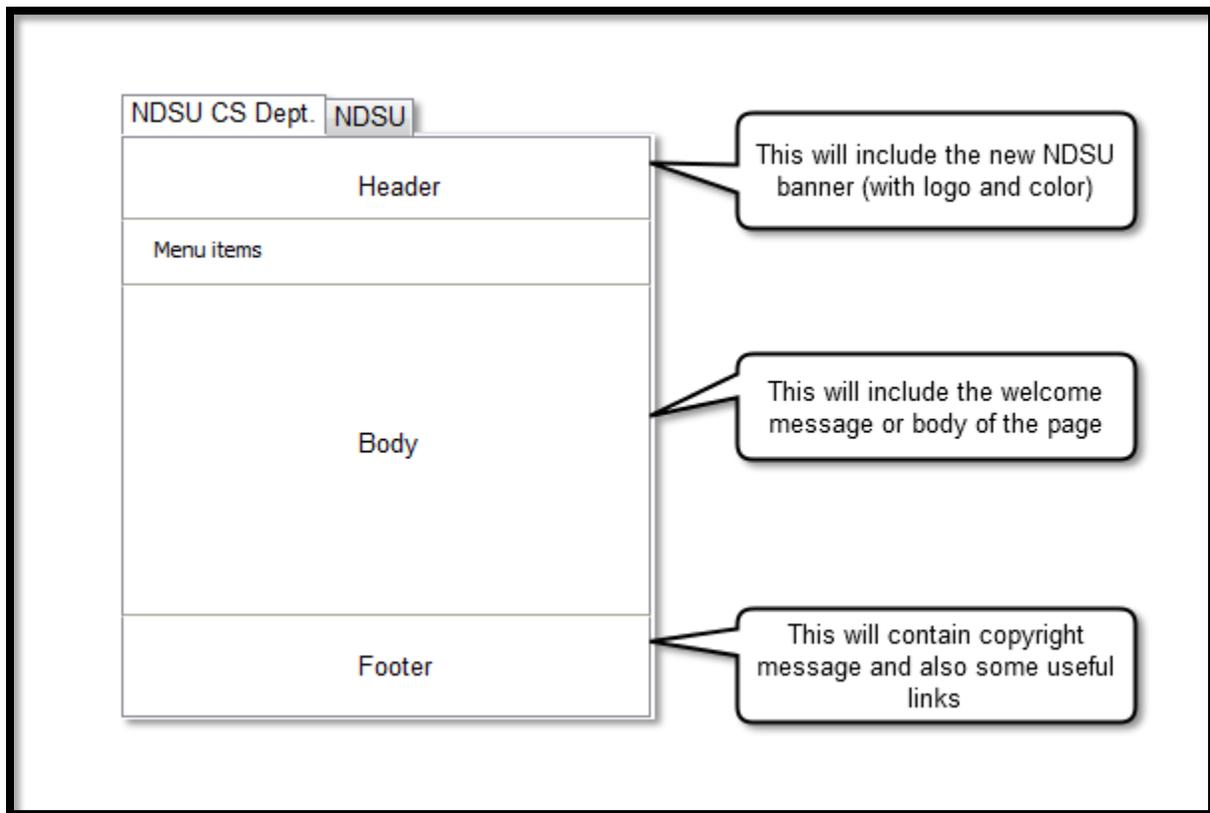


Figure 1. Fundamental Wireframe for Home Page.

Wireframe 2. The wireframe in Figure 2 can be used to give the user an easier way to browse pictures instead of scrolling a long web page. This wireframe can be used to show news headlines (with a slideshow) or faculty information. See Figure 14 for a screenshot of how this wireframe will look like on a web page.

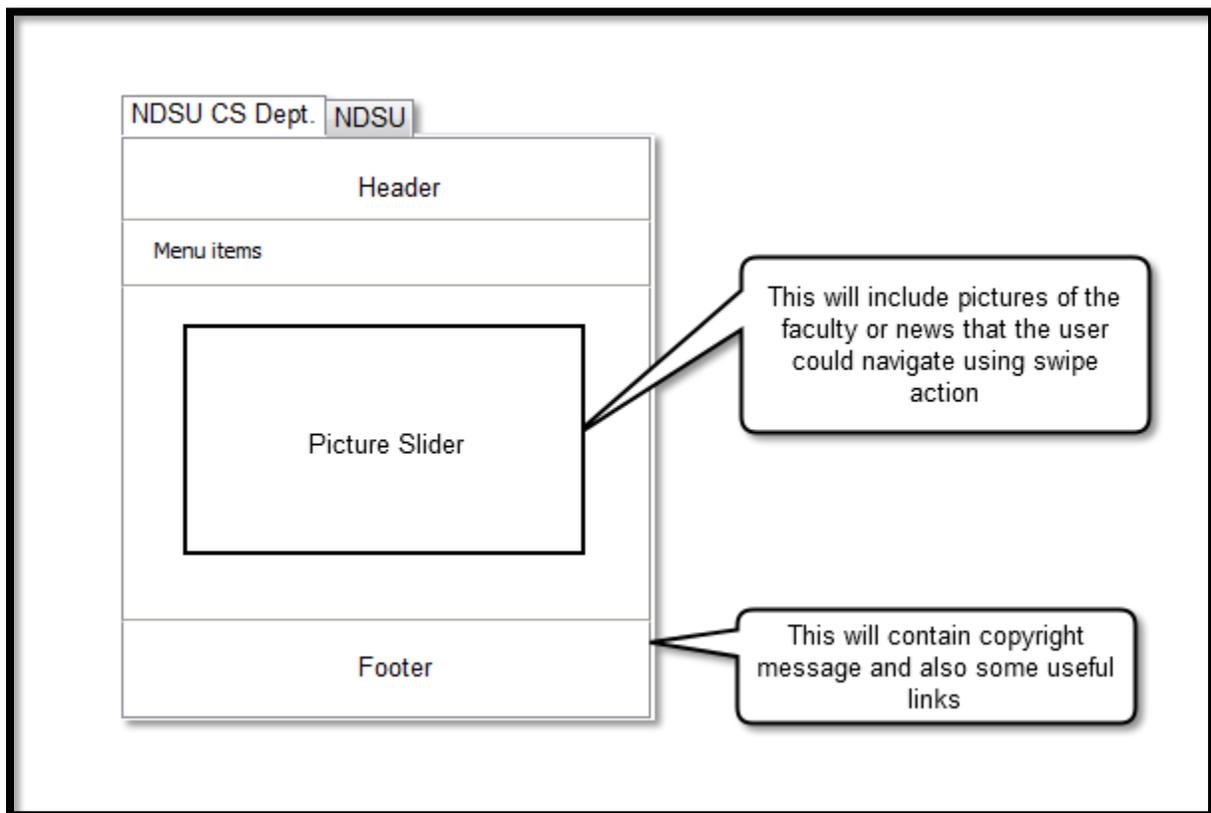


Figure 2. Wireframe for Showing News or Other Information as a Slider.

Wireframe 3. The wireframes in Figure 4 could be used to filter down the results according to users and to show menu items according to the user's choice. (See Figure 15 for a screenshot of various filters.) We could show a stripped-down version of the menu items or the content so that the user is able to find the content he needs in less time, and the menu could be divided according to various users (current, prospective, or graduate students). See Figure 8 for details. The wireframe in Figure 3 is an example to show how the graduate students' checklist should look.

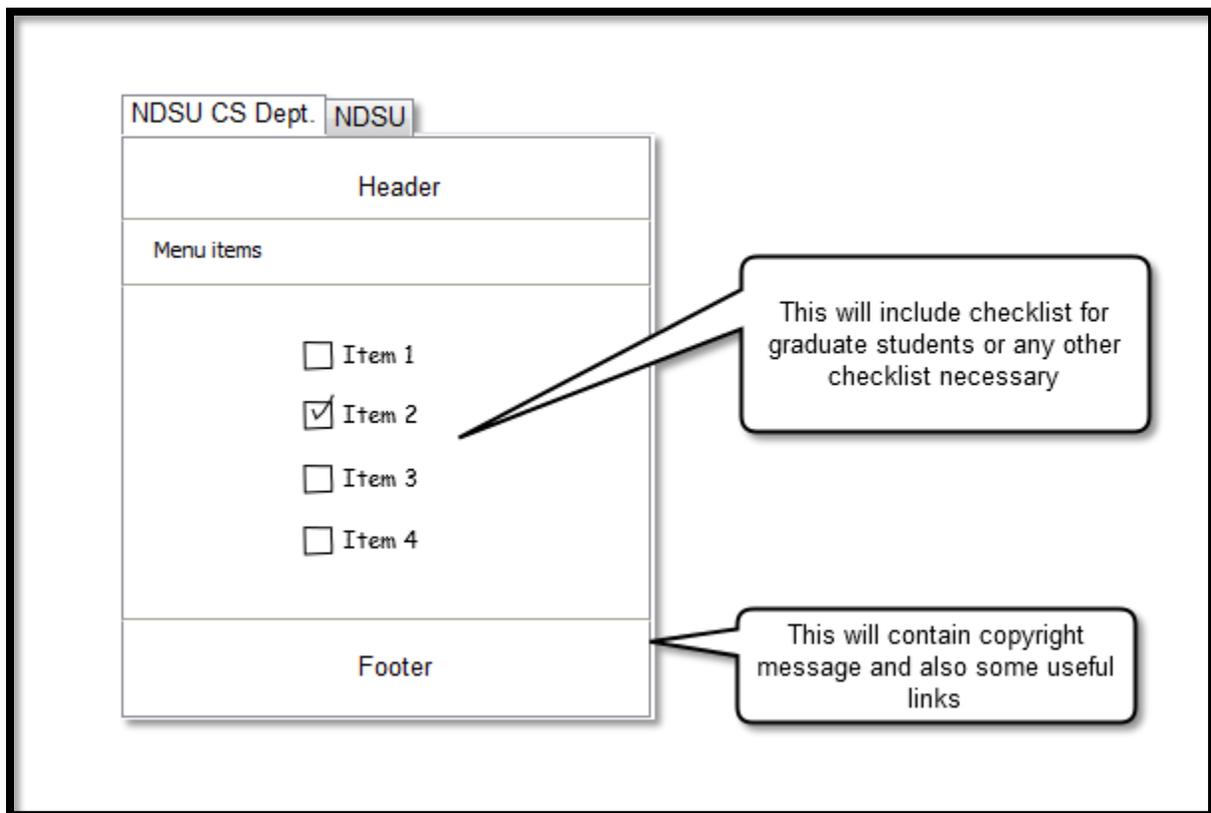


Figure 3. Wireframe for Showing Checklist to Graduate Students.

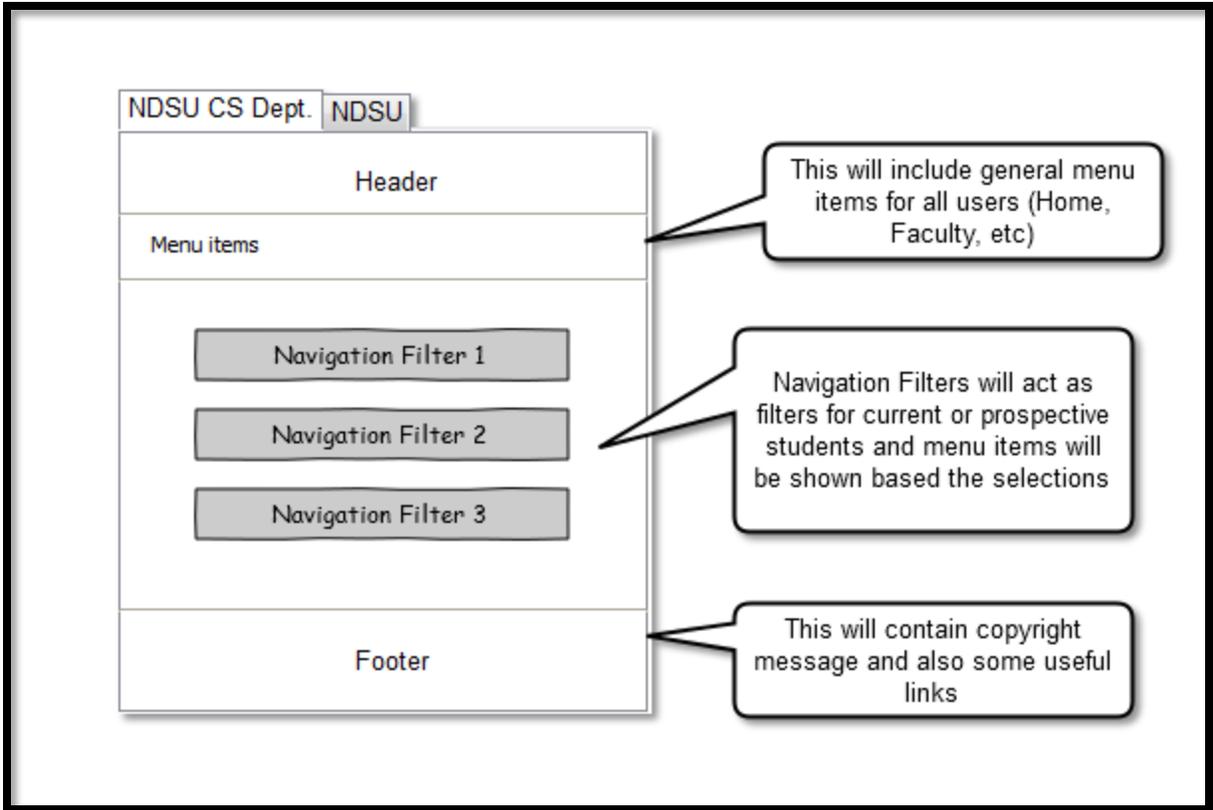


Figure 4. Wireframe With Various Filters for Showing Specific Menu According to the Choice Made.

CHAPTER 5. USE CASES

In this chapter, I will discuss the use case diagram (Figure 5) for the mobile website.

There are two types of actors in the use cases. Normal users include the prospective students and even current students. Graduate student users can see the checklist for the Plan of Study or other documents that need to be submitted to the Graduate School.

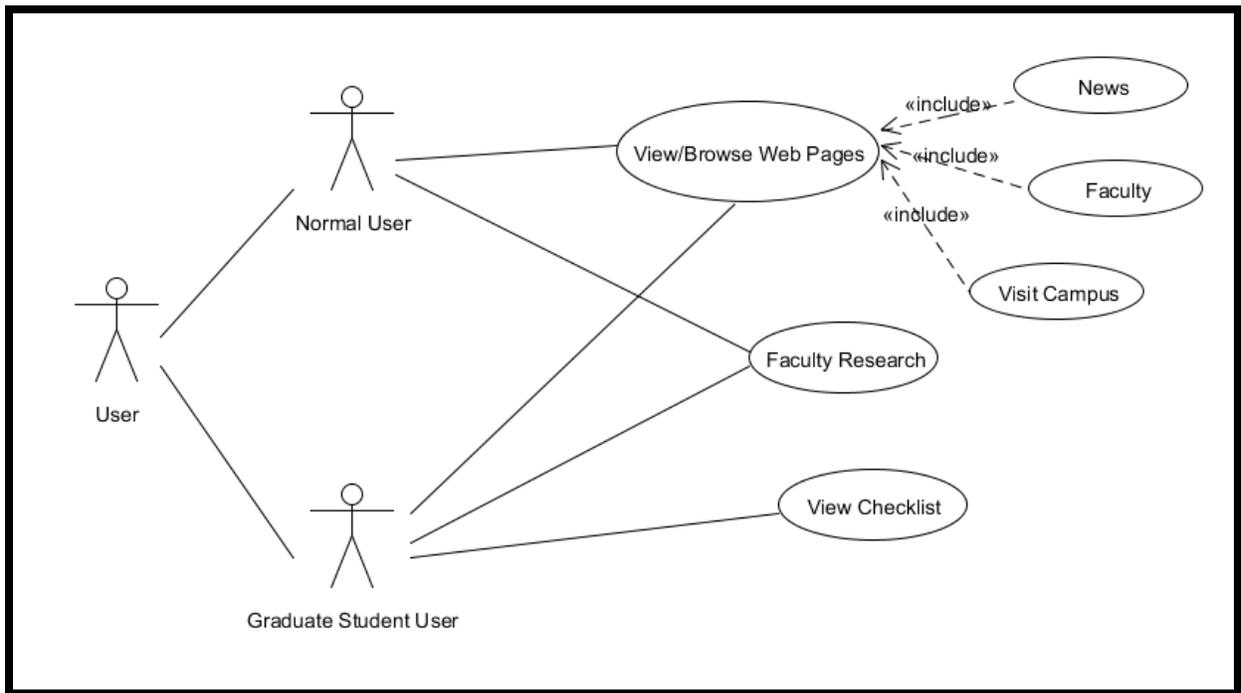


Figure 5. Use Case for Mobile Website Prototype.

CHAPTER 6. ARCHITECTURE AND TECHNOLOGIES

I have used two-tier client-server architecture [6]. The client/server model is a computing model that acts as distributed application which partitions tasks or workloads between the providers of a resource or service, called servers, and service requesters, called clients. Often, clients and servers communicate over a computer network on separate hardware, but both the client and server may reside on the same system.

I decided to use two-tier rather than three-tier architecture due to simplicity. The two tiers will be easier to maintain and update the website because we do not have to maintain a database server which adds to the cost of hosting the website.

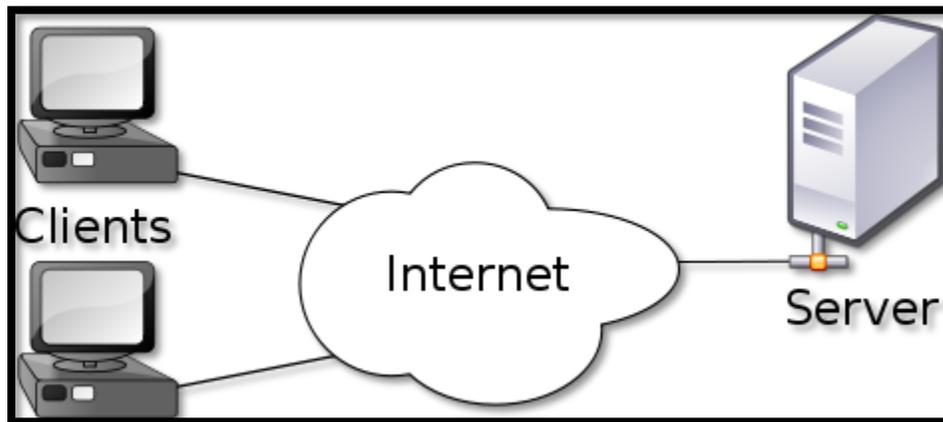


Figure 6. Two-Tier Client Server Architecture.

I have also divided the web pages into different component to make the maintenance easier. The components are as follows:

1. Header
2. Menu Items
3. Slider

4. Body
5. Footer

Technologies Used. The technology used to create the website is as follows:

1. Apache Tomcat Server
2. Netbeans 7.0.1
3. PHP 5.0
4. Browsers (Firefox, Chrome, IE, Safari, Dolphin, and Opera)
5. Pencil Project (used for GUI prototyping)
6. Zymic.com (for free website hosting)
7. Firebug

CHAPTER 7. WORK

The source code for the website is included by the name of NDSUCSMobile.zip file. Because the project is created with PHP, a server-side scripting or compiling language, it requires a PHP-capable server to run the website.

I decided to develop the Computer Science Department's mobile website based on the same template as the new North Dakota State University (NDSU) banner (Figure 7) at the top. The most visited/most important menu items are first, and there is a dropdown menu for other items.

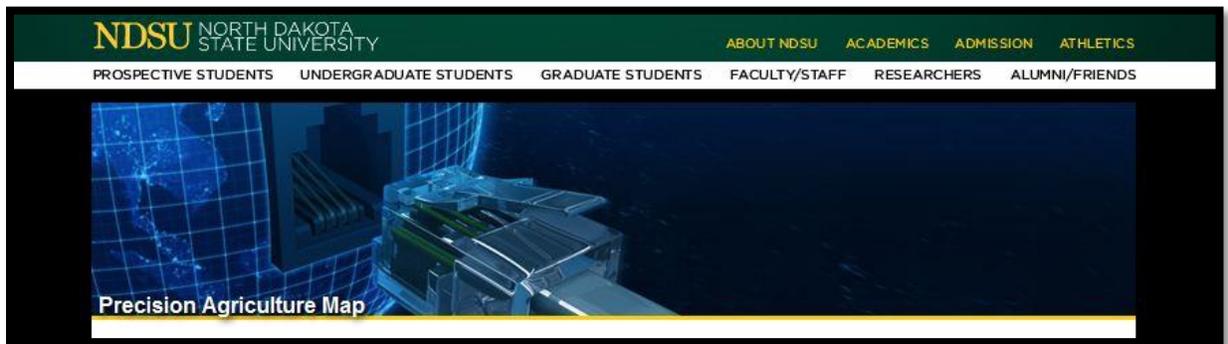


Figure 7. Header for Secondary Pages for NDSU.

Menu Simplification. The menu has to be simplified in order to show it on a mobile device. See Figure 8 below to see details.



The degree programs could be classified according to undergraduate or graduate students. Relevant degree programs should be shown using Figure 4.

Figure 8. Current Menu on the NDSU CS Department website.

CHAPTER 8. TESTING

For testing the website, I followed several guidelines:

Testing Task 1. Test the site on a desktop browser first to ensure the functionality behaves as expected.

Testing Task 2. Once the site is functionally complete, test it using some device emulators.

Testing Task 3. When the site is working correctly across all emulators, begin testing on real handsets.

I tested the website on Android [11]/iPhone [12] emulators and also on various Android/Apple devices (e.g., Samsung Galaxy S2, Galaxy S3, iPhone 4, iPhone 4S, Samsung Precedent, Blackberry Torch, etc.). It passed all the tests. I also used a World Wide Web Consortium [W3C] mobile checker (<http://validator.w3.org/mobile/>) to see if the site passed the guidelines set up by them. The W3C mobileOK Checker performs various tests on a web page to determine its level of mobile friendliness. The tests are defined in the mobileOK Basic Tests 1.0 specification. A web page is mobileOK when it passes all the tests (Figure 9).

I used this test because it is offered by World Wide Web Consortium which develops open standards to ensure the long-term growth of the web. This test will examine the formation of the HTML on the webpage as well as check for CSS errors and if the images used are in the JPEG or GIF format (Objective 5). As you can see in Figure 9, there are some errors because the alternate text for the image is missing and because there were white-space characters or comments on the webpage.

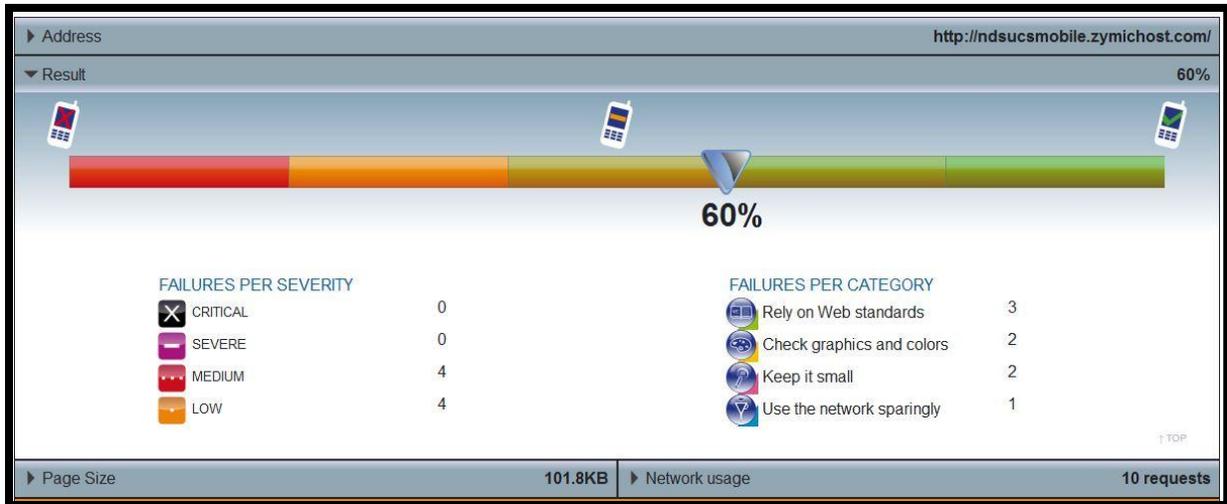


Figure 9. MoileOK Test Results.

I also performed tests to learn how the website ranks according to various industry standards using mobiReady. The mobiReady testing tool evaluates mobile readiness using the industry’s best practices and standards. The free report provides both a score (from 1 to 5) and in-depth analysis of pages to determine how well a site performs on a mobile device (Figure 10).

I decided to do this test because it makes sure that I met my Objective 1 and Task 3 because the mobiReady test checks if I used a table as the layout for the webpage or if I used a nested table for anything. It even checks if I used a table in general for anything on the webpage. It checks for things such as character encoding, if the website opens popup windows, or if the page redirects to some other website. This test even checks if a CSS is used, confirming Objective 2 of using a CSS for to make the mobile website compatible with various browsers. The score for this test is “bad” (Figure 10) because the test assigns more weight to the image size that increases the size of the webpage (Objective 6 was to keep the size of the webpage under 20 kilobytes.) and also the alternate text for the images. The webpage I used to for this test has bigger images and the alternate text. I tried the same test after making the image size smaller.

The website passed the test, but images did not look proper due to resizing. (I have very limited knowledge about picture resizing using Adobe Photoshop.)



Figure 10. MobiReady Test Results.

Also, I performed the Gomez free Mobile Readiness Test. The Gomez free Mobile Readiness Test evaluates how well a site performs on a mobile phone. Instantly, the user gets a score from 1 to 5 based on an in-depth analysis of 30 proven mobile web-development best practices, ranging from standard-compliant code and style-sheet use to caching techniques. It takes about 24 hours for this test to complete (Figure 11).

I conducted this test because it considers multi-browser performance which includes compatibility with the major browsers; it even checks cross-device compatibility to verify if the website renders properly across all devices (Objectives 1, 2, and 3). It combines these two checks and displays them as Gomez Page Composition. As you can see in Figure 11, the prototype website scored 73%, and it does have 4 medium failures due to some Internet Explorer (IE) Javascript issues which were on the webpage when I ran the test. The test also compares how fast is the website loads compared to other page, such as the Google page, and the result is 31% (Figure 11) due to the larger images on the prototype website.

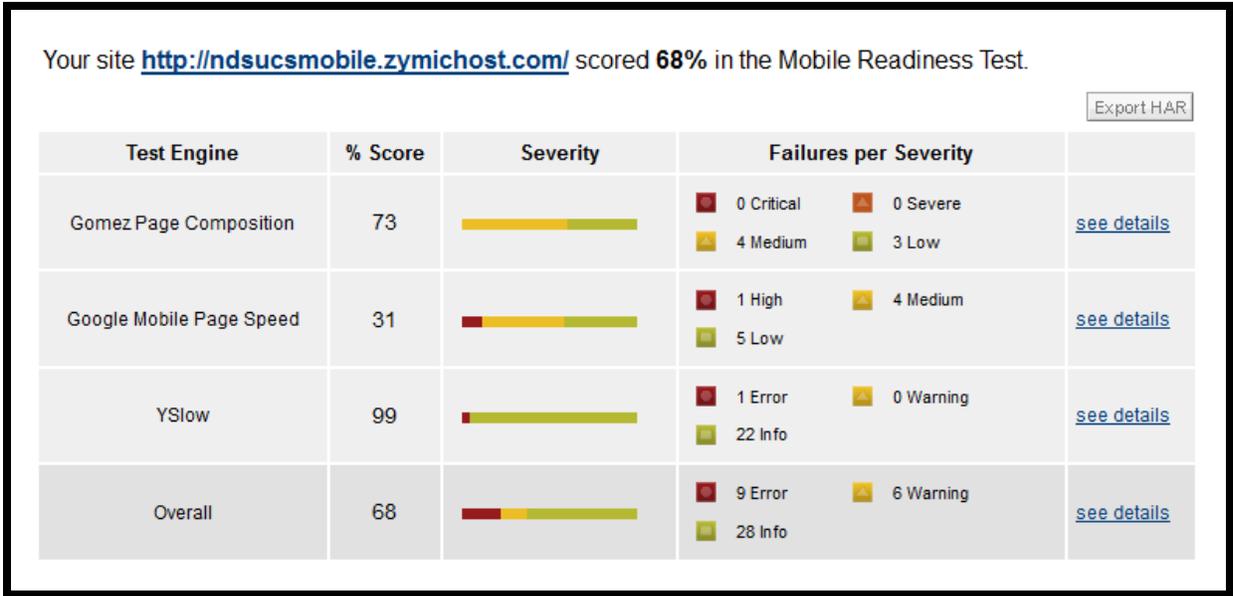


Figure 11. Mobile Website Readiness Test Results.

I used the above-mentioned tests because they cover most of my objectives and tasks and because they validate that I was able to achieve the goals set in Chapter 2. I did these tests when I was developing the website. Therefore, some of the test numbers are not what one would expect them to be, and also, there are some errors because image sizes makes the page bigger than 20 kilobytes (KB).

CHAPTER 9. SCREENSHOTS

In Figure 12, you can see the modified menu items. (See Figure 8 for the original menu items.) I have shown a few of the menu items, thus fulfilling Goal 1 of simplified navigation. This page also makes the site task oriented because it could be used for particular content such as full news content. In this screenshot (Figure 12), the webpage is used for displaying the North Dakota State University Computer Science Department building.

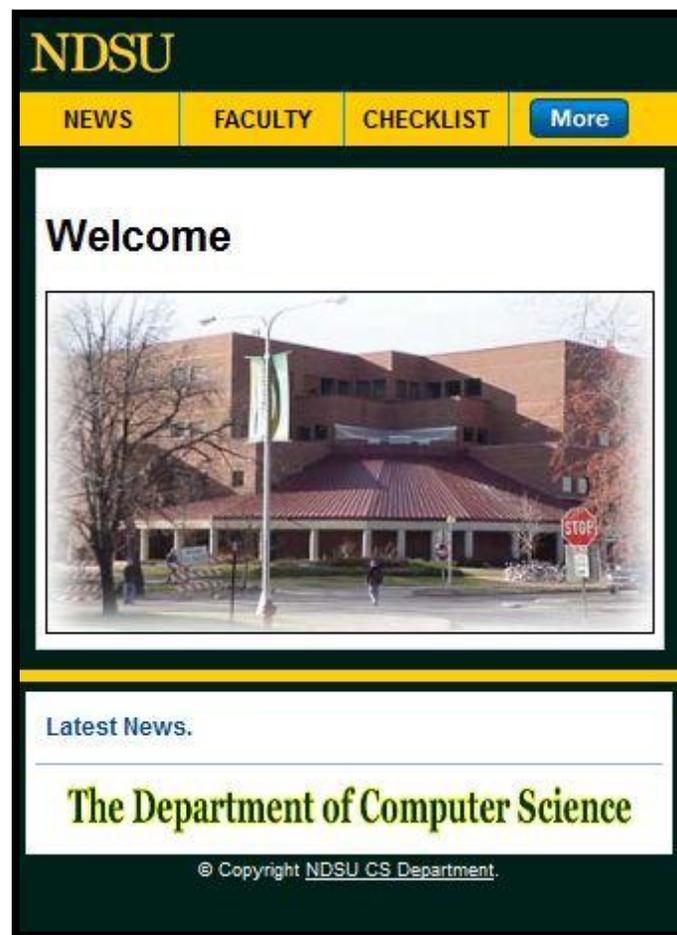


Figure 12. Home Page of the Prototype Website.

Figure 13 shows the expanded menu item for the menu simplification (Goal 1) and shrinks the content because only a few menu items are shown (Goal 3). This expanded-menu concept makes it easier for the user to make a selection, thus it making the website task oriented (Goal 2) because the user will be able to select the desired menu item.



Figure 13. Expanded Menu from More Text Button.

In Figure 14, I have shown a screenshot of the slider given to display news or some other information, such as faculty information, at equal intervals (slideshow). This slider functionality shrinks the content because a picture is worth a thousand words [17], thus shrinking the content

which fulfills Goal 3. I have used a Javascript-based slider called Flexslider [18] instead of using a Flash-based slider (Task 1).

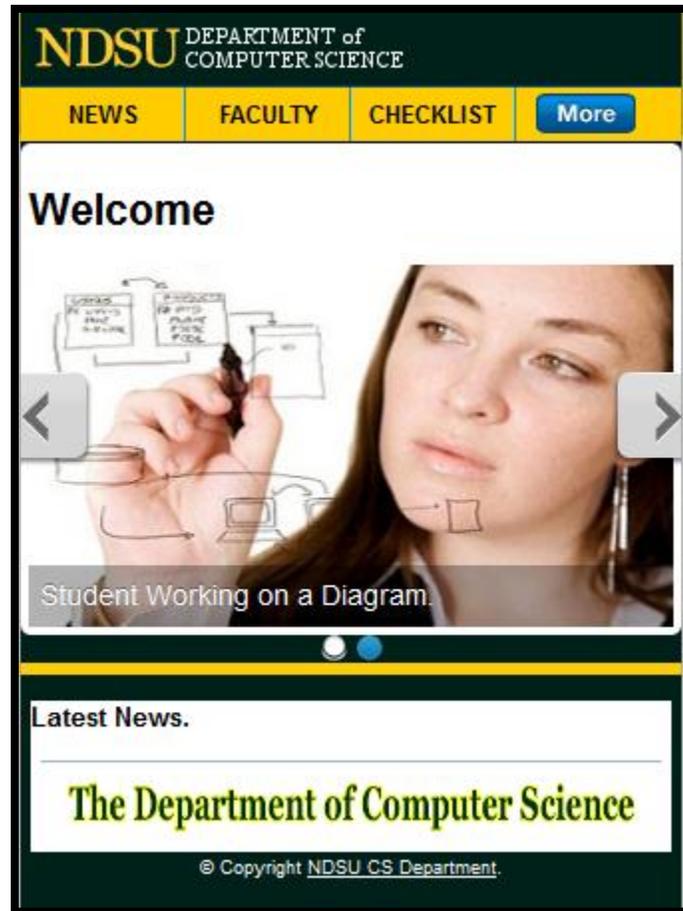


Figure 14. Slider Page for News or Other Information.

Figure 15 shows navigation filters for different menu items according to the user's choice. These filters serve the purpose of Objectives 3 and 4, making the website more structured, and the content is more organized according to the user's selection. This page also displays a list for the user's input, fulfilling Objective 5.

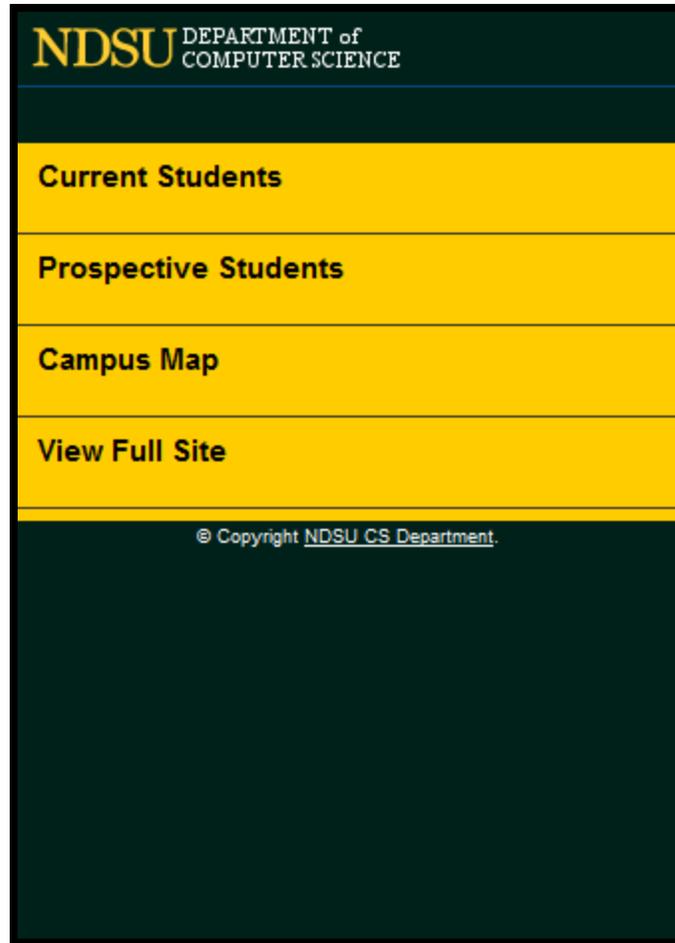


Figure 15. Mobile Website Page for Showing the Menu According to Various Filters.

CHAPTER 10. QUESTIONS LEFT UNANSWERED

I was discussing something with my boss (Dr. James Ross, Senior Developer, Enterprise Computing and Infrastructure, NDSU) [5], and he suggested that I could utilize a streamlining process of differentiating among different user types (prospective students, undergraduate students, graduate students, etc.) using some filters and then show information specific to the user. The main question is how to define those filters for a mobile website because they already exist for a desktop-based website.

The current Computer Science Department website needs to be changed to the new NDSU template, and it needs to be adjusted for mobile devices. Also, there is a question about how to handle the Apple iPad and Android tablets because they are available in different sizes. Should the same website (desktop version) be used, or should a different one be designed for mobile devices?

CHAPTER 11. CONCLUSION AND FUTURE WORK

Conclusion. Mobile website development is very challenging because the screen size is limited and less data is good. (Content is good but in small bites.) The Department of Computer Science website meets most of the industry guidelines and has passed various tests that I conducted to validate website design. I followed a test-and-learn approach to develop this website because I had to come up with a design that works best on most mobile phones. Now, it is all about 128 x 160 pixels, and it is growing at a faster pace. I was able to meet my goal of simplifying the navigation by providing various filters (Figure 15). I provided a slider and more buttons on the menu in order to shrink the website content. (Content is king but in small bites.) Providing navigation filters also made the website task oriented because it classifies users based on the various tasks performed.

I was able to fulfill all of my objectives and tasks except for the objective to keep the size of the webpage under 20 kilobytes. The page with a slider (Figure 14) has many images, thus making it difficult to keep the page under 20 kilobytes.

In order to make the website compatible on the commonly used web browsers and not to give it a fixed-boxes look, I put the width and height in the CSS file in percentages rather than in pixels. I also avoided using any form of table on the web pages because, on smaller screens, it becomes difficult for users to view the content when they have to scroll horizontally or vertically.

I was able to make this prototype easy to maintain and update by dividing the content into different files. If there is a need to change the menu, we can update it in one file rather than going to each single file.

Future Work. Future work for the prototype website would include actual implementation on the NDSU CS Department servers, in other words making it live and accessible to everyone. The prototype would also include some changes to the website images because they need some work for the webpage size to be less than 20 kilobytes.

Instead of using a Javascript slider, we could use an HTML 5 slider if browsers start supporting it fully. Make the mobile website for all the users other than the student like the desktop version.

We can make the website live and certain steps need to be followed. We need to get the approval from all the members of the Computer Science Department as to whether all of them agree that there a need to have a mobile website and also agree on the design/look of the website.

We also need to figure out on the images as well as whether to use the filters for different users in order for them to see menu specific to that user group or to show same menu with frequently accessed menu items first. We need to find out what the actual frequently accessed items are by doing some study on the current desktop version website using Google Analytics or other tools for getting an accurate data.

Once we have finalized the design we need to contact the University Public Relations team for their approval and to check with them if there are certain guidelines that need to be followed or is there anything that need to be changes in order for us to make the website live.

After getting the approval we need to make a decision at the Department Level as to whether we need to use how we are going to maintain the website as no wants to make the make the changes twice for a website. We need to make a decision whether to make use of a Content Management System (CMS) or to have a database in order to make the changes only once. We

need to make sure if the current desktop version of the website need to be restructured for making use of the CMS or the database.

After making sure that all the university guidelines are met and all the maintenance related questions are answered we need to finalize the content that needs to be shown on the mobile website and how often is that content is going to be updated and also form a committee in charge for the update process.

We need to make sure that we have rules setup on the server to see what devices the user is trying to access the website with so that we can show the appropriate website. We need to install PHP server do that the pages are rendered properly to the device. We need to make sure that appropriate ports are opened for accessing the website and also the https certificates are installed and ready if the mobile website is going to use it.

REFERENCES

1. Ryan Stemkoski "How to Design Website for Mobile Phones" [Online]. Available: <http://www.stemkoski.com/how-to-design-websites-for-mobile-phones/>. [Accessed May 2012].
2. Yaswanth Potla, Ramakanth Annadi, Jun Kong, Gursimran Walia and Kendall Nygard. "Adapting Web Page Tables on Mobile Devices," 2011.
3. Z. Hua and H. Lu. "Web Browsing on Small-Screen Devices: A Multiclient Collaborative Approach," IEEE Pervasive Computing, Vol. 5(2), 2006.
4. S. Bhaluja. "Browsing on Small Screens: Recasting Web-Page Segmentation into an Efficient Machine Learning Framework," Proc. 5th International Conference on World Wide Web, 2006.
5. Personal Communication with Dr. James Ross, 2011.
6. Sun Microsystems. "Distributed Application Architecture," 2012.
7. "The Best & Worst of the Mobile Web," EBook Mobithink.com Panelist. [Online]. Available: <http://mobithinking.com/>. [Accessed May 2012].
8. "What is CSS?" World Wide Web Consortium. [Online]. Available: <http://www.w3.org/standards/webdesign/htmlcss#whatcss>. [Accessed November 2012].
9. World Wide Web Consortium. "Tags Used in HTML," November 3, 1992.
10. Robert Bringhurst. "The Elements of Typographic Style," 1992.
11. "Android SDK" [Online]. Available: <http://developer.android.com/sdk/index.html>. [Accessed November 2012].
12. "Xcode 4" [Online]. Available: <https://developer.apple.com/xcode/index.php>. [Accessed November 2012].

13. “Application Software” [Online]. Available:
http://en.wikipedia.org/wiki/Application_software. [Accessed November 2012].
14. “An Update on Flash Player and Android” [Online]. Available:
<http://blogs.adobe.com/flashplayer/2012/06/flash-player-and-android-update.html>.
[Accessed November, 2012].
15. “Mobile Web Design” [Online]. Available: <http://www.noupe.com/how-tos/mobile-web-design-tips-and-best-practices.html>. [Accessed May 2012].
16. “How to Design and Build a Mobile Web Site” [Online]. Available:
<http://www.webpagefx.com/design-build-mobile-web-site.html>. [Accessed May 2012].
17. “A Picture is Worth a Thousand Words” [Online]. Available:
http://en.wikipedia.org/wiki/A_picture_is_worth_a_thousand_words. [Accessed
November 2012].
18. “Flexslider” [Online]. Available: <http://www.woothemes.com/flexslider/>. [Accessed
November 2012].

APPENDIX. CODE

This code for the index page with a slider is as follows:

```
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01//EN"
"http://www.w3.org/TR/html4/strict.dtd">

<html lang="en">

<head>

<title>NDSU CS - Mobile</title>

<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">

<meta name="language" content="EN">

<meta name="viewport" content="width=320, user-scalable=yes">

<meta name="format-detection" content="telephone=no">

<link rel="stylesheet" type="text/css" href="css/style.css" media="all">

<script src="scripts/jquery.js"></script>

<script type="text/javascript" src="scripts/ndsucsmobile.js"></script>

<script type="text/javascript" src="scripts/s_code.js"></script>

<!-- Slider Pieces -->

<link rel="stylesheet" href="css/flexslider.css" type="text/css" media="screen" />
```

```
<script src="https://ajax.googleapis.com/ajax/libs/jquery/1/jquery.min.js"></script>
```

```
<script src="jquery.flexslider-min.js"></script>
```

```
<script type="text/javascript">
```

```
$(window).load(function() {
```

```
    $('.flexslider').flexslider();
```

```
});
```

```
</script>
```

```
</head>
```

```
<body>
```

```
<div id="header">
```

```
<div id="masthead">
```

```
<div id="usoLogo">
```

```
<a href="index.php">
```

```

```

```

```

```
</a>
```

</div>

</div>

<div id="siteNav">

<div>

News

</div>

<div>

Faculty

</div>

<div>

Checklist

</div>

<div id="menuButton">

</div>

</div>

```
<div style="display: none;" id="fullMenu">
```

```
<a href="index.php">NDSU CS Department Mobile Home</a>
```

```
<a href="index.php">Prospective Students</a>
```

```
<a href="index.php">Current Students</a>
```

```
<a href="index.php">Faculty Research</a>
```

```
<a href="index.php">Visiting The Campus</a>
```

```
<a href="http://cs.ndsu.nodak.edu/">NDSU CS Department <span  
class="detail">(View full site)</span></a>
```

```
</div>
```

```
</div>
```

```
<div id="mobileContent">
```

```
<div class="playerPhotoContainer">
```

```
<div class="flexslider">
```

```
<h1>Welcome</h1>
```

```
<ul class="slides">
```

```
<li>
```

```

```

```
<p class="flex-caption">Computer Science Building (IACC).</p>
```

```
</li>
```

```
<li>
```

```

```

```
<p class="flex-caption">Student Working on a Diagram.</p>
```

```
</li>
```

```
</ul>
```

```
</div>
```

```
</div>
```

```
<br style="clear: both;">
```

```
<div id="dynamicMsg">
```

```
<a href="index.php"></a>
```

```
</div>
```

```
<div id="mobilePage">
```

```
<!--?xml version="1.0" encoding="UTF-8"?-->
```

```
<div class="articleTeaser">
```

```
<h3>
```

```
Latest News.
```

```
</h3>
```

```
<div class="clear">&nbsp;</div>
```

```
</div>
```

```
<div id="usOpenRadioPromo">
```

```
<a href="">
```

```

```

```
</a>
```

```
</div>
```

```
</div>
```

```
</div>
```

```
<div id="footer">
```

```
© Copyright <a href="http://cs.ndsu.nodak.edu/">NDSU CS Department</a>.
```

</div>

<noscript>

Please Unable java script.

</noscript>

</body>

</html>