|  |
| --- |
| Thomson Reuters |
| Bisk Admin Manage A Firm |
| User Story 11214 |

|  |
| --- |
| Williams, Lenny  Tuesday, May 19, 2015 |

Table of Contents

[User Story 11214 Overview 1](#_Toc419788334)

[Task 12018 – Create UI wireframe for Bisk Admin Firm Management 2](#_Toc419788335)

[Search 5](#_Toc419788336)

[Hyperlink Names 6](#_Toc419788337)

[Hyperlink Emails 6](#_Toc419788338)

[Disabled Checkbox 6](#_Toc419788339)

[“VCR” Controls 6](#_Toc419788340)

[Page Controls 6](#_Toc419788341)

[Firm Administration – Update Firm 6](#_Toc419788342)

[Firm Admins 8](#_Toc419788343)

[Firm User 9](#_Toc419788344)

[Firm Purchases 11](#_Toc419788345)

[Firm Seat Purchase 12](#_Toc419788346)

[Divisions 14](#_Toc419788347)

[Firm Division Form 15](#_Toc419788348)

[Division Admin 16](#_Toc419788349)

[Task 12228 – Implement UI Framework 17](#_Toc419788350)

[Task 11502 – Create Functionality for Update Firm 26](#_Toc419788351)

[Form Changes – Grids 27](#_Toc419788352)

[Task 12219 – Create Functionality for Firm Users 28](#_Toc419788353)

[Roles 28](#_Toc419788354)

[Session Variables 29](#_Toc419788355)

[UI Form 29](#_Toc419788356)

[Temp User Name and Password 33](#_Toc419788357)

[UI Grid 34](#_Toc419788358)

[Task 11504 - Create Functionality for Global Firm Administrators 35](#_Toc419788359)

[Firms 36](#_Toc419788360)

[Firm Administrators 37](#_Toc419788361)

[Firm Administrator 38](#_Toc419788362)

[User Role 41](#_Toc419788363)

[Division Users 42](#_Toc419788364)

[Task 11503 – Create Functionality for Firm Purchases 43](#_Toc419788365)

# User Story 11214 Overview

As a **Bisk Administrator** I would like the ability to manage the various firms so that I can easily perform admin duties.

**Acceptance Criteria**

* The ability to search existing firms by (partial keyword).
  + Name
  + E-mail
* The ability to edit (Add, Delete, Modify) a firms contact information (See Create a Firm story; all fields). Sub Admins.
* The ability to assign (add) and remove firm administrators for the firm.
* Ability to Assign and Edit a primary Firm Administrator
* The ability to disable an entire firm's access to the site including its candidates.
  + Use Case Scenario:

Term has expired

Non-payment

* Ability to view all firm admins
* Ability to view all firm information

This iteration is geared towards an administrative component for the CPAReview Administrators. As a Bisk Administrator, you will be able to add new firms, add purchases, and accomplish other tasks the actual Firm administrator should be doing.

Once the Global Firm administrator has been added, he/she will be able to add new Global administrators, division administrators, divisions, and work with the seats purchased for their organization. The intent of this application is to NOT have Bisk Administrators doing anything more than creating the firm, adding an administrator, and adding seat purchases.

It is expected that the Bisk Administrators will be adding seat purchases on a regular basis, so the UI should be easily available from the CPAReview admin location and it should be simple to use without much training.

# Task 12018 – Create UI wireframe for Bisk Admin Firm Management

A large mouthful, but this task is paramount in terms of defining the “How” for navigating the application from a Bisk Administrators point of view. This wireframe will be adapted to the actual Firm user experience, so the paradigms exposed here will be important for the overall look and feel.

The wireframe has been created in the following application

Balsamiq Mockups

Version: 3.0.3 - 03/11/2015 10:31

Flash Player Version: WIN 17,0,0,128

In the beginning, the following process was coded for the Create Firm user story:

|  |
| --- |
| C:\Users\uc190433\AppData\Local\Temp\flaFCFC.tmp\Snapshot.png |

By selecting the Firm button, the user was sent to a form that allowed them to enter the data, and then press a Create Firm button to create a record in the FirmParent data table. Additional Search functionality was added to support the User Story for Create Firm.

All of this work actually provided complex data structures and behind the scenes functionality that will allow development for all of the data table requirements, LINQ capabilities, and some very sophisticated UI implementations. With that in mind, this is the expected flow to begin sorting out Task 12018, which has the following description:

*As a Firm Administration I need a UI to access actions to create, manage, and update firms.*

*This framework should provide easy access to create, search for, update firms, update firm purchases, disable, and view firms. It will be used to tie together all of the Bisk Administration functionalities regarding the firm administration application.*

The previous wireframe illustration brought the user to the Firm Selection button. The following are wireframe that extend that process…

|  |
| --- |
| C:\Users\uc190433\AppData\Local\Temp\fla2516.tmp\Snapshot.png |

The text that the user will first see for this application appears as follows:

|  |
| --- |
| Welcome To CPAReview Firm Administration    If this is your first visit, you will need to create a firm to associate with purchases, firm administrators, divisions, division administrators, etc. Each time you select an item that does not have any records associated with the firm selected, you will be presented with a form to enter information in for that item.    After you have successfully entered the information, the next time you select that item, you will be presented with a grid that will provide a link to create new items, an ability to disable any of the items, and a hyperlink under the name of the item that will navigate you to the form so that you can edit the item.    When you click on the Firm menu item at the left, you will be taken to the "Create a Firm" form. If there are already Firm records, you will see the grid UI that will allow you to select a firm to work with. |

This expresses the experience that the user can expect throughout their interaction with this Firm Management application. They will see Forms or Grids representing an item or a list of those items. The navigation actions are clearly accessible by the user. A tree selection always exists on the left, and the items under the tree would represent information linked directly to the firm selected.

|  |  |
| --- | --- |
| The next thing if the Firm database has never had any records added would be that user would be presented with a form to fill in.  Once the Firm information has been added correctly, the next time the user clicks on the **Firm** Tree navigation is that they would see a grid instead of a form. | C:\Users\uc190433\AppData\Local\Temp\flaDDDB.tmp\Snapshot.png |

The grid that would appear for Firms after the very first firm was entered has a number of features and functionality that mirror what almost every other Firm specific table will have. The following features can be considered common to almost every other grid you will encounter in this application:

* Search
* Hyperlink Names
* Hyperlink Emails
* Disabled Checkbox
* “VCR” Controls
* Page Controls

|  |
| --- |
| C:\Users\uc190433\AppData\Local\Temp\flaA060.tmp\Snapshot.png |

You can expect the following from each of these bulleted items:

### Search

The search will be specific to the context of the grid. If it is a firm grid, the search will go through multiple field columns to try and match the search string. We can add or remove column references for the search pattern, if the search appears to work slowing. The search uses an optimistic pattern – if Name is a primary field, it would be first in the search, address might be the second, email address is a possible search column, etc…

Search uses internal LINQ coding techniques, which allow for a more sophisticated search patterning than SQL.

### Hyperlink Names

Any name that is hyperlinked would navigate to the Firm Users form for that user. If the hyperlink is a Firm, Department, Seat Purchase, or Seats - it would navigate to that item’s grid or form if there are no records found for that navigation item.

### Hyperlink Emails

Any hyperlinked email would launch an email to that user.

### Disabled Checkbox

Once an item has been disabled, there will be business logic reference to understand how to decouple the record from any associated children and/or disable those associated records.

### “VCR” Controls

A stated – first record, last record, next record, previous record

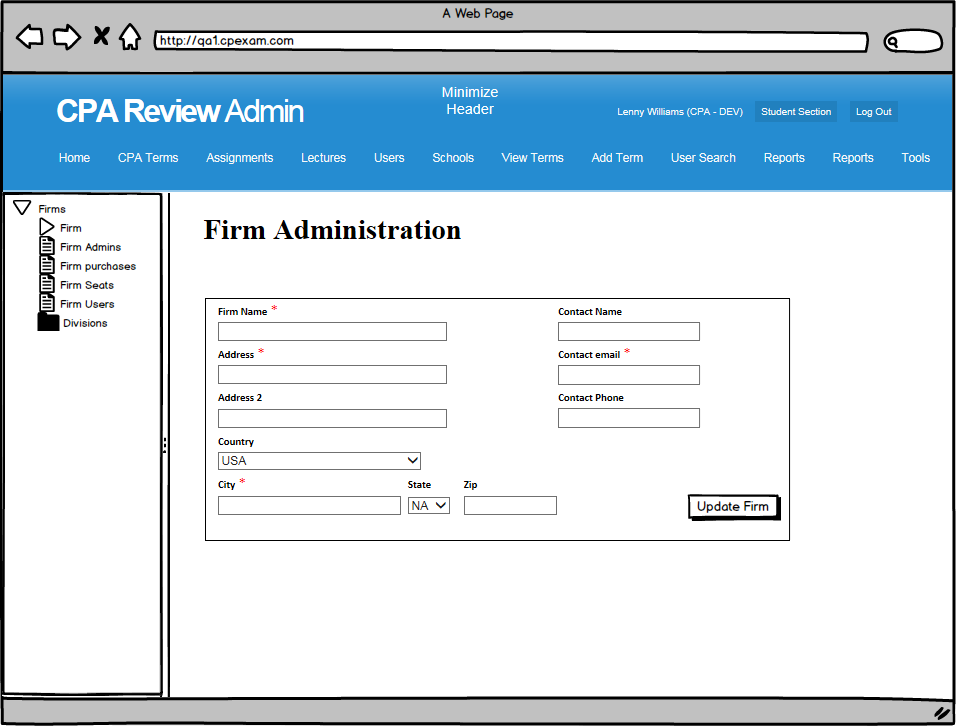
### Page Controls

Clicking on the Page number link will take you to the set of records on that page. Paging is arbitrarily set to 5 records per page. Setting this limit by each user has been discussed, but this is a larger discussion. We would need to have user available settings which would need to be saved in some fashion (cookie, xml file, database table).

|  |  |
| --- | --- |
|  | Please note that the Tree navigation on the left side of the screen is intended to be contextually driven.  You would not see or be able to reference Firm Admins, Firm purchases, Divisions, etc. unless you had actually selected or created a firm. |

## Firm Administration – Update Firm

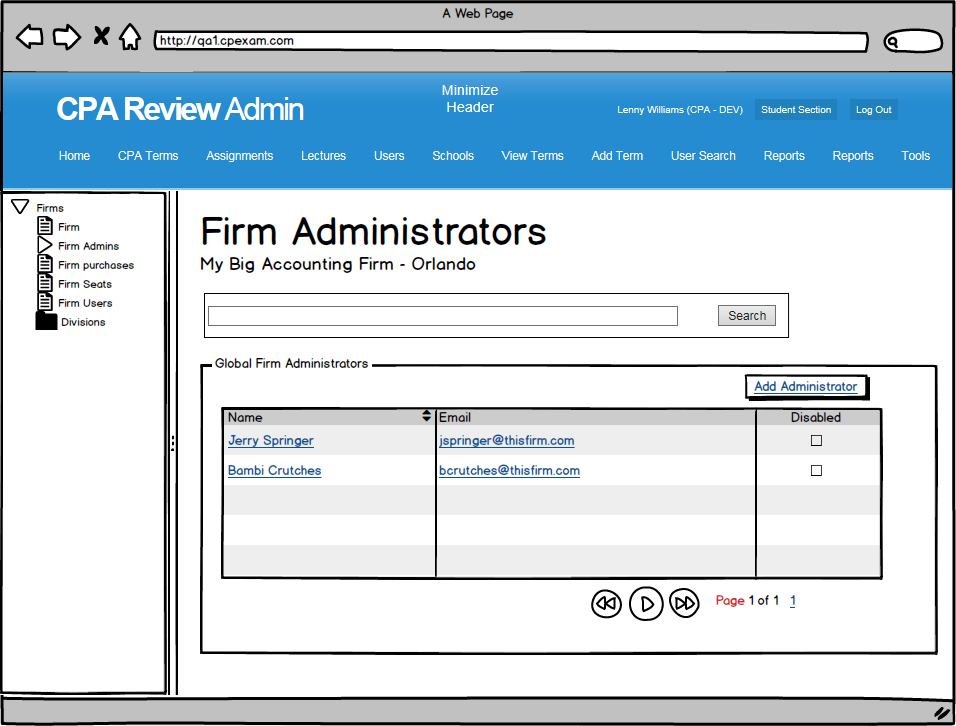
As mentioned in the *Hyperlink Names* reference previously – if you select a Firm by name, you will get a form (not filled here – but it would be) with the Update Firm button available to update this record if there any changes.



The same process is true for the Firm Administrator (Design change might involve adding an Administrator for the Firm and displaying with the Firm grid information).

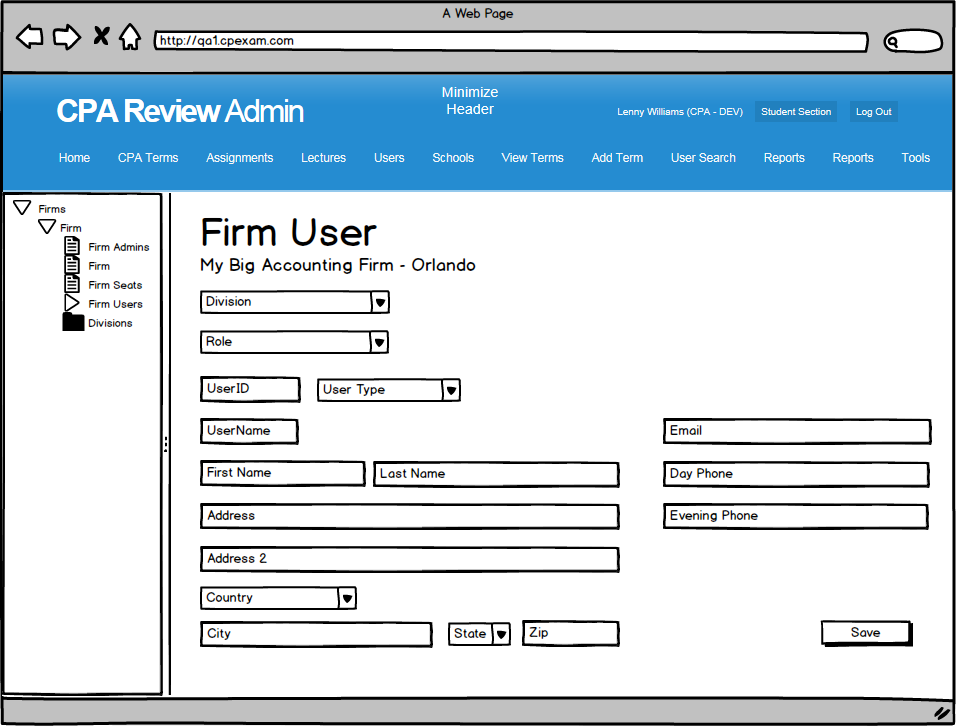
## Firm Admins

If there are no Firm Administrators, you would be taken to a Firm Administrator’s form entry. If there are Firm Administrators for the selected firm, you would see a Firm Administrators screen such as the following:



If there are no records for the **Firm Administrator** for this firm, you would see the **Firm User** form first, instead of the Firm Administrator grid. This is the same form you would see if you clicked on the name hyperlink.

## Firm User



This table is actually a link between **FirmUsers** and the CPAReview table – **User**.

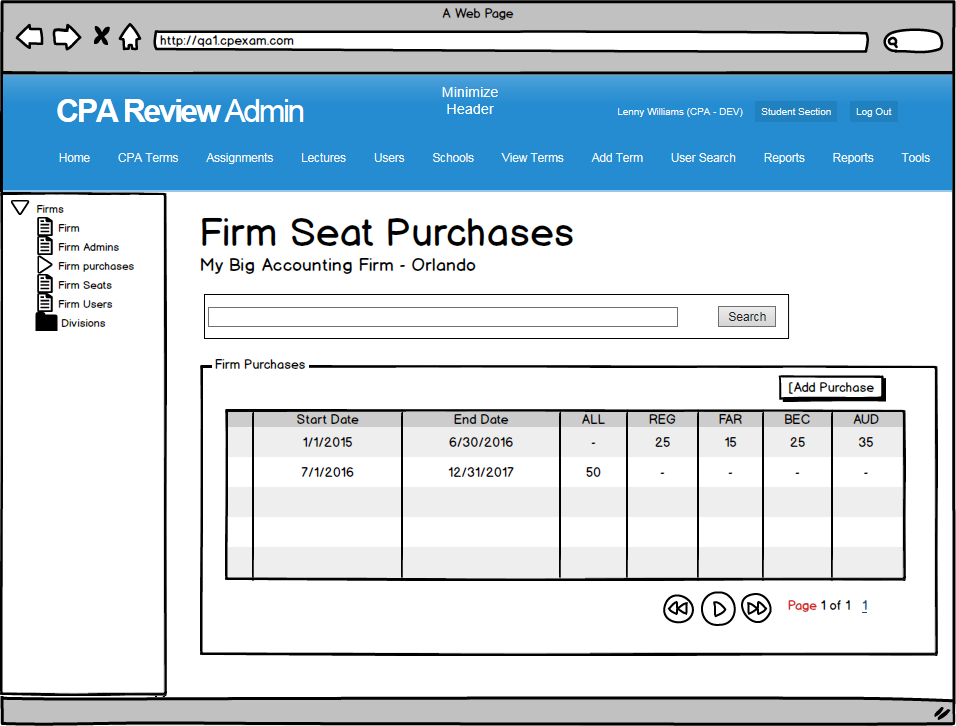
|  |  |
| --- | --- |
| FirmUsers | Users |
| SELECT [FirmUsersID]  ,[UserID]  ,[FirmParentID]  ,[FirmDivisionID]  ,[FirmRoleID]  ,[DateCreated]  ,[LastModified]  ,[ModifiedBy]  ,[IsDisabled]  FROM [CPAExam3].[dbo].[FirmUsers] | SELECT [UserID]  ,[TypeKey] - autoset  ,[UserName] – readonly?  ,[Password]  ,[FName]  ,[LName]  ,[CustomerNum]  ,[DivisionKey]  ,[Email]  ,[Address]  ,[Address2]  ,[City]  ,[State]  ,[Zip]  ,[CountryID]  ,[DayPhone]  ,[EvePhone]  ,[Fax]  ,[SchoolAttended]  ,[SalesRep]  ,[bBioVisible]  ,[Biography]  ,[CDROM]  ,[UPN]  ,[OfficeKey]  ,[ServiceLineKey]  ,[UserDateCreated]  ,[UserLastModified]  ,[UserModifiedBy]  ,[WebsiteID]  ,[PasswordExpiration]  ,[bShowQuestionsExplanations]  ,[SpecialShippingInstructions]  ,[SalesRepEmail]  ,[University]  ,[GraduationDate]  ,[SalesRepUPN]  ,[WizardStatus]  ,[GreatPlainsRepID]  ,[CandidatePostNotificationTypeID]  ,[FacultyPostNotificationTypeID]  ,[EnableBookmarks]  ,[EnableBooknotes]  ,[IsDisabled]  ,[TestUser]  FROM [CPAExam3].[dbo].[Users] |

If the user already exists, you would only change the ***Division*** and the ***Role*** from the drop-downs. The system will do a search on ***UserName***, ***Fname***, ***Lname***, and the ***Email*** fields to insure that this is a unique user and would present the Users grid to select from if there is more than one match. If there is only one match – the user’s details would be filled in, and if there is no match – you would enter data as expected.

The routine for adding a new user would mirror the routine in CPAReview, so that the system can take advantage of that already established process. It is important to remember that this user is exactly the same as any user in the CPAReview application. That can be viewed in CPAReview like any other user and they can be assigned to classrooms like any CPAReview user.

The primary difference is that their *UserID* has been added as part of the FirmUsers table records and will be referenced in the Firm Administration as a member of a firm and possibly a division (not all firms will have divisions)

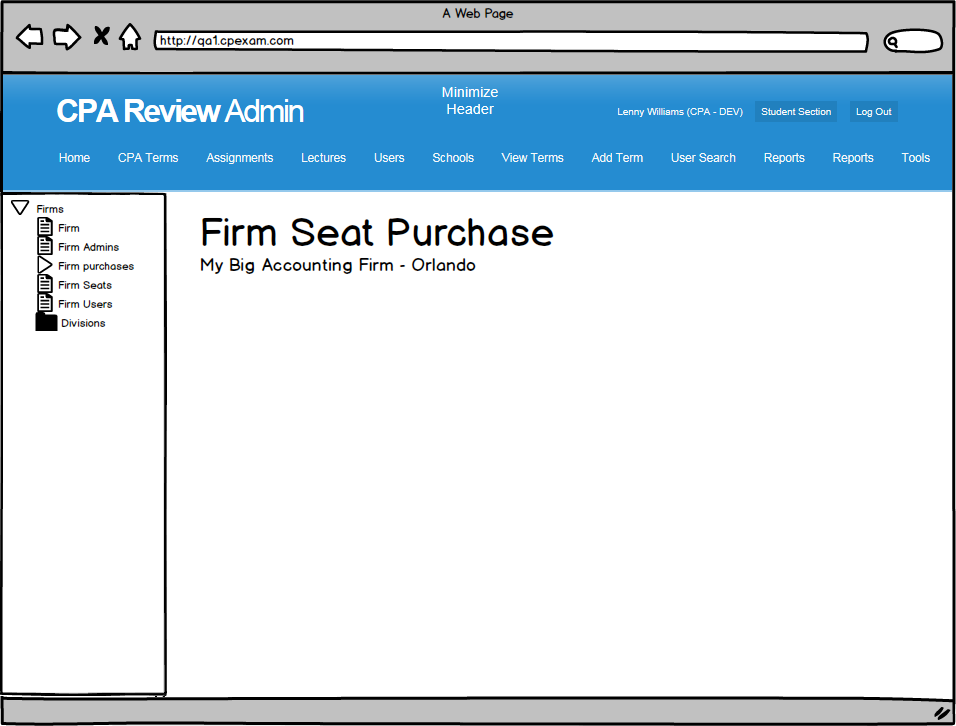
## Firm Purchases



This form has some complex events that will follow the Add Purchase button action. There are business rules that might come into play with this particular table set. The user may need to juggle around the seat totals between each of the Functional seat areas (REG, FAR, BEC, or AUD). They also may be allowed to change or move the Purchase to a different time set.

As with other grids, clicking the Add… button should take you to a form that represents a single Firm purchase record. Currently – there is no cross-reference between the Firm Seat Purchases and some order record. This should be considered here.

## Firm Seat Purchase



This is a lot of TBD since this is a plumbing issue. The Tables that are impacted extend beyond the Firm Administration application. The link between Firm and CPAReview starts with the CPAExam3.dbo.**FirmSeatPurchase** and the CPAExam3.dbo.**FirmUsers** table to the CPAExam3.dbo.**Registration** table. There are tertiary links from the CPAExam3.dbo.**Registration** table and the CPAExam3.dbo.**Calendar** table and it’s links.

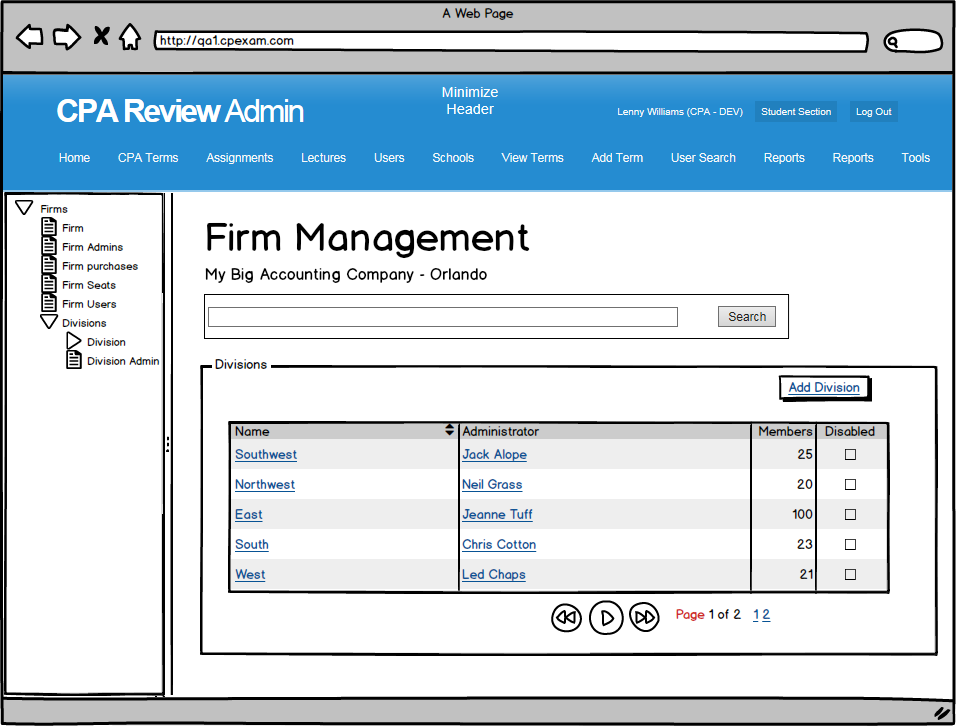
|  |  |
| --- | --- |
| FirmSeatPurchase | Registration |
| SELECT [FirmSeatPurchaseID]  ,[FirmParentID]  ,[RegistrationKey]  ,[DateTimeStamp]  FROM [CPAExam3].[dbo].[FirmSeatPurchase] | SELECT [RegistrationKey]  ,[CalID]  ,[UserID]  ,[RegistrationStatus]  ,[RegistrationDate]  ,[TransferID]  ,[DateShipped]  ,[CPAOnlinePaid\_to\_be\_deleted]  ,[CPAOnlinePaidStatus]  ,[eCommOrderID]  ,[IsDeleted]  ,[NewsDate]  ,[CourseStartDate]  ,[CourseEndDate]  FROM [CPAExam3].[dbo].[Registration] |

|  |  |
| --- | --- |
| Calendar | WebVersion |
| SELECT [CalID]  ,[TermKey]  ,[WebVersionID]  ,[SectionNum]  ,[StartDate]  ,[EndDate]  ,[SurveyWeek]  ,[EvalWeek]  ,[bolInstructor]  FROM [CPAExam3].[dbo].[Calendar] | SELECT [WebVersionID]  ,[Version]  ,[CourseVersionID]  ,[MediaVersion]  ,[WebVersionStatusKey]  ,[LectureVersionID]  ,[ParentCourseID]  ,[WebVersionTypeID]  FROM [CPAExam3].[dbo].[WebVersion] |
|  |  |

Yipes – lots of dependencies as we transition through the links. The documentation for this will be in the next User Story – Bisk Admin Seat Assignment Management.

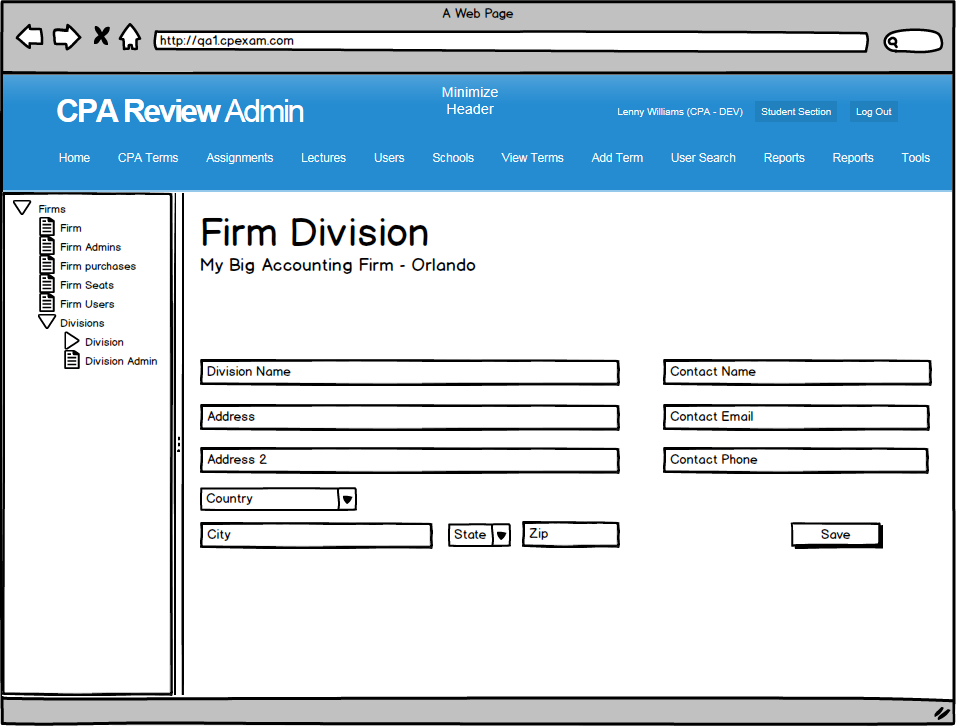
## Divisions

Like previous grid, the functionality here allows you to add a new division, or click on the Division name to edit/update that division’s information. If there are no existing divisions, you would see a Division form instead of the grid.



Clicking on the **Add Division** button or clicking on the Division *Name* hyperlink will take you to the **Division** form, where you can add and/or change information. Clicking on the *Administrator* name hyperlink will take you to the **Firm Users** form, where you can edit that information.

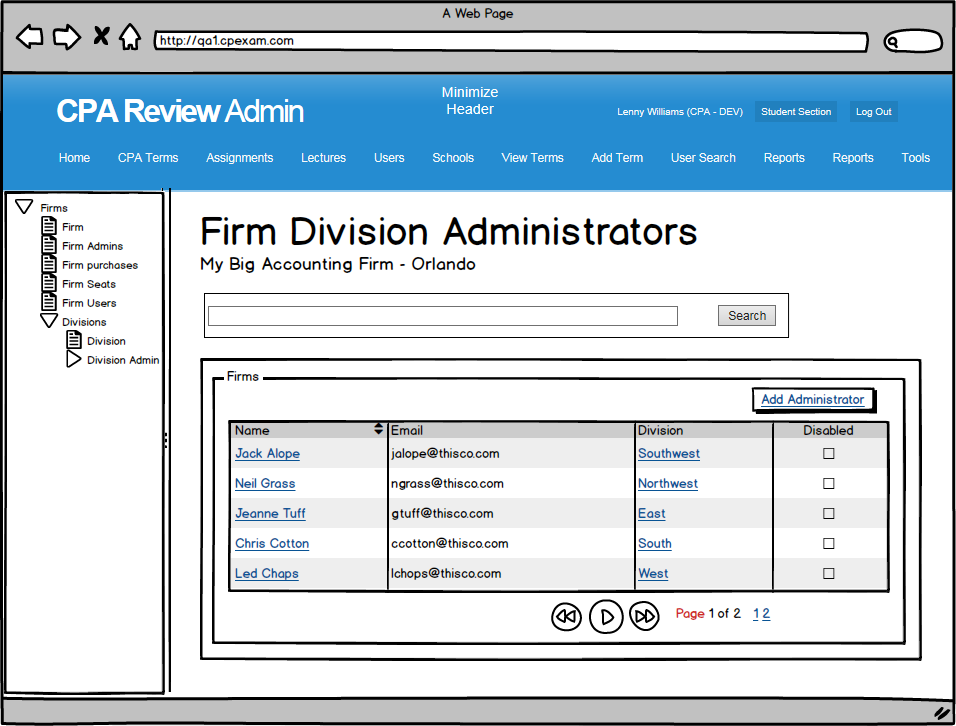
## Firm Division Form



|  |  |
| --- | --- |
| The FirmParentID will be passed when this form appears as it would be an expected filter.  When adding a division name, the system will look to see if this division exists in this firm first. | SELECT [FirmDivisionId]  ,[FirmParentID]  ,[DivisionName]  ,[Address]  ,[Address2]  ,[City]  ,[State]  ,[Zip]  ,[CountryID]  ,[ContactName]  ,[ContactEmail]  ,[ContactPhone]  ,[DateCreated]  ,[LastModified]  ,[ModifiedBy]  ,[IsDisabled]  FROM [CPAExam3].[dbo].[FirmDivision] |

## Division Admin

This grid would be used to view/review the Division Administrators that are associated with each Division. If there are no users associated with the Firm’s Divisions, an update form would appear for this firm.



As mentioned previously, clicking on the *Name* hyperlink will take you to the **Firm Users** form, and clicking on the *Division* hyperlink will take you to that division in the **Division** form. Clicking on the **Add Administrator** button will again take you to the **Firm Users** form, where the **FirmParentID** and the **Role** as *Global Administrator* will be set.

The **Users** table record will also have the **UserType** field set to *Firm Administrator* so that the system can differentiate when a firm administrator (or division administrator for the firm) logs into CPAReview.

# Task 12228 – Implement UI Framework

This is a complex set of technology implementations. We will be adding Routing for ASP.NET web forms, and Navigation processes to allow for State management routing.

In the first implementation of this, I am installing author *Graham Mendick’s* Navigation component that is detailed on Github[[1]](#endnote-1). Installing it, I used the following Nuget command:

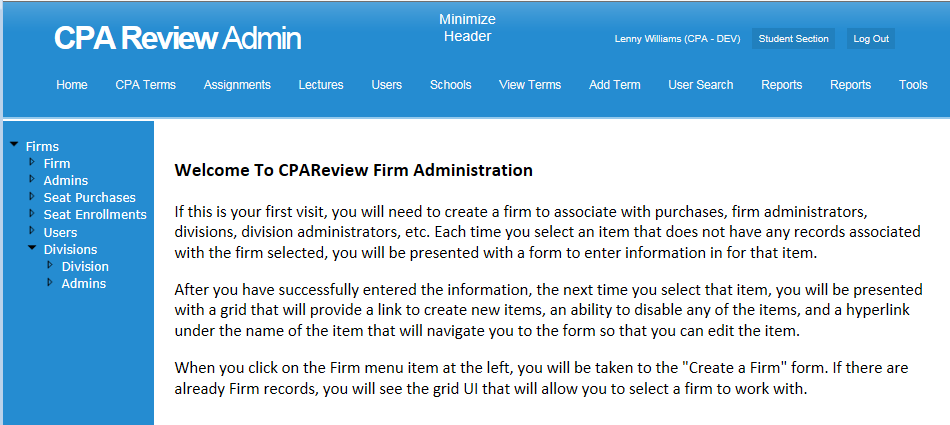
|  |
| --- |
| PM> Install-Package Navigation  Attempting to resolve dependency 'WebActivatorEx (≥ 2.0.0)'.  Attempting to resolve dependency 'Microsoft.Web.Infrastructure (≥ 1.0.0.0)'.  Installing 'Microsoft.Web.Infrastructure 1.0.0.0'.  Successfully installed 'Microsoft.Web.Infrastructure 1.0.0.0'.  Installing 'WebActivatorEx 2.0'.  Successfully installed 'WebActivatorEx 2.0'.  Installing 'Navigation 1.11'.  You are downloading Navigation from Graham Mendick, the license agreement to which is available at http://navigation.codeplex.com/license. Check the package for additional dependencies, which may come with their own license agreement(s). Your use of the package and dependencies constitutes your acceptance of their license agreements. If you do not accept the license agreement(s), then delete the relevant components from your device.  Successfully installed 'Navigation 1.11'.  Adding 'Microsoft.Web.Infrastructure 1.0.0.0' to CPAExamWebsite.  Successfully added 'Microsoft.Web.Infrastructure 1.0.0.0' to CPAExamWebsite.  Adding 'WebActivatorEx 2.0' to CPAExamWebsite.  Successfully added 'WebActivatorEx 2.0' to CPAExamWebsite.  Adding 'Navigation 1.11' to CPAExamWebsite.  Successfully added 'Navigation 1.11' to CPAExamWebsite. |

Web.config

|  |
| --- |
| <modules>  …        <add name="UrlRoutingModule"             type="System.Web.Routing.UrlRoutingModule,                 System.Web.Routing,                 Version=3.5.0.0,                 Culture=neutral,                 PublicKeyToken=31BF3856AD364E35"/>        </modules>  And under <handlers>        <handlers>          <add name="UrlRoutingHandler"               preCondition="integratedMode"               verb="\*"               path="UrlRouting.axd"               type="System.Web.HttpForbiddenHandler,                 System.Web, Version=2.0.0.0,                 Culture=neutral,                 PublicKeyToken=b03f5f7f11d50a3a" />        </handlers> |

The UI Framework relies on an asp:Treeview control and the site.master page which contains the page links. Navigation could be provisioned by a xml file or through a database, but the requirements do not call for this more complex approach. The site.master navigation links can easily be changed and deployed without deploying the CPAExam.dll.

The following is the Treeview and landing page for the CPAExam Firm administration application:



Note that the Treeview control is placed into its own content area on the left panel. The content to the right is in a ContentPlaceHolder control with an ID=”MainContent”. This site is using master pages, and new pages are added to the MainContent area when a menu item is clicked on from the left content area that is referenced as :

asp:ContentPlaceHolder ID="LeftNav" runat="server">

This content area is static through all pages, making it easy to transition between pages. The current TreeView implementation items are only draft suggestions. In code the nodes appear as follows:

|  |
| --- |
| <div id="leftcontent">      <asp:ContentPlaceHolder ID="LeftNav" runat="server">            <div>              <asp:TreeView ID="TreeView1" runat="server"                  ImageSet="Arrows">                  <Nodes>                      <asp:TreeNode Text="Firms" NavigateUrl="default.aspx">                          <asp:TreeNode Text ="Firm" NavigateUrl="Views\Firms.aspx"></asp:TreeNode>                          <asp:TreeNode Text ="Admins" NavigateUrl="NotThereYet.aspx"></asp:TreeNode>                          <asp:TreeNode Text ="Seat Purchases" NavigateUrl="NotThereYet.aspx"></asp:TreeNode>                          <asp:TreeNode Text ="Seat Enrollments" NavigateUrl="NotThereYet.aspx"></asp:TreeNode>                          <asp:TreeNode Text ="Users" Value="NotThereYet.aspx"></asp:TreeNode>                          <asp:TreeNode Text ="Divisions" Value="NotThereYet.aspx">                              <asp:TreeNode Text ="Division" Value="NotThereYet.aspx"></asp:TreeNode>                              <asp:TreeNode Text ="Admins" Value="NotThereYet.aspx"></asp:TreeNode>                          </asp:TreeNode>                      </asp:TreeNode>                  </Nodes>                  <HoverNodeStyle Font-Underline="True" ForeColor="#5555DD" />                  <NodeStyle Font-Names="Verdana" Font-Size="10pt" ForeColor="white" HorizontalPadding="5px" NodeSpacing="0px" VerticalPadding="0px" />                  <ParentNodeStyle Font-Bold="False" />                  <SelectedNodeStyle Font-Underline="True" ForeColor="#5555DD" HorizontalPadding="0px" VerticalPadding="0px" />              </asp:TreeView>          </div>      </asp:ContentPlaceHolder>  </div> |

The styles have not been optimized, but a system of CSS styles has been implemented in the Site.Master page. These CSS styles will eventually be contained in a separate CSS file that will be referenced from the Site.Master page, but they have been placed in the Site.Master page for ease of development. This is also true of the jQuery/JavaScript code – which will also be moved to a separate file that will be referenced from the Site.Master file.

The CSS styles and jQuery code that is contained in the Site.Master file are as follows:

|  |
| --- |
| CSS |
| <style type="text/css" media="screen">      #leftcontent{          position:fixed !important;          position:absolute;          top:0;          right:0;          bottom:0;          left:0;          border:5px;          border-color:#95C9EC;          border-style:inset;          background-color:#258CD1;          color:white;          margin-top:auto;          margin-bottom:auto;      }      .lbltitle {          font-family: Calibri,sans-serif;          font-size: 9pt;          font-weight: 800;      }        .auto-style1 {          width: 120px;      }        .errorTitle {          font-family: Calibri,sans-serif;          font-size: 8pt;          color: red;      }        .error {          font-family: Calibri,sans-serif;          font-size: 8pt;          color: red;      }        body {          margin: 0;          padding: 0;          background: #fff;      }        h1 {          font-family:Calibri, sans-serif;          font-size:18pt;          font-weight:700;      }     h3 {          font-family:Calibri, sans-serif;          font-size:16pt;      }      p {          font-family:Calibri, sans-serif;          font-size:14pt;      }        #leftcontent {          width: 15%;          float: left;          color: #000;          padding: 5px;          height:100%;          border:1px;      }        #rightcontent {          float: right;          width: 80%;          padding: 20px;      }      .FirmGrid {          font-family: Calibri,sans-serif;          font-size: 11pt;      }        .gridHead {          background-color: #258CD1;          font-family: Calibri, sans-serif;          font-size: 10pt;          color:white;      }      .gridRow {          background-color: #95C9EC;      }      .gridAltRow {          background-color: white;      }      .gridContainer {          margin-top: 40px;          width: 850px;          border: 1px;          border-style: inset;          background-color: white;      }  </style> |

|  |
| --- |
| jQuery/JavaScript |
| <script type="text/javascript" src="http://ajax.googleapis.com/ajax/libs/jquery/1.7.2/jquery.min.js"></script>  <script type="text/javascript">      $(document).ready(function () {          // would need to have regular class name assignments to use the .keyup-<whatever>          $('.keyup-blank').keyup(function () {              $('span.error-keyup-0').remove();              var inputVal = $(this).val();              var numericReg = /^\d\*[0-9](|.\d\*[0-9]|,\d\*[0-9])?$/;              if (!numericReg.test(inputVal)) {                  $(this).after('<span class="error error-keyup-0">Enter Information</span>');              }          });          $('.keyup-numeric').keyup(function () {              $('span.error-keyup-1').remove();              var inputVal = $(this).val();              var numericReg = /^\d\*[0-9](|.\d\*[0-9]|,\d\*[0-9])?$/;              if (!numericReg.test(inputVal)) {                  $(this).after('<span class="error error-keyup-1">Numeric characters only</span>');              }          });          $('.keyup-characters').keyup(function () {              $('span.error-keyup-2').remove();              var inputVal = $(this).val();              var characterReg = /^\s\*[a-zA-Z0-9,\s]+\s\*$/;              if (!characterReg.test(inputVal)) {                  $(this).after('<span class="error error-keyup-2">No special characters</span>');              }          });          $('.keyup-limit-8').keyup(function () {              $('span.error-keyup-3').remove();              var inputVal = $(this).val();              var characterReg = /^([a-zA-Z0-9]{0,8})$/;              if (!characterReg.test(inputVal)) {                  $(this).after('<span class="error error-keyup-3">Maximum 8 characters</span>');              }          });          $('.keyup-phone').keyup(function () {              $('span.error-keyup-4').remove();              var inputVal = $(this).val();              if (inputVal == "") {                  $('span.error-keyup-4').remove();              }              var characterReg = /^[2-9]\d{2}-\d{3}-\d{4}$/;              if (!characterReg.test(inputVal)) {                  $(this).after('<span class="error error-keyup-4">Format xxx-xxx-xxxx</span>');              }          });        });    </script> |

The jQuery referenced above is a template for the actual Validation jQuery referenced in the Firm.aspx file. That file implements the following jQuery.

|  |
| --- |
| <script type="text/javascript" src="http://ajax.googleapis.com/ajax/libs/jquery/1.7.2/jquery.min.js"></script>   <script type="text/javascript">       $(document).ready(function () {             // focus to firm           $('#MainContent\_tbFirm').focus();                 $('#MainContent\_tbFirm').keydown(function () {               if (event.which == 13) {                   event.preventDefault();               }               $('span.error-keyup-firm').remove();               var inputVal = $(this).val();               var CharReg = /\S/; // this will match any non-blank character               if (!CharReg.test(inputVal)) {                   $(this).after('<span class="error error-keyup-firm">Enter Firm Information</span>');               }           });             $('#MainContent\_tbAddress').keydown(function () {               if (event.which == 13) {                   event.preventDefault();               }                 $('span.error-keyup-address').remove();               var inputVal = $(this).val();               var CharReg = /\S/; // this will match any non-blank character               if (!CharReg.test(inputVal)) {                   $(this).after('<span class="error error-keyup-address">Enter Address Information</span>');               }           });             $('#MainContent\_tbCity').keyup(function () {               if (event.which == 13) {                   event.preventDefault();               }                 $('span.error-keyup-city').remove();               var inputVal = $(this).val();               var CharReg = /\S/; // this will match any non-blank character               if (!CharReg.test(inputVal)) {                   $(this).after('<span class="error error-keyup-city">Enter a City</span>');               }           });             $('#MainContent\_tbContactEmail').keyup(function () {               if (event.which == 13) {                   event.preventDefault();               }                 $('span.error-keyup-7').remove();               var inputVal = $(this).val();               var CharReg = /\S/; // this will match any non-blank character               if (!CharReg.test(inputVal)) {                   $(this).after('<span class="error error-keyup-7">Enter an Email Address</span>');               }               else {                   var emailReg = /^([\w-\.]+)@((\[[0-9]{1,3}\.[0-9]{1,3}\.[0-9]{1,3}\.)|(([\w-]+\.)+))([a-zA-Z]{2,4}|[0-9]{1,3})(\]?)$/;                   //var emailReg = /^([\w-\.]+@([\w-]+\.)+[\w-]{2,4})?$/;                   if (!emailReg.test(inputVal)) {                       $(this).after('<span class="error error-keyup-7">Invalid Email Format.</span>');                   }               }             });                   $('#MainContent\_CreateFirm').click(function () {                   if ($('span.error').length > 0) {                   alert('Errors!');                   return false;               } else {                   $("#CreateFirm").after('<span class="error">Form Accepted</span>');                   return true;               }               });           });     </script> |

The actual difference here is that instead of using a “*Class=*” reference as seen in the Site.Master reference, I am using an “ID=” reference such as (#MainContent\_tbContactEmail) id reference for a MainContent asp:TextBox with an ID of tbContactEmail. Standardization of textbox field names might minimize the need for individual jQuery code in each of the Form pages that require validation for their fields, but this creates a dependency that may or may not work as Form requirements evolve. This is a possible area where optimization might occur. At the moment, it is more import to have working validation on each of the forms without external dependencies.

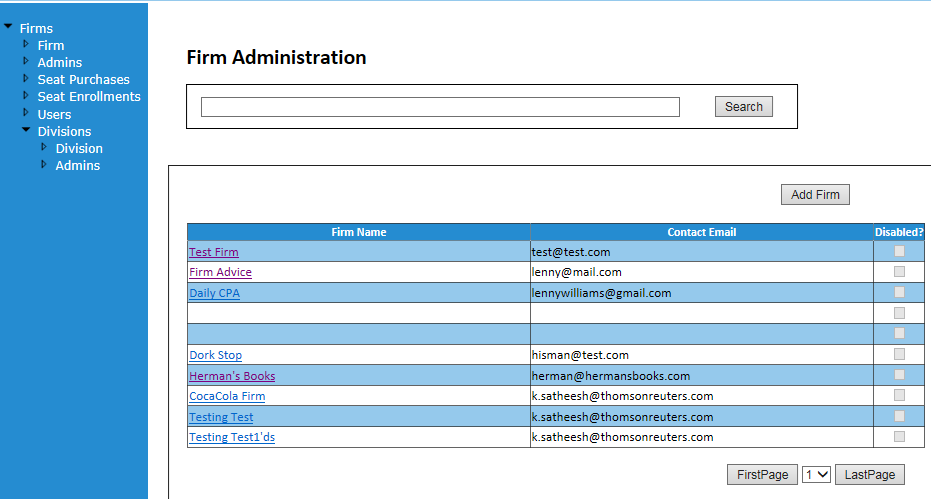
Another item that may not be apparent – the containers can be (and are) referenced to allow for changes in styles at the CSS level. This makes it much easier to change colors, fonts, and font sizes without too much trouble. When/if the requirement is implemented to have the capability to allow firms to change the look & feel, this built-in capability will allow for an easier implementation for that requirement.

# Task 11502 – Create Functionality for Update Firm

In this task, the user will be able to Update Firm information after the very first Firm has been created. As stated previously, when there are no records for Firm, Firm Divisions, Firm Administrators, Division Administrators, etc. – the first item the user will see when clicking on the associated menu link, the user will need to add information in a Form associated with that item type.

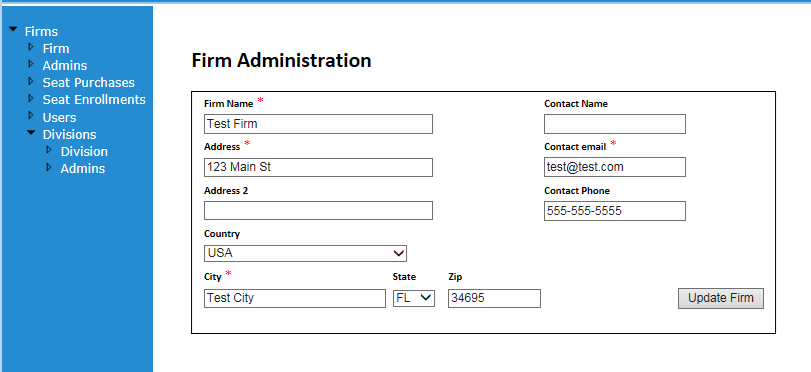
After a record has been added to any of the associated Firm or firm children records, the user will be presented with a grid that has hyperlinks on the Name column item. The Hyperlink will take the user to a Form that will allow them to update that record.

Here is an example of clicking on the Firm navigation menu item from the TreeView control on the left:



Note that you can click on the hyperlink to the **Firm Name**, or enter an item in the Search Text area and click on the Search button to locate the Firm, and then click on the hyperlinked **Firm Name** to update that firm.

Here is an example of the Update Firm form:



Clicking on the Firm navigation menu item will take you back to the Firm Grid. This behavior will be common in each of the subsequent Firm children form areas.

## Form Changes – Grids

This iteration of forms should be considered a draft approach, as some changes are expected in future coding. For example, the following list provides a view into some features that should be added:

1. Paging control will change from”



To something like…

<< | < Page 1 2 3 4 5 6 7… >| >>

1. Contact Email may be a hyperlink that will allow the user to click and popup an email client to email to the contact email address.
2. The Disabled button will be enabled with the program logic that accompanies this function. It should not be implemented until all child features have been added, since this function should undo (disable) all children from this level.

# Task 12219 – Create Functionality for Firm Users

The description for this task states:

Provide the following processes and implement functionality as required:

* Role filtering
* Seat assignments, adjusting, and business rules
* Sync or Add to CPAReview Users table
* Mimic CPAReview User registration process
* Linq and/or Stored Procedures
* UI Grid
* UI Form
* Business Rules
* Testing

In terms of complexity, this task rates high. It is crucial that the FirmUser table syncs perfectly with the CPAReview User table. When a FirmUser is established initially, they do not have any particular abilities, nor have they been assigned roles, seats, or classes.

## Roles

When first created, a ***Firm User*** has no role assignment. The associated ***User*** record has the TypeKey column for role set to 2 – which is a student designation. To reiterate information contained in the *Firm Support – Create Firm – Dev.docx[[2]](#footnote-1)* word document, these are the possible roles for Users in CPAReview:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| TypeKey | TypeName | TypeDesc | TextColor | HelpRoleID |
| 0 | Demo | Demo User | royalblue | 1 |
| 1 | Guest | Guest Account | MidnightBlue | 1 |
| 2 | Student | Student | SteelBlue | 1 |
| 3 | Professor | Professor | Sienna | 2 |
| 4 | Sales | CPAExam Sales Rep | OrangeRed | 1 |
| 5 | UA Rep | University Alliance Rep | YellowGreen | 1 |
| 6 | AdminSales | Administrator for Sales Dept | #FF0099 | 3 |
| 7 | Technical Support | Tech Support | blue | 1 |
| 8 | Guest Admin | Guest Admin | DarkRed | 3 |
| 9 | SystemAdmin | System Administrator | OrangeRed | 3 |
| 10 | E&Y Admin | Ernst & Young Admin | blue | 3 |
| 15 | Super Admin | Ernst & Young Super Admin | #ff0099 | 3 |
| 20 | Question Editor | Question Editor | Peru | 1 |
| 32 | Site Admin | Site Administrator | DarkViolet | 3 |
| 45 | Bisk Admin | Bisk Administrator | OrangeRed | 3 |
| 46 | CPAOnline\_Rep | CPA Online Reps | OrangeRed | 4 |
| 47 | CPAOnline\_Student | CPA Online Student | OrangeRed | 4 |
| 48 | CPAOnline\_Professor | CPA Online Professor | OrangeRed | 5 |
| 49 | KPMG\_Admin | KPMG Area Administrator | SteelBlue | 3 |
| 50 | KPMG\_AreaAdvisor | KPMG Area Advisors | SteelBlue | 1 |
| 51 | FirmAdminUser | Type of Firm Administrator | #cc3300 | 1 |

An additional reference from the *Firm Support – Create Firm – Dev.docx[[3]](#footnote-2)*  Word document details the expected role assignments for members of the FirmUsers table

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  |  | | --- | --- | --- | | 1 | FirmAdmin | Primary administrator for the firm | | 2 | DivAdmin | Division administrator | | 3 | ReportAdmin | Primary Firm readonly administrator | | 4 | DivReportAdmin | Division readonly administrator | | 5 | FirmUser | Firm designated user | |  |

## Session Variables

The Firm Administration application for Thomson Reuters Administrators has been enabled with session variables – most notably to support the Firm ID, since all child forms rely on this value.

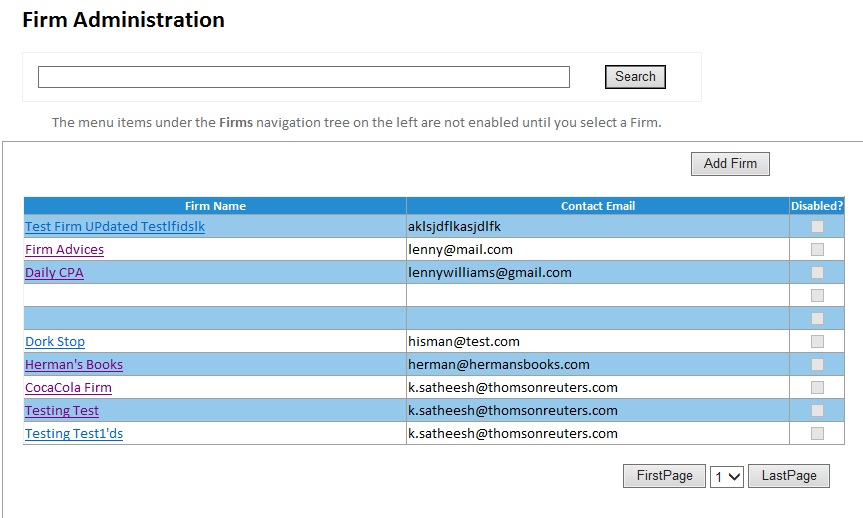
## UI Form

This is a complex form, in that there needs to be an association between the Firm and FirmUser. This is a state that is also true for all of the children of Firms – FirmUsers, Departments, Seats, Admins, etc. After stumbling about for a proper resolution to this situation, I decided to disable the TreeView child menu items under *Firms* whenever the Session[“ FirmID”] variable was not set.

To accomplish bit of program wizardry – the code-behind on the Site.Master page was altered to look for the proper Firm ID being set. Using this approach, it is unnecessary to have each of the child forms that are associated with a Firm force a redirect to the Firm Grid page to have the user select a Firm.

|  |  |
| --- | --- |
| The users will only be able to select any of the menu items under the Firms parent menu item if they select a firm.  If you try to select any of the child menu items under parent without first selecting a Firm – there will be no action available. |  |

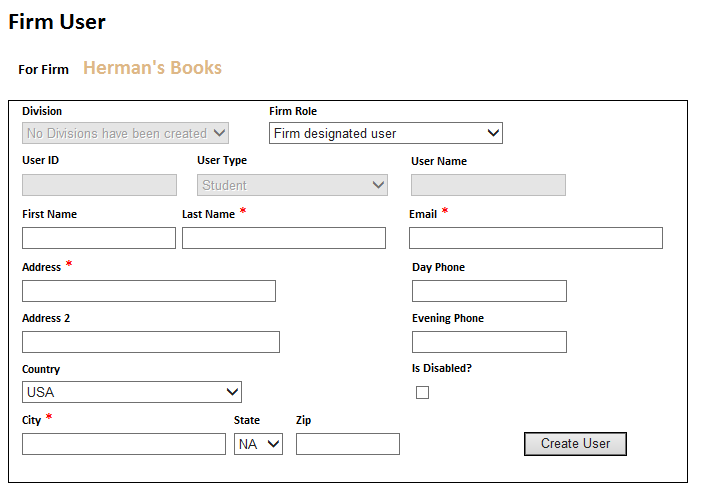
The Firm Grid should appear as follows:



Please note the message that prompts the user about selecting a firm. Each time the Firm menu item has been clicked – the global Firm ID variable is set to -1. This clears the application and allows for selecting a new firm.

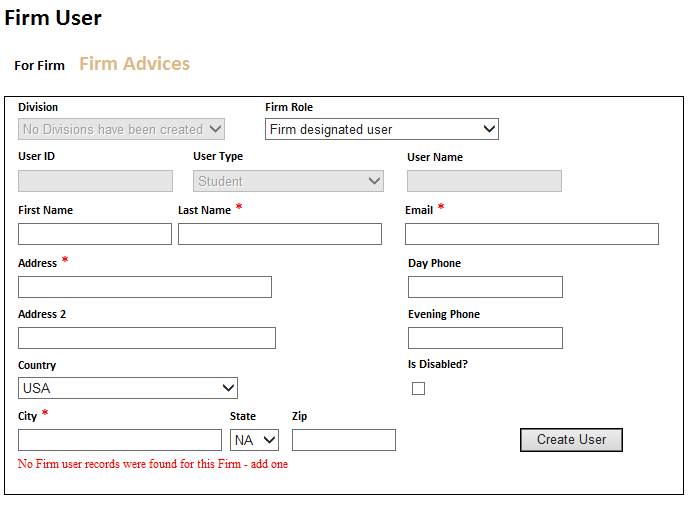
|  |  |
| --- | --- |
|  | Once a firm has been selected, the user is taken to the selected Firm form. |

|  |  |
| --- | --- |
|  | Now the user can select from any of the child menu items. These records will be associated with the Firm that was selected. |



The above form for *FirmUser* will only appear after clicking the *Users* navigation link when there are no FirmUsers records for the selected Firm. After there are one or more records in the **FirmUsers** table, you would see a *FirmUsers* grid first, and then you could select a Firm User or add a new Firm User.

This is an example of no records existing for the firm selected:



A new record will be added so there can be a grid display for the FirmUsers records associated with this firm.

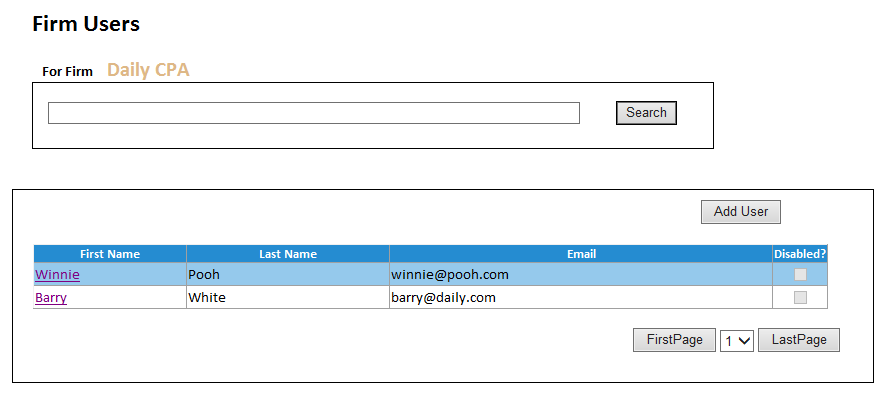
### Temp User Name and Password

When creating a new user, the following business rule is followed

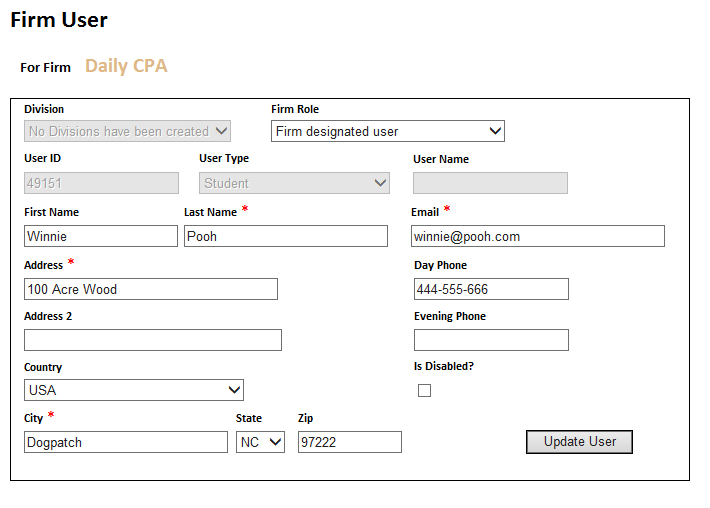
|  |
| --- |
| // have to create a new user  // need the process for creating new user (Register User?)  var tmpUserName = tbFirstName.Text.Substring(0,1) +      tbLastName.Text + tbZip.Text;  // first letter of First Name, last name, zip  // password is initials + "password"  var tmpPass = tbFirstName.Text.Substring(0, 1) +      tbLastName.Text.Substring(0, 1) + "password"; |

This is only a temporary solution. The CPAExam site user ***Add User*** functionality has a different approach and the best path forward is to try and mirror this…

## UI Grid



By clicking on a name in the *First Name* column, you will navigate to the Firm User record associated with that name.



# Task 11504 - Create Functionality for Global Firm Administrators

The brief requirements are stated as following:

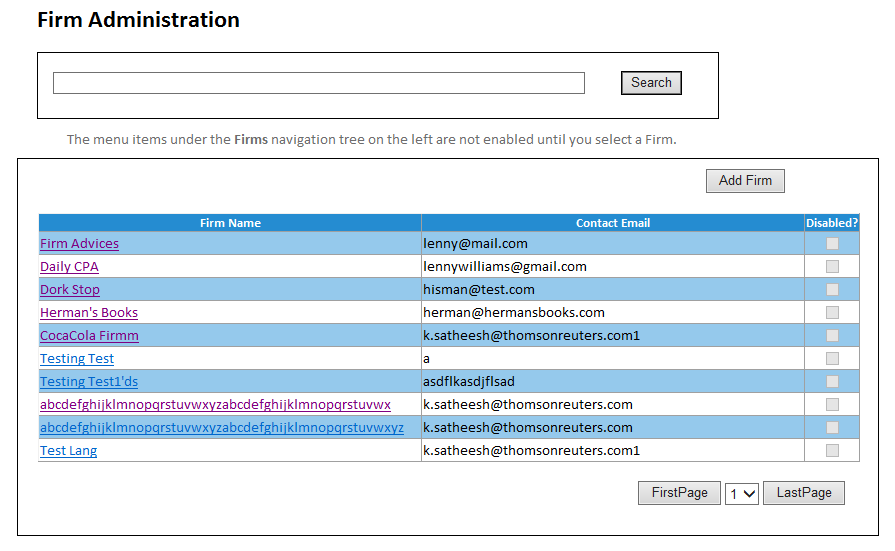
|  |
| --- |
| Provide the following processes and implement functionality as required:   * Role filtering * Linq and/or Stored Procedures * UI for Update Firm * Business Rules * Testing |

In many ways, the functionality and UI will mirror that of Firm User, since an administrator is really a Firm User with a role assignment. The difference will be that this user will be recognized by the CPAReview system when they log in and will be taken to the Firm Administration application directly with all of the associations directly linked to the Firm Administrator’s Firm.

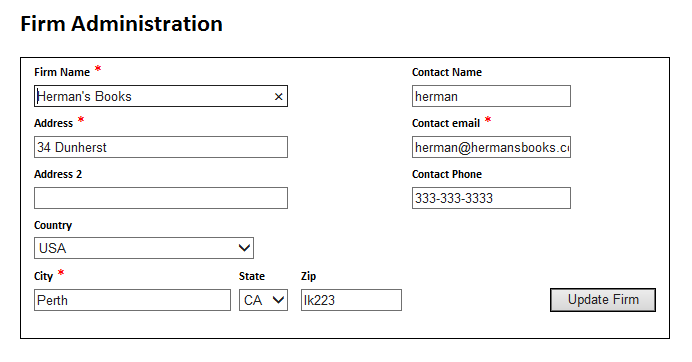
It is because of this that a Firm Administrators cannot be a CPAReview student. The anomaly could be resolved by having a screen after login recognition that gives the user a choice to login to the CPAReview system or to their Firm. This would be a business decision.

|  |  |
| --- | --- |
| Taking this Firm by Selecting from the Firms Navigation |  |

## Firms

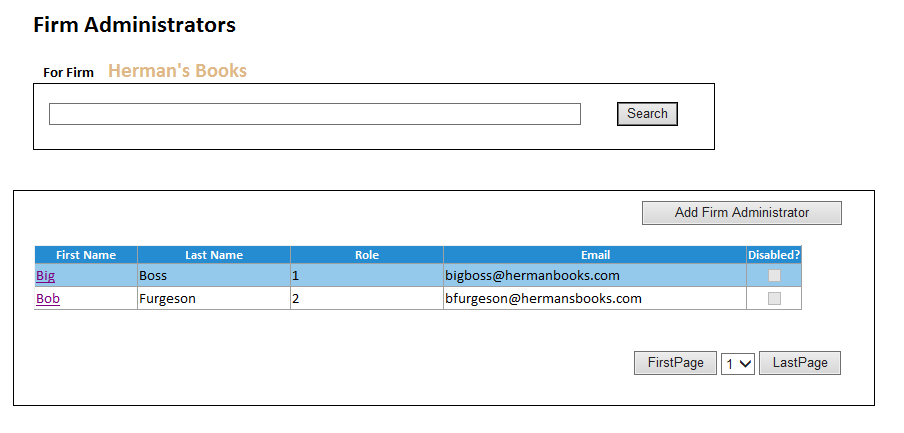


We Get



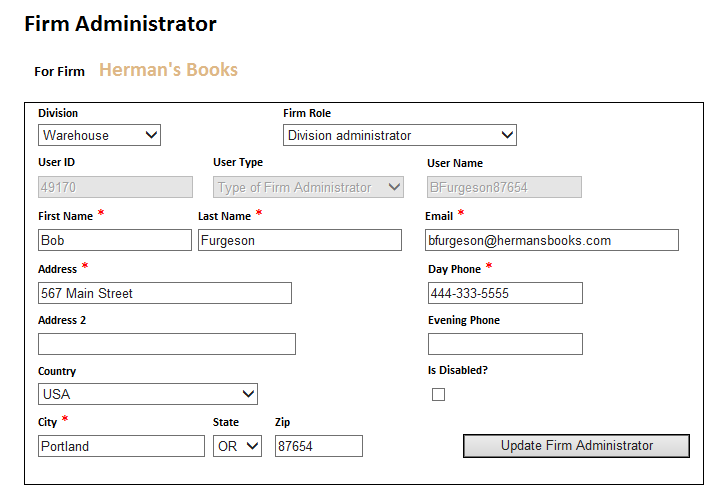
|  |  |
| --- | --- |
|  | Now All of the Children of Firm are accessible via the Navigation tree menu.  We will concentrate on the Admins selection. |

## Firm Administrators



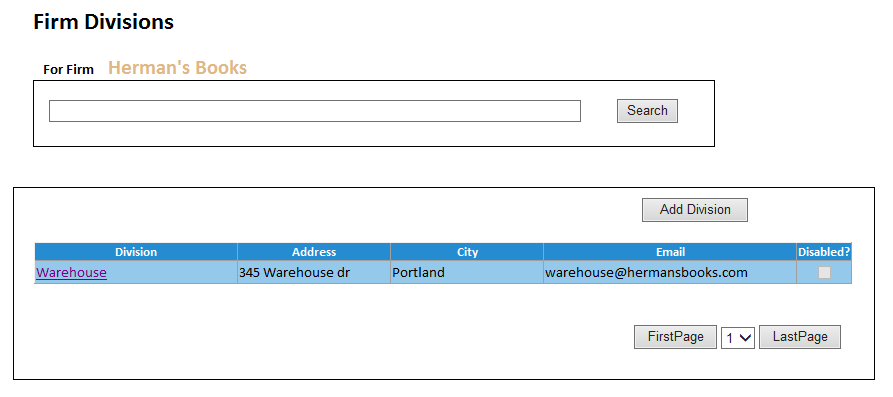
As you can see, there are two items with different Role IDs. Future versions of this grid will have the actual role name that associates with the Role ID. In this case, the Role IDs are for Firm Administrator and Division Administrator. Clicking on either will take you to the Firm Administrator form.

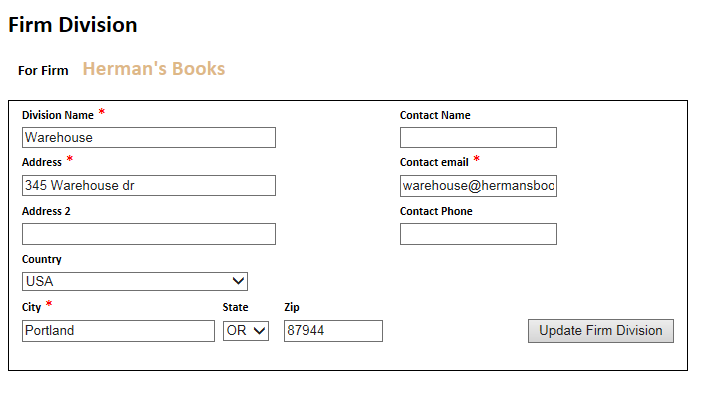
## Firm Administrator



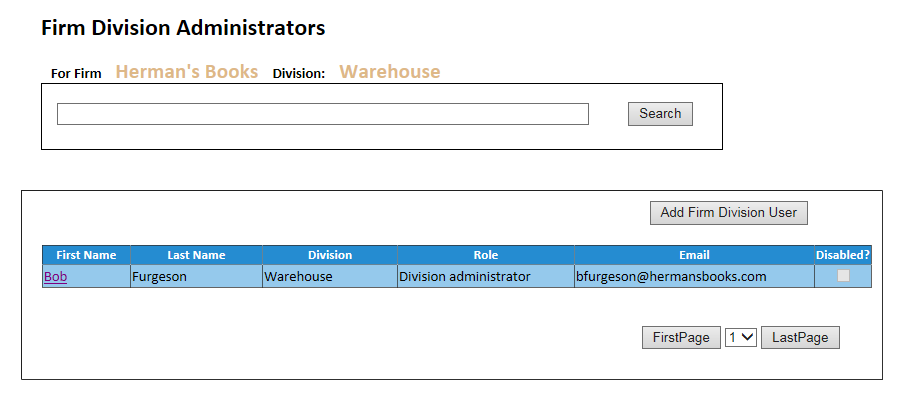
As you can see, this user has been designated as an administrator for the *Warehouse* division of the **Herman’s Books** firm.

|  |  |
| --- | --- |
| Clicking on the Divisions link will give us the next grid for Divisions associated with this firm. |  |

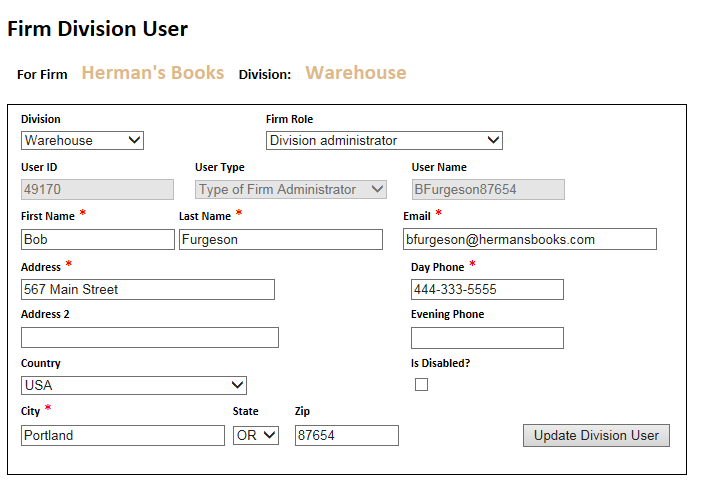
Select the Division



Now you can see the Division Administrators associated with this Firm’s Division.



Clicking on the First Name column item will take you to the user form for this user.



## User Role

You can change this user’s role by selecting a new role from the **Firm Role** drop down list. Once you do this, this user record will no longer be in the *Division Admins* grid. They will still be in the *Division Users* list, unless you change the Division association from the **Division** drop down list.

The following fields are read-only:

* User ID

The User ID field is set automatically for and by the CPAReview Users table (**CPAExam3.dbo.Users**).

* User Type

This is automatically set in the code. If a user is designated as any type of administrator, they will be assigned as Type of Firm Administrator. This allows the user the ability to log into the CPAExam site and be taken to their Firm association.

If a user is designated as a Firm Designated User, they will be automatically assigned a role as a CPAReview user. When they log into the CPAExam site, they will be taken to the normal CPAReview student screen. If they have been assigned to classes, their schedule will show on their screen.

|  |
| --- |
| // have to create a new user                          // UserType should be = 2 (Student)                          // when user is designated as FirmAdministrator - their TypeKey is = 51 (FirmAdminUser)                          // a Firm Administrator cannot be a user in the system - it would not be possible to login                          // to the CPAReview as both a user and a FirmAdministrator                          // If that is a requirement, the FirmAdministrator would need to set up a different user account                          // as either administrator or student - they need to pick...                          //  ddlSalesRep.Items.Add(new ListItem("Unknown / No Rep", "7000"));                          // check with CPAExam3.dbo.UpdateUser stored procedure for synchronization |

* User Name

This is automatically assigned by a business rule in the initial Firm User creation routine. The system will use the following rules to assign a User Name and Password. The user can change these at any time when they log onto CPAExam.

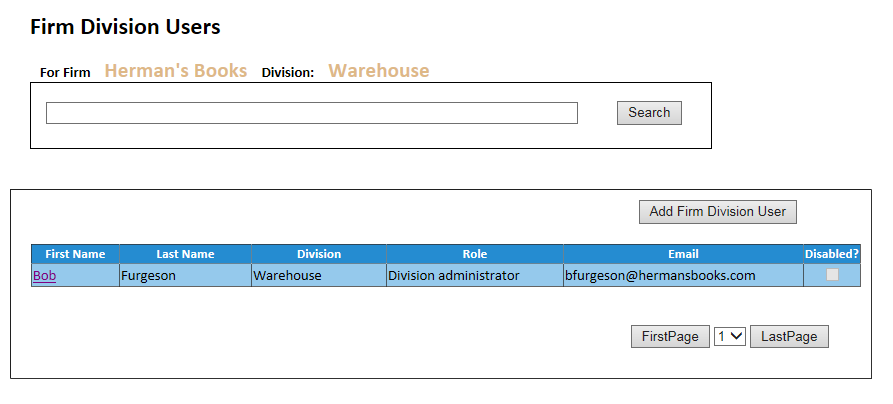
Rule

|  |
| --- |
| // need to check for similar username passwords...    var tmpUserName = tbFirstName.Text.Substring(0, 1) +        tbLastName.Text + tbZip.Text;    // first letter of First Name, last name, zip    // password is initials + "password"    var tmpPass = tbFirstName.Text.Substring(0, 1) +        tbLastName.Text.Substring(0, 1) + "password"; |

(Note – this may be an issue for users designated as Admins. They would not be able to easily change username and/or password as they are not taken to the CPAReview/CPAExam user screen)

### Division Users

The first time you access Division Users, if there are no users, the application will take you to a Division User Form and ask you to create a new Division User. This is true for either Division Admin or Division User. After you have created one new user for the Firm’s Division, you will see a grid of all of the Division’s users.

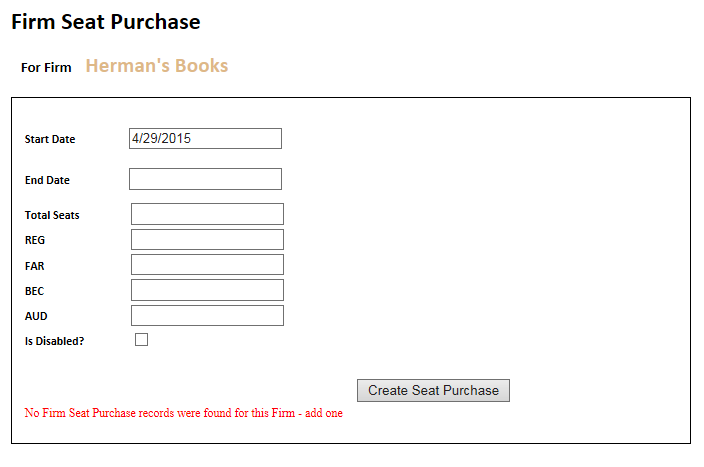


# Task 11503 – Create Functionality for Firm Purchases

The Description of the requirements for this task is as follows:

|  |
| --- |
| Provide the following processes and implement functionality as required:   * Role filtering * Seat assignments, adjusting, and business rules * Linq and/or Stored Procedures * UI Grid * UI Form * Business Rules * Testing |

The implementation of this task was fairly complex from a development standpoint. Containing a mix of Dates and numbers resulted in some advanced business rules. The screens themselves are deceptively simple. For the Firm Purchase form itself, it appears as follows:



|  |  |
| --- | --- |
| The start date is automatically set to Today’s date. You can easily change this by clicking on the date text field and then changing the date using the Calendar control. |  |

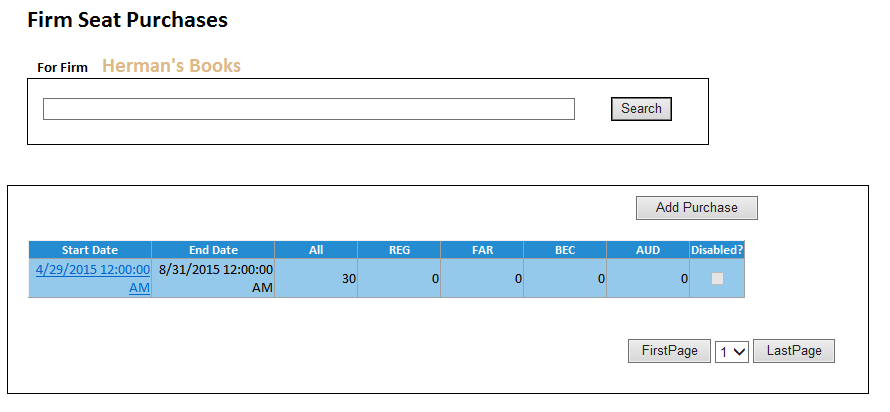
|  |  |
| --- | --- |
|  | The same type of control exists for the End Date. |

If you try to enter an incorrect date (example …a date before the start date) then the system will prompt you with an error message.



|  |  |
| --- | --- |
| If you enter a number into any of the sections – then click the **Create Seat Purchase** button - and do not have a total that is greater than ) and/or is less than the number of seats set in any of the sections – you will get an error message.  If you cause an error, you will not be able to create or update this seat purchase until the error is resolved. |  |

Once you have created at least one Seat Purchase record, you will see the following when you click on the Seat Purchases navigation menu item:



Only a Thomson Reuters administrator can access this area of the application to update everything. A Firm Administrator would only be able to change the sections (REG, FAR, BEC, and AUD).

Once a seat purchase block has been completed, you would be able to assign seats to Firm Users.

1. Mendick, Graham.(2015).grahammendick/navigation.Referenced on April 8, 2015 from from Github. <https://github.com/grahammendick/navigation>. [↑](#endnote-ref-1)
2. *Firm Support – Create Firm – Dev.docx* - Modify CPAReview to support View based on Role login, pp 34-38 [↑](#footnote-ref-1)
3. *Firm Support – Create Firm – Dev.docx – Firm Role Table View – p6* [↑](#footnote-ref-2)