```csharp
using System;
using System.ComponentModel;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Data;
using System.Data.Common;
using WA_EmsgSMS.bin;
using WA_EmsgSMS.bin.Database.Entity;

namespace WA_EmsgSMS.bin.Database
{
    public class DALtoMgntUrlMappings:DALtoDbObject<EntityMgntUrlMapping>
    {
        private const string ITEM_SEPERATOR_SYMBOL = "+";
        private const string DB_DATETIME_FORMAT_STRING = "yyyy/MM/dd hh:mm:ss";

        private DbProviderFactory _dbProvider;
        private DbConnection _dbConnection;
        private DbCommand _dbCommand;

        public DALtoMgntUrlMappings() {

            Initialize();
        }

        protected override void Initialize()
        {
            _dbProvider = DbProviderFactories.GetFactory(AppResource.GLOBAL_DB_PROVIDER_NAME);
            _dbConnection = _dbProvider.CreateConnection();
            _dbCommand = _dbProvider.CreateCommand();
        }

        [DataObjectMethod(DataObjectMethodType.Select,true)]
        public override IList<EntityMgntUrlMapping> GetResultSet()
        {
            IList<EntityMgntUrlMapping> resultSet = new List<EntityMgntUrlMapping>();
            DbDataAdapter dbDataAdapter = _dbProvider.CreateDataAdapter();
            DataTable dataTable = new DataTable();
            string commandText;

            _dbConnection.ConnectionString = AppResource.GLOBAL_CONN_STRING;
            _dbCommand.Connection = _dbConnection;

            commandText = "SELECT * FROM sys_MgntUrlMappings ";

            _dbCommand.CommandText = commandText;

            dbDataAdapter.SelectCommand = _dbCommand;
```

```csharp
      dbDataAdapter.Fill(dataTable);

      bool hookFlag = false;
      hookFlag = dataTable.Rows.Count > 0 ? true : false;

      if (hookFlag)
      {
         DataRow[] dataRows = dataTable.Select("parentId=0");
         hookFlag = dataRows.Count() > 0 ? true : false;

         if (hookFlag)
         {
            EntityMgntUrlMapping entity;

            DataRow dataRow = dataRows[0];
            entity = MapDataRowToObject(dataRow);
            string parentId = entity.Id.ToString();
            int level = 1;

            resultSet.Add(entity);
            GenerateChildEntity(resultSet, dataTable, parentId, level);
         }//end if
      }//end if
      return resultSet;
}//end method: GetResultSet

[DataObjectMethod(DataObjectMethodType.Select,false)]
public IList<EntityMgntUrlMapping> GetResultSet(int? id)
{
   IList<EntityMgntUrlMapping> resultSet = new List<EntityMgntUrlMapping>();
   DbDataAdapter dbDataAdapter = _dbProvider.CreateDataAdapter();
   DataTable dataTable = new DataTable();
   string commandText;

   _dbConnection.ConnectionString = AppResource.GLOBAL_CONN_STRING;
   _dbCommand.Connection = _dbConnection;

   commandText = "SELECT * FROM sys_MgntUrlMappings WHERE id={0}";
   _dbCommand.CommandText = string.Format(commandText, id.Value.ToString());

   dbDataAdapter.SelectCommand = _dbCommand;
   dbDataAdapter.Fill(dataTable);

   bool hookFlag = false;
   hookFlag = dataTable.Rows.Count > 0 ? true : false;

   if (hookFlag) {
      DataRow dataRow = dataTable.Rows[0];
      EntityMgntUrlMapping entity = MapDataRowToObject(dataRow);
```

```csharp
            resultSet.Add(entity);
        }

        return resultSet;
    }

    [DataObjectMethod(DataObjectMethodType.Insert,true)]
    public override int InsertSingleEntity(EntityMgntUrlMapping entity)
    {
        string parentId;
        string applicationId;
        string text = entity.Text;
        string value = entity.Value;
        string imgUrl = entity.ImgUrl;
        string url = entity.Url;
        string type = entity.Type;
        string sort = Convert.ToString(entity.Sort);
        string visible = entity.Visible == true ? "1" : "0";
        string unlock;
        string errorUrl = entity.ErrorUrl;
        string createdDate;

        if (entity.ParentId.HasValue)
            parentId = Convert.ToString(entity.ParentId.Value);
        else parentId = "0";

        if (entity.ApplicationId.HasValue)
            applicationId = entity.ApplicationId.Value.ToString();
        else applicationId = AppResource.GetCurrentApplicationId();

        if (entity.Unlock.HasValue)
            unlock = entity.Unlock == true ? "1" : "0";
        else unlock = "1";

        if (entity.CreatedDate.HasValue)
            createdDate = entity.CreatedDate.Value.ToString(DB_DATETIME_FORMAT_STRING);
        else createdDate = DateTime.Now.ToString(DB_DATETIME_FORMAT_STRING);

        string commandText;
        commandText = "INSERT INTO
sys_MgntUrlMappings(parentId,applicationId,text,value,imgUrl,url,type,sort,visible,unlock,errorUrl,createdDate)
" +
                "VALUES({0},'{1}','{2}','{3}','{4}','{5}','{6}',{7},{8},{9},'{10}','{11}')";

        _dbConnection.ConnectionString = AppResource.GLOBAL_CONN_STRING;
        _dbCommand.Connection = _dbConnection;
        _dbCommand.CommandText =
            string.Format(commandText, parentId, applicationId, text, value, imgUrl, url, type, sort, visible,
```

```
unlock, errorUrl, createdDate);

        int affectedCount = 0;

        try
        {
            _dbConnection.Open();
            affectedCount = _dbCommand.ExecuteNonQuery();
        }
        catch (Exception ex)
        {

            throw ex;
        }
        finally {

            _dbConnection.Close();

        }//end try catch
        return affectedCount;
    }//end method: InsertSingleEntity

    [DataObjectMethod(DataObjectMethodType.Update,true)]
    public override int UpdateSingleEntity(EntityMgntUrlMapping entity)
    {
        string id = Convert.ToString(entity.Id);
        string parentId = Convert.ToString(entity.ParentId);
        string text = entity.Text;
        string value = entity.Value;
        string imgUrl = entity.ImgUrl;
        string url = entity.Url;
        string type = entity.Type;
        string sort = Convert.ToString(entity.Sort);
        string visible = entity.Visible == true ? "1" : "0";
        string errorUrl = entity.ErrorUrl;

        string commandText;
        commandText = "UPDATE sys_MgntUrlMappings " +
                "SET
parentId={0},text='{1}',value='{2}',imgUrl='{3}',url='{4}',type='{5}',sort={6},visible={7},errorUrl='{8}' " +
                "WHERE id={9}";

        _dbConnection.ConnectionString = AppResource.GLOBAL_CONN_STRING;
        _dbCommand.Connection = _dbConnection;
        _dbCommand.CommandText = string.Format(commandText, parentId, text, value, imgUrl, url, type,
sort, visible, errorUrl, id);

        int affetcedCount = 0;
```

```
    try
    {
      _dbConnection.Open();
      affetcedCount = _dbCommand.ExecuteNonQuery();
    }
    catch (Exception ex)
    {

      throw ex;
    }
    finally {
      _dbConnection.Close();
    }//end try catch
    return affetcedCount;
}//end method: UpdateSingleEntity

[DataObjectMethod(DataObjectMethodType.Delete,true)]
public override int DeleteSingleEntity(EntityMgntUrlMapping entity)
{
    string id = Convert.ToString(entity.Id);

    string commandText = "DELETE FROM sys_MgntUrlMappings WHERE id={0}";

    _dbConnection.ConnectionString = AppResource.GLOBAL_CONN_STRING;
    _dbCommand.Connection = _dbConnection;
    _dbCommand.CommandText = string.Format(commandText, id);

    int affectedCount = 0;

    try
    {
      _dbConnection.Open();
      affectedCount = _dbCommand.ExecuteNonQuery();
    }
    catch (Exception ex)
    {
      throw ex;
    }
    finally {
      _dbConnection.Close();
    }//end try catch

    return affectedCount;
}//end method: DeleteSingleEntity

protected override EntityMgntUrlMapping MapDataRowToObject(DataRow dataRow)
{
    int id = (int)dataRow["id"];
```

```
        int parentId = (int)dataRow["parentId"];
        Guid applicationId = (Guid)dataRow["applicationId"];
        string text = dataRow["text"] as string;
        string value = dataRow["value"] as string;
        string imgUrl = dataRow["imgUrl"] as string;
        string url = dataRow["url"] as string;
        string type = dataRow["type"] as string;
        int sort = (int)dataRow["sort"];
        bool visible = (bool)dataRow["visible"];
        bool unlock = (bool)dataRow["unlock"];
        string errorUrl = dataRow["errorUrl"] as string;
        DateTime createdDate = (DateTime)dataRow["createdDate"];

        EntityMgntUrlMapping entity = new EntityMgntUrlMapping();
        entity.Id = id;
        entity.ParentId = parentId;
        entity.ApplicationId = applicationId;
        entity.Text = text;
        entity.Value = value;
        entity.ImgUrl = imgUrl;
        entity.Url = url;
        entity.Type = type;
        entity.Sort = sort;
        entity.Visible = visible;
        entity.Unlock = unlock;
        entity.ErrorUrl = errorUrl;
        entity.CreatedDate = createdDate;

        return entity;
    }//end method: MapDataRowToObject

    private void GenerateChildEntity(IList<EntityMgntUrlMapping> resultSet, DataTable dataTable, string
_parentId, int level) {

        string sortExpression = "sort";
        string filterExpression = "parentId=" + _parentId;

        DataRow[] dataRows = dataTable.Select(filterExpression, sortExpression);
        bool hookFlag = dataRows.Count() > 0 ? true : false;

        level++;
        if (hookFlag) {
            foreach (DataRow dataRow in dataRows) {

                EntityMgntUrlMapping entity = MapDataRowToObject(dataRow);
                string parentId = Convert.ToString(entity.Id);
                string text = entity.Text;
```

```
                for (int start = 1; start < level; start++)
                    text = ITEM_SEPERATOR_SYMBOL + text;
                //end foreach
                entity.Text = text;

                resultSet.Add(entity);
                GenerateChildEntity(resultSet, dataTable, parentId, level);
            }//end foreach

        }//end if
    }//end method: GenerateChildEntity

  }//end class
}
```