Corporate UI Standards

Release Version 1.0.1

Disclaimer: In case of a conflict between the guidelines in this document and functional specs /mockups; the functional specs/mockups will dictate the UI.

Contents

1   Revision History

Revision #	Action	Date	Author
Ver 1.0.0	Document Created	5/20/2010	Farid Kajani
Ver 1.0.1	Document updated	6/10/2010	Farid Kajani

2   Layout

The Layout will have 4 main sections and 2 optional sections depending on the need.
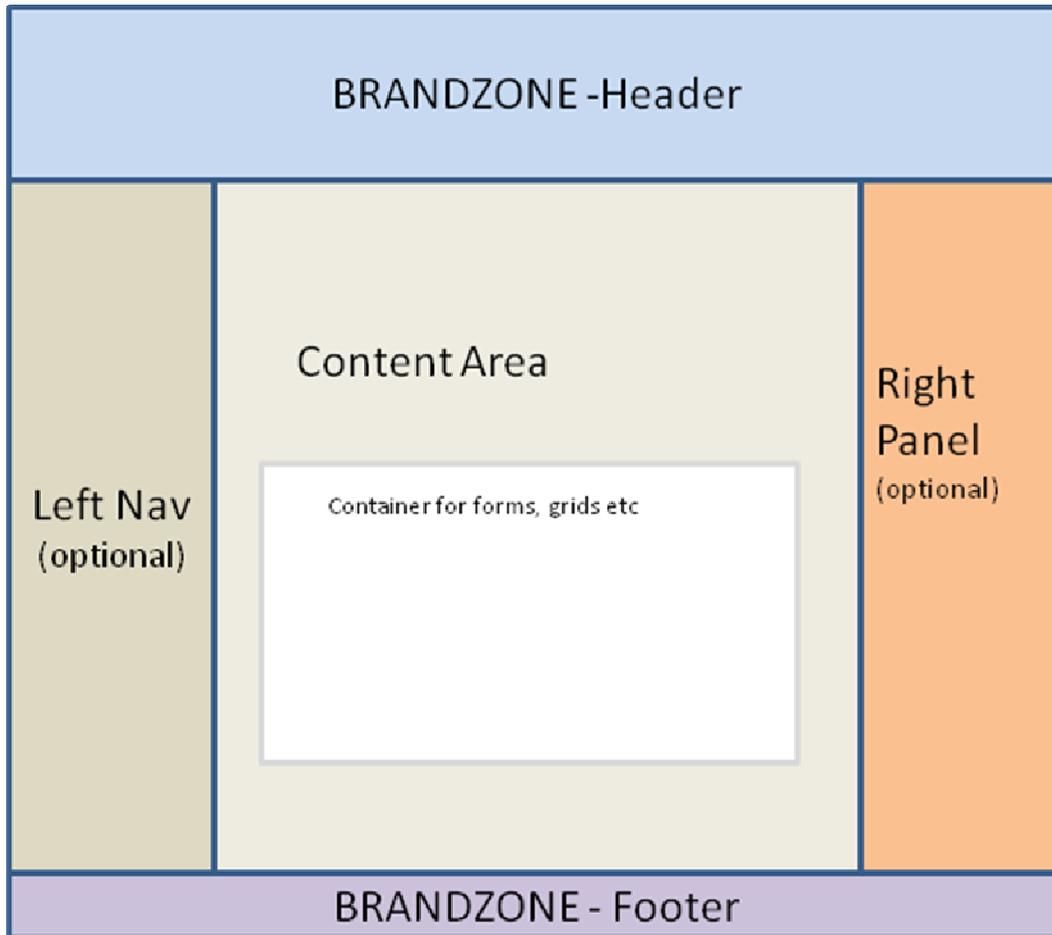
The layout should be designed for **1024px** width screen resolution. The height should be a **minimum** of **618px** (This is to accommodate the browser menu, toolbars, tabs etc). If the application is designed to cover the full window i.e. eliminating the menu, toolbar and tabs, then the minimum height should be 768px. The browser can have a vertical scrollbar but horizontal scrollbar should be avoided as much as possible. The scrollbars should be present only if there is useful content beyond the viewable area.

The brandzone should not extend beyond 1024px. In scenarios where the content area extends beyond 1024px width the **brandzone** should span the maximum available width.

2.1 Brandzone

- The Brandzone is the area which comprises of the Header (with Product branding, company branding, Navigation - Menus & Sub menus), and Footer for Web Applications. Other related Graphics and contents are also part of the Brandzone.

- The detailed information for Brandzone and its Guidelines are available at http://nrs12vdbuild03/GUI/Clf/Brandzone

- The Brandzone is a standard component that is already designed and ready for implementation. All products are expected to use it directly from the site or replicate it depending on the limitations of the product.

- The tab Name should always say 'MedAssets - <Product name>'
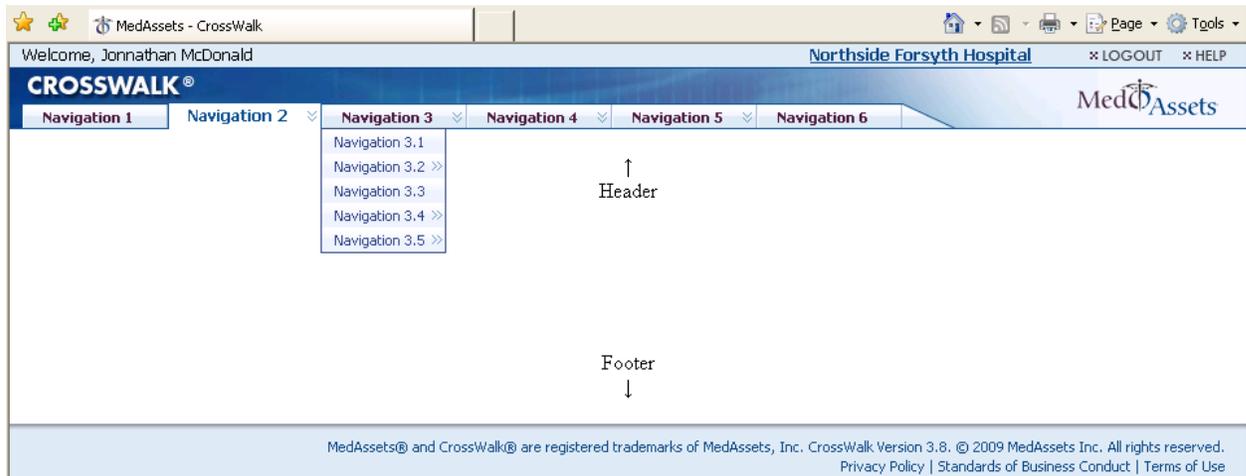
2.2 Content Area

- Page Title

- Sub title (optional)

- Page Trail (optional)

- Main Content
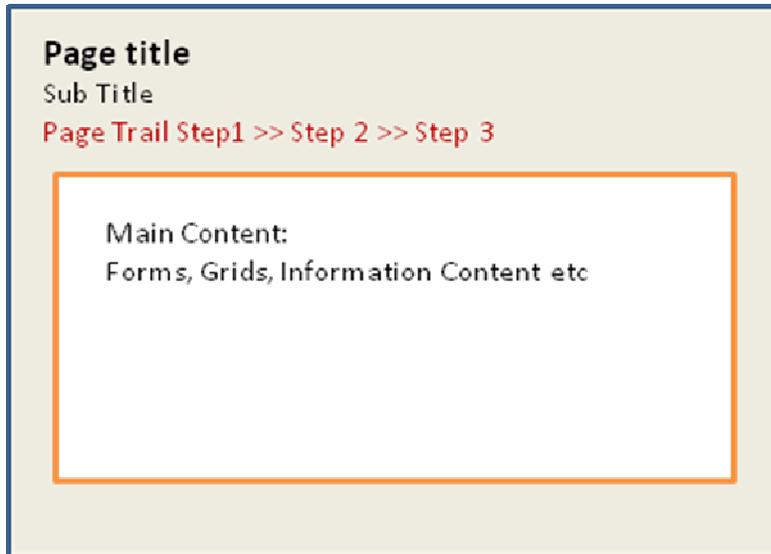
2.3 Left navigation (optional)

The left navigation can have either or both; THE LEFT NAV LINKS AND/OR TREE depending on the requirement.

- There should be only 2 levels of links and sub links

- No limit to the number of nodes the Tree can have. However the tree should be enclosed in its own enclosure and have scrollbars if needed (set to Auto).
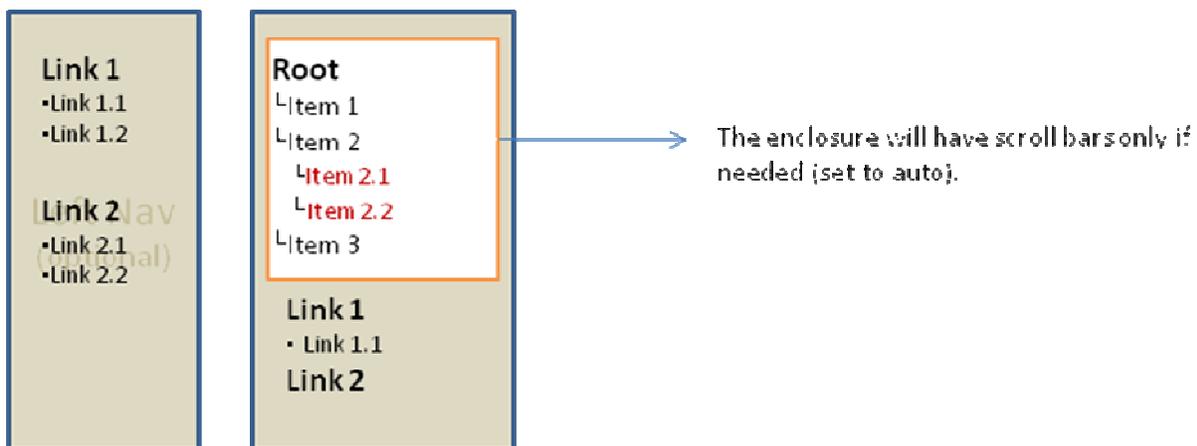
2.3.1 Accordion Control

The left nav can be done in the form of an Accordion control also. However the necessity of it will be determined by the UI Designer.

- All blocks in the accordion layout should look the same

- Only one section will be open at a time. If the other section is activated the previous one will collapse.

- The order in which the headings appear will always remain the same no matter which section is open.

- All headings should be visible on the screen height of 618px. If a section is tall enough to push all the other headings beyond the screen height then that section should have a vertical scrollbar.
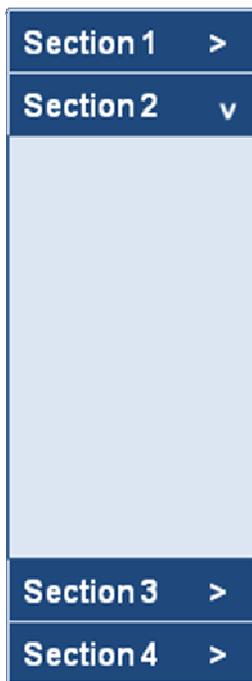
- All headings should have a directional icon (arrow) to indicate the state of the section (open or close).

- The state of the section can be changed by clicking the heading or the direction arrows

3   Colors

The color scheme should consist of the following colors:

3.1   Primary or Main

- Dark Blue (#165790) – This color should be used for Page Title, for background on Grid Headers and any other box that requires a header with a dark background. Any text on such background should be White. This color should also be used for instructional messages that are to appear on the Page itself and not as alert boxes.

- Light Blue (# dfe9f7) – This color should be used for alternating row color for Grids and background for any message or box that needs to be differentiated from the rest of the content.

- Black (#000000) – This will be primarily used for text.

- White (#FFFFFF) – This will be the background of the application and also the alternating row color for grid.

3.2 Secondary colors

- Maroon (#501d47) – This color should be used on hyperlinks only.
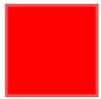
- Red (#ff0000) – This color will be used for all error messages that appear on the page.

- Green (#2a7000) – This color should be used for any positive feedback to the user. E.g. 'The Data was saved successfully'

4 Fonts

4.1 Font Family

The primary font family should be Tahoma by default; the secondary font should be verdana. If there is a need for fonts other than Tahoma or Verdana it will be defined by the UI designer. The font rule does not hold true for any graphical or creative elements.

4.2 Font-Size

Body Content: 11px, normal
Page Title: 14px bold
Grid or content titles: 12px Bold
Error messages: 10px
Instructional Messages: 10px
Success Messages: 10px
other messages: 10px

Text Colors:

- Black (#000000). Default color for all text unless specified.

- Maroon (#501d47) – This color should be used on hyperlinks only.

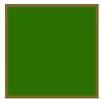- Red (#ff0000) – This color will be used for all error messages that appear on the page.

- Green (#2a7000) – This color should be used for any positive feedback to the user. E.g. 'The Data was saved successfully'

5   Areas and Elements

5.1   Page title, sub title

The Page title and sub title length should be a maximum of half the viewable size of the content area without scrolling.
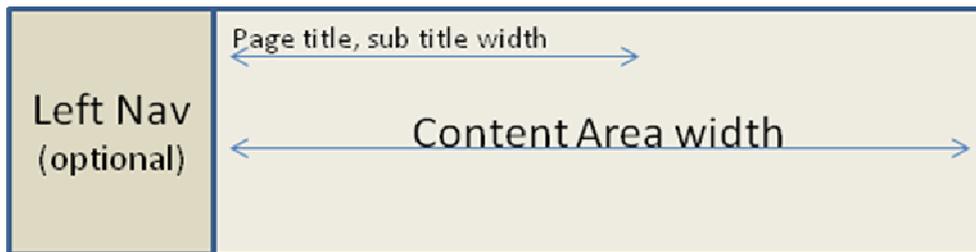
5.2   Content Area

There is no fixed width defined for a Content Area (the dotted line is shown just for reference and is not a part of the layout). However it should adjust itself or have definite width depending on the content.

The Content Area will typically hold the Messages, Title, Icons, Form or Grid or any other content, Buttons and any footer information necessary.
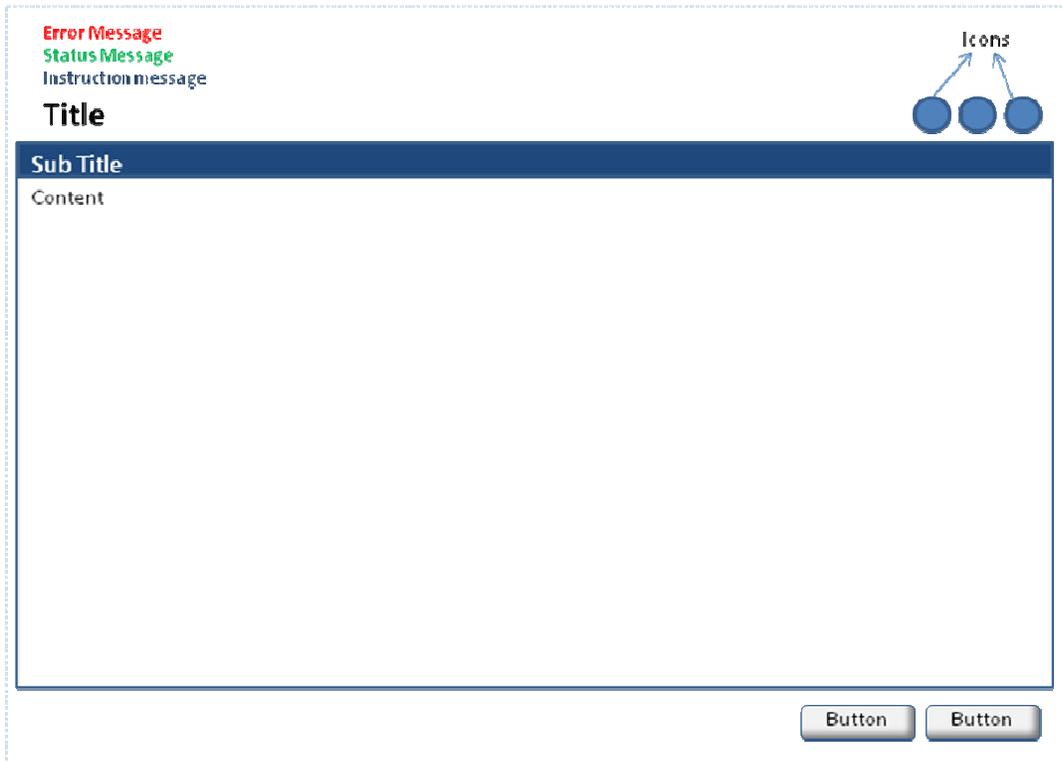
5.3 Hyperlinks

Hyperlinks will always have an underline. The color for hyperlinks will be Maroon (#501d47). By default the Hyperlinks will not be bold unless it is required to be prominent.

6 Error Handling

- Try to avoid the use of word 'Error'.

- Use a subtle language.

- Use consistent behavior to show error messages.

- Try to have Error messages displayed on the page of action itself and not as separate popup, alerts or floating divs.

- Have common library for error messages, like a resource file that is updated by a single person.

- If an error occurs a Generic Error message should be placed on the top of the content on the page like 'All required fields were not filled in'. (For visual guide to placement see section 5.2 Content Area). The fields where error occurs should have red border surrounding it and message in red explaining the error for each field if it needs an explanation.

- The cursor should be in the first field that needs to be corrected.

- Color for error message should be Red (#ff0000)

- Font size for error message should be 10px.

6.1 Server side errors

- Display multiple error messages at the same time.

- Minimize server trips

- Allow users to correct several errors at once.

6.2 Client side errors

Each field that needs a validation should be validated on exit from the field. i.e when the cursor moves out of the field and the required condition for the field is not satisfied it should display appropriate message next to the field in red.

6.3 Messages and Feedback to the user

Messages and Feedback to the User can be displayed in the following ways:

Message area – This Area is usually above the main action area but below the page information like page title, page trail etc. However depending on the need and appropriateness, a page can have the message Area in somewhere else as long as it is closer to the action area. The message area should be used to display - Error messages, Informational messages.

Status Bar of a window – This area should be used for non critical information.

Too tips – Tool tips should be used on all Icons and if needed on elements that need more explanation. These should be used to supplement the main help and not be used extensively throughout the page.

Pop ups and Alert boxes - These should be used only in case of critical message or warning. They should have a non-destructive default action (pressing enter). i.e. If the message is 'Your data is not saved. Are you sure you want to Exit.' The focus should be on the 'No' button by default.

The following icons should be used based on the type of message

⚠ - Warning
⛔ - Critical
❓- Question
ℹ - Information

The Alert and message pop ups should always be modal. No other action on the application should be allowed on the application unless an action is taken on these pop ups or alerts.

7   Actions

7.1   Drag and Drop

When executing a Drag and Drop the item being moved should be either made lighter or opacity be reduced to 60%. The element should remain stuck to the mouse until the item is dropped in the target area. If the item cannot be dropped in the area, the mouse should have an addition icon on top '⊘'

7.2   Moving a window or a section

When moving a window or section with a mouse, the cursor should change to a moveable cursor ✥

7.3   Progress bars

- Short waits (less than 4 seconds) – Use hour glass

- Long waits (greater than 4 seconds) – Use Bar with progress indicator with percent completed and if possible estimated time left. Also give option to cancel.

- Unknown wait – show long the process is going on with a timer and give option to cancel and/or pause.

7.4   Expand/Collapse Actions

In cases where information needs to be hidden or collapsed there should be a graphic or sign (+) or (-) to change the state. If the content is displayed the sign will be (-) to indicate you can collapse it and vice versa.

7.5   Save

Save will be triggered when a Save button is clicked and on successful save a Save message should be displayed on the page above the area that was modified.

A Save will also be triggered if a change is made to a page and the user is navigating to the different page without saving. In that case an alert will be popped asking the user for a confirmation to save. In that case the behavior will be the same as mentioned above about displaying the message. The exception to this rule is that if the user is simply selecting another tab to go back and forth between tabs

to copy/paste, research, review, and the like while filling in data. So in short the confirmation dialogs should be thrown only when a data loss can occur by the user's action.

7.6 Delete

Any kind of delete action should be interrupted with a Delete confirmation (alert) and then the action should be performed. On a successful delete there should be a delete message displayed above the area that was modified.

7.7 Incomplete form/fields

The instructions/message for incomplete form/fields should be displayed above the form on the same page. It should not be an alert message

8 Form Elements

The forms will always have the labels left aligned and the fields left aligned in their respective columns. They should always have

- Instruction message (if any)

- Title

- Icons

- Form labels and fields

- Action Buttons
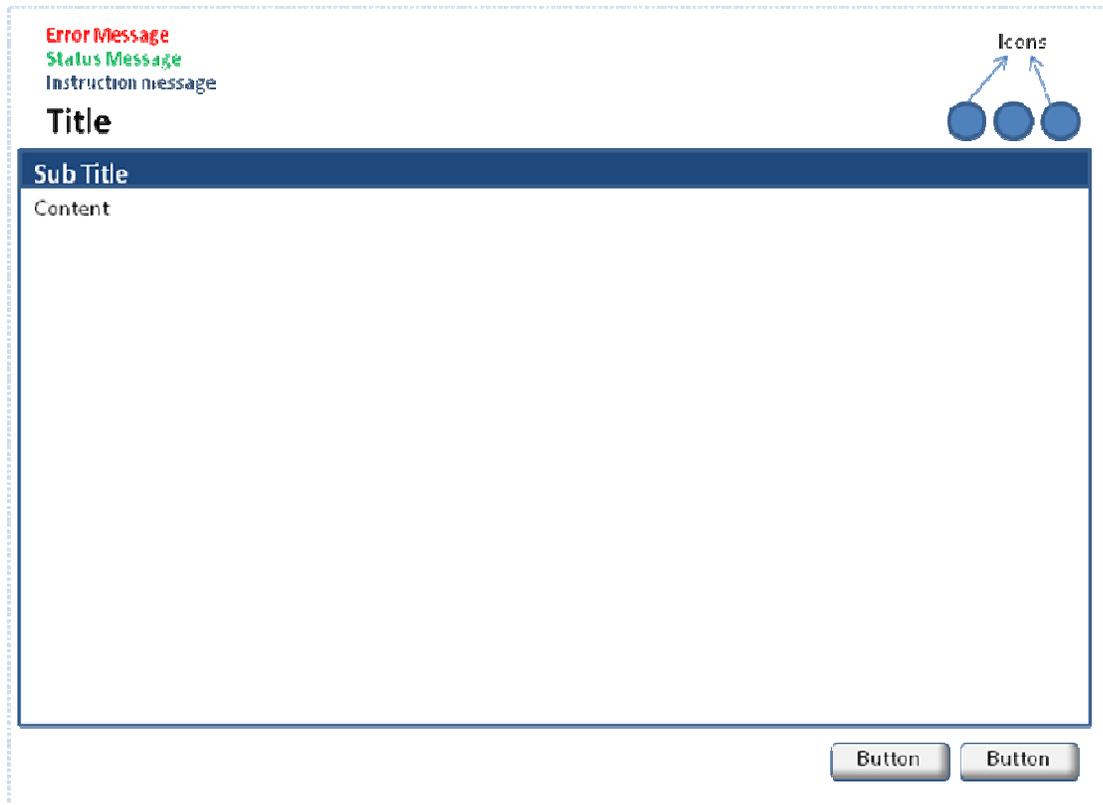
8.1.1 Radio Buttons

The maximum number of items for use with radio buttons should be 6. If there are more than 5 items then a dropdown should be used. There is a exception to this rule in scenarios where there is a list of item details for e.g. in a grid.

8.1.2 Check Boxes

Checkboxes can be used if multiple selections are necessary. However the maximum number of items should be 8. If more than 8 items are needed then the Summary list box control should be used. There are some exceptions to this rule for e.g. checkboxes on a grid. Any other exceptions will be governed by the mockup design.

Checkboxes under normal circumstances will have 2 states – checked and unchecked. However for situations where a checkbox is used to make multiple selections 3 states can be allowed to indicate the mixed status. The 3rd state should be indicated with a solid color filled checkbox. Typical examples are tree node, checkbox in a column header.

Checkboxes should always be on the left side of the label and there should be one character space between the checkbox and the label. If possible allow the user to click on the label (in addition to the checkbox) to select the respective checkbox. Do not make the labels a hyperlink to perform some other action.

8.1.3 List Boxes

There are 5 different variations of list boxes:

- The standard List box

- Summary list box

- The dropdown

- The dropdown combo box

- Progressive search list

8.1.3.1 Standard List box

A standard list box with multi-select should be used for 8-12 items. If the number of items is less than 8 then checkboxes, radio buttons or Dropdown should be used. If the list has more than 12 items and requires multiple selections then a summary list box should be used.

8.1.3.2 Summary List Box

Summary List Box should be used for more than 12 items and requires multiple selections. The list Box should always be in the form of pairs:

The available Items list box and the selected list with 4 buttons between them to transfer items between them. The list menus should always allow you to select multiple items.

8.1.3.3 Dropdowns

Dropdowns should be used for more than 5 items. If there are only 5 items or less then radio buttons should be used. However the downside to using too many dropdowns is that it requires 2 clicks to select. And all items are not visible until the 1st click. If the items of the dropdown are getting generated dynamically i.e. the application has to make server trips depending on fulfillment of some other fields then in that case a maximum of 8 items should be in the list box.

8.1.3.4 Dropdown Combo Box

This is a standard control for Windows applications but not for the web. For the web it is encouraged to use this if possible by using Ajax. However performance should not be compromised.

8.1.3.5 Progressive Search list

This is a special kind of dropdown combo list. By default it is a textbox and as the user starts typing, a list of close matches gets generated under it. E.g. Google search. The list should not start getting generated until a minimum of 3 characters are typed in.

8.1.4   Date picker

Date fields should always have a date picker and should also be allowed free form entering of dates unless specified in the functional specs. The format should be MM/DD/YYYY.
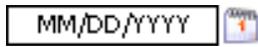
8.1.5   Tab Order

By default the cursor should be available on the first field of the form. And the tab order should run from left to right.

Hitting Enter should simulate as the form is submitted without having to tab to and click the submit button.

8.1.6   Buttons

All forms will have a submit button and forms with more than 5 fields should have a clear button. The clear button will reset the form to its original state.

All buttons should be regular form buttons and not images.

8.2   Grid

Grid should be a minimum of 400px wide. But there is no limit to the maximum width. There should be no width defined for any column unless necessary. The columns are expected to adjust among themselves. However if it does not look aesthetically right then a width should be set for a column. The column header text should never be clipped.

8.2.1   Grid Layout

The grid will have the following areas/elements.

- Title – Cannot be more than one line

- Action Icons

- Dropdown for # of items to be displayed at a time – The option will be 10, 25, 50, 100. The default will be 100 unless specified in the functional specs document or mockups.

- Sub-title or Column Headers - This is optional depending on the need;

- Paging dropdown.

- Footer(optional)

- The grid will have alternating row colors and a column divider. For color information see the section 3.1 Primary Colors

- Amount column will always be right aligned.

8.2.2 Grid Behavior

- The checkboxes on Grid should have only 2 states. Checked or Unchecked. The 3rd state (mixed) is allowed if needed.

- The checkbox column will have a checkbox in the column header to select/deselect all.

- When a User selects some checkboxes on the first page of results and then navigates to the next page to select a few more items, all selections should be recorded in the session (or something similar). The Action like save or Delete should be performed on the entire set of selections. The user should not be prompted to take an action for each page of results.

8.2.3 Rules for operations on grid items:

Grids will never have a horizontal scroll bar unless specified in the functional specs or mockups. If the grid extends beyond the viewable space then the browser can have a horizontal scroll bar.

Checkboxes or Radio buttons if needed should always be present on the first column.

Editing Items -

The value in the cells will be editable on a single click. When navigating from one page to another of the grid the changed values will be saved in the session/cache. The changes will be committed only when save is clicked or when navigating to a different section, the user should be prompted to save the changed values.

In case when editing is not allowed on the grid itself or in cases where editing only one row at a time is allowed then the 2nd column will have a icon which will take the user to edit page.

Delete -

Deleting multiple rows will be allowed by selecting the checkboxes (or radio button for single item deletion)

9   Popup Windows

Popup windows should always be modal i.e. they will appear on top of all other screens and the screens behind cannot be accessed unless the popup is closed. However there are exceptions depending on the requirements and actions to be taken.

There should be only one level of popup. If anymore levels are required then they should be available through the div fly outs.

All popup windows should be resizable.

Popup windows should not have any browser toolbars or status bar.

Pop ups can be closed by clicking the 'X' on the popup window.

Scrollbars should appear only if needed

If any changes have been made by the user and popup Close 'X' is clicked, a alert message should popup asking for User confirmation to save before closing the window.