APPENDIX

```
namespace HeterogeneousApp
{
   public partial class DecisionManager : System.Web.UI.Page
   {
      dbaccess obj = new dbaccess();
      Security objSecurity = new Security();
      protected void Page_Load(object sender, EventArgs e)
      {
         if (IsPostBack == false)
         {
            obj.GridLoad(GridView1,obj.DataadapterQuery("select parameter,1
'from_value',1 'to_value' from input_field_parameters"));
            //string str=
obj.ConvertDataSetToXMLString(obj.DataadapterQuery("select parameter,1
'from_value',1 'to_value' from input_field_parameters"));
         }
      }
  protected void btn_request_Click(object sender, EventArgs e)
      {
         // DataSet ds = obj.ConvertGridviewToDataSet(GridView1);
         stringXMLString=
obj.ConvertDataSetToXMLString(Grid1_DatatableUpdatation());
         XMLString = XMLString.Replace(" ", "");
         //localhost1.Service1 objWebService1 = new
HeterogeneousApp.localhost1.Service1();
         //objWebService1.PreAuthenticate = true;
```

```
        //objWebService1.Credentials =
System.Net.CredentialCache.DefaultCredentials;
        Samsung.Samsung objwebSamsung = new
HeterogeneousApp.Samsung.Samsung();
        objwebSamsung.PreAuthenticate = true;
        objwebSamsung.Credentials =
System.Net.CredentialCache.DefaultCredentials;
        DataSet ds1 =
objwebSamsung.ResultOutput(txt_product_name.Text.Trim(), XMLString);
        //Samsung
        ds1.Merge(ds1);
        SearchResult objexpert = new SearchResult();
        DataSet dsexpert =
objexpert.ResultXML(txt_product_name.Text.Trim(), XMLString);
        ds1.Merge(dsexpert);
        XML String = obj.ConvertDataSetToXMLString(ds1);
        DataSet ds_distinct = objexpert.AllResultXML(XMLString);
        XMLString = obj.ConvertDataSetToXMLString(ds_distinct);
        DataSet ds_decision = objexpert.decison_manager(XMLString);

 DataSet ds2 = objwebHP.ResultOutput(txt_product_name.Text.Trim(),
XMLString);
        ACS.ACS objwebACS = new HeterogeneousApp.ACS.ACS();
        Objweb ACS.PreAuthenticate = true;
        Objweb ACS.Credentials =
System.Net.CredentialCache.DefaultCredentials;
        DataSet ds2 = objweb
ACS.ResultOutput(txt_product_name.Text.Trim(), XMLString);
        // ACS
        Ds1.Merge(ds2);
```

```
        SearchResult objexpert = new SearchResult();
        DataSet dsexpert =
objexpert.ResultXML(txt_product_name.Text.Trim(), XMLString);
        ds1.Merge(dsexpert);
        XML String = obj.ConvertDataSetToXMLString(ds1);
        DataSet ds_distinct = objexpert.AllResultXML(XMLString);
        XMLString = obj.ConvertDataSetToXMLString(ds_distinct);
        DataSet ds_decision = objexpert.decison_manager(XMLString);

 DataSet ds3 = objwebHP.ResultOutput(txt_product_name.Text.Trim(),
XMLString);
        LOP.LOP objwebLOP = new HeterogeneousApp.LOP.LOP();
        objwebSamsung.PreAuthenticate = true;
        objwebSamsung.Credentials =
System.Net.CredentialCache.DefaultCredentials;
        DataSet ds3 = objweb
LOP.ResultOutput(txt_product_name.Text.Trim(), XMLString);
        //LOP
        Ds1.Merge(ds3);
        SearchResult objexpert = new SearchResult();
        DataSet dsexpert =
objexpert.ResultXML(txt_product_name.Text.Trim(), XMLString);
        ds1.Merge(dsexpert);
        XML String = obj.ConvertDataSetToXMLString(ds1);
        DataSet ds_distinct = objexpert.AllResultXML(XMLString);
        XMLString = obj.ConvertDataSetToXMLString(ds_distinct);
        DataSet ds_decision = objexpert.decison_manager(XMLString);
```

```
DataSet ds4 = objwebHP.ResultOutput(txt_product_name.Text.Trim(),
XMLString);
        AXL.AXL objwebAXL = new HeterogeneousApp. AXL.AXL ();
        Objweb AXL.PreAuthenticate = true;
        objweb AXL.Credentials =
System.Net.CredentialCache.DefaultCredentials;
        DataSet ds4 = objweb
AXL.ResultOutput(txt_product_name.Text.Trim(), XMLString);
        //AXL
        ds1.Merge(ds4);
        SearchResult objexpert = new SearchResult();
        DataSet dsexpert =
objexpert.ResultXML(txt_product_name.Text.Trim(), XMLString);
        ds1.Merge(dsexpert);
        XML String = obj.ConvertDataSetToXMLString(ds1);
        DataSet ds_distinct = objexpert.AllResultXML(XMLString);
        XMLString = obj.ConvertDataSetToXMLString(ds_distinct);
        DataSet ds_decision = objexpert.decison_manager(XMLString);

        if (ds_decision != null)
        {
           if (ds_decision.Tables[0].Rows.Count > 0)
           {
              GridView2.DataSource = ds_decision.Tables[0];
              GridView2.DataBind();
              GridView2.Visible = true;
              if (ds_decision.Tables[1].Rows.Count > 0)
              {
                 GridView3.DataSource = ds_decision.Tables[1];
                 GridView3.DataBind();
```

```
            GridView3.Visible = true;
        }
        else
        {
            GridView3.Visible = false;
        }
    }
    else
    {
        GridView2.Visible = false;
    }
  }
  else
  {
    Response.Write("ds is null");
  }
}

private DataTable Grid1_SetDatatableColumns(DataTable dt)
{
  dt.Columns.Clear();
  dt.Columns.Add(new DataColumn("parameter", typeof(string)));
  dt.Columns.Add(new DataColumn("from_value", typeof(string)));
  dt.Columns.Add(new DataColumn("to_value", typeof(string)));
  return dt;
}
private DataSet Grid1_DatatableUpdatation()
{
  DataTable dt = new DataTable();
  dt = Grid1_SetDatatableColumns(dt);
```

```
        foreach (GridViewRow row in GridView1.Rows)
        {
            TextBox txt_parameter =
(TextBox)row.Cells[0].FindControl("txt_parameter");
            TextBox tx_from_value =
(TextBox)row.Cells[1].FindControl("txt_from_value");
            TextBox txt_to_value =
(TextBox)row.Cells[2].FindControl("txt_to_value");
            DataRow dr = null;
            dr = dt.NewRow();
            dr["parameter"] = txt_parameter.Text.Trim();
            dr["from_value"] = tx_from_value.Text.Trim();
            dr["to_value"] = txt_to_value.Text.Trim();
            dt.Rows.Add(dr);
        }
        DataSet ds = new DataSet();
        ds.Tables.Add(dt);
        return ds;
    }
    protected void btn_Consumer_approval_Click(object sender,
                                                EventArgs e)
    {
        Response.ContentType = "application/pdf";
        Response.AddHeader("content-disposition",
"attachment;filename=Export.pdf");
        Response.Cache.SetCacheability(HttpCacheability.NoCache);
        StringWriter sw = new StringWriter();
        HtmlTextWriter hw = new HtmlTextWriter(sw);
        HtmlForm frm = new HtmlForm();
        GridView2.Parent.Controls.Add(frm);
```

```
            frm.Attributes["runat"] = "server";
            frm.Controls.Add(GridView2);
            frm.RenderControl(hw);
            StringReader sr = new StringReader(sw.ToString());
            Document pdfDoc = new Document(PageSize.A4, 10f, 10f, 10f, 0f);
            HTMLWorker htmlparser = new HTMLWorker(pdfDoc);
            PdfWriter.GetInstance(pdfDoc, Response.OutputStream);
            pdfDoc.Open();
            htmlparser.Parse(sr);
            pdfDoc.Close();
            Response.Write(pdfDoc);
            Response.End();
        }
    }
}
CREATE procedure sp_descision_manager
@xml xml
as
begin
DECLARE @Temp_ds TABLE
(
Ventor nvarchar(50),
product_code nvarchar(20),
product_name nvarchar(50),
product_price numeric(12,2),
parameter nvarchar(20),
fuzzy_set nvarchar(50),
from_value numeric(18,2),
to_value numeric (18,2),
set_value numeric (18,2)
```

```
)

insert into @Temp_ds
SELECT
                  Tbl.Col.value('Ventor[1]', 'nvarchar(50)')  ,
                 Tbl.Col.value('product_code[1]', 'nvarchar(20)'),
                   Tbl.Col.value('product_name[1]', 'nvarchar(50)'),
                  Tbl.Col.value('product_price[1]', 'numeric(12,2)'),
                  Tbl.Col.value('parameter[1]', 'nvarchar(20)')  ,
                  Tbl.Col.value('fuzzy_set[1]', 'nvarchar(50)'),
                 Tbl.Col.value('from_value[1]', 'numeric(18,2)'),
                  Tbl.Col.value('to_value[1]', 'numeric(18,2)')       ,
                 Tbl.Col.value('set_value[1]', 'numeric(18,2)')
    FROM   @xml.nodes('//NewDataSet//Table')  Tbl(Col)

--select distinct  from @Temp_ds
DECLARE @Temp_decision TABLE
(
product_code nvarchar(20),
parameter nvarchar(20),
set_value numeric (18,2),
fuzzy_set nvarchar(50)
)
--select * from @Temp_expert_advise
--select product_code ,Ventor from @Temp_expert_advise
declare @Ventor  nvarchar(50),@product_code nvarchar(50),
@product_name nvarchar (50),@parameter nvarchar(50),@set_value
numeric(18,2)
declare cur cursor for select distinct product_code  from @Temp_ds
--select @cur =cursor for select product ,ventor from @Temp_expert_advise
```

```
open cur
fetch next from cur into @product_code
while  @@FETCH_STATUS = 0
begin
		declare @cost numeric (18,2), @Duration numeric
(18,2),@Quality numeric(18,2)
		select @cost = set_value   from @Temp_ds where
parameter='cost' and product_code= @product_code
		select @Duration = set_value   from @Temp_ds where
parameter='Duration' and product_code= @product_code
		select @Quality = set_value   from @Temp_ds where
parameter='Quality' and product_code= @product_code

		--select @cost ,@Duration ,@Quality
		-- find the cost parameter
		declare @m_cost nvarchar(50)
		select @cost =  isnull (@cost,0)
		if  ((@cost < 1000) or ( @cost between 1000 and 2999))
		begin
			select @m_cost = 'Low'
		end
		else if ( ( @cost between 3000 and 5999) or ( @cost = 6000) or
( @cost between 6000 and 7999))
		begin
			select @m_cost = 'Medium'
		end
		else if ((@cost between 8000 and 25999) or (@cost >=
150000))
		begin
			select @m_cost = 'High'
```

```
            end
      insert into @Temp_decision
            select  @product_code,'Cost',@cost, @m_cost

            -- to find the duration parameter

            declare @m_duration nvarchar(50)

            select @Duration = isnull (@service,0)

            if ((@Duration <2) or (@Duration between 2 and 3))
            begin
                  select  @m_duration ='Low'
            end
            else if ( (@Duration=4) or (@Duration between 4 and 6)  )
            begin
                  select @m_duration='Mediumn'
            end
            else if ( (@Duration between 7 and 9) or (@Duration=10) or
(@Duration between 10 and 12))
            begin
                  select @m_duration = 'High'
            end

            insert into @Temp_decision
            select  @product_code,'Duration',@Duration, @m_duration

            -- to find the quality
            declare @m_quality nvarchar(50)
            select @Quality = isnull (@Quality,0)
```

```
                if ((@Quality <2) or (@Quality between 1 and 2))
                begin
                        select  @m_quality ='Low'
                end
                else if ( (@Quality=3) or (@Quality between 3 and 5)  )
                begin
                        select @m_quality='Mediumn'
                end
                else if ( (@Quality between 6 and 8) or (@Quality=9) or
(@Quality between 9 and 10))
                begin
                        select @m_quality = 'High'
                end
                 insert into @Temp_decision
                select  @product_code,'Quality',@Quality, @m_quality
fetch next from cur into @product_code
end
DECLARE @Temp_decision_tmp TABLE
(
product_code nvarchar(20),
parameter nvarchar(20),
set_value nvarchar (20),
fuzzy_set nvarchar(50)
)
insert into @Temp_decision_tmp select * from @Temp_decision
--select * from @Temp_decision_tmp
select distinct  b.ventor, a.product_code,b.product_name,a.parameter,
case when a.set_value ='0.00' then 'No Match' else a.set_Value  end
'set_Value',a.fuzzy_set from @Temp_decision_tmp a , @Temp_ds b
where a.product_code=b.product_code
```

```
select distinct b.* from @Temp_decision a, expert_advise_flag b where
a.product_code =b.product_code
CLOSE cur
DEALLOCATE cur
end
```