

APPENDIX U

GRAPHICAL USER INTERFACE STANDARDS FOR WEB APPLICATIONS

GUI Standards for Web Applications

for Legacy Modernization

Pennsylvania Department of Transportation
Bureau of Business Solutions & Services
Data Administration Division



Table of Contents

TABLE OF CONTENTS	3
DOCUMENT INFORMATION.....	5
REVISION HISTORY	5
PURPOSE.....	6
INTENDED AUDIENCE.....	6
INTRODUCTION	7
USER INTERFACE DESIGN	8
RESPONSIVE AND NON-RESPONSIVE DESIGN.....	8
SELECTING AN APPROACH.....	9
TEMPLATES.....	9
PAGE LAYOUT	11
REQUIRED CORE ELEMENTS	11
<i>Top Navigation</i>	11
<i>Banner Header</i>	11
<i>Primary Display Area</i>	12
<i>Footer Area</i>	15
VERTICAL AND HORIZONTAL SCROLL BARS	16
BODY GUTTER.....	16
USABILITY AND FORMATTING	17
BODY COLORS.....	17
<i>Multiple Rows</i>	17
FONTS	17
<i>Acceptable Font Families</i>	17
<i>Font Styling</i>	17
<i>Relative Font Size</i>	17
TABBING ORDER.....	18
LINK TREATMENT.....	18
<i>New Browser Windows</i>	18
<i>Link Clickability Cues</i>	18
<i>Text Links</i>	19
<i>Link Names</i>	19
<i>Embedded Links</i>	19
<i>Link Length</i>	19
<i>Accessibility</i>	19
BUTTON TREATMENT.....	20

<i>Icons</i>	21
<i>Accessibility</i>	21
WAIT TIME.....	21
TABLES	22
<i>Accessibility</i>	22
ACCORDIONS	22
<i>Accessibility</i>	22
FORM CONTROLS.....	23
<i>Text Input</i>	23
<i>Dropdowns</i>	24
<i>Checkboxes</i>	24
<i>Radio Buttons</i>	25
ALERTS (MESSAGES)	25
<i>Content</i>	26
<i>Placement</i>	26
<i>Accessibility</i>	26
ACCESSIBILITY	27
VARIOUS DISABILITIES TO CONSIDER.....	27
ACCESSIBILITY AND RWD	28
ARIA	29

Document Information

Revision History

Date	Version	Author	Description of Change
9/10/2015	0.1	Don Chavey	Initial draft of document assembled.
9/22/2015	0.2	Don Chavey	Update Introduction to address needs of user groups and clarify scope of GUI standards. Provide considerations for selecting responsive or non-responsive approach. Update standard elements in top navigation, banner and footer.
10/6/2015	0.3	Don Chavey	Simplify organization of Usability and Formatting section. Add new content on accordions, checkboxes, and radio buttons. Add accessibility guidance from U.S. Web Design Standards and intersperse accessibility guidance at appropriate sections.
3/16/2016	1.0	Don Chavey	Replaced placeholders with GUI element examples from PennDOT PDJF Framework and Placards application being redeveloped. Removed reference to future UI template library and added PDJF responsive CSS Grid. Added guidance for when responsive template is not used. Updated standard banner header and footer descriptions based on Placards. Addressed tab order and conditionally required fields. Provided additional guidance on button icons, wait time, tables, text input, and data formatting examples. Updated guidance on alerts, notably adding severity level descriptions.

Purpose

This document details the graphical user interface (GUI) standards and requirements for web applications produced and maintained for the purpose of conducting official PennDOT business.

Intended Audience

This is a moderately technical document intended for user interface designers and developers. It is also intended for use by project managers and architects who must plan development efforts and ensure that PennDOT standards are followed.

Introduction

The Legacy Modernization program recognizes that modern and consistent GUI design approaches are vital to PennDOT's strategic goals of transforming business operations and citizen services delivery. This document provides graphical user interface (GUI) design standards for applications delivered through a web browser to employee, customer and business partner end users working on network-connected and mobile devices.

These standards aim to provide a standard look and feel and, at the same time, to effectively address the needs of these different user groups:

- **Customers** need a simplified user interface and less complex language. These users come from many different backgrounds and employ a large variety of hardware and software to access applications. To serve them, web sites need to be compatible with popular browser choices and to operate similar to popular public and commercial web sites.
- **Business partners** need an efficient yet simplified user interface as well. Web sites should be arranged for many user types but can contain more business-related terminology as these users are familiar with the business. Again, they need to be compatible with popular browser choices.
- **Employees** primarily need an efficient user interface. But at the same time, this interface needs to make it easy to train new employees. It also should provide enhanced features to increase productivity. While a single standard browser can be expected, employees vary in preferences and background, so the application must accommodate keyboard as well as mouse/trackpad users.
- **All user groups** have come to expect that modern web applications have the ability to adapt to their need or desire to resize their browser and type size at will. Also, it is broadly required and expected that web applications are accessible, so that persons with disabilities can navigate and obtain the same or equivalent information available to persons without disabilities

This document discusses an approach to GUI design that considers the application type, end users and devices targeted. It provides standard page layouts and provides guidelines on usability and formatting. Last, but definitely not least, it discusses requirements to serve users with various disabilities.

This document features UI element examples from the PrimeFaces UI component library within the PennDOT Java Framework (PDJF), which was created as a generalized framework for building modern Java applications. The standards and guidelines in this document, however, are generally applicable to web applications regardless of detailed technical underpinnings.

This document does not address native (iOS or Android) mobile apps. It is common to confuse mobile web sites with mobile apps. The latter require users to download a local copy of app from an app store (or receive an app push), as opposed to accessing a remote application via a browser. A mobile app is indicated when there is a need to leverage built-in functions of the device, such as dialing, geolocation calendar integration, photos, and so forth.

Acknowledgement

The initial version of this document borrowed heavily from the [State of Michigan Look and Feel Standards for Web Applications and Sites](#), Version 7.0, August 2015 for responsive web design.

It is also based on accessibility guidelines for Federal web sites in the [U.S. Web Design Standards](#), alpha version published September 2015.

User Interface Design

Responsive and Non-Responsive Design

A 2015 Cisco white paper¹ points out that mobile traffic now accounts for more than half of total internet traffic, calling for new UI design approaches. Designers are quickly getting to the point of being unable to keep up with the endless new resolutions and devices. For many web sites, creating a web site version for each resolution and new device would be impossible, or at least impractical.

Responsive web design (RWD)² is a design approach that suggests that web sites/applications should respond to the user's behavior and environment based on screen size, platform and orientation. This eliminates the need for a different design and development phase for each new gadget on the market.

A site designed with RWD adapts the layout to the viewing environment by using fluid, proportion-based grids, flexible images, and CSS3 media queries, in the following ways:

- The fluid grid concept calls for page element sizing to be in relative units like percentages, rather than absolute units like pixels or points.
- Flexible images are also sized in relative units, so as to prevent them from displaying outside their containing element.
- Media queries allow the page to use different CSS style rules based on characteristics of the device the site is being displayed on, most commonly the width of the browser.

A responsive web site might show a mobile-style top menu with a single column layout when viewed with a smart phone, but tabs or global navigation buttons and a 2-column layout when viewed with a tablet, and a 3-column layout when viewed on a laptop or desktop monitor. All three devices can see the same content; the server "responds" to the device with a layout optimized for viewing on that device.

RWD has gained rapidly in acceptance in the general marketplace, and it has been adopted a number of state governments, including Pennsylvania's. Google is now judging how "mobile-friendly" a site is and is giving preference to RWD in search results served to mobile devices. RWD is not a panacea, however. It should be recognized that:

1. Responsive design first works out how an application will display and work on mobile devices, and it can take more time to develop and test up front. This may represent the best or worst ROI³, depending whether mobile will figure strongly or at all in application usage.
2. Although RWD adapts the display to a smaller device like a smartphone, it still sends all of the data to that device that it would to a desktop. If this is an insurmountable issue, a dedicated mobile site or mobile app may be a better approach than RWD.

Accordingly, it is expected that RWD will be appropriate for some applications, while a traditional, non-responsive design approach will be appropriate for others, including many core PennDOT applications.

¹ Cisco Visual Networking Index: Global Mobile Data Traffic Forecast Update 2014–2019 White Paper

² Proposed by Ethan Marcotte in "[Responsive Web Design](#)"

³ "return on investment"

Selecting an Approach

A site that was built traditionally cannot be effectively “converted” to responsive. It needs to be baked into the design. When planning for responsive web, the need is to design for mobile first, ensuring the content is organized and easy-to-read when compressed to a mobile device.

Accordingly, it is suggested that, at the outset, each development project select a design approach. It is impractical to provide a hard and fast decision tree, but considerations should include:

- Expected end user groups – members of the public (individuals and organizations), PennDOT business partners, and PennDOT or other Commonwealth employees
- Devices that may be used to access the application -- desktop/laptop, tablet, smartphone
- Density of the functionality and data presented to the user by the web site/application

In general, a responsive approach is best when a variety of device types may be used. Conversely, a non-responsive approach may be best for internal, data-rich applications intended for delivery through a desktop browser.

Templates

Most projects should use a responsive *template*, even to develop an application that will not use responsive *design approach*. The primary reasons for this are:

1. It should represent no additional effort.
2. All users expect, and sometimes rely on, the ability to resize windows, zoom in/out, and change text size in their browsers. The responsive templates will allow the application to adapt to these changes, automatically adjusting presentation of the application functionality.
3. PennDOT has set emphasis on a mobile workforce, with more tablets and smartphones deployed, as an enterprise strategic focus area for FY 2016 – 2018. The responsive templates would give the application better ability to work effectively across devices and form factors, even if this is not a current requirement.

Following is an example of a responsive template provided by PDJF UI component library. The PrimeFaces Grid CSS is optimized for mobile devices, tablets and desktops. Up to 12 columns are supported based on fluid and fixed modes.

Sample Grid CSS from PDJF UI Component Library

1	1	1	1	1	1	1	1	1	1	1	1
2		2		2		2		2		2	
3			3			3			3		
4				4				4			
6						6					
12											
4				8							
3			9								
2		5					5				
4			2		2		3			1	

If a responsive template is not available, or will not be used for any reason, pages should be optimized for a screen resolution of 1024 x 768 pixels, as the great majority of desktop displays are set at it or above. The layout of page elements should appear “as intended” at this resolution, and no horizontal scroll bar should be presented for the page. If possible, the presentation should accommodate a ‘liquid’ or dynamic width and should resize itself to accommodate higher resolution displays.

Page Layout

Required Core Elements

All application designs must incorporate these core elements for a common, consistent presentation layer: Top Navigation, Banner Header, Primary Display Area, and Footer.

Top Navigation

The top navigation is a key element to all online service sites representing PennDOT. The goal is to present the user an official PennDOT unified, brand appearance that carries across to all applications.

To maintain brand appearance, color requirements include white font and graphics overlaying the branded background color which is dark blue, HTML color code 002D5A.

Sample Top Navigation Rendered on Desktop



The top navigation includes right-aligned embedded links, some of which are required. The top navigation serves as the primary space for the following utility links:

- A link to the application home (landing) page, enabling direct return from other pages (Required)
- A link to user help (Required)
- A link to a site map to aid navigation for users with screen readers or non-JavaScript compliant browsers (Optional)
- A contact link to give users an access to contact information regarding the application, such as contact email, mail or fax information (Required)
- A link back to the CoPA portal site represented by the PA.gov brand logo (Required)

Considerations for Responsive Top Navigation

For ALL instances the top navigation is 44px high, background extends the page width, and content is restricted to parent container dimensions.

Phone & Tablet Portrait

- A dropdown menu system is required to access responsive top navigation utility links.
- The required height of the menu system links is 44px.
- PA.gov is never nested in the menu system.

Banner Header

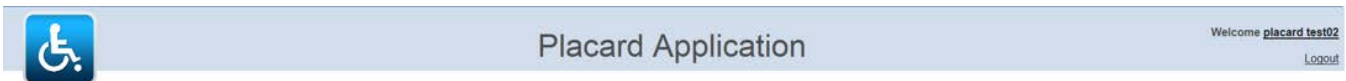
In Department *web content sites*, the banner is a large area (minimum 1295px x 460px) with the PennDOT logo and several agency links superimposed on images appropriate to PennDOT's mission. For *web applications*, the banner header is shorter to help keep functionality above the fold, and it generally does not contain imagery to help reduce visual distraction. It contains:

- Left-aligned, either an application-specific logo or the PennDOT logo plus tagline acting as an active link back to the PennDOT home page
- The web site title, which can be the application name or the name of a board, commission, or program

- Right-aligned, if login is required, the logged-in user and, as appropriate, role(s) right-justified
- Right-aligned, if login is required, the login or logout link, depending on the user's signed-in state

If a sub-agency name is used in the web site title, the PennDOT tagline must be displayed. Using the PennDOT tagline clearly communicates to the user which department is responsible for the application, regardless of internal acronyms or program names. It promotes the department's legal authority to provide the service or transaction.

Sample Banner Header Rendered on Desktop



The banner header may be solid-filled or, for desktop or tablet, may contain a white and colored gradient. To maintain brand appearance, color requirements include black font and graphics overlaying the light blue background color, which is HTML color code D1DDEC.

Additional Banner Header Sizes

If the application display area needs to be maximized when on desktop, smaller versions of the banner header may be used. The smaller version is equally sized and proportioned and 100% width of the parent container but may differ in height dimensions. The key elements of the banner components, must remain clear and consistent – even as the header reduces in size.

Considerations for Responsive Banner Header

The banner header will always be 100% width of the parent container. The logo, web site title, and image tagline must be embedded into the image and NOT individual DOM elements.

Desktop and Tablet Landscape

- Standard height: 130px
- Additional size: 100px

Tablet Portrait

- Minimum height scale: 100px

Phone

- Minimum height scale: 50px
- No background image or gradients allowed
- Only solid filled agency color may be used

Primary Display Area

The primary application display area includes the primary user interface and functionality. Elements include a combination of navigation, main body, and right sidebar.

- The main body area is required.

- Main navigation is needed depending on size and complexity of the application. If needed, it is located horizontally under the banner header area or vertically in the left sidebar.
- The right sidebar is optional.

These factors will help determine which of the above components should be used:

1. Will the application have a single, dedicated process flow where the user will be guided through a set of screens, from beginning to end, resulting in a final submission page?
 - a. If YES, then a Body Area Only design will best accommodate the application.
2. Will the application contain distinct, multiple sections with different results or inputs for each section?
 - a. If YES, then incorporating main navigation will best accommodate your design; horizontal or vertical depending on content and real-estate.
3. Will the site include distinct separate sections, supporting or related content, links to outside sites, or include help files?
 - a. If YES, then incorporating the right sidebar will best accommodate the design.

Body Area Only

In cases where the application requires no navigation and will utilize the entire body area for functionality and user introduction information, the Body Area Only template can best accommodate this design.


An example scenario would include a single page application or home screen.

Considerations for Responsive Body Area

- Body area will always be 100% width of the parent container.

Sample Body Area Only

Home Help Site Map ContactUs
PA.GOV



Placard Application

Welcome [placard_test02](#)
[Logout](#)

Search Customer

WID:

Placard Number:

DL/ID Number:

OR

Last Name:

First Name:

Date of Birth: (MM/DD/YYYY)

Business Name:

Business Id:

City:

Search Results

Showing 1-4 out of 4

Name	Address	City	State	ZIP
SAMUEL ABATE	143 BUTTUNWOOD ST	JESSUP	PA	19434
SANDRA ABRAHAM	5 COLONIAL ACRES	WYOMING	PA	19644
SARAH ABEL	111 HARPERS FERRY RD	BEAVER FALLS	PA	15010
SADEK ABDEL	45 W PHILADELPHIA AVE	MORRISVILLE	PA	19067

Release: 1.0
PennDOI Home Help Site Map ContactUs
Copyright © 2014 Pennsylvania Department of Transportation. All Rights Reserved.
[Web Policies](#)
Tue, Mar 1, 2016 03:32:33 PM

Main Navigation

Main navigation is optional and can be displayed horizontally or vertically within the primary display area. Horizontal and vertical main navigation cannot be used together.

- Horizontal Main Navigation

Choosing to use horizontal main navigation will depend on content and real-estate. Smaller menus will best accommodate this design. At no time should a navigation element wrap, breaking the design.

An example scenario would include multiple navigation elements (5 or less) and/or navigation elements with short text.

- Horizontal Main with Left Child Navigation

There may be scenarios in which the main navigation categories will require child navigation. Child navigation only exists after navigating away from the homepage.

An example scenario would include a multiple page application with nested navigation.

- Vertical Main Navigation

Choosing to use vertical main navigation will allow for greater flexibility with less real-estate concerns as using horizontal main nav.

An example scenario would include a multi-page app with many menu items or lengthy text that cannot fit horizontally.

- Vertical Main with Child Navigation

There may be scenarios in which the main navigation categories will require child navigation. Child navigation only exists after navigating away from the homepage and will always reside nested within vertical main navigation.

Right Sidebar

If a design requires an isolated section for supporting or related content, using the optional right sidebar will best accommodate this design. This section can be used for links outside of the site that related to the body content or for help files and tips. Right sidebar can exist in body area only, with vertical main navigation, or horizontal main navigation.

Footer Area

The footer area contains links to PennDOT.gov and the PennDOT Privacy Policy. In addition, it provides users with easy access to all utility top navigation links even after scrolling to the bottom of a page:

- A link to the application home (landing) page, enabling direct return from other pages
- A link to user help
- A link to a site map to aid navigation for users with screen readers or non-JavaScript compliant browsers
- A contact link to give users an access to contact information regarding the application, such as contact email, mail or fax information

Final elements include display of a required copyright statement and optional release, date and time information.

Sample Footer



Considerations for Responsive Footer Area

For ALL instances the footer must appear beneath the primary application display area and content must be center-aligned. When minimal body content is present, the footer must remain at the bottom of the screen.

Desktop, Tablet Landscape and Tablet Portrait

- Links appear inline
- Copyright ending with current year

Phone and Tablet Portrait

- “Back to Top” Link

Phone

- Links are block elements and 44px high
- Copyright hidden

Vertical and Horizontal Scroll Bars

Applications should attempt to include as much information “Above the Fold” and limit vertical scrolling when possible. This offers the user ease of access to information. Critical features such as login and core navigation should never appear “Below the Fold”.

“Below the Fold” refers to that area at the bottom of the browser screen that limits what can be displayed to the user.

Applications should avoid horizontal scroll bars, especially when on a mobile device. Horizontal scrolling causes many usability conflicts and is considered a poor application of information architecture. Users will often miss details or valuable information if displayed off the screen and will constantly be required to move the screen to see all of the information. The net result is users tend to avoid sites that require too much scrolling.

Body Gutter

The body area should also contain built-in margin accommodations at the far left and right of the display area. A gutter ranging from 10-20 pixels is recommended.

Usability and Formatting

Body Colors

WCA (Web Content Accessibility) standards ensure that content is accessible by everyone, regardless of disability or user device. To meet these standards, text and interactive elements should have a color contrast ratio of [at least 4.5:1](#). This ensures that viewers who cannot see the full color spectrum are able to read the text.

For this reason, it is suggested to use black text with white as the background for the body of the application.

If you choose another color combination, this [color contrast tool](#) is a useful resource for testing the compliance of any color combination.

Multiple Rows

- When displaying multiple rows of output, row background should alternate between light gray and white to distinguish between the rows.

Fonts

Acceptable Font Families

- Verdana
- Arial

Font Styling

- Use non-bold black text for the font weight and color.
- Ensure that the format of common items such as headings, subheadings, labels and body text are consistent throughout the application.
- Use proper sentence casing; do not use all-caps fonts or text.
- Text should be left-aligned. Center-aligned text must be avoided.
- Font styles such as but not limited to: comic sans, *calligraphy*, *scripts*, *brushes*, **BLOCK**, or **over-styled** typefaces must be avoided and are not business appropriate for official business.
- Blinking and moving text are an accessibility problem for people with photosensitive epilepsy and visual impairments and must be avoided.
- Font colors should adopt the look and feel of the parent agency site and must provide optimum contrast against the background color. Body text default colors, unless otherwise specified within the application style or server side includes, should be black (#000000) text on a white background.

Relative Font Size

Different browsers may not display font size consistently. If you want your site to be usable and accessible you will want to allow users to resize text.

The base font-size for the site should be set up in the body element in the site style sheet. The default browser base font is 16 pixels. Most developers see this as too large. Designers use the body element in the CSS to set the initial size of the text.

Using CSS, take the default browser text size of 16pt and work with this to control the size of typography. First the body tag is used to reduce the default size to 62.5% of 16px , 10 px

```
body {font-size:87.25%} /*This sets the base font to 14px (14 x .8725)*/
```

This results in a default size of 10pt, which makes creating new rules and managing CSS rules easy. Notice in the example below how sizes are computed using the em measurement to become larger or smaller than the base font size. An em is a sliding measurement of the width of a font.

```
p {font-size:1em} /* This keeps the font at 10pt */
```

```
H1 {font-size:2em} /* displayed at 24px */
```

```
H2 {font-size:1.5em} /* displayed at 18px */
```

```
H3 {font-size:1.25em} /* displayed at 15px */
```

```
H4 {font-size:1em} /* displayed at 12px */
```

```
.small {font-size: 0.8em} /* This decreases the font to 8pt */
```

When using CSS it is recommended that font size be set using either the ems or percentage.

Tabbing Order

It is important to design a logical tabbing order for all windows or fields that are capable of obtaining focus at any given time. Automatically place the cursor in the first data entry field and provide auto-tabbing functionality.

Within a page, users should be able to navigate back and forth between the page elements using the TAB key and the SHIFT + TAB without mouse intervention.

The tabbing order should flow left to right, from top to bottom. There may be a few places where the tabbing order will need to move in a different manner based on a specific functional need. These situations should be approved with business analysis and/or acceptance staff when they arise.

Link Treatment

New Browser Windows

A new browser window should be opened for:

- Help links
- Supporting Information links
- Links to non-web documents (e.g., PDF, Word, Excel, PowerPoint, etc.)
- Whenever the look-and-feel of the navigation changes. (Destinations having the same header and navigation menus should open in the same window.)

Link Clickability Cues

- Text that is not navigable must not be blue or underlined. This ensures that items that are not clickable or tapped do not have characteristics that suggest they are.
- Provide sufficient cues to clearly indicate that an item is navigable using underline or reverse color when in focus.
- Make sure banners are not shaded to look like buttons.
- Be consistent in using underlining, bullets, arrows, and other symbols to indicate navigation.

Text Links

- Use text links rather than image links.
- Should be readable by text-only browsers, mobile devices, and assistive technology software such as a JAWS reader.
- More easily recognized as navigable.
- Usually download faster.
- Preferred by users.
- More readily identified as links by users.

Link Names

- Make link text consistent with the title or headings on the destination (i.e., target) page.
- Closely matched links and destination targets help provide the necessary feedback to users that they have reached the intended page.
- Do not confuse users by using the same or similar text for links with different destinations.

Embedded Links

- Ensure that embedded links (in the code) are descriptive.
- Link text should accurately and succinctly describe the link's destination. Example: "[Read More About CoPA Application Standards](#)"

Link Length

- Make text links long enough to be understood, but short enough to minimize wrapping. Links should not wrap to more than one line.
- Single-word links may lack needed description but additional words increase reading time. Create text links comprised of short words or phrases rather than entire sentences.

Accessibility

- Users should be able to tab to navigate between links.
- Users should be able to activate a link when pressing 'Enter' on their keyboard.
- Users should be able to identify links without relying on color alone.
- Users should be able to activate hover and focus states with both a mouse and a keyboard.
- When labels are used to call out new content that is dynamically loaded onto a page, be sure to use ARIA live regions to alert screen readers of the change. The aria-live attribute has three possible values:
 - Off (no notification).
 - Polite (screen reader notifies user once current task is complete).
 - Assertive (screen reader interrupts current task to notify user).
- Be sure the clickable or tappable text never just says "click here." This text (whatever is contained in the "<A href..." tag) is read by assistive technology software. Hearing only "click here" frustrates vision impaired users by giving them no clue about where the link goes or why it should be clicked.
- Inform users via tool tips or link attributes whenever a link target opens in a new window.
- There should be no links that leave the Commonwealth's Web domain without a warning to the user that will enable the user to proceed or return based on their choice.

Button Treatment

Buttons are the primary way for actions to be taken in the application. The label of a button must indicate the action that will be applied when clicked. It is important that the labels and the intended actions remain consistent across applications. Common buttons include those in the following table.




Label	Functionality
Accept	Indicates the user's choice to use a presented data value, which may also entail selecting a data value from among multiple options. For example, choosing to use an address as entered or as received from an address validation service.
Add	Similar to Create. Produces a new form or record for a particular entity, usually allowing the user to enter information in the new instance. Implies that the new information will be appended to existing records of the same type.
Approve	Indicated confirmation, endorsement, and/or support of the information being displayed.
Back	Takes the user to the previous page or screen.
Calculate	Starts the process of calculating information in form fields and giving a result.
Cancel	Permits users to leave the form or dialog without making changes or taking actions. Navigation required, returns to the original location from which the form or dialog was called.
Checkout	Starts the process of accepting payment for products or services.
Clear	Removes all values displayed on the form, returning fields to empty, null or default values. No navigation required.
Continue	Navigation required, proceeds to the next step in a process. Implies at least temporary acceptance/ submission of data entered in the current form or screen.
Create	Produces a new form or record for a particular entity, usually allowing the user to enter information in the new instance.
Details	Drills down to display or allow editing of more specific attributes or components of a higher-level entity.
Edit	Opens an existing form or field in a mode that allows data to be added or altered.
Email	Opens the default email client or sends a message to a pre-defined recipient.
Filter	Subsets information based on field input.
Generate	Starts the process of generating whatever was selected on a page.
Home	Returns the user to the main, welcome or parent screen of the site or application.
Login	Logs the user into the application.
Next	Takes the user to the next step in a set of screens.
Previous	Takes the user to the previous step in a set of screens.
Print	Allows the user to print a page. Should open a printer-friendly version of the current page.
(Print) Preview	Provides users with a capsule view of their changes or actions, often in what-you-see-is-what-you-get, or WYSIWIG, format, prior to printing or submitting.
Reset	Resets a form to the way it was when the page was loaded.
Save	Should only be used when the user can save data and does not trigger anything else to happen. Save should be used when a user can come back to a transaction to update it. Stores all (changed) field values to the database for later retrieval. No navigation required, stays on current screen.
Search	Launches a query to find and return records based on the criteria entered by the user.
Submit	Navigation required. Should be used when triggering other actions to take place, and when the transaction should be put in a state that is not updateable by the user entering the data. Commits form data for processing, approval, or storage. Passes data from a Web page to a back-end process or server.
Update	Starts the process of updating the information entered on a page.
View	Displays additional information about something on a page.

Icons

In the battle of clarity between icons and labels, labels almost always win. Icons can be hard to memorize, are often surprisingly ambiguous, and can add little to user comprehension. As a general rule, use only a button label, with no icon, to avoid confusing the user and reduce clutter. Exceptions to this rule include:

- Icons can be used effectively *in combination with text labels* in a few cases when they are likely to increase rapid comprehension. These include:
 - Universally understood icons (e.g., Print, Search, Help)
 - For differentiation among a set of buttons (e.g. upload photo, video, text file, or spreadsheet)
- Icons can be used *without text labels* to conserve space in these specific cases:
 - Universally understood icons (e.g., Print, Search, Help)
 - Icons serving as bullet points (e.g., file type icons for PDFs, DOCs).

Sample Icon Only Buttons

Transaction Date/Time	Transaction Type	User ID	User Name	Previous Placard No.	WID
02/12/2016 12:00 AM	Placard Application is Recalled	Test	Test User	MRN3969	160434900000012001 
02/12/2016 12:00 AM	Placard was renewed	Test	Test User	MRN3969	160434900000012001 
02/12/2016 12:00 AM	New Placard issued	Test	Test User		160434900000012001 

Accessibility

- Buttons should display a visible focus state when users tab to them.
- Avoid using <div> or tags to create buttons. Screen readers do not automatically know either is an usable button.
- When styling links to look like buttons, remember that screen readers handle links slightly differently than they do buttons. Pressing the Space key triggers a button, but pressing the Enter key triggers a link.
- If the button has a graphic label only, assign its Name property to an appropriate text label. This enables assistive technology like screen readers to provide users with alternative information about the graphic.

Wait Time

If users may experience wait time for a particular action to complete, provide a visual indication of activity. The following table provides feedback method guidelines for various wait times.

Wait Time	Use
Less than 5 seconds	Hourglass or similar to indicate activity
5 to 60 seconds	Progress bar to show status relative to completion
Over 1 minute	<ul style="list-style-type: none"> • Warning to user of expected duration (with opt-out choice) • Progress indicator • Audible beep on completion if available (to allow alternate activities while waiting)

Sample Hourglass Equivalent and Progress Bar from PDJF UI Component Library



Tables

Tables show tabular data in columns and rows

- Columns or rows must have headings, and visually distinct from the table body.
- Heading information should be retained when users scroll data tables.
- Responsive tables should be built with the correct DOM elements to allow scaling and displaying labels for values on the device.
- Titles of sortable columns must provide visual hints that they are clickable and whether they are currently sorted in ascending or descending sequence.

Accessibility

- All data tables should have a caption, a summary, and properly scoped header cells to allow screen readers to render them intelligently.
- Simple tables can have two levels of headers. Each header cell should have `scope='col'` or `scope='row'`.
- Complex tables are tables with more than two levels of headers. Each header should be given a unique id and each data cell should have a `headers` attribute with each related header cell's id listed.

Accordions

Accordions are a list of headers that can be clicked to hide or reveal additional content. They are useful when:

- Users only need a few specific pieces of content within a page.
- Information needs to be displayed in a small space.

Sample Accordions

Placard 1							
Placard #	Placard Type	Tag Type	Issue Date	Expiration Date	Status	WID	Actions
MRN3969	Permanent	G1	02/12/2016	2/2021	Recalled	160434900000012001	
Placard #	Placard Type	Tag Type	Issue Date	Expiration Date	Status	WID	Actions
JNTEM12	Temporary	JN	02/12/2016	5/2016	Issued	160434900000012001	

Accessibility

- Code header areas in the accordion as `<buttons>` so that they are usable with both screen readers and the keyboard.
- Buttons should state whether they are expanded or not with the appropriate attribute: use either `aria-expanded='true'` or `aria-expanded='false'`.
- Each button has a unique name `aria-controls='collapsible-#'` that associates the control to the appropriate region by referencing the controlled elements id.
- Each content area has an `aria-hidden` attribute set to either true or false. When false, the element (and all children) are neither visible or perceivable, and assistive technologies will skip this content.

Form Controls

Form controls allow users to enter information into a page.

Text Input

Text inputs allow people to enter any combination of letters, numbers, or symbols of their choosing (unless otherwise restricted). Text input boxes can span single or multiple lines.

Labels

- Labels should include adequate space between the label and input field so users are quickly able to recognize the label as describing the correct field.
- Labels must have a colon after the label so users can visually distinguish between the label and the input field or displayed data.
- Capitalize first characters (title case). Example: Control Section:
- Required-entry fields must have a red asterisk with a space after the asterisk before the field label. Example: * First Name:
- Some fields may be required only if another condition is met. If this is the case, the field determining whether the condition is true or false must be required and must precede the conditionally required field on the page. The red “required” asterisk of the conditionally required field is turned on and off dynamically as appropriate.
- A key of “* = Required” in all bolded red text is located at the top of the form as well as at the bottom if scrolling is necessary for completion.
- Place data formatting examples in parentheses, following the label and before the colon. Use black text, rather than gray, to maintain color contrast. Example: Expiration Date (MM/YYYY):

Input Boxes

- Match input box sizes with dimensions of the requested data.
- A user should not have to scroll to the right or left to see data that is being displayed inside a particular field.
- Alphanumeric fields should be left justified.
- Numeric fields should be right justified.
- Right-justify displayed currency fields.
- Format displayed currency fields as \$_,____.____ with “\$” as part of the label.
- Currency should be input as one field long enough to cover the maximum number length plus the decimal point.
- After inputting, currency should be re-displayed as described above (\$_,____.____).
- Calendar dates should be displayed as mm/dd/yyyy.
- Calendar dates should be input as "mm/dd/yyyy" in one field (or with a calendar widget at the user's option).
- In validating email address fields, check for an @ symbol and verify there are 2-4 characters after the “.” (dot).
- Require users to enter an email address twice and compare entries to validate.
- For free-form text fields (normally used for ‘Comments’) character limit must be shown at the bottom right of the field, in addition to one or both of the following:
 - Display the number of characters the user has typed Example: 250/3000 max.
 - Stop the user from typing when they have reached the maximum characters.

Accessibility

- Providing formatting examples outside labels, such as before or below the input box, allows more flexible positioning but should not be done. The user can sometimes miss it. It is also not supported by some web browsers (typically older versions) and assistive technologies that do not implement the WAI-ARIA aria-labelledby attribute.⁴
- Placeholder text should also be avoided. It provides an example of the required data format inside form fields that have not yet been edited by the user. Placeholder text is usually displayed with lower color contrast than text provided by users, and it disappears from form fields when users start entering text. Most browsers' default rendering of placeholder text does not provide a high enough contrast ratio, and having it disappear can make it more difficult for users to check their responses prior to submitting. Also, assistive technologies, such as screen readers, do not treat placeholder text as labels, and placeholder text is not broadly supported across assistive technologies and is not displayed in older web browsers.⁵
- Avoid breaking numbers with distinct sections (such as phone numbers, Social Security Numbers, or credit card numbers) into separate input fields. For example, use one input for phone number, not three (one for area code, one for local code, and one for number). Each field needs to be labeled for a screen reader and the labels for fields broken into segments are often not meaningful.

Dropdowns

A dropdown allows users to select one option from a list.

- Do not limit the number of viewable list box options.
- Ensure drop-down values are displayed entirely without being cut off.
- Avoid overusing making options in one dropdown menu change based on the input to another. Users often don't understand how selecting an item in one impacts another.
- Do not use JavaScript to automatically submit the form (or do anything else) when an option is selected. Offer a "submit" button at the end of the form instead. Users often change their choices multiple times. Auto-submission is also less accessible.

Accessibility

- Make sure your dropdown has a label. Don't replace it with the default menu option (for example, removing the "State" label and just having the dropdown read "Select a state" by default).
- Don't use JavaScript to automatically submit the form (or do anything else) when an option is selected. Auto-submission disrupts screen readers because they select each option as they read them.

Checkboxes

Checkboxes allow users to select one or more options from a visible list. Useful when users need to see all the available options at a glance or for when a user needs to choose "yes" or "no" on only one option, e.g., to toggle a setting on or off.

- Users should be able to tap on or click on either the text label or the checkbox to select or deselect an option.
- List options vertically; horizontal listings can make it difficult to tell which label pertains to which checkbox.

⁴ <http://www.w3.org/WAI/tutorials/forms/instructions/>

- Avoid using negative language in labels as they can be counterintuitive. For example, “I want to receive a promotional email” instead of “I don’t want to receive promotional email.”

Accessibility

- Surround a related set of checkboxes with a <fieldset>. The <legend> provides context for the grouping. Do not use fieldset and legend for a single check.
- Each input should have a semantic id attribute, and its corresponding label should have the same value in it’s for attribute.
- The title attribute can replace <label>.

Radio Buttons

Radio buttons allow users to see all available choices at once and select exactly one option.

- Users should be able to tap on or click on either the text "label" or the radio button to select or deselect an option.
- List options vertically; horizontal listings can make it difficult to tell which label pertains to which radio button.
- Use caution if you decide to set a default value. Setting a default value can discourage users from making conscious decisions, seem pushy, or alienate users who don’t fit into your assumptions. If you are unsure, leave nothing selected by default.

Accessibility

- Group related radio buttons together with <fieldset> and describe the group with <legend>.
- Each radio button should have a <label>. Associate the two by matching the <label>’s for attribute to the <input>’s id attribute.
- The title attribute can replace <label>.

Alerts (Messages)

Alerts act as a notification that keeps users informed of the status of the system and which may or may not require them to respond, for example that they just did something that needs to be corrected or as confirmation that a task was completed successfully.

Consistency in assigning severity level to alerts is an important usability consideration. Some guidelines are:

- **Information:** Should be used when conveying general information that is not urgent (i.e., the user can still get along fine using the application if they don’t bother to read the message).
- **Warning:** Should be used in the scenario where an action may have irreversible consequences if the user chooses to proceed (e.g, “if you proceed with format you will lose all data on the disk, do you wish to continue?”).
- **Error:** Should be used when an error has occurred and the user will not be allowed to continue until it is corrected (e.g., a mandatory value was not supplied).
- **Fatal:** Should be used when an application or system failure has occurred.

Sample Message Icons from PDJF UI Component Library



Content

- Use consistent icons in front of messages of the same severity, e.g., “i” in blue circle, exclamation point in yellow triangle. (Exact images may appear somewhat different across UI component libraries.)
- Error messages should be standardized within the application and be table-driven.
- Keep error messages short but include the specific field and the data issue that need correcting (except for the Sign In and Password error messages).
- Phrase messages in users’ terms. Do not use words or symbols such as ‘null’, ‘>’ or ‘<’.
- Adopt neutral wording for error messages; do not imply blame to the user, or personalize the computer, or attempt to make a message humorous.
- Messages containing programming codes should be meaningful and restricted to (code execution) errors that will be referred to support staff.

Placement

- In general, place messages at the top of the form page itself. (On long forms, include in-line validation in addition to any error messages that appear at the top of the form.)
- Mark the location of the first detected error by positioning the cursor at that point on the display, i.e., at that data field.
- Ensure that a displayed error message is removed after the error has been corrected; do not continue to display a message that is no longer applicable.
- If an action will result in destroying a user’s work (for example, deleting an application) use a more intrusive pattern, such as a confirmation modal dialogue, to allow the user to confirm that this is what they want.

Accessibility

- Color changes can highlight errors, but should not be the only indication (per ADA rules).
- Set “Focus” on error message so assistive technology devices such as JAWS will read correctly.
- Use the ARIA role=“alert” to inform assistive technologies of a time-sensitive and important message that is not interactive. If the message is interactive, use the alertdialogue role instead.
- Do not visually hide alert messages on the page and then make them visible when they are needed. Users of older assistive technologies may still be able to perceive the alert messages even if they are not currently applicable.

Accessibility

The Commonwealth has instituted a policy, described in ITP ACC-001, to ensure that when an agency, board or commission provides information through the web, it is taking reasonable measures to ensure that persons with disabilities can access, navigate, and otherwise obtain the same or equivalent information as those persons without disabilities. Specific web site requirements and validation techniques are added in OPD-ACC001A.

Various Disabilities to Consider

In order to build an accessible web site, you need to include everyone regardless of their disability. Below is a list of different types of disabilities that some web users have and some of the tools they use to view the web with:

Visual

- Blindness
- Assistive technologies that “read” text elements in a page
- Use Keyboard to navigate
- Use Tab key to get from link to link
- Low Vision
- Use software to enlarge the screen
- Color Blindness
- Approximately 10% of men and 0.5% of women have some color blindness

Hearing

- Hard of Hearing/Deaf
- Cannot hear audio content
- Don’t assume they know sign language

Mobility / Motor Impairments

- May use assistive technologies like:
- Raised spaces in between keys
- Puff and Sip switch
- Head switch
- Head wand
- Voice recognition
- Iris recognition
- Mouth stick and
- Trackball

Cognitive

- Confused by complex visual layouts
- Difficulty understanding lengthy text
- Problems that affect ability to process visual information
- Seizure Disorder
- Stimulated by quick movements such as animations, causing seizures

Accessibility consists of a system that can be accessed and operated in a variety of ways with a variety of tools and does not rely on a single method or ability of the user. This promotes maximum user community participation, and allows the web site to appeal to the general public as well as internal employees.

In practice, building an accessible web site includes:

- Structuring content using code that indicates information types, hierarchies and relationships.
- Using semantic markup that enables a screen reader and other assistive technologies to effectively interpret a web page's meaning and utilize its content and controls.
- Creating a user interface that is device-neutral and supports keyboard alternatives to mouse-based interaction.
- Designing a Web resource that functions with diverse user agents and operating systems, and degrades gracefully when it does not.

Accessibility and RWD

Both responsive Web design and accessibility are ways of making a site flexible. Responsive Web design starts by thinking about a page as a collection of elements that can be rearranged, not as a static layout. Accessibility starts by thinking about different ways someone might interact with a web site/application and ensures that, no matter what senses a user employs, the site supports all of them.

A responsive site—that is, a site that is designed and coded to respond to devices with different screen sizes—is not automatically accessible. Following are considerations for RWD and accessibility:⁵

On Desktop Browsers

- Pages must respond correctly to custom CSS and removal of CSS. Enabling high contrast mode in browsers such as Internet Explorer will turn off background images to ensure the proper contrast level of text.
- The page must still be understandable without CSS Background-images.
- The page must still be readable without specified colors.
- When a background color is specified, a foreground color must also be specified, and vice versa.
- Position and reading order of content must remain correct.
- Using CSS to hide content in different contexts may not always provide the desired results with assistive technology. Manipulation of the DOM to add or remove page content should be used (e.g appendChild).

On Mobile Devices

- The viewport must allow users to pinch zoom (scale) up to 200%.
- The minimum level of contrast between foreground and background colors may need to be greater when displayed on a mobile device. For example, the luminosity of standard text must be a ratio of 4.5:1 for WCAG Level AA conformance. On a mobile device the contrast needs will likely be more, for example a luminosity ratio of 7:1 indicated by WCAG 2 Level AAA criteria.
- It is a common misconception that focus order (AKA tab order) is not important on touch screen devices. Focus order is still important to users of assistive technology and alternative input devices. For example, the swipe gesture with a mobile screen reader will move to the next content unit such as a link, form field,

⁵ "What Does Responsive Web Design Have to do with Accessibility?" by Jonathan Avila, July 11, 2013

or text based on the DOM order. Users with mobility impairments may likely use an external keyboard or device that simulates a keyboard to access a tablet.

ARIA

Accessible Rich Internet Applications defines ways to make Web content and Web applications (especially those developed with Ajax and JavaScript) more accessible to people with disabilities. ARIA landmark roles identify significant page areas, giving them meaning and making them more keyboard navigable. There are several landmark roles available, but simply adding the following five, can greatly enhance the accessibility of your page:

- Main
- Navigation
- Search (If applicable)
- Banner
- Contentinfo (Footer)

To include the landmark roles, add a “Span” or “Div” tag and a “Role” attribute. For example: `<div role="Banner"> banner html code here </div>`

For additional information, see: <http://webaim.org/techniques/aria/>

Specific accessibility guidance is provided in this document as it applies to various user interface elements.