In 2011, the School of Information and Library Science at the University of North Carolina at Chapel Hill installed a 46-inch touchscreen computer in the Manning Hall lobby. After discussing the system’s role with stakeholders, a Drupal-based website was developed with the following system goals: showcase the contributed work and accomplishments of current library and information science students, inspire unique and creative student work for the touchscreen, demonstrate educational opportunities to prospective students, provide an introduction for SILS newcomers and when idle, display event announcements or community interest topics.

This document is for content editors, site developers, system administrators, and web designers who will maintain and expand the site. The manual is divided into eight sections: Introduction, Overview of Site Navigation and Site Sections, Adding and Editing Content, Basic Site Maintenance and Administration, Overview of Site Technology, Reference for Site Builders/Developers, Reference for Site Designers and Future Work.

Headings:

- Human-computer interaction
- Graphical user interfaces (Computer systems)
- Web development
- Web design
- Websites -- Management
CONTENT EDITING, SITE DEVELOPING AND WEB DESIGNING GUIDE FOR A DRUPAL 7 BASED, LARGE TOUCHSCREEN SITE

by
Robert L Cooper

A Master’s paper submitted to the faculty of the School of Information and Library Science of the University of North Carolina at Chapel Hill in partial fulfillment of the requirements for the degree of Master of Science in Information Science.

Chapel Hill, North Carolina
July 2013

Approved by

_______________________________________
Jeffrey Pomerantz
# Table of Contents

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Table of Contents</td>
<td>1</td>
</tr>
<tr>
<td>Introduction</td>
<td>4</td>
</tr>
<tr>
<td>1.1 About this Document</td>
<td>6</td>
</tr>
<tr>
<td>1.1.1 What it Covers</td>
<td>6</td>
</tr>
<tr>
<td>1.1.2 What it Does Not Cover</td>
<td>7</td>
</tr>
<tr>
<td>Overview of Site Navigation and Site Sections</td>
<td>10</td>
</tr>
<tr>
<td>1.2 User Interface</td>
<td>10</td>
</tr>
<tr>
<td>1.3 Administration Interface</td>
<td>11</td>
</tr>
<tr>
<td>1.4 Site Sections</td>
<td>12</td>
</tr>
<tr>
<td>1.4.1 Student Projects</td>
<td>12</td>
</tr>
<tr>
<td>1.4.2 Conference Posters</td>
<td>14</td>
</tr>
<tr>
<td>1.4.3 Promotional Material</td>
<td>16</td>
</tr>
<tr>
<td>Adding and Editing Content</td>
<td>17</td>
</tr>
<tr>
<td>1.5 Accessing the System Via Windows</td>
<td>17</td>
</tr>
<tr>
<td>1.6 Adding Student Projects</td>
<td>17</td>
</tr>
<tr>
<td>1.6.1 Before You Begin</td>
<td>17</td>
</tr>
<tr>
<td>1.6.2 Adding a Single Page</td>
<td>20</td>
</tr>
<tr>
<td>1.6.3 Adding a Mini-site</td>
<td>21</td>
</tr>
<tr>
<td>1.6.4 Adding an Existing Site Via an Iframe</td>
<td>22</td>
</tr>
<tr>
<td>1.6.5 Where Files Are Stored</td>
<td>23</td>
</tr>
<tr>
<td>1.7 Adding Conference Posters</td>
<td>23</td>
</tr>
<tr>
<td>1.7.1 Before You Begin</td>
<td>23</td>
</tr>
<tr>
<td>1.7.2 Adding a Conference Poster</td>
<td>23</td>
</tr>
<tr>
<td>1.7.3 Where Files Are Stored</td>
<td>23</td>
</tr>
<tr>
<td>1.8 Adding Promotional Material</td>
<td>24</td>
</tr>
<tr>
<td>1.8.1 Before You Begin</td>
<td>24</td>
</tr>
<tr>
<td>1.8.2 Adding Promotional Material</td>
<td>24</td>
</tr>
<tr>
<td>1.8.3 Where Files Are Stored</td>
<td>24</td>
</tr>
<tr>
<td>1.9 Editing Content</td>
<td>24</td>
</tr>
<tr>
<td>Basic Site Maintenance and Administration</td>
<td>25</td>
</tr>
<tr>
<td>1.10 Backing Up and Restoring Drupal</td>
<td>25</td>
</tr>
<tr>
<td>1.10.1 Backing Up the Database</td>
<td>25</td>
</tr>
<tr>
<td>1.10.2 Backing Up System Files</td>
<td>25</td>
</tr>
<tr>
<td>1.10.3 Restoring the Database</td>
<td>26</td>
</tr>
<tr>
<td>1.10.4 Restoring System Files</td>
<td>26</td>
</tr>
<tr>
<td>1.11 Updating Add-on Modules</td>
<td>27</td>
</tr>
<tr>
<td>1.12 Updating Drupal Core Software</td>
<td>27</td>
</tr>
<tr>
<td>Overview of Site Technology</td>
<td>29</td>
</tr>
<tr>
<td>Section</td>
<td>Page</td>
</tr>
<tr>
<td>------------------------------------------------------------------------</td>
<td>------</td>
</tr>
<tr>
<td>Reference for Site Designers</td>
<td>51</td>
</tr>
<tr>
<td>Reference for Site Builders/Developers</td>
<td>34</td>
</tr>
<tr>
<td>1.12 Design Resources</td>
<td>51</td>
</tr>
<tr>
<td>1.12.1 Foundational Design Skills</td>
<td>51</td>
</tr>
<tr>
<td>1.12.2 Design Theory</td>
<td>51</td>
</tr>
<tr>
<td>1.12.3 CSS Techniques</td>
<td>52</td>
</tr>
<tr>
<td>1.12.4 Design Inspiration</td>
<td>52</td>
</tr>
<tr>
<td>1.12.5 Quick Start for New Designians</td>
<td>52</td>
</tr>
<tr>
<td>1.13 Current Site Design</td>
<td>52</td>
</tr>
<tr>
<td>1.13.1 Wireframes</td>
<td>53</td>
</tr>
<tr>
<td>1.13.2 Custom Page Layouts with Panels</td>
<td>55</td>
</tr>
<tr>
<td>1.13.3 Design Philosophy</td>
<td>56</td>
</tr>
<tr>
<td>1.13.4 Style Guide</td>
<td>59</td>
</tr>
<tr>
<td>1.13.5 Large Screen and Touchscreen Design and Usability Considerations</td>
<td>59</td>
</tr>
<tr>
<td>1.14 Custom Drupal Subtheme: Genesis Touchscreen</td>
<td>62</td>
</tr>
<tr>
<td>1.14.1 HTML5 and CSS3 features</td>
<td>64</td>
</tr>
<tr>
<td>1.14.2 Theme CSS Files</td>
<td>64</td>
</tr>
<tr>
<td>1.14.3 Custom Template Files</td>
<td>65</td>
</tr>
<tr>
<td>1.14.4 JavaScript</td>
<td>65</td>
</tr>
<tr>
<td>1.15 Image Guidelines</td>
<td>68</td>
</tr>
<tr>
<td>1.15.1 Source Files</td>
<td>68</td>
</tr>
<tr>
<td>1.15.2 Optimizing and Saving Files for the Web</td>
<td>69</td>
</tr>
<tr>
<td>Future Work</td>
<td>75</td>
</tr>
<tr>
<td>1.16 Alternate Browsing Paths</td>
<td>75</td>
</tr>
<tr>
<td>1.16.1 Browse by Taxonomy</td>
<td>75</td>
</tr>
</tbody>
</table>
Introduction

When Apple launched the iPhone in 2007, it also launched a touchscreen revolution, and as large-format display screens rapidly fell in price throughout the late 2000s and early 2010s, their use as digital signage become nearly ubiquitous as well. The large-format touchscreen computer is an emerging product that is a natural combination and extension of the trends based on the popularity of the iPhone and large-format display screens.

During 2011 summer, the School of Information and Library Science (SILS) at the University of North Carolina at Chapel Hill (UNC-Chapel Hill) installed a 46-inch touchscreen computer in the lobby of their building, Manning Hall. At the time, I was looking for a master’s project that would incorporate the fullest range of skills I’d learned at SILS, and I could not resist opportunity to develop the interactive software and to create the visual design of such a rare device. After discussing the role of the touchscreen computer with all of the stakeholders, Dean Gary Marchionini, Director of Information Technology Aaron Brubaker and SILS Director of Communications Wanda Monroe, I identified the following system goals:

- to showcase the contributed work and accomplishments of current students in the library science and information science programs
- to inspire unique work designed specifically for the new touchscreen interface and to provide an interesting venue for student creativity
- to demonstrate what can learned at SILS to prospective students
- to provide some orientation or introduction for newcomers to SILS
• when idle, to display announcements related to upcoming events or topics of community interest.

Since the final project needed to support a wide and flexible range of student content and since most development courses at SILS are web-oriented, a website a natural fit. The project also needed to be easy for student administrators to maintain and update, and as I began evaluating technologies to build the project with, I decided that the Drupal 7 content management system (CMS) would provide the necessary flexibility and ease of use. The touchscreen site can include work from nearly any digital medium that can be displayed on a webpage, including single HTML pages with embedded style sheets, PHP and JavaScript code, student-produced videos, Adobe Flash projects, and images such as diagrams and conference posters.

Although a custom solution in a server-side scripting language such as PHP seemed like a good fit at first, Drupal 7 offered an opportunity to learn trending technology and to address one of the shortcomings of a wholly custom student-developed solution: often when a student developer leaves an organization, no one in the organization has the time or perhaps the expertise to modify the system, and so the organization either adapts to using a system that is decreasingly able to meet their evolving needs or they abandon it. Instead, a content management system strikes a balance between competing goals: the need for a custom, flexible system that can be managed and extended by future students with a variety of skill sets, and yet still be a system that can accommodate a wide range of student submissions.

The final project runs a dedicated website, built in Drupal 7, in a browser in kiosk mode. The system has three main parts: Student Projects, a section to showcase student
work; Conference Posters, a section to showcase posters designed by students; and Promotional Materials, a section to display announcements related to upcoming events or topics of community interest.

1.1 About this Document
This document is meant for anyone who intends to fulfill any of these roles with the system: content editor, site builder/front-end/backend developer, system administrator, and web designer.

1.1.1 What it Covers
The manual is divided into eight main sections: Introduction, Overview of Site Navigation and Site Sections, Adding/Editing Content, Basic Site Maintenance and Administration, Overview of Site Technology, Reference for Site Builders/Developers, Reference for Site Designers and Future Work. The topics are presented in approximately the order that a new site editor or administrator will need to learn them and roughly in order of increasing complexity or difficulty.

The Introduction aims to quickly acclimate the reader to the touchscreen site and this document.

Content updates will likely be one of the first tasks a reader will need to do, and the Adding/Editing Content section provides detailed instructions for working with site content.

The Site Maintenance/Administration provides a guide for readers who are new to Drupal to keeping the system up-to-date with security and maintenance patches.
The Reference for Site Builders and Developers section details the site customization for readers who need to modify or expand site structure, features or organization.

The Reference for Site Designers provides readers who want to modify or substantially change the site design or page layouts with the rationale for the current design, a style guide, wireframes, and information about the custom visual theme for the site. The section also details graphic design considerations unique to large screen and touchscreen design. Finally, the section provides an annotated list of design resources for readers who are new to design.

The document concludes with the Future Work section which provides readers with suggestion for fixes and expansions to the site including ways to enhance traffic to the system.

1.1.2 What it Does Not Cover
This guide assumes the reader has basic familiarity with core web technologies such as HTML, CSS, JavaScript, PHP and MySQL so these topics will not be directly addressed by this document. Recommended resources for select topics will be provided throughout the document.

1.1.2.1 Drupal Basics
The touchscreen site was built with the Drupal 7 CMS, http://drupal.org/, and this manual does not cover the basics of using, editing or developing sites in Drupal. As necessary, key Drupal topics will be briefly described to aid Drupal newcomers. See the recommended resources below for training materials to learn Drupal fundamentals.
1.1.2.1.1 Recommended Drupal Resources

1.1.2.1.1.1 Books
Falk’s book is the most comprehensive, concise and clear introduction to key Drupal 7 topics available. The book is logically structured to guide one from simple site-user topics to advanced site-building and site administration topics.

Tomlinson’s book provides a firm grounding in basic site building with Drupal 7. At the time of this writing, the book is available in electronic format through the UNC-Chapel Hill library.

The book is geared towards experienced programmers who need to write custom modules or themes for Drupal. At the time of this writing, the book is available in electronic format through the UNC-Chapel Hill library.

1.1.2.1.1.2 Online Resources
In addition to the books above, the following online Drupal resources are helpful.

The Drupal community maintains help pages that include online documentation, question-and-answer forums, community chat, mailing lists, event listings, Drupal-related books, training resources and professional services at http://drupal.org/community and http://drupal.org/support.

Johan Falk, author of *Drupal 7: the Essentials*, has also created free training videos that cover the topics in his book and many more for NodeOne, a Scandinavian Drupal development firm. The videos can be found in NodeOne's Learning Library at http://nodeone.se/en/learn-drupal.
StackOverflow.com, a question-and-answer site for programming and related disciplines, has an extensive library of Drupal topics at http://stackoverflow.com/tags/drupal.

Acquia, a Drupal support company founded by the creator of Drupal, offers free training webinars. Upcoming Acquia webinars can be found at https://www.acquia.com/resources/webinars and previously recorded webinars can be found at https://www.acquia.com/resources/recorded_webinars.
Overview of Site Navigation and Site Sections

Before making any changes, a site editor or builder should take a moment to become familiar with the user interface, the administration interface, and the major content sections.

1.2 User Interface

1) Page Title/Section Heading
2) Navigation Area
3) SILS logo
4) Links to main site sections
5) Content area for current page
1.3 Administration Interface

1) Administration dropdown menu
2) Shortcut bar
3) Logout link
4) Hide/Show shortcut bar
5) Content overlay
1.4 Site Sections
The site has three major content sections: Student Projects, Conference Posters and Promotional Material. The information below will help you to become familiar with the basic user experience and help you determine which section to add new content to. Step-by-step instructions for adding and editing content are covered in Adding and Editing Content section on p. 17.

1.4.1 Student Projects

When a user presses the Student Projects button, a grid with all projects listed in reverse chronological order appears. A user can click a thumbnail or title to view the selected project.
Once a user has selected a project, the resulting page will display the contributed work and related metadata including title, description and student name. Users will be able to interact with the project if the student designed the work with interactivity in mind. For instance, a static diagram would have no interactivity, while a HTML and JavaScript reworking of the same diagram would allow users to click on aspects of the diagram for further information.
1.4.2 Conference Posters

Conference Poster grid view

When a user presses the Conference Posters link, a grid appears with all posters listed in reverse chronological order. A user can click a thumbnail or title to view the selected poster.
Sample poster node page

Once a user has selected a conference poster, the resulting page will display a large representation of the poster and related metadata including title, description and student name.

Sample poster enlarged in lightbox

To see the poster in more detail, a user can tap the poster to enlarge it to nearly full screen. When finished viewing the enlarged image, a user can then tap the X icon in the upper right corner of the image to close it.
1.4.3 Promotional Material

The SILS Director of Communications develops posters that alert the SILS community to upcoming events and items of interest. When the system has been idle for 10 minutes, it will display an auto-rotating gallery of posters. Since displaying promotional material is the default behavior when the system is idle, the section does not have a navigation link.
Adding and Editing Content

The first and primary activity most system editors or administrators will want to do is manage content on the site. The following section explains how to login in to the administrative backend of Drupal and how to add and edit items in each of the primary content sections of the site.

1.5 Accessing the System Via Windows

1. The SILS Desktop Support and Help Desk Manager or the Director of Information Technology will provide login credentials for the touch screen system and instructions for either logging in directly to the touchscreen or remotely accessing it.

2. Once you have gained access to the Windows desktop, use the web browser to go to http://localhost/drupal/user/login and use the provided credentials to gain access to the Drupal backend.

1.6 Adding Student Projects

1.6.1 Before You Begin

1.6.1.1 Analyze the Project

Since projects can employ a wide variety of technologies, the system has three separate ways to handle adding projects to cover all situations: a single page, mini-site and iframe.

1.6.1.1.1 Single Page

Use this approach if a project can be hosted on a single page such as Flash embed code, video embed code or one page of HTML/PHP/JavaScript code.

1.6.1.1.2 Mini-site
If a project has several pages of content, each page can be added to the system, and the pages can be linked together via taxonomy. See the Adding a Mini-site section on p. 21 for complete instructions.

1.6.1.3 Iframe or Object
If a project cannot practically be added directly to the system, such as a project that relies on a remote database server that will not accept external connections, or if a website already exists on another server and is cannot be moved, an iframe can be used to embed the project in a page on the touchscreen site. Ideally, the student should have sole control of and access to the content and agree that any future revisions will not violate any of the requirements detailed in the next section. See the Adding an Existing Site Via an Iframe section on p. 22 for complete instructions.

1.6.1.2 Sanitize the Project
The site was purposefully designed to be a touchscreen-only system, so no provision has been made to allow text entry by users. Prospective projects should not have any text entry fields, because no keyboard –physical or virtual– is available to users.

The touchscreen site is also a closed system so all links to external sites should be removed before adding a project. If a project is to be added as an iframe, the project pages on the external pages can have links to related project pages, but not to other sites.

If the submitted project contains scripting, such as PHP or JavaScript, a trusted person familiar with the language in question should first review the code. Ideally, site administrators will maintain a development version of the touchscreen site that can be used to test projects with custom code and backup the live site before committing a new project.
The site editor or administrator should closely review any submitted projects with the student and have the student remove any text fields or external links and revise custom code for stability and security as necessary before adding the project to the system. Refer any questions to the SILS Desktop Support and Help Desk Manager or the Director of Information Technology.

1.6.1.3 Get a Representative Image
Obtain an image file, often a screen capture of the project, and a description from the student who produced the project. The description should provide a viewer with context to enhance his/her understanding of the work and its relevance. Edit the image based on the Image Guidelines section on p. 68.

1.6.1.4 Add Taxonomy Terms

1.6.1.4.1 Student Projects Taxonomy
Student projects are organized with two simple taxonomies: a flat list of course names and a two-level, hierarchical taxonomy based on student name and project name. Student names form the top-level and project names form the second level. For instance, the taxonomy for a student named Jane Doe with two projects called My first project and My second project would be structured as:

   Jane Doe
   - My first project
   - My second project

If this is the first project for the student:

1. Click Structure > Taxonomy > Student Projects from the administration dropdown menu.
2. Click Add term.
3. Add the student’s name as a taxonomy term in the Name field.
4. Do not change any other settings and Click Save.
5. Once the student name has been added as a term, follow the instructions below to associate a project name.

To add a project term to an existing student taxonomy term:
1. Click Structure > Taxonomy > Student Projects from the administration dropdown menu.
2. Click Add term.
3. Add the project name as a taxonomy term in the Name field. Per the site Style Guide section on p. 59, the project name should be entered in sentence case.
4. Expand the Relations accordion and in the Parent terms selection list, click on the name of the related appropriate student.
5. Do not change any other settings and Click Save.
6. To reorder taxonomy terms, click Structure > Taxonomy > Student Projects from the administration dropdown menu, drag terms to the desired location in the list and drop.

### 1.6.1.4.2 Course Taxonomy
The site has an optional flat taxonomy for courses. The taxonomy is currently unused by the site, but if enough course-related projects were to be added, pages to browse projects-by-course could easily be added. See Future Work section on p. 75 for more details.

1. Click Structure > Taxonomy > Course from the administration dropdown menu.
2. Click Add term.
3. Add the course number followed by a colon and the course name in the Name field. Example: INLS 501: Information Resources and Services
4. Do not change any other settings and Click Save.

### 1.6.2 Adding a Single Page
1. Click Content in the administration toolbar.
2. Click the + Add content link in the upper-left corner of the administration overlay.
3. Select Student Project from the list of content types.
4. Enter the title and subtitle of the project in the corresponding fields. Please note that the site Style Guide section on p. 59 specifies sentence case for titles.
   - Example:
     - Original title in title case: My Student Project: A Study About My Information Science Topic
     - Title: My student project
     - Subtitle: A study about my Information Science topic
5. Enter the project description in the About field.
6. Enter the HTML code for the page into the Page Body field.
7. Set the **Text format** to *Full HTML* for most pages. In some cases, the *Full HTML* text format will apply some HTML filtering. If a project contains PHP code or if Drupal applies undesired filtering, set the text format to *PHP code*.

8. Leave the **Page Type** set to the default *Home* value for all single-page projects.

9. Click the **Browse** button under the **Thumbnail** section. Locate the edited image and click the **Upload** button. The file upload may take several seconds or longer depending on file size. The site will create all necessary thumbnails from the original image.

10. Do not change any settings under the following vertical tabs: URL path settings, Revision information, Comment settings, Authoring information and Publishing options.

11. Click **Save**.

### 1.6.3 Adding a Mini-site

The mini-site option allows you to add several linked pages that function as a small site.

1. Click **Content** in the administration toolbar.

2. Click the + **Add content** link in the upper-left corner of the administration overlay.

3. Select **Student Project** from the list of content types.

4. Enter the title of the project in the title field. Please note that the site Style Guide section on p. 59 specifies sentence case for titles.

   Example:
   
   Original title in title case: My Student Project: A Study About My Information Science Topic
   
   Title: My student project
   
   Subtitle: A study about my Information Science topic

5. Enter the project description in the **About** field.

6. Enter the HTML code for the page into the **Page Body** field.

7. Set the **Text format** to *Full HTML* for most project pages. In some cases, the *Full HTML* text format will apply some HTML filtering. If project contains PHP code or if Drupal applies undesired filtering, set the text format to *PHP code*.

8. For projects that will have multiple, related pages added to the content management system, designate one page to be the **Home** and set all subsequent pages to **Subpage**.

9. Click the **Browse** button under the **Thumbnail** section. Locate the edited image and click the **Upload** button. The file upload may take a few seconds. The site will create all necessary thumbnails from the original image.

10. Do not change any settings under the following vertical tabs: URL path settings, Revision information, Comment settings, Authoring information and Publishing options.

11. The site automatically creates URLs for all added pages from the page titles. Any navigation links for related pages in the mini site will have to be revised to reflect the URLs generated by the system.
Example: A mini-site with three pages

<table>
<thead>
<tr>
<th>Page title</th>
<th>Generated URL</th>
</tr>
</thead>
<tbody>
<tr>
<td>My home page</td>
<td>content/my-home-page</td>
</tr>
<tr>
<td>My first page</td>
<td>content/my-first-page</td>
</tr>
<tr>
<td>My second page</td>
<td>content/my-second-page</td>
</tr>
</tbody>
</table>

Any navigation links on the pages would need to be edited to match the generated URLs.

1.6.4 Adding an Existing Site Via an Iframe

If a project cannot be practically added directly to the system, such as a project that relies on a remote database server that will not accept external connections, or if a website already exists on another server and cannot be moved, an iframe can be used to embed the project in a page on the touchscreen system.

1. Click **Content** in the administration toolbar.
2. Click the **+ Add content** link in the upper-left corner of the administration overlay.
3. Select **Student Project** from the list of content types.
4. Enter the title of the project in the title field. Please note that the site Style Guide section on p. 59 specifies sentence case for titles.
   Example:
   - Original title in title case: My Student Project: A Study About My Information Science Topic
   - Title: My student project
   - Subtitle: A study about my Information Science topic

5. Enter the project description in the **About** field.
6. Set the **Text format** to **PHP Code**.
7. Enter the iframe code for the page into the **Page Body** field. Note that the maximum width of an iframe is 1202 pixels.
   Sample code:
   ```html
   <iframe id="frametest" width="1202" height="1900" scrolling="auto" frameborder="0" src="http://ruby.ils.unc.edu/~onyen/project_name/"></iframe>
   ```

8. Leave **Page Type** set to the default **Home** setting for all iframe pages.
9. Click the **Browse** button under the **Thumbnail** section. Locate the edited image and click the **Upload** button. The file upload may take a few seconds. The site will create all necessary thumbnails from the original image.
10. Do not change any settings under the following vertical tabs: URL path settings, Revision information, Comment settings, Authoring information and Publishing options.
11. Click **Save**.

### 1.6.5 Where Files Are Stored

All student project files are stored in the

C:\xampp_final\drupal\sites\default\uploads\projects folder.

### 1.7 Adding Conference Posters

#### 1.7.1 Before You Begin

Obtain an image file and a description of the poster from the student who produced the poster. The description should include information about the poster to provide a viewer with context to enhance his/her understanding of the poster and its relevance. Edit the image based on the Image Guidelines section on p. 68.

#### 1.7.2 Adding a Conference Poster

1. Click **Content** in the administration toolbar.
2. Click the **+ Add content** link in the upper-left corner of the administration overlay.
3. Select **Conference Poster** from the list of content types.
4. Enter the title of the poster in the title field. Please note that the site Style Guide section on p. 59 specifies sentence case for titles.
   
   Example:

   Original title in title case: My Student Project: A Study About My Information Science Topic

   Title: My student project
   Subtitle: A study about my Information Science topic

5. Enter the poster description in the About field.
6. Click the **Browse** button under the **Poster Image** section. Locate the edited image and click the **Upload** button. The file upload may take a few seconds. The site will create all necessary thumbnails from the original image.
7. Do not change any settings under the following vertical tabs: URL path settings, Revision information, Comment settings, Authoring information and Publishing options.
8. Click **Save**.

#### 1.7.3 Where Files Are Stored

All conference poster files are stored in the C:\xampp_final\drupal\sites\default\posters folder.
1.8 Adding Promotional Material

1.8.1 Before You Begin
Obtain a promotional image file from the SILS Director of Communications and edit the image based on the Image Guidelines on p. 68.

1.8.2 Adding Promotional Material
1. Click **Content** in the administration toolbar.
2. Click the **+ Add content** link in the upper-left corner of the administration overlay.
3. Select **Promotional Material** from the list of content types.
4. Enter a title in the title field. Please note that the site Style Guide section on p. 59 specifies sentence case for titles.
   
   **Example:**
   
   Original title in title case: New SILS Promotional Material for the Touchscreen
   Title: New SILS promotional material for the touchscreen

5. Click the **Browse** button under the **Image** section. Locate the edited image and click the **Upload** button. The file upload may take a few seconds. The site will create all necessary thumbnails from the original image.
6. Do not change any settings under the following vertical tabs: URL path settings, Revision information, Comment settings, Authoring information and Publishing options.
8. Click **Save**.

1.8.3 Where Files Are Stored
All student promotional material files are stored in the C:\xampp_final\drupal\sites\default\uploads folder.

1.9 Editing Content
The process for editing existing nodes is the same for all content types. Reference the Adding and Editing Content section on p. 17 and Custom Content Types section on p. 34 for information about specific fields.

1. Click the **Find Content** link in the shortcut bar.
2. To limit the node list to a particular content type, choose an option from the **Type** dropdown list and click the **Filter** button. Note the type filter will remain selected – even after logging out– until it has been reset.
3. Find the node to be edited and click the **Edit** button to the far left of the node title.
4. Edit the values of any of the available fields.
5. Click **Save**.
Basic Site Maintenance and Administration

The following section explains essential site maintenance tasks necessary to stay current with bug fixes and security updates to the Drupal 7 core software and add-on modules. The database and any relevant files must be backed up before conducting any updates.

1.10 Backing Up and Restoring Drupal
All custom files; including site themes, add-on modules, configuration files and uploaded files; are stored in the sites folder. All site data is stored in the database.

1.10.1 Backing Up the Database
1. Click Configuration > Development > Performance from the administration menu.
2. Click the Clear Cache button.
3. Click Configuration > System > Backup and Migrate from the administration menu.
4. Click the Backup now button.
5. Select a location to save backup file to in the resulting dialog box.
6. Click OK or Save depending on browser.

1.10.2 Backing Up System Files
To backup the entire system, including the Xampp webserver, make a copy of the c:\xampp_final folder to external media such as a USB flash drive. To backup just the Drupal 7 site, copy the c:\xampp_final\htdocs\drupal folder to external media. To backup just the custom files for the touchscreen site, make a copy of the c:\xampp_final\htdocs\drupal\sites folder to external media.
1.10.3 Restoring the Database
If the Drupal backend is accessible, the database can be restored with the Backup and Migrate module. If the Drupal system is not functioning the database can be restored with phpMyAdmin.

1.10.3.1 Restoring Via the Drupal Backend
1. Click Configuration > System > Backup and Migrate > Restore from the administration menu.
2. Click Browse and use the resulting dialog box to select a backup file.
3. Click the Restore now button.
4. Click OK or Save depending on browser.

Note that the database restoration process can take a minute or more to complete.

1.10.3.2 Restoring Via phpMyAdmin
1. The SILS Desktop Support and Help Desk Manager or the Director of Information Technology will provide login credentials for phpMyAdmin.
2. After accessing the Windows desktop, open http://localhost/phpmyadmin/ with a browser and log in to phpMyAdmin with the provided credentials.
3. Select drupal from the list of databases on the left.
4. Click the Check All link below the list of tables and ensure that all tables have been selected.
5. Select Drop from the dropdown list next to the Check All link.
6. Click Yes to confirm that all tables should be dropped from the database.
7. Select the Import tab.
8. Click the Browse button and select to select a backup file.
9. Click Go.

Note that the database restoration process can take a minute or more to complete.

1.10.4 Restoring System Files
To restore the all system files, replace the c:\xampp_final folder on the touchscreen hard drive with the xampp_final folder from external media. To restore just the Drupal 7 site, replace the c:\xampp_final \drupal folder on the touchscreen hard drive with the drupal folder on external media. To restore just the custom files for the touchscreen system,
replace the c:\xampp_final \drupal \sites folder on the touchscreen hard drive with the
sites folder on external media.

1.11 Updating Add-on Modules
1. Back up the Drupal database and the entire Drupal site folder: c:\xampp_final
   \htdocs\drupal.
2. Click Modules > Update from the administration menu to see a list of modules that
   need to be updated.
3. Place a check next to all modules to be updated and click Download these updates.
4. Check the Perform updates with site in maintenance mode box.
5. Click Continue.
6. Click the Run database updates link.
7. Click Continue.
8. If database updates are required, click Apply pending updates. If not, click
   Continue to the next step.
9. Click the Administration pages link and then click Configuration > Development
   > Maintenance mode from the administration menu.
10. Uncheck the Put site into maintenance mode box.
11. Click Save Configuration.

1.12 Updating Drupal Core Software
Drupal 7 security updates should be installed as soon as conveniently possible after
release.

1. Download and expand latest update version of the Drupal 7 Core software from
   http://drupal.org/download and expand the ZIP file.
3. Back up the Drupal database and the entire Drupal folder: c:\xampp\htdocs\drupal.
4. Click Configuration > Development > Maintenance mode form the administration
   menu.
5. Check the Put site into maintenance mode box.
6. Click Save Configuration.
7. Delete everything from c:\xampp\htdocs\drupal except the sites folder, and the
   .htaccess and robots.txt files if they have been customized.
8. Copy all of the new files except the sites folder and the .htaccess and robots.txt file
   if customized from the expanded Drupal Core folder from the expanded folder to
   c:\xampp\htdocs\drupal.
9. Restore the Drupal database.
11. Click the Administration pages link and then click Configuration > Development
    > Maintenance mode from the administrative menu.
12. Uncheck the Put site into maintenance mode box.
13. Click Save Configuration.
Note that the above instructions are meant for incremental updates to Drupal 7. When Drupal 8 is released, the Drupal Core developers will also release details on an upgrade path.
Overview of Site Technology

1.13 XAMPP
Xampp is an easily installed and configured bundle of the most commonly used open-source web server software: the MySQL database, the Apache HTTP server, and the PHP and Perl scripting languages. Xampp is an ideal tool for a web development because all of the included software components have been configured to work together, so setting up Xampp can be completed in minutes, whereas a web server can hours or days to set up. More importantly, Xampp is portable; it can be moved to another computer running the same operating system by merely copying the xampp folder to the new computer. Xampp is located in the C:\xampp_final\ folder.

1.14 Browser
Since the touchscreen software is a website, a browser is needed to display it. The browser used must have a kiosk mode, a mode that displays the browser window full-screen, prevents access to the underlying operating system and suppresses messages from the operating system. Some browsers, however, only support full-screen mode, a mode that merely maximizes the browser’s viewing window without preventing access to the operating system. Browsers that only support full-screen mode are unsuitable for the system. The features of kiosk mode, however, vary by browser, and after extensive evaluation and feature comparison, Google Chrome 23 performed the best overall. Since browser features steadily change, site administrators will want to periodically review the available browsers and make sure the system is using the best available choice. Appendix
D on p. 94 has a brief overview of each browser’s strengths and weaknesses for use with the touchscreen site.

1.15 Drupal
As mentioned in the introduction, during the discovery phase of the project I identified what the project should do and what technology it should use, and I quickly realized that the project should be based on web technology. Most of the development courses at SILS focus on web technologies and the browser have rapidly evolved as a medium to deliver the widest range of content possible.

Drupal offers several advantages over a custom solution written in a server-side scripting language such as PHP and a custom database:

- Drupal is easy to install, and its core functionality and module system allows for rapid site development. New sites can be built much quicker in Drupal than a custom-coded site and allows for more time for experimentation.
- Drupal functionality is extensible with over 21,000 easily installed community-contributed modules.
- Since Drupal is open-source software, it has a large, dedicated and enthusiastic community that maintain the code base and regularly release security and bug fixes, whereas a custom solution will only receive fixes as needs are identified and matched to available student developers with the necessary skills.
- Drupal empowers people without a technical background; it doesn’t require special skills to build and extend sites.
- Drupal runs on standard web technologies: It is written in PHP and can use several different databases, HTML, CSS and JavaScript, so developers and designers are
equally empowered to create custom solutions and designs with technology they know
while leveraging the work already done by the community. One can write his/her own
custom modules in PHP and design his/her own visual themes with a mixture of PHP,
HTML and CSS.

Drupal doesn’t require special skills to manage content. Site editors don’t even
need to know HTML or CSS—though knowledge of them is handy. Instead, adding and
editing content is form-driven and WYSIWYG (What You See Is What You Get) editors
can even be added to Drupal.

Drupal files are located in the C:\xampp_final\htdocs\drupal.

1.15.1 Drupal Modules
Much of the core features of Drupal can turned on or off via enabling or disabling the
core modules. Downloading and installing community-contributed modules can also
extend Drupal’s feature set. The Drupal core modules necessary for the touchscreen site
to function and the third-party modules installed to extend site functionality are detailed
in Appendix A on p.86.

1.15.2 Drupal Themes
At minimum, a Drupal theme is a bundle of template files that provide the HTML, CSS
and PHP necessary for page structure, visual design and data. Themes often contain
image files to enhance the visual design and JavaScript to enhance interactivity. At the
time of this writing, the Drupal community has created over 1600 free Drupal themes that
can be downloaded and installed to quickly and easily change the look-and-feel of a site,
but these themes often offer limited customization. For complete flexibility, a new Drupal
theme can be written, but one must be familiar with HTML, CSS, PHP and the Drupal
application programming interface (API) to do so, and the process can also be time consuming. Starter themes offer a middle path between a community-written theme and a completely custom theme by providing base HTML structure, simple CSS for most HTML elements and basic layout, and the PHP code to retrieve database content. Starter themes provide much more flexibility and customization than a full theme, and are much quicker to edit than creating a custom theme.
The touchscreen site with Responsive Blog, a community-contributed theme, enabled

The touchscreen site uses the starter theme Genesis, http://drupal.org/project/genesis, as a base. More information can be found in the Custom Drupal Subtheme: Genesis Touchscreen section on p. 62.
Reference for Site Builders/Developers

When discussing Drupal developers, many in the community make a distinction between coders and builders. When a need arises for new site functionality, coders are likely to write a custom PHP extension, called a module, to handle the need, whereas builders look for a community-contributed module that handles the need. In reality, the distinction is somewhat arbitrary. Coders will readily use third-party modules because they feel no need to reinvent a known solution, and they know that the whole community has vetted third-party modules for bugs. In contrast site builders may never need to write, or even learn to write, PHP because of the vast number of community-contributed modules available –over 21,000 at the time of this writing.

The touchscreen site was designed with builders in mind to allow students with diverse skills and backgrounds to extend site functionality. Future developers are encouraged to follow suit to maintain the widest range of potential developers. Please note that students who are creating projects to be showcased by the site should not be similarly constrained in their approach.

The following information details the all of the current site customizations for future developers who need to modify or expand site structure, features or organization.

1.16 Custom Content Types
Drupal 7 uses content types to organize related subject matter with similar data structure into groups. A site builder can use the two default content types, article and basic page, or
define custom content types. For instance, a site builder could create a Book content type to organize book listings on a site. The site builder could then add fields to the new content type that are common to all books: title, author(s), publisher, publication date and description—among others.

The default content types, article and basic page, are not used by the touchscreen site. Instead, three custom content types organize content nodes into the major site sections: Student Project, Conference Poster and Promotional Material. The fields for each content type are described below.

To add or edit custom content types, select **Structure > Content Types** from the administration menu content types.

<table>
<thead>
<tr>
<th>Field Name</th>
<th>Data Type</th>
<th>Form Element</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Title</td>
<td>text</td>
<td>text field</td>
<td>required field for the name of the project</td>
</tr>
<tr>
<td>Subtitle</td>
<td>text</td>
<td>text field</td>
<td>optional field for projects with multipart titles</td>
</tr>
<tr>
<td>Course</td>
<td>term</td>
<td>dropdown list</td>
<td>optional field, based on the Course taxonomy, for projects related to a particular course</td>
</tr>
<tr>
<td>About</td>
<td>long text</td>
<td>text box</td>
<td>recommended field for a brief explanation of the project</td>
</tr>
<tr>
<td>Head Tag</td>
<td>long text</td>
<td>text box</td>
<td>unused field to allow custom HTML to be added to the head section of page, See the Future Work section on p. 75 for more details</td>
</tr>
<tr>
<td>Page Body</td>
<td>long text</td>
<td>text box</td>
<td>required field for the content or custom code of a project</td>
</tr>
<tr>
<td>Page Type</td>
<td>list</td>
<td>radio buttons</td>
<td>required field for multi-page projects to specify if a given page is the main page or subpage of a project</td>
</tr>
<tr>
<td>Thumbnail</td>
<td>image</td>
<td>file upload</td>
<td>required field for thumbnail upload</td>
</tr>
</tbody>
</table>
### 1.16.2 Conference Poster

<table>
<thead>
<tr>
<th>Field Name</th>
<th>Data Type</th>
<th>Form Element</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Title</td>
<td>text</td>
<td>text field</td>
<td>required field for the name of the conference poster</td>
</tr>
<tr>
<td>Subtitle</td>
<td>text</td>
<td>text field</td>
<td>optional field for posters with multi-part titles</td>
</tr>
<tr>
<td>About</td>
<td>long text</td>
<td>text box</td>
<td>recommended field for a brief explanation of the conference poster</td>
</tr>
<tr>
<td>Poster Image</td>
<td>image</td>
<td>file upload</td>
<td>required field for uploading an image of the conference poster, See Image Guidelines section on p.68</td>
</tr>
</tbody>
</table>

### 1.16.3 Promotional Material

<table>
<thead>
<tr>
<th>Field Name</th>
<th>Data Type</th>
<th>Form Element</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Title</td>
<td>text</td>
<td>text field</td>
<td>required field for the name of the promotional poster</td>
</tr>
<tr>
<td>Image</td>
<td>image</td>
<td>file upload</td>
<td>required field for uploading an image of the promotional, See Image Guidelines section on p.68</td>
</tr>
</tbody>
</table>

### 1.17 Custom Taxonomy

The system employs two custom taxonomies to organize student projects. Instructions for adding new taxonomy terms can be found in the Before You Begin section on p. 17.

Select **Structure > Taxonomy** from the administration menu to add and edit taxonomy terms or add a new taxonomy.

### 1.17.1 Student Projects Taxonomy

The Projects taxonomy uses a two-level hierarchy based on student name and project name. Student names form the top-level hierarchy and project names form the second level. For instance, the taxonomy for a student named John Doe with two projects called *My first project* and *My second project* would be structured as:
Jane Doe
- My first project
- My second project

1.17.2 Course Taxonomy
The site has an optional flat taxonomy for courses. The taxonomy is currently unused by
the site, but if enough course-related projects were to be added, pages to browse projects-
by-course could easily be added. See the Future Work section on p. 75 for more details.

1.18 Nodequeues
The Nodequeue add-on module, http://drupal.org/project/nodequeue, allows hand-sorted
lists of content nodes to be created. The system has one nodequeue for the Promotional
Material content type. The nodequeue allows a site editor to set the rotation order of the
promotional posters that are displayed when the system is idle.

Select Structure > Nodequeues from the administration menu and then the View
link to the right of Promotional Material to edit the nodequeue. Nodes can be easily
added to the nodequeue by typing the name of the node in the autocomplete textbox that
appears below the node list and then clicking the Add content button. Dragging nodes to
the desired location in the list can change node order. Click the Save button or changes
will be lost.

1.19 Custom Views
The Views add-on module, http://drupal.org/project/views, is a feature-rich query builder
and results formatter, that allows site builders to quickly create custom queries and
displays without any knowledge of scripting languages like PHP or database management
systems like MySQL –though a general knowledge of relational database theory is
helpful. The touchscreen site uses the Views module to generate displays that allow users
to browse student projects and conference posters nodes and a display that provides the images for the rotating gallery of promotional material.

<table>
<thead>
<tr>
<th>VIEW NAME</th>
<th>DESCRIPTION</th>
<th>TAG</th>
<th>PATH</th>
<th>OPERATIONS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Conference Posts</td>
<td>Provides all of the list views for the Conference Posters content type</td>
<td>default</td>
<td>/conference-posters-list,</td>
<td>edit</td>
</tr>
<tr>
<td>Display: Page</td>
<td></td>
<td></td>
<td>/conference-posters-grid,</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>/conference-posters-gallery</td>
<td></td>
</tr>
<tr>
<td>Show: Page</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Type: Content</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Taxonomy term</td>
<td>A view to emulate Drupal core's handling of taxonomy/term.</td>
<td>default</td>
<td>/taxonomy/term/%,</td>
<td>edit</td>
</tr>
<tr>
<td>Displays: Feed, Page</td>
<td></td>
<td></td>
<td>/taxonomy/term/%/feed</td>
<td></td>
</tr>
<tr>
<td>Database overriding code</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Type: Content</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Promotional Material</td>
<td>Provides all of the list views for the Promotional Material content type.</td>
<td>default</td>
<td>/promotional-material-list,</td>
<td>edit</td>
</tr>
<tr>
<td>Display: Page</td>
<td></td>
<td></td>
<td>/promotional-material-grid,</td>
<td></td>
</tr>
<tr>
<td>In database</td>
<td></td>
<td></td>
<td>/promotional-material-gallery</td>
<td></td>
</tr>
<tr>
<td>Type: Content</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>nodequeue_1</td>
<td>Display a list of all nodes in queue 'Promotional Material'</td>
<td>nodequeue</td>
<td>/nodequeue/1</td>
<td>edit</td>
</tr>
<tr>
<td>Displays: Block, Page</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>In code</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Type: Content</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Student Project Taxonomy</td>
<td>Provides all of the views for the Student Projects Taxonomy</td>
<td>default</td>
<td>/student-projects-list,</td>
<td>edit</td>
</tr>
<tr>
<td>Display: Page</td>
<td></td>
<td></td>
<td>/student-projects-grid</td>
<td></td>
</tr>
<tr>
<td>In database</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Type: Content</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Main administration page for the Views module

Select **Structure > Views** from the administration menu to add new views or edit settings for existing Views.
1.19.1 Student Projects Taxonomy View

The Grid view generates a list in reverse chronological order by date posted of all student projects that have been added to the system. Each list entry can have the following fields: title, subtitle, thumbnail image, about (a description) and a related course. If an optional field, such as the description, is empty for a given entry, the field will be hidden. The list is formatted as a grid layout going from left to right and then top to bottom. Only eight entries display at a time, but the user can use the pager navigation below (not shown here) to browse through older entries.

Users can navigate to the view by pressing the Student Projects button from the main navigation, but site editors can go directly to the view at the following address: http://localhost/drupal/student-projects-grid.
The custom settings necessary to create the view are detailed below.

**Format**

- Format: Grid
  - Number of Columns: 4
  - Fill up single line

- Show Fields

**Fields**

- Thumbnail Image: Formatter: Image
  - Image style: grid_4col_325x200
  - Link image to: Content

- Title: Link this field to the original piece of content
  - Customize field HTML: H4

- Subtitle: Formatter: Default
  - Customize field HTML: H5

- About: Formatter: Trimmed
  - Trim length: 300

- Course: Formatter: Link

**Filter criteria**

- Content Published (Yes)
- Content Type (= Student Project)

**Sort criteria**

- Content Post date (desc)

**Page settings**

- Path /student-projects-grid

**Pager**

- Use pager Full | Paged, 8 items
1.19.2 Conference Posters

The Grid view generates a list in reverse chronological order by date posted of all conference posters that have been added to the system. Each list entry can have the following fields: poster image (a thumbnail), title, subtitle and about (a description). If an optional field, such as the description, is empty for a given entry, the field will be hidden. The list is formatted as a grid layout going from left to right and then top to bottom. Only eight entries display at a time, but the user can use the pager navigation below to browse through older entries.

Users can navigate to the view by pressing the Conference Posters button from the main navigation, but site editors can go directly to the view at the following address: http://localhost/conference-posters-grid.
The custom settings necessary to create the view are detailed below.

**Format**
Format: Grid  
Number of Columns: 4  
Fill up single line
Show: Fields

**Fields**
Poster Image:  
Formatter: Image  
Image style: grid_4col_325x200  
Link image to: Content
Title:  
Link this field to the original piece of content  
Customize field HTML: H4
Subtitle:  
Formatter: Default  
Customize field HTML: H5
About:  
Formatter: Trimmed  
Trim length: 300  
Customize field HTML: P

**Filter criteria**
Content: Published (Yes)
Content: Type (= Conference Poster)

**Sort criteria**
Content: Post date (desc)

**Page settings**
Path: /conference-posters-grid

**Pager**
Use pager: Full | Paged, 8 items

The Gallery view is an optional view that is currently unused, and like the Grid view, it also generates a list in reverse chronological order by date posted of all conference
posters. Instead of a grid, the list is formatted to display an auto-rotating gallery of the full-sized poster images. Gallery settings are defined by the Conference Posters option set for the Galleria module. See the Galleria section on p. 67 and Galleria Option Sets section on p. 49 for more information. Site editors can go directly to the view at the following address: http://localhost/drupal/conference-posters-gallery.

### 1.19.3 Promotional Materials

The Gallery view generates a list of all promotional posters sorted by the related nodequeue (see Nodequeue section on p. 37) that have been added to the system. The list is formatted to display an auto-rotating gallery of the full-sized poster images. Gallery settings are defined by the Promotional Material option set for the Galleria module. See the Galleria section on p. 67 and Galleria Option Sets section on p. 49 for more information. Site editors can go directly to the view at the following address: http://localhost/drupal/promotional-material-gallery.
The custom settings necessary to create the view are detailed below.

**Format**

Format: Galleria  
Option set: Promotional Material

**Fields**

Promotional Image:  
Formatter: Image  
Image style: None (original image)  
Link image to: Nothing

**Filter criteria**

Content: Published (Yes)  
Content Type (= Promotional Material)

**Page settings**

Path: /promotional-material-gallery

**Pager**

Use pager: Display all items | All items

1.20 Page Manager and Panels

The Page Manager module can be used to control the layout of pages, such as node pages and taxonomy pages, independent of the regions defined by the site theme. By default, Drupal displays all of a node’s fields in a single column and in the same order as the related content type. Page Manager, a part of the Chaos Tools add-on module, http://drupal.org/project/ctools, allows a suite builder to define rules to determine when an alternate layout should be provided for a page type. When a node page is viewed, Page Manager determines if it satisfies a selection rule, and is so, passes the page to the Panels module, http://drupal.org/project/panels, for content and layout. The Panels module allows a site builder to choose the node fields to be displayed, the order of the fields, and
page layout. If the viewed page does not satisfy a selection rule, however, the page is passed back to the main Drupal system and the default layout for the theme is used.

Select Structure > Pages from the administration menu and then the Edit link to the right of the node_view entry to edit the selection rule and panel layouts.

### 1.20.1 Selection Rule and Panel Layout for Student Projects Nodes

The touchscreen site has a selection rule to override the display for Student Project and Conference Poster nodes, so the layout for all node pages of these two content types are handled by the Panels module. Compare the two images below. The first image shows the default display of a Student Project node. As mentioned above, the node page displays all fields and field labels in their default order in a single column. The second image shows the same node, but the selection rule has passed the node to Panels to handle content and formatting. Panels has been set to display only certain fields in a defined order, to hide all field labels and to format the nodes with a two-column layout.
Selection rules
The following selection rules are defined in Page Manager:

The panel will be selected if the node being viewed is type "Student Project".

This panel will be selected if Node being viewed is type "Conference Poster".
1.20.1.1 Panel Layouts
The Panels module has a number of alternate 2-column and 3-column layouts that divide the page into regions to organize a selected page. Once a layout is selected, the site builder can decide which node fields to add to the page and which region the field values will appear in.

![Layout selection option in the Panels module](image)

Once a layout has been selected, each of the data fields of the content type can be placed in any of the regions.
1.20.1.2 Custom Panel Layout: Two column, Main Left

The layout and panel regions defined by the custom Panels layout: Two Column, Main Left

In addition to the default panel layouts, the touchscreen site has a custom layout; Two Column, Main Left; used for Student Project and Conference Poster nodes. The layout consists for four files located in the layouts folder of the Genesis Touchscreen theme.

- `panels-twocol-mainleft.tpl.php`: template file that provides base HTML for the layout and the PHP code necessary to retrieve node content
- `twocol-mainleft.css`: stylesheet defining region sizes and locations
- `twocol-mainleft.inc`: data file describing the layout, including layout name and regions
- `twocol-mainleft.png`: icon for the layout

See the Custom Drupal Subtheme: Genesis Touchscreen section on p. 62 for more information about the Genesis Touchscreen theme. Information about creating custom panel layouts can be found at http://drupal.org/node/495654.

1.21 Custom Image Formats

When images are uploaded via a file-upload field of a content type, Drupal can be configured with custom image formats that automatically apply transformations such as scaling and cropping to the images. The original image and one new image for each
format selected will be saved during the file upload process. Custom image formats can be found by selecting **Configuration > Media > Image styles** from the administration menu.

The touchscreen site uses the following custom image formats:

- grid_4col_325x200  used by grid views
- poster_medium  Scale width 1202, scale to content column width
- poster_large  full-size poster image, Scale width 1920
- promotional_medium  Scale width 1527

### 1.22 Galleria Option Sets

The Galleria module, [http://drupal.org/project/galleria](http://drupal.org/project/galleria), allows a site builder to quickly and easily make custom sets of parameters for Galleria, a jQuery-based framework for image galleries, without any JavaScript programming. On the touchscreen site, Galleria handles the gallery of promotional material that displays when the system is idle. Since the Promotional Material gallery shows advertising posters, it is not meant to be interactive, and therefore most of the parameters in the option set are designed to configure the slideshow player to a static gallery of rotating images.

The Promotional Material option set contains the following parameters:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Theme</td>
<td>classic_minimal</td>
<td>minimal theme</td>
</tr>
<tr>
<td>Thumbnail image style</td>
<td>None (original image)</td>
<td></td>
</tr>
<tr>
<td>Normal image style</td>
<td>None (original image)</td>
<td></td>
</tr>
<tr>
<td>Big image style</td>
<td>None (original image)</td>
<td></td>
</tr>
<tr>
<td>height</td>
<td>950</td>
<td>sets heights of player to 950 pixels</td>
</tr>
<tr>
<td>width</td>
<td>1560</td>
<td>sets the width of the players to 1560 pixels, the maximum width of the content area</td>
</tr>
</tbody>
</table>
imageMargin 0 By default, images are scaled to fill the entire player, but a margin can be set if needed. The Promotional Material gallery does not have a margin.

preload 3 maintains a buffer of three images so that the gallery does not pause to load new images during rotation.

transition flash causes current image to fade into the background when transitioning to next image.

imageCrop Scale down so entire image fits If an image is too large for the gallery, it will be scaled to fit.

showInfo disabled disables the display of image metadata such as a caption and description.

showImagenav disabled disables and hides the next and previous arrows so that the gallery cannot be controlled by users.

thumbnails No thumbnails disables thumbnail creation and hides thumbnail navigation.

autoplay 25000 sets the gallery to auto-advance to the next image after 25,000 milliseconds (25 seconds).

idleMode disabled The player can be configured to behave differently after a set time limit. The option is disabled because the gallery is the idle mode for the touchscreen site. In brief, the idle mode does not need its own idle mode.

clicknext disabled disables the option for a click event to advance the slideshow to the next image.

fullscreenDoubleTap disabled disables the double-tap event.

The touchscreen site also has a similarly configured option set for conference posters if such a gallery is desired at a future date.

Galleria option sets can be found by selecting Configuration > Media > Galleria from the administration menu.
Reference for Site Designers

Visual design changes are excellent cues to users that a site has fresh offerings or undergone major renovation, so designers will need to periodically change the site design. Whether the changes are to be minor tweaks or a completely new design, the following section details my visual design decisions here so that future designers can make informed changes that purposefully improve the design in the case of tweaks and give food for thought for a new design. The section also provides some resources for newcomers to design and a detailed explanation of the custom theme developed for the site.

1.23 Design Resources

Designers should be familiar with HTML and CSS before beginning. Novices should consume these books in roughly the order listed below. More experienced designers can browse topic areas of interest.

1.23.1 Foundational Design Skills


1.23.2 Design Theory


1.23.3 CSS Techniques

1.23.4 Design Inspiration


1.23.5 Quick Start for New Designers
Read The Non-designer's Design Book first, because it is a crash course in design and can be read in an evening or two. Browse a layout guide like The New Big Book Of Layouts and search the Internet for examples of current design trends to draw inspiration from.

1.24 Current Site Design
The following section details the layout of all major pages via wireframe diagrams, explains the role of Panels layout in the design, explains the philosophy of the current design and provides a brief style guide.
1.24.1 Wireframes

Student Projects and Conference Posters Grid View

Wireframe for Student Project Nodes
Wireframe for Conference Poster Nodes

Please note that the wire frames for student project nodes and conference poster nodes are nearly identical. While conference posters should have a landscape orientation, and thus completely fit onscreen when scaled to fit 1202 pixels wide, whereas student projects can have any size or orientation as long as the width is 1202 pixels or less. Student projects can flow below the fold, so room for a scrollbar is necessary in the layout. If a scrollbar appears, is the navigation shifts to the left shrinking the margin between it and the description area.
1.24.2 Custom Page Layouts with Panels
Drupal default layout system is cumbersome and limited. Theme creators define regions that are applied by default to every page on the site. If a different layout is needed for some pages, the theme creator must write a template override file using HTML, PHP and the Drupal API for each page that requires a different layout. The touchscreen site uses the Panels module instead, because a designer or site builder can chose from a number of layouts that can work within or ignore the theme’s default regions. Any number of layouts can be used and are applied to pages based on rules defined by the designer. The Panels module also provides a drag-and-drop interface to add and remove content from the chosen layout. See the Page Manager and Panels section on p. 44 for more information.
1.24.3 Design Philosophy

An example of the touchscreen site’s visual design

The touchscreen site is meant to have a minimalist design so that users focus on student content and not the visual design. Users subjectively judge content based on site design, however, so the design does need to be appealing. Since the touchscreen site has a prominent place and large visual impact in the lobby of Manning Hall, it may be a part of newcomers’ first experience with SILS, so the visual design needs to help orient them to SILS and subtly communicate SILS culture. To that end, the SILS logo has a prominent location at the top of the screen to ground the user in a sense of place: he/she is at SILS.

The navigation area uses shades of blue derived the official Carolina Blue, PMS 542 or hex code #56A0D3 (UNC Creative). Carolina blue is understandably a popular color for student work, so the navigation background color is a much darker shade to prevent the navigation area from blending in with any projects that use the official color.

SILS teaches two disciplines, library science and information science, and the site’s design is meant to subtly show the harmonious interplay between the approaches.
The aspects of the designed influence by the library focus on the most iconic element of the library: the book. As mentioned in the introduction, large-format digital signage is becoming commonplace, but large-format touchscreens are comparatively rare, so to encourage users to touch the screen, I took a photograph of a book with a rich-textured leather binding for the background image. The texture is meant to evoke the tactile pleasure one receives when browsing a library shelf and berry-picking books based on the binding. The navigation area has a ribbon-like shape and appearance similar to a bookmark, and the SILS logo is placed at the top of the navigation to reinforce the bookmark imagery. Together, the background image and ribbon navigation are meant to look like a book with a bookmark casually laid upon it. For typography, I used a serif font from the Palatino family, Palatino Linotype, for page title and body text to represent the rich history of the printed page and the technological advance of the printing press.

The user interface elements are meant to represent information science and are sparse, crisp stark, modern and minimalist. Design has a vocabulary, and user interfaces are meant to readily draw upon recognizable elements to reduce the cognitive load for users so that they may spend more time consuming and interacting with content rather than learning the quirks of a visual design. Since most users only experience with touchscreens are mobile devices, I drew inspiration from common user interfaces like Apple’s iOS and Windows 8. In particular, the grid layout of icons on the Student Projects and Conference Posters are reminiscent of mobile touchscreen app pages. While I mentioned in the previous section that the navigation area is meant to be reminiscent of a bookmark, I avoided a skeuomorphic representation in favor of a simpler, modern look similar to tiles in Windows 8. While designers successfully use skeuomorphism to
translate a metaphor from the physical world to the digital world, they do so at the risk of creating a kitschy, overwrought style that distracts a user from the content to focus on the design (Carr, 2012). The navigation links have a pill-like shape and slight gradient background for 3D appearance that have become a commonplace graphical representation of a button. While the blue gradient is carefully color matched to Carolina Blue, it is also reminiscent of Apple’s Aqua look and iOS buttons. For typography, I used a sans serif font, Verdana, for headings because Bernard, et al. showed that Verdana was a strongly preferred font from a group of twelve popular online fonts among test subjects because it was perceived as legible and business-like and had a comparatively quick reading time during testing (2001).
1.24.4 Style Guide

Colors

- **#56A0D3 (Carolina Blue)**: Base Color
- **#2E536F**: Navigation Background
- **#9AC9FF**: Navigation Button Border
- **#6093BC**: Textual Links
- **#C77C7A**: Textual Links when clicked
- **#868938**: Page Title
- **#000000 (Black)**: Background Image
- **#FFFFFF (White)**: SILS Logo, Text

Fonts

- Palatino Linotype: Page Title, Body Text
- **Verdana Bold**: Grid View Titles, Navigation Buttons
- Verdana: Grid View Subtitles, Pager Links

Text Treatments

- **Page Title**: All Capital Letters
- **Grid View Titles**: Sentence Case

Assets

- SILS Logo: sils-logo.png
- Site Background: touchscreen-background.jpg

1.24.5 Large Screen and Touchscreen Design and Usability Considerations

While many ways the look-and-feel for this project was just one more website design, I
had to adapt my skills to suit touch technology that I was unfamiliar with and learn a new
set of design considerations for a 46-inch, vertically-mounted touchscreen computer. For
instance, Nielson’s eye-tracking research provides strong evidence for placing a site’s
main navigation at the top or left sides of a page, because users tend to quickly scan a
webpage in an F pattern, scanning the top of the page from left to moving and then moving down the left side while doing more left-to-right scans (2006). On the touchscreen site, however, simple ergonomics complicate the issue. Top navigation is impractical because the size of screen and the mounting height would require people of average height to reach above eye-level to touch links while bottom navigation would be noticed last, if at all, by users scanning in an F pattern. Since most people are right-handed, left-side navigation proved awkward as well, because users would have to reach across their bodies with their right hand or step back-and-forth to touch navigation links. Right-side navigation, then, is the best choice, because while it is not as readily noticed during an F-scan as top and left navigation, it is much easier to reach. Interestingly, Outing and Ruel found during their eye-tracking research that users had “more eye fixations and longer viewing duration” on right-side navigation than left and suggest that the “novelty” of right-side navigation may have caused it (2004). To ensure that users noticed the navigation and to reduce their cognitive load, I made the navigation a large, colorful area that reaches from top to bottom and is persistent on all pages.
The red arrows indicate the likely scan pattern of a sample page from the touchscreen site. The size, color, persistence and location of the navigation make it likely to be noticed by users.

Touchscreen buttons offer interesting challenges too. When using a mouse, web users scroll the page looking for elements with hover states to discover clickable elements, but touchscreens do not support hovering. Instead, buttons and links need to be obviously clickable and visually enticing to encourage users to poke at the element to see if it is interactive. Clicking buttons can be difficult for users if the button is smaller than their fingers (Forlines, Wigdor, Shen, & Balakrishnan, 2007, p. 647), but vertical screen orientation encouraged accurate pressing, whereas horizontal plane accuracy was affected by the distance of the target button from the user (p. 652), so I gave the navigation buttons and the grid-view icons large target areas. Touchscreens cannot provide the haptic feedback that physical keyboards can, so Lee and Zhai recommend visual, auditory and vibrato-tactile feedback to alert a touchscreen user of a successful click (2009, pp. 310-311). The hardware for the touchscreen site is capable of vibrato-tactile feedback, and the traffic and noise level of the Manning Hall lobby make sound cues implausible.
much of the time, so, the touchscreen site relies solely on visual feedback to register
successful clicks. All buttons and icons highlight when touched and shift slightly down
and right to simulate the look of a pressed button. Text-only links highlight when pressed,
but do not shift, because they do not resemble buttons enough to make the visual illusion
appear realistic.

The range of motion required for scrolling on such a large screen proved awkward,
so scrolling behavior was eliminated wherever possible. Instead, grid layouts show
multiple rows and columns to fill available screen space and pager navigation links allow
users to browse through multiple pages of icons instead of scrolling. As discussed in the
Wireframes section on p. 53, student project nodes are the only pages that allow
scrolling, because student projects do not have height limitation and can flow below the
fold of the page.

1.25 Custom Drupal Subtheme: Genesis Touchscreen
As mentioned in the Overview of Site Technology section, a Drupal theme is a bundle of
template files that provide the HTML, CSS and PHP necessary for page structure, visual
design and data retrieval. Themes often contain image files to enhance the visual design
and JavaScript to enhance interactivity. The touchscreen site uses the starter theme
Genesis, http://drupal.org/project/genesis, as a base. To customize Genesis, a copy –
called a subtheme– must be made, and modifications are made to the subtheme. The
subtheme for the touchscreen site is called Genesis Touchscreen.
The touchscreen site with Genesis, a starter theme, enabled but not customized

The touchscreen site with Genesis Touchscreen, a custom subtheme of Genesis, enabled

The following subsections explain: the use of the emerging HTML5 and CSS3 web standards all CSS files for the theme, brief information about custom template files, and the JavaScript used by the theme.
1.25.1 HTML5 and CSS3 features
Even though HTML5 and CSS3 are still developing standards at the time of this writing, the web development and design community has enthusiastically embraced them, and these standards enjoy solid support from some browsers like Google Chrome and Mozilla Firefox. While the touchscreen site does not use any new HTML5 elements, such as nav or section, and only few new CSS3 properties; background gradient, border-radius, and box-shadow; the site was built with these standards in mind to future-proof it as much as possible. As the HTML5 and CSS3 standards evolve, the theme can be modified to enhance site functionality and visual design.

1.25.2 Theme CSS Files
The Genesis starter theme comes with the following CSS files to provide basic page layout and styling. The files can be found in the C:\xampp_final\drupal\sites\all\themes\genesis\genesis_touchscreen\css folder.

blocks.css The stylesheet provides an outline for CSS rules for all Drupal blocks. The rules are mostly empty of properties because the site was designed to use panels (see the Page Manager and Panels section on p. 44) instead of blocks.

comments.css The stylesheet provides an outline for CSS rules for the Drupal commenting system, but the file is unused because the commenting system is disabled.

custom.css The stylesheet provides a place for custom or miscellaneous CSS rules that do not belong in any of the other CSS files.

fields.css The stylesheet provides global CSS rules for the default Drupal fields. The file is unused.

html-elements.css The stylesheet provides rules for all HTML elements. The file contains much of the custom design for the Genesis Touchscreen subtheme.

navigation.css The stylesheet contains rules for navigation elements, including the main menu, breadcrumb and pager navigation.

nodes.css The page contains global CSS rules to be applied to all nodes
The stylesheet contains layout rules for the Drupal regions provided by the Genesis theme. The file contains much of the custom design for the Genesis Touchscreen subtheme.

views-styles.css The stylesheet contains custom CSS rules for views (see the Custom Views section on p.37).

1.25.3 Custom Template Files
The Genesis Touchscreen theme does not use template overrides files for custom page layouts, because designers need to be familiar with PHP and the Drupal API to create template overrides. See the Page Manager and Panels subsection of the Reference for Site Builders/Developers section for more information on page layouts. Minor modifications have been made to the default template files for the Genesis Touchscreen theme, but a detailed explanation of the changes is beyond the scope of this document. The changes are easy to spot by comparing Genesis Touchscreen template files to the original Genesis files.

1.25.4 JavaScript
The following section provides resources for learning JavaScript and jQuery, describes the theme’s custom JavaScript, and describes the use of the jQuery library and two third-party jQuery projects: Galleria and Colorbox. More information about using JavaScript and jQuery with Drupal 7 can be found at http://drupal.org/node/171213.

1.25.4.1 JavaScript Resources
The following books JavaScript and jQuery books cover a broad spectrum of topics and should be useful for beginning and advanced programmers.

McFarland’s book is an excellent starting point for newcomers to JavaScript and jQuery. The book is written in an easily grasped, conversational and concise style with excellent concrete examples and hand-on opportunities.


White’s book is a successful blend of reference and how-to guide. The book progresses from beginning topics to advanced concepts such as object-oriented JavaScript and integrating JavaScript with other dynamic web technologies.


Flanagan’s book is the most comprehensive JavaScript reference book available. The book has been recently revised to include topics such as jQuery and server-side JavaScript.

### 1.25.4.2 Custom JavaScript Files

The Genesis Touchscreen theme uses two custom JavaScript files. The files can be found on the C:\xampp_final\drupal\sites\all\themes\genesis\js folder. An explanation of each file is given below, and the code for each file is in Appendix B on p.89.

#### 1.25.4.2.1 timed-redirect.js

When the system has been idle for ten minutes, a JavaScript file, timed-redirect.js, causes the system to randomly pick one of the main sites sections and redirects the browsers to the main page for the selected section. The redirect serves two purposes: If the promotional material page is selected, the system goes into idle mode and displays a gallery of SILS advertising posters. If either the student projects or conference posters grid view is selected, the redirect provides passersby with an opportunity to see a site section they might not have seen before.
1.25.4.2.2  *page-transition.js*
When a user selects a link, *page-transition.js* causes just the document body to fade in
during the page transition while the background image and site navigation remains
visible. Since the site background and navigation are persistent and static throughout the
entire site, the code achieves the effect by hiding the columns div until after the page has
fully loaded and then uses a fade-in effect that lasts 500 milliseconds. The fade in effect
is meant to make the system feel more like an application than a traditional website.

1.25.4.3  *Installed jQuery Projects*

1.25.4.3.1  *Galleria*
*Galleria*, http://galleria.io/, is a jQuery-based framework for image galleries. The
framework makes setting up and customizing the look-and-feel and the functionality of
an image slideshow quick and easy even for beginners. Customization is handled via
Galleria themes, bundles of CSS and JavaScript files that define the style of the gallery,
such as a slideshow player or thumbnail grid, and option sets, parameters like the size of
the player and auto-rotation of images. *Galleria* has an accompanying Drupal module that
allows a site editor to quickly create custom sets of gallery options via the Drupal
administrative backend. More information can be found in the *Galleria Option Sets*
section on p. 44.

1.25.4.3.2  *Colorbox*
*Colorbox*, http://drupal.org/project/colorbox, is a jQuery plugin to create a lightbox effect
for images. The lightbox allows a user to tap a thumbnail image to see a large version of
the image. The lightbox feature is used on the touchscreen site to allow users to see
conference posters at the largest proportional size that will fit on the screen. Colorbox has an accompanying Drupal module that allows for simple configuration via the Drupal administrative backend.

1.26 Image Guidelines
The touchscreen site makes extensive use of images, and therefore it requires good images for visual appeal. Files received for the touchscreen site will almost certainly need to be processed for web use. The saving of a file for use on a website is a process that must balance file size and image quality, and the following section will provide background information and guidelines for using the Save for Web feature in Adobe Photoshop to create web-ready files that are good quality, quick-loading and tailored to the specific needs of the touchscreen site.

1.26.1 Source Files
The suitability or quality of source file places the single greatest limitation on the final files. The following chart provides quick reference for ideal sources files for conference posters promotional material and icons for student project images.

**Conference Posters and Promotional Material**
- Page orientation: landscape
- Preferred format for document files: PDF
- Preferred formats for image files: PNG or TIF
- Ideal image resolution: 1920 x 1080 pixels or as close as possible

**Student Project Icons**
- Page orientation: landscape
- Preferred format for document files: PDF
- Preferred formats for image files: PNG or TIF
Ideal image resolution 325 x 200 pixels or larger

All source files will need to be in a format that Photoshop can open: an image file format or an Adobe PDF. Otherwise, a designer or site editor would need access to every—often obscure—software package that a submitter could have used to create the work. Many file formats, such as Microsoft Word or PowerPoint, are also unsuitable because font substitution or text reflow problems are likely to occur if the file is opened on a computer other than the one that was used to create it. Overall, Adobe PDFs are the best source file for any document file because they will look exactly like the original file on the original computer. If a document cannot be exported to PDF, most programs can export to image formats, but care must be taken to ensure that the resolution of the resulting file is large enough for the desired use on the touchscreen site. PNG-24 and TIFF formats are the best source formats for photos because they support lossless compression and most programs can export to these formats. Appendix C describes likely source file formats—especially image file formats—in more detail.

1.26.2 Optimizing and Saving Files for the Web
As mentioned previously, saving a file for use on the web is a process that must balance file size and image quality. File size can be adjusted by the use of image compression. Lossy compression will result in smaller files, but potentially affect image quality. If a file has been saved at too low quality the image looks terrible, whereas if the quality has been set too high the file size can be quite large and will be slow to download. The touchscreen loads files from the local hard drive instead of downloading them from a remote server, so the file sizes can be larger for the touchscreen site than most sites. As
discussed in the previous section, any file format that Photoshop can open will work, but PDFs and lossless image formats such as TIFF and PNG work best.

Using the example image below, the following steps will show how to create files that are good quality, quick loading and tailored to the specific needs of the touchscreen site.

1. Click **File > Open** from the menu bar in Photoshop and select the file to be opened in the resulting dialog box. Click **Open**.
2. Most file types will open, but not PDFs and Camera Raw files require further input.

![A sample poster in PDF format](image-url)
PDF

PDF Import dialog box

If the selected file is a PDF, the Import PDF dialog box will appear. The dialog box allows you to adjust the width and height of the imported file. The screen resolution of the touchscreen site is 1920 pixels wide by 1080 pixels high (1920 x 1080), and the size of the imported conference posters and promotional material need to be adjusted as close as possible screen resolution. Student project icons only need to be 325 x 200 pixels. Ensure that the dropdown lists of unit types are set to pixels for width and height.

If the width and height are significantly less than 1920 x 1080 pixels (or 325 x 200 for student project icons), the images is too small and will need to be resubmitted at a higher resolution.

If the width and height are approximately 1920 pixels and 1080 pixels respectively, but not larger, the values should be left as-is.

If either the width is greater than 1920 pixels or the height is greater than 1080 pixels, set the width to 1920 pixels and the height will be automatically calculated. If the resulting height is larger than 1080 pixels, set the height to 1080 pixels, and the width will be recalculated. The resulting dimensions should be the largest resolution that will completely fit on screen. Click OK.

Camera Raw
If the selected file is a camera raw file, the Camera Raw plugin-in will open. Click Open Image.

3. If the image was not resized during import, select Image > Image Size from the menu bar, and follow the same process for resizing a PDF in step 2.
4. Press **Command+1** in OS X or **Control+1** in Windows to view the image at 100 percent. Scroll around the image looking for any visual problems such as pixilation. If you discover any problems, report them to the student to see if she can fix the problem.

   For instance, the SILS logo in our sample poster is a small image—just 40 x 240 pixels—obtained from the Internet and stretched to twice that size. To fix the problem, a vector copy of the SILS logo could most likely be obtained from the SILS Director of Communications to replace the low-resolution logo in the original poster. The poster would then need to be converted to a PDF again.

5. From the menu bar, select **File > Save For Web & Devices** and a dialog box will appear. In the top right corner, select the **2-Up tab** to see the original file in one pane and the export image in the other pane.
6. Each pane has a label and file size below it. One pane will be labeled Original and one will be labeled with the export file type. Click in the export pane.

<table>
<thead>
<tr>
<th>Original: &quot;sample poster&quot;</th>
<th>PNG-24</th>
</tr>
</thead>
<tbody>
<tr>
<td>7.15 MB</td>
<td>6 sec @ 768 Kbps</td>
</tr>
</tbody>
</table>

The original image is displayed in the left panel and the export version is displayed on the left.

7. The image can be saved in a number of formats and a balance between file size and image quality must be found. Select the dropdown menu for file type in the upper right corner just below the Preset dropdown menu. Try setting the file type to PNG-24, because it provide the best image quality. Check the file size below the export pane; if the file size is below 750K, the image will load on the touchscreen site without a noticeable pause for the user. Continue to step 9.

8. Click and hold in the export frame; drag the mouse around the image and compare the two images. Pay special attention for image distortion, called artifacting, especially around areas with full color and text. If artifacting has occurred, increase the quality until the artifacting disappears. See the right panel in the image below for an example of artifacting. Continue to step 9.

PNG-24 selected

If the file size for the PNG-24 is above 750K, change the file type to JPEG and set the quality to 80. If the file size is above 750K, adjust the quality down until the file size is below 750K, but do not go below a quality of 50.

JPEG selected and Quality set to 69
Export frame (on right) shows artifacting

If the image has a simple range of colors, such as a poster without any photos or gradients, try PNG-8 or GIF to create the smallest file possible. Note that PNG-8 and GIF file types only support 256 colors.

9. Click **Save**.
Future Work

While I had used with other CMSs before beginning work on this project, I had only briefly experimented with Drupal. I learned an immense amount about Drupal and web content management while working on the project, but like most systems, the touchscreen site can be improved. I identified the following improvements and fixes after the development phase of the project was completed and during the evaluation phase: add alternate browsing paths for content discovery via the student and course taxonomies, perform technical fixes to the system, add new technological features, investigate methods to increase site traffic and student submissions.

1.27 Alternate Browsing Paths

1.27.1 Browse by Taxonomy
By default, each taxonomy term automatically receives a page with links to all nodes associated with the term. Therefore, the taxonomy terms are more than just an organizational tool; they provide alternate browsing paths for site users. Since the Student Project taxonomy uses a student’s name as a top-level term and project name as a second level term, projects could be browsed by student name. The touchscreen site also uses an optional flat taxonomy for courses so that projects could be browsed by related course.

As mentioned above, the taxonomy pages already exist, but are unused. Since Drupal’s default behavior provides almost no visual styling, Views or Page Manager and Panels should be used to create multi-columnar layout and text formatting.
1.27.2 Browse by Breadcrumb
Once taxonomy pages are in use, breadcrumb navigation should be added to allow a user to navigate back along a path she has been exploring. For instance, if a user has browsed to Jane Doe’s taxonomy page to see all of her projects, and then selected *Project one* for viewing, the user should be able to use the breadcrumb navigation to travel back to Jane Doe’s project list or back to the student list.

While Drupal Core does support breadcrumb navigation, numerous add-on modules are available to modify or extend the system so that the breadcrumb navigation can be tailored to meet future system needs. Add-on modules can be found at [http://drupal.org/project/modules](http://drupal.org/project/modules).

1.28 Improvements and Fixes
Most systems have opportunities to improve functionality, and the touchscreen site is no exception. I identified the following improvements and fixes after the development phase of the project was completed and during the evaluation phase.

1.28.1 Head Tag for Student Projects
During the planning stage, I identified a potential need for HTML entities, such as links to external CSS files, to be added to the head element of student project nodes. I added a text field for head tags to the Student Project content type; however, the Head Tag field is not rendered in the head of the document, because Drupal maintains a rigid separation of content and markup. Drupal considers all field values to be content, so it renders them exclusively in the body section of a page. Browsers will still use head elements in the body section, but the page will have invalid markup and may render improperly.
The Head tag field has been excluded from the Panel layout for student project nodes to prevent the field content from rendering until a solution is found. After further experience with Drupal, I have the following three suggestions:

1. A template override file for the student project nodes could be added to the system to insert the Head Tag field value for the current node into the document.
2. A custom add-on module could be written to add the Head Tag values to the head element of the document.
3. The Head Tag field could be deprecated and alternate solutions found on a case-by-case basis. For instance, an embedded stylesheet could be used in place of a link tag for an external stylesheet. The link tag could also be added to the body of the document if pages display properly—even though the resulting markup would be invalid.

Refer to Pro Drupal 7 Development for instructions for template overrides and custom module creation.

1.28.2 Modify the Idle Timer and Redirect System

The touchscreen site has a JavaScript-based timer that will redirect the browser to a randomly selected section once the site has become inactive. The idle timer begins when a page is loaded and is reset when a user-triggered event, such as screen tap, occurs. Therefore, the idle timer should prevent the touchscreen site from redirecting as long as a user is interacting with the system; however for security purposes, the browser is unaware of any interaction with embedded content such as an iframe or a Flash file.
To illustrate the security risk, consider the following scenario: If user interaction with embedded content did cause event triggers, a hacker could create a fake wrapper page for a password-protected site and use the wrapper page to capture user input.

As a workaround, the idle timer has been set at a high value, twenty-five minutes, because a user is unlikely to interact with an embedded object for more than twenty-five minutes without triggering an event such as a stray touch or navigation to a new page. Therefore, the timer is unlikely to expire while a user is interacting with the system, but it could happen occasionally, so the idle timer should be reworked to overcome this shortcoming. JavaScript may not be able handle the problem because of the above-mentioned security prohibition, and another solution may need to be found such as recreating the idle timer and redirect system at the operating system level where the security concern will not exist.

If a JavaScript solution is found, the timed-redirect.js file should be removed from the theme and a custom module should be written for it so that a site editor can change the idle delay and edit the list of destination pages without having to know any JavaScript. Instead, the editor would be able to change the values via the configuration option for the module in the GUI.

1.29 Technological Enhancements

1.29.1 Drush
Drush, http://drupal.org/project/drush, provides a command-line shell for Drupal. Site administrators who are familiar with the command line will find that Drush can simplify many time-consuming tasks, such as updating Drupal core and downloading, enabling, disabling and updating modules. Scripting can provide further convenience by
automating complex or repetitive tasks. A complete list commands and an easy Windows
installer for Drush can be found at http://drush.ws/.

1.29.2 Roles
Drupal is configured by default with three roles: anonymous user, administrator, and
authenticated user. An anonymous user is anyone who browses the site. An administrator
can log in to the Drupal backend and perform all backend tasks such as adding content or
updating modules. An authenticated user can log in to the backend and perform any tasks
delegated by an administrator in the People > Permissions section.

Drupal can also be configured with custom roles in the People > Roles section. At
the time of this writing, the site was not configured to use the authenticated user role or
any custom roles, but it may be desirable to configure custom roles to divide labor. For
instance, a site editor role could be created with only enough to add, edit and delete
content. That way, one or more site editor accounts could be created to maintain content
so that administrators could focus on site building and maintenance.

1.29.3 Extend the Backup System
1.29.3.1 Schedule backups
While backups of the database should be performed at minimum when the Drupal core
software and modules are updated, occasional backups are unlikely to offer enough
protect against an unexpected corruption of the database. In the event of database
corruption, all new content since the last backup will be lost. The Backup and Migrate
module, http://drupal.org/project/backup_migrate, can be configured to perform
scheduled backups via a simple interface at Configuration > Backup and Migrate >
Schedules.
1.29.3.2 Add file backup module
By default, the Backup and Migrate module can perform backups of the database, but not
the site files. A companion module, Backup and Migrate Files, extends the Backup and
Migrate module to allow file backups as well. More information can be found at
http://drupal.org/project/backup_migrate_files.

1.29.3.3 Managing backup files
Automated backups can create a large build-up of files, but another companion module,
Backup and Migrate Prune automatically removes old backup files based on the
following logic:

- Keep all backups for the current day
- Keep a backup per day for the current week
- Keep a backup per week for the current month
- Keep a backup per month for the current year
- Keep a backup per year for previous years

(Drupal)

More information can be found at http://drupal.org/project/backup_migrate_prune.

1.29.4 Redirects for outside links
The touchscreen site is meant to be a walled garden so that users can only have access to
site content, but since the site runs in a standard browser, any external link will direct the
users off site. All student projects and other content should be scrubbed of external links
before posting to the touchscreen site, but eventually external links will be missed. The
Page Manager module has a HTTP Response code plugin,
http://drupal.org/project/page_manager_redirect that should be able to redirect all
external links back to a specified page on the touchscreen site to prevent users from
accidentally browsing to external pages.
1.29.5 Evaluate kiosk mode, CSS3 & HTML 5 features
As mentioned in Browser section on p. 29, Google Chrome 23 performed best overall in kiosk mode at the time of this writing, but it did have drawbacks: Chrome does not always suppress dialog boxes from the browser and does not disable all keyboard shortcuts such as Esc to exit full-screen display or Alt+Tab to change switch windows. Browser development proceeds at a steady pace, so site administrators should periodically evaluate all major browsers for new kiosk mode features that will enhance the security of the touchscreen system and for emerging CSS3 and HTML5 features that will provide an enhanced browsing experience for users. The following sections detail the strengths and weaknesses of each major browser at the time of this writing and should serve as criteria for site administrators when evaluating newer versions of the browsers. Appendix D on p. 94 has a brief overview of each browser’s strengths and weaknesses for use with the touchscreen site.

1.30 Enhance Site Appeal
The touchscreen site has the potential to be a centerpiece of the lobby of Manning Hall. The key is to increase student participation so that the touchscreen site regularly receives good, fresh content and to increase awareness that the touchscreen site is interactive. Those two goals are complementary: increasing awareness will increase traffic and increased participation will increase awareness and traffic.

1.30.1 Site Traffic Analysis
The site does not currently have any method to track user interaction, but such data would be useful to determine which pages of the site receive traffic. This data could be used to help site administrators understand what content appeals to users and then tailor future content submission requests to better match user preferences.
Google Analytics is the most used web analytics reporting software, and therefore enjoys the most documentation. The software uses a page tagging system to transfer user interaction data to its third-party suite of tools.

Set up is straightforward:
1. Create a Google account for the system at http://www.google.com/analytics/. Be sure to make note of the unique web-property ID that Google provides.
2. Download and install the Google Analytics module:
   http://drupal.org/project/google_analytics.
3. Configure the module using the web-property ID.

Since Google Analytics is a third-party service and data is transferred to their servers, Google’s privacy policy should be carefully reviewed with the SILS Director of Information Technology before installing. If Google Analytics unsuitable for any reason, an open-source logfile analysis tool like AWStats,
http://awstats.sourceforge.net/, can be installed directly on the XAMPP server to provide analysis tools without transferring data to third-party serves. Both Google Analytics and AWStats maintain good online documentation at

1.30.2 Motion Sensing Camera
Coupling a motion sensor device to a large-format touchscreen system in public thoroughfare presents interesting opportunities to entice users to interact with the site and enhance their experience. If a motion-sensing device were added to the system, it could be used to trigger an attention-getting signal to attract passersby, add an extra layer of interactivity to the site and add another element for student developers and designers to incorporate into projects.
While any motion-sensing camera will likely work, the Microsoft Kinect for Windows, http://www.microsoft.com/en-us/kinectforwindows/, has several characteristics that make it a compelling choice. The Kinect is widely available and affordable. Since the Kinect sensor was designed with variable room size in mind, it works well at close and long ranges. Users who are familiar with touch technology may be somewhat disappointed by the current touchscreen site because the hardware does not support touch gestures, but the Kinect can provide enhanced interactivity and additional input methods with three-dimensional, motion-based gestures and speech input. The Kinect is particularly well suited to large-screen systems that do not have keyboards or mice such as the touch screen site (Liebling & Meredith, 2012, p. 105).

The Kinect’s popularity has also spawned several developer toolsets, and JavaScript is the ideal choice to integrate with the Kinect with the touchscreen site. The following development tools are available.

**Kinect for Windows SDK**  
A software development kit that allows development of Windows applications with gesture and voice recognition in in the C++, C# and Visual Basic programming languages  

**DepthJS**  
an open source browser extension that provides JavaScript support for Chrome and Firefox  
http://depthjs.media.mit.edu/ (MIT Media Lab, 2010)

**Kinesis**  
a framework to provide Kinect support in any browser, requires Windows 7 and the Kinect for Windows SDK  
http://kinesis-docs.heroku.com/

**KinectJS**  
A project to provide JavaScript support for the Chrome 16+ and Firefox 8+ browsers, requires Windows and the Kinect for Windows SDK  
http://kinect.childnodes.com/

Designers and developers looking for inspiration for motion-based ideas should investigate the Link Media Wall at Duke University and the Bystander motion-based
interactive museum gallery. The Link Media Wall is a tiled wall of 18 display screens that can react to the motion. The wall is available for the Duke community to create “engaging, dynamic and interactive exhibits” (Duke University). The Bystander project used projectors to display historical, Australian crime scene photos grouped in “flocks” on the wall of a museum gallery (Robertson, Mansfield, & Loke, 2006, pp. 32-33), and the “room itself would respond to the presence and activity of those in the room by changing what material was revealed and how it was displayed (Kan, Robertson, Muller, & Sadler, 2005, p. 16).

1.30.3 Enhance Student Participation and Site Traffic
Student participation and audience growth are inextricably related and complementary goals for the touchscreen site. Increased content will cause increased interest and vice versa, because at minimum, developers and designers will want to show off their accomplishments. One key to growing student participation and site traffic is to increase student awareness of the touchscreen site and its development opportunities through the many communication channels available at SILS such as the SILS listservs, mailings and Manning Hall bulletin boards. Of course, the best opportunity to promote the touchscreen site is the site itself; posters that advertise development opportunities and encourage students to submit project proposals should be added to the site’s promotional material slideshow. If properly promoted, development opportunities could become a talking point among SILS students and create word-of-mouth advertising for the site.

Site administrators need to work with the Director of Information Technology and SILS Director of Communications to engage the larger SILS community and seek out natural partnerships to create student opportunities and excitement for the touchscreen
site. When promoting the site, present it as an opportunity for students who are interested in semester-long independent studies or undergraduate honors theses. Engage student groups such as ISSUE and ASIS&T to create enthusiasm for the touchscreen site with events like coding contests.

Content from any course that teaches any aspect of web development or design of any sort could be showcased on the touchscreen site. As instructors elect to create or retool projects to work with the touchscreen site, whole classrooms become word-of-mouth advertisers. Site traffic will increase as students discover that they can use the touchscreen system as a tool to evaluate what is learned in a course, and as a result, interest in courses that showcase projects should increase.
Appendix A: Enabled Modules

The Drupal core modules necessary for the touchscreen site to function and the third-party modules installed to extend site functionality are detailed below. Each entry in the lists includes a module title, installed version number and a brief description provided by the module author.

### Core Modules

<table>
<thead>
<tr>
<th>Module</th>
<th>Version</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Contextual links</td>
<td>7.12</td>
<td>Provides contextual links to perform actions related to elements on a page.</td>
</tr>
<tr>
<td>Dashboard</td>
<td>7.12</td>
<td>Provides a dashboard page in the administrative interface for organizing administrative tasks and tracking information within your site.</td>
</tr>
<tr>
<td>Database logging</td>
<td>7.12</td>
<td>Logs and records system events to the database.</td>
</tr>
<tr>
<td>Field UI</td>
<td>7.12</td>
<td>User interface for the Field API.</td>
</tr>
<tr>
<td>Help</td>
<td>7.12</td>
<td>Manages the display of online help.</td>
</tr>
<tr>
<td>Menu</td>
<td>7.12</td>
<td>Allows administrators to customize the site navigation menu.</td>
</tr>
<tr>
<td>Number</td>
<td>7.12</td>
<td>Defines numeric field types.</td>
</tr>
<tr>
<td>Overlay</td>
<td>7.12</td>
<td>Displays the Drupal administration interface in an overlay.</td>
</tr>
<tr>
<td>PHP filter</td>
<td>7.12</td>
<td>Allows embedded PHP code/snippets to be evaluated.</td>
</tr>
<tr>
<td>RDF</td>
<td>7.12</td>
<td>Enriches your content with metadata to let other applications (e.g. search engines, aggregators) better understand its relationships and attributes.</td>
</tr>
<tr>
<td>Search</td>
<td>7.12</td>
<td>Enables site-wide keyword searching.</td>
</tr>
<tr>
<td>Shortcut</td>
<td>7.12</td>
<td>Allows users to manage customizable lists of shortcut links.</td>
</tr>
<tr>
<td>Syslog</td>
<td>7.12</td>
<td>Logs and records system events to syslog.</td>
</tr>
<tr>
<td>Trigger</td>
<td>7.12</td>
<td>Enables actions to be fired on certain system events, such as when new content is created.</td>
</tr>
<tr>
<td>Update manager</td>
<td>7.12</td>
<td>Checks for available updates, and can securely install or update modules and themes via a web interface.</td>
</tr>
<tr>
<td><strong>Add-on Modules</strong></td>
<td><strong>Version</strong></td>
<td><strong>Description</strong></td>
</tr>
<tr>
<td>--------------------</td>
<td>-------------</td>
<td>-----------------</td>
</tr>
<tr>
<td>Administration menu</td>
<td>7.x-3.0-rc2</td>
<td>A better Toolbar.</td>
</tr>
<tr>
<td>Toolbar style</td>
<td>7.x-0.0</td>
<td></td>
</tr>
<tr>
<td>Advanced help</td>
<td>7.x-1.0</td>
<td>Allow advanced help and documentation.</td>
</tr>
<tr>
<td>Backup and Migrate</td>
<td>7.x-2.2</td>
<td>Backup or migrate the Drupal Database quickly and without unnecessary data.</td>
</tr>
<tr>
<td>Colorbox</td>
<td>7.x-1.3</td>
<td>A light-weight, customizable lightbox plugin for jQuery 1.4.3+.</td>
</tr>
<tr>
<td>Custom content panes</td>
<td>7.x-1.0</td>
<td>Create custom, exportable, reusable content panes for applications like Panels.</td>
</tr>
<tr>
<td>Custom rulesets</td>
<td>7.x-1.0</td>
<td>Create custom, exportable, reusable access rulesets for applications like Panels.</td>
</tr>
<tr>
<td>Date All Day</td>
<td>7.x-2.5</td>
<td>Adds 'All Day' functionality to date fields, including an 'All Day' theme and 'All Day' checkboxes for the Date theme and Date popup widgets.</td>
</tr>
<tr>
<td>Date Popup</td>
<td>7.x-2.5</td>
<td>Enables jQuery popup calendars and time entry widgets for selecting dates and times.</td>
</tr>
<tr>
<td>Date Repeat Field</td>
<td>7.x-2.5</td>
<td>Creates the option of Repeating date fields and manages Date fields that use the Date Repeat API.</td>
</tr>
<tr>
<td>Date Tools</td>
<td>7.x-2.5</td>
<td>Tools to import and auto-create dates and calendars.</td>
</tr>
<tr>
<td>Date Views</td>
<td>7.x-2.5</td>
<td>Views integration for date fields and date functionality.</td>
</tr>
<tr>
<td>Features</td>
<td>7.x-1.0-rc2</td>
<td>Provides feature management for Drupal.</td>
</tr>
<tr>
<td>Feeds Admin UI</td>
<td>7.x-2.0-alpha4</td>
<td>Administrative UI for Feeds module.</td>
</tr>
<tr>
<td>Feeds SQL</td>
<td>7.x-1.0</td>
<td>Imports data from an external database using Feeds.</td>
</tr>
<tr>
<td>Feeds XPath Parser</td>
<td>7.x-1.0-beta3</td>
<td>Parse an XML or HTML document using XPath.</td>
</tr>
<tr>
<td>Galleria</td>
<td>7.x-1.0-beta2</td>
<td>Turns image fields of nodes into Galleria galleries with this JavaScript (JQuery) image gallery.</td>
</tr>
<tr>
<td>Mini panels</td>
<td>7.x-3.2</td>
<td>Create mini panels that can be used as blocks by Drupal and panes by other panel modules.</td>
</tr>
<tr>
<td>Module filter</td>
<td>7.x-1.6</td>
<td>Filter the modules list.</td>
</tr>
<tr>
<td>------------------------</td>
<td>---------</td>
<td>-------------------------</td>
</tr>
<tr>
<td>Nodequeue Service</td>
<td>7.x-2.0-beta1</td>
<td>Provides a nodequeue service.</td>
</tr>
<tr>
<td>Page manager</td>
<td>7.x-1.0</td>
<td>Provides a UI and API to manage pages within the site.</td>
</tr>
<tr>
<td>Panel nodes</td>
<td>7.x-3.2</td>
<td>Create nodes that are divided into areas with selectable content.</td>
</tr>
<tr>
<td>Pathauto</td>
<td>7.x-1.0</td>
<td>Provides a mechanism for modules to automatically generate aliases for the content they manage.</td>
</tr>
<tr>
<td>Site map</td>
<td>7.x-1.0</td>
<td>Display a site map.</td>
</tr>
<tr>
<td>Stylizer</td>
<td>7.x-1.0</td>
<td>Create custom styles for applications such as Panels.</td>
</tr>
<tr>
<td>Views content panes</td>
<td>7.x-1.0</td>
<td>Allows Views content to be used in Panels, Dashboard and other modules which use the CTools Content API.</td>
</tr>
<tr>
<td>Views UI</td>
<td>7.x-3.3</td>
<td>Administrative interface to views. Without this module, you cannot create or edit your views.</td>
</tr>
</tbody>
</table>
Appendix B: Custom JavaScript Code

1.31 Code for timed-redirect.js

```javascript
var timer;
var redirectDelay = 600000;
var randomNumber;
var urlArray = [
  "http://localhost/drupal/",
  "http://localhost/drupal/promotional-material-gallery/",
  "http://localhost/drupal/conference-posters-grid/"
];
var redirectURL;

selectURL();
startTimer();

document.body.onblur = resetTimer;
document.onmousedown = resetTimer;
document.onkeydown = resetTimer;
document.onscroll = resetTimer;

function selectURL() {
  redirectURL = '0';
  randomNumber = Math.floor(Math.random() * (urlArray.length));
  //alert("length: " + urlArray.length + ' ' + "Random: " + randomNumber);
  redirectURL = urlArray[randomNumber];
}// end selectURL

function startTimer() {
  timer = setTimeout('redirectPage(redirectURL)', redirectDelay);
}

function endTimer() {
  clearTimeout(timer);
}

function resetTimer() {
  endTimer();
  startTimer();
}

function redirectPage(targetURL) {
  var currentURL = window.location.href;  //local variable scope
  ensures use of current URL.
  // Redirect will not occur if page has an admin overlay. If
  Overlay module
  // is disabled, the redirect will not occur on any admin pages.
  if (currentURL.indexOf("#overlay") == -1) {
    endTimer();
    window.location.assign(targetURL);
  }
}
```
1.32 Code for page-transition.js

jQuery(document).ready(function() {
    jQuery("#columns").css("display", "none");
    jQuery("#columns").fadeIn(500);
});
Appendix C: Image and Graphic File Formats

Before discussing specific file formats, some general information about graphic file formats is helpful.

Raster images, often-called bitmap images, store data as a grid of pixels with assigned color values. The size of the grid is called resolution, and it is an important consideration for image files for the touchscreen site, because images with a small resolution will look pixelated if enlarged. For instance, the touchscreen site has a resolution of 1920 x 1080 pixels, and if a 300 x 400 pixel image were displayed full screen, it would look terrible because of the pixilation.

Raster images generally contain a large amount of data to map the location and color of each pixel and therefore often have large file sizes. Most image file formats use compression techniques to reduce the file size for faster downloading. Lossy compression techniques can produce small file sizes when compared to the original by discarding color, but can also result in a loss of image quality. Lossless compression techniques do not discard color data, but achieve much more modest compression results than lossy compression.

In contrast, vector images mathematically describe graphics and therefore can be scaled to any size without becoming pixelated. Browsers offer no support for vector graphics except for limited support for SVG (scalable vector graphics), so vector files almost always need to be converted to a raster format such as JPEG, PNG or GIF.

While Photoshop supports an exhaustive list of image and graphics formats, the following are some common ones likely to be submitted.
GIF  a raster image format that generally creates small files because it only uses 256 colors, supported by all browsers, best for images with simple color palettes such as web graphics, logos and posters with colors and no photos file extension: .gif

PNG-8  a raster image format designed specifically for the web to replace GIFs, generally creates small files because it only uses 256 colors, only supported by modern browsers, best for images with simple color palettes such as web graphics, logos and posters with colors and no photos file extension: .png

JPEG  a raster image format with lossy compression that supports millions of colors, is a common export format for most programs, supported by all browsers, best used for photos or images with complex color palettes, some digital cameras only capture images in JPEG format file extensions: .jpg, .jpeg

TIFF  a raster image format that supports millions of colors with lossless or no compression, used primarily for print-ready documents, is a common export format for most programs, Safari is the only major browser to support TIFFs, a preferred format for image source files file extension: .tif

PNG-24  a raster image format with lossless compression and support for millions of colors, designed specifically for the web to replace JPEGs, only supported by modern browsers, file size is generally larger than JPEGs, a preferred format for image source files file extension: .png

Camera Raw  a variety of raster image formats that store native, uncompressed data captured by a digital camera, support for each camera must be added to Photoshop so newer cameras are only supported by newer versions for Photoshop, visit http://www.adobe.com/products/photoshop/extend.html to see a complete list of cameras supported by Photoshop CS6 common file extensions: .dng, .cr2, .nef

PSD  the native file format for Photoshop that can contain raster and vector data, can suffer from font substitution so it is best to convert the file to an image format on the computer that created it. file extension: .psd
PDF a universal file format for document distribution that can contain raster and vector data, prevents font substitution and text reflow so that the PDF looks like the original document, can be converted to web-friendly formats by Photoshop, the overall best format for document source files file extension: .pdf

EPS a format used mostly for print-ready documents that can contain raster and vector data, can suffer from font substitution file extension: .eps

For more detailed information about all file formats that Photoshop can open go to:

Overall, the PNG-24 and TIFF formats are the best source formats for photos because they support lossless compression and most programs can export to these formats. Lossless compression formats are preferable for source files because the final files created for the touchscreen site will likely be compressed as well, and the resulting file will have lost color data during both compression processes. Adobe PDFs are the best source file for any document file because they will look exactly like the original file on the original computer. If a document cannot be exported to PDF, most programs can export to image formats, but care must be taken to ensure that the resolution of the resulting file is large enough for the desired use on the touchscreen site.
Appendix D: Evaluation of Browsers for Kiosk Mode

1.33 Opera
Opera 12 provided the most configurable and secure kiosk mode. Opera hides all operating system messages including notifications about updated software or virus definitions, suppresses dialog boxes from the browser, hides all browser toolbars and chrome and disables all keyboard shortcuts such as Esc to exit full-screen display or Alt+Tab to change switch windows. Opera was eliminated from consideration because scroll bars are hidden by kiosk mode, and scroll bars are a necessary element for the touchscreen site.

1.34 Mozilla Firefox
Firefox 16 does not have a kiosk mode, but one can be added through third party add-ons. Even though most of the third-party add-ons do function well, Firefox was eliminated from consideration because add-ons can cease to function after Firefox has been updated until the add-on has been updated as well.

1.35 Microsoft Internet Explorer
Internet Explorer 9 was eliminated from consideration because it has limited supported for CSS3.

1.36 Google Chrome
Like Opera, Google Chrome 23 hides all operating system messages including notifications about updated software or virus definitions, hides all browser toolbars and most chrome elements such as toolbars. Unlike Opera, Chrome does not always suppress dialog boxes from the browser, does not disable all keyboard shortcuts such as Esc to exit
full-screen display or Alt+Tab to change switch windows, but it does support scroll bars in kiosk mode. Since the system does not have a keyboard attached, the active keyboard shortcuts are not problematic. The scroll bars are a necessary element so Chrome was the best overall choice for kiosk mode.
Bibliography


