

```
1 using System;
2 using System.Data;
3 using System.Configuration;
4 using System.Collections;
5 using System.Web;
6 using System.Web.Security;
7 using System.Web.UI;
8 using System.Web.UI.WebControls;
9 using System.Web.UI.WebControls.WebParts;
10 using System.Web.UI.HtmlControls;
11
12 public partial class Logon : System.Web.UI.Page
13 {
14
15     protected void Login_Click(object sender, EventArgs e)
16     {
17
18         try
19         {
20             Boolean _loginStatus = false;
21             string _user = txtUserName.Text;
22             string _password = txtPassword.Text;
23
24             //Validate the user
25             if (ValidateUser(_loginStatus, _user, _password))
26             {
27                 //Create an Admin cookie which will expire at sessions end (30 min
28                 sliding) //HttpCookie ARICookie = new HttpCookie("ARIALDCookie", "Admin");
29                 //Response.Cookies.Add(ARICookie);
30                 //Response.SetCookie(ARICookie);
31
32                 //Forms authenticate user with cookie expiration at end of session (30
33                 min sliding) System.Web.Security.FormsAuthentication.RedirectFromLoginPage(_user,
34                 false);
35             }
36         }
37         catch (Exception)
38         {
39             /* Need to setup some email mechanism to return or log errors
40             * lblLoginStatus.Text = ee.ToString();
41             */
42             lblLoginStatus.Text = "An error has occurred. Please try to login later.";
43         }
44     }
45
46     private Boolean ValidateUser(Boolean _loginStatus, string _user, string _password)
47     {
48         try
49         {
50             /*
51             * Declare the Table Adapters and the Data Table for to check if user is
52             in admin table.
53             * Call the table adapter and instancate a new adapter object. Call the
54             * data table first then assign it to a variable .
55             */
56
57             ARITableAdapters.tbl_AdminTableAdapter AdminTableAdapter = new
58             ARITableAdapters.tbl_AdminTableAdapter();
59             ARI.tbl_AdminDataTable AdminTable;
60
61             // Count the user
62             int CountUser;
```

```
1 using System;
2 using System.Data;
3 using System.Configuration;
4 using System.Collections;
5 using System.Web;
6 using System.Web.Security;
7 using System.Web.UI;
8 using System.Web.UI.WebControls;
9 using System.Web.UI.WebControls.WebParts;
10 using System.Web.UI.HtmlControls;
11
12 public partial class admCtl_HelpDesk : System.Web.UI.UserControl
13 {
14     //GVSubmission Index
15     int _gvSubIndex;
16
17     //Store the Admin id for lookup on other views
18     int AdminID;
19
20
21     protected void GVSubmissions_RowDataBound(object sender, GridViewRowEventArgs e)
22     {
23         try
24         {
25             //Trim string data in the GV
26             CommonCode TrimString = new CommonCode();
27
28             //Truncate the text in GV and write the complete text to the tooltip
29             if (e.Row.RowType == DataControlRowType.DataRow)
30             {
31                 Label myComment = (Label)e.Row.FindControl("lblItemComment");
32                 Label mySolution = (Label)e.Row.FindControl("lblItemSolution");
33                 myComment.ToolTip = myComment.Text;
34                 mySolution.ToolTip = mySolution.Text;
35                 string temp = myComment.Text;
36                 string temp2 = mySolution.Text;
37                 int len = temp.Length - 1;
38                 int len2 = temp2.Length - 1;
39                 temp = TrimString.Truncate(temp);
40                 temp2 = TrimString.Truncate(temp2);
41                 myComment.Text = temp;
42                 mySolution.Text = temp2;
43                 temp = null;
44                 temp2 = null;
45
46                 //Capture the index of each row on build of the gridview so we can call the selectindex
47                 //on a button click.
48                 ImageButton butHistory = (ImageButton)e.Row.Cells[7].FindControl("imgButHistory");
49                 butHistory.CommandArgument = _gvSubIndex.ToString();
50                 _gvSubIndex = _gvSubIndex + 1;
51             }
52         }
53         catch (Exception ex)
54         {
55             lblStatus.Text = ex.Message;
56         }
57     }
58
59 }
60
61 protected void GVSubmissions_RowCommand(object sender, GridViewCommandEventArgs e)
62 {
63     try
64     {
65
```

```
66         //Ensure we have a valid session **Now checked from Default page on load.
67         // HttpCookie cookieSessionState = Request.Cookies["ARISessionState"].Value
;
68         // string ARISessionState = Server.HtmlEncode(cookieSessionState.Value);
69         // if (ARISessionState != "Valid")
70         // {
71         //     Response.Redirect("../Login.aspx");
72         // }
73
74         if (e.CommandName == "History")
75         {
76             //Get the current row index
77             int _RowID = Convert.ToInt32(e.CommandArgument);
78
79             //Using the row index get the datakey (RecordID)
80             DataKey myData = GVSubmissions.DataKeys[_RowID];
81
82             //Read the DataKey Value and assign it to the text box.
83             lblID.Text = myData.Values["sub_ID"].ToString();
84
85             //Read the CommandArgument Index assinged on RowDataBound to
86             //use to get the assinged index of the row.
87             int _RowIndex = Convert.ToInt32(e.CommandArgument.ToString());
88             GVSubmissions.SelectedIndex = _RowIndex;
89
90             mvSub.ActiveViewIndex = 2;
91
92         }
93     }
94
95     catch (Exception ex)
96     {
97         lblStatus.Text = ex.Message;
98     }
99
100 }
101
102 protected void GVSubmissions_SelectedIndexChanged(object sender, EventArgs e)
103 {
104     try
105     {
106         string _subID = GVSubmissions.SelectedValue.ToString();
107
108         lblID.Text = _subID;
109
110         ARITableAdapters.tbl_SubmissionsTableAdapter SubmissionTableAdapter = new
ARITableAdapters.tbl_SubmissionsTableAdapter();
111         ARI.tbl_SubmissionsDataTable SubmissionTable;
112
113         SubmissionTable = SubmissionTableAdapter.GetProblemTicketByID(Convert.
ToInt32(_subID));
114         mvSub.ActiveViewIndex = 1;
115     }
116     catch (Exception ex)
117     {
118         lblStatus.Text = ex.Message;
119     }
120
121 }
122
123 protected void GVSubmissions_Sorted(object sender, EventArgs e)
124 {
125     try
126     {
127         //Display an direction arrow on sort
128         switch (GVSubmissions.SortDirection)
129         {
```

```
130         case SortDirection.Ascending:
131             GVSubmissions.Columns[3].HeaderText = "Date <img src='images/down-
blue.gif' border='0' width='14' height='14' />";//_Subdatedown;
132             GVSubmissions.Columns[14].HeaderText = "FAQ <img src='images/down-
blue.gif' border='0' width='14' height='14' />";//_Substatusdown;
133             GVSubmissions.Columns[15].HeaderText = "Status <img src='images/
down-blue.gif' border='0' width='14' height='14' />";//_Substatusdown;
134             break;
135         case SortDirection.Descending:
136             GVSubmissions.Columns[3].HeaderText = "Date <img src='images/up-
blue.gif' border='0' width='14' height='14' />";//_Subdateup;
137             GVSubmissions.Columns[14].HeaderText = "FAQ <img src='images/up-
blue.gif' border='0' width='14' height='14' />";//_Substatusup;
138             GVSubmissions.Columns[15].HeaderText = "Status <img src='images/up-
-blue.gif' border='0' width='14' height='14' />";//_Substatusup;
139             break;
140         default:
141             GVSubmissions.Columns[3].HeaderText = "Date";
142             GVSubmissions.Columns[14].HeaderText = "FAQ";
143             GVSubmissions.Columns[15].HeaderText = "Status";
144             break;
145     }
146 }
147 catch (Exception ex)
148 {
149     lblStatus.Text = ex.Message;
150 }
151 }
152 }
153
154 protected void GVHistory_RowDataBound(object sender, GridViewRowEventArgs e)
155 {
156
157     try
158     {
159         //Trim string data in the GV
160         CommonCode TrimString = new CommonCode();
161
162         //Truncate the text in GV and write the complete text to the tooltip
163         if (e.Row.RowType == DataControlRowType.DataRow)
164         {
165             Label myCommentUSR = (Label)e.Row.FindControl("ItemCommentUSRLabel");
166             Label myCommentsFAQ = (Label)e.Row.FindControl("ItemCommentsFAQLabel");
167
168             Label mySolution = (Label)e.Row.FindControl("ItemSolutionLabel");
169             Label myNote = (Label)e.Row.FindControl("ItemNoteLabel");
170             myCommentUSR.ToolTip = myCommentUSR.Text;
171             myCommentsFAQ.ToolTip = myCommentsFAQ.Text;
172             mySolution.ToolTip = mySolution.Text;
173             myNote.ToolTip = myNote.Text;
174
175             string temp = myCommentUSR.Text;
176             string temp1 = myCommentsFAQ.Text;
177             string temp2 = mySolution.Text;
178             string temp3 = myNote.Text;
179
180             int len = temp.Length - 1;
181             int len1 = temp1.Length - 1;
182             int len2 = temp2.Length - 1;
183             int len3 = temp3.Length - 1;
184
185             temp = TrimString.Truncate(temp);
186             temp1 = TrimString.Truncate(temp1);
187             temp2 = TrimString.Truncate(temp2);
188             temp3 = TrimString.Truncate(temp3);
189
190             myCommentUSR.Text = temp;
```

```
190         myCommentsFAQ.Text = temp1;
191         mySolution.Text = temp2;
192         myNote.Text = temp3;
193         temp = null;
194         temp1 = null;
195         temp2 = null;
196         temp3 = null;
197
198         //Add admin name
199         //Get the adminID from the cookie on the client
200         //AdminID = Convert.ToInt32(Server.HtmlEncode(Request.Cookies[
"ARICookie"]["AdminID"]));
201
202         ARITableAdapters.tbl_AdminTableAdapter AdminTableAdapter = new
ARITableAdapters.tbl_AdminTableAdapter();
203         ARI.tbl_AdminDataTable AdminDataTable;
204
205         Label _AdminID = (Label)e.Row.FindControl("ItemAdminIDLabel");
206         int ID = Convert.ToInt32(_AdminID.Text);
207
208         AdminDataTable = AdminTableAdapter.GetAdminData();
209
210         //Lookup the Admin from the admin table by the AdminID;
211         DataRow _adminRow = AdminDataTable.FindByAdminID(ID);
212
213         Label _AdminName = (Label)e.Row.FindControl("ItemAdminNameLabel");
214
215         //Grab the Admin first name and last name from the datarow
216         string _firstname = _adminRow["AdminFirstName"].ToString();
217         string _lastname = _adminRow["AdminLastName"].ToString();
218
219         _AdminName.Text = _lastname + ", " + _firstname;
220     }
221 }
222 }
223 catch (Exception ex)
224 {
225     lblHistoryStatus.Text = ex.Message;
226 }
227 }
228
229 protected void GVHistory_Sorted(object sender, EventArgs e)
230 {
231     switch (GVHistory.SortDirection)
232     {
233         case SortDirection.Ascending:
234             GVHistory.Columns[2].HeaderText = "AdminID <img src='images/down-blue.
gif' border='0' width='14' height='14' />"; // _Subdatedown;
235             GVHistory.Columns[3].HeaderText = "ModDate <img src='images/down-blue.
gif' border='0' width='14' height='14' />"; // _Substatusdown;
236             GVHistory.Columns[4].HeaderText = "Status <img src='images/down-blue.
gif' border='0' width='14' height='14' />";
237             break;
238         case SortDirection.Descending:
239             GVHistory.Columns[2].HeaderText = "AdminID <img src='images/up-blue.
gif' border='0' width='14' height='14' />"; // _Subdateup;
240             GVHistory.Columns[3].HeaderText = "ModDate <img src='images/up-blue.
gif' border='0' width='14' height='14' />"; // _Substatusup;
241             GVHistory.Columns[4].HeaderText = "Status <img src='images/up-blue.gif
' border='0' width='14' height='14' />"; // _Substatusup;
242             break;
243         default:
244             GVHistory.Columns[2].HeaderText = "AdminID";
245             GVHistory.Columns[3].HeaderText = "ModDate";
246             GVHistory.Columns[4].HeaderText = "Status";
247             break;
248     }
}
```

```
249     }
250
251     protected void DVSubmissions_DataBound(object sender, EventArgs e)
252     {
253         try
254         {
255             //Get the adminID from the cookie on the client
256             AdminID = Convert.ToInt32(Server.HtmlEncode(Request.Cookies["ARICookie"]["AdminID"]));
257
258             ARITableAdapters.tbl_AdminTableAdapter AdminTableAdapter = new ARITableAdapters.tbl_AdminTableAdapter();
259             ARI.tbl_AdminDataTable AdminDataTable;
260
261             AdminDataTable = AdminTableAdapter.GetAdminData();
262
263             //Lookup the Admin from the admin table by the AdminID;
264             DataRow _adminRow = AdminDataTable.FindByAdminID(AdminID);
265
266             if (DVSubmissions.CurrentMode == DetailsViewMode.Edit)
267             {
268                 //Find and lookup admin by id and display and write date to the DB
269                 Label myAdminID = (Label)DVSubmissions.Rows[0].Cells[0].FindControl("EditAdminIDLabel");
270                 Label myModDate = (Label)DVSubmissions.Rows[0].Cells[0].FindControl("EditModDateLabel");
271
272                 myAdminID.Text = AdminID.ToString();
273                 myModDate.Text = DateTime.Now.ToString();
274
275                 //Grab the Admin first name and last name from the datarow
276                 string _firstname = _adminRow["AdminFirstName"].ToString();
277                 string _lastname = _adminRow["AdminLastName"].ToString();
278
279                 Label myAdmin = (Label)DVSubmissions.Rows[0].Cells[0].FindControl("EditAdminNameLabel");
280
281                 myAdmin.Text = _lastname + ", " + _firstname;
282             }
283             else if (DVSubmissions.CurrentMode == DetailsViewMode.ReadOnly)
284             {
285                 //Show some fields we want to see in read mode
286                 DVSubmissions.Fields[10].Visible = true;
287                 DVSubmissions.Fields[11].Visible = true;
288                 DVSubmissions.Fields[14].Visible = true;
289                 DVSubmissions.Fields[18].Visible = true;
290                 DVSubmissions.Fields[25].Visible = true;
291                 DVSubmissions.Fields[26].Visible = true;
292
293                 //DVSubmissions.Caption = "<h1>Submission Details</h1>"; //Change the caption for the header
294
295                 //lookup admin by ID from the DB and display admin name.
296                 string _firstname = _adminRow["AdminFirstName"].ToString();
297                 string _lastname = _adminRow["AdminLastName"].ToString();
298
299                 Label myAdmin = (Label)DVSubmissions.FindControl("ItemAdminNameLabel");
300
301                 myAdmin.Text = _lastname + ", " + _firstname;
302             }
303         }
304         catch (Exception ex)
305         {
306             lblSubmissionStatus.Text = ex.Message;
307         }
308     }
```

```
309
310 protected void DVSubmissions_ItemUpdating(object sender,
311     DetailsViewUpdateEventArgs e)
312 {
313     try
314     {
315         if (DVSubmissions.CurrentMode == DetailsViewMode.Edit)
316         {
317             //Prep to write to the history table
318             ARITableAdapters.tbl_HistoryTableAdapter HistoryTableAdapter = new
319                 ARITableAdapters.tbl_HistoryTableAdapter();
320
321             Label _TicketID = (Label)DVSubmissions.Rows[0].Cells[0].FindControl(
322                 "EditTicketIDLabel");
323             Label _AdminID = (Label)DVSubmissions.Rows[0].Cells[0].FindControl(
324                 "EditAdminIDLabel");
325             Label _ModDate = (Label)DVSubmissions.Rows[0].Cells[0].FindControl(
326                 "EditModDateLabel");
327             Label _CommentsUSR = (Label)DVSubmissions.Rows[0].Cells[0].FindControl(
328                 "EditCommentsUSRLabel");
329             TextBox _CommentsFAQ = (TextBox)DVSubmissions.Rows[0].Cells[0].
330                 FindControl("EditCommentsFAQTextBox");
331             TextBox _Solution = (TextBox)DVSubmissions.Rows[0].Cells[0].
332                 FindControl("EditSolutionTextBox");
333             TextBox _CCMail = (TextBox)DVSubmissions.Rows[0].Cells[0].FindControl(
334                 "EditCCMailTextBox");
335             DropDownList _Status = (DropDownList)DVSubmissions.Rows[0].Cells[0].
336                 FindControl("ddlStatus");
337             TextBox _Note = (TextBox)DVSubmissions.Rows[0].Cells[0].FindControl(
338                 "EditNoteTextBox");
339
340             if (_AdminID.Text != null)
341             {
342                 int TicketID = Convert.ToInt32(_TicketID.Text);
343                 int AdminID = Convert.ToInt32(_AdminID.Text);
344                 DateTime ModDate = Convert.ToDateTime(_ModDate.Text);
345
346                 //Write to the history table
347                 HistoryTableAdapter.Insert(TicketID, AdminID, ModDate,
348                     _CommentsFAQ.Text, _Solution.Text, _CCMail.Text, _Status.SelectedValue, _Note.Text,
349                     _CommentsUSR.Text);
350             }
351         }
352     }
353     catch (Exception ex)
354     {
355         lblSubmissionStatus.Text = ex.Message;
356     }
357 }
358
359 protected void DVSubmissions_ItemUpdated(object sender,
360     DetailsViewUpdatedEventArgs e)
361 {
362     try
363     {
364         //Details view updated - rebind the GV
365         GVSubmissions.DataBind();
366     }
367     catch (Exception ex)
368     {
369         lblSubmissionStatus.Text = ex.Message;
370     }
371 }
372
373 protected void DVSubmissions_ItemDeleted(object sender,
374     DetailsViewDeletedEventArgs e)
```

```
361     {
362         try
363         {
364             //DV updated - rebind the GV
365             GVSubmissions.DataBind();
366         }
367         catch (Exception ex)
368         {
369             lblSubmissionStatus.Text = ex.Message;
370         }
371     }
372
373     protected void DVSubmissions_ItemCommand(object sender,           ↙
374     DetailsViewCommandEventArgs e)
375     {
376         try
377         {
378             //User closed - reset index to activeview 0
379             if (e.CommandName == "Close")
380             {
381                 mvSub.ActiveViewIndex = 0;
382             }
383             //Hide some fields we don't want to see in edit mode
384             else if (e.CommandName == "Edit")//Respond
385             {
386                 DVSubmissions.Fields[10].Visible = false;
387                 DVSubmissions.Fields[11].Visible = false;
388                 DVSubmissions.Fields[14].Visible = false;
389                 DVSubmissions.Fields[18].Visible = false;
390                 DVSubmissions.Fields[25].Visible = false;
391                 DVSubmissions.Fields[26].Visible = false;
392
393                 //Reset the caption to Response Details
394                 DVSubmissions.Caption = "<h1>Response Details</h1>"; //change the   ↙
395                 caption for the header
396             }
397             //Update the database a send mail
398             else if (e.CommandName == "Update")//Send Mail
399             {
400                 Mail_Util mail = new Mail_Util();
401
402                 ARITableAdapters.tbl_AdminTableAdapter AdminTableAdapter = new   ↙
403                 ARITableAdapters.tbl_AdminTableAdapter();
404                 ARI.tbl_AdminDataTable AdminDataTable;
405
406                 AdminDataTable = AdminTableAdapter.GetAdminData();
407
408                 //Lookup the Admin from the admin table by the AdminID;
409                 DataRow _adminRow = AdminDataTable.FindByAdminID(AdminID);
410
411                 string _AdminFirstName = _adminRow["AdminFirstName"].ToString();
412                 string _AdminLastName = _adminRow["AdminLastName"].ToString();   ↙
413
414                 string _AdminName = _AdminLastName + ", " + _AdminFirstName;
415                 Label _TicketID = (Label)DVSubmissions.Rows[0].Cells[0].FindControl(   ↙
416                 "EditTicketIDLabel");
417                 Label _FirstName = (Label)DVSubmissions.Rows[0].Cells[0].FindControl(   ↙
418                 "EditFirstNameLabel");
419                 Label _LastName = (Label)DVSubmissions.Rows[0].Cells[0].FindControl(   ↙
420                 "EditLastNameLabel");
421                 Label _EmailTo = (Label)DVSubmissions.Rows[0].Cells[0].FindControl(   ↙
422                 "EditEmailLabel");
423                 TextBox _CCEmail = (TextBox)DVSubmissions.Rows[0].Cells[0].FindControl(   ↙
424                 ("EditCCMailTextBox"));
425                 Label _Subject = (Label)DVSubmissions.Rows[0].Cells[0].FindControl(   ↙
```

```

    "EditSubjectLabel");
419     Label _Body1 = (Label)DVSubmissions.Rows[0].Cells[0].FindControl(
    "EditCommentsUsrLabel");
420     TextBox _Body2 = (TextBox)DVSubmissions.Rows[0].Cells[0].FindControl(
    "EditNoteTextBox");
421     String _BodyFormatted = "Your Comments: " + _Body1.Text + "<br /><br
/>Response: " + _Body2.Text;
422     String _NameFormatted = _FirstName.Text + " " + _LastName.Text;
423     string EmailLink = ConfigurationManager.AppSettings["AppURL"] +
    "TicketStatus.aspx?Key1=" + _TicketID.Text;
424
425     mail.ARiMailer("Response", _EmailTo.Text, "admin@arildru.net",
    _AdminName, _Subject.Text, _BodyFormatted, EmailLink, _CCEmail.Text);
426     string mailstatus = mail.ARIMailStatus;
427
428     //Write alert to client browser on mailstatus
429     lblSubmissionStatus.Text = "<script>alert('" + mailstatus + "');</
script>"; // mail.ARIMailStatus;
430
431     //Set the activeview index to 3
432     else if (e.CommandName == "SubmitRemoveFAQ")
433     {
434         mvSub.ActiveViewIndex = 3;
435     }
436 }
437 catch (Exception ex)
438 {
439     lblSubmissionStatus.Text = ex.Message;
440 }
441
442 }
443
444 protected void imgBtnCloseHistory_Click(object sender, ImageClickEventArgs e)
445 {
446     try
447     {
448         mvSub.ActiveViewIndex = 0;
449     }
450     catch (Exception ex)
451     {
452
453         lblHistoryStatus.Text = ex.Message;
454     }
455 }
456
457 protected void ddlCategory_SelectedIndexChanged(object sender, EventArgs e)
458 {
459     try
460     {
461         if (DVFAQDetails.CurrentMode == DetailsViewMode.Edit)
462         {
463             //Update the DB from the users selection in the combo box
464             TextBox myCategory = (TextBox)DVFAQDetails.Rows[0].Cells[0].
FindControl("EditCategoryTextBox");
465             DropDownList myDDLCategory = (DropDownList)DVFAQDetails.Rows[0].Cells
[0].FindControl("ddlCategory");
466
467             myCategory.Text = myDDLCategory.SelectedItem.Text;
468             myDDLCategory.SelectedItem.Text = myCategory.Text;
469         }
470     }
471     catch (Exception ex)
472     {
473         lblFAQDetailsStatus.Text = ex.Message;
474     }
475 }
476
```

```
477     protected void ddlCategory_DataBound(object sender, EventArgs e)
478     {
479         try
480         {
481             if (DVFAQDetails.CurrentMode == DetailsViewMode.Edit)
482             {
483                 //After the databound - assign the value from the DB of the CR to the
484                 DDL
485                 TextBox myCategory = (TextBox)DVFAQDetails.Rows[0].Cells[0].
486                 FindControl("EditCategoryTextBox");
487                 DropDownList myDDLCategory = (DropDownList)DVFAQDetails.Rows[0].Cells
488                 [0].FindControl("ddlCategory");
489                 //myDDLCategory.Items.Insert(0, new ListItem("Select Category", "0"));
490                 if (myCategory.Text != "")
491                 {
492                     string _currentVal = myCategory.Text;
493                     ListItem item = myDDLCategory.Items.FindByText(_currentVal);
494                     myDDLCategory.SelectedItem.Selected = false;
495                     item.Selected = true;
496                 }
497             }
498             catch (Exception ex)
499             {
500                 lblFAQDetailsStatus.Text = ex.Message;
501             }
502         }
503     }
504     protected void ImgBtnClose_Click(object sender, ImageClickEventArgs e)
505     {
506         try
507         {
508             mvSub.ActiveViewIndex = 0;
509         }
510         catch (Exception ex)
511         {
512             lblFAQDetailsStatus.Text = ex.Message;
513         }
514     }
515     protected void ImgBtnUpdate_Click(object sender, ImageClickEventArgs e)
516     {
517         try
518         {
519             mvSub.ActiveViewIndex = 0;
520         }
521         catch (Exception ex)
522         {
523             lblFAQDetailsStatus.Text = ex.Message;
524         }
525     }
526     protected void DVFAQDetails_ItemUpdated(object sender, DetailsViewUpdatedEventArgs
527     e)
528     {
529         try
530         {
531             GVSubmissions.DataBind();
532         }
533         catch (Exception ex)
534         {
535             lblFAQDetailsStatus.Text = ex.Message;
536         }
537     }
538     protected void EditFAQDateLabel_DataBinding(object sender, EventArgs e)
```

```
540     {
541         try
542         {
543             if (DVFAQDetails.CurrentMode == DetailsViewMode.Edit)
544             {
545                 //Update the DV with the current date / time for db update
546                 Label _FAQDate = (Label)DVFAQDetails.FindControl("EditFAQDateLabel");
547                 DateTime FAQDate = DateTime.Now;
548                 _FAQDate.Text = FAQDate.ToString();
549             }
550         }
551         catch (Exception ex)
552         {
553             lblFAQDetailsStatus.Text = ex.Message;
554         }
555     }
556
557     protected void EditFAQCheckBox_DataBinding(object sender, EventArgs e)
558     {
559         try
560         {
561             //Set the Radio button to the default of the value read from the DB
562             CheckBox _ChkFAQ = (CheckBox)DVFAQDetails.FindControl("EditFAQCheckBox");
563             RadioButtonList _YesNoFAQ = (RadioButtonList)DVFAQDetails.FindControl("EditFAQRadioButton");
564
565             if (_ChkFAQ.Text == "True") _YesNoFAQ.SelectedIndex = 0;
566             else if (_ChkFAQ.Text == "False") _YesNoFAQ.SelectedIndex = 1;
567             else _YesNoFAQ.SelectedIndex = 1;
568         }
569         catch (Exception ex)
570         {
571             lblFAQDetailsStatus.Text = ex.Message;
572         }
573     }
574
575     protected void EditFAQRadioButton_SelectedIndexChanged(object sender, EventArgs e)
576     {
577         try
578         {
579             //Update the DB with the users new selected option
580             CheckBox _ChkFAQ = (CheckBox)DVFAQDetails.FindControl("EditFAQCheckBox");
581             RadioButtonList _YesNoFAQ = (RadioButtonList)DVFAQDetails.FindControl("EditFAQRadioButton");
582
583             if (_YesNoFAQ.SelectedIndex == 0) _ChkFAQ.Text = "True";
584             else if (_YesNoFAQ.SelectedIndex == 1) _ChkFAQ.Text = "False";
585         }
586         catch (Exception ex)
587         {
588             lblFAQDetailsStatus.Text = ex.Message;
589         }
590     }
591 }
592
593     protected void DVFAQDetails_DataBound(object sender, EventArgs e)
594     {
595         try
596         {
597             ARITableAdapters.tbl_AdminTableAdapter AdminTableAdapter = new
598             ARITableAdapters.tbl_AdminTableAdapter();
599             ARI.tbl_AdminDataTable AdminDataTable;
600
601             AdminDataTable = AdminTableAdapter.GetAdminData();
602
603             DataRow _adminRow = AdminDataTable.FindByAdminID(AdminID);
```

```
604         //lookup admin by ID and display admin name.
605         string _firstname = _adminRow["AdminFirstName"].ToString();
606         string _lastname = _adminRow["AdminLastName"].ToString();
607
608         Label myAdmin = (Label)DVFAQDetails.FindControl("EditAdminNameLabel");
609         myAdmin.Text = _lastname + ", " + _firstname;
610
611     }
612     catch (Exception ex)
613     {
614         lblFAQDetailsStatus.Text = ex.Message;
615     }
616 }
617
618 protected void EditAdminIDLabel_DataBinding(object sender, EventArgs e)
619 {
620     try
621     {
622         if (DVFAQDetails.CurrentMode == DetailsViewMode.Edit)
623         {
624             Label myAdminID = (Label)DVFAQDetails.FindControl("EditAdminIDLabel");
625             myAdminID.Text = AdminID.ToString();
626         }
627     }
628     catch (Exception ex)
629     {
630         lblFAQDetailsStatus.Text = ex.Message;
631     }
632 }
633
634 protected void EditCancelImgBtn_Command(object sender, CommandEventArgs e)
635 {
636     try
637     {
638         if (e.CommandName == "Cancel") mvSub.ActiveViewIndex = 0;
639     }
640     catch (Exception ex)
641     {
642         lblFAQDetailsStatus.Text = ex.Message;
643     }
644 }
645
646 }
647
648 protected void lnkBtnLogout_Click(object sender, EventArgs e)
649 {
650     try
651     {
652         HttpCookie ARICookie = Request.Cookies.Get("ARICookie");
653         ARICookie.Expires = DateTime.Now.AddDays(-1);
654         Response.Cookies.Add(ARICookie);
655         Response.SetCookie(ARICookie);
656         Response.Redirect(ConfigurationManager.AppSettings["AppURL"]);
657     }
658     catch (Exception ex)
659     {
660         lblAdminStatus.Text = ex.Message;
661     }
662 }
663
664 }
665
666 protected void imgButNewSubmission_Click(object sender, ImageClickEventArgs e)
667 {
668     mvSub.ActiveViewIndex = 4;
669 }
670
671 protected void FVSubmission_ItemInserting(object sender, FormViewInsertEventArgs e) ↵
```

```
e)
{
671     {
672         try
673         {
674             //Get the adminID from the cookie on the client
675             AdminID = Convert.ToInt32(Server.HtmlEncode(Request.Cookies["ARICookie"]["AdminID"]));
676
677             ARITableAdapters.tbl_AdminTableAdapter AdminTableAdapter = new ARITableAdapters.tbl_AdminTableAdapter();
678             ARITableAdapters.tbl_AdminTableAdapter();
679             ARI.tbl_AdminDataTable AdminDataTable;
680             AdminDataTable = AdminTableAdapter.GetAdminData();
681
682             //Lookup the Admin from the admin table by the AdminID;
683             DataRow _adminRow = AdminDataTable.FindByAdminID(AdminID);
684
685             ARITableAdapters.tbl_SubmissionsTableAdapter SubmissionsTableAdapter = new ARITableAdapters.tbl_SubmissionsTableAdapter();
686             ARITableAdapters.tbl_SubmissionsTableAdapter();
687             ARI.tbl_SubmissionsDataTable SubmissionsTable;
688             SubmissionsTable = SubmissionsTableAdapter.GetClientContactSubData();
689
690             TextBox my_subFirstName = (TextBox)FVSubmission.Row.FindControl("sub_FirstNameTextBox");
691             TextBox my_subLastName = (TextBox)FVSubmission.Row.FindControl("sub_LastNameTextBox");
692             TextBox my_subEmail = (TextBox)FVSubmission.Row.FindControl("sub_EmailTextBox");
693             TextBox my_subTele = (TextBox)FVSubmission.Row.FindControl("sub_TeleTextBox");
694             TextBox my_subLocation = (TextBox)FVSubmission.Row.FindControl("sub_LocationTextBox");
695             TextBox my_subSubject = (TextBox)FVSubmission.Row.FindControl("sub_SubjectTextBox");
696             TextBox my_subComments = (TextBox)FVSubmission.Row.FindControl("sub_CommentsTextBox");
697
698             DateTime CurrentDate = DateTime.Now;
699             my_subFirstName.Text = _adminRow["AdminFirstName"].ToString();
700             my_subLastName.Text = _adminRow["AdminLastName"].ToString();
701             my_subEmail.Text = _adminRow["AdminEmail"].ToString();
702
703             int? subID = 0;
704
705             SubmissionsTableAdapter.InsertProblemTicket(CurrentDate, "ProblemTicket", my_subFirstName.Text, my_subLastName.Text, my_subEmail.Text, my_subTele.Text, my_subLocation.Text, my_subSubject.Text, my_subComments.Text, "Open", ref subID);
706
707             my_subEmail.Text = null;
708             my_subFirstName.Text = null;
709             my_subLastName.Text = null;
710             my_subComments.Text = null;
711             my_subTele.Text = null;
712             my_subSubject.Text = null;
713             my_subLocation.Text = null;
714
715
716             GVSubmissions.DataBind();
717             mvSub.ActiveViewIndex = 0;
718
719         }
720         catch (Exception ex)
721         {
722             lblFVSubmissionStatus.Text = ex.Message;
723         }
724     }
}
```

```
725
726     protected void FVSubmission_ModeChanging(object sender, FormViewModeEventArgs e)
727     {
728         // Needed for the event e.Cancel
729     }
730
731     protected void InsertCancelButton1_Command(object sender, CommandEventArgs e)
732     {
733         mvSub.ActiveViewIndex = 0;
734     }
735 }
736 }
737
```

VisuMAX ©2008

```
63         //Return the row (if exist) where we have a user with the assigned password
64         AdminTable = AdminTableAdapter.GetDataByNameAndPassword(_user, _password);
65
66         //Get the number of rows returned
67         CountUser = AdminTable.Count;
68
69         //If we don't have at least one return false
70         if (!(CountUser == 1))
71         {
72             lblLoginStatus.Text = "Invalid Login!";
73             return false;
74         }
75         else
76         {
77             // We have at least one row - Store some session info and return true
78             int _AdminID = AdminTable[0].AdminID;
79             string _FirstName = AdminTable[0].AdminFirstName;
80             string _LastName = AdminTable[0].AdminLastName;
81             string _Email = AdminTable[0].AdminEmail;
82
83             //Create an Admin cookie which will expire at sessions end (30 min sliding)
84             HttpCookie ARICookie = new HttpCookie("ARICookie", "Admin");
85             ARICookie.Expires = DateTime.Now.AddDays(1);
86             ARICookie.Values["AdminID"] = _AdminID.ToString(); ;
87             ARICookie.Values["AdminFirstName"] = _FirstName;
88             ARICookie.Values["AdminLastName"] = _LastName;
89             ARICookie.Values["AdminEmail"] = _Email;
90             ARICookie.Values["ARISessionState"] = "Valid";
91             Response.Cookies.Add(ARICookie);
92             Response.SetCookie(ARICookie);
93
94             //Session.Add("AdminID", _AdminID.ToString());
95             //Session.Add("AdminFirstName", _FirstName);
96             //Session.Add("AdminLastName", _LastName);
97             //Session.Add("AdminEmail", _Email);
98             //Session.Add("ARISessionState", "Valid");
99             return true;
100         }
101     }
102 }
103
104 catch (Exception ex)
105 {
106     /* Need to setup some email mechanism to return or log errors
107     * lblLoginStatus.Text = ee.ToString();
108     */
109     lblLoginStatus.Text = ex.Message + " <br />An error has occurred. Please
110     try to login later.";
111     return false;
112 }
113 }
114 }
115 }
```

```
1 using System;
2 using System.Data;
3 using System.Configuration;
4 using System.Collections;
5 using System.Web;
6 using System.Web.Security;
7 using System.Web.UI;
8 using System.Web.UI.WebControls;
9 using System.Web.UI.WebControls.WebParts;
10 using System.Web.UI.HtmlControls;
11
12 public partial class usrCtl_ProblemTicket : System.Web.UI.UserControl
13 {
14     protected void fvProblemTicket_ItemInserting(object sender,
15     FormViewInsertEventArgs e)
16     {
17         try
18         {
19             Mail_Util mail = new Mail_Util();
20             string EmailSupport = ConfigurationManager.AppSettings["ContactEmail"];
21             string EmailSupportName = ConfigurationManager.AppSettings["ContactName"];
22
23             //instanciate problem ticket table
24             ARITableAdapters.tbl_SubmissionsTableAdapter SubmissionsTableAdapter = new
25             ARITableAdapters.tbl_SubmissionsTableAdapter();
26             ARI.tbl_SubmissionsDataTable SubmissionsTable;
27
28             SubmissionsTable = SubmissionsTableAdapter.GetClientContactSubData();
29
30             TextBox my_subFirstName = (TextBox)fvHelpDeskTicket.Row.FindControl(
31             "sub_FirstNameTextBox");
32             TextBox my_subLastName = (TextBox)fvHelpDeskTicket.Row.FindControl(
33             "sub_LastNameTextBox");
34             TextBox my_subEmail = (TextBox)fvHelpDeskTicket.Row.FindControl(
35             "sub_EmailTextBox");
36             TextBox my_subTele = (TextBox)fvHelpDeskTicket.Row.FindControl(
37             "sub_TeleTextBox");
38             TextBox my_subLocation = (TextBox)fvHelpDeskTicket.Row.FindControl(
39             "sub_LocationTextBox");
40             TextBox my_subSubject = (TextBox)fvHelpDeskTicket.Row.FindControl(
41             "sub_SubjectTextBox");
42             TextBox my_subComments = (TextBox)fvHelpDeskTicket.Row.FindControl(
43             "sub_CommentsTextBox");
44
45             DateTime CurrentDate = DateTime.Now;
46
47             int? subID = 0;
48
49             //ProblemTicketTableAdapter.InsertProblemTicket("ContactUs", "Open",
50             my_subFirstName.Text, my_subLastName.Text, my_subEmail.Text, my_subTele.Text,
51             my_subLocation.Text, my_subSubject.Text, my_subComments.Text, CurrentDate, ref
52             subID);
53
54             SubmissionsTableAdapter.InsertProblemTicket(CurrentDate, "ProblemTicket",
55             my_subFirstName.Text, my_subLastName.Text, my_subEmail.Text, my_subTele.Text,
56             my_subLocation.Text, my_subSubject.Text, my_subComments.Text, "Open", ref
57             subID);
58
59             //SubmissionsTableAdapter.Insert(CurrentDate, "ProblemTicket",
60             my_subFirstName.Text, my_subLastName.Text, my_subEmail.Text, my_subTele.Text,
61             my_subLocation.Text, my_subSubject.Text, my_subComments.Text, "Open");
62
63 #region Submitter email
64
65     //Send email to submitter
66     string EmailType = "Generic";
67     string EmailTo = my_subEmail.Text;
68     string EmailFrom = EmailSupport; // "admin@ARILDROU.NET";
69     string EmailName = EmailSupportName;
70     string EmailSubject = "Your submission was received";
71 }
```

```
52     string EmailBody = "<font size=\"2\" face=\"Verdana, Arial, Helvetica,
    sans-serif\"><p>Date: " + CurrentDate + "</p></font>";
53     EmailBody = EmailBody + "<font size=\"2\" face=\"Verdana, Arial, Helvetica
    , sans-serif\">You submitted the following Help Desk Ticket. <p> Submission:" +
    my_subComments.Text +
54         "<br /><br />Thank you for your submission. We will be in contact with
    you. You may track the status of your ticket from the link below.</p></font>";
55     string EmailLink = ConfigurationManager.AppSettings["AppURL"] +
    "TicketStatus.aspx?Key1=" + subID;
56
57     mail.ARMailer(EmailType, EmailTo, EmailFrom, EmailName, EmailSubject,
    EmailBody, EmailLink, null);
58     lblStatus.Text = mail.ARMailStatus;
59
60 #endregion
61
62 #region Receptient email
63
64     // Send email to receipient
65     EmailType = "ContactUs";
66     EmailTo = EmailSupport;
67     EmailFrom = my_subEmail.Text;//"admin@ARILDRU.NET";
68     EmailName = my_subFirstName.Text + " " + my_subLastName.Text;
69     EmailSubject = " Subject: " + my_subSubject.Text + "."; // "Contact Us:
    Submitted by: " + my_subEmail.Text +
70     EmailBody = "<font size=\"2\" face=\"Verdana, Arial, Helvetica, sans-serif
    \"><p>Date: " + CurrentDate + "</p></font>";
71     EmailBody = EmailBody + "<font size=\"2\" face=\"Verdana, Arial, Helvetica
    , sans-serif\"><p>The following Help Desk Ticket was just submitted. <br /><br />
    Submission:<br />" + my_subComments.Text + "</p></font>";
72     EmailLink = ConfigurationManager.AppSettings["AppURL"] + "Admin/Default.
    aspx"; // "Admin/ContactUs.aspx?ID=" + ProblemID;
73
74     mail.ARMailer(EmailType, EmailTo, EmailFrom, EmailName, EmailSubject,
    EmailBody, EmailLink, null);
75
76 #endregion Receptient
77
78     my_subEmail.Text = null;
79     my_subFirstName.Text = null;
80     my_subLastName.Text = null;
81     my_subComments.Text = null;
82     my_subTele.Text = null;
83     my_subSubject.Text = null;
84     my_subLocation.Text = null;
85     fvHelpDeskTicket.Visible = false;
86 }
87 catch (Exception ex)
88 {
89     lblStatus.Text = ex.Message;
90 }
91
92 }
93
94 protected void fvHelpDeskTicket_ModeChanging(object sender, FormViewModeEventArgs
    e)
95 {
96     // Needed for the event e.Cancel
97
98 }
99
100 protected void imgButCancelHelp_Click(object sender, ImageClickEventArgs e)
101 {
102     Response.Redirect(ConfigurationManager.AppSettings["AppURL"]);
103 }
104 protected void fvHelpDeskTicket_PageIndexChanging(object sender,
    FormViewPageEventArgs e)
```

```
105 {  
106  
107 }  
108 }  
109
```

VisuMAX ©2008

```

1 using System;
2 using System.Data;
3 using System.Configuration;
4 using System.Collections;
5 using System.Web;
6 using System.Web.Security;
7 using System.Web.UI;
8 using System.Web.UI.WebControls;
9 using System.Web.UI.WebControls.WebParts;
10 using System.Web.UI.HtmlControls;
11
12 public partial class User_Ctls_ursCtl_FAQs : System.Web.UI.UserControl
13 {
14     protected void Page_Load(object sender, EventArgs e)
15     {
16         try
17         {
18             ARITableAdapters.tbl_CategoryTableAdapter CategoryTableAdapter = new
19             ARITableAdapters.tbl_CategoryTableAdapter();
20             ARI.tbl_CategoryDataTable CategoryDataTable;
21
22             CategoryDataTable = CategoryTableAdapter.GetCategoryData();
23
24             ARITableAdapters.tbl_Submissions_FAQTableAdapter FAQTableAdapter = new
25             ARITableAdapters.tbl_Submissions_FAQTableAdapter();
26             ARI.tbl_Submissions_FAQDataTable FAQDataTable;
27
28             //Loop through each category and populate the placeholder
29             foreach (ARI.tbl_CategoryRow CatRow in CategoryDataTable)
30             {
31                 string _Category;
32                 Label CategoryName = new Label();
33                 _Category = CatRow.cat_Name;
34                 CategoryName.Text = "<h1><p>" + CatRow.cat_Name + "</h1></p>";
35                 plcHolderContent.Controls.Add(CategoryName);
36                 Label HTMLBreak = new Label();
37                 HTMLBreak.Text = "<p>&nbsp; </p>";
38
39                 FAQDataTable = FAQTableAdapter.GetDataByCategory(_Category);
40
41                 //For each category - Loop through a populate the placeholder with the
42                 faq info
43                 foreach (ARI.tbl_Submissions_FAQRow FAQRow in FAQDataTable)
44                 {
45                     Label Subject = new Label();
46
47                     if (FAQRow.Issub_SubjectNull())//test dataset field for null
48                     {
49                         Subject.Text = "<p><strong>Subject: </strong>No Subject</p>";
50                     }
51                     else
52                     {
53                         Subject.Text = "<p><strong>Subject: </strong>" + FAQRow.
54                         sub_Subject + "</p>";
55                     }
56                     plcHolderContent.Controls.Add(Subject);
57
58                     Label CommentFAQ = new Label();
59
60                     if (FAQRow.Issub_CommentsFAQNull())//test dataset field for null
61                     {
62                         CommentFAQ.Text = "<p><strong>Comments:</strong> No FAQ
63                         Comments Entered </p>";
64                     }
65                     else
66                     {
67                         CommentFAQ.Text = "<p><strong>Comments: </strong>" + FAQRow.

```

```
63     sub_CommentsFAQ + "</p>";
64     }
65     plcHolderContent.Controls.Add(CommentFAQ);
66     Label SolutionFAQ = new Label();
67
68     if (FAQRow.Issub_SolutionNull())
69     {
70         SolutionFAQ.Text = "<p><strong>Solution:</strong> No Solution
Entered </p><p>&nbsp; </p>";
71     }
72     else
73     {
74         SolutionFAQ.Text = "<p><strong>Solution: </strong>" + FAQRow.
sub_Solution + "</p><p>&nbsp; </p>";
75     }
76     plcHolderContent.Controls.Add(SolutionFAQ);
77
78     }
79     plcHolderContent.Controls.Add(HTMLBreak);
80
81     }
82     }
83     catch (Exception ex)
84     {
85         lblFAQusrCtlStatus.Text = ex.Message;
86     }
87 }
88 }
89 }
```

VisuMAX ©2008