Staff User Interface Design Principles

Draft for Community Comment, 10/6/2011

StI-1: General scope / description

This specification outlines the features and functions to be included in the staff user interface of the new application. It also describes the relationship of the staff user interface to the public user interface. Information about the public user interface can be found in the separate specification, PI.

The staff user interface will provide a method to browse, search, display, and edit records contained within an instance of the application. This document establishes an overall framework that supports the required interactions, providing a consistent user experience. Where the specific features mandated cannot be accommodated within the design principles suggested here, developers should consult with the technical architect and interface design consultants.

The staff user interface will be generated using templates and themes, two mechanisms for displaying output and making the output editable. Templates allow a repository to specify, order, and provide user controls for the individual data elements and labels comprising a particular type of view (e.g. a resource or digital object editing view). Themes allow a repository to style this output and place the rendered output in header and footer templates. Depending on the specific user permissions or record object being edited or displayed, parts of the header will vary in the specific content to be displayed. In other words, the header will not contain fixed context, but will be generated dynamically in response to the current state of certain application level or object-level variables.

Themes and templates are to be used as part of the systems-level architecture for the application. A 'default' staff UI theme will be supplied with the application. In addition, system administrators may implement their own staff UI theme using the system configuration panel.

For each record type that is viewable by the public (e.g. resources, resource components, digital objects, etc.) the staff user interface will provide a link to the public user interface's view of the data.

StI-2: Theme system: Description and Business Rules

A theme system will control the graphical appearance, display properties, and layout of the application's web-based staff interface. While templates generate the HTML output that is found in the main content section of each page (see section StI-4), a theme will

stipulate the style for that output. The header will contain a set of controls to search, display, save, and augment existing work, as well as a method to display public user views of the data.

A 'default' staff theme will be supplied with the application, but system administrators can customize the staff interface and graphical elements for local use. Each repository may have only one active theme at any given time. To allow for customization, repository staff must be able to create a new theme or modify the default theme and save it as new theme. To implement the new theme, a system administrator will change a setting within the application's system configuration panel.

Business Rules for Themes System:

1. The 'default' staff theme will be supplied as a subdirectory of the web application, in a 'staffthemes' folder.

2. System administrators must be able to create new themes as subdirectories in the 'staffthemes' folder

3. Once a new theme has been created and added to the 'staffthemes' folder, it must be available for selection in the system configuration panel.

The default theme and any additional themes developed by a repository or other implementer will provide styling information for content in the following areas:

1. **Header:** Scripts or include files to generate a header. The header must include the following elements:
    a. Logo/branding and Repository area
    b. User profile area.
2. **Main User Interface:**
    a. System-wide navigation area. Will provide tabs to local menu and editing screens for the following record types:
        i. Accessions
        ii. Resources/Resource Components
        iii. Digital Objects/Digital Object Components
        iv. Name Authority Records
        v. Subject Authority Records
        vi. System Configuration/Settings (tab visible ONLY to system administrators)
    b. Record-level user control area.
    c. Record-level navigation area. Includes:
        i. Filter sub-area to narrow the list of existing content records, allowing the user to pick one for editing

ii. A browsing area, that provides a filterable list or a tree hierarchy, depending on the context. This area will allow users to navigate, load, reorder, and transfer records found within the list of all records of a particular type or within the tree.

d. Record-level display/editing area.

3. **Footer**

Sections StI-3 to StI-5 describe the specific information to be included in each of these areas, as well as the general operation of the template system that produces the HTML output that is styled by the themes

Figure one in section StI-6 includes a non-prescriptive wireframe illustrating a potential layout scheme for the sections listed above.

StI-3: Staff Output Templates Overview

An output template system will specify the content of a particular type of page within the staff user interface, listing and ordering the XHTML elements, labels, application data, display features, and user controls that will be presented to the user. This content will be styled and placed in a header and footer by the theme system (see section StI-2).

The system will require the following templates:

4. **Header Templates**
    a. Logo/branding template
    b. User profile template
5. **Main User Interface Templates**
    a. Record-level user control templates
    b. System-level navigation templates. Will provide tabs to local menu/editing screens, for the following record types:
        i. Accessions
        ii. Resources/Resource Components
        iii. Digital Objects/Digital Object Components
        iv. Name Authority Records
        v. Subject Authority Records
        vi. System Configuration/Settings (tab visible ONLY to system administrators)

    Each of the above tabs will include submenus to create specific record and sub-record types, which will also need to be generated by the template system.
    c. Record-level navigation templates, including:

        i. Filter sub-area, allowing the user to narrow the list of existing content records, to pick one for editing
        ii. A browsing area, that provides a filterable list or a tree hierarchy, depending on the context. This area will allow users to navigate, load, reorder, and transfer records found within the list of all records of a particular type or within the tree.
    d. Record-level display/editing templates. The display/editing templates must include standard form controls such as select boxes, text fields, and the like.
    e. They may also use modal windows, or another suitable user interface element, to provide 'quick add' facilities and allow users to upload digital content.
    f. They may also use "tooltips" to show help text.
6. **Footer Template**
    a. Scripts or include files to generate a page footer, to include the following elements:
        i. Copyright statement,
        ii. "Powered by ArchivesSpace" linked to project website
        iii. Versioning information
        iv. Page generation statistics (load time, memory usage, etc.)

General Business Rules for Template System

a. All staff user interactions with the system, including the submission and display of passwords and form data, must take place using an SSL connection.

b. The template system must produce non-styled XHTML output. All styling and layout is to be done by the theme system.

c. The editing screen for each record must provide an item-specific URL that includes the system ID of each item type, and other appropriate parameters, to uniquely identify the page used to edit the record. This will allow direct links to the editing interface from the public side interface, when the staff user has authenticated.

d. We suggest that template system utilize a fluid layout, with the main user controls occupying 75% of the width of the screen, subject to a minimum and maximum width set via configuration directives (default min=600 px, maximum=1,200 px).

e. To the extent possible given browser support (see section f below), the template system must use HTML5.

f. The user interface must comply with relevant accessibility standards, including:
   a. Web Content Accessibility Guidelines (WCAG) 2.0
   b. iCITA HTML Best Practices
   c. WAI-ARIA (Web Accessibility Initiative - Accessible Rich Internet Applications)

g. The template system must be extensible, allowing plug-in developers to add or edit tabs/menu options and editing screens without requiring any modifications to the core table structure, API/object model, templates, or default stylesheet.

h. The template system must use the smallest possible number of id and class selectors. It must consistently apply those selectors to similar boxes, form controls, and other elements found within the header, menus and display/editing screens, allowing for extensibility and an easily maintained theme system.

i. All display properties, forms, and user controls must operate correctly in the latest versions of the following browsers, at the time of release: Internet Explorer, Firefox, Opera, Safari, and Chrome.

j. The template system should use open source libraries for user interaction elements and dynamic components. All open source libraries included in the application's template system must be eligible for inclusion under the terms of the ECL2 License under which the application will be released.

k. The API supporting the display/editing templates must include a function allowing developers and plug-in writers to load, view, and, in some cases, edit specific content in modal windows as illustrated in StI-6, figure two. Modal windows might be used to display digital object files, to allow users to 'quick add' records in another application module (for example, create a name record while creating a resource record, upload digital content, or complete other functions. "Tooltips" might be used to provide help text.

l. The application should provide mechanisms for internationalization and localization of the form labels, button names, help text, and other translatable elements in the interface. Common mechanisms, such as using gettext, or a feature-compatible equivalent, are recommended. Ideally, users should be

able to develop and ultimately contribute translations of interface elements for the application.

The remainder of section StI-3 provides a non-prescriptive description of the content and user interaction features to be generated by each of the above templates. Developers should collaborate with members of the technical team and consult other archival and digital object interface designs to determine optimal layout design for a user friendly and contemporary user interface

StI-3a: Header Templates: Description and Business Rules

Logo/Branding Template: Positioned in the upper left corner of the user's browser, the logo/branding template includes the ArchivesSpace logo and name, hyperlinked to the main 'dashboard' view of the staff interface. The name of the repository under which the user is currently authenticated will appear to the right of the ArchivesSpace logo, centered in the open browser window.

User Profile Template: Positioned in the upper right corner of the user's browser, the user profile template will list the current user's login name and will provide a method to edit the user's profile. This link should load a form in the display/editing area, so that the user can edit his or her profile/user record and save changes using the record-level controls.

In the case where an external authentication system (e.g. LDAP or Active Directory) has not been configured by the system administrator, the user should have the ability to change his/her password, using its built-in authentication protocol, and to resend his/her activation email or account information to the specified email address.

Business Rules for Header Templates:

1. All content contained in the header templates must appear in a fixed area of the browser window. In other words, it must be positioned relative to the top left corner of the browser's display window. It must not move around within the open browser window, except to the extent that the operator changes the horizontal width, relocates the window, or scrolls within the loaded page.

StI-3b: Main Interface Templates: Overview

The main user interface templates will generate a UI that allows users to browse, navigate, edit, and delete content records. They will also provide users the ability to manage relationships between records and to get help using the system. Finally, they will provide users logged in with Administrator credentials the ability to set system configuration options and vales (see System Configuration Specification SC, for a complete description of these settings.)

StI3-c: System-Level Navigation Area: Description and Business Rules

The system-level navigation area provides users a method to access all of the system's functions and to load particular editing screens. It consists of a set of tabs with sub-menus, running horizontally across the screen immediately below the header information, with zero or more sub-menu items, in the order listed:

- Accessions
  - Accession Records
  - Deaccession Records
- Resources
  - Resource Records
  - Resource Component Records
- Digital Objects
  - Digital Object Records
  - Digital Object Component Records
  - File Upload?
- Names
- Subjects
- Administration/Configuration
  - Repository Records
  - Location Records
  - Staff User Records
  - Configurable Menus/Lookups
  - System Configuration Directives (full list is provided in System Configuration Specification SC).

Business Rules for System Level Navigation Area

1. The top-level tabs must appear in a fixed area of the browser window relative to the vertical position of the header box. The tabs must not move around within the open browser window as the user moves within a page or between pages, except to the extent that the operator changes the horizontal width, relocates the window, or scrolls within the loaded page.

2. When a user clicks on a top-level tab or hits "Enter" while the focus is on that tab, the system must load a clickable list of all submenu items for that tab, in the main user editing area, preceded by a short message explaining the function of that particular area and the menu items.

3. The sub-level menus should appear in response to hover events on the parent tab, or the clicking of a down arrow or other control to activate the menu.

4. When a user clicks on a sub-level tab, the system should load the main browse/filter list for records of the specified type, with the 'create new' button enabled.

StI-3d: Record-Level User Controls: Description and Business Rules

Positioned immediately below system-level navigation area, the record-level user control area includes a record title sub-area and a record controls sub-area:

This area will supply the user with the following standard control set, which regulates actions that can be *taken on the record or records that are shown, as a whole*:

- Add New
- Bulk Add/Edit
- Save
- Cancel Unsaved Changes
- View Public (launches associated public page in new tab)
- Context-specific controls
- Delete (left justified)

The above controls will be visible but not functional if an operation is not available to the user, given the current state of the application. We recommend that the controls be visible but visually distinct from functional controls (e.g., by graying them out). For example, if the user does not have a record loaded for editing in the display/editing area, the 'save' button will be disabled. Similarly, if the user does not have permission to delete records, the 'delete' button will be disabled. If more than one record is highlighted in the record level navigation area, the bulk/add edit button will be enabled.

The following user controls for functions that may affect records will be displayed immediately to the right of the five standard controls, if warranted given the existing context/state of application and existing system features. Otherwise, they will not be displayed.

- Duplicate Record
- Add/edit Component Record
    - Shown only when resource or digital object record, or a corresponding component record, is loaded
- Transfer Component Record (to another resource or digital object parent)
    - Shown only when resource or digital object record is loaded or multiple records selected in browse pane
- Import record
    - Shown when any record with a data importer is loaded

- Export record
    - Shown when any record with a data exporter is loaded
- Merge record
    - Shown when a name, subject, digital object, or resource record is loaded

Note that additional user controls, for features affecting only a portion of the record (e.g. input forms, relation/linking interfaces, file upload controls, and sub-record creation systems, etc) will be available within the display/editing area.

Business Rules for Record Level User Controls:

1. The control buttons must appear in a fixed area of the browser window, i.e. use absolute positioning relative to the top of the browser window. They must not move around within the open browser window, except to the extent that the operator changes the horizontal width of the window.

2. The “Save” button’s operation must be the only method by which users can save any record changes to the database. In other words, changes to a constituent part of the record must not be saved until the user clicks the “Save” button.

3. The ability to delete records having children (e.g. resource or resource component records having children) shall be subject to the value of a configuration directive “allow complete tree deletions.” The default value of this directive shall be “no” for all users EXCEPT administrators.

4. Any operations executed using the ‘Delete’ button must be confirmed by the user, via a modal window or other suitable user interface element. The ‘Yes’ option to confirm the deletion should be non-operative for at least 2000 milliseconds after the pop-up launches, to dissuade the user from making accidental deletions. This default duration shall be editable via a configuration directive.

5. Each user control/button should be a tabbable element, in the order it appears in the list of user controls.

A wireframe/mockup, providing a non-prescriptive representation of the user controls area, is provided in the first image provided in section StI-6.

StI3-e: Record-Level Navigation Area: Description and Business Rules

Positioned immediately below the System-Level Navigation Area along the left hand edge of the usable interface, the Record-Level Navigation Area will allow users to

locate, browse, and move between all records of a given content type, for the repository role under which the user is authenticated.

For example, if a user has clicked the tab to edit resource records, the user will be presented with a filter/search box and a list of all collections available for editing, truncated by the value of the display limit configuration setting. As the user types in the filter box, records meeting the criteria will appear in a drop-down list and in the browsing sub-area. When a user clicks a particular record in either the drop-down list or the sub-area, it will load for editing in the record-level display/editing area.

In the case when a record has parents or children, (e.g. a Resource Record or Digital Object Record with components, or a component record itself) is loaded in the display/editing area, the Record Level control area will include a navigable tree hierarchy, so that users can browse, rearrange, open, and edit any record in the hierarchy, as well as create new component records in the hierarchy.

The following business rules apply to the Record-Level Navigation Area:

1. The area shall be horizontally resizable by users, ideally via a click, hold and drag action, and it should also be hideable.

The record-level user control area will include two sub-areas, a filter sub-area and a browsing sub-area.

The filter sub-area will allow the user to narrow the list of existing content records that are available for selection, allowing the user to pick one for editing. It will be located in a fixed location at the left edge of the user interface, immediately below the record-level controls area. It will consist of one or more facet limit boxes (where applicable) and a text filtering box into which the user will be able to type a search term. As the user types, it will dynamically filter the list of records that will be available for selection. The box will auto-suggest matches, using the filtering rules specified in section StI4. Using the cursor keys or mouse, an operator will select a matching record, and then load the selected record for editing in the display/editing pane.

The browsing sub-area will provide operators an alternate method to locate records for editing and will also allow them to reorder existing records within a hierarchy. It will be located in a fixed location at the left edge of the user interface immediately below the filter sub-area.

1. When a user is browsing ‘top level’ records, the browsing sub-area will provide a list of records, in columnar format. The following business rules apply specifically to the browsing sub-area*, when it is showing ‘top-level’ records* (i.e. root records having no parent):

- The area will display individual records in columnar format.
- As the user types a value in to the filter box, the records shown will dynamically update to apply the filter criteria.
- Each record will display on a single-line row.
- Users will be able to reorder, hide, or remove columns from view.
- The total number of records shown will be determined by a system configuration setting, but the user will be able to reset the setting to a different value, which will persist during the current session or until the user changes its value.

2. When the user is browsing records that have parents (such as resource component, digital object component, or nested location records), the browsing sub-area will show a navigable tree. The following business rules apply specifically to the browsing sub-area*, when it is showing 'child' records* (i.e. records having a parent):
   - Users will be able to expand and contract levels in the hierarchy, that is, to show or hide children, through the use of plus/minus boxes or right-pointing/down-pointing arrows;
   - Single-clicking a record will highlight it;
   - When items are reordered, all child records will be moved along with the 'root' record being reordered;
   - Right-clicking a record will bring up a context menu, supporting the following functions, subject to the user's permission levels: 1) edit, 2) add child, 3) add sibling, and 4) delete;
   - Double-clicking a record will load it in the display/editing pane, and the tree will adjust to show the current context;
   - Each record shall have a means to select it for bulk operations, e.g. a checkbox or other highlighting option;
   - It shall be possible for system operators to bulk move and transfer records, both within existing hierarchy and to a different root record.
   - It shall be possible for system operators to bulk move or transfer multiple records in a single operation **ONLY** if they share an immediate parent (i.e. if the selected records are siblings.
   - When a user selects one or more records, user controls to support to the following features shall be provided, immediately below the tree hierarchy, to support bulk actions: 1) delete; 2) renumber/reorder, and 3) transfer (to allow transfers both within the current tree, and to a tree having a different root record.)
   - Drag and drop features will be supported, to allow reordering of single items and multiple selected records;

A non-prescriptive wireframe illustrating some of these features is provided in section St-6.

All of the above features, with the exception of drag and drop, are currently supported in Archon version 3.21 and higher, so developers may wish to refer to that application for a non-prescriptive implementation example. Drag and drop is supported in all versions of Archivists' Toolkit.

StI3-f: Record Display and Editing Area: Description and Business Rules

The Record Display and Editing Area will include a record title sub-area and a record and a main editing sub-area. In addition, it will support the display of modal windows, "tooltips," or other user interface elements in response to specified events.

The record title sub-area will appear only when a user is editing an existing record or creating a new record.

- When a record is loaded for editing, the title sub-area will provide the user-supplied title, the user supplied identifier, and the system assigned identifier of the current record being edited.
- When a user is in the process of creating a new record but has not yet saved it, the title area will say 'Creating New XYZ Record,' where XYZ is the type of record being created, e.g. Resource, Subject Authority, Name Authority, etc.

The record display/editing sub-area will provide users the ability to create new records, edit records, manage relationships between records, create sub-records, export records, and run reports against records. It will be provided immediately to the right of the Record Level Navigation Area. It will include support for the following features, depending on the specific operation the user wishes to accomplish:

- Single record addition/editing
- Bulk Record addition/editing

Business Rules for Entirety of the Record Display/Editing Area:

- Upon initial load, the area shall cover the majority of the application's usable horizontal width. The remainder of the usable interface shall be used by the record navigation area on the left side of the interface, as specified above.
- The area shall be horizontally resizable by users in its relation to the record level navigation area, ideally via a click, hold and drag action.

- The area shall dynamically load content as needed for particular tab views or user operations. Testing must be completed to minimize load times for different types of content.

Single Record Display/Editing Mode:

When the user is in single record display/editing mode, the following types of data entry fields shall be provided, as applicable given the data type of each particular value or attribute:

  - Text boxes, with limited WYSIWYG support and tag/wrap editor.
  - Tag/wrap editors should be provided for only for elements that support mixed content, such as titles or notes.
  - Text areas, with limited WYSIWYG support and tag/wrap editor
  - Drop-down menus
  - Relation interfaces, to select items, relate them to the current content records being edited, and, when necessary, to define attributes specific to the relationship. See the business rules below, as well as section StI-5.
  - File viewer (provided in left column *for digital objects and digital object components ONLY;* to show a file name or thumbnails of associated digital content; and associated file metadata
  - Submit buttons, to launch a sub-editing or other interface features in an modal window or another suitable user interface element;
  - Modal windows, or their substitute, will allow for the following content/controls:
    - 'quick add' features (e.g. when a user is editing a resource record and needs to add a subject term, digital content item or other record to the database, before linking it to the resource record);
    - transfer interface (when moving from one root node to another root node)
    - Uploading files
    - Help text/system
    - Other functions as identified during application build phase

Business Rules for Single Record Display/Editing Mode:

- The interface shall provide the title and/or identifier of the item being edited, along with the system-assigned ID, at the top of the editing pane

- The interface shall use a set of local tabs at the top of the display/editing area, grouping information types logically.

- The display or editing pane for any record type shall provide the means to access, view, and edit all sub-records and link records that are related to the

record being viewed. For example, when the operators is viewing a resource record, s/he should be able to readily determine all the name subject and locations records linked to the resource record, as well as all of the external document, dates, rights, and other sub-records. Similarly when viewing a subject or name record (any record supporting a one-to-many relationship) one should be able to see all the records to which a name or subject record that is linked.

- The first tab shall be called "Overview" and shall include all fields required for data validation and other fields essential to the basic identification of the record. (Developers should consult with the technical lead/team or UI contractor to determine the content of the first tab and the titles/content of all subsequent tabs in each particular editing module, and must develop an API that allows for easy reordering and movement between tabs—see section StI-2.)

- When a user is editing a record and has changed a value on a particular tab, application should provide feedback in some manner that the record is unsaved. For example, the application could provide the user the option to save the record before navigating away from the record, or the application could highlight the tab containing the field that was edited.

- When a user is editing a record and has changed a value on a particular field, the application should provide feedback in some manner that the record is unsaved. For example, the application could provide the user the option to save the record before navigating away from the record, or the application could highlight the field that was edited.

Bulk Record editing mode/rapid data entry:

A spreadsheet view will be provided to allow the bulk adding of records, in rapid data entry mode. It will include the following features:

- Movable columns of resizable height and width, including column sort headers, paging, and the ability to hide/show columns;
- Ability to cut/paste records in from excel or other columnar applications
- Open source tools that can potentially support such features are described at: http://flexigrid.info/, http://www.trirand.com/blog/, and http://plugins.jquery.com/project/jqGridView

Business Rules for Spreadsheet/Bulk Add View:

1. Bulk add features will be provided for each of the following types of records:
    - Accessions
    - Resources

- Resource Components
- Digital Objects
- Digital Object Components
- Names
- Subjects
- Location Records

2. All records added via bulk add features shall be attached to a single context node, which shall be highlighted in the tree by the user prior to addition.
3. When using the spreadsheet view for Resource Component, Digital Object Component, or Location Records, users shall only be able to add/edit values at the same node in the tree (that is, siblings).
4. After records are saved in bulk add mode, the tree/hierarchy shall reload, with the parent record to which the records were added expanded, so that users can see where and what order the records were saved.

St-4: Facet Controls and Search Forms, Description, Scope and Business Rules

This section of the specification describes the methods in which users will be able to filter record sets to limit the scope of records being displayed in the navigation area. It also details the specific fields that must be queried when users enter data into a search form found in the local navigation area, listing the business rules that govern the filtering and searching behaviors. Section St1-3e provides information about how the filter and search forms' result set should be presented to the user.

Facet Dropdown Menus and Search Form: Description

The staff side application will include two types of controls: context-specific dropdown menus, which will limit the set of records being searched to those of a particular data type or Boolean value, and search forms, which will further limit the result set to records matching a particular string value or values

An application wide search form will not be presented. These forms will only be available when a user is browsing specified record types, as described above; each form will search only those records that are of the type currently being browsed.

For example, when a user is browsing name records, the user shall be provided a dropdown menu that will allow the user to limit the result records to names of a particular type (e.g. personal, corporate, of family names. Immediately below the dropdown menus will appear as search form that will allow the user to filter the list of name records to those that match a particular string value. The dropdown menu or menus and search form will be provided in the local navigation area of the staff user interface, as shown on the wireframe in Section StI-6, Figure 3.

Scope of Filter Operation

The following fields must be included in the index that is searched when a query is submitted against the filter form:

Resources/Resource Components

- Resource Identifier
- Title
- Note Sub-Records
- Linked Name Records
- Linked Subject Records

Digital Objects/Digital Object Components:

- Digital Object ID
- Title
- Note Sub-Records
- Linked Name Records
- Linked Subject Records
- File URI

Names:

- Primary/Family Name
- Variant Names (provide access to parent record via see also references)
- Description Notes
- Subordinate Names (for corporate Names)
- Fuller Form (for personal names)

Subjects:

- Subject Terms 1-4

Accessions:

- Accession Number
- Title
- Content Description
- Condition Description
- Inventory
- User Defined Notes with string or numeric content

Business Rules for Filter Form

1. Filtering operations must not be case-sensitive.

2. Filters must be executed across the aggregated content of all fields indexed for the record type being filtered.

3. The form control must allow dynamic keyword filtering, wherein the master list of terms is dynamically narrowed or expanded in response to each keystroke.

4. Frequently occurring words (e.g. articles and conjunctions) must be excluded from the filter query before it is submitted against the index.

5. The search algorithm that is used should use a ranking or relevance system, so that results most likely of interest to the user are presented at the top of the result sets. Full parameters for weighting should be developed in consultation with the technical architect, but will include at minimum the following:

    a. Precedence of hit
        i. Hits on the following fields must be granted a very high degree of weight by the ranking algorithm:
            1. Resource Identifier, Title and Description (Scope/contents)
            2. Digital Object Identifier, and Title
            3. Accession Identifier and Title
        ii. Hits on the following fields must be granted increased precedence over hits in other fields in the same record.
            1. Resource Component Title
            2. Digital Object Component Title
    b. Frequency of hits within one resource or digital object and its components. Single records with multiple hits should be weighted more heavily, in increasing proportion to the number of times a hit is found within the record.

In certain cases, users will be able to refine filtering operations to specific types of records. For example, searches against subject terms should be able to be targeted only toward one subject type, and filters against names should be able to be targeted only toward one name record type (e.g. personal names vs. corporate body names).

StI-5: Record Relation Interfaces

The user interface must provide convenient, user-friendly ways for user to relate content records of one type to record of another type. For example, a resource record may have a relationship to one or more name, subject, accession, digital object, and location records. Developers should consult the other specifications and the Data Model for detailed in formation about linking requirements regarding related records and sub-records,

In some cases, there is only one allowable predicate value, which will be 'hard coded' into the system, determining the nature of the relationship between the two records. In others cases, the relationship may be one of a particular type, with the value selected by the user from a set list of 'lookup values'. For example, name records can be related to other name records as Parentof, Childof, SuccessorTo, PredecessorTo, etc.

In the case of other record relationships, additional link attribute values will need to be recorded. For example, the link between two name records may include a 'description' field providing some additional information describing the nature of the relationship.

Therefore, the application's API must include two functions supporting 1) Simple linking operations; and 2) Complex linking operations. These functions should be abstract functions, built into the core of the application, allowing for a consistent user experience in defining relationships and extending relationship interfaced in new plug ins and modules that might be developed by the user community.

We leave the developers and interface design consultants some discretion to suggest the most efficient linking mechanisms to support these two features, subject to the business rules below. For reference, the developers may wish to refer to the existing Archon and AT implementations to view how relation linking works in those applications. In addition, two non-prescriptive screenshots from the current Archon relation interface, illustrating simple and complex relationship, are included in section StI-6.

Business Rules for Linking Systems:

1. Record linkages should not be written to the database until the user clicks the 'save' button at the top of the screen.

2. Linking mechanisms shall provide users a way to select one or more record from the entire list of potential object records, then attach it to the subject record.

3. A dynamic filtering system and autosuggest features shall be provided.

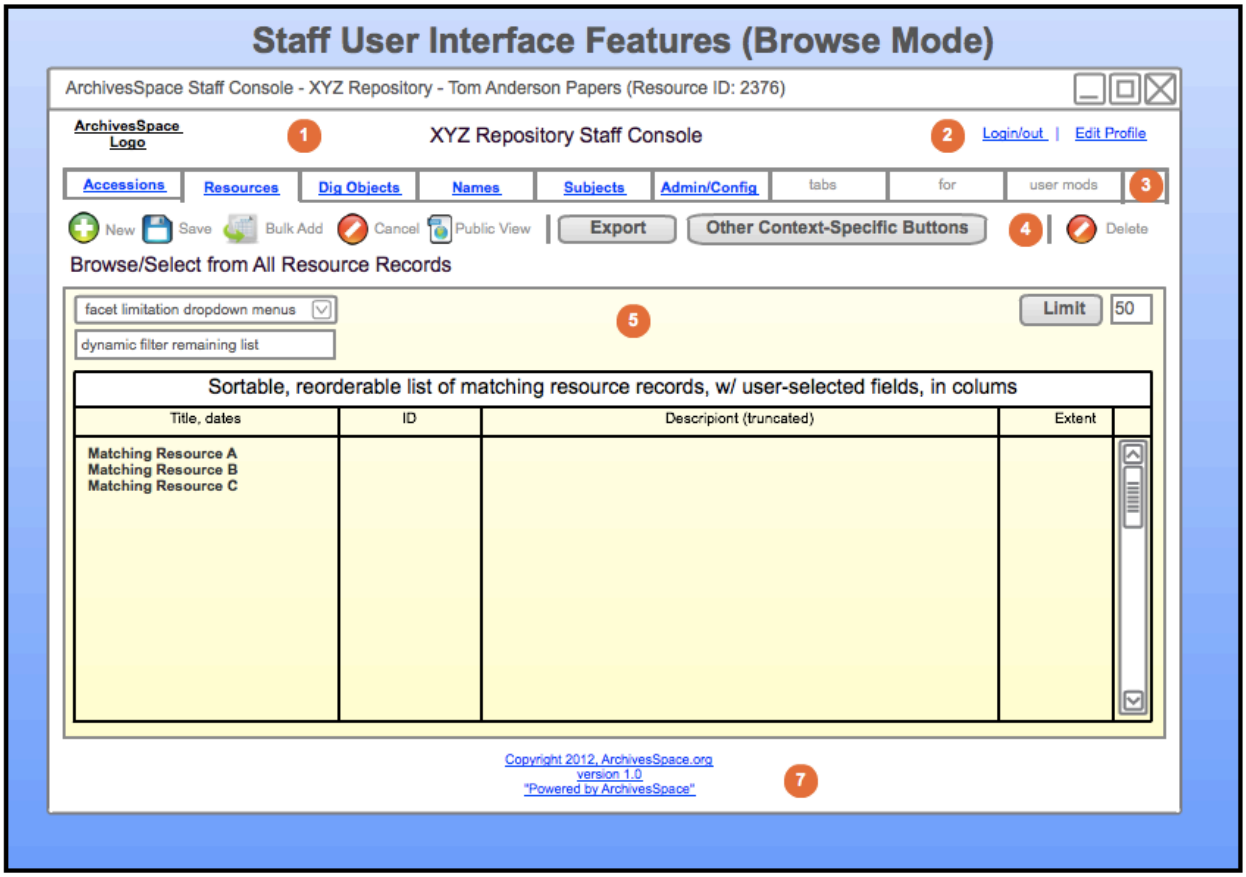
4. In the case where the potential objects to be linked are stored as hierarchical data (e.g. subjects and location records), the user may be provided a tree view to browse and select the correct node.

5. Complex linking mechanisms shall provide users a way to specify certain attribute values, which modify the relationship between the predicate operation. The following field types should be supported:
   - Drop-down menus (values from static or configurable lookup lists)
   - Text fields

StI-6: Non-Prescriptive Wireframes and Screenshots

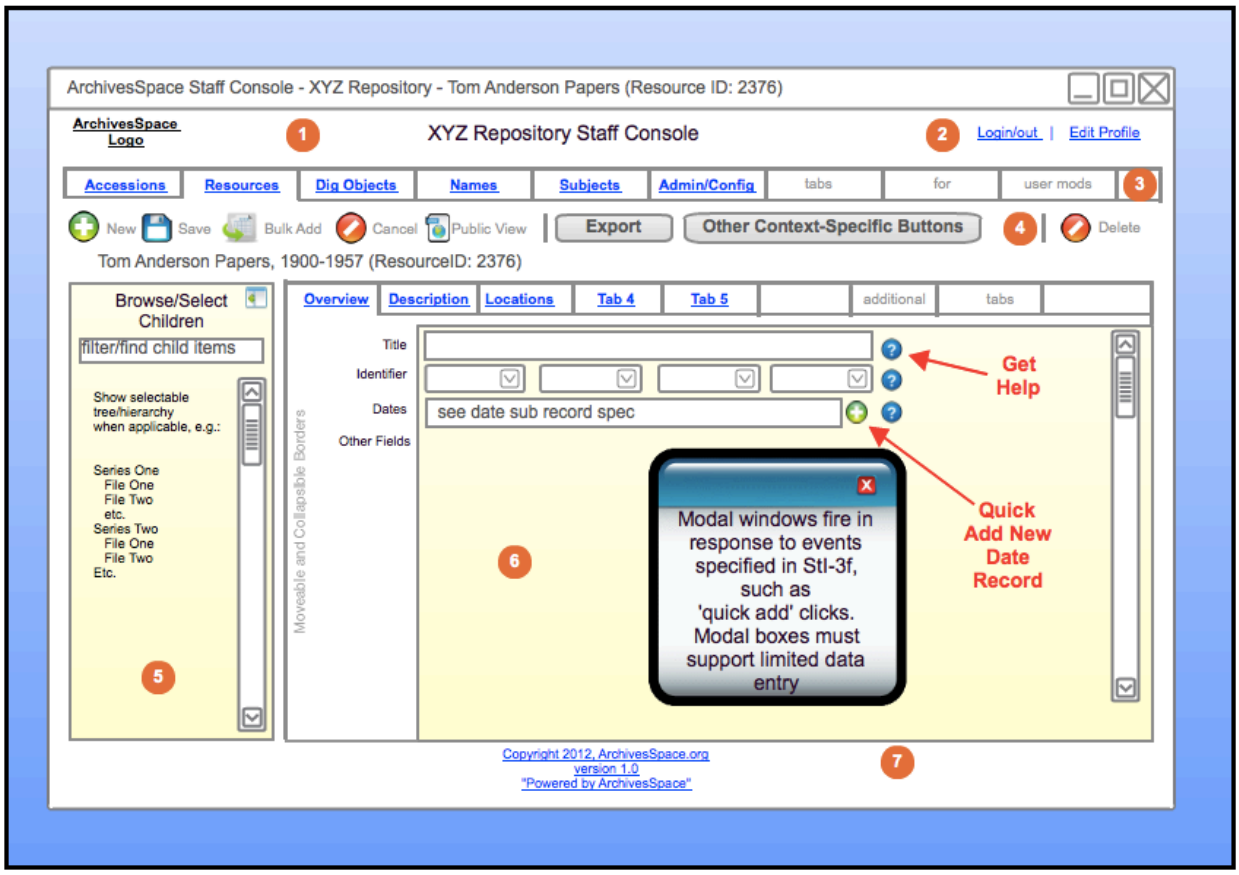
The following non-prescriptive wireframes and screenshots are provided to illustrate *potential* interface features, layout patterns and design elements; the actual interface must be subjected to an iterative design process and usability testing.

Figure One: Browse/Select View



1. Logo/Repository Area
2. User Profile Area
3. Application-Level Navigation Area/Tabs
4. Record-level Controls Area. Includes five fixed controls and additional context-specific controls.
5. Record-level Navigation Area/Browse
6. Editing/Display Area (not active in browse mode)
7. Footer

Figure 2: Single Record/Editing View



1. **Logo/Repository Area**
2. **User Profile Area**
3. **Application-Level Navigation Area/Tabs**
4. **Record-level Controls Area. Includes five fixed controls and additional context-specific controls.**
5. **Record-level Navigation Area/Browse**
6. **Editing/Display Area: Includes record title and single record, tabbed interface or bulk editing spreadsheets. Also supports AJAX-style popp-ups Described in detail in section Stl-**
7. **Footer**