Design, Efficiency Measurement and Comparative Analysis of a Dynamic Web User Interface

GRADUATE PROJECT REPORT

Submitted to the Faculty of
The Department of Computing and Mathematical Sciences
Texas A&M University- Corpus Christi
Corpus Christi, Texas

In Partial Fulfillment of the Requirements for the Degree of
Master of Science in Computer Science

By

Krishna K Molugu
Spring 2006

Committee Members

Dr. John D. Fernandez
Committee Chairperson

Dr. David Thomas
Committee Member

Dr. Long-zhuang Li
Committee Member
Design, Efficiency Measurement and Comparative Analysis of a Dynamic Web User Interface

GRADUATE PROJECT REPORT

Submitted to the Faculty of
The Department of Computing and Mathematical Sciences
Texas A&M University- Corpus Christi
Corpus Christi, Texas

In Partial Fulfillment of the Requirements for the Degree of
Master of Science in Computer Science

By

Krishna K Molugu
Spring 2006

Committee Members

Dr. John D. Fernandez
Committee Chairperson

Dr. David Thomas
Committee Member

Dr. Long-zhuang Li
Committee Member
ABSTRACT

This project includes the design and implementation of an effective dynamic interface for the Office of Planning and Institutional Effectiveness (PIE) at Texas A&M University-Corpus Christi (TAMU-CC). The PIE Website contains the results of various surveys conducted on campus and complete details of the students, their grades and other statistics. It also maintains details concerning faculty on campus. Unlike the current Website, new Dynamic PIE System computes the user specified options directly against the data and displays the results in a formatted pattern, thus making the user interface dynamic.

There were several steps involved in achieving success. The first step was timing the tasks required to browse through the current PIE Website at different levels and to rate the user friendliness and efficiency by testing with TAMU-CC employees and volunteers. Based on the results obtained, in the second step a new dynamic Website was developed using ASP.Net and C# which give the user a wide range of choices to select the data and contents he wants to retrieve and review. The newly designed system can also generate reports by using Crystal Reports (a report generating tool), which gives well formatted reports as output. Step three involved comparing the new Website against the current Website and tabulating the results. The final product is an effective and user friendly dynamic Web user interface for PIE using ASP.Net and C# which includes a capability for generating formatted reports.
# TABLE OF CONTENTS

Abstract..............................................................................................................................ii

Table of Contents...................................................................................................................iii

List of Figures.......................................................................................................................vii

List of Tables.......................................................................................................................ix

1. Introduction and Background.....................................................................................1
   1.1 Introduction...........................................................................................................1
   1.2 Overview of Current PIE System.................................................................1
       1.2.1 Student Enrollment....................................................................................2
       1.2.2 Semester Credit Hours.............................................................................3
       1.2.3 Degrees Awarded......................................................................................4
       1.2.4 Faculty and Staff.......................................................................................4
   1.3 Requirements for a New Dynamic System for PIE.......................................5

2. Dynamic PIE System..................................................................................................7
   2.1 Student Enrollment.............................................................................................8
   2.2 Student Enrollment Report................................................................................10
   2.3 Semester Credit Hours.....................................................................................11
   2.4 Semester Credit Hours Report.........................................................................12
   2.5 Degrees Awarded.............................................................................................13
   2.6 Degrees Awarded Report................................................................................15
   2.7 Faculty & Staff.................................................................................................16
   2.8 Faculty & Staff Report.....................................................................................18
   2.9 Other ASP.Net Pages.......................................................................................19
3. System Design........................................................................................................21

3.1 System Architecture............................................................................................21

3.1.1 User Interface Tier.........................................................................................22

3.1.2 Application Server Tier..............................................................................22

3.1.3 Database Tier...............................................................................................23

3.2 System Requirements.......................................................................................23

3.3 Software Requirements.....................................................................................24

3.3.1 Active Server Pages.Net (ASP.Net).........................................................24

3.3.2 Advantages of ASP.Net...............................................................................25

3.3.3 C-Sharp (C#)............................................................................................27

3.3.4 Advantages of C-Sharp..............................................................................27

3.3.5 Crystal Reports...........................................................................................28

3.3.6 Advantages of Crystal Reports.................................................................29

3.4 System Progression...........................................................................................29

3.5 Structure Chart..................................................................................................30

3.6 Database Tables..................................................................................................32

3.6.1 Table CBM001...........................................................................................33

3.6.2 Table CBM004...........................................................................................33

3.6.3 Table CBM008...........................................................................................33

3.6.4 Table CBM009...........................................................................................33
LIST OF FIGURES

Figure 1.1 Student Enrollment.................................................................2
Figure 1.2 Semester Credit Hours..........................................................3
Figure 1.3 Degrees Awarded.................................................................4
Figure 1.4 Faculty & Staff.....................................................................5
Figure 2.1 Home Page for Dynamic PIE System.................................7
Figure 2.2 Student Enrollment Page.....................................................8
Figure 2.3 Student Enrollment report....................................................10
Figure 2.4 Semester Credit Hours Page...............................................12
Figure 2.5 Semester Credit Hours Report.............................................13
Figure 2.6 Degrees Awarded Page.......................................................14
Figure 2.7 Degrees Awarded Report....................................................15
Figure 2.8 Faculty and Staff Page.......................................................16
Figure 2.9 Faculty & Staff Report.......................................................18
Figure 3.1 System Architecture..............................................................22
Figure 3.2 ASP.Net Architecture..........................................................25
Figure 3.3 Structure Chart for Dynamic PIE System............................31
Figure 3.4 ER Diagram for Student Enrollment.....................................37
Figure 3.5 ER Diagram for Semester Credit Hours...............................38
Figure 3.6 ER Diagram for Degrees Awarded.......................................39
Figure 3.7 ER Diagram for Faculty & Staff...........................................40
Figure 3.8 Context Level dataflow Diagram........................................42
Figure 3.9   Level 0 Dataflow Diagram.................................................................43
Figure 3.10  Level 1 Dataflow Diagram for Student Enrollment.........................44
Figure 3.11  Level 1 Dataflow Diagram for Semester Credit Hours......................45
Figure 3.12  Level 1 Dataflow Diagram for Degrees Awarded............................46
Figure 3.13  Level 1 Dataflow Diagram for Faculty & Staff..................................47
Figure 4.1   Usability Testing..............................................................................51
<table>
<thead>
<tr>
<th>Table</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Table 3.1</td>
<td>Hardware Requirements</td>
<td>23</td>
</tr>
<tr>
<td>Table 3.2</td>
<td>Codes and Design Files</td>
<td>48</td>
</tr>
<tr>
<td>Table 4.1</td>
<td>Average Time taken for Each Task</td>
<td>54</td>
</tr>
<tr>
<td>Table 4.2</td>
<td>User Reaction Survey Results</td>
<td>56</td>
</tr>
</tbody>
</table>
1. INTRODUCTION AND BACKGROUND

1.1 Introduction

A new system was designed for the Department of Planning and Institutional Effectiveness (PIE) to support unified access to a variety of queries related to students and faculty. The information required for the system is retrieved from a huge collection of data that is related to Texas A&M University – Corpus Christi (TAMU-CC). Staff, faculty, and students of TAMU-CC often use the data acquired from the system for work and research projects. The overall design of the system concentrated on user friendliness and ease in retrieving the required information with minimal number of clicks.

1.2 Overview of Current PIE System

The current PIE system uses Excel documents that are posted to the Website. The documents make the Website somewhat static; the documents change every semester with the new information that is entered into the database. The documents that are posted contain data related to student enrollment, semester credit hours, and more. The current system is hosted on a Web Server with the url: pie.tamucc.edu. The main purpose of that system is to display documents containing information about the students and faculty of TAMU-CC in Excel format. For a user aiming to get exact and accurate information, that approach is somewhat cumbersome. Some of the major reasons are as follows:

- The Website uses Excel documents which makes it static.
- The system has to be updated every semester whenever the data changes.
- The system administrator has to run a number of queries against the data to organize the results on the system.
Users who need specific information have to download the entire related document and search through it, to retrieve the required information.

1.2.1 Student Enrollment

The student enrollment section of the Website deals with the number of students enrolled for each semester. Figure 1.1 shows a screen capture of the current Student Enrollment page.

Figure 1.1 Student Enrollment [TAMUCC 2006]

In Figure 1.1, all the details are represented in the form of static Excel documents. The Web administrator has to do considerable work to create these documents and upload them to the server every semester. The segment, Student Enrollment, is categorized with the sections mentioned below:

- Enrollment By Class level (senior, junior, masters etc).
- Enrollment By Ethnicity
- Enrollment By Gender
➢ Enrollment By Age

➢ Enrollment By Geographic Origin

➢ Combinations of above categories

The headcount is the number of SSN’s calculated every semester to show the statistics of current enrollment. The data are analyzed and the results are generated, and then posted on the Website.

1.2.2 Semester Credit Hours

The Semester Credit Hours section is about the number of credit hours offered by a department in a particular major for that year. Figure 1.2 shows the structure of that page.

![Figure 1.2 Semester Credit Hours](TAMUCC 2006)

If the user wants to look at specific data, the user must select from the choices provided, which give the user a screen full of static Excel documents. In this segment, the credit hours offered by all departments from different colleges on campus are calculated for each student during a particular period of time. Subcategories divide the segment into groups for each college. Therefore, the administrator has the critical job of sorting and categorizing the data by college, department, level, and year.
1.2.3 Degrees Awarded

The segment ‘Degrees Awarded’ deals with critical information about the students in static Excel documents. The critical information consists of the grades earned by the student, student ID’s, and other details. Since the server holds the information, the Department of PIE must make it secure and prevent access to unauthorized users. Figure 1.3 shows an overview of this segment.

![Degrees Awarded](image)

Figure 1.3 Degrees Awarded [TAMUCC 2006]

The overall concept of the segment is to allow the user to find out how many students have graduated from a particular college in a given year. The headcount of the students is also categorized by their ethnicity, gender, level, and year. It is difficult for the end user to obtain the count with combinations of different options.

1.2.4 Faculty and Staff

Faculty and Staff is the only section in the Website that has details about faculty. In this section, the Full Time Equivalent (FTE) of faculty members is displayed. The screenshot of this section is depicted in Figure 1.4.
FTE calculations are done in a sequence of steps that include working with pivot tables, filtering the data, and categorizing into groups. The static nature of the page requires the administrator to redo the steps every semester, which consumes a lot of hours that could be utilized for other productive work.

1.3 Requirement for a New Dynamic System for PIE

The Department of Planning and Institutional Effectiveness plays an important role in any educational institution and is responsible for handling the critical information related to the campus. With rapid advancements in technology, educational institutions are moving towards more dynamic Websites with more efficiency and reduced work for the administrator. Advanced Web-based features offer many unique advantages over traditional resources, such as Excel spreadsheets and hyperlinks. An important aspect of this project is developing a system, which improves access to resources by providing sophisticated means to form queries and retrieve data.
The Dynamic User Interface was designed to solve the previously mentioned difficulties by providing a system that is not only interactive, but also has a user friendly interface. The new system includes the following features:

- It displays a variety of options in the form of drop down menus that give the end user the ability to get the data.
- When the user submits a query, the data updated by the central system (TAMU-CC) is filtered by the system which reduces the administrator’s responsibility to update the Website each semester.
- It contains a reporting tool called Crystal Reports .NET which generates well formatted reports.
2. DYNAMIC PIE SYSTEM

Dynamic PIE System designed for the Department of Planning and Institutional Effectiveness is an online Web-based system serving the needs of faculty, staff and students at Texas A&M University - Corpus Christi. Even though the whole Website was re-designed in ASP.Net, the major sections that were included in the system are:

1) Student Enrollment
2) Semester Credit Hours
3) Degrees Awarded
4) Faculty & Staff

The home page of the current system serves as the home page for the user interface as shown in Figure 2.1.

Figure 2.1 Home Page for Dynamic PIE System
The interface layout of the new system is subdivided into three parts. The top most part is occupied by the user control *banner*, which holds the banner for the newly designed system. On the left hand side of the system is placed the second user control *navigation*, which holds the navigation bar for the entire system. These two parts remain constant on the interface. The third part which is the main dynamic portion of the interface is the body of the system. This portion always changes by displaying the forms, content and reports generated during the course of the process of running the system. All the other pages on the Website apart from the above mentioned four sections are simple ASP.Net pages holding important information regarding the Department of Planning and Institutional Effectiveness.

### 2.1 Student Enrollment

This is the first Dynamic section of the PIE system. The screenshot of this section is shown in Figure 2.2.

![Figure 2.2 Student Enrollment Page](image)
This section basically deals with complete details of students who are enrolled in a given term. When a user clicks on the button *Student Enrollment* provided to him on the navigation bar it will take him to this section of the system. Here the user is given many options in the form of list boxes to select from. The options are listed below:

- *Class Level*
- *Ethnic*
- *Gender*
- *Geographic Origin*
- *College*
- *Major*
- *Year*
- *Term*

The contents of the list boxes are automatically populated from the respective tables stored in the database. The user is given an opportunity to make multiple selections from an individual list box by pressing the control key. For example, if the data is requested for the year 2001 and 2002, he will be able to select both years by holding the control key and clicking on the requested year. Once the user has completed selecting all the options needed, he will be given two additional options. At this state, if he clicks on the button *View Results* he will be diverted to the page which holds the results requested by him. Details of the results page is discussed in more detail in the following section. He can also print the results by clicking on the button *Print Results*. If in the future a new college is added to the university, the college is automatically listed in the List box *College* without any user administration required.
2.2 Student Enrollment Report

This is a dynamic ASP.Net page that is generated when a user clicks on View Results button from the previous section. The screen shot of this page is displayed in Figure 2.3.

<table>
<thead>
<tr>
<th>Student ID</th>
<th>Sex</th>
<th>Class</th>
<th>Race</th>
<th>Religion</th>
<th>Residence</th>
<th>Year</th>
<th>Grade</th>
<th>Nature of Student</th>
</tr>
</thead>
<tbody>
<tr>
<td>00000132</td>
<td>M</td>
<td>7</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>3</td>
<td>0</td>
</tr>
<tr>
<td>00000343</td>
<td>M</td>
<td>7</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>3</td>
<td>0</td>
</tr>
<tr>
<td>00000789</td>
<td>M</td>
<td>7</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>3</td>
<td>0</td>
</tr>
<tr>
<td>00000790</td>
<td>M</td>
<td>7</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>3</td>
<td>0</td>
</tr>
<tr>
<td>00000800</td>
<td>M</td>
<td>7</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>3</td>
<td>0</td>
</tr>
<tr>
<td>00000810</td>
<td>M</td>
<td>7</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>3</td>
<td>0</td>
</tr>
<tr>
<td>00000820</td>
<td>M</td>
<td>7</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>3</td>
<td>0</td>
</tr>
<tr>
<td>00000830</td>
<td>M</td>
<td>7</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>3</td>
<td>0</td>
</tr>
<tr>
<td>00000840</td>
<td>M</td>
<td>7</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>3</td>
<td>0</td>
</tr>
<tr>
<td>00000850</td>
<td>M</td>
<td>7</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>3</td>
<td>0</td>
</tr>
</tbody>
</table>

Figure 2.3 Student Enrollment Report

This page contains data depending on the user selection made from the previous page. The data is dynamically arranged in a tabular format. This part of the section is developed by Crystal Reports. Here the user is also given an option to increase the font of the text to his desired level. If the report is more then one page, one can easily navigate between the pages by click on the next and previous buttons available to him. If he knows
the exact page he is looking for, he also gets an option to directly jump to the page by entering the page number and clicking find. Search option is also available to find the specific student detail he is looking for.

2.3 Semester Credit Hours

The screenshot of this section is shown in Figure 2.4. The user is diverted to this page when he clicks on the button Semester Credit Hours available on the navigation bar. On entering this page (which is technically a form), the user is given many options to select from. This section gives the details on semester credit hours offered by every department of the specified college. From the series of list boxes available, the user is given a wide range of choices to obtain specific data he is looking for. The list boxes or the options available in this section are as follows:

- College
- Department
- Level
- Subject
- Term
- Year

Like the form Student Enrollment, the list boxes in this section are also automatically populated from the tables available in the database. Upon selecting the required options the results can be viewed in the tabulated form or can also be printed in formatted fashion. When clicked on the button View Results the selected options will be converted into a query and run against the database.
Some computations will be done with the acquired data and the final results will be displayed on the dynamic portion of the screen.

### 2.4 Semester Credit Hours Report

This page is dynamically generated when a user clicks on the button *View Results*. The contents of this page always changes depending on the options selected from the previous page. The screenshot of this page is shown in Figure 2.5. If there are more than one page of results, they are divided into several pages to avoid the scroll bar. The user can easily navigate through different pages by clicking on next page button available to him at the top of the page. He also has a search option which allows him to jump directly to the record he is looking for.
These results are again generated by using the tool Crystal reports. There is also an option available to increase the font of the text on the screen.

### 2.5 Degrees Awarded

This section gives the information on various degrees that are awarded depending on the specific selection. The screenshot of this page is shown in Figure 2.6. The user can navigate to this page by clicking on the button Degrees Awarded available in user control navigation. Similar to other pages, the user is given variety of options to select from once directed to this page. The options are displayed in the form of list boxes. The major advantage of using list boxes instead of dropdown menus is that multiple selections can be
made in list boxes which are not available in dropdown boxes. That is, one can select more than one option from the available choices.

Different options available in this section are categorized and listed below:

- **College**
- **Ethnicity**
- **Gender**
- **Level**
- **Major**
- **Degree**
- **Term**
- **Year**

**Figure 2.6 Degrees Awarded Page**

[Table showing various categories and options for degrees awarded]
The list boxes are populated from the table CBM009 and its respective lookup tables. When the primary database is updated with the new set of data the changes are automatically reflected on the page. This feature will totally eliminate the work of the administrator in updating the Website, thus saving him lots of time and energy.

2.6 Degrees Awarded Report

The screenshot of this page is shown in the Figure 2.7. The dynamic portion of the Website where this page is generated is used to display results obtained by selection made by the user from the previous page.

![Figure 2.7 Degrees Awarded Report](image-url)
The values are displayed in tabulated format. Results that are displayed can be screened to the level desired by cutting down the number of options that are available. One can again navigate through the pages by clicking on next or previous buttons available on the report. If the user wants to find a term of degree awarded for particular student from the list of results he can directly go to his record by typing his primary ID in the search space available at the top of the report.

2.7 Faculty & Staff

This section of the user interface is shown in Figure 2.8. The page Faculty & Staff deals with information regarding the faculty and staff working at TAMU-CC.

![Figure 2.8 Faculty and Staff Page](image-url)
The user can navigate to this page by clicking on the button *Faculty & Staff* available on the main navigation bar to his left. This section is further categorized into different subgroups or options that a user can select to get the information he requires. The options are available in the form of list boxes. The values in these list boxes are filled automatically from the table *CBM008* and its respective lookup tables available in the main database. Different options available on this page are listed below:

- *College*
- *Rank*
- *Gender*
- *Ethnicity*
- *Term*
- *Year*

Required results can be generated dynamically by using different combinations of the options available. The option *All* can be used from the available choices if that categorization is not required by the end user. After selecting from the available options, either the results can be viewed in a tabulated format by clicking on *View Results* Button, which will generate a dynamic page discussed in next section, or click on *Print Report* button available, which prints the formatted report using the default printer attached to the machine. Values in the list boxes are filled in without any human intervention. The administrator need not go through the hassle of updating the data every semester as the fields are automatically populated from the dataset available.
2.8 Faculty & Staff Report

This report is generated using ASP.Net and Crystal Reports. Initially a plain report is created as a template with empty rows and columns. Once the user selects the options available in the page prior to this, the columns and rows are automatically filled with the computed results and displayed in the dynamic portion of the Website. The screen shot of this page is shown in Figure 2.9.

Figure 2.9 Faculty & Staff Report

The figure above depicts the report format that is generated on clicking View Results button. Like other reports in the system, the user can increase the font of the text to his desired level. Navigation between the number of pages displayed, search option and go to specific page features are very much similar to the rest of the reports generated in the
designed system. These reports are stored in the system with an extension .rpt, and are displayed inside the dynamic portion of the Webpage developed by ASP.Net when requested.

2.9 Other ASP.Net Pages

The designed system also contains other ASP.Net pages apart from the ones discussed above. The various pages available in the system are listed below:

- History & Organization
- Budget and Finance.
- Campus Facilities
- Survey Results
- Performance Measures
- Strategic Planning
- Comparative Data
- Special Reports
- Assessments
- Common Dataset.

All the above mentioned pages are re-designed in ASP.net. These pages are used to hold data or information which is not related to the database. Information specific to every page is converted into PDF documents and posted on the Website. For example, Campus Facilities contains list of interactive PDF documents available which contain information regarding every building and different facilities available in that building on the campus.
The current Website which is hosted in the Department of Planning and Institutional Effectiveness is designed with a concept of frames. This makes it hard for Google to find the Website when someone is searching for related information on the internet. Keeping this concept in mind, the new Dynamic PIE System was developed by totally avoiding the use of frames.
3. SYSTEM DESIGN

This project is a Web based Dynamic PIE System for Department of Planning and Institutional Effectiveness at Texas A&M University – Corpus Christi. The project includes Active Server Pages.Net (ASP.Net) as front-end for designing the interface. C# is the language in which all the control statements required for running queries are implemented. The project also uses Crystal Reports.Net for generating formatted reports with the results obtained. The backend database is in MS-Access 2000.

3.1 System Architecture

A three-tier architecture was used in connecting functional building blocks together, in order to build a software system that performs according to specification. The three tier architecture consists of User Interface Tier, Application Server Tier and Database Tier [Corba 1998]. The diagrammatic representation of the system architecture is shown in Figure 3.1.

It is important to note that boundaries between tiers are logical. It is quite possible to run all three tiers on one and the same (physical) machine. It is important that the system be neatly structured, and that there is a well-planned definition of the software boundaries between the different tiers.
3.1.1 User Interface Tier

This tier is responsible for presenting the user data, retrieving the user events and controlling the user interface [Corba 1998]. The complete Web design with which the user interacts falls under this tier. This tier was designed using ASP.Net which contains different combinations of list boxes with options for the user to select.

3.1.2 Application Server Tier

This portion of the architecture contains the Windows 2000 server where the actual Website is hosted. Here is where the framework for .Net is installed which forms the base to run ASP.Net. Control statements written in C# in order to execute queries also reside here. It can be summarized that this tier handles all the Web application services.
3.1.3 Database Tier

The main responsibility of this tier is to store data [Corba 1998]. Here the data is stored in an MS-Access database. This database contains different tables and queries required for the application to work properly.

3.2 System Requirements

The project is hosted on a Windows 2000 Server operating system installed on the Planning and Institutional Effectiveness Web server. The minimum system requirements required for the Website to run are listed in Table 3.1.

Table 3.1 Hardware Requirements

<table>
<thead>
<tr>
<th>Hardware</th>
<th>Minimum System Requirements</th>
<th>Recommended</th>
</tr>
</thead>
<tbody>
<tr>
<td>Processor</td>
<td>Pentium III with at least 800MHz of processor speed</td>
<td>Pentium IV processor</td>
</tr>
<tr>
<td>Hard Disk Capacity</td>
<td>500 MB</td>
<td>2.5 GB</td>
</tr>
<tr>
<td>Operating System</td>
<td>Windows 2000 Server</td>
<td>Service Packs Installed</td>
</tr>
<tr>
<td>Memory</td>
<td>256 MB RAM</td>
<td>512 MB Ram</td>
</tr>
<tr>
<td>Ethernet Card</td>
<td>10 MBPS speed</td>
<td>100 / 1000 MBPS</td>
</tr>
<tr>
<td>Peripherals</td>
<td>Keyboard, Mouse, Monitor, CD Rom drive</td>
<td></td>
</tr>
</tbody>
</table>
3.3 Software Requirements

ASP.Net was used to develop the user interface for the Dynamic PIE System developed. The scripting required to execute queries which run against the database are developed in C#, which is a server side scripting language. MS Access 2000 is used as a database to hold various tables and queries required for the system to function. Finally, Crystal Reports is used as a reporting tool to generating formatted reports in a printable form.

The end-user can type the URL in his browser to go to the Website pie.tamucc.edu. From there he can select different options presented before him and submit the request. The submitted request will then be converted into a query to run against the database located on the server. The computed results are then presented in the format requested by the end-user.

3.3.1 Active Server Pages.Net (ASP.Net)

ASP.NET is a set of web development technologies developed by Microsoft. It can be used to build dynamic web sites, web applications and XML Web services. It is part of Microsoft's .NET platform and is the successor to Microsoft's Active Server Pages (ASP) technology. ASP.Net combined with the C# is the major part of the code in designing the system. ASP.Net is a programming framework built on the Common Language Runtime (CLR) that can be used on a server to build powerful Web applications. It provides a unified Web development model that includes the services necessary for developers to build enterprise-class Web applications. While ASP.Net is largely syntax compatible with ASP, it also provides a new programming model and infrastructure for more scalable and stable applications that help provide greater protection. ASP.Net is a compiled, .NET-based environment. You can author applications in any .NET compatible language, including
Visual Basic .Net, C#, and JScript .Net. This project used C# as the language to develop a Dynamic PIE System [ASP.Net 2006].

Figure 3.2 shows the architecture of ASP.Net. As the illustration shows, all Web clients communicate with ASP.Net applications through Microsoft Internet Information Services (IIS). This is, ASP.Net will not communicate directly with the Web clients, but will achieve this by integrating IIS in between.

![Figure 3.2 ASP.Net Architecture [Microsoft 2004]](image)

3.3.2 Advantages of ASP.Net

Benefits achieved by using ASP.Net in this project are listed below [startvbdotnet 2006]:

1) ASP.NET drastically reduces the amount of code required to build large applications

2) ASP.NET makes development simpler and easier to maintain with an event-driven, server-side programming model
3) ASP.NET pages are easy to write and maintain because the source code and HTML are together.

4) The source code is executed on the server. The pages have lots of power and flexibility by this approach.

5) The source code is compiled the first time the page is requested. Execution is fast as the Web Server compiles the page the first time it is requested. The server saves the compiled version of the page for use next time the page is requested.

6) The HTML produced by the ASP.NET page is sent back to the browser. The application source code one writes is not sent and is not easily compromised.

7) ASP.NET makes for easy deployment. There is no need to register components because the configuration information is built-in.

8) The Web server continuously monitors the pages, components and applications running on it. If it notices memory leaks, infinite loops, other illegal software or activities, it seamlessly kills those activities and restarts itself.

9) ASP.NET validates information (validation controls) entered by the user without writing a single line of code.

10) ASP.NET easily works with ADO .NET using data-binding and page formatting features.

11) ASP.NET applications run faster and counters large volumes of users without performance problems.

12) Multiple language support. Programmers can actually write their code in more than 25 .Net languages (including VB.Net, C#, and JScript.Net).
3.3.3 C-Sharp (C#)

C# is an object-oriented programming language developed by Microsoft as part of its .NET initiative. C# has a procedural, object oriented syntax based on C++ that includes aspects of several other programming languages with a particular emphasis on simplification. C# is designed to work with Microsoft’s .NET platform. Microsoft’s aim is to facilitate the exchange of information and services over the Web, and to enable developers to build highly portable applications. C# simplifies programming through its use of Extensible Markup Language (XML) and Simple Object Access Protocol (SOAP) which allow access to a programming object or method without requiring the programmer to write additional code for each step [C-Sharp 2006].

Because of its elegant object-oriented design, C# is a great choice for architecting a wide range of components—from high-level business objects to system-level applications. Using simple C# language constructs, these components can be converted into XML Web services, allowing them to be invoked across the Internet, from any language running on any operating system [C-Sharp 2006].

More than anything else, C# is designed to bring rapid development to the C++ programmer without sacrificing the power and control that have been a hallmark of C and C++. Because of this heritage, C# has a high degree of fidelity with C and C++. Developers familiar with these languages can quickly become productive in C# [C-Sharp 2006].

3.3.4 Advantages of C-Sharp

Advantages of C# are listed below:

1) Effective in resolving programming errors.
2) Garbage collection feature in C# will relieve the programmer of the burden of manual memory management.

3) Variables in C# are automatically initialized by the environment.

4) Variables are type-safe.

5) Reduces ongoing development costs with built-in support for versioning.

6) Includes native support for Windows-based applications.

3.3.5 Crystal Reports

Crystal Reports is a powerful reporting toolkit that helps to design flexible, feature-rich reports. It’s proven query technology gives access to virtually any data source, including XML, OLAP, and enterprise data sources. Extensive formatting options deliver complete control over how the data is presented to end-users [businessobjects 2005]. It lets the end-users refine their reports to meet their changing business needs. Crystal Reports includes more than 100 data presentation and interactivity options, including grouping, sorting, field highlighting, and running totals. For managed report delivery via the Web, a special offer of Crystal Enterprise Express Edition is included in Crystal Reports Professional Edition. Crystal Reports is available in different Editions to meet the needs of developers, report designers, and business users [businessobjects 2005].

Crystal Reports for .NET provides developers with a fast and highly productive way to create and integrate presentation-quality, interactive reports that scale to meet the end user demands. It is available as a tightly integrated feature in Microsoft Visual Studio .Net. With Crystal Reports for Visual Studio.Net, one can quickly create complex and professional-looking reports. Instead of coding, one uses the Crystal Report Designer
interface to create and format the report needed. The powerful Report Engine processes the formatting, grouping, and charting criteria one specifies.

3.3.6 Advantages of Crystal Reports

Crystal Reports are latest technology available in the market for designing formatted outputs in the form of reports. Some of the major advantages of Crystal Reports are listed below [businessobjects 2005]:

1) Transform your corporate data into powerful, dynamic information
2) Easily create reports for Windows, Web, and XML Web services applications
3) Maintain complete control over data access and presentation
4) Deliver dynamic, graphical reports in zero-client or rich-client environments
5) Empower end users with rich report interaction
6) Save time and write less code with existing Crystal reports and report design knowledge in your .NET projects
7) Minimize requests for report modifications

3.4 System Progression

A brief description of different steps that were involved in achieving a successful system is briefly outlined below.

- Recognized the need for a system in the Department of Planning and Institutional Effectiveness and acquired the required information.
- A simple prototype of the system was developed and reviewed with the staff.
- Necessary modifications were made from the review to get the final approval of the concept.
• Developed a rough sketch of structure charts and flow diagrams to get a detail view of the system.
• Social Security Numbers (SSN’s) were replaced with unique ID’s to protect the confidentiality of the system.
• Various tables required for the system were designed with the given data.
• Dynamic user interactive Web pages were developed using ASP.Net and C#.
• Different forms were created at each level to retrieve the data based on the user selections.
• Formatted reports were generated using Crystal Reports to process the results in a printable format.
• The system was tested on different scenarios and final changes were made.
• The project was demonstrated at Department of Planning and Institutional Effectiveness
• The new system was tested against the old one with similar tasks and the results were compared against the old system.
• Imported the actual SSN’s into the database.

3.5 Structure Chart

A structure chart is used to demonstrate how the inter related groups of the system are constructed. It gives the basic idea of what sub-groups or categories make up the system as a whole. The structure chart for the designed system is shown in Figure 3.3.
Figure 3.3 Structure Chart for Dynamic PIE System

As shown in Figure 3.3 the main PIE Website is divided into four different categories Student Enrollment, Semester Credit Hours, Degrees Awarded and Faculty & Staff. These four categories are again sub-divided into further groups like Ethnicity, Gender, Class Level etc. After selecting different options the user will then be directed to two options: View Reports or Print Results.


3.6 Database Tables

Microsoft Access is a relational database management system, packaged with Microsoft Office Professional which combines the relational Microsoft Jet Database Engine with a graphical user interface. It can use data stored in Access/Jet, SQL Server, Oracle, or any ODBC-compliant data container. It can be used to develop powerful, complex application software. On the other hand it can also be used to build simple applications without having to deal with features that need not be used. It supports substantial object-oriented (OO) techniques but falls short of being a fully OO development tool.

Microsoft Access was used to develop the backend of the project. Different tables and lookup tables required to run the designed system are as follows:

- CBM001
- CBM004
- CBM008
- CBM009
- Class
- College
- Department
- Ethnic
- Gender
- Level
- Rank
- SPE
• *Term_Names*

3.6.1 Table *CBM001*

*CBM001* is the main table used by the group Student Enrollment. The screen shot of this table is shown in the Appendix E. The primary id of this table is *Student_Id*, which is the Social Security Number of the students. This table is populated with data about enrollment of students at different levels in different departments for a given term.

3.6.2 Table *CBM004*

The table *CBM004* shown in the Appendix E forms the main table for the section Semester Credit Hours. Unlike the table CBM001 this table contains the data regarding credit hours offered by various departments in a particular college in a given term of a year. The major fields in this table are *Course Number, Course Title, College, Department, Term and Year*.

3.6.3 Table *CBM008*

As shown in the Appendix E the table *CBM008* has information regarding the Faculty & Staff of TAMU-CC. It covers the major fields about faculty like ranks, sex, department, ethnicity and salary. This table is mainly used to calculate the Full Time Equivalent (FTE) of a faculty and also the head count of the entire faculty for each department. The field *ssn* is the primary key and *term* and *year* are the foreign keys for this table. The screen shot of the table is shown in Figure 3.6.

3.6.4 Table *CBM009*

Appendix E shows the table *CBM009* that forms the main table for the section Degrees Awarded. This table contains data about graduation of students at TAMU-CC. Major fields that are concentrated in this table are *Student_ID, Sex, Ethnicity, Degree,*
Term. When the selected options are computed against this table it will display the number of degrees awarded according to that particular selection.

3.6.5 Table Class

Appendix E shows a lookup table Class with field names class (primary Key), level and class name. This is mainly used by Student Enrollment section to display names of class like senior, junior, masters and doctoral instead of their class codes.

3.6.6 Table College

Appendix E shows the lookup table College with field names ID (primary Key), college and college name. It is an important table which is used by almost every section of the project. It gives the name of the college depending upon its college ID.

3.6.7 Table Department

As shown in Appendix E the lookup table Department holds the names of different departments corresponding to their respective department codes. Important fields in the table include college (Foreign Key), dept (Primary Key), dept_name and admin_code (Foreign Key). The section Semester Credit Hours uses this lookup table.

3.6.8 Table Ethnic

Screenshot of the lookup table ethnic is shown in Appendix E. This is a small table with only two fields ethnic and ethnic_name in it. This table is used to display ethnic names for all faculty, staff and students of TAMU-CC instead of their ethnic codes. Although it is a small table, it was used by almost every section in the project.

3.6.9 Table Gender

Appendix E shows the lookup table gender holding two fields sex and gender with only two values male and female under them. This table was constantly used to make
further categorization in the displayed results. The field sex is the primary key for this table.

3.6.10 Table level

The lookup table level is similar to the lookup table class but is used by the different group Semester Credit Hours with different type of categorization. Here the levels are divided into remedial, lower, upper, masters and doctoral instead of senior, junior etc. The screenshot of this table is shown in Appendix E.

3.6.11 Table rank

Like the level of the student in the lookup table’s class and level, this table depicts the rank of the faculty. It is mainly used in combination with table CBM008 to calculate the FTE of faculty based on their rank. This table has only two fields rank and rank_name where rank is the primary id of the table. The screenshot of this table is shown in Appendix E.

3.6.12 Table term_names

The lookup table term_names determines the name of the term based on its term number. The term number ‘9’ is used for fall semester, ‘1’ is used for spring semester, ‘6’ is used for summer1 and ‘7’ is used for summer2. This table plays a major role when the user wants to look at data, specific to a particular semester. It has two fields: term and name with term as primary key. The screenshot of this table is shown in the Appendix E.
3.7 Entity Relationship Diagrams

The entity-relationship model or entity-relationship diagram (ERD) is a data model or diagram for high-level descriptions of conceptual data models, and it provides a graphical notation for representing such data models in the form of entity-relationship diagrams [iinet 2006]. Such models are typically used in the first stage of information-system design; they are used, for example, to describe information needs and/or the type of information that is to be stored in the database during the requirements analysis. The data modelling technique, however, can be used to describe an overview and classifications of used terms and their relationships.

Entity relationship diagrams are used to identify the data that must be captured, stored and retrieved from a designed system [iinet 2006]. They also are used to identify the data required to derive and generate reports on the performance measures that a system should be monitoring. ER diagrams have three different components [iinet 2006]:

1) Entities: Entity is anything that a system needs to store data about.

2) Attributes: Entities are further described with the attributes. These are the smallest units of data that can be described in a meaningful manner.

3) Relationships: Frequently, a meaningful relationship exists between the entities. There are potentially three types of relations that can exist between two different entities. They are:

   • One-to-One Relationships
   • One-to-Many Relationships
   • Many-to-Many Relationships
3.7.1 ER diagram for Student Enrollment

The entity relationship diagram for the section Student Enrollment is shown in Figure 3.4. One can see in the diagram the different tables used by this section and also the relations between them. *CBM001* is the important table of this section, which is mapped to various lookup tables to resolve the final results.

![Figure 3.4 ER Diagram for Student Enrollment](image-url)
3.7.2 ER diagram for Semester Credit Hours

Figure 3.5 shows the entity relationship diagram for the section Semester Credit Hours. The main table used by this section is CBM004. The diagram also shows all the lookup tables and their mapping structure with the main table CBM004.

Figure 3.5 ER Diagram for Semester Credit Hours
3.7.3 ER diagram for Degrees Awarded

The ER Diagram for the section Degrees Awarded is shown in Figure 3.6. CBM009 is main table that is used by this section. The diagram explicitly explains the relationship of CBM009 with the rest of the lookup tables that are used in this section. The system uses these tables to retrieve the data from database upon end-user request.

![ER Diagram for Degrees Awarded](image)

Figure 3.6 ER Diagram for Degrees Awarded
3.7.4 ER diagram for Faculty and Staff

Figure 3.7 shows the ER Diagram for the section Faculty and Staff. The main table used by this section is CBM008. The diagram shows different lookup tables used by this section with the links between each table.

Figure 3.7 ER Diagram for Faculty & Staff
3.8 Dataflow Diagrams

A data flow diagram (DFD) is a graphical representation of the flow of data through an information system. A data flow diagram can also be used for the visualization of data processing (structured design). It is common practise for a designer to draw a context-level DFD first which shows the interaction between the system and outside entities [Wiki 2006]. This context-level DFD is then "exploded" to show more detail of the system being modelled.

A data flow diagram illustrates the processes, data stores, and external entities in a business or other system and the data flows between these things. Four diagramatical components are used to develop a DFD. These are[Wiki 2006]:

1) **Data Flow:** It is represented by an arrow.

2) **Data Process:** This is represented by a circle or a rounded rectangle. A data process represents the transformation of data in the system. Generally, this represents something that happens in the system, such as Student Enrollment. Data that flows into a process should be different from the data that flows out of the process.

3) **External Entity:** Represented by square or oval. An external entity represents the source or sink of data external to the system. When modeling a DFD, the designer is not interested in the inner workings of the external entity, but only what data is produced/needed by the entity.

4) **Data Store:** Represented by two parallel line sometimes connected by a vertical lines. A data store is a repository for data. Data stores can be manual, digital, or temporary.
3.8.1 Context Level Dataflow Diagram

The top-level diagram is called a context diagram and contains one Functional Transform which represents the entire system. The context diagram also shows the external entities that interact with the system [Context DFD 2000]. The context level dataflow diagram is shown in the Figure 3.8.

Figure 3.8 Context Level Data Flow Diagram

Faculty, staff and students are three external entities used to define the data flow for the designed system. The end-user request for the information and system response for each request is shown with directed arrows. Next level of data flow diagrams are shown in the sections below.
3.8.2 Level 0 Dataflow Diagram for PIE System

The Level 0 dataflow diagram [Alan 2006] is shown in Figure 3.9. This dataflow diagram describes the system in more detail.

![Figure 3.9 Level 0 Dataflow Diagram](image)

It shows how all external entities communicate with the central database using different processes. The whole system can be summarized as follows: when a task is submitted, it is executed against the database and the computed results are displayed on the screen.
3.8.3 Level 1 Dataflow Diagram for Student Enrollment

Figure 3.10 shows level 1 dataflow diagram for section Student Enrollment. It shows the flow of data when a task, retrieve student enrollment for the current year, is executed. A faculty in this case is given an opportunity to select from various options he wants included in the request. Then the request is translated in the form of a query. The obtained query is then computed against data store Student Enrollment and results are obtained. The results are then arranged in a tabular format. The faculty is given an option to view the tabulated results or go further in the process and print a formatted report of information he requested.

Figure 3.10 Level 1 Dataflow Diagram for Student Enrollment
3.8.4 Level 1 Dataflow Diagram for Semester Credit Hours

A 1 level dataflow diagram for the section Semester Credit Hours is shown in Figure 3.11. The diagram represents student (an external entity for the system) requesting information to find number of credit hours offered by a department in the same year. His request is first converted in the form of a query and run against the data store Semester credit Hours. From the results obtained, the SCH offered by that department are calculated and the final results are tabulated. The student is again given an option to view the tabulated results at this point or further go ahead in the process and print a formatted report.

Figure 3.11 Level 1 Dataflow Diagram for Semester Credit Hours
3.8.5 Level 1 Dataflow Diagram for Degrees Awarded

The dataflow diagram in the Figure 3.12 shows two of external entities trying to access the same data store. In this case both faculty and staff submit their individual requests. The requests will then be processed and the headcount is calculated by the number of degrees awarded by a department in the same year. The results are displayed in tabulated format or reports are printed upon user request.

![Figure 3.12 Level 1 Dataflow Diagram for Degrees Awarded](image-url)
3.8.6 Level 1 Dataflow Diagram for Faculty & Staff

Figure 3.13 shows level 1 dataflow diagram for the section Faculty and Staff. The flow of data for different processes in the system is shown for a request to get FTE for the selected professors.
### 3.9 Essential Pages and Code Files

Table 3.2 shows important design files and code files used in the project development.

#### Table 3.2 Codes and Design Files

<table>
<thead>
<tr>
<th>File Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Global.asax</td>
<td>ASP.Net application file. Contains code to respond to application level events</td>
</tr>
<tr>
<td>Header1.ascx</td>
<td>User Control file used to define banner for PIE</td>
</tr>
<tr>
<td>LeftHeader.ascx</td>
<td>User Control file for Navigation Buttons</td>
</tr>
<tr>
<td>Web.config</td>
<td>Configuration file that stores global information (See Appendix A)</td>
</tr>
<tr>
<td>Index.aspx</td>
<td>Start page for the website. Actual ASP.Net file</td>
</tr>
<tr>
<td>Index.aspx.cs</td>
<td>Code behind file in C-sharp for index page</td>
</tr>
<tr>
<td>Index.aspx.resx</td>
<td>Resource file for index page</td>
</tr>
<tr>
<td>StudentEnrollment.aspx</td>
<td>ASP.Net form for Student Enrollment page</td>
</tr>
<tr>
<td>StudentEnrollment.aspx.cs</td>
<td>Code Behind page for StudentEnrollment</td>
</tr>
<tr>
<td>StudentEnrollment.aspx.resx</td>
<td>Resource file for StudentEnrollment</td>
</tr>
<tr>
<td>StudentEnrollmentReportForm.aspx</td>
<td>Displays results for StudentEnrollment</td>
</tr>
<tr>
<td>StudentEnrollmentReportForm.aspx.cs</td>
<td>Code Behind file for StudentEnrollmentReportForm</td>
</tr>
<tr>
<td>StudentEnrollment.rpt</td>
<td>Crystal Reports File for StudentEnrollment</td>
</tr>
<tr>
<td>File Name</td>
<td>Description</td>
</tr>
<tr>
<td>------------------------</td>
<td>--------------------------------------------------</td>
</tr>
<tr>
<td>CreditHours.aspx</td>
<td>ASP.Net form for Semester Credit Hours</td>
</tr>
<tr>
<td>CreditHours.aspx.cs</td>
<td>Code Behind page for CreditHours</td>
</tr>
<tr>
<td>CreditHours.aspx.resx</td>
<td>Resource page for CreditHours</td>
</tr>
<tr>
<td>CreditHoursReport.rpt</td>
<td>Crystal reports File for CreditHours</td>
</tr>
<tr>
<td>Degreesawarded.aspx</td>
<td>ASP.Net form for Degrees Awarded</td>
</tr>
<tr>
<td>Degreesawarded.aspx.cs</td>
<td>Code Behind page for Degreesawarded</td>
</tr>
<tr>
<td>Degreesawarded.aspx.resx</td>
<td>Resource file for Degreesawarded</td>
</tr>
<tr>
<td>DegreesawardedReportForm.aspx</td>
<td>Displays results for Degreesawarded</td>
</tr>
<tr>
<td>DegreesawardedReportForm.aspx.resx</td>
<td>Resource file for DegreesawardedReportForm</td>
</tr>
<tr>
<td>DegreesReport.rpt</td>
<td>Crystal Report File for Degreesawarded</td>
</tr>
<tr>
<td>Facultyandstaff.aspx</td>
<td>ASP.Net form for Faculty &amp; Staff</td>
</tr>
<tr>
<td>Facultyandstaff.aspx.cs</td>
<td>Code Behind file for Facultyandstaff</td>
</tr>
<tr>
<td>Facultyandstaff.aspx.resx</td>
<td>Resource file for Facultyandstaff</td>
</tr>
<tr>
<td>Facultyandstaffreportform.aspx</td>
<td>Displays results for Facultyandstaff</td>
</tr>
<tr>
<td>Facultyandstaffreportform.aspx.cs</td>
<td>Code Behind file for Facultyandstaffreportform</td>
</tr>
<tr>
<td>Facultyandstaffreportform.aspx.resx</td>
<td>Resource file for Facultyandstaffreportform</td>
</tr>
<tr>
<td>FacultyandStaffReport.rpt</td>
<td>Crystal Report file for Faculty &amp; Staff</td>
</tr>
</tbody>
</table>
4. EVALUATION AND RESULTS

The Dynamic PIE System developed for the Department of Planning and Institutional Effectiveness was tested and evaluated at every stage of development. The system was checked at every level to find errors and the respective queries were altered. Entity Relationship diagrams also played a major role in mapping the relations between tables and the respective lookup tables.

The testing performed on the system can be sub-divided into two categories. They are as follows:

1) Structural Testing
2) Usability Testing

4.1 Structural Testing

In this phase of testing, the code of the project was tested. Every module used in the project was tested against different combination of options to obtain the expected results. The written code was tested for different inputs to verify the output with the expected result [Uottawa 2005]. The main theme of this testing method is to focus on the functional requirements of the project rather than on the ease of user interface. By using this method it was easy to detect the deadlocks in the code which stopped the system from producing computed results. Since the internal state of the entity is being tested here, the errors in links between the different entities were also resolved.

Some of the error messages which occurred during the development stage of the project were resolved by using some online help available in the IT market. Special help was also taken from the online forms available at experts-exchange [EE 2006] to resolve a
particular error in the code found during this test phase. It was easy to debug the code in small chunks then trying to work with the entire code at once.

4.2 Usability Testing

This type of testing method plays an important role in any system which involves Web user interface. Usability testing is a technique for ensuring that the intended users of a system can carry out the intended tasks efficiently, effectively and satisfactorily [Info 2006]. Usability testing is carried out before the Website is released so that any significant issues that are identified can be modified accordingly to make the interface look more user friendly.

By applying this testing the human-computer interactions of the design were tested, and the required part of the code was enriched depending on the survey results. The usability test is typically attended by

- A representative user
- Test host

Figure 4.1 shows a typical layout of usability testing.

![Figure 4.1 Usability Testing [Info 2006]](image)
After the system was completely developed, a list of tasks was prepared for the user to accomplish. The following concerns were considered while preparing the tasks.

- Easy navigation system of the project
- Straightforwardness of the design in producing requested results
- Ability to print formatted results
- Flexibility in selecting from different options available.
- The overall view and design issues

4.3 Testing Process

The test process took several different steps. The sections in the project that were to be tested were identified and listed. There are three different types of external entities in the system, which are users of the system. They are faculty, staff and students. A sample of ten testers was selected, some of whom are regular users of the Website. The sample included five students, one faculty and four staff members of Texas A&M University-Corpus Christi.

The users were given three different tasks to be performed on the current Website and also on the new Dynamic PIE System that was developed. The three tasks were as follows:

1) You are doing a report for the State and need to find complete details about male students who are enrolled for doctoral program in Fall 2003 with ethnicity White.

2) You would like to print a detailed report on all female students who got there degrees in the year 2001 with a nursing major and ethnicity Black.
3) Assume that you are a faculty and want to know the details about *White* and *Hispanic Professors* in the college of *Arts & Humanities* that were present in *Fall 2004*

These three tasks were performed by the ten users in a sequence. First, the testing was done with the current Website and the start time and end time was noted for every task. Also the comments and suggestions which were given by the users are noted which were further turned into modifications for the new Dynamic PIE System developed.

In the second step the same three tasks were performed with the new Dynamic PIE System. Each task was again timed. Comments and suggestions made on new user interface were noted.

Finally, a User Reaction Survey was designed to get feedback from the testers. The questionnaire was developed with different questions pertaining to usability issues. The user reaction surveys filled by the testers are attached at the end of the report in Appendix C. Users that were involved in the usability testing are:

1) Ms. Phyllis Tedford, Instructor, College of Science and Technology.
2) Ms. Olivia Bayarena, Senior Academic Advisor, College of Science and Technology
3) Ms. Karrin Griffith, Transfer Counselor, Academic Advising Transition Center
4) Abel Cantu, Network Manager I, Library Operations
5) Ms. Gloria Ann Valdez, Business Co-Ordinator, College of Business
6) Jayanthi Kalvemula, Student, TAMU-CC
7) Nimitha Mukherjee, Student, TAMU-CC
8) Ramakrishna Vodetti, Student, TAMU-CC
9) Ankita Nellimarla, Student, TAMU-CC
All the users that were involved in the usability testing gave positive feedback. They liked the user interface and the fact that they can select multiple options from the list boxes available for them which was lacking in the current Website. They also liked the easy navigation system designed in the project. Most of all they liked the fact where they were able to get all the combinations of data they need in a single attempt which saved lot of there time and energy.

**4.4 Comparative Study for Two Websites**

When the test results were reviewed and analyzed, the new system proved much better then the existing one. The users retrieved the test data more efficiently and in much less time with the built-in list boxes then from the current Website. Usability assessment was conducted by recording the time taken to complete each of the three tasks discussed in the above section. The averaged results were tabulated as shown in Table 4.1.

**Table 4.1 Average Time Taken for Each Task**

<table>
<thead>
<tr>
<th>Tasks \ Websites</th>
<th>Current Website (minutes)</th>
<th>New Dynamic PIE System (minutes)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Task 1</td>
<td>3:15</td>
<td>1:03</td>
</tr>
<tr>
<td>Task 2</td>
<td>2:48</td>
<td>1:13</td>
</tr>
<tr>
<td>Task 3</td>
<td>3:50</td>
<td>0:41</td>
</tr>
</tbody>
</table>
By reviewing the results it is clear that the new Web User Interface is easy and quick for completing required tasks. The testers felt very comfortable with browsing the User Interface and also liked the fact that all the options were available to them and could be combined in different ways to obtain results in a single click.

4.5 User Reaction Survey

A small survey was conducted when the testing was completed. The main purpose of this survey was to determine the reactions of the testers to general and specific features of the system. The survey consisted of five questions to which the user respond by circling the appropriate options (Strongly Disagree, Disagree, Neutral, Agree, Strongly Agree). The questions in the survey were as follows:

1) Browsing the Website was as comfortable as the best Websites I have visited.

2) It was easy to follow the menus and navigation approaches used in the design

3) It was easy to accomplish the test tasks provided by the designer

4) The colors and graphics were pleasing to the eye

5) If this site were available to me, I would use it and recommend it to others

To analyze the collected results each option is tagged with a specific number so that it can be averaged at the end of the survey. For example Strongly Disagree is considered to be -2, Disagree is considered to be -1, Neutral is 0, Agree is +1 and Strongly Agree is +2.

Table 4.2 shows the tabulated results. Test results were computed and then averaged by ten users.
Table 4.2 User Reaction Survey Results

<table>
<thead>
<tr>
<th>Question Numbers</th>
<th>Average Rating</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>+ 1.8</td>
</tr>
<tr>
<td>2</td>
<td>+ 1.6</td>
</tr>
<tr>
<td>3</td>
<td>+ 1.6</td>
</tr>
<tr>
<td>4</td>
<td>+ 1.8</td>
</tr>
<tr>
<td>5</td>
<td>+ 2.0</td>
</tr>
</tbody>
</table>

By reviewing the results it can be seen that almost all the people either *Agree* or *Strongly Agree* that the Dynamic PIE System deserves high rating for usability.

In addition to these questions, there were also three open-ended questions that were included in the survey. They are listed below:

1) What two things did you like best and why?

2) What two things did you like least and why?

3) What would you suggest as changes to the design?

The comments given for these questions were studied carefully. Only one required a modification to the user interface to make it more user friendly. Ms. Phyllis Tedford, advised to add an on screen instruction saying “Hold Ctrl key to select more then one option”. After studying her request a label saying “Hold Control Key to make multiple Selections from the list boxes” was added to the Website.
5. FUTURE WORK

In the future, the functionality of the Dynamic PIE System can be expanded by adding new features. The flexible nature of ASP.Net and C-Sharp used to develop this system will make it easy to add any new module to the existing system. The system can be expanded by adding a new module that will generate graphs from the tabulated data projected on the screen. These graphs can then be printed in a formatted fashion using Crystal reports. The user can also be given an option to change the graph styles from bar graphs to pie charts and many more. It will also be a good idea to split the system into two parts, where the system administrators will have their own sub-system which is password protected. Using this approach, the administrators can look at more detailed data for each record which is not available for the regular users. The administrative segment can be protected by using logins and passwords.

It is also recommended to use secure licenses available to make the Website more secure. It is necessary to use this secure layer if the system contains critical data like Social Security Numbers and other information regarding TAMU-CC. Secure layer can be purchased from many third party vendors available like thawte, Verisign, entrust etc.
6. CONCLUSION

The Dynamic PIE System developed for the Department of Planning and Institutional Effectiveness will help staff, faculty and students to get the exact data they are looking for. The robust features of the system allow end-users to select from a variety of options that are available to them during the course of a process. With the enriched technology of ASP.Net, it really makes the user interface dynamic by automatically populating the list boxes from the respective tables, which are updated by the central authority of TAMU-CC. These new features will relieve the administrator from updating the Website every semester, when new data is added to the database. With the new system, the users can retrieve data with all possible combinations of options available in a single attempt. The new system is not only useful for staff, faculty and students of TAMU-CC, but is also very effective for the administrator.


APPENDIX A: Sample Code

Web.config file

```xml
<?xml version="1.0" encoding="utf-8"?>
<configuration>
  <appSettings>
    <add key="ConStr" value="Provider=Microsoft.Jet.OLEDB.4.0;Data Source="/>
  </appSettings>
  <system.web>

    <!-- DYNAMIC DEBUG COMPILATION
    Set compilation debug="true" to enable ASPX debugging.
    Otherwise, setting this value to false will improve runtime performance of this application.
    Set compilation debug="true" to insert debugging symbols (.pdb information)
    into the compiled page. Because this creates a larger file that executes
    more slowly, you should set this value to true only when debugging and to
    false at all other times. For more information, refer to the documentation about
    debugging ASP.NET files. -->
    <compilation defaultLanguage="c#" debug="true"><assemblies><add
      assembly="CrystalDecisions.CrystalReports.Engine, Version=9.1.5000.0,
      Culture=neutral, PublicKeyToken=692fbea5521e1304"/><add
      assembly="CrystalDecisions.ReportSource, Version=9.1.5000.0,
      Culture=neutral, PublicKeyToken=692fbea5521e1304"/><add
      assembly="CrystalDecisions.Shared, Version=9.1.5000.0, Culture=neutral,
      PublicKeyToken=692fbea5521e1304"/><add
      assembly="CrystalDecisions.Web, Version=9.1.5000.0, Culture=neutral,
      PublicKeyToken=692fbea5521e1304"/></assemblies></compilation>

    <!-- CUSTOM ERROR MESSAGES
    Set customErrors mode="On" or "RemoteOnly" to enable custom error mess_u103 ?es, "Off" to disable.
    Add <error> tags for each of the errors you want to handle.
    "On" Always display custom (friendly) messages.
    "Off" Always display detailed ASP.NET error information.
    "RemoteOnly" Display custom (friendly) messages only to users not running
    on the local Web server. This setting is recommended for security purposes, so
    that you do not display application detail information to remote clients. -->
    <customErrors mode="RemoteOnly"/>

    <!-- AUTHENTICATION
```
This section sets the authentication policies of the application. Possible modes are "Windows", "Forms", "Passport" and "None"

"None" No authentication is performed.
"Windows" IIS performs authentication (Basic, Digest, or Integrated Windows) according to its settings for the application. Anonymous access must be disabled in IIS.
"Forms" You provide a custom form (Web page) for users to enter their credentials, and then you authenticate them in your application. A user credential token is stored in a cookie.
"Passport" Authentication is performed via a centralized authentication service provided by Microsoft that offers a single logon and core profile services for member sites.

<authentication mode="Windows"/>

<!-- AUTHORIZATION
This section sets the authorization policies of the application. You can allow or deny access to application resources by user or role. Wildcards: "*" mean everyone, "?" means anonymous (unauthenticated) users.
-->

<authorization>
  <allow users="*"/> <!-- Allow all users -->
  <!-- <allow users="[comma separated list of users]" roles="[comma separated list of roles]"/> -->
  <deny users="[comma separated list of users]" roles="[comma separated list of roles]"/>
</authorization>

<!-- APPLICATION-LEVEL TRACE LOGGING
Application-level tracing enables trace log output for every page within an application.
Set trace enabled="true" to enable application trace logging.
If pageOutput="true", the trace information will be displayed at the bottom of each page.
Otherwise, you can view the application trace log by browsing the "trace.axd" page from your web application root.
-->

<trace enabled="false" requestLimit="10" pageOutput="false" traceMode="SortByTime" localOnly="true"/>

<!-- SESSION STATE SETTINGS
By default ASP.NET uses cookies to identify which requests belong to a particular session.
If cookies are not available, a session can be tracked by adding a session identifier to the URL.
To disable cookies, set sessionState cookieless="true". 
<sessionState mode="InProc"
stateConnectionString="tcpip=127.0.0.1:42424" sqlConnectionString="data source=127.0.0.1;Trusted_Connection=yes" cookieless="false"
timeout="20"/>

<!-- GLOBALIZATION
This section sets the globalization settings of the application.
-->
<globalization requestEncoding="utf-8" responseEncoding="utf-8"/>

</system.web>

</configuration>
APPENDIX B: Usability Test Scenario Form Used

Planning and Institutional Effectiveness Website
Task Plan

Assume you are a faculty, staff or student of Texas A&M University – Corpus Christi. You are to accomplish the following tasks:

1. You are doing a report for the State and need to find out complete details about Male students who enrolled for Doctoral Program in Fall 2003 with Ethnicity White.

2. You would like to print a detailed report on all Female students who got their degrees in the year 2001 with a Nursing major and Ethnicity Black.

3. Assume that you are a faculty and want to know the details about White and Hispanic Professors in the college of Arts & Humanities that were present in Fall 2004.
APPENDIX C: User Reaction Survey Form Used

New Dynamic PIE System

Now that you have completed the usability tasks, we would like to know some of your reactions, both in general and to specific features of the system.

Evaluator Name: _________________________________  Date: ________________

Please respond to the following items by circling the opinion that best corresponds to your own:

1. Browsing the Web site was as comfortable as the best Web sites I have visited.

   Strongly Disagree  Disagree  Neutral  Agree  Strongly Agree

2. It was easy to follow the menus and navigation approaches used in the design.

   Strongly Disagree  Disagree  Neutral  Agree  Strongly Agree

3. It was easy to accomplish the test tasks provided by the designers.

   Strongly Disagree  Disagree  Neutral  Agree  Strongly Agree

4. The colors and graphics were pleasing to the eye.

   Strongly Disagree  Disagree  Neutral  Agree  Strongly Agree

5. If this site were available to me, I would use it and recommend it to others.

   Strongly Disagree  Disagree  Neutral  Agree  Strongly Agree

What two things did you like best and why?

What two things did you like least and why?

What would you suggest as changes to the design?
Appendix D
Survey Results
Appendix E

Database Tables
### Table CBM001

<table>
<thead>
<tr>
<th>Field Name</th>
<th>Data Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>id</td>
<td>Text</td>
</tr>
<tr>
<td>name</td>
<td>Text</td>
</tr>
<tr>
<td>address</td>
<td>Text</td>
</tr>
<tr>
<td>phone</td>
<td>Text</td>
</tr>
<tr>
<td>email</td>
<td>Text</td>
</tr>
<tr>
<td>job</td>
<td>Text</td>
</tr>
<tr>
<td>city</td>
<td>Text</td>
</tr>
<tr>
<td>state</td>
<td>Text</td>
</tr>
<tr>
<td>zip</td>
<td>Text</td>
</tr>
<tr>
<td>country</td>
<td>Text</td>
</tr>
<tr>
<td>dob</td>
<td>Text</td>
</tr>
<tr>
<td>gender</td>
<td>Text</td>
</tr>
<tr>
<td>race</td>
<td>Text</td>
</tr>
<tr>
<td>marital</td>
<td>Text</td>
</tr>
<tr>
<td>education</td>
<td>Text</td>
</tr>
<tr>
<td>occupation</td>
<td>Text</td>
</tr>
<tr>
<td>salary</td>
<td>Number</td>
</tr>
<tr>
<td>benefits</td>
<td>Text</td>
</tr>
<tr>
<td>notes</td>
<td>Text</td>
</tr>
<tr>
<td>notes</td>
<td>Text</td>
</tr>
<tr>
<td>notes</td>
<td>Text</td>
</tr>
<tr>
<td>notes</td>
<td>Text</td>
</tr>
<tr>
<td>notes</td>
<td>Text</td>
</tr>
<tr>
<td>notes</td>
<td>Text</td>
</tr>
<tr>
<td>notes</td>
<td>Text</td>
</tr>
<tr>
<td>notes</td>
<td>Text</td>
</tr>
<tr>
<td>notes</td>
<td>Text</td>
</tr>
<tr>
<td>notes</td>
<td>Text</td>
</tr>
<tr>
<td>notes</td>
<td>Text</td>
</tr>
<tr>
<td>notes</td>
<td>Text</td>
</tr>
<tr>
<td>notes</td>
<td>Text</td>
</tr>
<tr>
<td>notes</td>
<td>Text</td>
</tr>
<tr>
<td>notes</td>
<td>Text</td>
</tr>
<tr>
<td>notes</td>
<td>Text</td>
</tr>
</tbody>
</table>

*Field Properties*
<table>
<thead>
<tr>
<th>Field Name</th>
<th>Data Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>subject</td>
<td>Text</td>
</tr>
<tr>
<td>paragraph</td>
<td>Text</td>
</tr>
<tr>
<td>title</td>
<td>Text</td>
</tr>
<tr>
<td>incep</td>
<td>Number</td>
</tr>
<tr>
<td>credit</td>
<td>Number</td>
</tr>
<tr>
<td>id</td>
<td>Text</td>
</tr>
<tr>
<td>remove</td>
<td>Text</td>
</tr>
<tr>
<td>nil</td>
<td>Text</td>
</tr>
<tr>
<td>resp</td>
<td>Number</td>
</tr>
<tr>
<td>url</td>
<td>Number</td>
</tr>
<tr>
<td>ipu</td>
<td>Number</td>
</tr>
<tr>
<td>master</td>
<td>Number</td>
</tr>
<tr>
<td>date</td>
<td>Number</td>
</tr>
<tr>
<td>spec</td>
<td>Number</td>
</tr>
<tr>
<td>exist</td>
<td>Number</td>
</tr>
<tr>
<td>eval</td>
<td>Number</td>
</tr>
<tr>
<td>exam</td>
<td>Number</td>
</tr>
<tr>
<td>instrmode</td>
<td>Text</td>
</tr>
<tr>
<td>terms</td>
<td>Text</td>
</tr>
<tr>
<td>college</td>
<td>Text</td>
</tr>
<tr>
<td>day</td>
<td>Text</td>
</tr>
<tr>
<td>year</td>
<td>Text</td>
</tr>
<tr>
<td>level1</td>
<td>Text</td>
</tr>
<tr>
<td>level2</td>
<td>Text</td>
</tr>
<tr>
<td>group_area</td>
<td>Number</td>
</tr>
<tr>
<td>end_date</td>
<td>Number</td>
</tr>
</tbody>
</table>

Field Properties:

- **Field Size**: 4
- **Format**:  
- **Input Mask**:  
- **Caption**:  
- **Default Value**:  
- **Validation Rule**:  
- **Validation Text**:  
- **Required**: No
- **Allow Zero Length**: No
- **Underscore**: Yes (Duplicates OK)
- **Unicode Compression**: Yes
- **BME Mode**: No Control
- **BME Sentence Mode**: None
### Table CBM008

<table>
<thead>
<tr>
<th>Field Name</th>
<th>Data Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>Accno</td>
<td>Text</td>
</tr>
<tr>
<td>Name</td>
<td>Text</td>
</tr>
<tr>
<td>Rank</td>
<td>Text</td>
</tr>
<tr>
<td>Function</td>
<td>Text</td>
</tr>
<tr>
<td>Sex</td>
<td>Text</td>
</tr>
<tr>
<td>Admin</td>
<td>Text</td>
</tr>
<tr>
<td>Ethnic</td>
<td>Text</td>
</tr>
<tr>
<td>Job</td>
<td>Text</td>
</tr>
<tr>
<td>Appr_01</td>
<td>Number</td>
</tr>
<tr>
<td>Appr_02</td>
<td>Number</td>
</tr>
<tr>
<td>Appr_11</td>
<td>Number</td>
</tr>
<tr>
<td>Appr_12</td>
<td>Number</td>
</tr>
<tr>
<td>Appr_13</td>
<td>Number</td>
</tr>
<tr>
<td>Salary_1</td>
<td>Number</td>
</tr>
<tr>
<td>Salary_2</td>
<td>Number</td>
</tr>
<tr>
<td>Salary_3</td>
<td>Number</td>
</tr>
<tr>
<td>Salary_4</td>
<td>Number</td>
</tr>
<tr>
<td>Salary_5</td>
<td>Number</td>
</tr>
<tr>
<td>Salary_6</td>
<td>Number</td>
</tr>
<tr>
<td>Fe</td>
<td>Text</td>
</tr>
<tr>
<td>Compliance</td>
<td>Text</td>
</tr>
<tr>
<td>New_Hire</td>
<td>Text</td>
</tr>
<tr>
<td>Form</td>
<td>Text</td>
</tr>
<tr>
<td>Ref</td>
<td>Text</td>
</tr>
<tr>
<td>Appl_Code</td>
<td>Text</td>
</tr>
<tr>
<td>Term</td>
<td>Text</td>
</tr>
</tbody>
</table>
Table CBM009

<table>
<thead>
<tr>
<th>Field Name</th>
<th>Data Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sex</td>
<td>Text</td>
<td>Sex</td>
</tr>
<tr>
<td>Ethnic</td>
<td>Text</td>
<td>Ethnic origin</td>
</tr>
<tr>
<td>Birth</td>
<td>Number</td>
<td>Birth month</td>
</tr>
<tr>
<td>Birthyear</td>
<td>Number</td>
<td>Birth year</td>
</tr>
<tr>
<td>Degree</td>
<td>Text</td>
<td>Degree earned</td>
</tr>
<tr>
<td>Level</td>
<td>Text</td>
<td>Degree level</td>
</tr>
<tr>
<td>Do</td>
<td>Text</td>
<td>Major Code</td>
</tr>
<tr>
<td>Sem</td>
<td>Text</td>
<td>Semester</td>
</tr>
<tr>
<td>Yr</td>
<td>Number</td>
<td>Year</td>
</tr>
<tr>
<td>Yr_Name</td>
<td>Text</td>
<td>Year name</td>
</tr>
<tr>
<td>Term</td>
<td>Text</td>
<td>Term</td>
</tr>
</tbody>
</table>

Field Properties

- Field Size: 9
- Format: 
- Input Mask: 
- Caption: 
- Default Value: 
- Validation Rule: 
- Validation Text: 
- Required: No
- Allow Zero Length: No
- Indexed: No
- Unicode Compression: Yes

Design View: F6 = Switch panes, F1 = Help.
## Lookup Table Class

<table>
<thead>
<tr>
<th>Field Name</th>
<th>Data Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>class</td>
<td>Text</td>
</tr>
<tr>
<td>level</td>
<td>Text</td>
</tr>
<tr>
<td>class_name</td>
<td>Text</td>
</tr>
</tbody>
</table>

Field Properties:
- Field Size:
- Format:
- Input Mask:
- Caption: Class Code
- Default Value:
- Validation Rule:
- Validation Text:
- Required: No
- Allow Zero Length: No
- Indexed: Yes (No Duplicates)
- Unicode Compression: No

Design view. F6 = Switch panes. F1 = Help.
# Lookup Table College

<table>
<thead>
<tr>
<th>Field Name</th>
<th>Data Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>college</td>
<td>AutoNumber</td>
</tr>
<tr>
<td>college_name</td>
<td>Text</td>
</tr>
</tbody>
</table>

**Field Properties**

- **Field Name**: college
- **Field Size**: Long Integer
- **New Values**: Increment
- **Indexed**: Yes (No Duplicates)

A field name can be up to 64 characters. Design view. F6 = Switch panes. F1 = Help.
## Lookup Table Department

<table>
<thead>
<tr>
<th>Field Name</th>
<th>Data Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>AllName</td>
<td>Text</td>
</tr>
<tr>
<td>dept</td>
<td>Text</td>
</tr>
<tr>
<td>dept_name</td>
<td>Text</td>
</tr>
<tr>
<td>admin_code</td>
<td>Text</td>
</tr>
<tr>
<td>First_term</td>
<td>Text</td>
</tr>
<tr>
<td>final_term</td>
<td>Text</td>
</tr>
</tbody>
</table>

### Field Properties

- **General**
  - Field Size: 3

- **Lookup**
  - Format: 
  - Input Mask: 
  - Caption: 
  - Default Value: 
  - Validation Rule: 
  - Validation Text: 
  - Required: No
  - Allow Zero Length: No
  - Indexed: No

- **A field name can be up to **

---

Design view. F6 = Switch panes. F1 = Help.
## Lookup Table Ethnic

<table>
<thead>
<tr>
<th>Field Name</th>
<th>Data Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>ethnic</td>
<td>Text</td>
</tr>
<tr>
<td>ethnic_name</td>
<td>Text</td>
</tr>
</tbody>
</table>

**Field Properties**

- **Field Size**: 1
- **Format**: 
- **Input Mask**: 
- **Caption**: Ethnic Code
- **Default Value**: 
- **Validation Rule**: 
- **Validation Text**: 
- **Required**: No
- **Allow Zero Length**: No
- **Indexed**: Yes (No Duplicates)
- **Unicode Compression**: No
Lookup Table Gender

<table>
<thead>
<tr>
<th>Field Name</th>
<th>Data Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>gender</td>
<td>Text</td>
</tr>
</tbody>
</table>

Field Properties:
- Field Size: 1
- Format: 
- Input Mask: 
- Caption: 
- Default Value: 
- Validation Rule: "F" Or "M"
- Validation Text: 
- Required: No
- Allow Zero Length: No
- Indexed: Yes (No Duplicates)
- Unicode Compression: No
## Lookup Table Level

<table>
<thead>
<tr>
<th>Field Name</th>
<th>Data Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>ID</td>
<td>AutoNumber</td>
</tr>
<tr>
<td>level1</td>
<td>Text</td>
</tr>
<tr>
<td>level2</td>
<td>Text</td>
</tr>
</tbody>
</table>

### Field Properties

- **Field Size**: Long Integer
- **New Values**: Increment
- **Format**: 
- **Caption**: 
- **Indexed**: Yes (No Duplicates)

*Design view. F6 = Switch panes, F1 = Help.*
## Lookup Table Rank

<table>
<thead>
<tr>
<th>Field Name</th>
<th>Data Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>rank</td>
<td>Text</td>
</tr>
<tr>
<td>rank_name</td>
<td>Text</td>
</tr>
</tbody>
</table>

**Field Properties**

- **General**
  - Field Size: 1
  - Format:
  - Input Mask:
  - Caption:
  - Default Value:
  - Validation Rule:
  - Validation Text:
  - Required: No
  - Allow Zero Length: No
  - Indexed: Yes (No Duplicates)
  - Unicode Compression: Yes

[Design view. F6 = Switch panes. F1 = Help.]
## Lookup Table term_names

<table>
<thead>
<tr>
<th>Field Name</th>
<th>Data Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>Term</td>
<td>Text</td>
</tr>
<tr>
<td>Name</td>
<td>Text</td>
</tr>
<tr>
<td>Date</td>
<td>Date/Time</td>
</tr>
</tbody>
</table>

### Field Properties

- **Field Size**: 5
- **Caption**: Term
- **Validation Rule**: Not set
- **Validation Text**: Not set
- **Required**: No
- **Allow Zero Length**: No
- **Indexed**: Yes (No Duplicates)
- **Unicode Compression**: Yes

*Design view. F4 = Switch panes. F1 = Help.*