

Really Big Wireless, Inc.

Contract Management System

Design Specification

Spring 2009



Chris Livdahl

**CSU Chico Online
CSCI 330 – Software Engineering
Instructor Dr. Len Fisk**

Change Control Page

The following information is being used to control and track modifications made to this document.

Revision & Date	Author	Sections	Summary of Change(s)
0.0 07/10/2009	C. Livdahl	All	<i>Document Creation</i>
0.1 07/11/2009- 07/12/2009	C. Livdahl	All	<i>Document Structure</i>
0.2 07/13/2009- 07/26/2009	C. Livdahl	1,2,3	<i>Added Content</i>
0.3 07/27/2009- 07/29/2009	C. Livdahl	3, 4, 5	<i>Added Content, Proofread</i>
1.0 07/30/2009	C. Livdahl	All	<i>Proofread, Finalize</i>

Table of Contents

1. Introduction	2
1.1 Purpose.....	2
1.2 Scope.....	2
1.3 Requirements	2
1.4 References	2
2. System Overview.....	4
2.1 System Architecture.....	4
2.2 Layered Architecture	6
2.3 Content Architecture.....	7
3. Program Level Description.....	9
3.1 Design Overview.....	9
3.2 Interface and Algorithm Design	10
3.2.1 Welcome Screen.....	11
3.2.2 Advanced Contracts Search.....	15
3.2.3 General Features – Advanced Companies Search.....	17
3.2.4 Contract Screen	19
3.2.5 Workflow Screen	27
3.2.6 Company Screen.....	29
3.2.7 Reports Screen.....	37
3.2.8 Last Visited Screen.....	39
3.2.9 Admin Screen	41
3.3 Data Structure Design.....	43
3.3.1 ERD – Color Key.....	43
3.3.2 ERD – General Data.....	44
3.3.3 ERD – Contracts Data	47
3.3.4 ERD – Company Data.....	55
3.3.5 ERD – Shared Data.....	59
3.3.6 ERD – User Data.....	65
3.3.7 ERD – Workflow Data	71
4. Quality Assurance Plan	79
4.1 Software Configuration Management.....	79
4.2 Requirements Changes.....	79
4.3 Design Changes	79
4.4 Formal Technical Reviews	80
4.5 Validation and Verification.....	80
4.6 SQA Personnel	81
5. Test Plan	82
5.1 Overview	82
5.2 Tracking Issues	83
5.3 Testing Outline.....	83
5.4 Releases and Future Defect Remedies	84

1. Introduction

1.1 Purpose

The purpose of this Design Specification document is to describe the design of the Contract Management System (CMS) that will be built for Really Big Wireless, Inc.

1.2 Scope

This document describes various technical aspects of the CMS in order to aid construction of this application. These aspects include the architecture, interfaces, and data structure of the application. In addition, the document presents both quality assurance and test plans.

1.3 Requirements

This document is a continuation of the Requirements Specification, previously delivered by the Contract Management Program (CMP) Team. The technical and application requirements are listed in detail in the Requirements Specification document and should be consulted as needed.

In addition, the Use Cases and Data Flow Diagrams (DFDs) are being kept in the Requirements Document only, in order to save space in this Design document. However, the User Interface mockups have been brought over to aid in the user interface design.

1.4 References

Sources of information/reference materials that were referenced in terms of content and format:

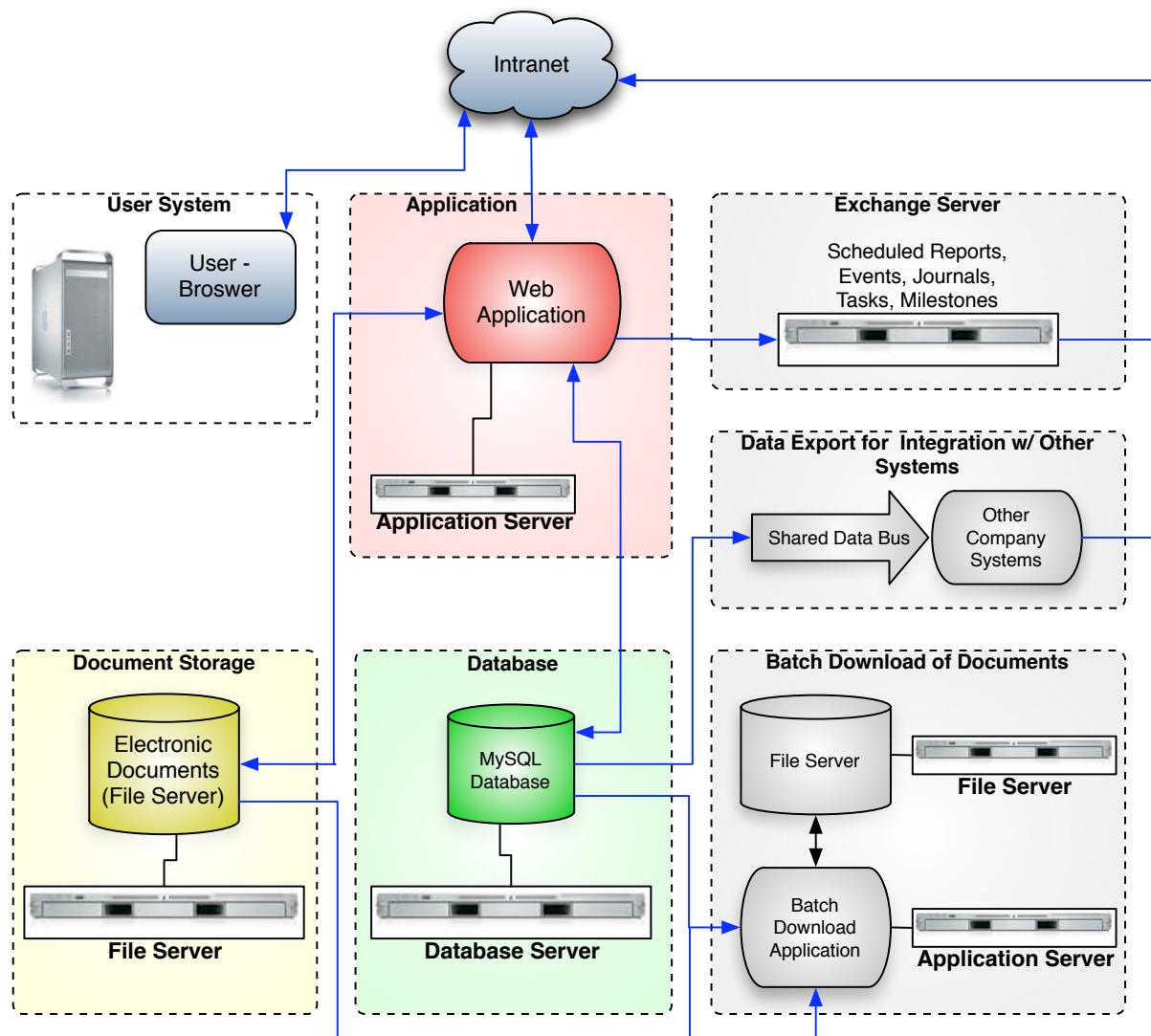
- *Pressman, Roger S.* "Software Engineering – A Practitioner's Approach", McGraw-Hill, New York, N.Y., 2005. ISBN: 0-07-285318-2
- *Tappero, Joseph.* "Terra Nova Investments, Property Management Software Design Description", March 2007.

- *Roberts, Carrie.* "Quality Treatment Systems, On-site Sewage Treatment Control, Design Specification", September 2005.
- *Padden, Ellen L.* "Naval Sea Logistics Center, Software Design Document", Spring 2006.
- *Grossman, Abby E.* "Dance Studio Technology Update Design Document", December 2005.

2. System Overview

2.1 System Architecture

The following diagram presents the Contract Management System (CMS) in the context of the applications, databases, and servers, as well as flows of data between them.



Intranet – The Intranet is the company-internal system of routers and cables, over which data traffic can flow between servers and user computers.

User System – The user system may be a desktop, laptop, or thin client. These computers should be equipped with the latest version of popular web browsers, such that the system works with Internet Explorer, Firefox, Google Chrome, and Safari. The operating system can be Windows or OS X. It is also necessary that the user have one of the Adobe PDF reader products, as well as Microsoft Word, for viewing documents, contracts, and reports. The user will use their browser to view the User Interface of the Web Application, contracts that are being created through the Workflow process, as well as to view search and report results before they are exported from the system.

Application – This application is the “Web Application” or “WebApp”; the functionality of which is discussed in the next section, “Program Level Description”. From a technical standpoint, the Web Application will be built using the popular Internet programming language PHP.

Document Storage – This area of document storage involves a standard file server for storing electronic documents pertaining to the Web Application.

Database – The Database is the main MySQL database for storing information pertaining to the Web Application. The Database server can be one of a wide variety of platforms, the most common at the company being variations of *nix and Windows servers.

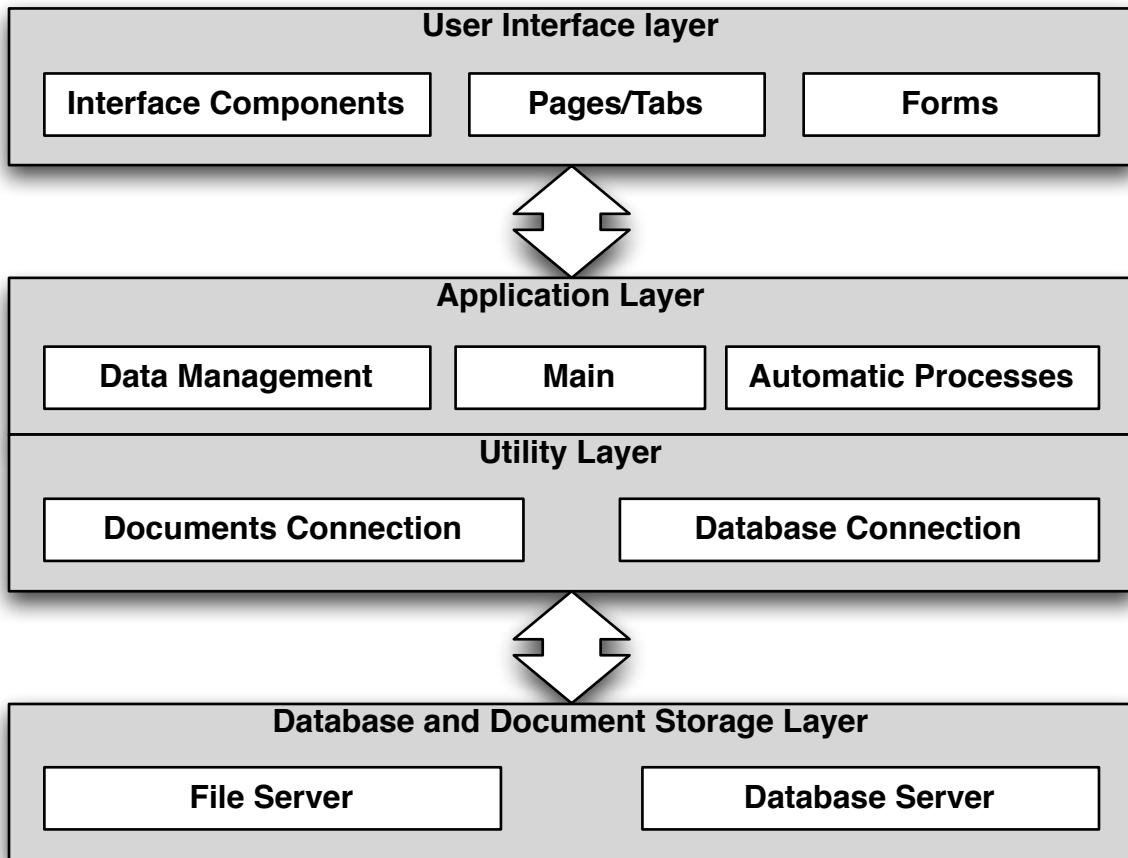
Exchange Server – The exchange server is used to email Scheduled Reports, Events, and Journals to users. The exchange server is Exchange 2003. But note that the functionality of this server is outside the scope of this system and design document.

Data Export for Integration w/ Other Systems – The data export is the sending of data from the Database over a shared bus, for use in other systems at Really Big Wireless. The interface consists of the tables of the Database being drawn out by the bus via a standard database connection. But note that the functionality of this process is outside the scope of this system and design document.

Batch Download of Documents – The batch download process is the process whereby sets of documents are delivered outside the system. The batch download process does need to interface with the CMS, but it can do so through a standard database connection, because the CMS is built using standard database technology. But note that the functionality of this application and database is outside the scope of this system and design document.

2.2 Layered Architecture

The CMS system uses a layered architecture, starting on the “outside” with the User Interface, then the Application layer, then the Utility layer, followed by the Database and Document Storage layer.



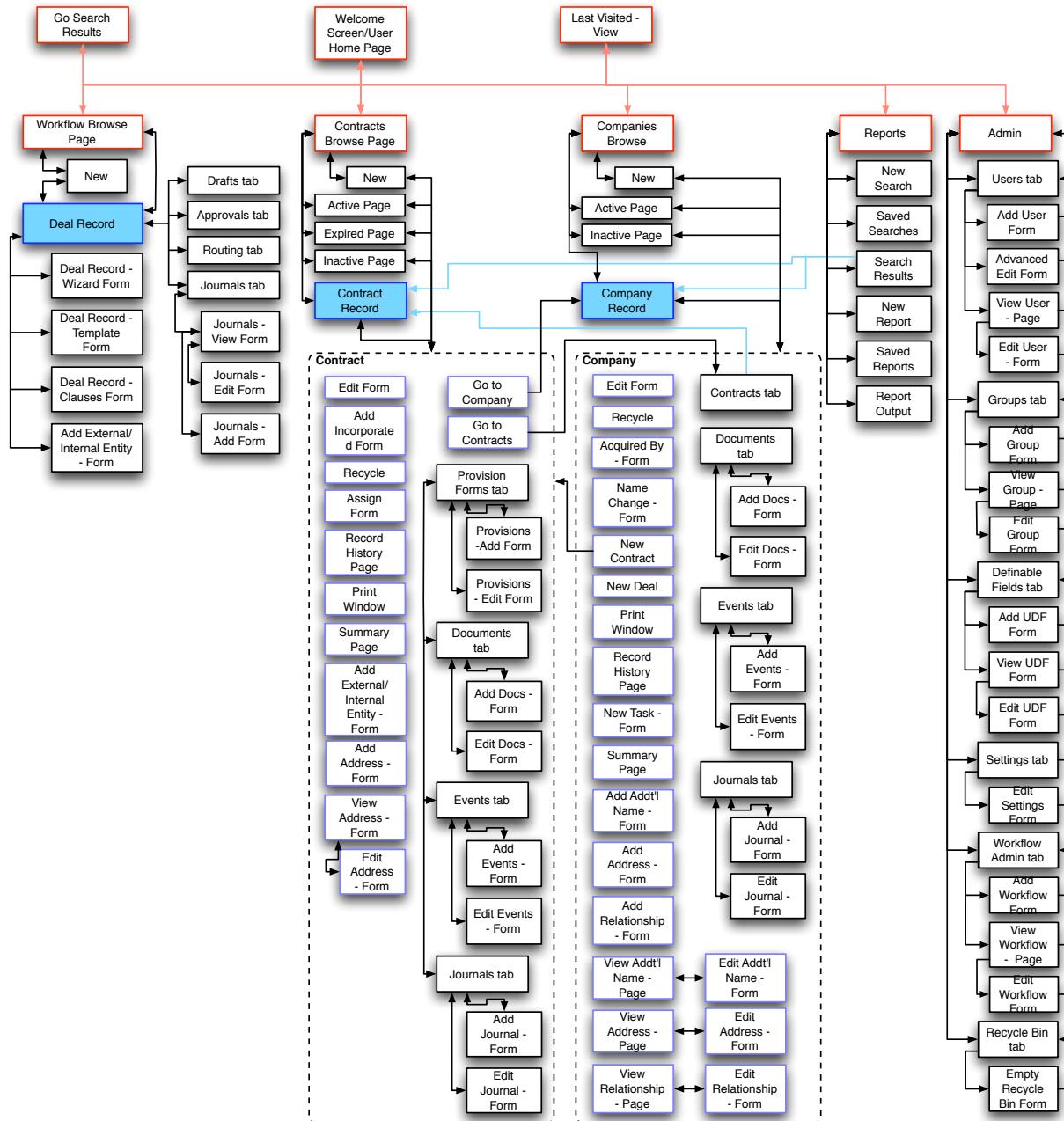
The User Interface Layer consists of the components seen by the user. These components include navigation elements (menus, buttons, dropdown boxes, search boxes, navigation items, etc.). There are forms that are presented to the user for data entry purposes.

The Application Layer consists of the data management or “business rules” as to how the application processes user data and commands. And, this layer also uses the Utility Layer to pass data to and from the Database and Document Storage Layer. This layer will also be responsible for sending data to the Exchange Server. The Application Layer also includes a Main area of functions that help to manage the entire Web Application. There are also Automatic Processes, such as where the system sends event alerts via email on certain pre-set dates.

The Utility Layer maintains a connection to the file server (documents connection) and database server (database connection).

The Database and Document Storage Layer contains the database server of the main CMS database, as well as the file server that contains electronic documents.

2.3 Content Architecture

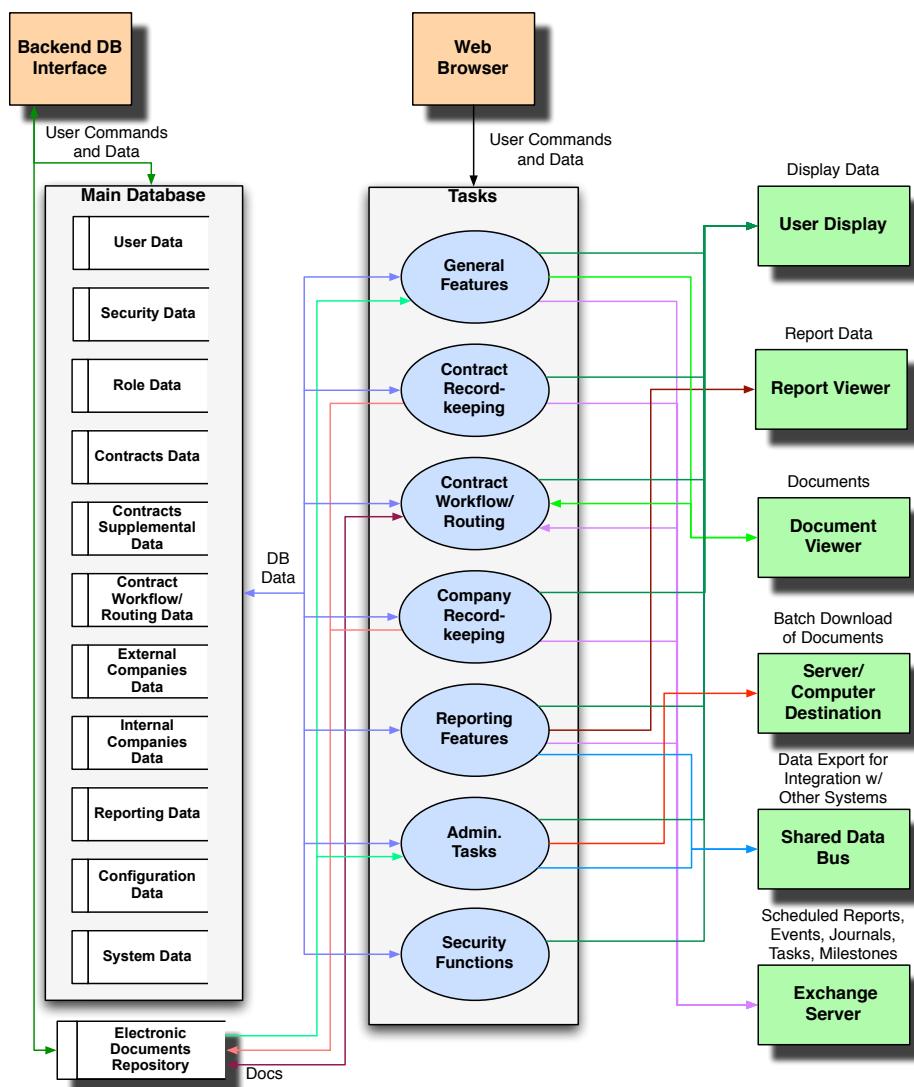


The above diagram shows a somewhat simplified map of the content architecture. The structure follows a hierarchical structure in general, with the navigational links from the main menu on the left side of the application showing in red. The paths in red denote that a user can move back and forth between each of the main areas of the main menu. Below each red box shows the pages and forms accessible for that respective area indicated in the red box. There is some navigation between the lower areas as indicated.

There types of pages that will be generated for the Web Application roughly fall into two categories: pages and forms. Pages and tabs are web pages that contain content that may come from multiple sources. Forms are web pages that have open "fields" where the user will input or change data and submit the data to the database portion of the system.

3. Program Level Description

3.1 Design Overview



The above diagram shows the main “Tasks” or functionality areas of the system from the Data Flow Diagram Level 1 in the Requirements document. The diagram also shows the Main database on the right, and then data output areas on the left. Note that the diagram has been somewhat modified for simplicity in that External/Internal Company Management functionality will be considered part of the Company Record-keeping area.

3.2 Interface and Algorithm Design

The user interfaces comprise the functionality available to a Read User, Write User, or Enterprise Administrator. (Note that the DB Admin shown in the use cases would access the system from backend off-the-shelf database tools, not the user interface.)

Note there are various modules & components that are used in the application, as indicated in Section 2.2 on layered architecture:

- Interface Components – Provides functionality for “drawing” the user interface, depending on where the user is at in the application.
- Pages/Tabs – These are pages/tabs that the user navigates to. The pages are assembled into HTML from PHP function calls that load data and from HTML pages for assembly into the viewable page.
- Forms – These forms are used when the user needs to input data. The forms are basically HTML pages and PHP code with form fields.
- Data Management – The data management area helps to store rules about loading data into the application. The Business Rules are enforced by using this module.
- Main – The Main module is the governor module of the Web Application. The Web Application accesses this module when the user first logs in. Other shared functionality for all modules lives here.
- Automatic Processes – Processes that are run in the background, such as Event Alerts and other such scheduled activity is managed by this area of functionality. Typically, “cron” jobs will be used in conjunction with PHP code to deliver Event Alerts.
- Documents Connection – This module handles the accessing of electronic documents.
- Database Connection – Handles the connection to the database. Though the standard MySQL commands are used to actually get the data, there are some functions for connecting to the database and getting at data. These functions can be used in order to keep code clean and manageable.

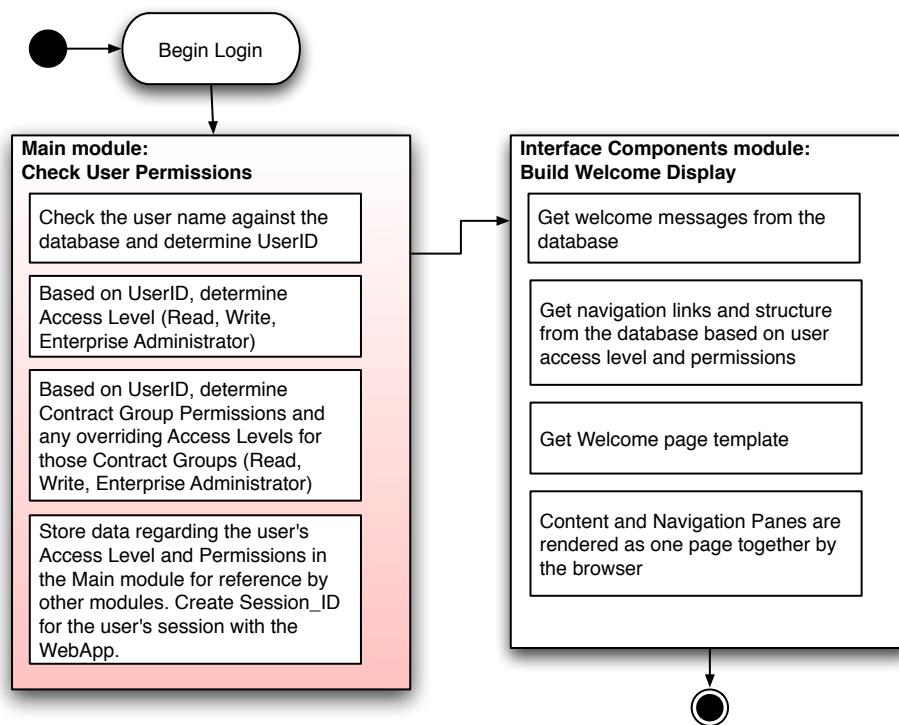
The components above, and the algorithms listed below are a 1.0 version, base-lined in the SCM tool, and will need to be made more robust as the construction of the system draws near. More detail needs to be added about when specific data is being read and written to the database in each module, as well as class names, function names, etc. In addition, more thought could be given to reusable components, as much system functionality follows repeatable patterns.

3.2.1 Welcome Screen

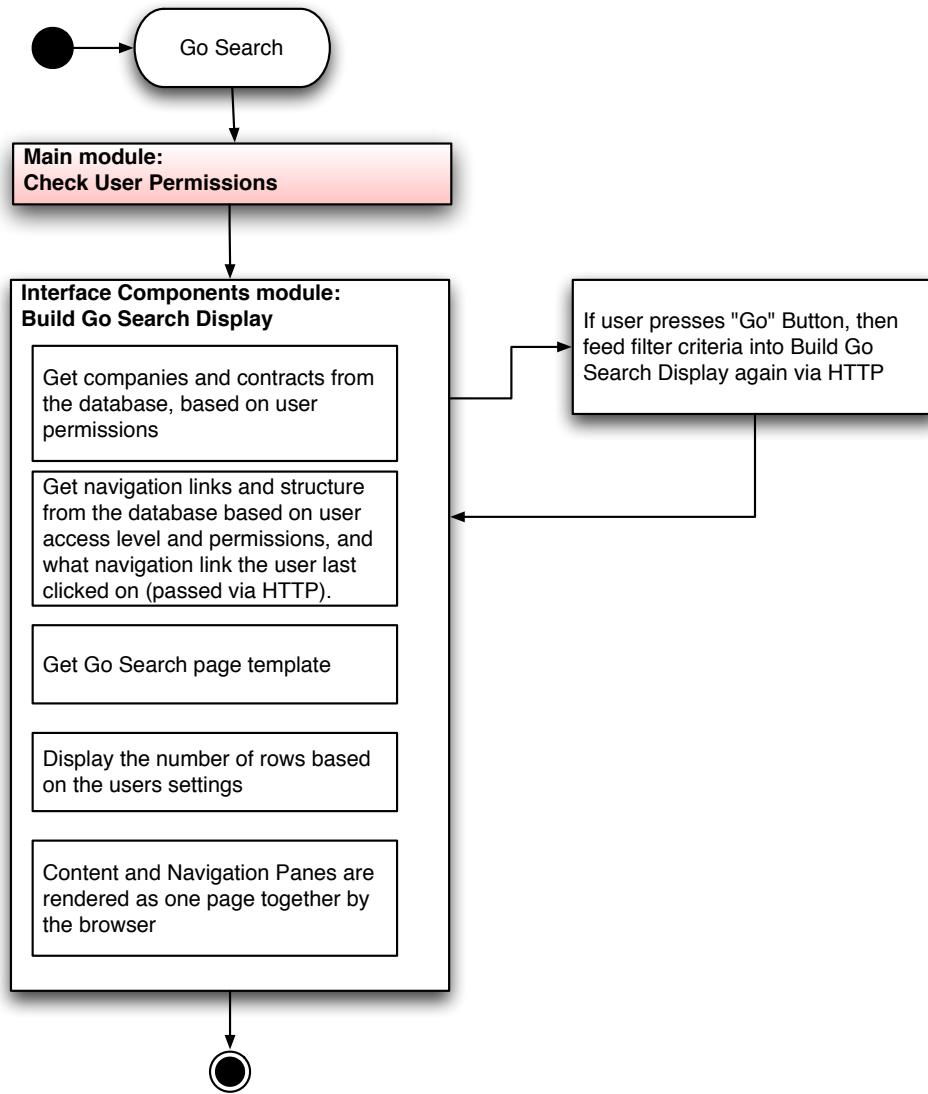
The user logs in to the application by a single sign-on process for all systems at Really Big Wireless. The password for a user is not stored in the CMS, but stored in the system that handles single sign-on. The handling of login information outside of the CMS is a benefit for the user, because they don't have to manage their account in multiple places.

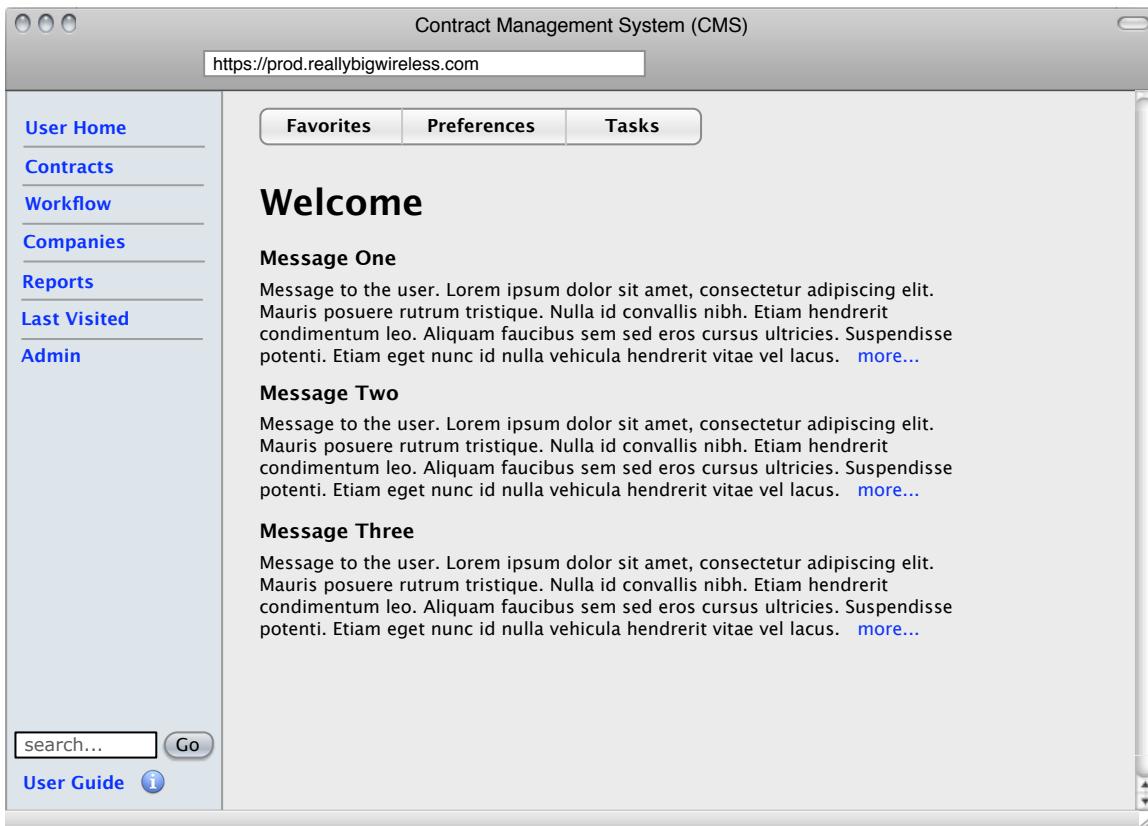
Based on the role of the user, upon login, the CMS displays the appropriate navigation pane at left, as well as the welcome page.

Note that the left hand area of the application is called the Navigation Pane. The main area of the application is called the Content Pane.



If the user puts data into the "Go Search" box on the lower left-hand side, then the contracts and companies database is queried for any relevant hits, much like a Google search.





In the above interface prototype, we see the welcome screen with several of the General Features from the use cases.

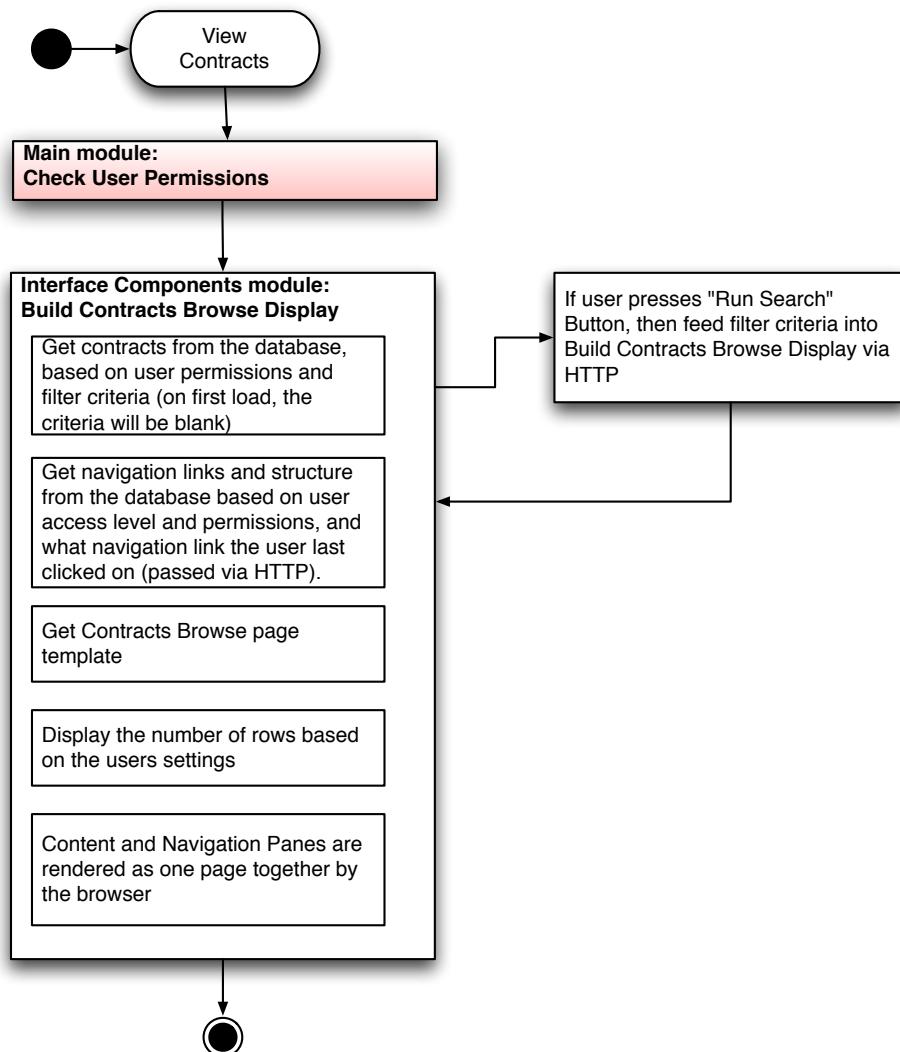
- Navigation – The area in the upper left has links to User Home, Contracts, Workflow, Companies, Reports, and Admin correspond to the major areas of the use cases.
- Search – The “search...” box with the “Go” button next to it is the area where a user performs the “Google-like” search discussed in the use cases.
- Advanced Search – This link takes the user to advanced searching/filtering.
- User Guide – This link takes the user to a user guide site out of the application, opening the guide in a new window. The user guide will be something like a wiki page with information about how to use the system, downloadable quick guides, and possibly even video of functionality.
- Favorites – This button takes the user to an area where they have “favorites” stored, such as links to saved reports, etc.

- Preferences – This button takes the user to an area that allows the user to customize the interface somewhat. The user could change the font size, colors, column width, number of search rows displayed, etc.
- Tasks – Note that this task functionality has been removed as a requirement by the stakeholders, and the user interface screen mockup needs to be updated. This decision was made because the enterprise implementation of Outlook contains a tasks feature.
- Messages – The messages that display on the welcome screen are messages to the user from the CMP Team, such as latest news regarding the system, and other updates.

3.2.2 Advanced Contracts Search

When the user clicks on the "Contracts" link in the Navigation Pane, the Contracts Browse page is displayed. In addition, the Contracts sub-menu displays in the Navigation Pane, i.e. navigational items that are one level lower display.

Initially, the page displays all contract records that the user has permissions to see. Then, the user can select from various filter options, and put in criteria to filter in, such as part of a contract name. When the user presses the Run Search button, then they will see a subset of contract records, based on their filter criteria as well as permissions.



The screenshot shows a web-based application window titled "Contract Management System (CMS)" with the URL "https://prod.reallybigwireless.com". On the left, there is a vertical navigation menu with links: "User Home", "Contracts" (with sub-links "New", "Active", "Expired", "Inactive"), "Workflow", "Companies", "Reports", "Last Visited", and "Admin". The main content area has two search boxes at the top, each with a "Filter options" dropdown and a plus sign icon for adding filters. Below the search boxes is a "Run Search" button. The central part of the screen displays a table of contract data with columns: "Contract Number", "Company Name", "Contract Purpose", "Effective Date", and "Expiration Date". Two rows of data are visible: one for contract 12345 (AB Company, Master Services Agmt, 01/01/2009, 01/01/2014) and another for contract 23456 (CDE LLC, Amendment #1, 03/24/2009, 03/24/2014). At the bottom of the table is a page navigation bar with icons for back, forward, and search, followed by "Page 1 of 1726". In the bottom left corner of the main area, there is a "search..." input field and a "Go" button. The bottom right corner contains a "User Guide" link with a help icon.

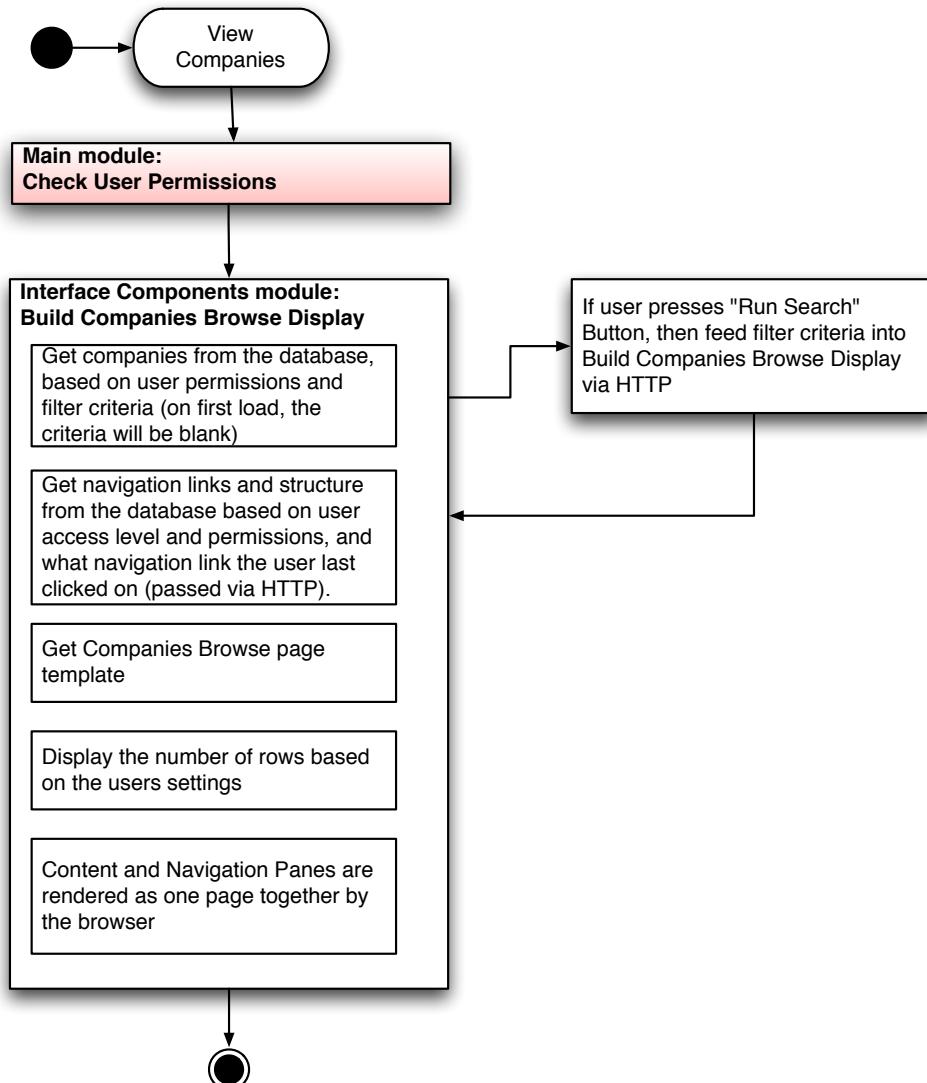
Contract Number	Company Name	Contract Purpose	Effective Date	Expiration Date
12345	AB Company	Master Services Agmt	01/01/2009	01/01/2014
23456	CDE LLC	Amendment #1	03/24/2009	03/24/2014

In the above screen mockup, we see the browsing and advanced search for contracts from the General Features use cases. The user gets to this screen by clicking on the “Contracts” link on the left hand panel. The user can add filter options by clicking on the plus sign to the right of the search boxes in the upper middle of the page. Search results are displayed as rows. The number of rows that display per screen are set per the user’s preferences. The user can access a contract record by clicking on one of the search result rows. (Note that this mockup only shows a couple results, whereas the table would be full in the real instance.) Note that the user can create a new contract by clicking on the “New” link at left.

3.2.3 General Features – Advanced Companies Search

When the user clicks on the “Companies” link in the Navigation Pane, the Companies Browse page is displayed. In addition, the Companies sub-menu displays in the Navigation Pane, i.e. navigational items that are one level lower display.

Initially, the page displays all company records that the user has permissions to see. Then, the user can select from various filter options, and put in criteria to filter in, such as the location of a company. When the user presses the Run Search button, then they will see a subset of company records, based on their filter criteria as well as permissions.

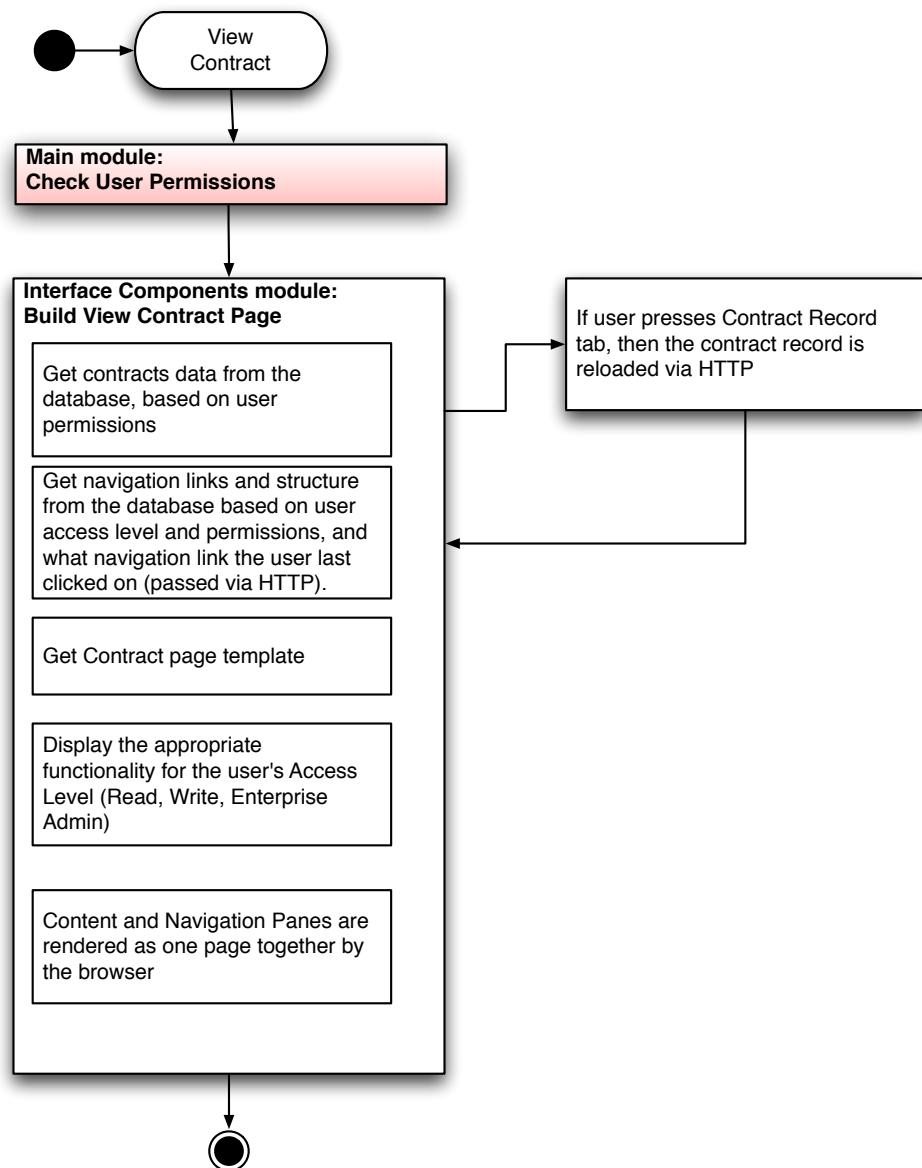


Company Name	Company ID	City	Country	Group
AB Company	54321	San Diego	United States	Trucking
CDE LLC	65432	Mexico City	Mexico	Primary

The above screen mockup shows a browse and advanced search screen for companies from the General Features use cases, much like the contract browse and advanced search. The user navigates to this page by the "Companies" link on the left panel. The user can add filter options by clicking on the plus sign to the right of the search boxes in the upper middle of the page. Search results are displayed as rows. The number of rows that display per screen are set per the user's preferences. The user can access a company record by clicking on one of the search result rows. (Note that this mockup only shows a couple results, whereas the table would be full in the real instance.) Note that a user can create a new company by clicking on the "New" link at left.

3.2.4 Contract Screen

If the user clicks on a contract record from the Go Search or from the Contracts Browse screen, then the user is brought into that contract record. The user can then maintain the contract data, based on if they have the correct access level and permissions.



Contract Management System (CMS)

<https://prod.reallybigwireless.com>

User Home	Contract Record	Provision Forms	Documents	Events	Journals		
Contracts	External Entities: XYZ Company, Inc.	Add External Entity					
New	Internal Entities: Really Big Wireless, Inc.	Add Internal Entity					
Active	Contract Details [ID: 6789]						
Expired	Contract Number: 12345						
Inactive	Contract Group: Trucking Sales						
Workflow	Agreement Type: Master						
Companies	Contract Type: Equipment						
Reports	Category 1: Loan						
Last Visited	Category 2: Tools						
Admin	Business Unit: Trucking						
	Contract Location: File Room						
	Contract Purpose: Use of Test Equipment						
	Contract Title: Equipment Loan Agreement						
	Contract Description: This is a test sentence. Other data could go into this field.						
	Edit Add Incorporated Recycle Assign Renew Record History Print Go to Company Go to Contracts Summary						
	Term Information						
	Effective Date: 01/01/2009	Original Expiration Date: 01/01/2009					
	Initial Term (in months): 60	Status: Active					
	Term Type: Fixed	Renewal Type:					
	POC Information						
	Current Contract Mgr: Smith, John	Current Attorney: Lawson, Lew					
	Address Information						
	Add Address						
	Location Type	Address Line 1	Address Line 2	City	State	Zip	Country
	Notice Address	54321 Your St	PO Box 123	San Diego	CA	92121	United States

search... User Guide [i](#)

The above screen mockup shows a contract record/profile.

At the top of the record are tabs for switching to various areas pertaining to the contract. Obviously, all the tabs will need to be prototyped as needed.

Contract Record tab – This screen. If the user clicks on this tab, then the contract record is loaded again.

Provision Forms tab – This tab contains additional data for the Contract Group managing this record. If the user clicks on this tab, then the provision forms for this contract record are displayed. The View, Add, Edit, Recycle paradigm is very repetitive throughout the application, so the processes are explained once below for the Address Information section, but the paradigm can be applied to this area.

Documents tab – This tab contains the electronic documents relevant to this record (including a scan of the signed contract). If the user clicks on this tab, then the electronic documents for this contract are displayed as links. If the user clicks on one of these document links, then the document will open in a document viewer, such as Adobe Acrobat or Word. The user can view, edit, and add documents from this tab. View, Add, Edit, Recycle.

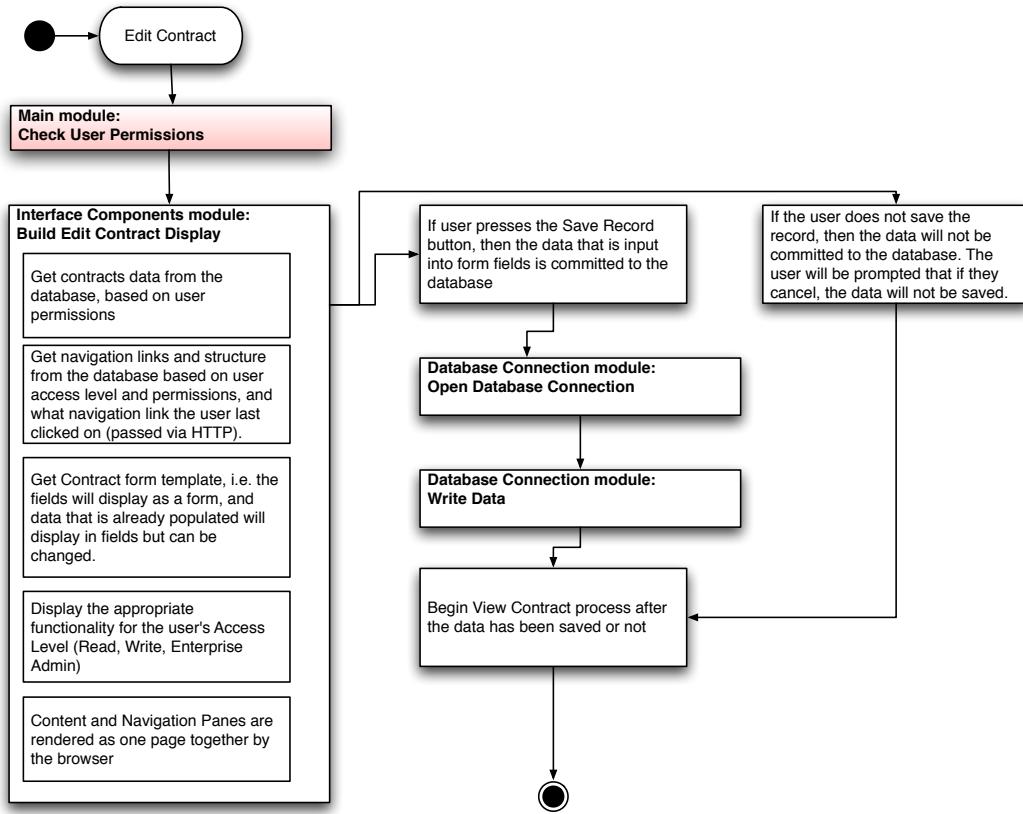
Events tab – This tab contains the email event alerts that are scheduled to be sent to a user (or group of users) per certain relevant dates. If the user clicks on this tab, then they will be shown a page with the event alerts scheduled to be sent, as well as closed events. View, Add, Edit, Recycle.

Journals tab – This tab contains functionality for users to be able to send messages to others “ad-hoc”, along with attaching one or more contracts pertaining to this contract record. If the user clicks on this tab, then they will be shown the journals for this contract record. They can also send a journal by typing a message, selecting recipients, and optionally attaching documents. View, Add, Edit, Recycle.

Below the tabs is an area with information about what external and internal entities (companies) pertain to this contract. The user can add additional companies by clicking on the buttons to the right of the section (the “plus signs”).

The next section is the Contract Details section, which contain the main data about the contract.

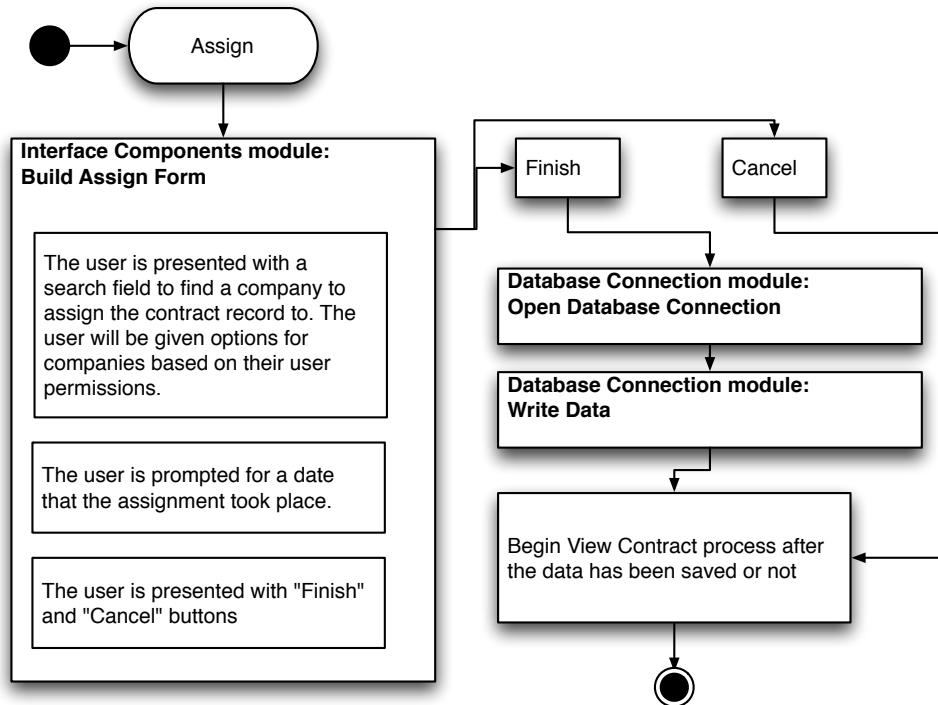
Edit button – The user can edit this data by clicking on the Edit button on the right hand side (which is the menu for performing contract maintenance). Note the user will not be presented this button if they do not have appropriate permissions.



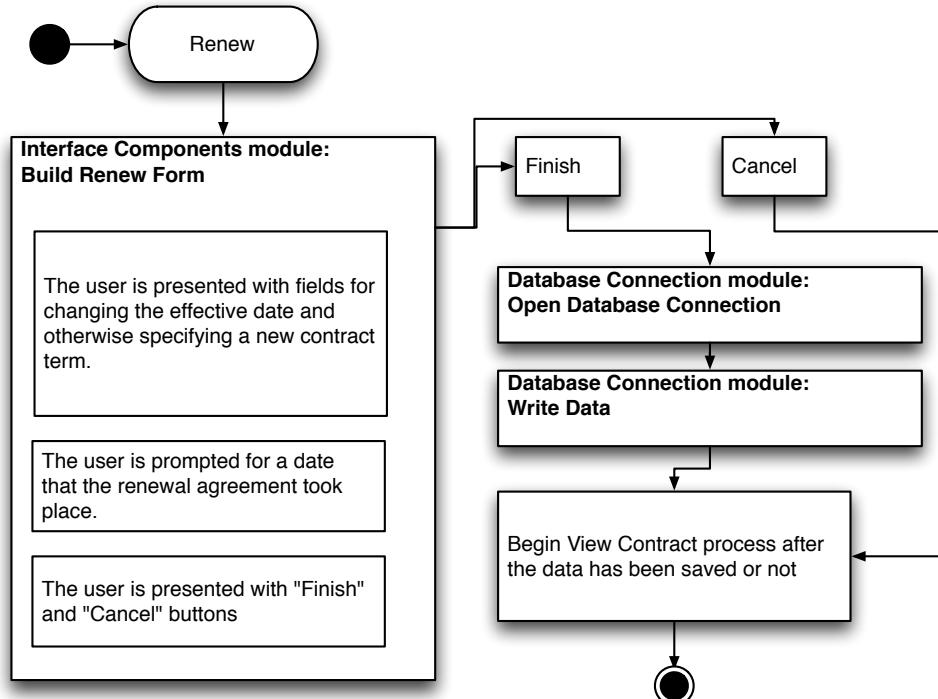
Add Incorporated button – Allows the user to add an amendment or other such “child” record to this contract record. This process is basically a duplicate of the Edit Contract functionality, with the exception that the amendment is linked to the “parent” contract record. Note the user will not be presented this button if they do not have appropriate permissions.

Recycle button – Lets the user recycle the record to the “recycle bin”. This button simply changes the contract record status to “recycled”. A message in red next to the Contract ID should display that the contract is recycled. The Enterprise Administrator will be able to clear the contract record from the recycle bin in the Admin functionality area. Note the user will not be presented this button if they do not have appropriate permissions.

Assign button – Will track the assignment of this record from one external/internal entity to another. Note the user will not be presented this button if they do not have appropriate permissions.

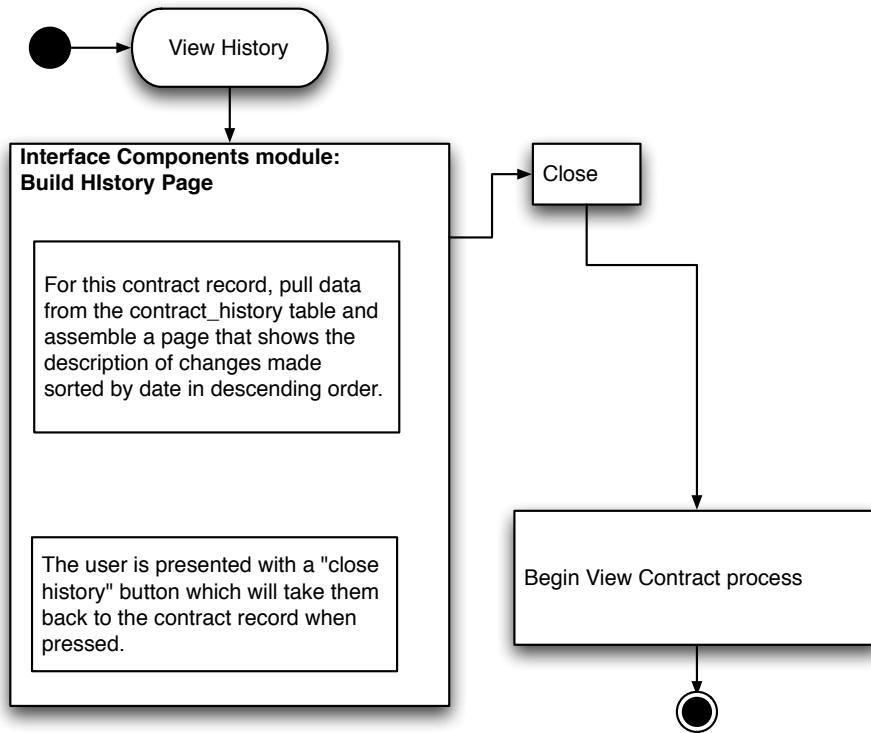


Renew button – Presents a “wizard” that walks the user through steps to extend the expiration date of the contract. Note the user will not be presented this button if they do not have appropriate permissions.



Record History button – Shows the user the history of all the changes made to the contract record, who made the changes, and when they made the

changes.

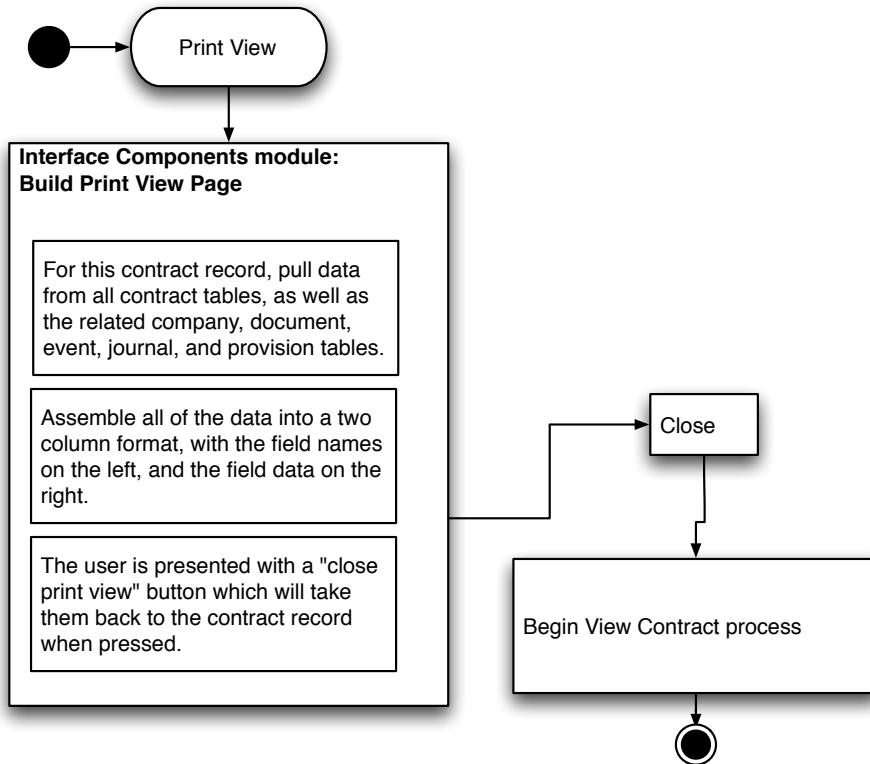


Print button – Prints this screen of information.

Go to the Company – Takes the user to the internal entity company record. If there are multiple internal entities, then the user is presented with an option of which one to visit. Viewing the company record is covered in the Company Screen section.

Go to the Contracts – Takes the user to the list of contracts for the internal entity company record. If there are multiple internal entities, then the user is presented with an option of which contract list to visit. Viewing the contracts in the company record is covered in the Company Screen section.

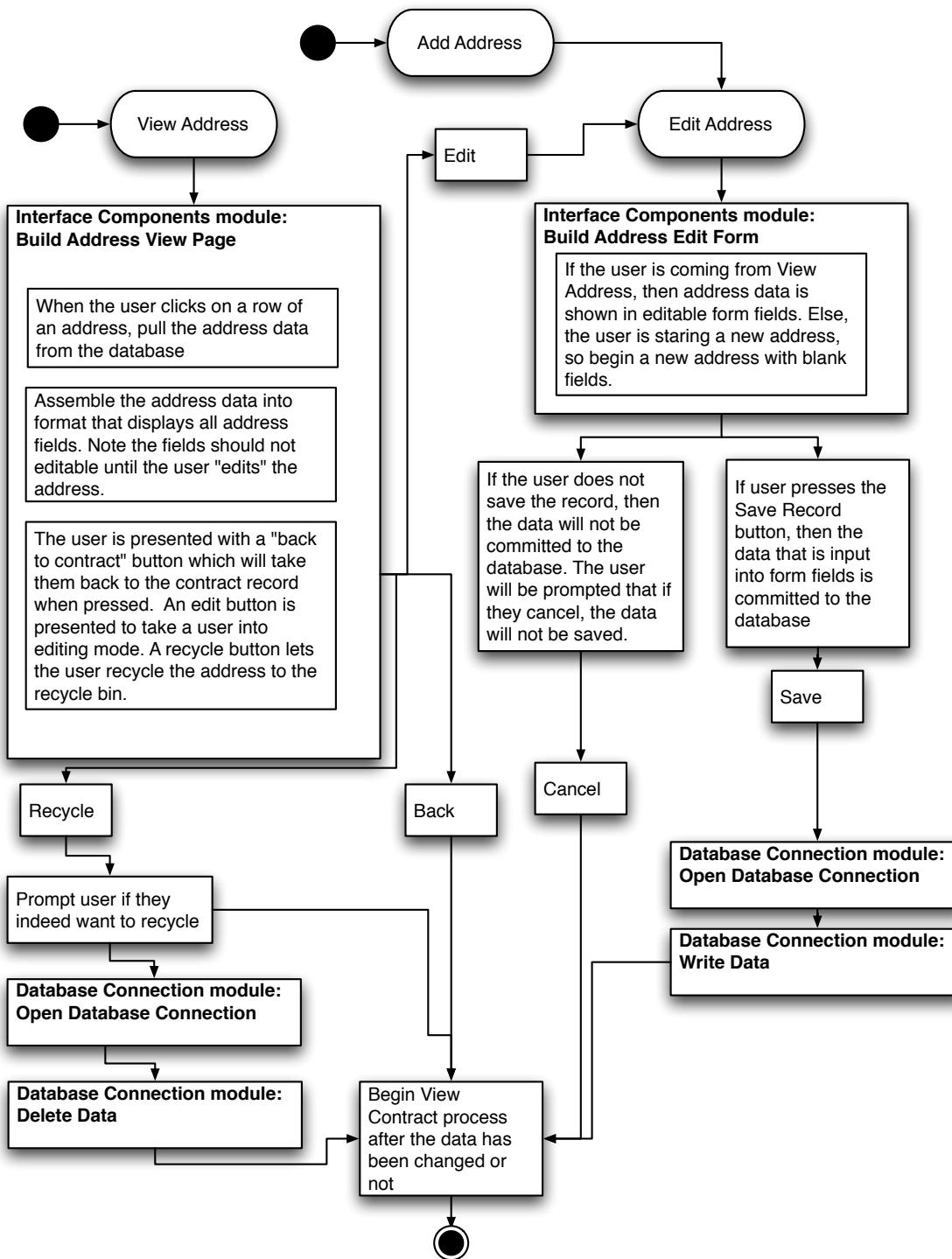
Summary – Shows the user a summary of all contract data (from all tabs), along with the option to print all data, or a subset of data. Note that the user will use their browser's print facility to print the page(s) of summary data.



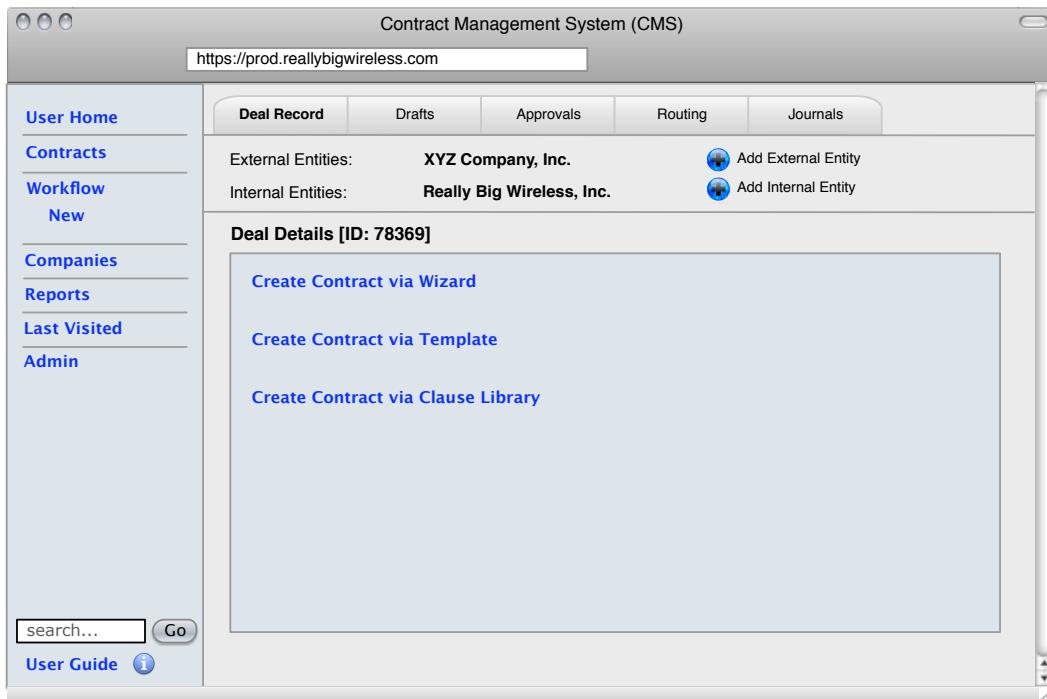
The Term Information section contains the data about the term of the contract. The effective date and expiration date have blue boxes next to them that present the user with a calendar to choose dates from (or the user can just type in the date). The dates cannot be modified unless the Edit button has been pressed. This data (along with the other contract data) can be edited with the contract Edit button described above.

The POC Information section contains the data about who is managing this contract. This data (along with the other contract data) can be edited with the contract Edit button described above.

The Address Information section contains the data about where to find the notice address for this contract (as explained in an earlier section). The user can view an address by click on the row for that address. The address can then be edited by pressing the Edit button in the address view. A new address can be added by pressing on the blue "plus sign" button, and the user is taken to a form for entering a new address.



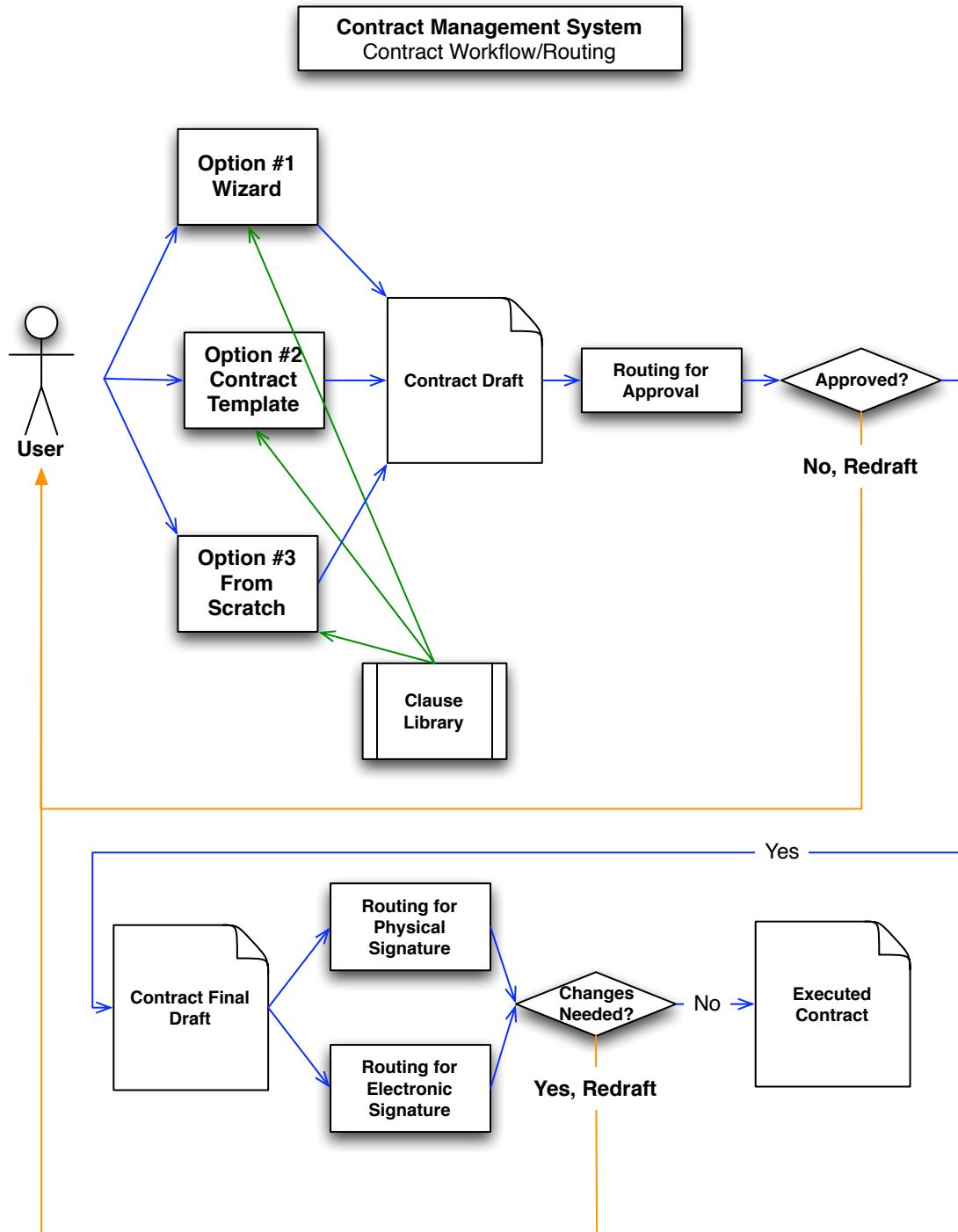
3.2.5 Workflow Screen



The above screen mockup shows the start of a new “Deal”, which is basically an object that leads to a created contract record.

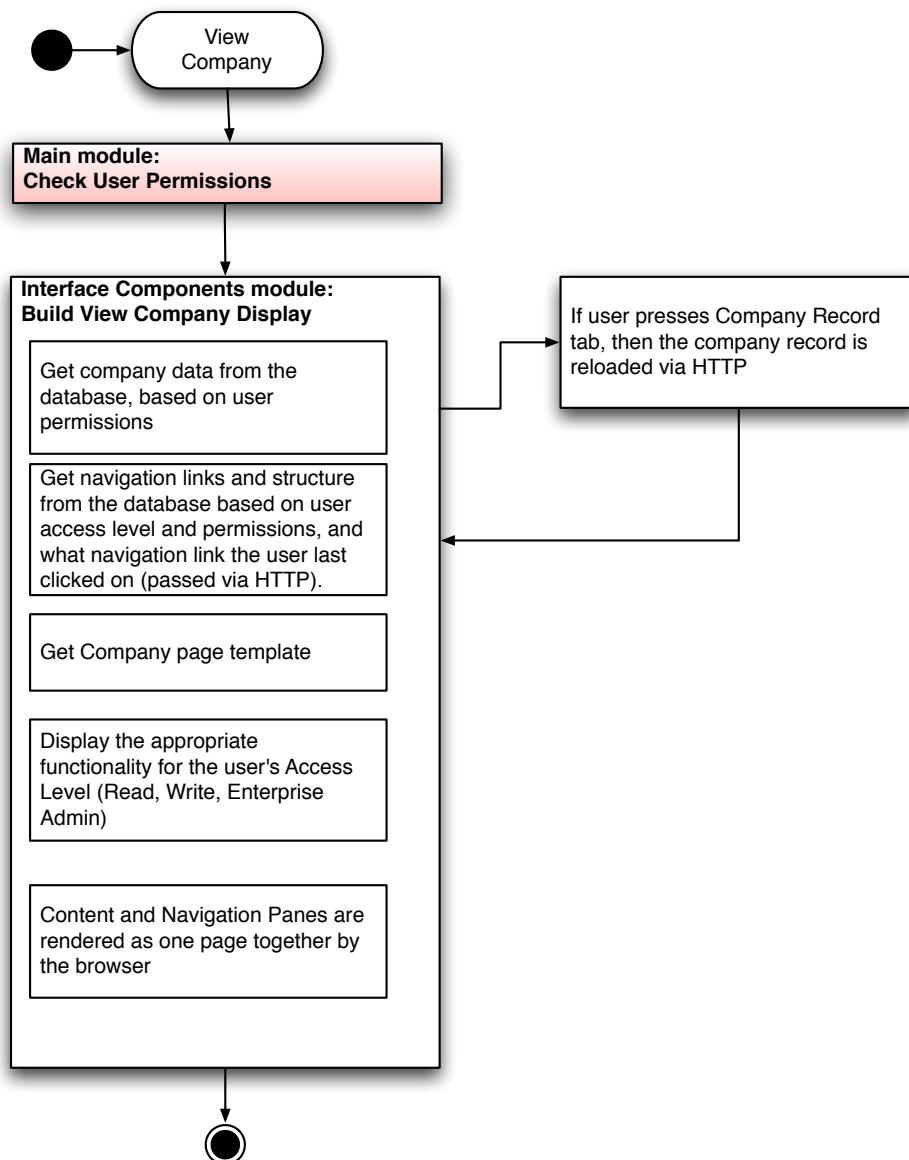
Note the change from the Requirements document where the first item was “Create Contract via Workflow” and has been replaced by “Create Contract via Wizard”.

Section 3.3.7 goes into great detail about the design of this area of functionality. The below flow chart should help the reader understand the process from a user’s perspective.



3.2.6 Company Screen

If the user clicks on a company record from the Go Search or from the Company Browse screen, then the user is brought into that company record. The user can then maintain the company data, based on if they have the correct access level and permissions.



Contract Management System (CMS)

https://prod.reallybigwireless.com

User Home Contracts Workflow Companies New Active Inactive Reports Last Visited Admin	<table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 15%;">Company Record</td> <td style="width: 15%;">Contracts</td> <td style="width: 15%;">Documents</td> <td style="width: 15%;">Events</td> <td style="width: 15%;">Journals</td> </tr> </table> <div style="border: 1px solid #ccc; padding: 5px; margin-bottom: 10px;"> <p>Company Details [ID: 9876]</p> <table border="0" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 25%;">Internal/External:</td> <td style="width: 75%;"><input type="text" value="External"/></td> </tr> <tr> <td>Company Name:</td> <td><input type="text" value="XYZ Company, Inc."/></td> </tr> <tr> <td>Address Line 1:</td> <td><input type="text" value="55564 Their Street"/></td> </tr> <tr> <td>Address Line 2:</td> <td><input type="text"/></td> </tr> <tr> <td>City:</td> <td><input type="text" value="San Diego"/></td> </tr> <tr> <td>State:</td> <td><input type="text" value="CA"/></td> </tr> <tr> <td>Zip/Postal Code:</td> <td><input type="text" value="92121"/></td> </tr> <tr> <td>Country:</td> <td><input type="text" value="United States"/></td> </tr> <tr> <td>Phone Number:</td> <td><input type="text" value="1-619-555-1234"/></td> </tr> <tr> <td>Fax Number:</td> <td><input type="text" value="1-619-555-4321"/></td> </tr> <tr> <td>Company Group:</td> <td><input type="text" value="Trucking"/></td> </tr> <tr> <td>Status:</td> <td><input type="text" value="Active"/></td> </tr> <tr> <td>Web Address:</td> <td><input type="text" value="www.xyzcompanyinc.com"/></td> </tr> <tr> <td>Accounting Number:</td> <td><input type="text" value="67894"/></td> </tr> <tr> <td>Description:</td> <td><input type="text" value="This is a free text field."/></td> </tr> </table> </div> <div style="border: 1px solid #ccc; padding: 5px; margin-bottom: 10px;"> <p>Additional Names</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="width: 25%;">Company Name</th> <th style="width: 25%;">Name Type</th> <th style="width: 25%;">Name Effective Date</th> <th style="width: 25%;">Description</th> </tr> </thead> <tbody> <tr> <td>ZZZ Company, LLC</td> <td>Formerly Known As</td> <td>07/01/2008</td> <td>Company decided to chang...</td> </tr> </tbody> </table> </div> <div style="border: 1px solid #ccc; padding: 5px; margin-bottom: 10px;"> <p>Additional Addresses</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="width: 12.5%;">Location Type</th> <th style="width: 12.5%;">Address Line 1</th> <th style="width: 12.5%;">Address Line 2</th> <th style="width: 12.5%;">City</th> <th style="width: 12.5%;">State</th> <th style="width: 12.5%;">Zip</th> <th style="width: 12.5%;">Country</th> </tr> </thead> <tbody> <tr> <td>Billing Address</td> <td>PO Box 555</td> <td></td> <td>San Diego</td> <td>CA</td> <td>92121</td> <td>United States</td> </tr> </tbody> </table> </div> <div style="border: 1px solid #ccc; padding: 5px; margin-bottom: 10px;"> <p>Company Relationships</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="width: 33%;">Company Name</th> <th style="width: 33%;">Relationship Type</th> <th style="width: 33%;">Description</th> </tr> </thead> <tbody> <tr> <td>ZWZ Corporation</td> <td>Joint Relationship</td> <td>New joint venture...</td> </tr> </tbody> </table> </div>	Company Record	Contracts	Documents	Events	Journals	Internal/External:	<input type="text" value="External"/>	Company Name:	<input type="text" value="XYZ Company, Inc."/>	Address Line 1:	<input type="text" value="55564 Their Street"/>	Address Line 2:	<input type="text"/>	City:	<input type="text" value="San Diego"/>	State:	<input type="text" value="CA"/>	Zip/Postal Code:	<input type="text" value="92121"/>	Country:	<input type="text" value="United States"/>	Phone Number:	<input type="text" value="1-619-555-1234"/>	Fax Number:	<input type="text" value="1-619-555-4321"/>	Company Group:	<input type="text" value="Trucking"/>	Status:	<input type="text" value="Active"/>	Web Address:	<input type="text" value="www.xyzcompanyinc.com"/>	Accounting Number:	<input type="text" value="67894"/>	Description:	<input type="text" value="This is a free text field."/>	Company Name	Name Type	Name Effective Date	Description	ZZZ Company, LLC	Formerly Known As	07/01/2008	Company decided to chang...	Location Type	Address Line 1	Address Line 2	City	State	Zip	Country	Billing Address	PO Box 555		San Diego	CA	92121	United States	Company Name	Relationship Type	Description	ZWZ Corporation	Joint Relationship	New joint venture...
Company Record	Contracts	Documents	Events	Journals																																																												
Internal/External:	<input type="text" value="External"/>																																																															
Company Name:	<input type="text" value="XYZ Company, Inc."/>																																																															
Address Line 1:	<input type="text" value="55564 Their Street"/>																																																															
Address Line 2:	<input type="text"/>																																																															
City:	<input type="text" value="San Diego"/>																																																															
State:	<input type="text" value="CA"/>																																																															
Zip/Postal Code:	<input type="text" value="92121"/>																																																															
Country:	<input type="text" value="United States"/>																																																															
Phone Number:	<input type="text" value="1-619-555-1234"/>																																																															
Fax Number:	<input type="text" value="1-619-555-4321"/>																																																															
Company Group:	<input type="text" value="Trucking"/>																																																															
Status:	<input type="text" value="Active"/>																																																															
Web Address:	<input type="text" value="www.xyzcompanyinc.com"/>																																																															
Accounting Number:	<input type="text" value="67894"/>																																																															
Description:	<input type="text" value="This is a free text field."/>																																																															
Company Name	Name Type	Name Effective Date	Description																																																													
ZZZ Company, LLC	Formerly Known As	07/01/2008	Company decided to chang...																																																													
Location Type	Address Line 1	Address Line 2	City	State	Zip	Country																																																										
Billing Address	PO Box 555		San Diego	CA	92121	United States																																																										
Company Name	Relationship Type	Description																																																														
ZWZ Corporation	Joint Relationship	New joint venture...																																																														
<input type="text" value="search..."/> <input type="button" value="Go"/> User Guide	<input type="button" value="Edit"/> <input type="button" value="Recycle"/> <input type="button" value="Acquired By"/> <input type="button" value="Name Change"/> <input type="button" value="New Contract"/> <input type="button" value="New Deal"/> <input type="button" value="Print"/> <input type="button" value="Record History"/> <input type="button" value="New Task"/> <input type="button" value="Summary"/>																																																															

The above screen mockup shows a company record/profile.

At the top of the record are tabs for switching to various areas pertaining to the company. Obviously, all the tabs will need to be prototyped as needed.

Company Record tab – This tab shows the main data area for the company. If the user presses on this tab, then the company record page is refreshed.

Contracts tab – This tab contains simply a sequential listing of the contracts pertaining to the company. The user can use the data on this tab to navigate to the relevant contract records, by simply clicking on the contract record listed.

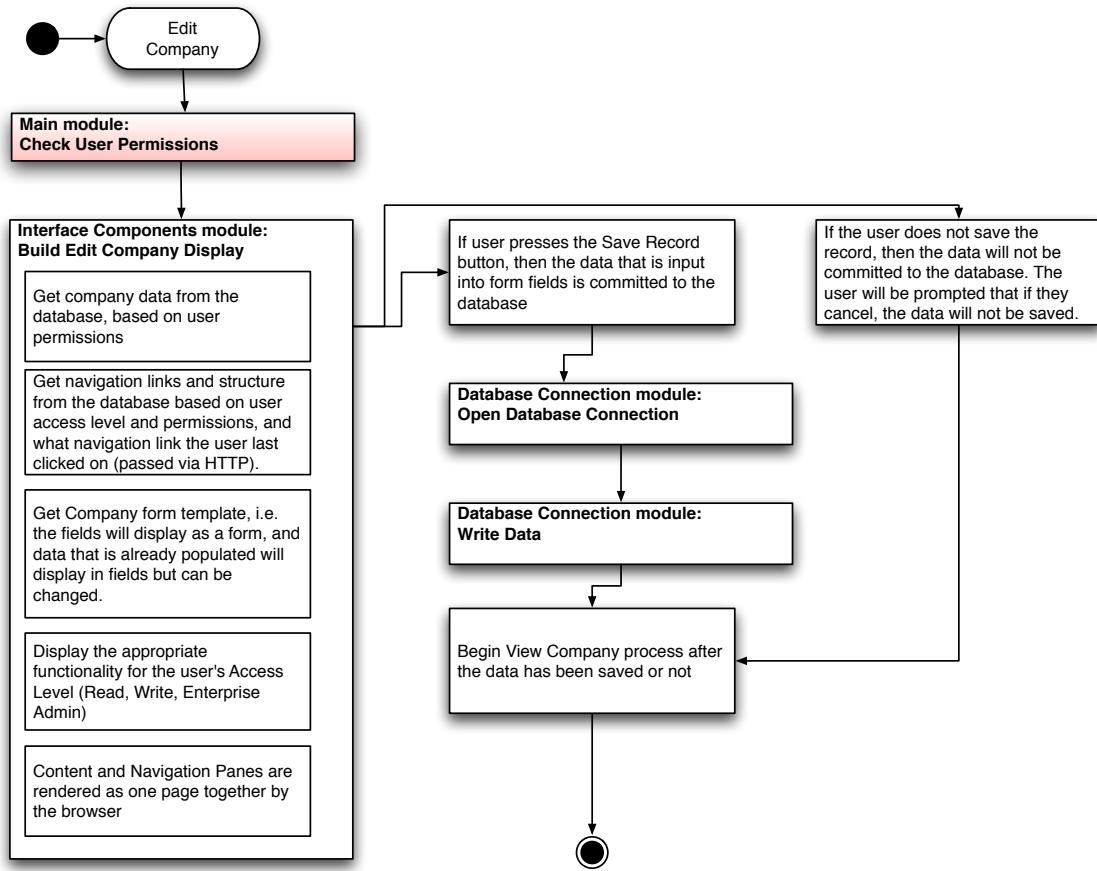
Documents tab – This tab contains any uploaded electronic documents pertaining to the company, such as certificates of incorporation, etc. If the user clicks on this tab, they will see a sequential listing of documents, which can be viewed, edited, and added. The View, Add, Edit, Recycle paradigm is very repetitive throughout the application, so the processes are explained once below for the Additional Address section, but the paradigm can be applied to this area.

Events tab – This page shows event alerts that are set up to be emailed to people per certain dates pertaining to the company (such as milestones). View, Add, Edit, Recycle.

Journals tab – This tab contains functionality for users to be able to send messages to others “ad-hoc”, along with attaching one or more contracts pertaining to this company record. View, Add, Edit, Recycle.

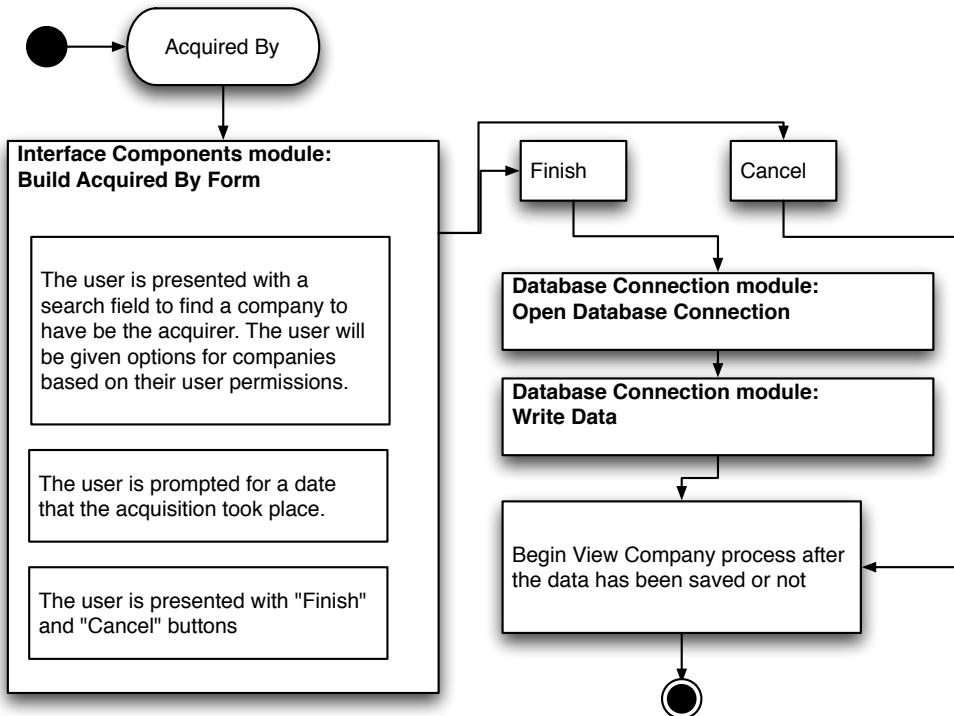
The next section is the Company Details section, which contain the main data about the company.

Edit button – The user can edit this data by clicking on the Edit button on the right hand side (which is the menu for performing company maintenance).

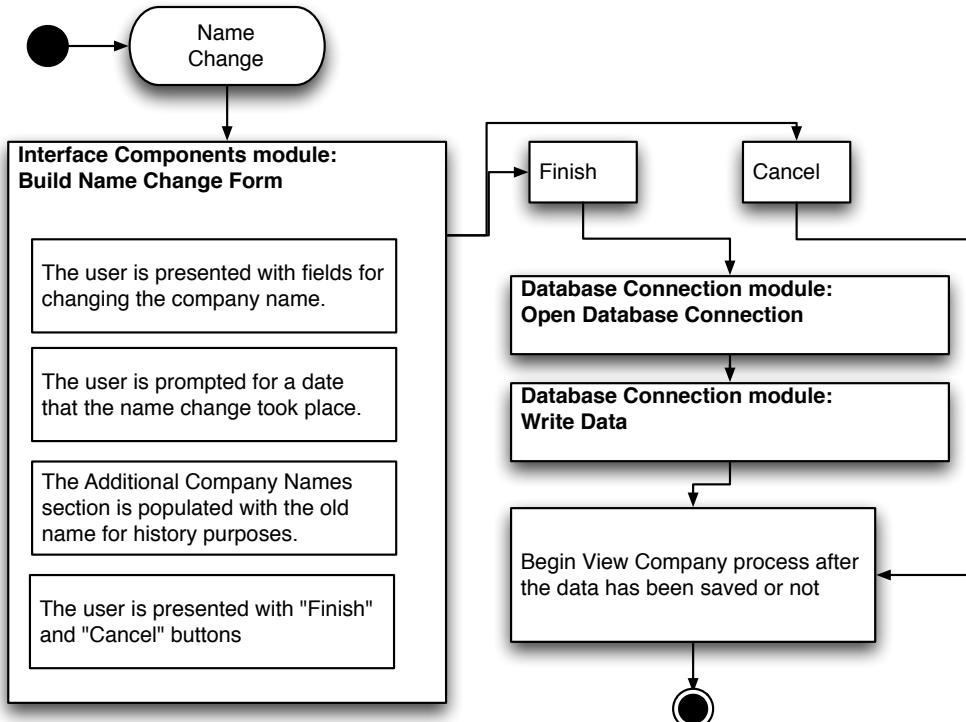


Recycle button – Lets the user recycle the record to the “recycle bin”. This button simply changes the company record status to “recycled”. A message in red next to the Company ID should display that the company is recycled. The Enterprise Administrator will be able to clear the company record from the recycle bin in the Admin functionality area. Note the user will not be presented this button if they do not have appropriate permissions.

Acquired By button – Will track the acquisition of this entity/company to another external/internal entity/company.



Name Change button – Presents a “wizard” that walks the user through steps to process a company that has changed their name.

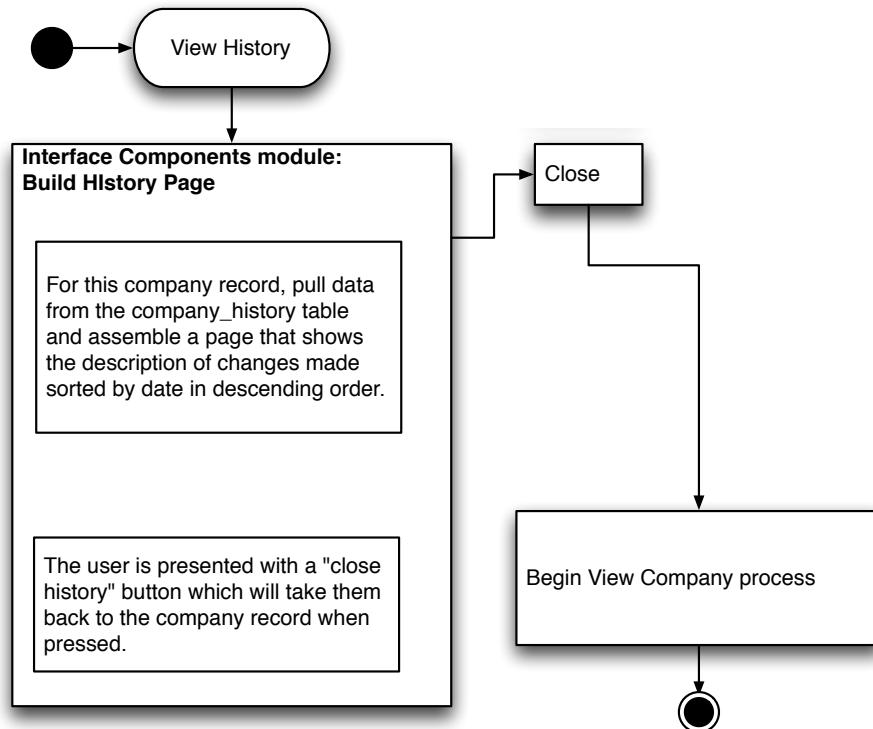


New Contract button – Creates a new contract record attached to this company and puts the user into the contract record edit screen.

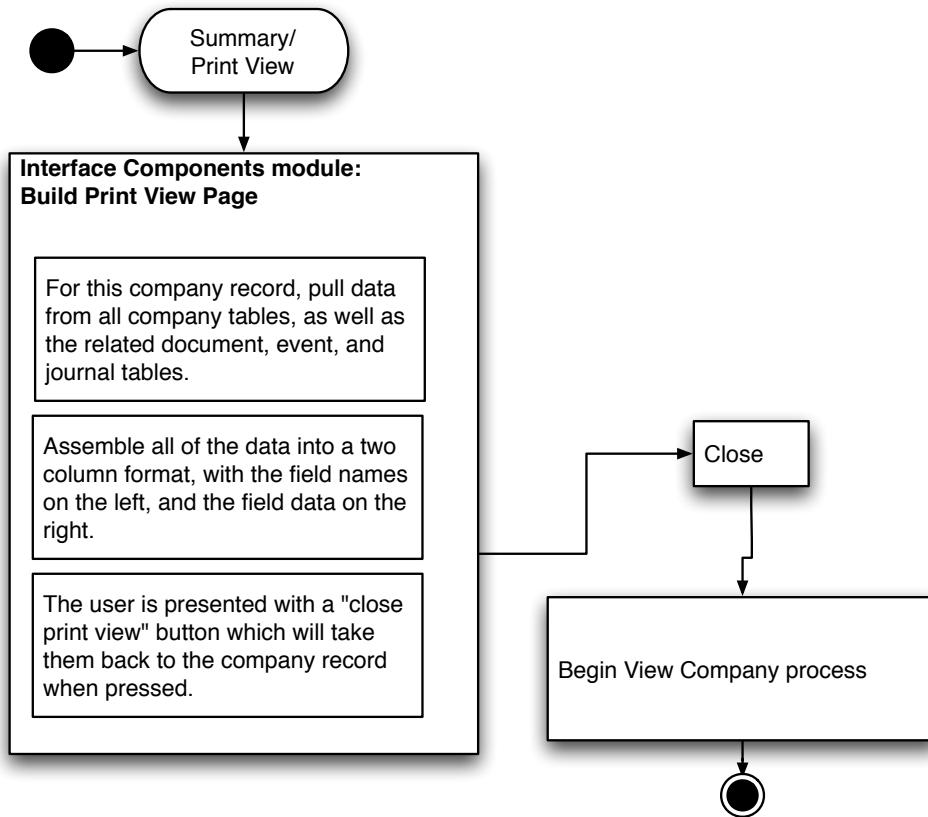
New Deal button – Puts the user into the workflow area to create a new contract through the workflow process.

Print button – Prints this screen of information using the browser's print functionality.

Record History button – Shows the user the history of all the changes made to the company record, who made the changes, and when they made the changes.

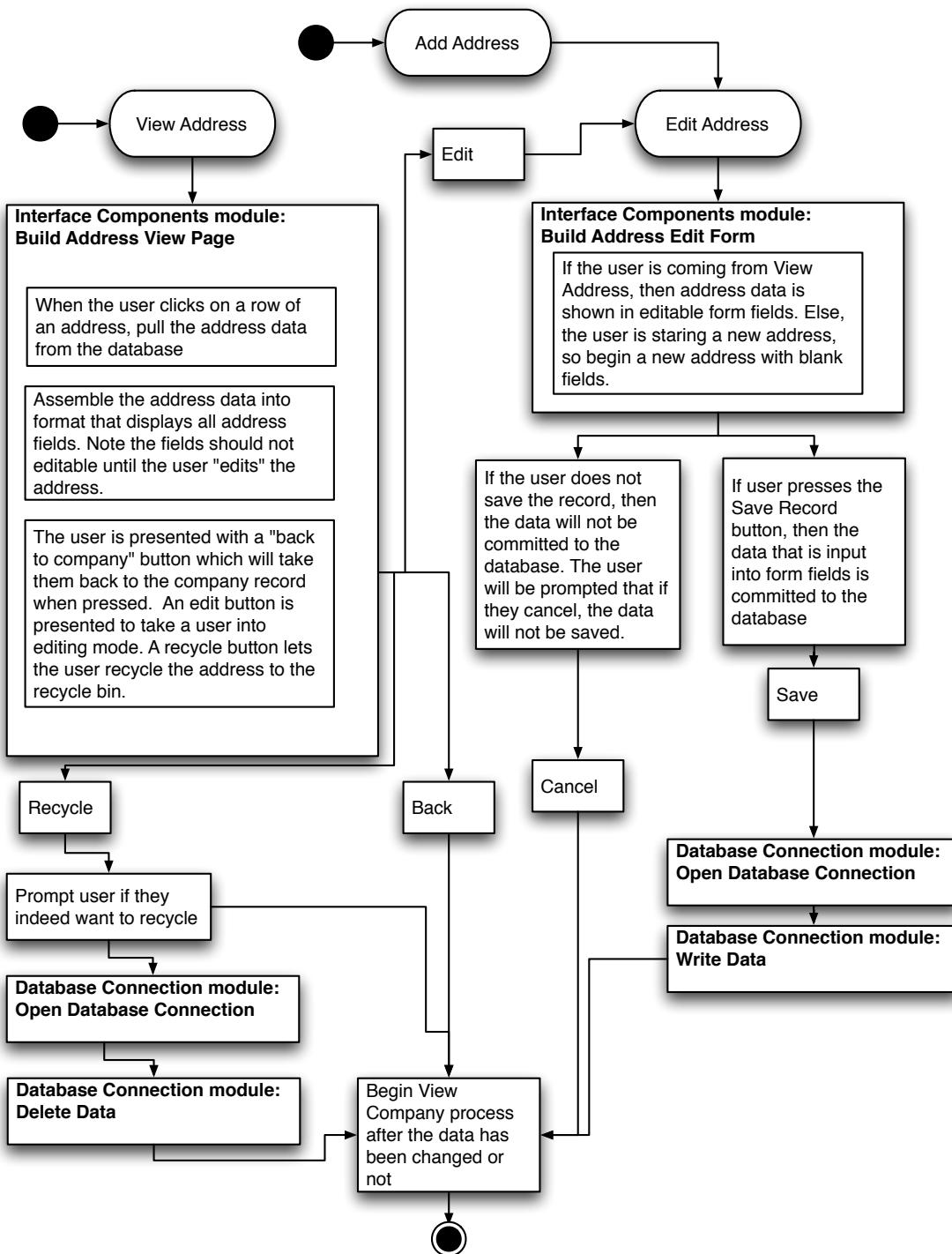


Summary – Shows the user a summary of all company data (from all tabs), along with the option to print all data, or a subset of data.



The Additional Names section shows other names that the company is known by, or was formerly known as. The user can add one by clicking on the blue plus sign to the right. View, Add, Edit, Recycle.

The Additional Addresses section shows other addresses relevant to the company. The user can view an address by click on the row for that address. The address can then be edited by pressing the Edit button in the address view. A new address can be added by pressing on the blue “plus sign” button, and the user is taken to a form for entering a new address. View, Add, Edit, Recycle



The Company Relationships section shows relationships to other internal and external entities/companies. The user can add one by clicking on the blue plus sign to the right. They will be prompted to search for a company to establish a relationship with, as well as to specify the relationship type. View, Add, Edit, Recycle

3.2.7 Reports Screen

The screenshot shows the CMS Reports screen. On the left is a sidebar with navigation links: User Home, Contracts, Workflow, Companies, Reports (with sub-links: New Search, Saved Searches, New Report, Saved Reports), Last Visited, and Admin. Below the sidebar are search fields: 'search...' and 'Go'. The main area has a title 'Contract Management System (CMS)' and a URL 'https://prod.reallybigwireless.com'. It features a search interface with a 'Search for Field' dropdown set to 'Contract Group'. A scrollable list of available fields includes Company Data, Contract Data, Contract Group (which is selected and highlighted in grey), Deal Data, and Admin Data. Below this are 'Run' and 'Save' buttons. To the right, there's a section for filtering search results with dropdowns for Companies, Company Status, Company Events, Contracts, Contract Status, Contract Events, and Provisions, all set to 'Pull-down Menu'. Below these filters is a 'Search Results' dropdown set to 'Contracts'. At the bottom is a search configuration table with columns: Table, Field, Sort, Option, Criteria, and Next Operand. The first row shows 'Contracts' as the Table, 'Contract Group' as the Field, 'None' as the Sort, 'is =' as the Option, 'Select' as the Criteria, and 'AND' as the Next Operand. There is also a small orange 'X' icon next to the 'Select' button.

The above screen mockup shows what a search might look like in the reports area. Search and report is a two-step process, in that (1) users build a search to narrow down to a set of company, contract, or provision records and then (2) they use a report to layout the records, choosing various fields to display.

Search

The “Search for Field” area allows the user to find a specific field they maybe looking for in building a search.

The box below the Search for Field shows the fields that are available for a search. Each main item on the menu represents a table of data, which when clicked on present the fields within that table.

At the top right are additional options for filtering search results. Each box allows the user to select active/recycled companies/contracts, active/inactive companies/contracts, open/closed companies/contract events, and active/recycled provisions.

The area in white below shows the fields that have been selected for query. The user can select additional ways to filter the data, such as sorting the data alphabetically, finding the records that fit (or don't fit) certain criteria, as well as Boolean logic rules.

The user can remove a field from the search by clicking on the orange "X" to the right.

Run button – Allows the user to execute the search.

Save button – Allows the user to save the search criteria.

Report

The report layout looks the same as the search layout, except that the verbiage indicates "report" instead of search, and there is slightly different functionality. The report functionality uses the field selection at the top to select what fields will be displayed on the report (whereas the search selects what rows of data to be pulled from the database).

Search & Report Process

When the user builds a search and saves it, they can then run the search through a saved report format in order to output certain fields for a given search rows.

3.2.8 Last Visited Screen

The screenshot shows the 'Contract Management System (CMS)' interface. On the left is a vertical navigation menu with links: User Home, Contracts, Workflow, Companies, Reports, Last Visited (which is highlighted in blue), and Admin. The main content area is titled 'Last Visited Contracts' and displays a table with two rows:

Contract Number	Company Name	Contract Purpose	Effective Date	Expiration Date
12345 23456	AB Company CDE LLC	Master Services Agmt Amendment #1	01/01/2009 03/24/2009	01/01/2014 03/24/2014

Below this is a section titled 'Last Visited Companies' with a table:

Company Name	Company ID	City	Country	Group
AB Company CDE LLC	54321 65432	San Diego Mexico City	United States Mexico	Trucking Primary

Next is a section titled 'Last Visited Deals' with a table:

Deal Number	Company Name	Contract Purpose	Effective Date	Expiration Date
55555 66666	ZX Company LRE LLC	Licensing Agreement Amendment #4	03/01/2009 06/15/2009	03/01/2014 06/15/2014

Below these sections are 'Search Favorites' and 'Report Favorites' sections, each listing one item:

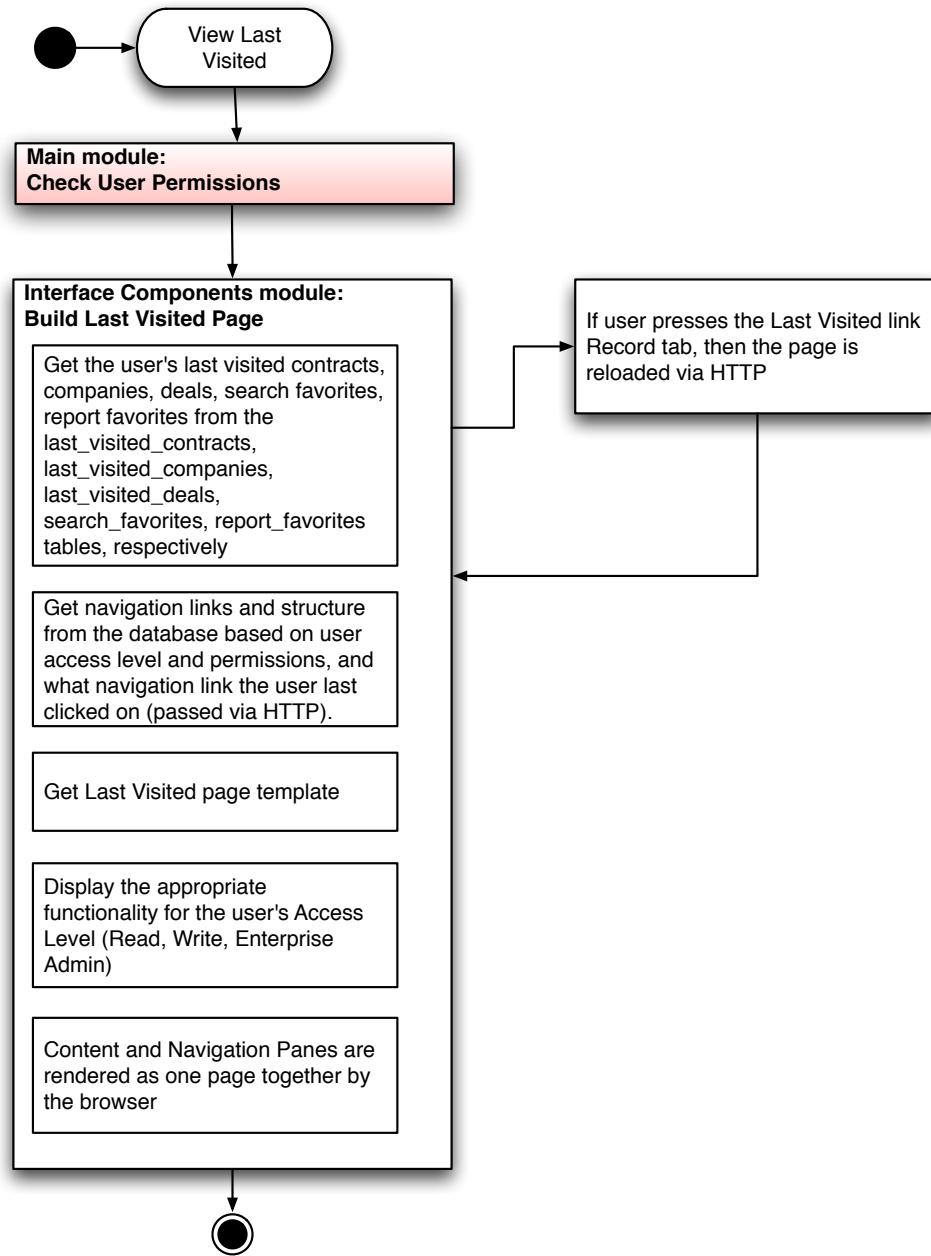
Search Favorites:
All Trucking Companies (Created 02/27/2009)
All Chips Contracts (Created 04/21/2009)

Report Favorites:
Simple Report (Created 02/01/2009)
Report of All Field (Created 03/21/2009)

At the bottom left are search and go buttons: and a link to 'User Guide'.

The above screen mockup shows what the screen looks like when the user clicks on the "Last Visited" link at left.

The first group presented is the last contracts visited. The next group is the last visited companies. Following these groups are last visited deals, favorite searches and favorite reports. The user can visit an item in one of the groups by clicking on a row.



3.2.9 Admin Screen

The screenshot shows the 'Contract Management System (CMS)' interface. At the top, there's a header bar with the title 'Contract Management System (CMS)' and a URL field containing 'https://prod.reallybigwireless.com'. Below the header is a navigation menu on the left with links: 'User Home', 'Contracts', 'Workflow', 'Companies', 'Reports', 'Last Visited', and 'Admin'. The 'Admin' link is highlighted. The main content area has tabs at the top: 'Users', 'Groups', 'Definable Fields', 'Settings', 'Workflow Admin', and 'Recycle Bin'. The 'Users' tab is selected. Below the tabs is a search bar with 'Filter options' and a dropdown, a search input field, a 'Go' button, and two blue buttons: 'Add User' and 'Advanced Edit'. A table follows, showing user data with columns: Login, First Name, Last Name, Access Level, Last Login, and Language. Two rows are visible: one for 'BLOBLAW' (Bob Loblaw) and another for 'MJONES' (Mike Jones). At the bottom of the main content area are buttons for 'search...', 'Go', and 'User Guide'. On the right side, there are page navigation controls: back, forward, and a page number indicator 'Page 1 of 1726'.

Login	First Name	Last Name	Access Level	Last Login	Language
BLOBLAW	Bob	Loblaw	Write User	07/06/2009 1:30p	English
MJONES	Mike	Jones	Read User	07/06/2009 8:30a	English

The above screen mockup shows what a user with Enterprise Administrator privileges sees after clicking on the "Admin" link in the menu at left.

At the top are various tabs for performing various admin functions, as described in the Use Cases.

Users tab – This tab lists the users sequentially, and presents the Enterprise Administrator with functionality to create and manage user accounts. They can add users with the "Add User" button. The "Advanced Edit" function provides for moving events from one user to another. The search box allows for filtering to find a specific user. To go to a user's account, the admin can click on the row representing the user. They will then be able to View, Edit, and Recycle a user.

Groups tab – This tab presents the Enterprise Administrator with the ability to add, remove and otherwise manage Contract Groups in the system. View, Edit, Add, Recycle.

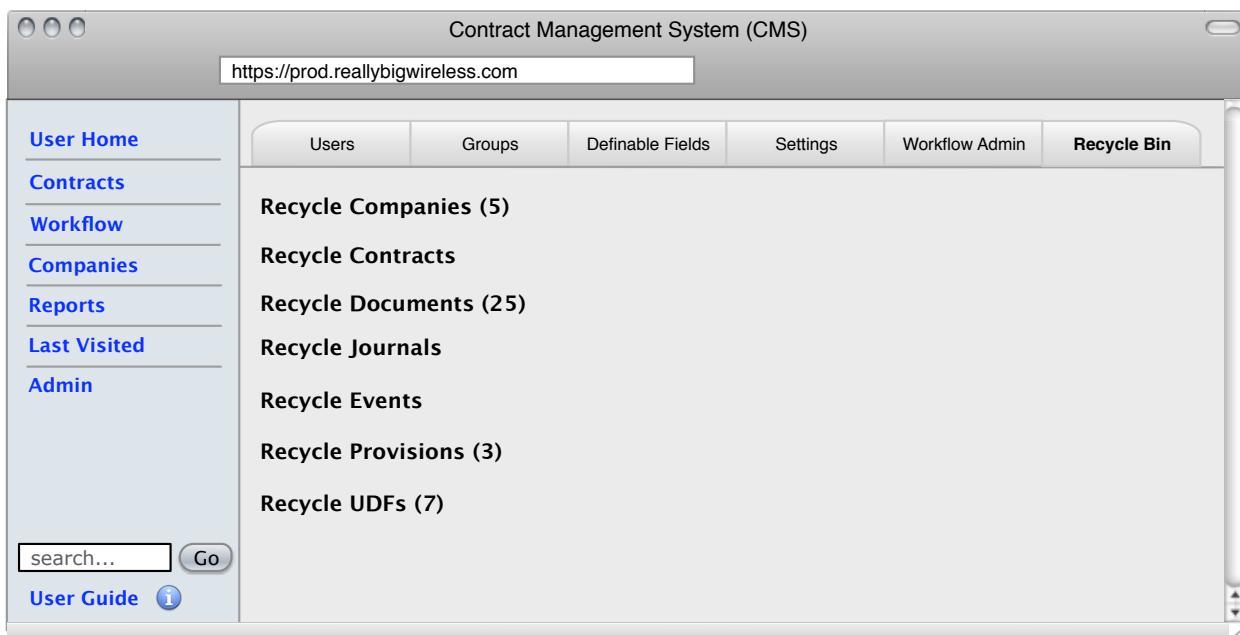
Definable Fields tab – This tab contains functionality for creating, naming, and managing user defined fields. View, Edit, Add, Recycle.

Settings tab – This tab is where the Enterprise Administrator would go to manage various settings in the application, such as how the application

behaves or looks in certain areas. View, Edit.

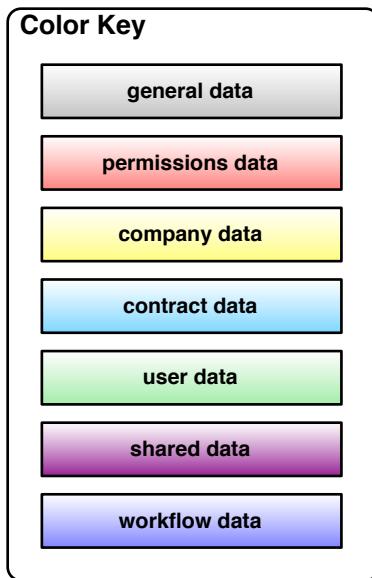
Workflow Admin tab – In this tab, there are features that allow the Enterprise Administrator to set up how the Workflow area performs. View, Edit, Add, Recycle.

Recycle Bin tab – This tab is where all items (contract, companies, provisions, etc.) that have been recycled will show up. The Enterprise Administrator then can review the items before they are deleted completely from the system. The below screen mockup shows the Recycle Bin tab. The Enterprise Administrator is given a link for each area of potential recycled records. They can see how many records are in the recycle bin for each area by the respective number in parenthesis. When they click on the “Recycle ...” link, they are taken to a page where the recycled records are listed sequentially with check boxes next to each one. They can delete the records from this page (permanently deleted from the system). They can check one or do a “Select All” to check all boxes, and then delete the checked items.



3.3 Data Structure Design

3.3.1 ERD – Color Key



As shown in the above color key, the data sources from the requirements document have been put into data groups in the entity relationship diagrams (ERD) shown in the subsequent sections. Each data group contains related tables that have been colored to hopefully make the ERD more readable.

General data – This group contains data that is “global” to the system.

Permissions data – This group can be seen in a number of areas on the ERD, and shows how “group level” permissions data will be stored. This data shows how users are given permission to work with various contracts if they belong to certain Contract Groups.

Company data – This group contains company data (both internal and external) to Really Big Wireless.

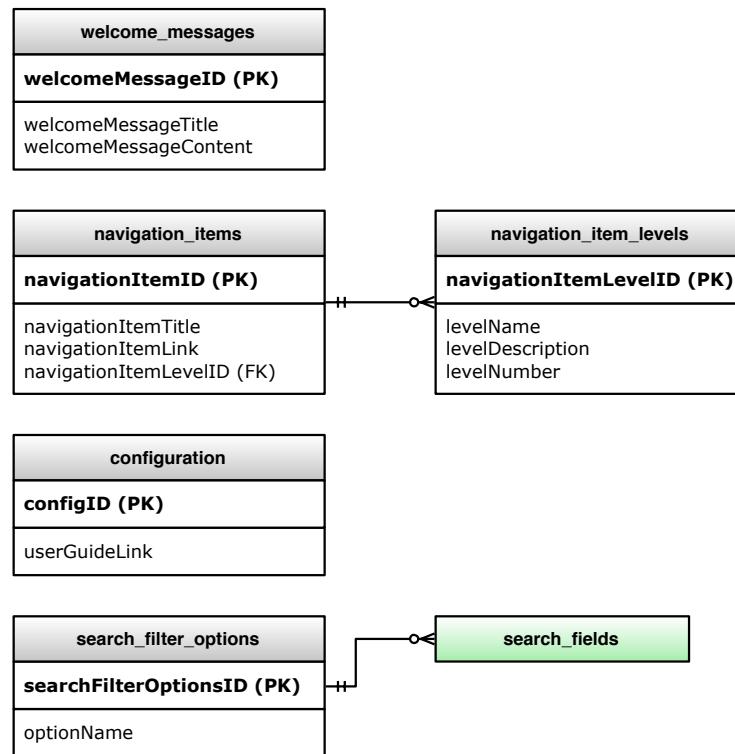
Contract data – This group holds contract data, as well as provision forms (extra fields) for contracts.

User data – This group holds data about user accounts, as well as saved searches and reports. A user’s “access level” is their overall level of access (Read, Write, Enterprise Admin), and trumps any access level they could have to a Contract Group.

Shared data – This group contains tables that are shared between contracts and companies. The foreign key from the contracts table or company table will be used respective of what kind of data is being stored (i.e. the foreign key from the contracts table would be used if the row represents contracts data).

Workflow data – This group represents the workflow data, or the underlying data set for creating a contract via the workflow process.

3.3.2 ERD – General Data



The above image shows the general data tables group. Each table is described below.

Table Name: welcome_messages

Description: This table represents the data for the welcome messages presented to the user when they login.

Field Name	Type	Description/Notes
welcomeMessageID (PK)	integer	primary key (auto-generated)
welcomeMessageTitle	string	the title of a welcome message

welcomeMessageContent	string	the content of a welcome message
-----------------------	--------	----------------------------------

Table Name: navigation_items

Description: This table represents the data for the navigation structure at the left side of the application. Each “navigation item” is a link that a user can click on to go to a certain part of the application.

Field Name	Type	Description/Notes
navigationItemID (PK)	integer	primary key (auto-generated)
navigationItemTitle	string	the name of the navigation item, i.e. the title that shows to the user
navigationItemLink	string	the link that the user follows in the application to a new area
navigationItemLevelID	integer	the level of a navigation item, i.e. there are several levels of hierarchy, giving an “outline” structure to the navigation menu

Table Name: navigation_item_levels

Description: This table contains the data to represent several levels of navigation hierarchy in the menu on the left side of the application.

Field Name	Type	Description/Notes
navigationItemLevelID (PK)	integer	primary key (auto-generated)
levelName	string	the name of the level of navigation
levelDescription	string	the description of the level of navigation
levelNumber	integer	the numeric representation of the level of navigation, i.e. level 1 is the highest level, level 2 the next highest, and so on

Table Name: configuration

Description: This table holds data pertaining to the configuration of the system.

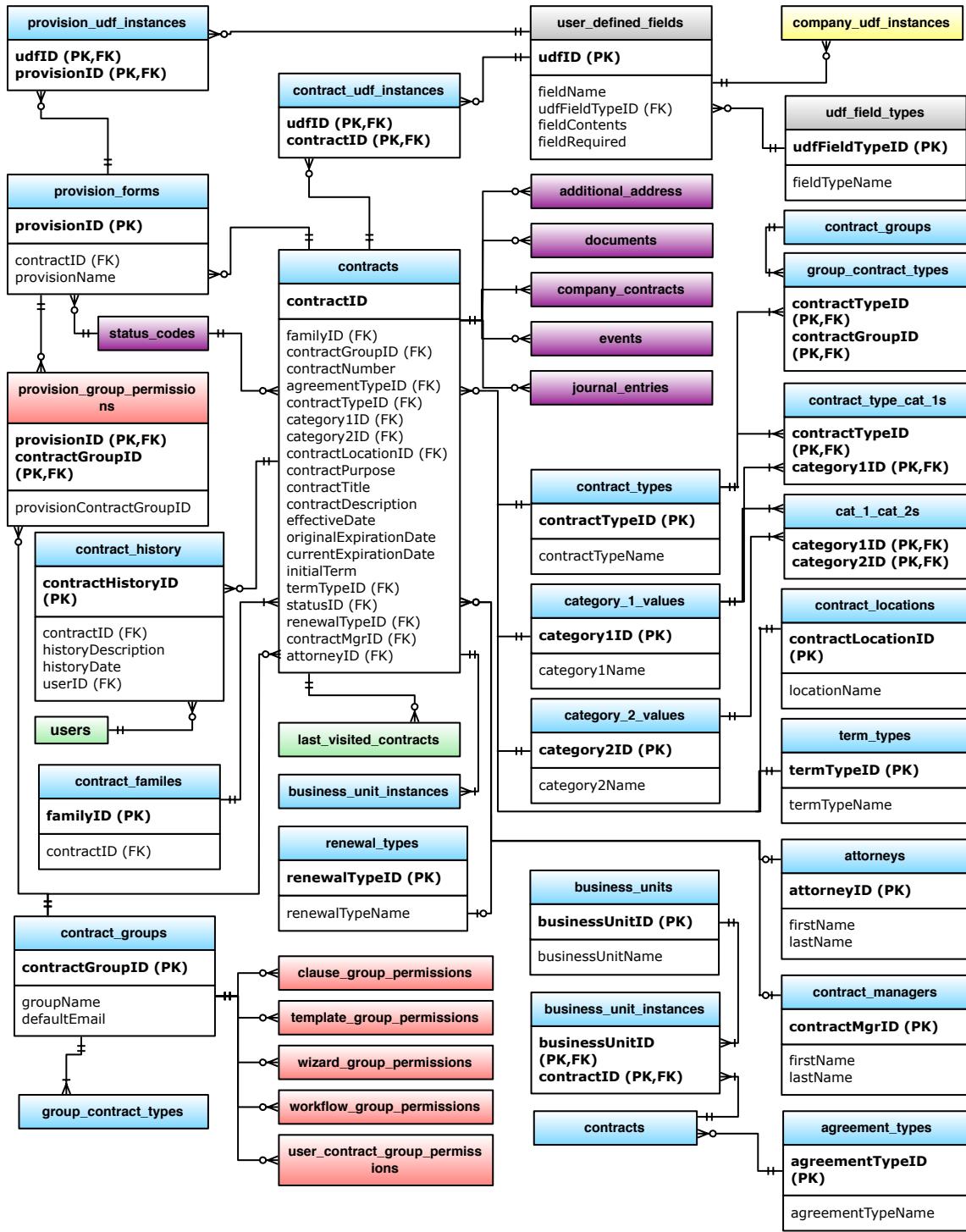
Field Name	Type	Description/Notes
configID (PK)	integer	primary key (auto-generated)
userGuideLink	string	this link takes the user to the user guide (a PDF file)

Table Name: search_filter_options

Description: This table holds data pertaining to options used in the advanced search filters.

Field Name	Type	Description/Notes
searchFilterOptionsID (PK)	integer	primary key (auto-generated)
optionName	string	this is the name of an option for advanced search filtering

3.3.3 ERD – Contracts Data



The above image shows the contract data tables, as well as stubs of related tables in other data groups. Each table is described below.

Table Name: contracts

Description: This table represents the data for contract records.

Field Name	Type	Description/Notes
contractID (PK)	integer	primary key (auto-generated)
familyID (FK)	integer	foreign key
contractGroupID (FK)	integer	foreign key
contractNumber	string	the number the business uses to track a contract, which could include numbers or characters
agreementTypeID (FK)	string	foreign key
contractTypeID (FK)	integer	foreign key
category1ID (FK)	integer	foreign key
category2ID (FK)	integer	foreign key
contractLocationID (FK)	integer	foreign key
contractPurpose	string	the purpose of the contract, usually a short statement about the contract
contractTitle	string	the exact title of the contract
contractDescription	text	a field where the user can input any text of additional information about the contract
effectiveDate	date	the date the contract is effective
originalExpirationDate	date	the date the contract originally expired
currentExpirationDate	date	the date the contract is currently set to expire
initialTerm	integer	the term (in months) of a contract
termTypeID (FK)	integer	foreign key
statusID (FK)	integer	foreign key
renewalTypeID (FK)	integer	foreign key

contractMgrID (FK)	integer	foreign key
attorneyID (FK)	integer	foreign key

Table Name: provision_udf_instances

Description: This table is a linking table that contains instances of a User Defined Field to be used on a provision form.

Field Name	Type	Description/Notes
udfID (PK,FK)	integer	primary key that is a foreign key from another table
provisionID (PK,FK)	integer	primary key that is a foreign key from another table

Table Name: provision_forms

Description: This table holds the provision forms.

Field Name	Type	Description/Notes
provisionID (PK)	integer	primary key (auto-generated)
contractID (FK)	integer	foreign key
provisionName	string	the name of the provision form

Table Name: provision_group_permissions

Description: This table holds the permissions for the provision forms, which is according to Contract Group.

Field Name	Type	Description/Notes
provisionID (PK,FK)	integer	primary key that is a foreign key from another table
contractGroupID (PK,FK)	integer	primary key that is a foreign key from another table
provisionContractGroupID	integer	auto-generated ID to give a number to this relationship

Table Name: contract_history

Description: This table contains the history of changes to a contract record.

Field Name	Type	Description/Notes
contractHistoryID (PK)	integer	primary key (auto-generated)
contractID (FK)	integer	foreign key
historyDescription	string	a description of what changes were made
historyDate	date	the date the changes were made
userID (FK)	integer	foreign key

Table Name: contract_groups

Description: This table holds the Contract Groups.

Field Name	Type	Description/Notes
contractGroupID (PK)	integer	primary key (auto-generated)
groupName	string	the name of the group
defaultEmail	string	the default email address that group communication should be sent to

Table Name: contract_udf_instances

Description: This table is a linking table that contains instances of a User Defined Field to be used on a contract record.

Field Name	Type	Description/Notes
udfID (PK,FK)	integer	primary key that is a foreign key from another table
contractID (PK,FK)	integer	primary key that is a foreign key from another table

Table Name: renewal_types

Description: This table holds the renewal types for a contract record (auto-renewal, manual-renewal).

Field Name	Type	Description/Notes
renewalTypeID (PK)	integer	primary key (auto-generated)
renewalTypeName	string	the name of the type of renewal

Table Name: user_defined_fields

Description: This table contains the user-defined fields (UDFs) that are customized fields that can be added to provision forms, contract records, or company records.

Field Name	Type	Description/Notes
udfID (PK)	integer	primary key (auto-generated)
fieldName	string	the name of the UDF
udfFieldTypeID (FK)	integer	foreign key
fieldContents	string	the contents of the UDF
fieldRequired	bool	whether or not the UDF is required to be filled out

Table Name: udf_field_types

Description: This table contains types of fields that a UDF can be.

Field Name	Type	Description/Notes
udfFieldTypeID (PK)	integer	primary key (auto-generated)
fieldTypeName	string	the name of the field type

Table Name: contract_types

Description: This table contains the types of contracts.

Field Name	Type	Description/Notes
contractTypeID (PK)	integer	primary key (auto-generated)
contractTypeName	string	the name of a Contract Type

Table Name: category_1_values

Description: This table contains the category 1 values for a contract.

Field Name	Type	Description/Notes
category1ID (PK)	integer	primary key (auto-generated)
category1Name	string	the name of a category 1

Table Name: category_2_values

Description: This table contains the category 2 values for a contract.

Field Name	Type	Description/Notes
category2ID (PK)	integer	primary key (auto-generated)
category2Name	string	the name of a category 2

Table Name: business_units

Description: This table contains the business units that can be associated with a contract.

Field Name	Type	Description/Notes
businessUnitID (PK)	integer	primary key (auto-generated)
businessUnitName	string	the name of a business unit

Table Name: business_unit_instances

Description: This table is a linking table that contains instances of a business unit for a contract record (i.e. multiple business units can be on a contract record).

Field Name	Type	Description/Notes
businessUnitID (PK,FK)	integer	primary key that is a foreign key from another table
contractID (PK,FK)	integer	primary key that is a foreign key from another table

Table Name: group_contract_types

Description: This table is a linking table that contains instances of a Contract Type for a Contract Group (i.e. when a Contract Group is selected, then a subset of Contract Type values can be selected).

Field Name	Type	Description/Notes
contractTypeID (PK,FK)	integer	primary key that is a foreign key from another table
contractGroupID (PK,FK)	integer	primary key that is a foreign key from another table

Table Name: contract_type_cat_1s

Description: This table is a linking table that contains instances of a Category 1 for a Contract Type (i.e. when a Contract Type is selected, then a subset of Category 1 values can be selected).

Field Name	Type	Description/Notes
contractTypeID (PK,FK)	integer	primary key that is a foreign key from another table
category1ID (PK,FK)	integer	primary key that is a foreign key

		from another table
--	--	--------------------

Table Name: contract_type_cat_2s

Description: This table is a linking table that contains instances of a Category 2 for a Category 1 value (i.e. when a Category 1 is selected, then a subset of Category 2 values can be selected).

Field Name	Type	Description/Notes
category1ID (PK,FK)	integer	primary key that is a foreign key from another table
category2ID (PK,FK)	integer	primary key that is a foreign key from another table

Table Name: contract_locations

Description: This table contains the locations that paper copy of a contract can be stored.

Field Name	Type	Description/Notes
contractLocationID (PK)	integer	primary key (auto-generated)
locationName	string	the name of a contract location

Table Name: term_types

Description: This table contains the term types for a contract (i.e. perpetual, fixed).

Field Name	Type	Description/Notes
termTypeID (PK)	integer	primary key (auto-generated)
termTypeName	string	the name of a term type

Table Name: attorneys

Description: This table contains the attorneys that can be associated to a contract record.

Field Name	Type	Description/Notes
attorneyID (PK)	integer	primary key (auto-generated)
firstName	string	the first name of an attorney
lastName	string	the last name of an attorney

Table Name: contract_managers

Description: This table contains the contract managers that can be associated to a contract record.

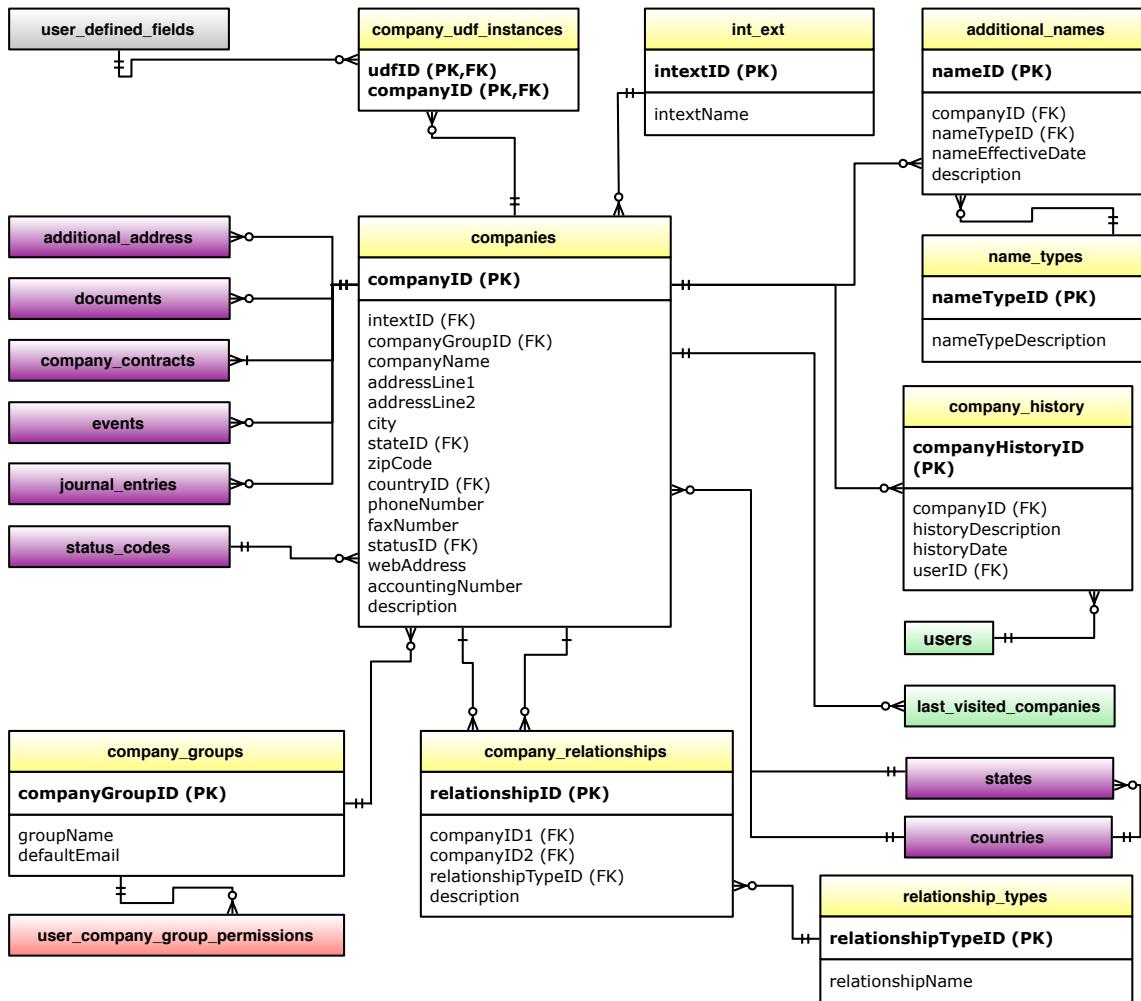
Field Name	Type	Description/Notes
contractMgrID (PK)	integer	primary key (auto-generated)
firstName	string	the first name of a contract manager
lastName	string	the last name of a contract manager

Table Name: agreement_types

Description: This table contains the type of agreement that a contract can be. These values are more generic than Contract Type. For example, Agreement Type values are Purchase Order, Master, Amendment, etc.

Field Name	Type	Description/Notes
agreementTypeID (PK)	integer	primary key (auto-generated)
agreementTypeName	string	the name of an agreement type

3.3.4 ERD – Company Data



The above image shows the company data tables, as well as stubs of related tables in other data groups. Each table is described below.

Table Name: companies

Description: This table represents the data for contract records.

Field Name	Type	Description/Notes
companyID (PK)	integer	primary key (auto-generated)
intextID (FK)	integer	foreign key
companyGroupID (FK)	integer	foreign key
companyName	string	the name of a company
addressLine1	string	the first line of an address for a company

addressLine2	string	the second line of an address for a company
city	string	the city of the company address
stateID (FK)	integer	foreign key
zipCode	string	the zip code for a company address
countryID (FK)	integer	foreign key
phoneNumber	string	the phone number of a company
faxNumber	string	the fax number of a company
statusID (FK)	integer	foreign key
webAddress	string	the internet address for a company's website
accountingNumber	string	the accounting number or identification number for the company in another system
description	text	an open text field for entering other information about a company

Table Name: company_udf_instances

Description: This table is a linking table that contains instances of a User Defined Field to be used on a company record.

Field Name	Type	Description/Notes
udfID (PK,FK)	integer	primary key that is a foreign key from another table
companyID (PK,FK)	integer	primary key that is a foreign key from another table

Table Name: company_groups

Description: This table contains the groups of companies.

Field Name	Type	Description/Notes
companyGroupID (PK)	integer	primary key (auto-generated)
groupName	string	the name of a company group
defaultEmail	string	the default email address for communication relating to a group

		of companies
--	--	--------------

Table Name: company_relationships

Description: This table contains the relationships between two companies.

Field Name	Type	Description/Notes
relationshipID (PK)	integer	primary key (auto-generated)
companyID1 (FK)	integer	foreign key
companyID2 (FK)	integer	foreign key
relationshipTypeID (FK)	integer	foreign key
description	string	the description of the relationship between two companies

Table Name: relationship_types

Description: This table contains the types of relationships between companies.

Field Name	Type	Description/Notes
relationshipTypeID (PK)	integer	primary key (auto-generated)
relationshipName	string	a name of the relationship type

Table Name: int_ext

Description: This table contains the relationship of a company to Really Big Wireless (i.e. internal or external).

Field Name	Type	Description/Notes
intextID (PK)	integer	primary key (auto-generated)
intextName	string	internal or external

Table Name: additional_names

Description: This table contains additional names for a company (i.e. "Also Known As", "Formerly Known As", etc.)

Field Name	Type	Description/Notes
nameID (PK)	integer	primary key (auto-generated)
companyID (FK)	integer	foreign key
nameTypeID (FK)	string	foreign key
nameEffectiveDate	date	the effective date for the additional name
description	text	extra information about an

		additional name
--	--	-----------------

Table Name: name_types

Description: This table contains the types of additional names for a company.

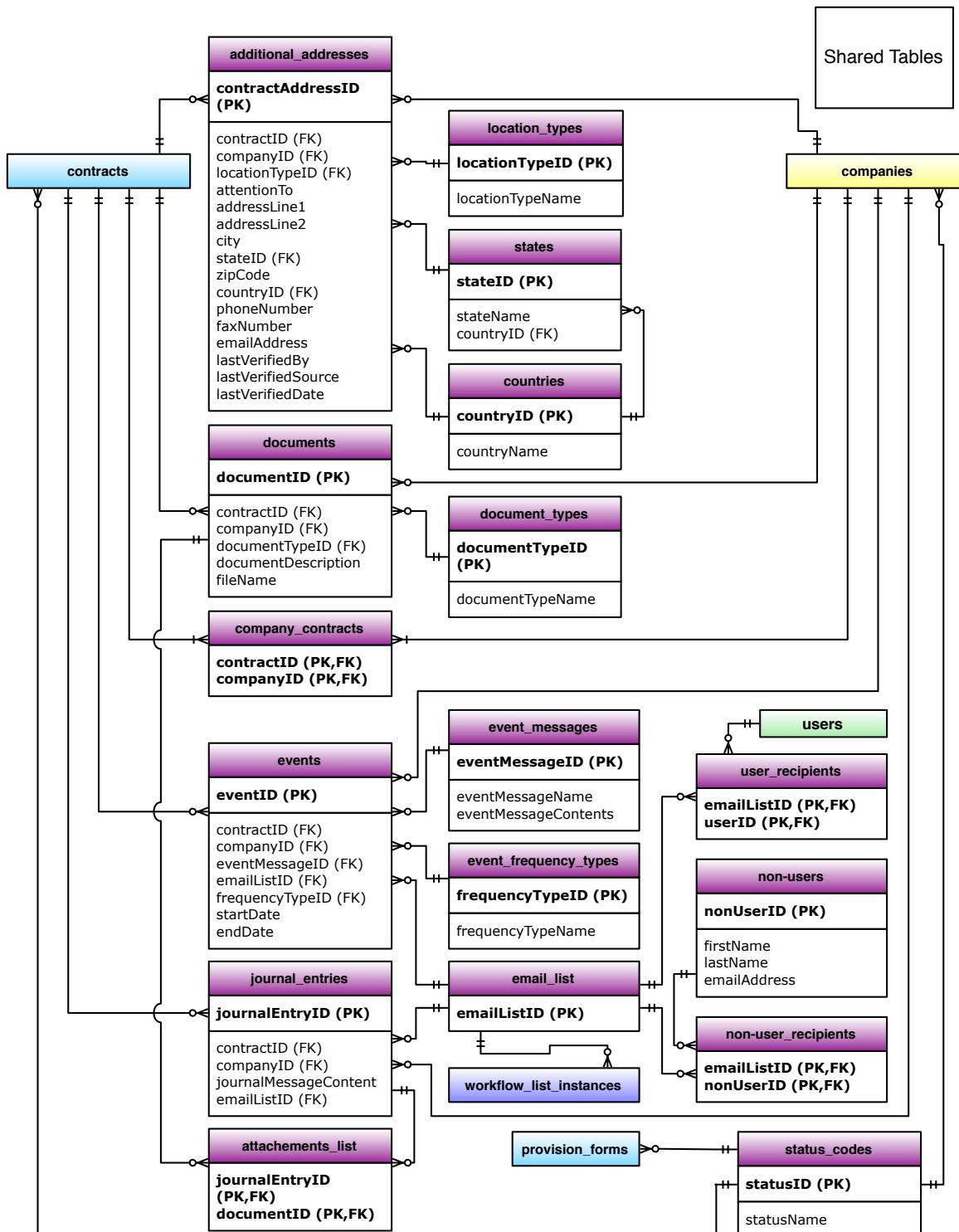
Field Name	Type	Description/Notes
nameTypeID (PK)	integer	primary key (auto-generated)
nameTypeDescription	string	a name of the name type

Table Name: company_history

Description: This table contains the changes that were made to company records.

Field Name	Type	Description/Notes
companyHistoryID (PK)	integer	primary key (auto-generated)
companyID (FK)	integer	foreign key
historyDescription	text	the description of the changes that were made to a company record
historyDate	date	the date the changes were made
userID (FK)	integer	foreign key

3.3.5 ERD – Shared Data



The above image shows the shared data tables, as well as stubs of related tables in other data groups. Each table is described below.

Table Name: additional_addresses

Description: This table contains additional addresses for a company or contract.

Field Name	Type	Description/Notes
contractAddressID (PK)	integer	primary key (auto-generated)
contractID (FK)	integer	foreign key
companyID (FK)	integer	foreign key
locationTypeID (FK)	integer	foreign key
attentionTo	string	
addressLine1	string	
addressLine2	string	
city	string	
stateID (FK)	integer	foreign key
zipCode	string	
countryID (FK)	integer	foreign key
phoneNumber	string	
faxNumber	string	
emailAddress	string	
lastVerifiedBy	string	the person that verified the address
lastVerifiedSource	string	where the person found the address
lastVerifiedDate	date	the date that the person verified the address

Table Name: location_types

Description: This table contains the location types for additional addresses.

Field Name	Type	Description/Notes
locationTypeID (PK)	integer	primary key (auto-generated)
locationTypeName	string	the name of a location type

Table Name: states

Description: This table contains states for a given country.

Field Name	Type	Description/Notes
stateID (PK)	integer	primary key (auto-generated)
stateName	string	the name of a state
countryID (FK)	integer	foreign key

Table Name: countries

Description: This table contains countries for addresses.

Field Name	Type	Description/Notes
countryID (PK)	integer	primary key (auto-generated)
countryName	string	the name of the country

Table Name: documents

Description: This table contains uploaded electronic documents for contracts or companies.

Field Name	Type	Description/Notes
documentID (PK)	integer	primary key (auto-generated)
contractID (FK)	integer	foreign key
companyID (FK)	integer	foreign key
documentTypeID (FK)	integer	foreign key
documentDescription	text	the description of the electronic document
fileName	string	the name of the file of the electronic document

Table Name: document_types

Description: This table contains the list of types of electronic documents that can be uploaded to the system.

Field Name	Type	Description/Notes
documentTypeID (PK)	integer	primary key (auto-generated)
documentTypeName	integer	the name of a document type

Table Name: company_contracts

Description: This table is a linking table that contains the instances of contracts for a company (and vice-versa).

Field Name	Type	Description/Notes
contractID (PK,FK)	integer	primary key that is a foreign key from another table
companyID (PK,FK)	integer	primary key that is a foreign key from another table

Table Name: events

Description: This table contains the "event alerts" or scheduled email

messages to be sent out for a company or contract.

Field Name	Type	Description/Notes
eventID (PK)	integer	primary key (auto-generated)
contractID (FK)	integer	foreign key
companyID (FK)	integer	foreign key
eventMessageID (FK)	integer	foreign key
emailListID (FK)	integer	foreign key
frequencyTypeID (FK)	integer	foreign key
startDate	date	the start date of a recurring event alert
endDate	date	the end date of a recurring event alert

Table Name: event_messages

Description: This table contains the messages that can be contained within an event alert.

Field Name	Type	Description/Notes
eventMessageID (PK)	integer	primary key (auto-generated)
eventMessageName	string	the name of the message
eventMessageContents	text	the contents of the message

Table Name: event_frequency_types

Description: This table contains the types of frequencies for an event alert (i.e. single, weekly, monthly, yearly).

Field Name	Type	Description/Notes
frequencyTypeID (PK)	integer	primary key (auto-generated)
frequencyTypeName	string	the name of the frequency type

Table Name: email_list

Description: This table contains the email lists for event alerts, journals, and workflow steps.

Field Name	Type	Description/Notes
eventMessageID (PK)	integer	primary key (auto-generated)

Table Name: user_recipients

Description: This table is a linking table for users contained on an email list.

Field Name	Type	Description/Notes
emailListID (PK,FK)	integer	primary key that is a foreign key from another table
userID (PK,FK)	integer	primary key that is a foreign key from another table

Table Name: non-users

Description: This table contains people that are not users in the system, but may need to be recipients of event alerts, journals, and workflow steps.

Field Name	Type	Description/Notes
nonUserID (PK)	integer	primary key (auto-generated)
firstName	string	
lastName	string	
emailAddress	string	

Table Name: non-user_recipients

Description: This table is a linking table for non-users contained on an email list.

Field Name	Type	Description/Notes
emailListID (PK,FK)	integer	primary key that is a foreign key from another table
nonUserID (PK,FK)	integer	primary key that is a foreign key from another table

Table Name: attachments_list

Description: This table is a linking table for electronic documents sent with a journal.

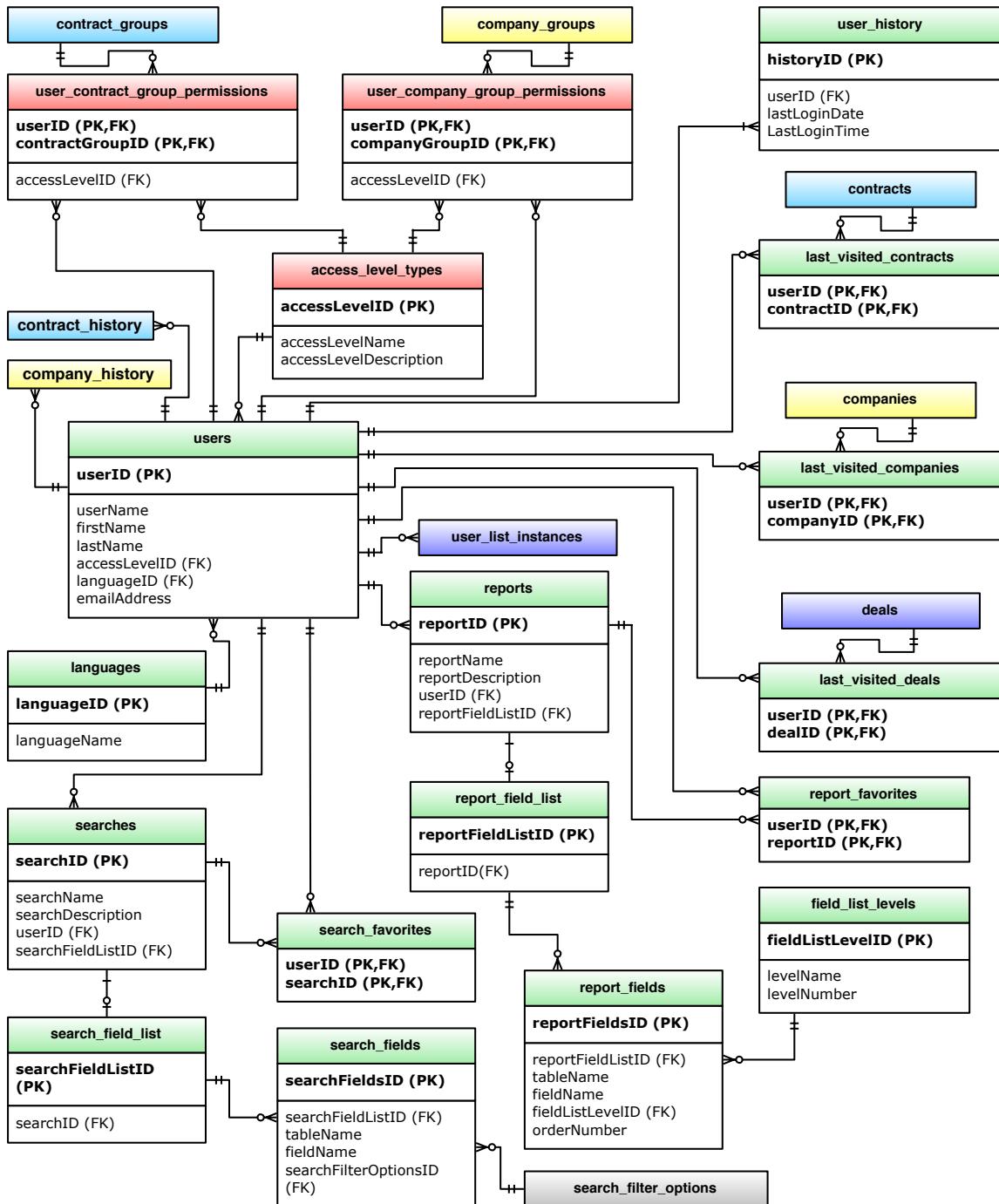
Field Name	Type	Description/Notes
journalEntryID (PK,FK)	integer	primary key that is a foreign key from another table
documentID (PK,FK)	integer	primary key that is a foreign key from another table

Table Name: status_codes

Description: This table contains the status for contracts, companies, or provisions (i.e. active, inactive, recycled).

Field Name	Type	Description/Notes
statusID (PK)	integer	primary key (auto-generated)
statusName	string	the name of the status

3.3.6 ERD – User Data



The above image shows the user data tables, as well as stubs of related tables in other data groups. Each table is described below.

Table Name: **users**

Description: This table contains the users of the system. Their overall access

level (Read, Write, or Enterprise Administrator) is indicated by the accessLevelID. Note that there is no place for passwords in the system, because the passwords are maintained by an external system.

Field Name	Type	Description/Notes
userID (PK)	integer	primary key (auto-generated)
userName	string	
firstName	string	
lastName	string	
accessLevelID (FK)	integer	foreign key
languageID (FK)	integer	foreign key
emailAddress	string	

Table Name: user_contract_group_permissions

Description: This table is a linking table that contains Contract Groups that user has access to, as well as the level of access they have to each Contract Group.

Field Name	Type	Description/Notes
userID (PK,FK)	integer	primary key that is a foreign key from another table
contractGroupID (PK,FK)	integer	primary key that is a foreign key from another table
accessLevelID (FK)	integer	foreign key

Table Name: user_company_group_permissions

Description: This table is a linking table that contains Company Groups that user has access to, as well as the level of access they have to each Company Group.

Field Name	Type	Description/Notes
userID (PK,FK)	integer	primary key that is a foreign key from another table
companyGroupID (PK,FK)	integer	primary key that is a foreign key from another table
accessLevelID (FK)	integer	foreign key

Table Name: access_level_types

Description: This table contains the levels of access for a user overall, and for an individual Contract or Company Group (Read, Write, or Enterprise Administrator).

Field Name	Type	Description/Notes
accessLevelID (PK)	integer	primary key (auto-generated)
accessLevelName	string	the name of the access level
accessLevelDescription	string	the description of the access level

Table Name: user_history

Description: This table contains the login history of users.

Field Name	Type	Description/Notes
historyID (PK)	integer	primary key (auto-generated)
userID (FK)	integer	foreign key
lastLoginDate	date	the date of the last user login
LastLoginTime	time	the time of the last user login

Table Name: languages

Description: This table contains the languages that users may want their system interface set to.

Field Name	Type	Description/Notes
languageID (PK)	integer	primary key (auto-generated)
languageName	string	the name of the language

Table Name: searches

Description: This table contains the saved searches for a user.

Field Name	Type	Description/Notes
searchID (PK)	integer	primary key (auto-generated)
searchName	string	the name of the search
searchDescription	string	the description of the search
userID (FK)	integer	foreign key
searchFieldListID (FK)	integer	foreign key

Table Name: search_field_list

Description: This table contains the “header” for a list of fields on a search.

Field Name	Type	Description/Notes
searchFieldListID (PK)	integer	primary key (auto-generated)
searchID (FK)	integer	foreign key

Table Name: search_fields

Description: This table contains the list of fields on a search.

Field Name	Type	Description/Notes
searchFieldsID (PK)	integer	primary key (auto-generated)
searchFieldListID (FK)	integer	foreign key
tableName	string	the name of the table where the field comes from
fieldName	string	the name of the field for the search
searchFilterOptionsID (FK)	integer	foreign key

Table Name: reports

Description: This table contains the saved reports for a user.

Field Name	Type	Description/Notes
reportID (PK)	integer	primary key (auto-generated)
reportName	string	the name of the report
reportDescription	string	the description of the report
userID (FK)	integer	foreign key
reportFieldListID (FK)	integer	foreign key

Table Name: report_field_list

Description: This table contains the “header” for a list of fields on a report.

Field Name	Type	Description/Notes
reportFieldListID (PK)	integer	primary key (auto-generated)
reportID(FK)	integer	foreign key

Table Name: report_fields

Description: This table contains the list of fields on a report.

Field Name	Type	Description/Notes
reportFieldsID (PK)	integer	primary key (auto-generated)
reportFieldListID (FK)	integer	foreign key
tableName	string	the name of the table where the field comes from
fieldName	string	the name of the field for the report
fieldListLevelID (FK)	integer	foreign key

orderNumber	integer	the order that the field will show up on a report section
-------------	---------	---

Table Name: field_list_levels

Description: This table contains the levels of fields on a report, i.e. the sections (or hierarchy) of fields.

Field Name	Type	Description/Notes
fieldListLevelID (PK)	integer	primary key (auto-generated)
levelName	string	name of the level of fields on a section of a report
levelNumber	integer	the number of a level on a report, effectively giving the report different sections of fields

Table Name: last_visited_contracts

Description: This table is a linking table that contains the last visited contract records for a given user.

Field Name	Type	Description/Notes
userID (PK,FK)	integer	primary key that is a foreign key from another table
contractID (PK,FK)	integer	primary key that is a foreign key from another table

Table Name: last_visited_companies

Description: This table is a linking table that contains the last visited company records for a given user.

Field Name	Type	Description/Notes
userID (PK,FK)	integer	primary key that is a foreign key from another table
companyID (PK,FK)	integer	primary key that is a foreign key from another table

Table Name: last_visited_deals

Description: This table is a linking table that contains the last visited deals (in the workflow section) for a given user.

Field Name	Type	Description/Notes
------------	------	-------------------

userID (PK,FK)	integer	primary key that is a foreign key from another table
dealID (PK,FK)	integer	primary key that is a foreign key from another table

Table Name: reportFavorites

Description: This table is a linking table that contains the “favorites” list of reports for a user.

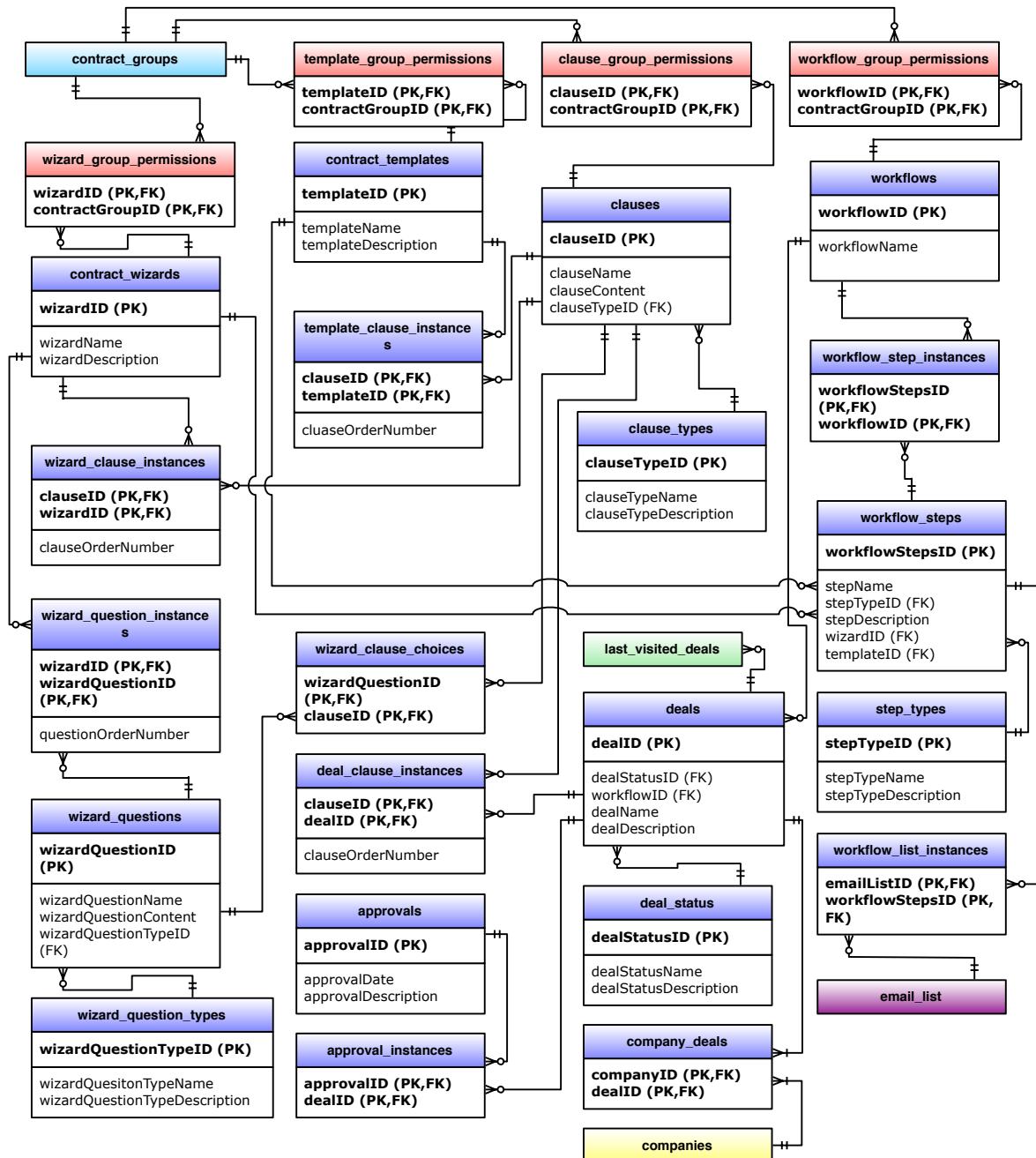
Field Name	Type	Description/Notes
userID (PK,FK)	integer	primary key that is a foreign key from another table
reportID (PK,FK)	integer	primary key that is a foreign key from another table

Table Name: searchFavorites

Description: This table is a linking table that contains the “favorites” list of searches for a user.

Field Name	Type	Description/Notes
userID (PK,FK)	integer	primary key that is a foreign key from another table
searchID (PK,FK)	integer	primary key that is a foreign key from another table

3.3.7 ERD – Workflow Data



The above image shows the workflow data tables, as well as stubs of related tables in other data groups. Each table is described below.

Table Name: workflows

Description: This table contains the Workflows. A Workflow is the overarching process a contract takes from conception through final

signature. During this process a contract is referred to as a “deal”. The outline for this process encompasses a number of “Wizard steps” as follows:

- Build – the contract/deal is built through one of three ways:
 - Wizard – A series of questions the user answers. The answers to the questions will pull in various relevant contract clauses.
 - Template – A complete “boilerplate” contract with the clauses already selected.
 - Clauses – The atomic pieces of a contract, i.e. the sections that make up a contract.
- Route – The contract is “routed” around to various people
 - for Approval – the contract/deal is sent to people who will approve the language, dollar values, etc.
 - for Signature – the contract is sent to external and internal people for signature (i.e. the contract is “executed”).
- Approve – During approval, the contract/deal can either be accepted or rejected. A rejection will cause the contract/deal to enter a prior step again.
 - Accept
 - Reject

Field Name	Type	Description/Notes
workflowID (PK)	integer	primary key (auto-generated)
workflowName	string	name of the workflow

Table Name: workflow_steps

Description: This table contains the steps specified in a Workflow process. Note that a step may include the use of a Wizard or Template. Other steps include pulling in Clauses (without using a Wizard or Template), as well as the other steps mentioned in the outline for a Workflow above.

Field Name	Type	Description/Notes
workflowStepsID (PK)	integer	primary key (auto-generated)
stepName	string	the name of a workflow step
stepTypeID (FK)	integer	foreign key
stepDescription	string	a description of a workflow step
wizardID (FK)	integer	foreign key
templateID (FK)	integer	foreign key

Table Name: workflow_step_instances

Description: This table is a linking table that contains the actual instances of steps in a Workflow.

Field Name	Type	Description/Notes
workflowStepsID (PK,FK)	integer	primary key that is a foreign key from another table
workflowID (PK,FK)	integer	primary key that is a foreign key from another table

Table Name: step_types

Description: This table contains the types of steps for a Workflow as mentioned above (build, route, approve steps).

Field Name	Type	Description/Notes
stepTypeID (PK)	integer	primary key (auto-generated)
stepTypeName	string	the name of a step type
stepTypeDescription	string	the description of a step type

Table Name: workflow_list_instances

Description: This table is a linking table that contains the email lists for a Workflow (i.e. the list of people that a contract/deal will be routed to in a Workflow step).

Field Name	Type	Description/Notes
emailListID (PK,FK)	integer	primary key that is a foreign key from another table
workflowStepsID (PK, FK)	integer	primary key that is a foreign key from another table

Table Name: clauses

Description: This table contains the Clauses (i.e. sections) of prepackaged language/verbiage for a contract/deal.

Field Name	Type	Description/Notes
clauseID (PK)	integer	primary key (auto-generated)
clauseName	string	the name of a clause
clauseContent	text	the content of a clause
clauseTypeID (FK)	integer	foreign key

Table Name: clause_types

Description: This table contains the types of Clauses in the system (such as Confidentiality, Indemnity, Notices, etc.).

Field Name	Type	Description/Notes
clauseTypeID (PK)	integer	primary key (auto-generated)
clauseTypeName	string	the name of a clause type
clauseTypeDescription	text	the description of a clause type

Table Name: deals

Description: This table contains the deals (contracts) that are going through the Workflow process.

Field Name	Type	Description/Notes
dealID (PK)	integer	primary key (auto-generated)
dealStatusID (FK)	integer	foreign key
workflowID (FK)	integer	foreign key

Table Name: deal_status

Description: This table contains the deals (contracts) that are going through the Workflow process.

Field Name	Type	Description/Notes
dealStatusID (PK)	integer	primary key (auto-generated)
dealStatusName	string	the name of the deal
dealStatusDescription	text	a longer description of the deal

Table Name: company_deals

Description: This table is a linking table that contains the company (or companies) associated with a contract/deal.

Field Name	Type	Description/Notes
companyID (PK,FK)	integer	primary key that is a foreign key from another table
dealID (PK,FK)	integer	primary key that is a foreign key from another table

Table Name: approval_instances

Description: This table is a linking table that contains the approvals of a contract/deal for various Workflow Steps.

Field Name	Type	Description/Notes

approvalID (PK,FK)	integer	primary key that is a foreign key from another table
dealID (PK,FK)	integer	primary key that is a foreign key from another table

Table Name: approvals

Description: This table contains the approval history for a contract/deal.

Field Name	Type	Description/Notes
approvalID (PK)	integer	primary key (auto-generated)
approvalDate	date	the date of an approval
approvalDescription	text	the description of an approval

Table Name: deal_clause_instances

Description: This table is a linking table that contains the clauses that are put into a contract/deal.

Field Name	Type	Description/Notes
clauseID (PK,FK)	integer	primary key that is a foreign key from another table
dealID (PK,FK)	integer	primary key that is a foreign key from another table
clauseOrderNumber	integer	the order the clause comes in the deal

Table Name: contract_wizards

Description: This table contains the Wizards (a Wizard is a set of questions that walk the user through creating a contract/deal).

Field Name	Type	Description/Notes
wizardID (PK)	integer	primary key (auto-generated)
wizardName	string	the name of the Wizard
wizardDescription	text	the description of the Wizard

Table Name: wizard_clause_instances

Description: This table contains the Clauses in a Wizard. These Clauses are decided on and added to this table after the user answers questions in the Wizard.

Field Name	Type	Description/Notes

clauseID (PK,FK)	integer	primary key that is a foreign key from another table
wizardID (PK,FK)	integer	primary key that is a foreign key from another table
clauseOrderNumber	integer	the order that the Clause comes for answered questions in a Wizard

Table Name: wizard_question_instances

Description: This table is a linking table that contains the Questions in a Wizard.

Field Name	Type	Description/Notes
wizardID (PK,FK)	integer	primary key that is a foreign key from another table
wizardQuestionID (PK,FK)	integer	primary key that is a foreign key from another table
questionOrderNumber	integer	the order that the question comes in the Wizard

Table Name: wizard_questions

Description: This table contains the questions that may be in a Wizard.

Field Name	Type	Description/Notes
wizardQuestionID (PK)	integer	primary key (auto-generated)
wizardQuestionName	string	the name of a question
wizardQuestionContent	text	the content of a question
wizardQuestionTypeID (FK)	integer	foreign key

Table Name: wizard_question_types

Description: This table contains the types of questions that may be in a Wizard.

Field Name	Type	Description/Notes
wizardQuestionTypeID (PK)	integer	primary key (auto-generated)
wizardQuestionTypeName	string	the name of a question type
wizardQuestionTypeDescription	text	the description of a question type

Table Name: wizard_clause_choices

Description: This table is a linking table that contains the Clauses that maybe linked to a question in a Wizard. Based on the user's answer to a question, one of among several Clauses will be picked as a Clause to use.

Field Name	Type	Description/Notes
wizardQuestionID (PK,FK)	integer	primary key that is a foreign key from another table
clauseID (PK,FK)	integer	primary key that is a foreign key from another table

Table Name: contract_templates

Description: This table contains the contract/deal Templates (a Template is an already-decided collection of Clauses).

Field Name	Type	Description/Notes
templateID (PK)	integer	primary key (auto-generated)
templateName	string	the name of a template
templateDescription	text	the description of a template

Table Name: template_clause_instances

Description: This table is a linking table that contains the Clauses that maybe linked to a Template.

Field Name	Type	Description/Notes
clauseID (PK,FK)	integer	primary key that is a foreign key from another table
templateID (PK,FK)	integer	primary key that is a foreign key from another table
clauseOrderNumber	integer	the order a Clause comes in the template

Table Name: wizard_group_permissions

Description: This table is a linking table that contains the Contract Group permissions for a Wizard, i.e. a user can use a Wizard if they are part of the respective Contract Group for that Wizard.

Field Name	Type	Description/Notes
wizardID (PK,FK)	integer	primary key that is a foreign key from another table
contractGroupID (PK,FK)	integer	primary key that is a foreign key

		from another table
--	--	--------------------

Table Name: template_group_permissions

Description: This table is a linking table that contains the Contract Group permissions for a Template, i.e. a user can use a Template if they are part of the respective Contract Group for that Template.

Field Name	Type	Description/Notes
templateID (PK,FK)	integer	primary key that is a foreign key from another table
contractGroupID (PK,FK)	integer	primary key that is a foreign key from another table

Table Name: clause_group_permissions

Description: This table is a linking table that contains the Contract Group permissions for a Clause, i.e. a user can use a Clause if they are part of the respective Contract Group for that Clause.

Field Name	Type	Description/Notes
clauseID (PK,FK)	integer	primary key that is a foreign key from another table
contractGroupID (PK,FK)	integer	primary key that is a foreign key from another table

Table Name: workflow_group_permissions

Description: This table is a linking table that contains the Contract Group permissions for a Workflow, i.e. a user can use a Workflow if they are part of the respective Contract Group for that Workflow.

Field Name	Type	Description/Notes
workflowID (PK,FK)	integer	primary key that is a foreign key from another table
contractGroupID (PK,FK)	integer	primary key that is a foreign key from another table

4. Quality Assurance Plan

The Software Quality Assurance Plan is a plan that is put in place to assure software quality for the Contract Management System (CMS). The plan should help to “find errors before they become defects”, and the overall goal is to produce as high quality a system as possible.

4.1 Software Configuration Management

As mentioned in the Requirements Specification document for this project, a Software Configuration Management (SCM) tool will be used to track changes to work products in a “check-in/check-out” fashion. This SCM tool will be used to track changes to the code, data and documentation regarding all phases of the project. The SCM should track who made changes and when they made the changes. Also, any changes at a certain level will require approvals by relevant personnel. In regards to the Software Quality Assurance (SQA) Plan, the SCM will be a repository for all SQA activities.

4.2 Requirements Changes

Along the way, even into the development and testing phase of the project, there may be changes to the Requirements Specification. The natural flexibility of a Web Application lends well to changes, as could happen due to business reasons, technical reasons, political reasons, and so forth.

Any changes to the Requirements Specification will require stakeholder discussion and approval. The changes should be logged into the SCM tool, and the actual Requirements document should be modified accordingly.

4.3 Design Changes

In addition to changes to the Requirements Specification, changes to the Design Specification may occur as well. Like changes to the Requirement Specification, changes to the Design Specification should be discussed, agreed upon, and properly logged in the SCM. Any major changes will obviously require much more effort on the part of the stakeholders to work through the implications.

4.4 Formal Technical Reviews

Formal Technical Reviews (FTRs) should also be put in place to help to provide a mechanism to review work products. The FTRs can be conducted in a rather agile manner, which lends well to the Web Application style of development. As noted in Pressman, FTRs are used to catch defects early in the development process, and consist of:

- Small teams that meet to review a work product.
- The team members are given a chance to review the work product before the meeting.
- During the meeting, members walk through the work product.
- The individual responsible for the work product should be treated with courtesy and should be allowed to see the problems themselves through a Socratic approach.
- Defects or problems are noted via a note taker (the notes are subsequently added to the SCM).
- The work product is accepted or rejected as appropriate.

The work product to be reviewed could be documentation, changes to the Requirements or Design, code, testing plan, or even this SQA plan itself. If the work product is rejected, then it must be “fixed” by the appropriate people responsible. The people responsible should be given an estimated date to rectify any problems by.

The FTR schedule should vary depending on the phase of the project. During the actual coding phase, the FTRs should be conducted more frequently, such as every week, or even daily as in smoke testing. The code problems should be caught early, so that they don’t blossom into larger bugs.

4.5 Validation and Verification

As the code comes together, and the Requirements and Design become more set in stone, there will be a need to make sure that the system meets the plan set out in the Requirements and Design Specifications.

In order to validate and verify that the Web Application conforms to the Requirements and Design, testing will need to occur at various times during the coding phase, and afterwards through delivery.

Not only should the explicit requirements be tested, but the Web Application should comply with implicit requirements as well. Implicit requirements include expected paradigms for a web application, such as generally

accepted formats for dropdown boxes, link style, etc. In other words, those testing the application can and should note problems that don't necessarily involve a specific requirement listed in the Requirements Specification.

Through the Test Plan (which is covered in Section 5 below), any issues and defects should be noted and logged into the SCM tool. In addition, major issues should be reported to middle and possibly upper management as soon as possible.

4.6 SQA Personnel

The people who are performing QA activities are generally the development personnel, except where it makes sense to have the IT Support Team and CMP Team step in. However, during the later stages of the validation and verification phase, the CMP Team will directly assist in testing the application. The involvement by the CMP Team will free up the development staff to focus of fixing issues instead of being bogged down by testing activity, since the development team is rather small.

5. Test Plan

The Test Plan is presented below and provides an overview of the testing activities that are part of the overall Software Quality Assurance (SQA) plan.

5.1 Overview

The Test Plan will cover testing of the main areas of functionality of the Contract Management System (CMS) as mentioned in the Requirements and Design Specifications:

- General Features
- Contract Record-keeping
- Contract Workflow/Routing
- Company Record-keeping
- Reporting Features
- Administrative Tasks
- Security Functions

In addition, attention should be given during functionality testing to the "General and Nonfunctional Requirements" as listed in Section 3.9 in the Requirements Specification. These requirements include usability, reliability, efficiency and capacity, maintainability, availability, and interoperability.

The functionality testing will happen in three stages:

- Module testing – Primarily conducted during the early part of the development stage, module testing will be conducted by the developers and will consist of testing the functionality of a given module and making sure that it performs as expected.
- Integration testing – This integration testing will occur as the individual modules come together. The CMP Team will be brought in to assist the developers and to verify that the system behaves as expected.
- Validation testing – Occurs during the final stages of development and delivery of the system. The CMP Team, developers, IT Support Team, as well as general system users will be brought in to validate that the requirements have been met and that the system functions as expected.

5.2 Tracking Issues

At any time testing is being conducted, issues should be tracked. The issues and problems should be logged into the Software Configuration Management (SCM) system and given “defect ID” numbers so that the issues can be tracked and resolved. The output from an error situation should be recorded and logged along with the defect ID and defect description. Logging the issues will also provide a history of problem solving so that any future similar occurrences of an issue can be fixed more quickly. Issues that require changes to the Requirement, Design or Code should be handled in accordance with the SQA plan.

5.3 Testing Outline

For integration testing, the team will follow a general outline of functionality to test. This outline covers the most used features of the application and will catch the majority of errors. Note that all areas of the system, and all possible inputs, cannot be tested due to time.

The testers, which will include developers, and at times the CMP Team, IT Support Team, and the general user community, will use the below outline.

Screen/Feature	Test Details	Results/Comments
Navigation	Verify that navigational elements are correct and that links take you to the proper destination	
“Go Search” Feature	Make sure this feature is indexed so that it works as expected. Run several queries from this feature (e.g., company name, contract number)	
Browse screens for companies and contracts	Test “Filter by” feature, sorting, and navigational features	
Companies & Contracts	Create a company and attach a contract to ensure these screens work as expected	
Event Alerts	Test by creating a contract with an expiration date to ensure it works as expected	

Documents	(1) Open documents to ensure they can be launched as expected, (2) Upload a document that less than 20 MBs, (3) Upload a document that is greater than 20 MBs	
Journals	Send out a journal using your email address to ensure it is sent as expected	
Searches / Reports	Verify and validate that saved searches and reports work as expected	
Acquired By	Process an acquisition of a company to another	
Assignment	Process an assignment of a contract record from one external company to another	
Name Change	Process a name change of an external and internal company record	
Workflow	Process a new deal to ensure deal workflow is working as expected	
Admin	Create and delete a user account, a contract group, a UDF, a Workflow. Adjust system settings. Empty the recycle bin.	

After testing an area of functionality, they will note any issues for logging into the SCM tool. In the case of a user from the general user population, they will give their test results to a member of the CMP Team so that the issue can be logged into the SCM tool. A series of "demonstrations" will be conducted to serve as the testing session for general users to participate in.

5.4 Releases and Future Defect Remedies

As the stability of the modules and system is achieved at each testing phase, the code and work products will be base-lined, that is, given a version number and locked in the SCM.

When the CMS is delivered as a release for use by Really Big Wireless, defect tracking will still continue. As before, the defects will be logged and tracked. The defects will be prioritized in terms of which ones are critical to be fixed first. Developers will work from this defects list to correct the system. If there are non-defect enhancements requested, then these will be taken into consideration, and prioritized as well by the CMP Team.