INTERACTIVE INFORMATION CENTER

Charles M Comella, CET
James K Pugh, CET
Professor Haas, Advisor

College of Engineering & Applied Science
ECET Final Report
Submitted to: Professor Michael Haas

April 25th, 2013
Dear Professor Michael Haas,

Attached is our final report on “Interactive Information Center.” This report was requested by the ECET faculty to be completed by each group.

This final report covers the details that led to the competition of our design project. In this report you will find information including our problem, solution and methodology. After these sections, we will put our code that made our project work. Our code is quite long since we have a software only project.

During the process of developing our project, we were advised by you as well as Charles’ co-workers. Charles’ co-workers did not directly assist in this project; however they did teach him many techniques used in this design.

Sincerely,

Charles M Comella
330-447-7038
comellcm@gmail.com

James K Pugh
419-467-7394
pughjs@mail.uc.edu
Department of Electrical and Computer Engineering Technology

Interactive Information Center

Charles Comella and James Pugh

April 19, 2013

Submitted in partial fulfillment of the degree of Bachelor of Science in Computer Engineering Technology

Student Signature______________________________

Student Signature______________________________

Advisor Signature______________________________
Acknowledgments

There are many people who helped with this project. Knowingly or unknowingly many people helped motivate us in completing this project.

- Professor Michael Haas, for helping with the idea as well as hosting the SQL Server.

- Professor Xuefu Zhou, for being in charge of the senior design class fall 2012.

- Professor James Everly, for being in charge of the senior design class spring 2013.

- Professor Max Rabiee, Kathy Ossman, David Tashjian, Roy Ford, Elvin Stepp, for the education throughout the years making projects like this possible.

- Michael Comella, for assistance building a stand to house part of our project.

- Jon and Julie Pugh for the general idea of this project and gathering various coupons and advertisements.

- Bob Hendershot, for his advice with web applications.

- Ted Gentsch, for working with Charles so that he may be in Cincinnati whenever needed.

- Microsoft and the University of Cincinnati, for giving software like visual studio and SQL management studio at no cost to us.
# Table of Contents

Abstract -------------------------------------------------- 6  
Introduction --------------------------------------------- 6  
  Problem----------------------------------------------- 7  
  Solution----------------------------------------------- 7  
  Credibility-------------------------------------------- 7  
    Charles M Comella --------------------------------- 7  
    James K Pugh -------------------------------------- 8  
Goals and Methodology------------------------------------ 8  
Overview------------------------------------------------- 8  
Discussion----------------------------------------------- 9  
  Project Concept---------------------------------------- 9  
  Design Objectives-------------------------------------- 12  
  Technical Approach------------------------------------- 12  
Budget--------------------------------------------------- 15  
Timeline------------------------------------------------ 15  
Problems Encountered------------------------------------- 15  
Future Plans--------------------------------------------- 16  
Conclusion----------------------------------------------- 17  
References----------------------------------------------- 18  
Appendix A---------------------------------------------- 19  
  Main Screen Code--------------------------------------- 18  
Appendix B---------------------------------------------- 47  
  Touch Screen Code-------------------------------------- 47  
Appendix C---------------------------------------------- 65  
  Data Grabber Application------------------------------- 65  
Appendix D---------------------------------------------- 71  
  Microsoft SQL Server Stored Procedures---------------- 71
## List of Illustrations

- Figure 1 – Project Sample .................................................. 6
- Figure 2 – Main Screen .......................................................... 10
- Figure 3 – Navigational Flow Chart ..................................... 11
- Figure 4 – Technical Flow Chart ......................................... 14
- Figure 5 – Timeline ............................................................. 15
- Figure 6 – Budget ............................................................... 15
Abstract

The Interactive Information Center delivers timely information to hotel guests using a friendly and fast touch-screen interface. It can be easily placed in hotel lobbies, restaurants, and conference centers. It provides up-to-date local weather forecasts, tide clocks, beach conditions, and hotel activities, plus targeted advertising for local businesses and services.

Introduction

This report outlines the project and the time spent planning and executing the necessary parts to get the Interactive Information Center online and functional.

Below in Figure 1 is the picture of the finished product as it was presented at the Technical Exposition. The main objective was to stimulate a privately owned hotel's information centers and give them a chance in which they can offer the same information center as the larger chain hotels with the available funding.

Figure 1 – Project Sample
The system contains a small computer that fits inside the stand that drives both the touch screen and the display board. The idea was to make as little visible as possible and to simulate what a tablet would work like. The touch screen is a USB powered monitor that has a primitive touch style. This was cheap and fit in our budget nicely; however it had some issue with the operation during Tech Expo.

More detailed analysis can be found in this report.

**Problem**

As a tourist, you need to be able to access information on where you are traveling. Most importantly is finding information on dining, weather, and even seeing some advertisements based on the location of where you are at. All of these things contribute to a local economy and the success of a tourist town. Larger chain hotels can afford to have computer systems that offer all this information due to their large disposable income. But the smaller hotels cannot afford to do so. This gives a smaller locally owned hotel to display weather, restaurants, local events, and hotel events to their customer in a timely and simple manner.

**Solution**

Create an application that is simple, cost effective and very user friendly to display this needed information. Paired with all the advertising, this makes a very affordable and effective solution for a local hotel. There will be an overhead LCD display that has a screen with basic quick information for anyone walking by and curious what the weather might be. Then for the users who want to see what the town they are touring has to offer, the tablet provides a quick and simple navigation solution for browsing the Interactive Information System.

**Credibility**

Charles Comella and James Pugh both cooped in the C# world. Together they have 8 years experience programming.

**Charles M Comella**

Charles completed all of his co-op terms at Lauren International working in software development. During his time with Lauren, he used ASP, ASP.NET, C#, SQL, HTML, CSS, AJAX, JQuery, Javascript. He also dealt with fixing broken computers and whatever various tasks were required of him.
James discovered his passion for software in high school which lead him to pursue his degree in CET at The University of Cincinnati. With help from his father and his family’s background in entrepreneurship the idea of the Interactive Information Center became possible. James competed 4 of his co-op terms at Intelligrated working in Software R&D and Software Controls. During his time with R&D is when he developed his C#, SQL, and XML skills that were utilized throughout this project.

Goals and Methodology

The main goal of this project is to create an application that will take all the weather, advertising, and local information and be able to display it in an orderly manner. This information will be accessible through a simple navigational menu that is on a tablet or touch screen. The interface needs to be friendly and approachable. A lot of people are scared of change, worried about new technology, or flat out want nothing to do with technology. This application and user interface will be friendly and simple. This system will also create a new business opportunities for other local businesses. It takes networking to advertise, this will hopefully tie together more local businesses through advertising. The system is also “greener”. It should replace any sort of brochures and keep the hotel workers from getting caught talking to the customer who never stops talking and ties up the desk attendant.

Overview

The remainder of this document outlines in great detail how the project was completed. The rest of this document covers the following sections: Project Concept, Design Objectives, Technical Approach, Timeline, Budget, Problems, and Future Changes.
DISCUSSION

Project Concept

The concept behind the project was to create a simple information solution for a hotel or resort that does not have the funding to offer any sort of a technological information center. Most privately owned hotels or resorts only offer a brochure stand, small amounts of information in the rooms about dining, and then any other needed information (weather) on the television. This product offers a fast and easy interactive solution to gathering data and advertising for local businesses. The Interactive Information Center also provides a friendly user interface for any user who approaches and uses it. A view of the main screen on the overhead display is on the next page on Figure 2.

In 2011, $496.9 billion was spent on advertising. So as one can assume, being self-employed and being able to afford advertising isn’t easy. So on top of helping the customers, we are also able to help stimulate the local economy. This product will help the local businesses not only network, but gives a cheaper alternative to advertising. As Figure 3, the navigational flow chart shows how simple navigating through the application really is.
Figure 2 – Main Screen
Interactive Information Center

User Selects Menu Item
Dining
Weather
Points of Interest
Events
User Selects Category
Timeout or User Selects "Back" and goes to Default Screen
Display list of restaurants based on category
User Selects Restaurant
Diner displays information on overhead screen
User Selects Restaurant
Selected POI displays on large screen
User Selects Category
Selected event displays on large screen
User Selects Category
Points of Interest
Events
User Selects Category
Weather
Menu Selects "Home" and goes to Default Screen
Figure 3 – Navigational Flow Chart
Design objectives

The design objectives were very simple. Grab data from the internet and information from local businesses to display to tourists who visit. All at keeping it to a low cost for any hotel or resort who wants a technological advancement in their lobby. The weather information is updated every 5 minutes from (www.weatherunderground.com) API. The information provided will be current weather, future weather, current radar, and tide information. Tourists need to know the weather and this is one of the simplest and quickest approaches to give them such data.

The data will be stored on a server locally and the customer will access a webpage that will allow them to present their data. This is ideal because the customer has no need for any sophisticated technology. The customer only needs a 1080p television, computer, tablet, and internet access. This helps keep costs at an absolute minimum. This prevents any sort of issues with the data so the administrator can verify or fix any of the errors on site, so at the end of the day the customer won’t need much assistance to keep the system up 100% of the time.

Technical Approach

How the project works is there are two screens, one will be a touch screen and one will be a large display screen. The touch screen will be how the user would navigate the large display screen through the system. The user will have the following options; weather, dinning, activities, and local. Each category will have their own screen displayed after the user has made their selection. Each screen will vary depending on the users selection.

In order to accomplish the main navigation and goal of this project, it is based around ASP.NET with C# code behind. This allowed flexibility and access to JavaScript, JQuery, HTML5, CSS3, AJAX, and Microsoft SQL. All eight of these languages are designed to work with one another.

ASP.NET with C# is a server side language that is being used to gather the data off the SQL Server. Once ASP.NET is loaded, it is required to have a reload in order to gather new data. The way to work around this problem is with AJAX. AJAX is a JavaScript and JQuery technique that gives us the ability to call another background ASP.NET page so that we may check our database without having an effect on our visual page. JavaScript is client-side scripting language used in most HTML pages that gives it more flexibility. JQuery is also a client-side scripting language and is a JavaScript add-on with various functions and abilities that has become invaluable to our project. HTML5 is the basic structure and markup of our page. CSS3 is the design of our page. Microsoft SQL is our database and gives us a place to store our data as well as manipulate the data.
Once the user has made a selection on the touch screen, the program will update the database telling where it wishes to navigate to. The main screen uses AJAX to check the database and see if it should navigate the main screen elsewhere. There is a timeout function throughout the entire project, meaning that if the display screen stays at one page too long, it will return the system to the first page. This was put in place so that the next user will be able to start their experience from the main screen.

The weather information is updated every 5 minutes by the “Data Grabber”. The information gathered is the weather and tides based on the city or zip code. The data came from an API call made to www.weatherunderground.com. The data returned is in XML so the data grabber gathers the data, parses out the necessary information, and stores it to our database for the program to use. This application also gets an up to date radar image that is displayed on the main screen and weather screen.
Figure 4 – Technical Flow Chart
Timeline

The timeline for this project was pretty straightforward, since it was 100% software-based, there was a lot of coding. With coding comes lots of testing.

![Timeline](image)

Figure 5 – Timeline

Budget

Throughout the project the budget stayed exactly as planned. Since it was all software-based, we had everything we needed. Our man hours were right around 200 hours plus or minus a few.

<table>
<thead>
<tr>
<th>Hotel Information Center</th>
<th>Budget</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Item</strong></td>
<td><strong>Cost</strong></td>
</tr>
<tr>
<td>Computers</td>
<td>400</td>
</tr>
<tr>
<td>Television</td>
<td>700</td>
</tr>
<tr>
<td>USB Screen</td>
<td>125</td>
</tr>
<tr>
<td>Display Casing</td>
<td>100</td>
</tr>
<tr>
<td>Poster Board</td>
<td>50</td>
</tr>
<tr>
<td>Server + Server Licenses</td>
<td>1000</td>
</tr>
<tr>
<td>MS SQL Server 2008 R2 Dev. License</td>
<td>150</td>
</tr>
<tr>
<td>Visual Studio 2012</td>
<td>450</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>People</th>
<th>Hours</th>
<th>Hourly Rate</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>2</td>
<td>$20/HR</td>
<td>$6,000</td>
</tr>
</tbody>
</table>

Total: $11,575

Figure 6 – Budget

Problems Encountered

Some of the problems encountered throughout the process of creating the Interactive Information Center were mainly due to working on this remotely. Charles was living over 3 hours away from Cincinnati, so most complications occurred due to software and hardware malfunctions. In order for Charles and
James to work on this project remotely, ‘Google Drive’ was used to synchronize the files that were being edited. This was a simple design decision because it was free and handled all the synchronizing automatically, but there were several occasions where Google Drive application on the PC would crash the changes would not update. A couple times this posed a large problem due to one person needed those changes to view/test/fix them.

Along with the distance between James and Chas, meeting deadlines wasn’t as easy as it would be if in person. If something got behind, there wasn’t much one could do but wait for it to get done. Due to each members skill set, one couldn’t make up for lost time on the other members portion where the deadline wasn’t met. Some deadlines were just not feasible since each member couldn’t sit down any day of the week to plan things out.

Our “Data Grabber” was used to grab real time information such as the weather and store it locally to the machine for use in our webpage that is displayed. The biggest complication was that different machines have different file path and folder structures. So hard coding the information to be saved on one machine was different from second machine. This caused exceptions and for the application to crash, it took a little bit to solve but then had to decide on a code change that required changes based on each machine it being run on.

One of the biggest challenges and problems of this project is that it required internet at all time and VPN access to The University of Cincinnati’s Internet. The basic navigation of the web page and all the information was held on a SQL database that the computer would remotely connect to, to display desired information. The second our server went down, we lost internet connection, or had SQL errors; the application would stop working instantly.

**Future Plans**

Continuing this product would be an adventure. However, both of us understand that it would take more work and improvements. Our database needs to be better thought out and constructed to handle multiple systems at the same time. There are a few flaws with the main screen design as well as the way we are designing it. The touch screen flashes and could be considered distracting to the end goal.

The Interactive Information Center has a lot of room for expansion and upgrades as one could assume. The finished product as it is now provides a simple user interface, navigation, and data for the customer. But there is talk of many upgrades if this product were to be marketed. Anywhere from printing abilities, administrative page, application for the hotel to maintain the database, and user interface rework.
The biggest improvement would be the ability to print the desired information or make it available via Quick Response Code (QR Code). Once the user finds their desired destination, they can easily print off a directions list with contact information or use a QR Code to have that information sent directly to their Smart Phone (Android or iPhone). This gives more versatility to the end user of how to access their data for their trip.

The idea behind the project is to have different companies work with the hotel for advertising and getting their information displayed on the Interactive Information Center. This problem at hand is that one hotel could have dozens of advertisements, restaurants, and events. Being that each advertisement would cost money, someone needs to keep track of which companies have paid for each month. This is a lot of work for one administrator who is overlooking several Information Centers, so it would be easier if the hotels would manage their own advertisements and information displayed.

The User Interface could use a lot of re-work. Right now it is basic and to the point, but it does not have the modern feel that it really needs. The coloring and shapes need to flow better compared to the squares and rectangles. Each system installed can have its own formatting; the formatting just needs to be appropriate for each location. The only restriction would be that “too loud” of a formatting theme might cause people to feel intimidated and not want to use the system.

A product like this would only be as good as the advertising that powered it. If we were not able to gather a profitable level of advertising then the product would not be able to happen. However, I feel that we are both skilled and have a good grasp for relative green people to the advertising world.

Conclusion

The final outcome of this project may not have been exactly what we were attempting to finish. However, we both believe that there is a good base that could be turned into a serviceable product someday. Marketability as it is right now may not far off, but this must be more dynamic in how it pulls data. Given more time, we would have liked to improve how we navigate and make it more acceptable to operate more than one system. We did accomplish most of the goals we set and exceeded many expectations we had. The future of this product appears bright and we would like to test how this product will fair someday.
References

Main Screen Page Code

MainPage.aspx

```csharp

<!DOCTYPE html>
<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
    <link href="StyleSheet.css" type="text/css" rel="stylesheet" />
    <script src="JavaScript.js"></script>
    <script src="http://code.jquery.com/jquery-1.9.1.min.js"></script>
    <script src="http://code.jquery.com/jquery-migrate-1.1.0.min.js"></script>
    <title></title>
    <style type="text/css">
        .tr{
            height: 100%;
        }
    </style>
</head>
<body class="blahblah">

<script type="text/javascript">
    $(document).ready(function () {
        setInterval('jquerytest()', 600);
        setInterval('timeout()', 30000);
    });

    function jquerytest() {
        var response;
        //$('#tablechecker').click();
        $.get("databasecheck.aspx", function (response) {
            //alert(response);

            if (response == "1") {
                //alert("useable");
                window.location.replace("activites.aspx");
            }
            if (response == "2") {
                //alert("useable");
                window.location.replace("local.aspx");
            }
            if (response == "3") {
                //alert("useable");
                window.location.replace("dinning.aspx");
            }
            if (response == "4") {
                //alert("useable");
                window.location.replace("weather.aspx");
            }
        });
    }

</script>
```
<form id="form1" runat="server">
<table style="width: 100%; flex-align: center;">
<tr>
<td>
<iframe id="weather" src="weathermini.aspx" width="520px" height="520px"></iframe>
</td>
<td class="auto-style1">
<iframe src="dinningmini.aspx" width="520px" height="520px"></iframe>
</td>
<td colspan="2">
<asp:AdRotator ID="AdRotator1" AdvertisementFile="Ad1.xml" runat="server"
Target="_blank" />
</td>
</tr>
<tr>
<td>
<iframe src="activitesmini.aspx" width="520px" height="520px"></iframe>
</td>
<td class="auto-style1">
<iframe src="localmini.aspx" width="520px" height="520px"></iframe>
</td>
<td colspan="2">
<asp:AdRotator ID="AdRotator2" AdvertisementFile="Ad2.xml" runat="server"
Target="_blank" />
</td>
</tr>
</table>
</form>
</body>
</html>
MainPage.aspx.cs

using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Data;
using System.Data.SqlClient;
using System.Data.Common;
using System.Data.Linq;
using System.Security.Principal;
using System.Configuration;
using System.Data.SqlClient;
using System.Xml.Linq;
using System.Web.UI.WebControls.WebParts;
using System.Web.Configuration;

public partial class MainPage : System.Web.UI.Page
{
    protected void Page_Load(object sender, EventArgs e)
    {
    }
    protected void tablechecker_Click(object sender, EventArgs e)
    {
        string hello;
        //4th try
        SqlConnection con = new SqlConnection("Data Source=10.63.1.116;Initial Catalog=JP_CM_SeniorDesign;User ID=comellcm; Password=qazwsx");
        con.Open();
        SqlCommand cmd = new SqlCommand("spNav_SelectNav", con);
        cmd.Connection = con;
        cmd.CommandType = CommandType.StoredProcedure;
        SqlParameter local2 = cmd.Parameters.Add("@local", SqlDbType.Int);
        local2.Direction = ParameterDirection.ReturnValue;
        cmd.ExecuteNonQuery();
        hello = local2.Value.ToString();
        //Response.Write("<script language=javascript>alert("");
        //Response.Write(hello);
        //Response.Write(""");
        if (hello == "1")
        {
            Response.Write("<script language=javascript>alert("");
            Response.Write(hello);
            Response.Redirect("activites.aspx");
        }
        if (hello == "2")
        {
            Response.Write("<script language=javascript>alert("");
            Response.Redirect("local.aspx");
        }
    }
}
if (hello == "3")
{
    Response.Write("<script language=javascript>alert('" + hello + 
"')</script>");
    Response.Redirect("dining.aspx");
}

if (hello == "4")
{
    Response.Write("<script language=javascript>alert('" + hello + 
"')</script>");
    Response.Redirect("weather.aspx");
}
Activites.aspx

@ Page Language="C#" AutoEventWireup="true" CodeFile="activites.aspx.cs"
Inherits="activites" %>

<!DOCTYPE html>
<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
  <script src="JavaScript.js"></script>
  <script src="http://code.jquery.com/jquery-1.9.1.min.js"></script>
  <script src="http://code.jquery.com/jquery-migrate-1.1.0.min.js"></script>
  <title></title>

  <style type="text/css">
    html {
      overflow-x: hidden;
      overflow-y: hidden;
    }

    .minititle {
      text-align: center;
      /*color:#D69345;*/ color: #58FA58;
      font-family: Arial;
      font-weight: bold;
      font-size: 24px;
    }

    .widget {
      width: 100%;
      background-color: #6ED1FF;
      color: black;
      opacity: .8;
      height: 480px;
    }

    .bottomforcer {
      width: 100%;
      position: absolute;
      bottom: 35px;
      text-align: center;
    }

    .blahblah {
      background-color: #323B66;
    }
  </style>
</head>
<body class="blahblah">
<form id="form1" runat="server" class="widget">
  <script type="text/javascript" runat="server" class="widget"  

$(document).ready(function () {
    setInterval('jquerytest()', 600);
    setInterval('timeout()', 300000);
});

function jquerytest() {
    var response;
    //$('#tablechecker').click();
    $.get("databasecheck.aspx", function (response) {
        //alert(response);
        if (response == "2") {
            //alert("useable");
            window.location.replace("local.aspx");
        } if (response == "3") {
            //alert("useable");
            window.location.replace("dinning.aspx");
        } if (response == "4") {
            //alert("useable");
            window.location.replace("weather.aspx");
        } if (response == "0") {
            window.location.replace("MainPage.aspx");
        }
    });
});
"<br />
</ItemTemplate>
<SeparatorTemplate>
   <hr color="#0431B4" size="5" width="100">
</SeparatorTemplate>
<FooterTemplate>
   </table>
</FooterTemplate>
</asp:DataList>
</table>
<table style="width: 100%;" class="bottomforcer">
<tr style="text-align: center;">
   <td>
      <asp:AdRotator ID="AdRotator2" AdvertisementFile="Ad1.xml" runat="server" Target="_blank" />
   </td>
   <td>
      <asp:AdRotator ID="AdRotator1" AdvertisementFile="Ad2.xml" runat="server" Target="_blank" />
   </td>
</tr>
</table>"
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Data;
using System.Security.Principal;
using System.Configuration;
using System.Data.SqlClient;
using System.Xml.Linq;
using System.Web.UI.WebControls.WebParts;
using System.Web.Configuration;

public partial class activites : System.Web.UI.Page
{
    protected void Page_Load(object sender, EventArgs e)
    {
    }

    protected void tablechecker_Clickactivites(object sender, EventArgs e)
    {
        string hello;
        //4th try

        SqlConnection con = new SqlConnection("Data Source=10.63.1.116;Initial Catalog=JP_CM_SeniorDesign;User ID=comellcm; Password=qazwsx");
        con.Open();
        SqlCommand cmd = new SqlCommand("spNav_SelectNav", con);
        cmd.Connection = con;
        cmd.CommandType = CommandType.StoredProcedure;
        SqlParameter local2 = cmd.Parameters.Add("@local", SqlDbType.Int);
        local2.Direction = ParameterDirection.ReturnValue;
        cmd.ExecuteNonQuery();
        //Response.Write("<script language=javascript>alert("");
        //Response.Write(local2.Value.ToString());
        //Response.Write("" + local2.Value + "]</script>");
        hello = local2.Value.ToString();

        if (hello == "0")
        {
            Response.Redirect("MainPage.aspx");
        }
    }
}
$(document).ready(function () {
    setInterval('jquerytest()', 600);
    setInterval('jquerygotorest()', 601);
    setInterval('timeout()', 300000);
});

function jquerygotorest() {
    //alert('hi');
    var response;
    $.get("databaserestcheck.aspx?check=type", function (response) {
        //alert(response);
        if (response == "0") {
            //alert('stay here');
        } else if (response == "1") {
            //alert('response is 1');
            window.location.replace("MainPage.aspx");
        } else {
            //alert('hi');
            window.location.replace("dinning_type.aspx?type=" + response);
        }
    });

});

function jquerytest() {
    var response;
    $.get("databasecheck.aspx", function (response) {
        //alert(response);

        if (response == "1") {
            //alert('useable');
            window.location.replace("activites.aspx");
        } else if (response == "2") {
            //alert('useable');
            window.location.replace("local.aspx");
        } else if (response == "4") {
            //alert('useable');
            window.location.replace("weather.aspx");
        } else if (response == "0") {
            window.location.replace("MainPage.aspx");
        } else {
            //alert("dinning_type.aspx?type=" + response);
        }
    });
}
```html
//window.location.replace("dinning_type.aspx?type="+ response);
}
})
</script>

<asp:SqlDataSource ID="SqlDataSource1" runat="server"
ConnectionString="%$ ConnectionStrings:JP_CM_SeniorDesignConnectionString %"
SelectCommand="SELECT distinct top 16 [Address], [Name], [Phone], [Description], [Category_Key], [Logo_FilePath], Map_FilePath FROM [Restaurants_Tbl] order by Phone ASC"/>

<div>
<table class="minitable" style="width:100%;">
<tr>
<td class="minititle">Restaurants</td>
</tr>
</table>
</div>

<asp:DataList ID="DataList1" runat="server">
<HeaderTemplate>
</HeaderTemplate>
<ItemTemplate>
<tr style="font-size: 20px; font-family: Cambria;">
<td style="width:20%">
<asp:Label ID="Label1" runat="server" Text='<%# Eval("Name")%>' /></td>
<td style="width:20%">
<asp:Label ID="Label2" runat="server" Text='<%# Eval("Phone")%>' /></td>
<td style="width:20%">
<asp:Label ID="Label3" runat="server" Text='<%# Eval("Address")%>' /></td>
<td style="width:20%">
<b><asp:Label ID="Label5" runat="server" Text='<%# Eval("Category_Key")%>' /></b></td>
</tr>
</ItemTemplate>
</asp:DataList>
</div>

<table style="width: 100%;" class="bottomforcer">
<tr style="text-align: center;">
<td>
<asp:AdRotator ID="AdRotator2" AdvertisementFile="Ad2.xml" runat="server" Target="_blank" />
</td>
</tr>
</table>
```
<asp:AdRotator ID="AdRotator1" AdvertisementFile="Ad1.xml" runat="server" Target="_blank" />
</td>
</tr>
</table>
</div>
</form>
</body>
</html>
Dinning_ind.aspx

<%@ Page Language="C#" AutoEventWireup="true" CodeFile="dinning_ind.aspx.cs"
Inherits="dinning_ind" %>

<!DOCTYPE html>
<html xmlns="http://www.w3.org/1999/xhtml">
<head id="Head1" runat="server">
<script src="http://code.jquery.com/jquery-1.9.1.min.js"></script>
<script src="http://code.jquery.com/jquery-migrate-1.1.0.min.js"></script>
<title></title>
<style type="text/css">
.minitable
{
    margin: 0 auto;
}

.iframe
{
    flex-align: center;
}

.blahblah
{
    background-color: #323B66;
}

.minititle
{
    text-align: center;
    /*color:#D69345;*/
    color: #58FA58;
    font-family: Arial;
    font-weight: bold;
    font-size: 24px;
}

.widget
{
    width: 100%;
    background-color: #6ED1FF;
    color: black;
    opacity: .8;
    height: 480px;
}

html
{
    overflow-x: hidden;
    overflow-y: hidden;
}

.bottomforcer
{
    width: 100%;
    position: absolute;
$(document).ready(function () {
    setInterval('jquerytest()', 600);
    setInterval('jquerygotorest()', 601);
    setInterval('jquerygotorest2()', 603);
    setInterval('timeout()', 300000);
});

function jquerygotorest() {
    //alert("hi");
    var response;
    $.get("databaserestcheck.aspx?check=ind", function (response) {
        var x, y;
        x = document.getElementById("Label5").innerText;
        y = response;

        var x2, y2;
        x2 = jQuery.trim(x);
        y2 = jQuery.trim(y);

        //alert(x2.length);
        //alert(y2.length);
        //alert(response);
        //alert(x2);
        //alert(y2);
        if (response == "0") {
            //alert("stay here");
        } else if (x2 == y2) {
            //alert("what??");
        } else if (x2.length == y2.length)
            //alert("response is 1");
            window.location.replace("MainPage.aspx");
        else {
            window.location.replace("dinning_ind.aspx?type=" + response);
        }
    });
}

function jquerytest() {
if (response == "1") {
    //alert("useable");
    window.location.replace("activites.aspx");
}
else if (response == "2") {
    //alert("useable");
    window.location.replace("local.aspx");
}
else if (response == "4") {
    //alert("useable");
    window.location.replace("weather.aspx");
}
else if (response == "0") {
    window.location.replace("MainPage.aspx");
} else {
    //alert("dinning_type.aspx?type=" + response);
    //window.location.replace("dinning_type.aspx?type=" +
    response);
}
}
});

function jquerygotorest2() {
    var response;
    $.get("databaserestcheck.aspx?check=type", function (response) {

        //alert(response);
        if (response == "0") {
            //alert("response is 0");
            window.location.replace("dinning.aspx");
        }
    }
});
</script>
<asp:SqlDataSource ID="SqlDataSource1" runat="server"
ConnectionStrings="%$ ConnectionStrings:JP_CM_SeniorDesignConnectionString
%$"></asp:SqlDataSource>
<table style="display:none;">
<tr>
    <td style="display:none;">
        <asp:Label ID="Label15" runat="server"></asp:Label>
    </td>
</tr>
</table>
</div>
<table border="0" class="minitable" style="width:100%">
<tr style="background-color: #0431B4" class="minititle">
    <td colspan="6" class="minititle">Restaurants</td>
</tr>
<table style="width: 100%;" class="bottomforcer">
<tr style="text-align: center;">
<td>
<asp:AdRotator ID="AdRotator2" AdvertisementFile="Ad2.xml" runat="server" />
</td>
</tr>
</table>
Dinne

g_ind.aspx.cs

using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Text;

public partial class dinning_ind : System.Web.UI.Page {
    protected void Page_Load(object sender, EventArgs e) {
        string type = Request.QueryString["type"];  
        //string sm_title = Rdr.GetString(1).ToString().replace("%20", "+"); 
        //type = type.GetString(1).ToString().replace("%20", " ");
        string ind = type.ToString().Replace("%20", "-");

        Label5.Text = type.ToString();
        //Response.Write ind;

        SqlDataSource1.SelectCommand = "SELECT top 1 * FROM [Restaurants_Tbl] where name like "+ ind + "%";";
    }
}
Dinning_type.aspx

```csharp
<%@ Page Language="C#" AutoEventWireup="true" CodeFile="dinning_type.aspx.cs" Inherits="dinning_type" %>

<!DOCTYPE html>
<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
    <script src="http://code.jquery.com/jquery-1.9.1.min.js"></script>
    <script src="http://code.jquery.com/jquery-migrate-1.1.0.min.js"></script>
    <title></title>
    <style type="text/css">
        .minitable
        {
            margin: 0 auto;
            width: 100%;
        }

        .iframe
        {
            flex-align: center;
        }

        .blahblah
        {
            background-color: #323B66;
        }

        .minititle
        {
            text-align: center;
            color: #58FA58;
            font-family: Arial;
            font-weight: bold;
            font-size: 24px;
        }

        .widget
        {
            width: 100%;
            background-color: #6ED1FF;
            color: black;
            opacity: .8;
            height: 480px;
        }

        html
        {
            overflow-x: hidden;
            overflow-y: hidden;
        }

        .bottomforcer
        {
            width: 100%;
            position: absolute;
            bottom: 35px;
        }
    </style>
</head>
<body>
</body>
</html>
```
$(document).ready(function () {
    setInterval('jquerytest()', 600);
    setInterval('jquerygotorest()', 601);
    setInterval('jquerygotorest2()', 601);
    setInterval('timeout()', 300000);
});

function jquerygotorest() {
    //alert("hi");
    var response;
    $.get("databaserestcheck.aspx?check=ind", function (response) {
        if (response == "0") {
            //alert("stay here");
        } else if (response == "1") {
            //alert("response is 1");
            window.location.replace("MainPage.aspx");
        } else {
            //alert("hi");
            //alert(response);
            window.location.replace("dinning_ind.aspx?type=" + response);
        }
    });
}

function jquerytest() {
    var response;
    $.get("databasecheck.aspx", function (response) {
        //alert(response);

        if (response == "1") {
            //alert("useable");
            window.location.replace("activities.aspx");
        } else if (response == "2") {
            //alert("useable");
            window.location.replace("local.aspx");
        } else if (response == "4") {
            //alert("useable");
            window.location.replace("weather.aspx");
        } else if (response == "0") {
            window.location.replace("MainPage.aspx");
        } else {
            //alert("dinning_type.aspx?type=" + response);
        }
    });
}
//window.location.replace("dinning_type.aspx?type="+ response);
} });

function jquerygotorest2() {
    var response;
    $.get("databasestcheck.aspx?check=type", function(response) {
        //alert(response);
        if (response == "0") {
            //alert("response is 0");
            window.location.replace("dinning.aspx");
        }
    });
}
</script>
<asp:SqlDataSource ID="SqlDataSource1" runat="server"
    ConnectionString="%$ ConnectionStrings:JP_CM_SeniorDesignConnectionString %$"></asp:SqlDataSource>
<table class="minitable" style="width: 100%;">
    <tr style="background-color: #0431B4; width: 100%;">
        <td colspan="4" class="minititle">Restaurants</td>
    </tr>
    <asp:DataList ID="DataList1" runat="server"
        DataSourceID="SqlDataSource1">
        <HeaderTemplate>
            <tr style="width: 100%;">
                <td style="width: 20%"><b>Name</b></td>
                <td style="width: 20%"><b>Phone</b></td>
                <td style="width: 20%"><b>Address</b></td>
                <td style="width: 20%"><b>Type</b></td>
            </tr>
        </HeaderTemplate>
        <ItemTemplate>
            <tr style="font-size: 20px; font-family: Cambria; width: 100%;">
                <td style="width: 20%"><asp:Label ID="Label1" runat="server" Text='"%# Eval("Name")"'></td>
                <td style="width: 20%"><asp:Label ID="Label2" runat="server" Text='"%# Eval("Phone")"'></td>
            </tr>
        </ItemTemplate>
    </asp:DataList>
<asp:Label ID="Label3" runat="server" Text='\$
\$
\$
\$
\$
\$
\$
\$
\$
\$
\$
\$
\$
\$
\$
\$
\$
\$
\$
\$
\$
\$
\$
\$
\$
\$
\$
\$
\$
\$
\$
\$
\$
\$
\$
\$
\$
\$
\$
\$
\$
\$
\$
\$
\$
\$
\$
\$
\$
\$
\$
\$
\$
\$
\$
\$
\$
\$
\$
\$
\$
\$
\$
\$
\$
\$
\$
\$
\$
\$
\$
\$
\$
\$
\$
\$
\$
\$
\$
\$
\$
\$
\$
\$
\$
\$
\$
\$
\$
\$
\$
\$
\$
\$
\$
\$
\$
\$
\$
\$
\$
\$
\$
\$
\$
\$
\$
\$
\$
\$
\$
\$
\$
\$
\$
\$
\$
\$
\$
\$
\$
\$
\$
\$
\$
\$
\$
\$
\$
\$
\$
\$
\$
\$
\$
\$
\$
\$
\$
\$
\$
\$
\$
\$
\$
\$
\$
\$
\$
\$
\$
\$
\$
\$
\$
\$
\$
\$
\$
\$
\$
\$
\$
\$
\$
\$
\$
\$
\$
\$
\$
\$
\$
\$
\$
\$
\$
\$
\$
\$
\$
\$
\$
\$
\$
\$
\$
\$
\$
\$
\$
\$
\$
\$
\$
\$
\$
\$
\$
\$
\$
\$
\$
\$
\$
\$
\$
\$
\$
\$
\$
\$
\$
\$
\$
\$
\$
\$
\$
\$
\$
\$
\$
\$
\$
\$
\$
\$
\$
\$
\$
\$
\$
\$
\$
\$
\$
\$
\$
\$
\$
\$
\$
\$
\$
\$
\$
\$
\$
\$
\$
\$
\$
\$
\$
\$
\$
\$
\$
\$
\$
\$
\$
\$
\$
\$
\$
\$
\$
\$
\$
\$
\$
\$
\$
\$
\$
\$
\$
\$
\$
\$
\$
\$
\$
\$
\$
\$
\$
\$
\$
\$
\$
\$
\$
\$
\$
\$
\$
\$
\$
\$
\$
\$
\$
\$
\$
\$
\$
\$
\$
\$
\$
\$
\$
\$
\$
\$
\$
\$
\$
\$
\$
\$
\$
\$
\$
\$
\$
\$
\$
\$
\$
\$
\$
\$
\$
\$
\$
\$
\$
\$
\$
\$
\$
\$
\$
\$
\$
\$
\$
\$
\$
Local.aspx

```csharp
<%@ Page Language="C#" AutoEventWireup="true" CodeFile="local.aspx.cs" Inherits="local" %>

<!DOCTYPE html>
<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
    <script src="JavaScript.js"></script>
    <script src="http://code.jquery.com/jquery-1.9.1.min.js"></script>
    <script src="http://code.jquery.com/jquery-migrate-1.1.0.min.js"></script>
    <title></title>
    <style type="text/css">
        .minitable
        {
            margin: 0 auto;
        }
        .iframe
        {
            flex-align: center;
        }
        .blahblah
        {
            background-color: #323B66;
        }
        .minititle
        {
            text-align: center;
            /*color:#D69345;*/ color: #58FA58;
            font-family: Arial;
            font-weight: bold;
            font-size: 24px;
        }
        .widget
        {
            width: 100%;
            background-color: #6ED1FF;
            color: black;
            opacity: .8;
            height: 480px;
        }
    html
    {
        overflow-x: hidden;
        overflow-y: hidden;
    }
    .bottomforcer
    {
        width: 100%;
        position: absolute;
    }
```
```html
<!DOCTYPE html>
<html>
<head>
    <style>
        body { text-align: center; }
    </style>
</head>
<body class="blahblah">
<form id="form1" runat="server" class="widget">
    <script type="text/javascript">
        $(document).ready(function () {
            setInterval('jquerytest()', 600);
            setInterval('timeout()', 300000);
        });

        function jquerytest() {
            var response;
            //$('#tablechecker').click();
            $.get("databasecheck.aspx", function (response) {
                //alert(response);
                if (response == "1") {
                    //alert("useable");
                    window.location.replace("activites.aspx");
                } else if (response == "3") {
                    //alert("useable");
                    window.location.replace("dinning.aspx");
                } else if (response == "4") {
                    //alert("useable");
                    window.location.replace("weather.aspx");
                } else if (response == "0") {
                    window.location.replace("MainPage.aspx");
                }
            });
        }
    </script>

    <asp:SqlDataSource ID="SqlDataSource1" runat="server"
        ConnectionStrings="<%$ ConnectionStrings:JP_CM_SeniorDesignConnectionString %>">
        SelectCommand="SELECT * FROM [Events_Tbl]"
    </asp:SqlDataSource>

    <div>
        <table class="minitable" style="width: 100%;">
            <tr style="background-color: #0431B4">
                <td class="minititle">Local Events</td>
            </tr>
        </table>
    </div>

    <asp:SqlDataSource ID="SqlDataSource2" runat="server"
        ConnectionStrings="<%$ ConnectionStrings:JP_CM_SeniorDesignConnectionString %>">
        SelectCommand="SELECT top 2 * FROM [Events_Tbl]"
    </asp:SqlDataSource>
</form>
</body>
</html>
```
<asp:DataList ID="DataList1" runat="server"
DataSourceID="SqlDataSource1">
  <HeaderTemplate>
    <table id="weatherminitable" style="font-size: 20px; font-family: Cambria; margin-left:50px;">
      </HeaderTemplate>
  <ItemTemplate>
    <b>Event:</b>
    <asp:Label ID="ActivityLabel" runat="server" Text='<%# Eval("Event_Name") %>' />
    <br />
    <b>Date and Time:</b>
    <asp:Label ID="Date_TimeLabel" runat="server" Text='<%# Eval("Date_Time") %>' />
    <br />
    <b>Description:</b>
    <asp:Label ID="LocationLabel" runat="server" Text='<%# Eval("Description") %>' />
    <br />
    <b>Address:</b>
    <asp:Label ID="Label1" runat="server" Text='<%# Eval("Address") %>' />
    <br />
  </ItemTemplate>
  <SeparatorTemplate>
    <hr color="#0431B4" size="5" width="100">
  </SeparatorTemplate>
  <FooterTemplate>
    </table>
  </FooterTemplate>
</asp:DataList>
</table style="width: 100%;" class="bottomforcer">
<tr style="text-align: center;">
  <td runat="server"
    <asp:AdRotator ID="AdRotator2" AdvertisementFile="Ad2.xml"
    Target="_blank" />
  </td>
</tr>
<td runat="server"
    <asp:AdRotator ID="AdRotator1" AdvertisementFile="Ad1.xml"
    Target="_blank" />
</td>"
Weather.aspx

```html
<!DOCTYPE html>
<html xmlns="http://www.w3.org/1999/xhtml">
<head id="Head1" runat="server">
  <script src="http://code.jquery.com/jquery-1.9.1.min.js"></script>
  <script src="http://code.jquery.com/jquery-migrate-1.1.0.min.js"></script>
  <script src="JavaScript.js"></script>
  <title></title>
  <style type="text/css">
    .minitable
    {
      margin: 0 auto;
    }
    .iframe
    {
      flex-align: center;
    }
    .blahblah
    {
      background-color: #323B66;
    }
    .minititle
    {
      text-align: center;
      /*color:#D69345;*/ color: #58FA58;
      font-family: Arial;
      font-weight: bold;
      font-size: 24px;
    }
    .widget
    {
      width: 100%;
      background-color: #6ED1FF;
      color: black;
      opacity: .8;
      height: 480px;
    }
    html
    {
      overflow-x: hidden;
      overflow-y: hidden;
    }
    .bottomforcer
    {
      width: 100%;
      position: absolute;
    }
  </style>
</head>
<body>
<!-- Your content here --></body>
</html>
```
$(document).ready(function () {
    setInterval('jquerytest()', 600);
    //setInterval('timeout()', 90000);
});

function jquerytest() {
    var response;
    //$('#tablechecker').click();
    $.get("databasecheck.aspx", function (response) {
        //alert(response);
        if (response == "1") {
            //alert("useable");
            window.location.replace("activites.aspx");
        }
        if (response == "2") {
            //alert("useable");
            window.location.replace("local.aspx");
        }
        if (response == "3") {
            //alert("useable");
            window.location.replace("dinning.aspx");
        }
        if (response == "0") {
            window.location.replace("MainPage.aspx");
        }
    });
};

</script>

<div class="minititle" style="width: 100%; height: 100%; text-align: center;">
    <asp:SqlDataSource ID="SqlDataSource1" runat="server" ConnectionString="<%$ ConnectionStrings:JP_CM_SeniorDesignConnectionString %>">
        SelectCommand="SELECT * FROM [Weather_Conditions_Tbl] where date between dateadd(dd,-1,getdate()) and dateadd(dd,1,getdate())"
    </asp:SqlDataSource>
    <asp:SqlDataSource ID="SqlDataSource2" runat="server" ConnectionString="<%$ ConnectionStrings:JP_CM_SeniorDesignConnectionString %>">
        SelectCommand="SELECT RIGHT(Time, 8) as time, Date, justtide FROM [Tides_Tbl] where date between dateadd(dd,-1,getdate()) and dateadd(dd,0,getdate()) order by ampm, time"
    </asp:SqlDataSource>
</div>
### Weather

<table>
<thead>
<tr>
<th>Image</th>
<th>Conditions</th>
<th>Temperature</th>
<th>Humidity</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="images/localRadar.jpg" alt="Local Radar" /></td>
<td><a href="#">Eval(&quot;Weather_Conditions&quot;)</a></td>
<td><a href="#">Eval(&quot;Temp_F&quot;)</a></td>
<td><a href="#">Eval(&quot;Humidity&quot;)</a></td>
</tr>
</tbody>
</table>

### Weather Details

- **Conditions**: [Eval("Weather_Conditions")](#)
- **Temperature**: [Eval("Temp_F")](#)
- **Humidity**: [Eval("Humidity")](#)
<table style="width: 100%;" class="bottomforcer">
  <tr style="text-align: center;">
    <td>
      <asp:AdRotator ID="AdRotator2" AdvertisementFile="Ad2.xml" runat="server"
      Target="_blank" />
    </td>
    <td>
      <asp:AdRotator ID="AdRotator1" AdvertisementFile="Ad1.xml" runat="server"
      Target="_blank" />
    </td>
  </tr>
</table>
</div>
</form>
</body>
<html>
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Data;
using System.Data.Common;
using System.Security.Principal;
using System.Configuration;
using System.Data.SqlClient;
using System.Xml.Linq;
using System.Web.UI.WebControls.WebParts;
using System.Web.Configuration;

public partial class weather : System.Web.UI.Page
{
    protected void Page_Load(object sender, EventArgs e)
    {
    }
    protected void tablechecker_Clickweather(object sender, EventArgs e)
    {
        string hello;
        //4th try
        SqlConnection con = new SqlConnection("Data Source=10.63.1.116;Initial Catalog=JP_CM_SeniorDesign;User ID=comellcm; Password=qazwsx");
        con.Open();
        //SqlCommand cmd = new SqlCommand();
        SqlCommand cmd = new SqlCommand("spNav_SelectNav", con);
        cmd.Connection = con;
        cmd.CommandType = CommandType.StoredProcedure;
        SqlParameter local2 = cmd.Parameters.Add("@local", SqlDbType.Int);
        local2.Direction = ParameterDirection.ReturnValue;
        cmd.ExecuteNonQuery();
        //Response.Write("<script language=javascript>alert('" + local2.Value + "')</script>");
        hello = local2.Value.ToString();

        if (hello == "0")
        {
            Response.Redirect("MainPage.aspx");
        }
    }
}
Touch Screen Code

Maintouchpage.aspx

```csharp
@ Page Language="C#" AutoEventWireup="true" CodeFile="maintouchpage.aspx.cs" Inherits="maintouchpage" %>

<!DOCTYPE html>
<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
    <link href="StyleSheet.css" rel="stylesheet" />
    <script src="JavaScript.js"></script>
    <script src="http://code.jquery.com/jquery-1.9.1.min.js"></script>
    <script src="http://code.jquery.com/jquery-migrate-1.1.0.min.js"></script>
    <title>Touch Screen Panel</title>
</head>
<style type="text/css">
    body {
        background-color: black;
    }
</style>
<body style="cursor: none;">
<script type="text/javascript">
    function back() {
        document.getElementById("firstset").style.display = 'block';
        document.getElementById("weatherset").style.display = 'none';
        document.getElementById("Activtiesset").style.display = 'none';
        document.getElementById("Localset").style.display = 'none';
        loadXMLDocChas("navigation.aspx");
        //$.post("navigation.aspx", function (response) { });
        window.close();
    }
    function weatherset() {
        document.getElementById("firstset").style.display = 'none';
        document.getElementById("weatherset").style.display = 'block';
        //loadXMLDocChas("gotoweather.aspx");
        $.post("gotoweather.aspx", function (response) { });
    }
    function Resturantsset() {
        document.getElementById("firstset").style.display = 'none';
        //document.getElementById("Resturantsset").style.display = 'block';
        loadXMLDocChas("gotoResturants.aspx");
        //$.post("gotoResturants.aspx", function (response) { });
        window.close();
    }
    function Activtiesset() {
        document.getElementById("firstset").style.display = 'none';
        document.getElementById("Activtiesset").style.display = 'block';
        //loadXMLDocChas("gotoActivites.aspx");
        $.post("gotoActivites.aspx", function (response) { });
    }
```
function Localset() {
    document.getElementById("firstset").style.display = 'none';
    document.getElementById("Localset").style.display = 'block';
    //loadXMLDocChas("gotoLocal.aspx");
    $.post("gotoLocal.aspx", function (response) { });
}

function loadXMLDocChas(url) {
    window.open(url);
}

</script>

<form id="maintouchpage1" runat="server">
  <div>
    <table id="firstset" style="width:800px;">
      <tr style="width: 100%;">
        <td style="width:240px">
          <div onclick="weatherset();" class="link" id="weatherbutton">
            <a onclick="weatherset();">Weather</a>
          </div>
        </td>
      </tr>
      <tr style="width: 100%;">
        <td style="width:240px">
          <div onclick="Resturantsset();" class="link">
            <a onclick="Resturantsset();">Restaurants</a>
          </div>
        </td>
      </tr>
      <tr style="width: 100%;">
        <td style="width:240px">
          <div onclick="Activtiesset();" class="link">
            <a onclick="Activtiesset();">Activities</a>
          </div>
        </td>
      </tr>
      <tr style="width: 100%;">
        <td style="width:240px">
          <div onclick="Localset();" class="link">
            <a onclick="Localset();">Local Events</a>
          </div>
        </td>
      </tr>
    </table>
    <table id="weatherset" style="display: none; width:800px">
      <tr>
        <td style="width:800px">
          <div onclick="back();" class="linkback">
            <a onclick="back();">Let's go Back</a>
          </div>
        </td>
      </tr>
    </table>
  </div>
</form>
<table id="Activtiesset" style="display: none; width:800px;"
width="100%" height="100%">
<tr style="width: 800px;">
<td>
<div onclick="back();" class="linkback">
<a onclick="back();">Let's go Back</a>
</div>
</td>
</tr>
</table>

<table id="Localset" style="display: none;" style="width: 800px;">
<tr style="width: 800px;">
<td>
<div onclick="back();" class="linkback">
<a onclick="back();">Let's go Back</a>
</div>
</td>
</tr>
</table>
</div>
</form>
</body>
</html>
MainTouchPage.aspx.cs

using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
using System.Data.Common;
using System.Data;
using System.Security.Principal;
using System.Configuration;
using System.Data.SqlClient;

public partial class maintouchpage : System.Web.UI.Page
{
    protected void Page_Load(object sender, EventArgs e)
    {
        /* string str = "Data Source=10.63.1.116;Initial Catalog=JP_CM_SeniorDesign;User ID=comellcm";
        SqlConnection conn = new SqlConnection(str);
        SqlCommand cmd = new SqlCommand("spNav_GoHome",conn);
        SqlDataAdapter da = new SqlDataAdapter();
        DataSet ds = new DataSet();
        da.Update(ds, "tbl_Navigation");
        this.GridView1.DataBind();
        Close*/
    }

    protected void Page_PreInit(object sender, EventArgs e)
    {
    }

    protected void weatherbutton_Click(object sender, EventArgs e)
    {
        DataTable dtUsers = new DataTable("tbl_Navigation");

        SqlConnection connection = new SqlConnection(ConfigurationManager.ConnectionStrings["JP_CM_SeniorDesignConnectionString"].ConnectionString);
        connection.Open();
        SqlCommand cmd = new SqlCommand("UPDATE tbl_Navigation SET Weather = 0, Resturants = 0, Activites = 0, Local = 0", connection);
        SqlDataAdapter da = new SqlDataAdapter(cmd);

        //da.Fill(dtUsers);
        da.Update(dtUsers);

        connection.Close();
    }
}
CatRest.aspx

```html
<!DOCTYPE html>
<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
    <link href="StyleSheet.css" rel="stylesheet" />
    <script src="JavaScript.js"></script>
    <title></title>

    <style type="text/css">
    body
    {
        background-color: black;
    }
    </style>
</head>

<script type="text/javascript">
    function gotothenextpagefunction(x) {
        loadXMLDocChas("gotoResturantstype.aspx?passvariable=" + x);
        window.close();
    }
    function back() {
        loadXMLDocChas("navigation.aspx");
        loadXMLDocChas("clearer.aspx");
        window.close();
    }

    function loadXMLDocChas(url) {
        window.open(url);
    }
</script>
<body>

    <asp:SqlDataSource ID="SqlDataSource1" runat="server" ConnectionString="@ ConnectionStrings:JP_CM_SeniorDesignConnectionString" SelectCommand="select distinct Category_Key  FROM [Restaurants_TBL]" />
    <form id="form1" runat="server">
        <div>
            <asp:DataList ID="DataList1" runat="server" DataSourceID="SqlDataSource1">
                <HeaderTemplate>
                    <table id="Resturantsset" width="100%" height="100%">
                        <tr style="width: 100%;">
                            <td>
                                <div onclick="back();" class="linkback">
                                    <a>Let's go Back</a>
                                </div>
                            </td>
                        </tr>
                    </table>
                </HeaderTemplate>
                <ItemTemplate>
                </ItemTemplate>
            </asp:DataList>
        </div>
    </form>
</body>
</html>
```
<tr style="width: 100%;">
  <td>
    <div onclick="gotothepagefunction('javascript:
      <% # Eval("Category_Key") %");" class="linkback">
      <a href="#javascript:
        <% # Eval("Category_Key") %"></a>
    </div>
  </td>
</tr>

FooterTemplate>
</table>
</FooterTemplate>
</asp:DataList>
</div>
</form>
</body>
</html>

Clearer.aspx.cs

using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
using System.Data.Common;
using System.Data;
using System.Security.Principal;
using System.Configuration;
using System.Data.SqlClient;
public partial class clearer : System.Web.UI.Page
{
  protected void Page_Load(object sender, EventArgs e)
  {
    SqlConnection con = new SqlConnection("Data Source=10.63.1.116;Initial Catalog=JP_CM_SeniorDesign;User ID=comellcm; Password=qazwsx");
    con.Open();
    SqlCommand cmd = new SqlCommand();
    cmd.Connection = con;
    cmd.CommandType = CommandType.StoredProcedure;
    cmd.CommandText = "spNav_Rest_Clearer";
    /*
     * SqlParameter p1 = new SqlParameter("variable", variable to send);
     * 2nd
     * cmd.Parameters.Add(p1);
     */
    SqlDataReader rd = cmd.ExecuteReader();
  }
}
public partial class clearerind : System.Web.UI.Page
{
    protected void Page_Load(object sender, EventArgs e)
    {
        SqlConnection con = new SqlConnection("Data Source=10.63.1.116;Initial Catalog=JP_CM_SeniorDesign;User ID=comellcm; Password=qazwsx");
        con.Open();
        SqlCommand cmd = new SqlCommand();
        cmd.Connection = con;
        cmd.CommandType = CommandType.StoredProcedure;
        cmd.CommandText = "spNav_Rest_Clearer";
        /*
        * SqlParameter p1 = new SqlParameter("variable", varible to send);
        * 2nd
        * cmd.Parameters.Add(p1);
        */
        SqlDataReader rd = cmd.ExecuteReader();
    }
}
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
using System.Data.Common;
using System.Data;
using System.Security.Principal;
using System.Configuration;
using System.Data.SqlClient;

public partial class gotoActivites : System.Web.UI.Page
{
    /*SqlConnection cn = new SqlConnection("$
     ConnectionStrings:JP_CM_SeniorDesignConnectionString ");
        SqlCommand cmd = new SqlCommand();
        SqlDataReader dr;*/

    protected void Page_Load(object sender, EventArgs e)
    {
        SqlConnection con = new SqlConnection("Data Source=10.63.1.116;Initial Catalog=JP_CM_SeniorDesign;User ID=comellcm; Password=qazwsx");
        con.Open();
        SqlCommand cmd = new SqlCommand();
        cmd.Connection = con;
        cmd.CommandType = CommandType.StoredProcedure;
        cmd.CommandText = "spNav_GoActivites";
        /*
         * SqlParameter p1 = new SqlParameter("variable", variable to send);
         * 2nd
         * cmd.Parameters.Add(p1);
         */
        SqlDataReader rd = cmd.ExecuteReader();
    }
}
GotoLocal.aspx.cs

using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
using System.Data.Common;
using System.Data;
using System.Security.Principal;
using System.Configuration;
using System.Data.SqlClient;

public partial class gotoLocal : System.Web.UI.Page
{
    /*SqlConnection cn = new SqlConnection("$
    ConnectionStrings:JP_CM_SeniorDesignConnectionString ");
    SqlCommand cmd = new SqlCommand();
    SqlDataReader dr;*/

    protected void Page_Load(object sender, EventArgs e)
    {
        SqlConnection con = new SqlConnection("Data Source=10.63.1.116;Initial
Catalog=JP_CM_SeniorDesign;User ID=comellcm; Password=qazwsx");
        con.Open();
        SqlCommand cmd = new SqlCommand();
        cmd.Connection = con;
        cmd.CommandType = CommandType.StoredProcedure;
        cmd.CommandText = "spNav_GoLocal";
        /*
        * SqlParameter p1 = new SqlParameter("variable", varible to send);
        * 2nd
        * cmd.Parameters.Add(p1);
        */
        SqlDataReader rd = cmd.ExecuteReader();
    }
}
GotoResturants.aspx.cs

using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Data.Common;
using System.Data;
using System.Security.Principal;
using System.Configuration;
using System.Data.SqlClient;

public partial class gotoResturants : System.Web.UI.Page
{
    protected void Page_Load(object sender, EventArgs e)
    {
        string x = Request.QueryString["passvariable"];

        if (x == null)
        {
            SqlConnection con = new SqlConnection("Data Source=10.63.1.116;Initial Catalog=JP_CM_SeniorDesign;User ID=comellcm; Password=qazwsx");
            con.Open();
            SqlCommand cmd = new SqlCommand();
            cmd.Connection = con;
            cmd.CommandType = CommandType.StoredProcedure;
            cmd.CommandText = "spNav_GoResturants";
            /*
             * SqlParameter p1 = new SqlParameter("variable", varible to send);
             * 2nd
             * cmd.Parameters.Add(p1);
             */
            SqlDataReader rd = cmd.ExecuteReader();
            Response.Redirect("catrest.aspx");
        }
        else
        {
            SqlConnection conn = new SqlConnection("Data Source=10.63.1.116;Initial Catalog=JP_CM_SeniorDesign;User ID=comellcm; Password=qazwsx");

            SqlCommand com = new SqlCommand( "dbo.spNav_GoResturants_type", conn);
            com.CommandType = CommandType.StoredProcedure;
            com.Parameters.Add(new SqlParameter("@type", x));

            conn.Open();
            SqlDataAdapter da = new SqlDataAdapter(com);
            DataSet ds = new DataSet();
            da.Fill(ds);
conn.Close();
Response.Redirect("catrest.aspx");

/*
SqlConnection con = new SqlConnection("Data Source=10.63.1.116;Initial
Catalog=JP_CM_SeniorDesign;User ID=comellcm; Password=qazwsx");
con.Open();
SqlCommand cmd = new SqlCommand();
cmd.Connection = con;
cmd.CommandType = CommandType.StoredProcedure;
cmd.CommandText = "spNav_GoResturants_type";
cmd.Parameters.Add("@type", SqlDbType.VarChar).Value = x;
cmd = new SqlCommand("spNav_GoResturants_type", con);
SqlDataReader rd = cmd.ExecuteReader();
*/
gotoResturantsind.aspx.cs

using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
using System.Data.Common;
using System.Data;
using System.Security.Principal;
using System.Configuration;
using System.Data.SqlClient;

public partial class gotoResturantsind : System.Web.UI.Page
{
    protected void Page_Load(object sender, EventArgs e)
    {

        string whichrest = Request.QueryString["whichrest"];    

        SqlConnection conn = new SqlConnection("Data Source=10.63.1.116;Initial Catalog=JP_CM_SeniorDesign;User ID=comellcm; Password=qazwsx");

        SqlCommand com = new SqlCommand("dbo.spNav_GoResturants_ind", conn);
        com.CommandType = CommandType.StoredProcedure;
        //com.Parameters.AddWithValue("@type", x);
        com.Parameters.Add("@name", SqlDbType.VarChar);
        com.Parameters["@name"].Value = whichrest;

        conn.Open();
        com.ExecuteNonQuery();

        conn.Close();
    }
}
GotoRestaurantstype.aspx.cs

```csharp
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
using System.Data.Common;
using System.Data;
using System.Security.Principal;
using System.Configuration;
using System.Data.SqlClient;

public partial class gotoRestaurantstype : System.Web.UI.Page
{
    protected void Page_Load(object sender, EventArgs e)
    {

        string x = Request.QueryString["passvariable"];  

        SqlConnection conn = new SqlConnection("Data Source=10.63.1.116;Initial Catalog=JP_CM_SeniorDesign;User ID=comellcm; Password=qazwsx");

        SqlCommand com = new SqlCommand("dbo.spNav_GoResturants_type", conn);
        com.CommandType = CommandType.StoredProcedure;
        //com.Parameters.AddWithValue(new SqlParameter("@type", x));
        //com.Parameters.AddWithValue("@type", x);
        com.Parameters.Add("@type", SqlDbType.VarChar);
        com.Parameters["@type"].Value = x;

        conn.Open();
        com.ExecuteNonQuery();
        //SqlDataAdapter da = new SqlDataAdapter(com);
        //DataSet ds = new DataSet();

        conn.Close();
        Response.Redirect("singlerest.aspx?restcat=" + x);
    }
}
```
gotoweather.aspx.cs

using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
using System.Data.Common;
using System.Data;
using System.Security.Principal;
using System.Configuration;
using System.Data.SqlClient;

public partial class gotoweather : System.Web.UI.Page
{
    /*SqlConnection cn = new SqlConnection("$
    ConnectionStrings:JP_CM_SeniorDesignConnectionString ");
    SqlCommand cmd = new SqlCommand();
    SqlDataReader dr;*/

    protected void Page_Load(object sender, EventArgs e)
    {
        SqlConnection con = new SqlConnection("Data Source=10.63.1.116;Initial
Catalog=JP_CM_SeniorDesign;User ID=comellcm; Password=qazwsx");
        con.Open();
        SqlCommand cmd = new SqlCommand();
        cmd.Connection = con;
        cmd.CommandType = CommandType.StoredProcedure;
        cmd.CommandText = "spNav_GoWeather";
        /*
         * SqlParameter p1 = new SqlParameter("variable", varible to send);
         * 2nd
         * cmd.Parameters.Add(p1);
         */
        SqlDataReader rd = cmd.ExecuteReader();
    }
}
Navigation.aspx.cs

using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
using System.Data.Common;
using System.Data;
using System.Security.Principal;
using System.Configuration;
using System.Data.SqlClient;

public partial class navigation : System.Web.UI.Page
{
    /*SqlConnection cn = new SqlConnection("$
    ConnectionStrings:JP_CM_SeniorDesignConnectionString ");
    SqlCommand cmd = new SqlCommand();
    cmd.Connection = cn;
    cmd.CommandType = CommandType.StoredProcedure;
    cmd.CommandText = "spNav_GoHome";
    /*
    * SqlParameter p1 = new SqlParameter("variable", varible to send);
    * 2nd
    * cmd.Parameters.Add(p1);
    */
    SqlDataReader dr = cmd.ExecuteReader();

    Response.Redirect("maintouchpage.aspx");
}
}
SingleRest.aspx

```html
<!DOCTYPE html>
<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
    <link href="StyleSheet.css" rel="stylesheet" />
    <script src="JavaScript.js"></script>
    <script src="http://code.jquery.com/jquery-1.9.1.min.js"></script>
    <script src="http://code.jquery.com/jquery-migrate-1.1.0.min.js"></script>
    <title></title>

    <style type="text/css">
        body {
            background-color: black;
        }
    </style>
</head>

<body type="text/javascript">

    <script type="text/javascript">
        function gotothenextpagefunction(x) {
            loadXMLDocChas("gotoResturantsind.aspx?whichrest=" + x);
            //$.post("gotoResturantsind.aspx?whichrest=" + x, function (response) {
        });
    }
    function back() {
        loadXMLDocChas("clearerind.aspx");
        //$.post("clearerind.aspx", function (response) { });
        //loadXMLDocChas("navigation.aspx");
        window.location.replace("catrest.aspx");
        //window.close();
    }
    function loadXMLDocChas(url) {
        window.open(url);
    }
    function etphonehome() {
        loadXMLDocChas("clearer.aspx");
        loadXMLDocChas("navigation.aspx");

        window.location.replace("maintouchpage.aspx");
        //window.close();
    }
</script>

    <form id="form1" runat="server">
        <asp:SqlDataSource ID="SqlDataSource1" runat="server" ConnectionString=""%>
        <div>
            <asp:DataList ID="DataList1" runat="server" DatasourceID="SqlDataSource1">
                <HeaderTemplate>
```
<table id="ResturantInd" style="width: 100%;">
  <tr>
    <td>
      <div onclick="etphonehome();" class="linkback">
        <a>Home</a>
      </div>
    </td>
  </tr>
  <tr>
    <td>
      <div onclick="back();" class="linkback">
        <a>Let's go Back</a>
      </div>
    </td>
  </tr>
</table>

ngleRest.aspx.cs
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;

public partial class singlerest : System.Web.UI.Page
{
    protected void Page_Load(object sender, EventArgs e)
    {
        string restcat = Request.QueryString["restcat"];

        SqlDataSource1.SelectCommand = "SELECT * FROM [Restaurants_Tbl] where category_key = '" + restcat + "';";
    }
}
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Data.SqlClient;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;
using System.IO;
using System.Xml.Linq;
using System.Xml;
using System.Net;

namespace WeatherUndergroundDataGrabber
{
    public partial class Form1 : Form
    {
        public string _UserName;

        public SqlConnection GNPtestConnection = new SqlConnection();
        public SqlCommand GNPtestCommand = new SqlCommand();

        public Form1()
        {
            InitializeComponent();
        }

        private void grabDataBtn_Click(object sender, EventArgs e)
        {
            ParseResponseCondXML();
            ParseResponseTideXML();
            SaveCurrentForecastImage();
        }

        private void ParseResponseTideXML()
        {
            const string highTideStr = "High Tide";
            const string lowTideStr = "Low Tide";

            Cursor.Current = Cursors.WaitCursor;

            var xml = "http://api.wunderground.com/api/be95499c5b98d7d6/tide/q/FL/Daytona.xml";
            var doc = new XmlDocument();
            doc.Load(xml);

            //start by drilling down to the "observation" level, this our tide data
            XmlNodeList currentLocation =
                doc.SelectNodes("/response/tide/tideSummary");
var nodes =
doc.SelectNodes("/response/tide/tideSummary/observation/data/type");

string[] observation = new string[nodes.Count];

//list to hold node numbers related to high tide or low tide
List<int> highNode = new List<int>();
List<int> lowNode = new List<int>();

//grabs our node numbers that contain high tide or low tide
for (int i = 0; i < nodes.Count; i++)
{
    //observation[i] = nodes[i];

    if (nodes[i].InnerText == "High Tide")
        highNode.Add(i);

    if (nodes[i].InnerText == "Low Tide")
        lowNode.Add(i);
}

nodes =
doc.SelectNodes("/response/tide/tideSummary/observation/date/pretty");

List<string> highTideTimes = new List<string>();
List<string> lowTideTimes = new List<string>();

for (int j = 0; j < highNode.Count; j++)
{
    int highNodeNum = highNode[j];

    highTideTimes.Add(nodes[highNodeNum].InnerText + " " + highTideStr);
}

for (int k = 0; k < lowNode.Count; k++)
{
    int lowNodeNum = lowNode[k];

    lowTideTimes.Add(nodes[lowNodeNum].InnerText + " " + lowTideStr);
}

var mergeTides = highTideTimes.Union(lowTideTimes).ToList();

//mergeTides.Sort();

//need to sort according to the date... damn...

******************SQL CONNECTION
SetConnectionString();

string counter;
counter = "1";

//open up a connection
using (SqlConnection conn = GNPtestConnection)
{ 
    foreach (var tide in mergeTides) 
    {
        if (counter == "1")
        {
            //delete tides
            SqlCommand com2 = new SqlCommand("dbo.spDelete_Tides", conn);
            com2.CommandType = CommandType.StoredProcedure;
            //open connection and execute it
            conn.Open();
            SqlDataAdapter da2 = new SqlDataAdapter(com2);
            DataSet ds2 = new DataSet();
            da2.Fill(ds2);
            conn.Close();
            //end delete tides
            counter = "2";
        }

        SqlCommand com = new SqlCommand("dbo.SPData_Tides", conn);
        com.CommandType = CommandType.StoredProcedure;
        //pass in the variable for the SP
        com.Parameters.Add(new SqlParameter("@tides", tide));
        //open connection and execute it
        conn.Open();
        SqlDataAdapter da = new SqlDataAdapter(com);
        DataSet ds = new DataSet();
        da.Fill(ds);
        conn.Close();
    }
}

private void ParseResponseCondXML()
{
    Cursor.Current = Cursors.WaitCursor;
    Conditions cond = new Conditions();

    //makes API call for XML data
    var xml = "http://api.wunderground.com/api/be95499c5b98d7d6/conditions/q/32169.xml";
    var doc = XDocument.Load(xml);

    //Pull out our current location (hard coded currently
    //stores to our object
    var currentLocation = doc.Element("response").Element("current_observation").Element("display_location")
        ;
    cond._City = (string)currentLocation.Element("full");

    //Pull information we desire for our current location
    //stores to our object
var currentObservation =
doc<Element("response").Element("current_observation");
cond._Temp = (string)currentObservation.Element("temp_f");
cond._Humidity = (string)currentObservation.Element("relative_humidity");
cond._Conditions = (string)currentObservation.Element("weather");
MessageBox.Show(cond._City + "\n" + cond._Conditions + "\n" +
cond._Temp + "\n" + cond._Humidity);

/****************** SQL CONNECTION 
SetConnectionString();
//open up a connection
using (SqlConnection conn = GNPtestConnection)
{
(SqlCommand com = new SqlCommand("dbo.SPData_Conditions", conn);
com.CommandType = CommandType.StoredProcedure;
//pass in the variable for the SP
com.Parameters.Add(new SqlParameter("@city", cond._City));
com.Parameters.Add(new SqlParameter("@conditions",
cond._Conditions));
com.Parameters.Add(new SqlParameter("@temp", cond._Temp));
com.Parameters.Add(new SqlParameter("@humidity", cond._Humidity));

//open connection and execute it
conn.Open();
SqlDataAdapter da = new SqlDataAdapter(com);
DataSet ds = new DataSet();
da.Fill(ds);
conn.Close();
}

Cursor.Current = Cursors.Default;
}

private void SaveCurrentForecastImage()
{
// function to query Yahoo finance for matching symbol data
string URL;
string saveLocation = @"C:\Users\admin\Google Drive\SeniorDesign\SeniorDesignpage\imageslocalRadar.jpg"; //Jake DESKTOP String
//string saveLocation = @"Z:\Google Drive\SeniorDesign\SeniorDesignpage\imageslocalRadar.jpg"; //Jake LAPTOP
//string saveLocation = @"C:\Users\comella002\Google Drive\SeniorDesignpage\imageslocalRadar.jpg"; //chas laptop
//string saveLocation = @"C:\Users\chas\Google Drive\SeniorDesignpage\imageslocalRadar.jpg"; //chas Server

// show wait cursor
Cursor.Current = Cursors.WaitCursor;

URL = "http://api.wunderground.com/api/be95499c5b98d7d6/radar/q/32169.gif?width=280&height=280&newmaps=1"; // build the request

WebClient webClient = new WebClient();
webClient.DownloadFile(URL, saveLocation);

    // show default cursor
    Cursor.Current = Cursors.Default;
    }

public void SetConnectionString()
{
    // set the connection string based on Config groupbox data
    // unbind the dataGridView from everything (blanks the table) since
    // dataGridView1.DataSource = null;
    _UserName = textBoxUserID.Text;

    GNPtestConnection.ConnectionString =
        "Data Source=" + comboBoxDataSource.Text + ";"
        +
        "Initial Catalog=" + textBoxInitialCatalog.Text + ";"
        +
        "User ID=" + textBoxUserID.Text;

    // show connection string without password
    textBoxConnectionString.Text =
        GNPtestConnection.ConnectionString;

    // add the password to the actual Connection String
    GNPtestConnection.ConnectionString +=
        ";Password=" + textBoxPassword.Text;

    // toolStripStatusLabel1.Text = "Connection Updated."
    // debug step; uncomment next line to verify connection string
    // listBoxStatus.Text = "Connection String: " +
    // GNPtestConnection.ConnectionString);
    }

}

public class Conditions
{
    private string conditions;
    private string humidity;
    private string temp;
    private string city;

    public string _Conditions
    {
        get { return conditions; }        // public property available to
    calling form
        set { conditions = value; }
    }

    public string _Humidity
    {
        get { return humidity; }        // public property available to
    calling form
        set { humidity = value; }
    }
}
public string _Temp
{
    get { return temp; }  // public property available to calling
    set { temp = value; }
}

public string _City
{
    get { return city; }  // public property available to calling
    set { city = value; }
}
Microsoft SQL Server Stored Procedures

SpData_conditions

ALTER PROCEDURE [dbo].[SpData_conditions]
   -- Add the parameters for the stored procedure here
   @city varchar(50),
   @conditions varchar(max),
   @temp nchar(50),
   @humidity nchar(50)
AS
BEGIN
   -- SET NOCOUNT ON added to prevent extra result sets from
   -- interfering with SELECT statements.
   SET NOCOUNT ON;

   delete from Weather_Conditions_Tbl

   insert into Weather_Conditions_Tbl (Full_City, Weather_Conditions, Temp_F, Humidity, date)
   values
   (@city, @conditions, @temp, @humidity, getdate())
END

SPData_Tides

USE [JP_CM_SeniorDesign]
GO
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
-- ===============================================================================
-- Author:  Chas
-- Create date: 2/22/2013
-- Description:
-- ===============================================================================
ALTER PROCEDURE [dbo].[SPData_Tides]
   -- Add the parameters for the stored procedure here
   @tides varchar(100)
AS
BEGIN

Declare
   --@tides varchar(50)

@hour varchar(2)
,@min varchar(2)
,@ampm varchar(2)
,@date varchar(50)
,@finaldate varchar(50)
Set @tides = '2:06 PM EST on February 18, 2013 High Tide'

Set @temptide = right(@tides, 6)
if (substring(@temptide, 1, 1) = 'W')
Begin
  set @finaltide = right(@tides, 8)
  set @date = right(@tides, 17)
End
else
Begin
  set @finaltide = right(@tides, 9)
  set @date = right(@tides, 18)
End
/*finish date*/
set @finaldate = substring(@date, 1, 8)

/*beginning substrings*/
if (substring(@tides, 2, 1) = ':')
begin
  set @hour = substring(@tides, 1, 1)
  set @min = substring(@tides, 3, 2)
  set @ampm = substring(@tides, 6, 2)
  set @monthstart = substring(@tides, 16, 3)
end
else
begin
  set @hour = substring(@tides, 1, 2)
  set @min = substring(@tides, 4, 2)
  set @ampm = substring(@tides, 7, 2)
  set @monthstart = substring(@tides, 17, 3)
end

if (@monthstart = 'Jan')
set @month = 'January'
if (@monthstart = 'Feb')
set @month = 'February'
if (@monthstart = 'Mar')
set @month = 'March'
if (@monthstart = 'Apr')
set @month = 'April'
if (@monthstart = 'Jun')
set @month = 'June'
if (@monthstart = 'Jul')
set @month = 'July'
if (@monthstart = 'Aug')
set @month = 'August'
if (@monthstart = 'Sep')
set @month = 'September'
if(@monthstart = 'Oct')
    set @month = 'October'
if(@monthstart = 'Nov')
    set @month = 'November'
if(@monthstart = 'Dec')
    set @month = 'December'

-- SET NOCOUNT ON added to prevent extra result sets from
-- interfering with SELECT statements.
SET NOCOUNT ON;

-- Insert statements for procedure here
-- Insert into Tides_Tbl (Tide_Conditions) values (@tides)
insert into tides_tbl (Tide_Conditions, time, date, justtide, ampm) values
(@tides, @hour + ':@min+ ' + @ampm, @month + ' ' + @finaldate, @finaltide, @ampm)
END

SpDelete_Tides

ALTER PROCEDURE [dbo].[spDelete_Tides]
AS
BEGIN
    delete from tides_tbl
END

SpNav_GoActivites

ALTER PROCEDURE [dbo].[spNav_GoActivites]
    -- Add the parameters for the stored procedure here
AS
BEGIN
    -- SET NOCOUNT ON added to prevent extra result sets from
    -- interfering with SELECT statements.
    SET NOCOUNT ON;

    UPDATE tbl_Navigation SET Weather = '0', Resturants = '0', Activites = '1', Local = '0'
END

SPNav_GoHome

ALTER PROCEDURE [dbo].[spNav_GoHome]
    -- Add the parameters for the stored procedure here
AS
BEGIN
    -- SET NOCOUNT ON added to prevent extra result sets from
    -- interfering with SELECT statements.
    SET NOCOUNT ON;

    UPDATE tbl_Navigation SET Weather = '0', Resturants = '0', Activites = '0', Local = '0'
END
SPNav_GoLocal

ALTER PROCEDURE [dbo].[spNav_GoLocal]
    -- Add the parameters for the stored procedure here
AS
BEGIN
    -- SET NOCOUNT ON added to prevent extra result sets from
    -- interfering with SELECT statements.
    SET NOCOUNT ON;
    UPDATE tbl_Navigation SET Weather = 0, Resturants = 0, Activites = 0, Local = 2
END

SPNav_GoResturants

ALTER PROCEDURE [dbo].[spNav_GoResturants]
    -- Add the parameters for the stored procedure here
AS
BEGIN
    -- SET NOCOUNT ON added to prevent extra result sets from
    -- interfering with SELECT statements.
    SET NOCOUNT ON;
    UPDATE tbl_Navigation SET Weather = 0, Resturants = 3, Activites = 0, Local = 0
END

SpNav_GoResturants_ind

ALTER PROCEDURE [dbo].[spNav_GoResturants_ind]
    -- Add the parameters for the stored procedure here
    @name varchar(50)
AS
BEGIN
    -- SET NOCOUNT ON added to prevent extra result sets from
    -- interfering with SELECT statements.
    SET NOCOUNT ON;
    UPDATE tbl_Navigation_rest_ind SET name = @name, [go_back] = 0
END

SPNav_GoResturants_type

ALTER PROCEDURE [dbo].[spNav_GoResturants_type]
    -- Add the parameters for the stored procedure here
    @type varchar(50)
AS
BEGIN
    -- SET NOCOUNT ON added to prevent extra result sets from
    -- interfering with SELECT statements.
    SET NOCOUNT ON;
    UPDATE tbl_Navigation_rest_types SET type = @type, [go_back] = 0
END
SPNav_GoWeather

ALTER PROCEDURE [dbo].[spNav_GoWeather]
-- Add the parameters for the stored procedure here
AS
BEGIN
-- SET NOCOUNT ON added to prevent extra result sets from
-- interfering with SELECT statements.
SET NOCOUNT ON;

UPDATE tbl_Navigation SET Weather = 4, Resturants = 0, Activites = 0, Local = 0
END

SpNav_Rest_Cleaner

ALTER PROCEDURE [dbo].[spNav_Rest_Clearer]
-- Add the parameters for the stored procedure here
AS
BEGIN
-- SET NOCOUNT ON added to prevent extra result sets from
-- interfering with SELECT statements.
SET NOCOUNT ON;

-- Insert statements for procedure here
update [JP_CM_SeniorDesign].[dbo].[tbl_Navigation_rest_types]
set type = 'none'

update [JP_CM_SeniorDesign].[dbo].[tbl_Navigation_rest_ind]
set name = '0'
END

SpNav_RestClearer_justind

ALTER PROCEDURE [dbo].[spNav_Rest_Clearer_justind]
-- Add the parameters for the stored procedure here
AS
BEGIN
-- SET NOCOUNT ON added to prevent extra result sets from
-- interfering with SELECT statements.
SET NOCOUNT ON;

-- Insert statements for procedure here
update [JP_CM_SeniorDesign].[dbo].[tbl_Navigation_rest_ind]
set name = '0'
END
### SpNav_SSelectNav

ALTER PROCEDURE [dbo].[spNav_SelectNav]
    -- Add the parameters for the stored procedure here
    -- SET NOCOUNT ON added to prevent extra result sets from
    -- interfering with SELECT statements.
    SET NOCOUNT ON;

    Declare @local varchar(50),
    @weather varchar(50),
    @resturants varchar(50),
    @activites varchar(50)

    select @local = local from [JP_CM_SeniorDesign].[dbo].[tbl_Navigation]
    select @weather = weather from [JP_CM_SeniorDesign].[dbo].[tbl_Navigation]
    select @resturants = resturants from [JP_CM_SeniorDesign].[dbo].[tbl_Navigation]
    select @activites = activites from [JP_CM_SeniorDesign].[dbo].[tbl_Navigation]

    if (@local <> '0')
        return @local
    if (@weather <> '0')
        return @weather
    if (@resturants <> '0')
        return @resturants
    if (@activites <> '0')
        return @activites
    if (@local <> '0' and @weather <> '0' and @resturants <> '0' and @activites <> '0')
        return 0

    END
SPNav_SelectNav_ind

ALTER PROCEDURE [dbo].[spNav_SelectNav_ind]
    -- Add the parameters for the stored procedure here
    @name varchar(50) output
AS
BEGIN
    -- SET NOCOUNT ON added to prevent extra result sets from
    -- interfering with SELECT statements.
    SET NOCOUNT ON;

    Declare
    --@type varchar(50),
    @go_back varchar(50)

    select @go_back = go_back,
    @name = name
    from [JP_CM_SeniorDesign].[dbo].[tbl_Navigation_rest_ind]
END

SpNav_SelectNav_rest

ALTER PROCEDURE [dbo].[spNav_SelectNav_rest]
    -- Add the parameters for the stored procedure here
    @type varchar(50) output
AS
BEGIN
    -- SET NOCOUNT ON added to prevent extra result sets from
    -- interfering with SELECT statements.
    SET NOCOUNT ON;

    Declare
    --@type varchar(50),
    @go_back varchar(50)

    select @go_back = go_back,
    @type = type
    from [JP_CM_SeniorDesign].[dbo].[tbl_Navigation_rest_types]

    --if (@go_back <> '0')
    --return 1

    --if (@type <> '')
    --return @type

    --if (@go_back = '0' and @type = '')
    --return 0
END