Spring 2015

Recipe Suggestion Tool

Tejaswi Patha  
*Governors State University*

Deepika Sandhadi  
*Governors State University*

Follow this and additional works at: [http://opus.govst.edu/capstones](http://opus.govst.edu/capstones)

🔗 Part of the [Databases and Information Systems Commons](http://opus.govst.edu/capstones), and the [Graphics and Human Computer Interfaces Commons](http://opus.govst.edu/capstones)

**Recommended Citation**

[http://opus.govst.edu/capstones/98](http://opus.govst.edu/capstones/98)
ABSTRACT

There is currently a great need for a tool to search cooking recipes based on ingredients, country and recipetype. Current search engines do not provide this feature. Most of the recipe search results in current websites are not efficiently clustered based on relevance or categories resulting in a user getting lost in the huge search results presented. They also do not provide links to view images of the ingredients of a recipe.

My project aims to combine the features like search based on ingredients, suggestions for similar recipes, and images for the ingredients under one search engine and provide an intuitive interface for the same. I explored different clustering algorithms to find an efficient algorithm that can be used to cluster recipe data matching user's queries. As part of this project, I also built FreeText search it help users can search Recipes by ingredients, country and recipe type. I created few charts for users to understand which ingredients are used more in recipes and which country ingredients are more. This website also provides articles to users for making tasty recipes. In this article page users can comment and rate the article. Our website is deployed to Microsoft azure platform.
# Table of Contents

ABSTRACT

1.0 INTRODUCTION

2.0 USER INTERFACE

2.1 Login Page:

2.2 Register Page:

2.3 Home Page:

2.4 Article Page:

3.0 DATABASE DESIGN

3.1 Data Definition

3.2 E-R DIAGRAM

4.0 DATA VISUALIZATION

4.1 GetRecipeByKeyWord:

4.2 GetRecipeCountByIngredients:

4.3 Get recipe count by country:

5.0 AZURE CLOUD:

5.1 Creating Virtual machine:

5.2 Host website on IIS:

6.0 Technical Environments

6.1 Development

References

Appendix 1

Sample Asp.net page

Sample c# code behind

Sample business layer
1.0 INTRODUCTION

In this project, we worked on a recipe suggestion tool, which suggests similar recipes. Users can also search Recipes based on ingredients, Country and RecipeType from Database and provide recipes list by order. We also provide the facility for the user to view the images of the recipes. We use Entity framework to retrieve the data from SQL server database. For keyword search we used SQL server FreeTextTable function and ranked the results based on the relevance.

We used ASP.net identity provider for login component and registration component. Only logged in users can view the dashboard page, comment on articles and rate articles.

We used Microsoft chart control to create charts for ingredients vs. count and country vs. count. I wrote few stored procedures to feed the data to charts. We also provided articles to the users by stored procedures. Users can read an article, rate the articles and comment on article if the user is logged in.

FreeTextTable Search helps users to access information easily. One only needs to get connected to the Internet to get the information one needs. When searching for cooking recipes, sometimes user may prefer to search based on ingredients, country and RecipeType.

We have deployed the website and SQL database on windows azure. We have created a Virtual machine in widows azure installed IIS, .NET framework 4.5, and set up the IIS to host the web site in the virtual machine.
2.0 USER INTERFACE

2.1 Login Page:
The Login Screen will remain in the right column of the web page.
The log in screen will have an option to enter username and password for existing users.

2.2 Register Page:
Register screen will allow users to create new account.
The account information section consists of fields as shown below, which are
required to create new account.

2.3 Home Page:

Once user login to the website they can search recipes in multiple ways like,
By selecting Ingredient,
By selecting country name,
By Keyword,
By alphabet

The search engine returns recipe image, recipe id, recipe name, category, and country.
You need a new dish in a hurry? Stumped by how to make that something special? We can help with your busy lifestyle. Take a look around and review our hundreds of free recipes or submit one of your own favorites.


Main Course Recipe

- Pork (212)
- SeaFoods (20)
- Veal (7)
- Veal (6)
- Seafood (9)
- Seafood (5)
- Sauces (58)
- Sauces (88)
- Sandwich (46)
- Pizza (12)
- Desserts (41)
- Desserts (45)
- Pies (67)
- Pies (63)
- Spices (35)
- Spices (15)
- Beef (36)
- Beef (20)
- Kebab (11)
- Kebab (11)
- Soup (194)
- Soup (194)
- Casseroles (66)
- Casseroles (60)
- Cakes (121)
- Cakes (121)
- Entrees (128)
- Entrees (128)
- One-pot (22)
- One-pot (22)
- Lamb (22)
- Lamb (22)
- Vegetarian (124)
- Vegetarian (124)
- Pasta (81)
- Pasta (81)
- Pudding (2)
- Pudding (21)
- Side dish (2)
- Side dish (24)
- Puddings (4)
- Puddings (14)
- Mayonnaise (12)
- Mayonnaise (12)
- Barbeque (24)
- Barbeque (24)
- Micro oven (14)
- Bacon (2)
- Bacon (2)
- Wild Game (12)
- Wild Game (12)
- Appetizers (46)
- Appetizers (46)
- Meat (81)
- Meat (81)
- Dressing (120)
- Dressing (120)
- Dips (42)
- Dips (42)
### Recipe Suggestion Tool

#### Chicken

<table>
<thead>
<tr>
<th>Image</th>
<th>RecipeID/Recipe Name</th>
<th>Category</th>
<th>Country</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image1.png" alt="Image" /></td>
<td>Chicken Kidney with Crab Meat</td>
<td>Poultry</td>
<td>China</td>
</tr>
<tr>
<td><img src="image2.png" alt="Image" /></td>
<td>Wild Chicken</td>
<td>Wild Game</td>
<td>China</td>
</tr>
<tr>
<td><img src="image3.png" alt="Image" /></td>
<td>Chicken Rolls with Quinoa Sauce</td>
<td>Poultry</td>
<td>China</td>
</tr>
<tr>
<td><img src="image4.png" alt="Image" /></td>
<td>Chicken with Cilantro</td>
<td>Casseroles</td>
<td>USA</td>
</tr>
<tr>
<td><img src="image5.png" alt="Image" /></td>
<td>Baked Chicken Delight</td>
<td>Casseroles</td>
<td>USA</td>
</tr>
<tr>
<td><img src="image6.png" alt="Image" /></td>
<td>Chopped chicken liver appetizer</td>
<td>Poultry</td>
<td>Israel</td>
</tr>
</tbody>
</table>

#### Beef

<table>
<thead>
<tr>
<th>Image</th>
<th>RecipeID/Recipe Name</th>
<th>Category</th>
<th>Country</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image7.png" alt="Image" /></td>
<td>Beef and Peppers in Hot-Gin Sauce</td>
<td>Beef</td>
<td>China</td>
</tr>
<tr>
<td><img src="image8.png" alt="Image" /></td>
<td>Beef Chow Mein</td>
<td>Beef</td>
<td>China</td>
</tr>
<tr>
<td><img src="image9.png" alt="Image" /></td>
<td>Beef in Tepalo Sauce Recipe</td>
<td>Beef</td>
<td>China</td>
</tr>
<tr>
<td><img src="image10.png" alt="Image" /></td>
<td>Beef with Ginger and Onions</td>
<td>Beef</td>
<td>China</td>
</tr>
<tr>
<td><img src="image11.png" alt="Image" /></td>
<td>Chinese Fried Rice</td>
<td>Side dish</td>
<td>China</td>
</tr>
<tr>
<td><img src="image12.png" alt="Image" /></td>
<td>Seafood Dumplings</td>
<td>Fish</td>
<td>China</td>
</tr>
</tbody>
</table>

#### Search Options

- **Recipe ID**: 805, 861, 855, 86, 1, 402
- **Categories**: Poultry, Wild Game, Casseroles
- **Countries**: China, USA, Israel
Description Page:
Users can see ingredients and preparation method of the recipe once they click on recipe id.

Lamb with Peppers
By Unknown (Internet), Country: India
Category: Lamb

Ingredients:
Shoulder of lamb (boned and cubed) 1½ kg.
Onion sliced 500 gms.
Button Mushroom 250 gms.
One each of green, red and yellow peppers (capsicums)
Chicken stock 150 gms.
Oil for frying
Pepper to taste
Salt to taste
Tripping:
Soured cream 150 ml

Parsley finely chopped 1 tbsp.

Preparation:
2.4 Article Page:

In this page user can read articles, can give rating and can enter comments. We have used user control for left section of the article and repeater to show the list on right section.

By clicking on the name you can go to full article page where logged in users can enter comments and rate the article.
Five Minutes a Day for Fresh-Baked Bread

The Secret: Keep Dough Refrigerated. It is easy to have fresh bread whenever you want it with only five minutes a day of active effort. Just mix the dough and let it sit for two hours. No kneading needed! Then shape and bake a loaf, and refrigerate the rest to use over the next couple weeks. Yes, really! The Master Recipe below makes enough dough for many loaves. When you want fresh-baked crusty bread, take some dough, shape it into a loaf, let it rise for about 20 minutes, then bake. Your house will smell like a bakery, and your family and friends will love you for it.

I was trained as a scientist, not as a chef. That helped in developing a new process for homemade bread, but I never could have brought the recipes to this level without the expertise and standards of a professional — my co-author Zia is a Culinary Institute of America trained pastry chef. Over several years, we found how to subtract the various steps that make the classic technique so time-consuming, and identified a few that couldn’t be omitted. This is the quick and easy way to get professional-style results with little effort.

How It All Began
Like most kids, my brother and I loved sweets, so dessert was our favorite time of day. We’d sit in the kitchen, devouring frosted supermarket doughnuts. “Those are too sweet,” my grandmother would say. “If I had a piece of good rye bread, with cheese on it, it’s better than cake.”

Suddenly, I knew she was right. I could finish half a loaf of very fresh, very crisp rye bread by myself, with or without butter. The right stuff came from a little bakery in Queens. The crust was crisp, this and caramelized brown. The crumb was moist and dense, chewy but never gummy, and bursting with tangy yeast, rye and wheat flavors. It made great toast, too — and yes, it was better than cake.

When I was a kid, handmade bread was available all over New York City, and it wasn’t a novelty delicacy. Everyone took it for granted. It was a simple comfort food brought here by modest immigrants, but now the dry, tasteless corner shops turning out great European breads are no longer so ubiquitous, and nobody’s grandmother would ever have paid as for a loaf of bread.

So Zia and I decided to do something about it. Our book, Artisan Bread in Five Minutes a Day, is our attempt to help people re-create the great ethnic breads of years past, in their own homes, without investing serious time or effort. Using our straightforward, fast and easy recipes, anyone can create artisan bread and avoid the boredom of supermarket imitation products.

There are (9) comments

Comment on Apr. 26, 2015 by rajesh

Comment on Apr. 26, 2015 by rajesh

Comment on Apr. 26, 2015 by rajesh

Comment on Apr. 26, 2015 by rajesh

Field mark with red asterisk(*) is required.

Username: deeksha
Email: deeksha

Comment:* Only 200 char allowed

Submit
3.0 DATABASE DESIGN

3.1 Data Definition
The main repository for Recipe data is a relational database utilizing Microsoft SQL server 2008 as the management system and hosted on a Microsoft Azure. This repository is where all Recipe data is stored and maintained. There are two instances, which exist to support the development and production Recipe sites. The Recipe database is normalized and converted to a MS SQL database, which is hosted on Windows Virtual machine in Azure. The External website uses this MSSQL database. The renormalization of the main Recipe database allows for faster query performance on the external web site.
3.2 E-R DIAGRAM

- **Recipe**
  - RecipeID
  - RecipeName
  - Instructions
  - ServingQuantity
  - Calories
  - Author

- **RecipeCategory**
  - RecipeID
  - CategoryID
  - RecipeCategoryDescription

- **Ingredient**
  - IngredientID
  - IngredientName
  - IngredientType

- **RecipeIngredient**
  - RecipeID
  - IngredientID
  - Amount

- **Category**
  - CategoryID
  - CategoryName
  - CategoryDescription
4.0 DATA VISUALIZATION

- GetRecipeByKeyWord
- GetRecipeCountByIngredients
- GetRecipeCountByCountry

4.1 GetRecipeByKeyWord:
- We used FreeTextTable for free text search on the table.
- Search all the columns based on our priority and it will give ranking.

```
ALTER PROCEDURE [dbo].[getRecipeByKeyWord]
    @keyWord varchar(100)
AS
BEGIN
SET FMTONLY OFF;
    if(@keyWord is not null)
    Begin
        if(LEN(@keyWord) = 1)
            BEGIN
                select  r.RecipeCounter,r.RecipeType,r.RecipeCountry,r.Name,r.Serves, r.Picture
                from Recipe r with(nolock) where r.Name LIKE @keyWord+'%'
            END
        ELSE
            BEGIN
                select  r.RecipeCounter,r.RecipeType,r.RecipeCountry,r.Name,r.Serves
                from Recipe r with(nolock)
                inner join FREETEXTTABLE(Recipe,(RecipeType,Ingredients,RecipeCountry,RecipeLanguage),@keyWord) k
                on r.RecipeCounter = k.[KEY]
                ORDER by k.RANK desc
            END
```
4.2 GetRecipeCountByIngredients:

Get recipe count by Ingredients

We created chart for top 20 ingredients vs. count. Collected data for this graph from stored procedure. We used ASP.NET chart controller for drawing charts. This is the graph for top 20 ingredients vs. count by descending order.

In out of 1600 recopies top 20 ingredients are Salt, Pepper, Ice, Onions, Egg, Sugar, Butter, Garlic, Flour, Water, Milk, Tomato, Cream, Nut, Cloves, Cheese, Parsley, Olive oil, Potato, Vinegar.
4.3 Get recipe count by country:

This is a Pie chart from asp.net chart controller. This chart shows top 15 counties by recipe count count in recipe database. Stored procedure use for this chart is getRecipeCountByCountry.
5.0 **AZURE CLOUD:**

We have deployed our code in Microsoft azure by creating a new virtual machine and setting up the IIS for hosting the website in that machine.

5.1 Creating Virtual machine:

5.2 Host website on IIS:
6.0 Technical Environments

6.1 Development

1) Entity framework:
The Entity Framework is a set of technologies in ADO.NET that support the development of data-oriented software applications. We have used entity framework to read data from database and execute stored procedures. We have created complex types for stored procedures and used them throughout the project.

2) Microsoft ASP.NET
3) SQL Server 2008
4) Microsoft Azure
References

http://www.vahrehvah.com/
http://www.recipe.com/
http://www.ingredientpairings.com/
Appendix 1

Sample Asp.net page
```csharp
<%@ Page Language="C#" MasterPageFile="~/Site.Master" AutoEventWireup="true" %>
<%@ Register TagPrefix="ucl" TagName="alphaletter" Src="Control/alphaletter.ascx" %>
<asp:Content ID="Content2" ContentPlaceHolderID="LeftPanel" Runat="Server">
    
</asp:Content>
<asp:Content runat="server" ID="BodyContent" ContentPlaceHolderID="MainContent">
    
    <div style="padding: 2px; text-align: left; margin-left: 40px; margin-bottom: 15px; margin-top: 16px; margin-right: 40px;">
        <asp:Image id="Myranimage" runat="server"
            Width="107" Height="74"
            AlternateText="Recipe Random Image"
            Style="float:left; padding-right: 5px;" />
        <span class="drecipe">
            You need a new dish in a hurry? Stumped by how to make that something special? We can help with your busy lifestyle. Take a look around and review our hundreds of free recipes or submit one of your own favorites.
        </span>
    </div>
    <br />
    <div style="float:left">
        <div style="padding: 2px; text-align: center; margin-left: 26px; margin-bottom: 12px; margin-right: 26px;">
            <ucl:alphaletter id="alpha1" runat="server"></ucl:alphaletter>
        </div>
        <div style="text-align: center; padding-top: 3px;"><asp:Label cssClass="content2" runat="server" id="lbltotalRecipe" />
    </div>
    <br />
    <div style="text-align: center; padding-bottom: 5px;" style="font-family: verdana,arial; font-size: 17px; color: #CC3300;">
        Main Course Recipe</div>
    </div>
</div>
<asp:DataList id="MainCourseCategory" RepeatColumns="3" RepeatDirection="Horizontal" runat="server" HorizontalAlign="Center">
    <ItemTemplate>
        <div style="margin-left: 60px; margin-top: 3px; margin-bottom: 3px; margin-right: 10px;">
            <span class="bluearrow"><span style="font-family: verdana,arial; font-size: 17px; color: #CC3300;">
                &raquo;</span></span>
            <a class="catlink" title="Browse all Recipe">% Eval("Catagory") % recipe" href="% Eval("Catagory",
```
Sample C# code behind
using RecipeSuggestionTool.DataLayer;
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;

namespace RecipeSuggestionTool
{
    public partial class _Default : Page
    {
        BusinessLayer businessLayer = new BusinessLayer();
        protected void Page_Load(object sender, EventArgs e)
        {
            string keyword = String.Empty;
            if (Request.QueryString["keyword"] != null)
            {
                keyword = Request.QueryString["keyword"]; 
                grdResults.DataSource = businessLayer.searchByKeyWord(keyword.Trim());
                grdResults.DataBind();
            }
        }
    }
}
protected void grdResults_RowDataBound(object sender, GridViewRowEventArgs e)
{
    if (e.Row.RowType == DataControlRowType.DataRow)
    {
        ImageButton img = (ImageButton)e.Row.FindControl("btnImg");
        getRecipeByKeyWord_Result item = new getRecipeByKeyWord_Result();
        item = (getRecipeByKeyWord_Result)e.Row.DataItem;
        img.ImageUrl = "~/Images/RecipeImage/" + item.RecipeCounter + ".jpg";  //
        e.Row.Cells[0].Controls.Add(img);
    }
}

protected void grdResults_RowCommand(object sender, GridViewCommandEventArgs e)
{
    if (e.CommandName == "item")
    {
        Response.Redirect("~/RecipeDetails.aspx?RecipeID=" + ((LinkButton)e.CommandSource).Text);
    }
}

protected void grdResults_SelectedIndexChanged(object sender, EventArgs e)
{
}
}
Sample business layer

```csharp
using RecipeSuggestionTool.DataLayer.Model;
using System;
using System.Collections.Generic;
using System.Data;
using System.Data.SqlClient;
using System.Linq;
using System.Web;

namespace RecipeSuggestionTool.DataLayer
{
    public class BusinessLayer
    {
        public List<getRecipeByKeyWord_Result> searchByKeyWord(string keyWord)
        {
            RecipiesRepoEntities recipeRepo = new RecipiesRepoEntities();
            List<getRecipeByKeyWord_Result> searchResult = new List<getRecipeByKeyWord_Result>();
            searchResult = recipeRepo.getRecipeByKeyWord(keyWord).ToList();
            return searchResult;
        }

        public List<getRecipeTypesandCounts_Result> getCategories(int type)
        {
            RecipiesRepoEntities recipeRepo = new RecipiesRepoEntities();
            List<getRecipeTypesandCounts_Result> types = new List<getRecipeTypesandCounts_Result>();
            types = recipeRepo.getRecipeTypesandCounts(type).ToList();
            return types;
        }

        public List<GetArticleCategoryListing_Result> GetArticleCatogories()
        {
            RecipiesRepoEntities recipeRepo = new RecipiesRepoEntities();
            List<GetArticleCategoryListing_Result> result = new List<GetArticleCategoryListing_Result>();
            result = recipeRepo.GetArticleCategoryListing().ToList();
            return result;
        }

        #region Get article details, Update hit counter, Get Category list, Add, Update and Delete Article
        
        /// <summary>
        /// Returns article category list.
        /// </summary>
        /// <returns></returns>
        public IDataReader GetArticleCategoryList()
        {
            get { return DataAccess.GetFromReader("GetArticleCategoryList"); } }

        /// <summary>
        /// Returns 10 newest cooking articles in the right side panel.
        /// </summary>
        /// <returns></returns>
        public IDataReader GetNewestArticleSidePanel(int Top)
        {
            SqlParameter prmTop = new SqlParameter("@Top", SqlDbType.Int, 4);
```
prmTop.Value = Top;

return DataAccess.GetFromReader("GetNewestArticleSidePanel", prmTop);
}

/// <summary>
/// Returns Article Category
/// </summary>
public IDataReader GetArticlesInCategory(int CATID, int OrderBy, int SortBy, int PageIndex, int PageSize)
{
    SqlParameter prmCatId = new SqlParameter("@CATID", SqlDbType.Int, 4);
    prmCatId.Value = CATID;
    SqlParameter prmOrderBy = new SqlParameter("@OrderBy", SqlDbType.Int, 4);
    prmOrderBy.Value = OrderBy;
    SqlParameter prmSortBy = new SqlParameter("@SortBy", SqlDbType.Int, 4);
    prmSortBy.Value = SortBy;
    SqlParameter prmPageIndex = new SqlParameter("@PageIndex", SqlDbType.Int, 4);
    prmPageIndex.Value = PageIndex;
    SqlParameter prmPageSize = new SqlParameter("@PageSize", SqlDbType.Int, 4);
    prmPageSize.Value = PageSize;
    return DataAccess.GetFromReader("GetCategoryArticle", prmCatId, prmOrderBy, prmSortBy, prmPageIndex, prmPageSize);
}

/// <summary>
/// Returns top 10 articles submitted by author/user
/// </summary>
public IDataReader GetTop10OtherArticlesByThisAuthor(int UserID, int AID)
{
    SqlParameter prmUserID = new SqlParameter("@UserID", SqlDbType.Int, 4);
    prmUserID.Value = UserID;
    SqlParameter prmAID = new SqlParameter("@ID", SqlDbType.Int, 4);
    prmAID.Value = AID;
    return DataAccess.GetFromReader("spSelectOtherArticlesByThisAuthor", prmUserID, prmAID);
}

/// <summary>
/// Returns 10 related articles
/// </summary>
public IDataReader GetRelatedArticle(int CatID, int AID)
{
    SqlParameter prmCatId = new SqlParameter("@CatID", SqlDbType.Int, 4);
    prmCatId.Value = CatID;
SqlParameter prmAID = new SqlParameter("@ID", SqlDbType.Int, 4);
prmAID.Value = AID;

return DataAccess.GetFromReader("spSelectRelatedArticle", prmCatId, prmAID);
}

/// <summary>
/// Returns IDataReader Article Category
/// </summary>
public IDataReader GetArticleCategory(int CATID, int OrderBy)
{
SqlParameter prmCatId = new SqlParameter("@CATID", SqlDbType.Int, 4);
prmCatId.Value = CATID;
SqlParameter prmOrderBy = new SqlParameter("@OrderBy", SqlDbType.Int, 4);
prmOrderBy.Value = OrderBy;

return DataAccess.GetFromReader("CategoryPage_Article", prmCatId, prmOrderBy);
}

/// <summary>
/// Returns article category name
/// </summary>
/// <returns></returns>
public string GetArticleCategoryName(int CAT_ID)
{
SqlParameter prmCatID = new SqlParameter("@CAT_ID", SqlDbType.Int, 4);
prmCatID.Value = CAT_ID;

return DataAccess.GetString("GetArticleCategoryName", prmCatID);
}

/// <summary>
/// Update article rating
/// </summary>
/// <returns></returns>
public static int AddArticleRating(int ID, int Rating)
{
SqlParameter prmID = new SqlParameter("@ID", SqlDbType.Int, 4);
prmID.Value = ID;
SqlParameter prmRating = new SqlParameter("@Rating", SqlDbType.Int, 4);
prmRating.Value = Rating;

return DataAccess.Execute("AddArticleRating", prmID, prmRating);
}

/// <summary>
/// Insert article
/// </summary>
/// <returns></returns>
public int AddArticle(article article)
{SqlParameter prmUserID = new SqlParameter("@UserID", SqlDbType.Int, 4);
    prmUserID.Value = article.UID;

SqlParameter prmTitle = new SqlParameter("@Title", SqlDbType.VarChar, 200);
    prmTitle.Value = article.Title;

SqlParameter prmContent = new SqlParameter("@Content", SqlDbType.VarChar, 8000);
    prmContent.Value = article.Content;

SqlParameter prmAuthor = new SqlParameter("@Author", SqlDbType.VarChar, 50);
    prmAuthor.Value = article.Author;

SqlParameter prmCatID = new SqlParameter("@CAT_ID", SqlDbType.Int, 4);
    prmCatID.Value = article.CatID;

SqlParameter prmKeyword = new SqlParameter("@Keyword", SqlDbType.VarChar, 255);
    prmKeyword.Value = article.Keyword;

SqlParameter prmSummary = new SqlParameter("@Summary", SqlDbType.VarChar, 500);
    prmSummary.Value = article.Summary;

    return DataAccess.Execute("spInsertArticle", prmUserID, prmTitle, prmContent, prmAuthor, prmCatID, prmKeyword, prmSummary);
}

/// <summary>
/// Update a users article
/// </summary>
/// <returns>
public int UpdateArticle(article article)
{
SqlParameter prmUserID = new SqlParameter("@UserID", SqlDbType.Int, 4);
    prmUserID.Value = article.UID;

SqlParameter prmID = new SqlParameter("@AID", SqlDbType.Int, 4);
    prmID.Value = article.ID;

SqlParameter prmTitle = new SqlParameter("@Title", SqlDbType.VarChar, 200);
    prmTitle.Value = article.Title;

SqlParameter prmContent = new SqlParameter("@Content", SqlDbType.VarChar, 8000);
    prmContent.Value = article.Content;

SqlParameter prmCatID = new SqlParameter("@CAT_ID", SqlDbType.Int, 4);
    prmCatID.Value = article.CatID;
SqlParameter prmKeyword = new SqlParameter("@Keyword", SqlDbType.VarChar, 100);
prmKeyword.Value = article.Keyword;

SqlParameter prmSummary = new SqlParameter("@Summary", SqlDbType.VarChar, 500);
prmSummary.Value = article.Summary;

return DataAccess.Execute("spUpdateArticle", prmUserID, prmID, prmTitle, prmContent, prmCatID, prmKeyword, prmSummary);

/// <summary>
/// Finalize Insert article
/// </summary>
/// <returns></returns>
public int FinalizeAddArticle(int ID)
{
SqlParameter prmID = new SqlParameter("@ID", SqlDbType.Int, 4);
prmID.Value = ID;

return DataAccess.Execute("FinalizeArticleSubmission", prmID);

/// <summary>
/// Returns last submitted article ID
/// </summary>
/// <returns></returns>
public IDataReader GetLastArticleID
{
    get { return DataAccess.GetFromReader("GetLastArticleID"); }
}

/// <summary>
/// Admin Recipe Manager Delete Recipe
/// </summary>
/// <returns></returns>
public int AdminDeleteArticle(article article)
{
SqlParameter prmID = new SqlParameter("@ID", SqlDbType.Int, 4);
prmID.Value = article.ID;

return DataAccess.Execute("ArticleDelete", prmID);

/// <summary>
/// Returns article search result
/// </summary>
/// <returns></returns>
public IDataReader GetArticleSearchResult(string Search, int CatId, int OrderBy, int SortBy, int PageIndex, int PageSize)
{
SqlParameter prmSearch = new SqlParameter("@Search", SqlDbType.VarChar, 20);
prmSearch.Value = Search;

SqlParameter prmCatId = new SqlParameter("@CATID", SqlDbType.Int, 4);
prmCatId.Value = CatId;
SqlParameter prmOrderBy = new SqlParameter("@OrderBy", SqlDbType.Int, 4);
    prmOrderBy.Value = OrderBy;
SqlParameter prmSortBy = new SqlParameter("@SortBy", SqlDbType.Int, 4);
    prmSortBy.Value = SortBy;
SqlParameter prmPageIndex = new SqlParameter("@PageIndex", SqlDbType.Int, 4);
    prmPageIndex.Value = PageIndex;
SqlParameter prmPageSize = new SqlParameter("@PageSize", SqlDbType.Int, 4);
    prmPageSize.Value = PageSize;
    return DataAccess.GetFromReader("GetArticleSearchResult", prmSearch, prmCatId, prmOrderBy, prmSortBy, prmPageIndex, prmPageSize);

    /// <summary>
    /// Returns all article submitted by an author/user.
    /// </summary>
    public IDataReader GetAllArticleByAuthor(string Author, int OrderBy, int SortBy, int pageIndex, int pageSize)
    {
        SqlParameter prmAuthor = new SqlParameter("@FindByAuthor", SqlDbType.VarChar, 20);
        prmAuthor.Value = Author;
SqlParameter prmOrderBy = new SqlParameter("@OrderBy", SqlDbType.Int, 4);
    prmOrderBy.Value = OrderBy;
SqlParameter prmSortBy = new SqlParameter("@SortBy", SqlDbType.Int, 4);
    prmSortBy.Value = SortBy;
SqlParameter prmPageIndex = new SqlParameter("@PageIndex", SqlDbType.Int, 4);
    prmPageIndex.Value = PageIndex;
SqlParameter prmPageSize = new SqlParameter("@PageSize", SqlDbType.Int, 4);
    prmPageSize.Value = PageSize;
    return DataAccess.GetFromReader("spSelectAllArticleByUser", prmAuthor, prmOrderBy, prmSortBy, prmPageIndex, prmPageSize);
    }

    /// <summary>
    /// Returns all article commented by user.
    /// </summary>
    public IDataReader GetAllArticleCommentedByUser(string Author, int OrderBy, int SortBy, int pageIndex, int pageSize)
    {
SqlParameter prmAuthor = new SqlParameter("@FindByAuthor", SqlDbType.VarChar, 20);
    prmAuthor.Value = Author;
SqlParameter prmOrderBy = new SqlParameter("@OrderBy", SqlDbType.Int, 4);
    prmOrderBy.Value = OrderBy;
SqlParameter prmSortBy = new SqlParameter("@SortBy", SqlDbType.Int, 4);
    prmSortBy.Value = SortBy;
SqlParameter prmPageIndex = new SqlParameter("@PageIndex", SqlDbType.Int, 4);
    prmPageIndex.Value = PageIndex;
SqlParameter prmPageSize = new SqlParameter("@PageSize", SqlDbType.Int, 4);
    prmPageSize.Value = PageSize;
return DataAccess.GetFromReader("spSelectGetUserArticleCommentByUser", prmAuthor, prmOrderBy, prmSortBy, prmPageIndex, prmPageSize);
}
prmCommentID.Value = comment.ID;
SqlParameter prmAuthor = new SqlParameter("@Author",
SqlDbType.VarChar, 50);
prmAuthor.Value = comment.Author;
SqlParameter prmEmail = new SqlParameter("@Email", SqlDbType.VarChar, 50);
prmEmail.Value = comment.Email;
SqlParameter prmComment = new SqlParameter("@Comments",
SqlDbType.VarChar, 350);
prmComment.Value = comment.Comments;
SqlParameter prmUserID = new SqlParameter("@UserID", SqlDbType.Int, 4);
prmUserID.Value = comment.UID;
return DataAccess.Execute("spInsertArticleComment", prmCommentID,
prmAuthor, prmEmail, prmComment, prmUserID);
}

/// <summary>
/// Update article comment
/// </summary>
/// <returns></returns>
public int UpdateArticleComment(Comment comment)
{
SqlParameter prmID = new SqlParameter("@ID", SqlDbType.Int, 4);
prmID.Value = comment.ID;
SqlParameter prmComment = new SqlParameter("@Comment",
SqlDbType.VarChar, 350);
prmComment.Value = comment.Comments;
return DataAccess.Execute("UpdateArticleComments", prmID, prmComment);
}

/// <summary>
/// Delete individual article comment
/// </summary>
/// <returns></returns>
public int DeleteArticleComment(Comment comment)
{
SqlParameter prmID = new SqlParameter("@ID", SqlDbType.Int, 4);
prmID.Value = comment.ID;
SqlParameter prmAID = new SqlParameter("@AID", SqlDbType.Int, 4);
prmAID.Value = comment.RECID;
return DataAccess.Execute("AdminDeleteArticleComments", prmID,
prmAID);
}

/// <summary>
/// Delete multiple/batch article comment
/// </summary>
/// <returns></returns>
public int DeleteMultipleArticleComment(string CsvID, string CsvItemID)
{
    SqlParameter prmCsvID = new SqlParameter("@CsvID", SqlDbType.VarChar, 1000);
    prmCsvID.Value = CsvID;

    SqlParameter prmCsvItemID = new SqlParameter("@CsvItemID", SqlDbType.VarChar, 1000);
    prmCsvItemID.Value = CsvItemID;

    return DataAccess.Execute("AdminDeleteArticleComments", prmCsvID, prmCsvItemID);
}