#region References

using System;

using System.Collections.Generic;

using System.Linq;

using System.Web;

using System.Web.UI;

using System.Web.UI.WebControls;

using SMART.CORE;

using SMART.Service;

using SMART.CORE.ScopeManagement;

using SMART.UTILITY;

using System.IO;

using System.Web.Services;

#endregion

namespace SMART.UI.Content

{

public partial class CreateMaterialScope : System.Web.UI.Page

{

#region Global Variables

static List<MaterialPartInfo> lstMaterialPartInfo = null;

CreateMaterialInfo objCreateMaterialInfo = null;

static int \_intUserID = 0;

int \_intUserRoleID = 0;

static int clientID = 1;

static string \_strReqPath = null;

static string \_strMarketId = null;

#endregion Global Variables

/// <summary>

/// Use to initialise Create Material scope page

/// </summary>

/// <param name="sender"></param>

/// <param name="e"></param>

protected void Page\_Load(object sender, EventArgs e)

{

int intUserRequestID = 0;

ValidatedUserInfo objValidatedUserInfo = null;

try

{

objValidatedUserInfo = new ValidatedUserInfo();

if (UtilSession.GetComplexTypeFromSession<ValidatedUserInfo>(UIConstants.SessionKeys.SESSION\_VALIDATED\_USER\_DETAILS, ref objValidatedUserInfo) == 1)

{

\_intUserRoleID = objValidatedUserInfo.RoleID;

\_intUserID = Convert.ToInt32(objValidatedUserInfo.UserID);

if (\_intUserRoleID.ToString() == UIConstants.PMROLEID || \_intUserRoleID.ToString() == UIConstants.ADMINROLEID

|| \_intUserRoleID.ToString() == UIConstants.SUPERADMINROLEID)

{

if (Session["TrackScopeHistory\_Market"] != null)

{

\_strMarketId = Session["TrackScopeHistory\_Market"].ToString();

}

if (Path.GetFileName(Request.Path) != null)

{

\_strReqPath = Path.GetFileName(Request.Path);

}

if (Request.QueryString["RequestID"] == null)

{

intUserRequestID = 0;

hdnUserRequestID.Value = Convert.ToString("0");

btnSaveAsTemplate.Visible = false;

}

else

{

intUserRequestID = Convert.ToInt32(Request.QueryString["RequestID"]);

hdnUserRequestID.Value = Request.QueryString["RequestID"];

btnSaveAsTemplate.Visible = true;

}

if (!IsPostBack)

{

if (intUserRequestID == 0 || Request.QueryString["tempID"] != null)

{

lblCreateMaterialScopeHeader.Text = "Create Material Scope";

btnSaveAsDraft.Text = "Save As Draft";

btnPreviewAndSubmit.Text = "Preview & Submit";

}

else

{

lblCreateMaterialScopeHeader.Text = "Update Material Scope";

btnSaveAsDraft.Text = "Update Draft";

btnPreviewAndSubmit.Text = "Update, Preview & Submit";

}

BindJobDetails(intUserRequestID);

}

}

else

{

Response.Redirect("AccessDenied.aspx");

}

}

else

{

Response.Redirect("AccessDenied.aspx");

}

}

catch (Exception objException)

{

ErrorInfo objErrorInfo = UtilFactory.CreateErrorInfo("CreateMaterialScope.aspx.cs", "page\_load");

ErrorLogManager.LogError(objException, objErrorInfo);

string strMessage = CustomMsgReader.GetCustomMessage(UIConstants.DisplayMSG.MODULE\_REPORTS,

UIConstants.DisplayMSG.MESSAGE\_ERROROCCURED).ToString();

Response.Redirect("Error.aspx?errMesg=" + strMessage);

}

finally

{

objValidatedUserInfo = null;

intUserRequestID = 0;

}

}

/// <summary>

/// For Updating Ajax call of Material Info

/// </summary>

/// <param name="intUserRequestID">Takes User Request ID</param>

/// <returns></returns>

[WebMethod]

public static CreateMaterialInfo GetMaterialPartDetails(int intUserRequestID)

{

BSLScopeManagementHelper objBSLScopeManagementHelper = null;

CreateMaterialInfo objCreateMaterialInfo = null;

string strJobNumber = string.Empty;

int intRequestTypeID = 0;

try

{

strJobNumber = System.Web.HttpContext.Current.Request.QueryString["jobNum"].ToString() != "" ? System.Web.HttpContext.Current.Request.QueryString["jobNum"].ToString() : "0";

intRequestTypeID = System.Web.HttpContext.Current.Request.QueryString["reqTypeID"].ToString() != "" ? Convert.ToInt32(System.Web.HttpContext.Current.Request.QueryString["reqTypeID"].ToString()) : 0;

objBSLScopeManagementHelper = new BSLScopeManagementHelper();

objCreateMaterialInfo = new CreateMaterialInfo();

objCreateMaterialInfo = objBSLScopeManagementHelper.GetDetailsToCreateMaterialScope(strJobNumber, intRequestTypeID, intUserRequestID);

}

catch (Exception objEx)

{

ErrorInfo objErrorInfo = UtilFactory.CreateErrorInfo("CreateMaterialScope.aspx.cs", "GetMaterialPartDetails");

ErrorLogManager.LogError(objEx, objErrorInfo);

string strErrMsg = CustomMsgReader.GetCustomMessage(UIConstants.DisplayMSG.MODULE\_REPORTS, UIConstants.DisplayMSG.MESSAGE\_ERROROCCURED);

HttpContext.Current.Response.Redirect("Error.aspx?errMesg=" + strErrMsg);

}

finally

{

objBSLScopeManagementHelper = null;

strJobNumber = string.Empty;

}

return objCreateMaterialInfo;

}

/// <summary>

/// To bind data to Grid, Table and Dropdown

/// </summary>

/// <param name="intUserRequestID"></param>

protected void BindJobDetails(int intUserRequestID)

{

BSLScopeManagementHelper objBSLScopeManagementHelper = null;

JobInfo objJobInfo = null;

try

{

objBSLScopeManagementHelper = new BSLScopeManagementHelper();

objCreateMaterialInfo = new CreateMaterialInfo();

objCreateMaterialInfo = objBSLScopeManagementHelper.GetDetailsToCreateMaterialScope(Convert.ToString(Request.QueryString["jobNum"])

, Convert.ToInt32(Request.QueryString["reqTypeID"]), intUserRequestID);

lstMaterialPartInfo = objCreateMaterialInfo.lstMaterialPartInfo;

grdJobDetails.DataSource = objCreateMaterialInfo.lstJobInfo;

grdJobDetails.DataBind();

objJobInfo = new JobInfo();

objJobInfo = objCreateMaterialInfo.lstJobInfo.First();

if (objCreateMaterialInfo.FlagRejected == 1)

{

spanRejectedMsg.Visible = true;

spanRejectedMsg.InnerText = objCreateMaterialInfo.RejectedReason;

spanRejectedMsg.Attributes.Add("class", "failuremessage");

}

if (objJobInfo.InConstruction == true)

{

chkInConstruction.Checked = true;

chkInConstruction.Enabled = false;

}

else

{

chkInConstruction.Checked = false;

chkInConstruction.Enabled = true;

}

txtRequestComments.Text = objJobInfo.RequestComments;

FillDropDown(ddlSelectWarehouse, objCreateMaterialInfo.lstWarehouseInfo, objCreateMaterialInfo.previousWarehouseInfo, "SELECT");

UpdatePanel2.Update();

}

catch (Exception objException)

{

ErrorInfo objErrorInfo = UtilFactory.CreateErrorInfo("CreateMaterialScope.aspx.cs", "BindJobDetails");

ErrorLogManager.LogError(objException, objErrorInfo);

}

finally

{

objBSLScopeManagementHelper = null;

objJobInfo = null;

objCreateMaterialInfo = null;

}

}

/// <summary>

/// To bind Warehouse dropdown list

/// </summary>

/// <param name="objDropdownList"></param>

/// <param name="lstWarehouseInfo"></param>

/// <param name="objPreviousWarehouseInfo"></param>

/// <param name="strDefaultText"></param>

protected void FillDropDown(DropDownList objDropdownList, List<WarehouseInfo> lstWarehouseInfo,

WarehouseInfo objPreviousWarehouseInfo, string strDefaultText)

{

WarehouseInfo objWareHouseInfo = null;

try

{

objWareHouseInfo = new WarehouseInfo();

if (objPreviousWarehouseInfo != null)

{

objWareHouseInfo.WarehouseID = objPreviousWarehouseInfo.WarehouseID;

objWareHouseInfo.WarehouseName = objPreviousWarehouseInfo.WarehouseName;

}

else

{

objWareHouseInfo.WarehouseID = 0;

objWareHouseInfo.WarehouseName = strDefaultText;

}

lstWarehouseInfo.Insert(0, objWareHouseInfo);

objDropdownList.DataSource = lstWarehouseInfo;

objDropdownList.DataTextField = "WarehouseName";

objDropdownList.DataValueField = "WarehouseID";

objDropdownList.DataBind();

}

catch (Exception objException)

{

ErrorInfo objErrorInfo = UtilFactory.CreateErrorInfo("CreateMaterialScope.aspx.cs", "FillDropDown");

ErrorLogManager.LogError(objException, objErrorInfo);

}

finally

{

objWareHouseInfo = null;

objDropdownList = null;

lstWarehouseInfo = null;

objPreviousWarehouseInfo = null;

strDefaultText = string.Empty;

}

}

/// <summary>

/// To pass data on selected values.

/// </summary>

/// <param name="objSubmitNoradStatus"></param>

/// <param name="objInConstruction"></param>

/// <param name="objRequestComments"></param>

/// <param name="objWarehouseId"></param>

/// <param name="objMaterialDetails"></param>

/// <param name="objButtonName"></param>

/// <returns>Return User request id</returns>

[WebMethod]

public static int SaveDataOnButtonClick(object objSubmitNoradStatus, object objInConstruction, object objRequestComments,

object objWarehouseId, SMART.CORE.ScopeManagement.CreateMaterialScope[] objMaterialDetails, object objButtonName)

{

int intRequestTypeID = 0, intUserRequestID = 0, count = 0;

BSLScopeManagementHelper objBSLScopeManagementHelper = null;

DataHolder objDataHolder = null;

string strUserRequestID = string.Empty, strJobNumber = string.Empty, strJobID = string.Empty;

KeyValue objLstValue = null;

string strItemATTPrice = string.Empty;

try

{

if (System.Web.HttpContext.Current.Request.QueryString["RequestID"] != null)

{

strUserRequestID = System.Web.HttpContext.Current.Request.QueryString["RequestID"].ToString() != "" ? System.Web.HttpContext.Current.Request.QueryString["RequestID"].ToString() : "0";

}

strJobNumber = System.Web.HttpContext.Current.Request.QueryString["jobNum"].ToString() != "" ? System.Web.HttpContext.Current.Request.QueryString["jobNum"].ToString() : "0";

strJobID = System.Web.HttpContext.Current.Request.QueryString["jobID"].ToString() != "" ? System.Web.HttpContext.Current.Request.QueryString["jobID"].ToString() : "0";

intRequestTypeID = System.Web.HttpContext.Current.Request.QueryString["reqTypeID"].ToString() != "" ? Convert.ToInt32(System.Web.HttpContext.Current.Request.QueryString["reqTypeID"].ToString()) : 0;

objBSLScopeManagementHelper = new Service.BSLScopeManagementHelper();

objDataHolder = new DataHolder();

objDataHolder.lstKeyValue = new List<KeyValue>();

objLstValue = new KeyValue();

objLstValue.Key = "UserJobId";

objLstValue.Value = Convert.ToString(strJobID);

objDataHolder.lstKeyValue.Add(objLstValue);

objLstValue = new KeyValue();

objLstValue.Key = "RequestTypeId";

objLstValue.Value = intRequestTypeID.ToString();

objDataHolder.lstKeyValue.Add(objLstValue);

objLstValue = new KeyValue();

objLstValue.Key = "UserRequestId";

objLstValue.Value = strUserRequestID == string.Empty ? "0" : strUserRequestID;

objDataHolder.lstKeyValue.Add(objLstValue);

objLstValue = new KeyValue();

objLstValue.Key = "WarehouseID";

objLstValue.Value = objWarehouseId.ToString();

objDataHolder.lstKeyValue.Add(objLstValue);

objLstValue = new KeyValue();

objLstValue.Key = "SubmitToNorad";

if (objSubmitNoradStatus.ToString().ToLower() == "true")

{

objLstValue.Value = "Checked";

}

else

{

objLstValue.Value = "Unchecked";

}

objDataHolder.lstKeyValue.Add(objLstValue);

objLstValue = new KeyValue();

objLstValue.Key = "InConstruction";

if (objInConstruction.ToString().ToLower() == "true")

{

objLstValue.Value = "Checked";

}

else

{

objLstValue.Value = "Unchecked";

}

objDataHolder.lstKeyValue.Add(objLstValue);

objLstValue = new KeyValue();

objLstValue.Key = "RequestComments";

objLstValue.Value = objRequestComments.ToString();

objDataHolder.lstKeyValue.Add(objLstValue);

objLstValue = new KeyValue();

objLstValue.Key = "ActionID";

objLstValue.Value = objButtonName.ToString().Contains("Update") ? "12" : "3";

objDataHolder.lstKeyValue.Add(objLstValue);

objLstValue = new KeyValue();

objLstValue.Key = "ActionFromPage";

objLstValue.Value = \_strReqPath;

objDataHolder.lstKeyValue.Add(objLstValue);

objLstValue = new KeyValue();

objLstValue.Key = "ActionFromButton";

objLstValue.Value = objButtonName.ToString();

objDataHolder.lstKeyValue.Add(objLstValue);

objLstValue = new KeyValue();

objLstValue.Key = "Userid";

objLstValue.Value = \_intUserID.ToString();

objDataHolder.lstKeyValue.Add(objLstValue);

objLstValue = new KeyValue();

objLstValue.Key = "UsermarketId";

objLstValue.Value = \_strMarketId;

objDataHolder.lstKeyValue.Add(objLstValue);

objLstValue = new KeyValue();

objLstValue.Key = "ClientID";

objLstValue.Value = clientID.ToString();

objDataHolder.lstKeyValue.Add(objLstValue);

count = objBSLScopeManagementHelper.CheckCountOfAvailableMaterialRequests(objDataHolder);

if (count > 0 && !(objButtonName.ToString().Contains("Update")))

{

intUserRequestID = 0;

}

else

{

intUserRequestID = objBSLScopeManagementHelper.CreateOrUpdateMaterialScope(objDataHolder, objMaterialDetails.OfType<SMART.CORE.ScopeManagement.CreateMaterialScope>().ToList());

}

}

catch (Exception objException)

{

ErrorInfo objErrorInfo = UtilFactory.CreateErrorInfo("CreateMaterialScope.aspx.cs", "SaveDataOnButtonClick");

ErrorLogManager.LogError(objException, objErrorInfo);

}

finally

{

intRequestTypeID = 0;

strUserRequestID = string.Empty;

strJobNumber = string.Empty;

strJobID = string.Empty;

objBSLScopeManagementHelper = null;

objDataHolder = null;

objSubmitNoradStatus = null;

objInConstruction = null;

objRequestComments = null;

objWarehouseId = null;

objMaterialDetails = null;

objButtonName = null;

objLstValue = null;

strItemATTPrice = string.Empty;

}

return intUserRequestID;

}

/// <summary>

/// Method for Template saving

/// </summary>

/// <param name="sender"></param>

/// <param name="e"></param>

protected void btnSave\_Click(object sender, EventArgs e)

{

int intUserRequestID = 0;

int intStatusValue = 0;

BSLScopeManagementHelper objBSLScopeManagementHelper = null;

string strTemplateName = string.Empty;

string strMessage = string.Empty;

string strClass = string.Empty;

try

{

intUserRequestID = Request.QueryString["RequestID"] != "" ? Convert.ToInt32(Request.QueryString["RequestID"]) : 0;

strTemplateName = txtTemplateName.Text;

objBSLScopeManagementHelper = new BSLScopeManagementHelper();

intStatusValue = objBSLScopeManagementHelper.AddTemplateDetails(\_intUserID, strTemplateName, intUserRequestID, "");

if (intStatusValue > 0)

{

strClass = "successmessage-add";

strMessage = "Template Saved Successfully";

}

else

{

strClass = "failuremessage-add";

strMessage = "Template saved Unsuccessfully";

}

SpanTemplate.Attributes.Add("class", strClass);

SpanTemplate.InnerText = strMessage;

SpanTemplate.Attributes.CssStyle[HtmlTextWriterStyle.Visibility] = "visible";

UpdatePanelCreateTemplate.Update();

}

catch (Exception obException)

{

ErrorInfo objErrorInfo = UtilFactory.CreateErrorInfo("CreateMaterialScope.aspx.cs", "btnSave\_Click");

ErrorLogManager.LogError(obException, objErrorInfo);

spnSuccessFailure.InnerText = CustomMsgReader.GetCustomMessage(UIConstants.DisplayMSG.MODULE\_GENERAL, UIConstants.DisplayMSG.MESSAGE\_ERROR);

spnSuccessFailure.Attributes.Add("class", "failuremessage");

spnSuccessFailure.Visible = true;

UpdatePanel2.Update();

}

finally

{

objBSLScopeManagementHelper = null;

objBSLScopeManagementHelper = null;

strTemplateName = string.Empty;

strMessage = string.Empty;

strClass = string.Empty;

}

}

/// <summary>

/// Return to Previous Scope

/// </summary>

/// <param name="sender"></param>

/// <param name="e"></param>

protected void btnBack\_Click(object sender, EventArgs e)

{

try

{

Response.Redirect("~/Content/TrackScopeHistory.aspx?jobid=" + Request.QueryString["jobId"]);

}

catch (Exception obException)

{

ErrorInfo objErrorInfo = UtilFactory.CreateErrorInfo("CreateMaterialScope.aspx.cs", "btnBack\_Click");

ErrorLogManager.LogError(obException, objErrorInfo);

}

}

}

}