CAGIS HelpDesk System
For the City of Cincinnati’s
Cincinnati Area Geographical Information System (CAGIS)

By

Sean K. Winfield

Submitted to
the Faculty of the Information Engineering Technology Program
in Partial Fulfillment of the Requirements for
the Degree of Bachelor of Science
in Information Engineering Technology

University of Cincinnati
College of Applied Science

March 2006
Acknowledgements

I would like to thank GOD for giving me the ability to use my mind in a productive, creative and useful capacity. I thank my fiancée, Maya for supporting me all the way while enduring a pregnancy. I thank my children, Ja’nae, Chanee, Briana and Mylan for their support by giving me time to focus on this project. Thanks to Robert Schlemmer and the IT Faculty for giving me the skills to create this application, making sure I met all deadlines, while being as flexible as possible. Thanks to the CAGIS Staff: Jay Erndt, Barbara Quinn, Michael Sullivan, Brian Scholl, Michael Love, Chad Beringhaus and Debbie Shaver for providing me constant feedback, giving me creative ideas and continuous testing throughout my project.

Special thanks to Weifen Luo of www.sourceforge.net for Dock Suite user control.
# TABLE OF CONTENTS

**ACKNOWLEDGEMENTS** i  
**TABLE OF CONTENTS** ii  
**LIST OF FIGURES** iii  
**ABSTRACT** iv  

1. **STATEMENT OF THE PROBLEM** 4  
2. **DESCRIPTION OF THE SOLUTION** 4  
3. **DELIVERABLES** 5  
4. **USER PROFILES** 5  
5. **DESIGN PROTOCOL** 8  

6. **PROOF OF DESIGN** 9  
   6.1 **DELIVERABLE #1 – UTILIZE ACTIVE DIRECTORY** 9  
   6.2 **DELIVERABLE #2 – ADMINISTRATOR’S INTERFACE: VIEW AND MODIFY DATA** 12  
   6.2.1 **View Tickets** 14  
   6.2.2 **View / Modify User Information** 15  
   6.2.3 **View / Modify Computer Information** 16  
   6.2.4 **View / Modify Department Information** 17  
   6.3 **DELIVERABLE #3 – INTEGRATE WITH MICROSOFT OUTLOOK** 18  
   6.4 **DELIVERABLE #4 – HELPDESK REPORTS** 19  
   6.5 **DELIVERABLE #5 – DESIGN A FLEXIBLE GUI** 21  

7. **CONCLUSIONS** 25  
8. **RECOMMENDATIONS** 25  

9. **DESIGN AND DEVELOPMENT** 26  
   9.1 **TIMELINE** 26  
   9.2 **BUDGET** 27  
   9.3 **HARDWARE REQUIREMENTS** 27  
   9.4 **SOFTWARE REQUIREMENTS** 27  

10. **SAMPLE CODE** 29  
   10.1 **PRESENTATION LAYER** 29  
   10.1.1 **Formatting Datagrid for Ticket Form** 29  
   10.2 **BUSINESS LOGIC LAYER** 31  
   10.2.1 **Ticket class from CH_DLL, the Dynamic Link Library file for the application** 31  

**REFERENCES** v
LIST OF FIGURES

Figure 1: Use Cases for User Profiles 7
Figure 2: Logging into the system 10
Figure 3: Group Listing for CAGIS Helpdesk System in Active Directory 11
Figure 4: TOAD Interface 12
Figure 5: Administrator’s Interface 13
Figure 6: View Tickets 14
Figure 7: View Users 15
Figure 8: View Modify Computers 16
Figure 9: View Departments 17
Figure 10: E-mail Created via Microsoft Outlook after Ticket Creation 18
Figure 11: View Tickets by User 19
Figure 12: Tickets by Department 20
Figure 13: My Tickets 21
Figure 14: Tabbed Ticket Form and Report Form with My Tickets Docked at Bottom 22
Figure 15: 3 Tabbed Forms and 2 Ticket Detail Forms Docked Right 23
Figure 16: My Tickets Docked Left and 2 Tabbed Forms 23
Figure 17: Ticket Form Floating and My Tickets Hidden 24

LIST OF TABLES

Table 1: User Profiles 6
Table 2: Budget 27
ABSTRACT

The Cincinnati Area Geographical Information System (CAGIS) utilized only three administrators to provide support to over 2300 users of their geographical information system and supporting applications. CAGIS had no way of tracking support calls completed by CAGIS administrators.

The CAGIS Helpdesk Application allows CAGIS Administrators to quickly enter the details of a support call and use information within the system to quickly resolve similar support calls. If users need immediate support, this application provides access to a remote connection tool in which an administrator can immediately connect to a user’s computer, resolve the issue and document the issue in the system. In addition, the CAGIS Helpdesk Application provides e-mail and reporting functionality. This gives its users a way to communicate and track the status of a problem or generate reports to determine whether problems are numerous in a specific application or area.

Overall, the CAGIS Helpdesk Application is an effective tool for tracking the issues and resolutions to support calls. It provides an efficient way for its users to quickly and easily resolve CAGIS users’ issues.
CAGIS HelpDesk System
For the City of Cincinnati’s
Cincinnati Area Geographical Information System (CAGIS)

1. Statement of the Problem

The Cincinnati Area Geographical Information System (CAGIS) was started in 1986 as a City/County “consortium created to share the cost of an automated mapping project.”(4) “By 1993, CAGIS had grown to 25 users, in which only half of the population was connected via network lines.”(6) There were no plans for maintaining GIS resources and no recorded benefits. “At one point, many top-level managers began doubting the potential of GIS to deliver any benefits. “(4)

Within a 7-year period, CAGIS was restructured and grew into a system that was used enterprise-wide. By the year 2000, CAGIS applications were installed on over 2000 user’s computers. (2) In addition, CAGIS technology was being used in virtually all of the City of Cincinnati and Hamilton County agencies, 7 townships and several businesses within Cincinnati. (4) At the time, CAGIS had only one administrator to provide technical support to its user community. (2) In addition, it had no helpdesk application to provide a structured means of providing support.

2. Description of the Solution

Design an issues and resolution tracking system to provide the following functionality:

- display ticket information based on current user’s needs;
- utilize Active Directory to authenticate users;
- display tickets for user currently logged into the application;
- view/modify application data;
- integrate with Microsoft Outlook to provide a generic e-mail template upon ticket creation which contains ticket information and allows the user to add additional recipients;
- generate reports based on various information.
3. Deliverables

In order to effectively define the deliverables for this project, the specific goals for this project are as follows:

- Create an issue and resolution tracking system
- Design an application to utilize Active Directory to authenticate users
- Design a module within the application to view and modify application data
- Integrate the application with Microsoft Outlook to provide a generic e-mail template upon ticket creation which contains ticket information and allows the user to add additional recipients
- Develop at least two reports using Crystal Reports in Visual Basic .NET to be displayed by CAGIS Managers and Secretary
- Design a flexible graphical user interface (GUI) that will allow users to display ticket information for the currently logged on user and save forms layout upon exiting the application.

4. User Profiles

There are four types of users for this application: CAGIS Managers, CAGIS Administrators, CAGIS Specialists, and a secretary. Table 1 outlines the differences between these groups.
<table>
<thead>
<tr>
<th>CAGIS User Group</th>
<th>Area of Expertise</th>
<th>IT Literacy Level</th>
</tr>
</thead>
<tbody>
<tr>
<td>Administrators</td>
<td>Systems Administration, Technical Support and Networking</td>
<td>Expert</td>
</tr>
<tr>
<td>Specialists</td>
<td>Geographical Information Systems (GIS), Permitting, Application Development, Oracle Database Administration, Unix Administration</td>
<td>Intermediate to Expert, depending on application</td>
</tr>
<tr>
<td>Managers</td>
<td>IT Support, Unix Administration, IT Management, Budgeting, Organizational Leadership</td>
<td>Intermediate to Expert</td>
</tr>
<tr>
<td>Secretary</td>
<td>Clerical tasks</td>
<td>Intermediate</td>
</tr>
</tbody>
</table>

**Table 1: User Profiles**

The CAGIS Administrators are the primary users of the system. They will use the system all day in order to enter support calls and perform software installations.

The CAGIS Specialists will use the system as “second level support” to the CAGIS Administrators, as necessary. They will take the support calls for which the CAGIS administrators cannot provide support. If a CAGIS Administrator determines that he cannot provide a solution to a particular problem, he will forward the request to the CAGIS Specialist that is best suited to provide a solution for the request. The CAGIS Specialist will be responsible for entering the solution to the problem in the system and closing the ticket.

The CAGIS Managers and Secretary will use the system from time to time, either on a monthly or bi-monthly basis to track the number of support calls and type of calls by application.

The Figure 1 lists the use cases for each user.
Figure 1: Use Cases for User Profiles
5. **Design Protocol**

The CAGIS Helpdesk System is multi-user application which was designed using Visual Basic .NET and a C# custom user control written by Weifen Luo from [www.sourceforge.net](http://www.sourceforge.net). The application invokes many user utilities such as:

- Radmin: a remote connection tool to simplify and consolidate CAGIS administrator’s daily tasks
- Windows command prompt, a utility used to execute shell commands
- Citrix Management Console, a utility used to manage users within Citrix Sessions
- Active Directory Users Computers, a utility used to manage users within an Active Directory domain

The backend database uses Oracle 9i to permanently store its data. The data is referenced using Oracle packages and stored procedures. The data can be used to view, create similar tickets or view reports.

Forms within the application are instantiated within a combination of a Multiple Document Interface (MDI) and a Tabbed Document Interface (TDI). This format was selected because of the following reasons:

- Many child windows do not fill up the OS task management interface, as they are hierarchically organized. Users simply switch applications.
- With MDI (and also TDI), a single menu bar and/or toolbar is shared between all child windows, reducing clutter and increasing efficient use of screen space.
- All child windows for an application can be hidden/shown/minimized/maximized as a whole.
- Without an MDI frame window, floating toolbars from one application can clutter the workspace of other applications, potentially confusing users with the jumble of interfaces.
- Features such as "Tile" and "Cascade" can be implemented for the child windows.
There are three components to this application: the database, the remote administration interface and the user interface. The user interface is the link to the database. The remote administration interface is the link to the user’s computer.

6. Proof of Design

This section explains in detail the design of the system and how the following project deliverables were met.

Project Deliverables

- Create an issue and resolution tracking system which will:
  - utilize Active Directory to authenticate users;
  - contain a module within the application to view and modify application data;
  - integrate with Microsoft Outlook to provide a generic e-mail template upon ticket creation which contains ticket information and allows the user to add additional recipients;
  - contain two reports using Crystal Reports in Visual Basic .NET to be displayed by CAGIS Managers and Secretary;
  - have a flexible graphical user interface (GUI) that will allow users to display ticket information for the currently logged on user and save forms layout upon exiting the application.

6.1 Deliverable #1 – Utilize Active Directory

When users log into the application, they are presented with a login window as displayed in Figure 2.
Users must enter a valid username, password and domain in order to be successfully authenticated. After the user enters this information and clicks the “Login” button, the application will contact active directory, authenticate the user’s username and password and determine if the user is in one of the following groups:

- **CHAdmin** – represents the group for the administrators of the application. These users have the ability to perform any function within the application.
- **CHSpecialist** – represents the group for the CAGIS Specialists. Members in this group can only view tickets, close tickets and generate reports.
- **CHMgrorSec** – represents the group for the CAGIS Secretary and CAGIS Managers. Users within this group can only generate reports.
Figure 3 displays an image of the groups using the Active Directory Users and Groups utility.

Figure 3: Group Listing for CAGIS Helpdesk System in Active Directory
6.2 Deliverable #2 – Administrator’s Interface: View and Modify Data

The Administrator’s Interface allows administrators to modify any information in the system. Previously, the administrator needed to login to an application such as TOAD (Figure 4), a database application used to connect to oracle databases and modify data.

![TOAD Interface](image)

**Figure 4: TOAD Interface**

The Administrator’s Interface eliminates this tedious step and provides a simple, easy way for CAGIS Administrators to modify application data. This interface as displayed in Figure 5, provides the user with four options:

- **View Tickets** – Provides the user a way to easily view tickets
- **View / Modify Computers** – Allows the user to modify computer names and IP addresses
- **View / Modify Users** – Allows the user to view and modify a user’s first and last name, phone number or department
- **View / Modify Departments** – Allows the user to view and add department names and descriptions

The information in the textboxes below each datagrid are populated once a user selects an item within the datagrid. Items listed in the datagrid are displayed by calling an Oracle stored procedure.

**Figure 5:** Administrator’s Interface
6.2.1 View Tickets

The View Tickets form, displayed in Figure 6, simply displays the tickets in the application. Tickets can only be created by accessing the Add Tickets form through the Main form.

Figure 6: View Tickets
6.2.2 View / Modify User Information

The View / Modify Users form, displayed in Figure 7, allows an administrator to change most of a user’s information, which includes the user’s first name, last name, phone number and department. The user’s username cannot be modified because it is linked within the database to ticket information.

![Figure 7: View Users](image)
6.2.3 View / Modify Computer Information

The View / Modify Computers form, displayed in Figure 8, allows an administrator to change a computer’s name or IP address. This is important because the application invokes a third-party remote connection application, Radmin, to connect to a user’s computer with this information.

![Figure 8: View Modify Computers](image-url)
6.2.4 View / Modify Department Information

The View / Modify Department form, displayed in Figure 9, allows administrators to add or update the name or description of a department.

Figure 9: View Departments
6.3 Deliverable #3 – Integrate with Microsoft Outlook

This application gives administrators the option of sending an e-mail notification to another administrator, the user for whom the call was taken, or any other interested party for the ticket. If the option is selected after ticket creation, Microsoft Outlook will automatically open with information regarding the ticket, and automatically input the user’s name to whom the ticket was assigned in the “To” field as displayed in Figure 10.

![Figure 10: E-mail Created via Microsoft Outlook after Ticket Creation](image-url)
6.4 Deliverable #4 – Helpdesk Reports

This application uses the Crystal Reports for Visual Basic.Net control to create two reports:

- **View Tickets by User** – This report, displayed in Figure 11 will display all tickets open and closed by a CAGIS Administrator. The information is grouped by ticket status.

![CAGIS HelpDesk Report](image)

**CAGIS HelpDesk Report**

*Tickets By Username*

bscholl

<table>
<thead>
<tr>
<th>TICKET STATUS: CLOSED</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ticket #</td>
</tr>
<tr>
<td>----------</td>
</tr>
<tr>
<td>142</td>
</tr>
<tr>
<td>166</td>
</tr>
</tbody>
</table>

**Figure 11:** View Tickets by User
• **View Tickets by Department** – This report, displayed in Figure 12 lists the total number of tickets opened and closed for a department. This report is useful because it helps CAGIS Managers determine if a department has a large number of support calls. If so, the Manager can determine if the number is high because the department is large or because the department needs additional training.

![Figure 12: Tickets by Department](image)

### CAGIS HelpDesk Report

**Tickets By Department**

- **Anderson Twp**
  - Total Tickets for Department: 99

- **Auditor**
  - Total Tickets for Department: 52

- **BlueAsh**
  - Total Tickets for Department: 18

- **CAGIS**
  - Total Tickets for Department: 257

- **Cinergy**
  - Total Tickets for Department: 5

- **CinPub Schools**
  - Total Tickets for Department: 3

- **City Buildings**
  - Total Tickets for Department: 262

- **City Health**
  - Total Tickets for Department: 264

- **City Manager**
  - Total Tickets for Department: 9
6.5 Deliverable #5 – Design a Flexible GUI

Designing a flexible GUI for the application was quite a challenge. I had to decide how I could display the current user’s tickets and more importantly, display information without making the forms look cluttered.

To solve this problem, I created a “My Tickets” form, displayed in Figure 13, which displays all the information for the current user. The user only views this information, therefore it can only be refreshed using the “Refresh” button on the form.

![Figure 13: My Tickets](image.png)

To solve the problem of managing the information on the user’s screen, I found a user control called, “Dock Panel Suite”, developed by Weifen Luo at [www.sourceforge.net](http://www.sourceforge.net). This user control provided a way for me to implement a form document style similar to Visual Studio .NET which uses a combination of a Tabbed Document Interface and a Multiple Document Interface. The Main form, displayed in Figure 14 – Figure 16, shows the various options for layouts within the application. In addition, the user has the option to save the layout upon exiting the system. If this option is selected, an extensible markup language (XML) file is created and saved that stores the current position of each form within the application. When the
user logs back into the system, the XML file is read and the forms are displayed in the same position as the last logged in session for each user.

Figure 14: Tabbed Ticket Form and Report Form with My Tickets Docked at Bottom
Figure 15: 3 Tabbed Forms and 2 Ticket Detail Forms Docked Right

Figure 16: My Tickets Docked Left and 2 Tabbed Forms
The Dock Panel user control allows windows to be docked in five positions with an additional “floating” and “Hidden” window property. The five docked positions are: dock left, dock right, dock bottom, dock top and document. The document property is similar to a docked “fill” property in Visual Studio. Figure 17 displays the floating and hidden windows.

Figure 17: Ticket Form Floating and My Tickets Hidden
7. Conclusions

The CAGIS Helpdesk Application allows CAGIS Administrators to quickly enter the details of a support call and use information within the system to quickly resolve similar support calls. It gives administrators added flexibility by providing them a way to communicate via e-mail and through the system, prioritize tickets so that they can work on the most important issues first and it’s new user interface allows administrators to view the information they want to see, when they want to see it.

8. Recommendations

This application was designed with “progressive functionality” in mind so that it can accommodate any future changes. One example of this is a web portal where CAGIS users would be able to submit ticket requests. This add-on would be fairly simple, provided that the programmer has a basic understanding of ASP.NET. The programmer would only need to gather the information about a ticket, add a reference to the business logic dynamic link library, then call the “CreateTicket” function in order to successfully create a ticket via the web.
APPENDIX

9. Design and Development

9.1 Timeline

Senior Design I Accomplishments

- Performed initial research
- Determined requirements for application
- Identified users of the system
- Increased knowledge of Visual Basic .Net
- Created Proposal and Presentation

Senior Design II Accomplishments

- Started development of User Interface, e-mail functionality, administrator profiles
- Restructured database
- Created Test environment on laptop
- Created Design Freeze and Oral Presentation

Senior Design III Accomplishments

- Completed / Tested design
- Presented Final Project
- Submitted Final Report
9.2 Budget

The following table lists the costs to develop the CAGIS Helpdesk System:

<table>
<thead>
<tr>
<th>Item</th>
<th>Estimated Cost</th>
<th>Actual Cost</th>
</tr>
</thead>
<tbody>
<tr>
<td>Labor</td>
<td>$14644.08</td>
<td>$14644.08</td>
</tr>
<tr>
<td>VB.NET Professional</td>
<td>$548.00</td>
<td><strong>$0</strong></td>
</tr>
<tr>
<td>Oracle DB*</td>
<td>$24,000.00</td>
<td><strong>$0</strong></td>
</tr>
<tr>
<td>Server*</td>
<td>$8,000.00</td>
<td><strong>$0</strong></td>
</tr>
<tr>
<td>Desktop*</td>
<td>$1132.00</td>
<td><strong>$0</strong></td>
</tr>
<tr>
<td>Crystal Reports*</td>
<td>$427.00</td>
<td><strong>$0</strong></td>
</tr>
<tr>
<td>Total Cost*</td>
<td>$48751.08</td>
<td>$14644.08</td>
</tr>
</tbody>
</table>

Table 2: Budget

Note: Labor was calculated using a typical Sr. Computer Programmer’s hourly salary times the number of hours estimated for the project.

*Server and desktop quotes approximate cost from Dell.com, pricing for software from corporate vendors.

**Software development uses existing licenses and hardware

9.3 Hardware Requirements

The hardware that used for this project was the current hardware within CAGIS. The application was stored on CAGIS’ file server and a shortcut to run the application was to the administrator’s desktop on their existing computers.

9.4 Software Requirements

Developer Workstation: The application will be written using the coding standard for CAGIS developers, Visual Basic .NET. Since the department distributes reports using Crystal Reports, the application will utilize existing Crystal Reports licenses for its reporting capabilities.

Database: The department stores all of its data on an Oracle 9i database. The application will initially require approximately 50MB space for its data.
Desktop:

- Oracle 9i client

- Microsoft Office 2000 or above

- Meet minimum requirements for running .NET framework applications:
  
  o **Processor:**
    
    - Client (a computer not working in a server capacity): 90-megahertz (MHz) Intel Pentium-class processor, or an AMD Opteron, AMD Athlon64 or AMD Athlon XP processor
    
    - Server (a computer working in a server capacity): 133-MHz Intel Pentium-class processor, or an AMD Opteron, AMD Athlon64 or AMD Athlon XP processor

  o **Operating System(s):**
    
    - Microsoft Windows® Server 2003* (.NET Framework 1.1 is installed as part of the operating system)
    
    - Windows XP Professional*
    
    - Windows XP Home Edition
    
    - Windows 2000*
    
    - Windows Millennium Edition (Windows Me)
    
    - Windows 98
    
    - Microsoft Windows NT® 4.0 Service Pack 6a

  o **Memory:**
    
    - Client: 32 megabytes (MB) of RAM, 96 MB recommended
    
    - Server: 128 MB of RAM, 256 MB recommended
10. Sample Code

10.1 Presentation Layer

10.1.1 Formatting Datagrid for Ticket Form

Begin Code

```vbnet
#Region "Configuring Datagrid Tablestyles"
Private Sub FormatDGTix()
    dgTix.TableStyles.Clear()
    Dim dgtblstyle As New DataGridTableStyle
    Dim textcol As DataGridLabelColumn
    With dgtblstyle
        .MappingName = "Table" ""Officers"
        .ReadOnly = True
        .AllowSorting = True
        .RowHeadersVisible = True
    End With

    textcol = New DataGridLabelColumn
    textcol.MappingName = "TICKETNO" : textcol.HeaderText = "Ticket #"
    textcol.Alignment = HorizontalAlignment.Center "TextCol.Format="
    dgtblstyle.GridColumnStyles.Add(textcol)

    textcol = New DataGridLabelColumn
    textcol.MappingName = "STATUS" : textcol.HeaderText = "Status"
    textcol.Alignment = HorizontalAlignment.Center
    dgtblstyle.GridColumnStyles.Add(textcol)

    textcol = New DataGridLabelColumn
    textcol.MappingName = "priority" : textcol.HeaderText = "PRIORITY"
    textcol.Alignment = HorizontalAlignment.Left
    dgtblstyle.GridColumnStyles.Add(textcol)
```
textcol = New DataGridLabelColumn
textcol.MappingName = "OPENEDON" : textcol.HeaderText = "Date Opened"
textcol.Alignment = HorizontalAlignment.Left 'TextCol.Format=
dgtbodystyle.GridColumnStyles.Add(textcol)

textcol = New DataGridLabelColumn
textcol.MappingName = "FIRSTNAME1" : textcol.HeaderText = "Opened By"
textcol.Alignment = HorizontalAlignment.Left 'TextCol.Format=
dgtbodystyle.GridColumnStyles.Add(textcol)

textcol = New DataGridLabelColumn
textcol.MappingName = "FIRSTNAME" : textcol.HeaderText = "Assigned To"
textcol.Alignment = HorizontalAlignment.Left 'TextCol.Format=
dgtbodystyle.GridColumnStyles.Add(textcol)

textcol = New DataGridLabelColumn
textcol.MappingName = "username" : textcol.HeaderText = "User/Contact Name"
textcol.Alignment = HorizontalAlignment.Left
dgtbodystyle.GridColumnStyles.Add(textcol)

textcol = New DataGridLabelColumn
textcol.MappingName = "compname" : textcol.HeaderText = "Computer Name"
textcol.Alignment = HorizontalAlignment.Left
dgtbodystyle.GridColumnStyles.Add(textcol)

textcol = New DataGridLabelColumn
textcol.MappingName = "ipaddr" : textcol.HeaderText = "IP Address"
textcol.Alignment = HorizontalAlignment.Left
dgtbodystyle.GridColumnStyles.Add(textcol)

textcol = New DataGridLabelColumn
textcol.MappingName = "phoneno" : textcol.HeaderText = "Phone Number"
textcol.Alignment = HorizontalAlignment.Left
dgtbodystyle.GridColumnStyles.Add(textcol)

Dim txtWrapColumn As New MultiLineColumn
With txtWrapColumn
    .MappingName = "PROBDESC" : .HeaderText = "Problem Description"
    .TextBox.WordWrap = True
    .Alignment = HorizontalAlignment.Left 'TextCol.Format=
End With
dgtbodystyle.GridColumnStyles.Add(txtWrapColumn)
dgTix.TableStyles.Add(dgtbodystyle)
If MainModule.isMgrOrSec = True Or MainModule.isSpecialist = True Then
    lblAddTicket.Enabled = False
    lblCopyTix.Enabled = False
End If
End Sub

End Code

10.2 Business Logic Layer

10.2.1 Ticket class from CH_DLL, the Dynamic Link Library file for the application

Begin Code

Imports CagisServices.OracleCall
Imports CagisServices.SqlHelperParameterCache
Imports System.Data
Imports System.Data.OracleClient
Imports Microsoft.Office.Interop
Public Class Tickets
    Dim Utility As New DBUtil
    Public Function GetTicket(ByVal Ticket As Integer) As DataSet
        Try
            Dim ds As DataSet
            Dim mSTATUS As String
            Dim param1 As New OracleParameter("Vin_Ticket", OracleType.Number)
            param1.Value = Ticket
            Dim paramStatus As New OracleParameter("nSTATUS", OracleType.VarChar, 4000)
            paramStatus.Direction = ParameterDirection.Output
            Dim paramCursor As New OracleParameter("io_cursor", OracleType.Cursor)
            paramCursor.Direction = ParameterDirection.Output
            ds = CagisServices.OracleCall.ExecuteDataset(Utility.ConnectionString, CommandType.StoredProcedure, "CH_PKG.CH_TICKET", param1, paramCursor, paramStatus)
            mSTATUS = paramStatus.Value.ToString()
            If mSTATUS <> "SUCCESS" Then
                Throw New ApplicationException(mSTATUS)
            Else
                Return ds
            End If
        End Try
    End Function
End Class

End Code
Public Function GetTicketAction(ByVal Ticket As Integer) As DataSet

    Try
        Dim ds As DataSet
        Dim mSTATUS As String

        Dim param1 As New OracleParameter("Vin_Ticket", OracleType.Number)
        param1.Value = Ticket

        Dim paramStatus As New OracleParameter("nSTATUS", OracleType.VarChar, 4000)
        paramStatus.Direction = ParameterDirection.Output

        Dim paramCursor As New OracleParameter("io_cursor", OracleType.Cursor)
        paramCursor.Direction = ParameterDirection.Output

        ds = CagisServices.OracleCall.ExecuteDataset(Utility.ConnectionString, CommandType.StoredProcedure, "CH_PKG.CH_TICKETACTION", param1, paramCursor, paramStatus)

        mSTATUS = paramStatus.Value.ToString()
        If mSTATUS <> "SUCCESS" Then
            Throw New ApplicationException(mSTATUS)
        Else
            Return ds
        End If
    Catch ex As Exception
        Throw New ApplicationException(ex.Message, ex)
    End Try

End Function

End Code
REFERENCES


