

NEAR EAST UNIVERSITY



Faculty Of Engineering

Department Of Computer Engineering

UNIVERSITY INFORMATION SYSTEM

**Graduation Project
COM- 400**

Student: Şeniz Varol-20041170

Supervisor : Ümit İlhan

Nicosia 2008

ACKNOWLEDGEMENTS

"First of all I would like to thank to my supervisor Mr.Umit Ilhan for his invaluable advice and belief in my work and myself over the course of Graduation Project..

I am then very much thankful to Ekrem Dermanel here who always motivated me when ever I got disappointed in any point while making this program.

*I then thank my parents who helped me and supported me till the last moment,
thank you very much.*

Finally ,I would also like to thank all my friends for their advise and support.."

ABSTRACT

The university information system which has been new yet is a system which keeps the whole information about a university's characteristics and presents them to use. This system which is stable unit of university or includes the functions which need to be updated every education term provides an opportunity for most of functions which are often done and it also provides an opportunity to use it handier.

A staff in a university provides the interaction between a teacher and a student. This system makes the functions of association easy and due to this system the information is taken fast and without interruption.

I want to explain the information about .Net which is a very new and unknown technology and presents Microsoft's software. The central component of the .NET platform is the .NET Framework. The .NET Framework Class Library can roughly be broken down into four groups—the Base Class Library (BCL), ADO.NET and XML, Windows Forms, and ASP.NET. The written ASP.Net pages 's functions is more suitable than explaining the written codes. The project includes 53 pages, Microsoft Access Database, config files, css pages and pictures.

TABLE OF CONTENTS

| | |
|--|-----|
| ACKNOWLEDGEMENTS | i |
| ABSTRACT | ii |
| TABLE OF CONTENTS | iii |
| INTRODUCTION | v |
| CHAPTER 1: .NET | 1 |
| 1.1. What Is .NET? | 1 |
| 1.2. Introduction | 1 |
| 1.3. The Vision | 2 |
| 1.4. Web Services | 2 |
| 1.5. The Platform | 3 |
| 1.5.1. The .NET Framework | 9 |
| 1.6. The .Net Framework | 4 |
| 1.7. Common Language Runtime | 4 |
| 1.8. Simplified Deployment | 5 |
| 1.9. Hardware Independence | 5 |
| 1.10. Automatic Memory Management | 6 |
| 1.11. Cross-Language Integration | 6 |
| 1.12. Metadata Driven | 6 |
| 1.13. Enhanced Security | 6 |
| 1.14. Interoperability | 7 |
| 1.15. Class Libraries | 7 |
| 1.16. Development Tools | 8 |
| 1.17. Programming Languages | 8 |
| 1.18. The .NET Framework SDK | 9 |
| 1.19. Visual Studio .NET | 9 |
| 1.20. Foundation Services | 9 |
| 1.21. What Does It All Mean? | 10 |
| CHAPTER 2: ASP.NET | 11 |
| 2. Introduction to ASP.NET | 11 |
| 2.1. What Is ASP.NET? | 11 |
| 2.2. What is an ASP.NET File? | 11 |
| 2.3. Why Do You Need ASP.NET? | 12 |
| 2.4. ASP.NET is Not ASP | 12 |
| 2.5. Framework for Processing HTTP Requests | 13 |
| 2.6. Comparison of Classic ASP and ASP.NET | 14 |
| 2.7. Accepts Requests | 14 |
| 2.8. Builds Responses | 15 |
| CHAPTER 3 : ADO.NET | 16 |
| 3.1. How Does ADO.NET Work? | 16 |
| 3.2. Is OLE DB Dead? | 16 |
| 3.3. Using the ADO.NET Object Model | 19 |
| 3.4. The Connection Object | 19 |
| 3.5. Connecting to a Database | 20 |
| 3.6. TheConnectionString Property | 21 |
| 3.7. Overloading the Connection Object Constructor | 21 |
| 3.8. The Open () Method | 22 |
| 3.9. The Close () Method | 22 |

| | |
|---|----|
| 3.10.The DataAdapter Object | 22 |
| 3.11.Updating a Data Source from a DataSet by Using the DataAdapter | 23 |
| 3.12. Insert Commands by Using the DataAdapter Object with Parameters | 23 |
| 3.13.The DataReader Object | 25 |
| 3.13.1. Creating the DataReader Object | 25 |
| 3.14. The DataSet Object | 25 |
| 3.14.1. Creating the DataSet Object | 27 |
| CHAPTER 4 : PROJECT | 28 |
| 4.The Details of Project | 28 |
| 4.1.The Files Which Are Used : | 28 |
| 4.2.Overview of Project | 31 |
| 4.3. Database | 51 |
| REFERENCES | 52 |
| APPENDIX | 53 |
| default.aspx | 53 |
| hata.aspx | 55 |
| help.aspx | 57 |
| detay_ogrenci.aspx | 64 |
| detay_ogretmen.aspx | 69 |
| ekle_ogrenci.aspx | 74 |
| login.aspx | 80 |
| liste_admin.aspx | 84 |
| CONCLUSION | 93 |

INTRODUCTION

The concept of the university information is originated from researches of the local software of the universities. Thus, the functions are improved by .NET vision. Also The university system is a kind of the technology which is secured, widened .Net.

This project consist of introduction, four chapters and conclusion.

Chapter One describes the definition of .NET. It is determined that .NET software does not depend on Distributed Internet Applications Architecture. .Net Framework provides the internet connections.

Chapter Two presents the ASP.NET. It connects respond and requests very well. ASP .Net uses the same items which the .Net uses.

Chapter Three presents the ADO.NET (ActiveX Data Object). ADO.NET is used to reach the data by the user whose program is improved in .NET. ADO.NET is the center of the DataSet items.

Chapter Four presents the details of University Information System.

Finally, the conclusion section presents the brief summary usage within the project.

CHAPTER 1: .NET

1.1. What Is .NET?

.NET is Microsoft's new initiative for building applications regardless of the platforms or languages in use. The .NET label applies to three distinct but related items: a vision for how information technology (IT) will evolve, a software platform to build .NET applications, and an application-hosting business designed to support the vision and market the platform. In this chapter, we inspect each of these items from a fairly nontechnical perspective.

By the end of the chapter, the readers will have a good idea of where Microsoft is going with the .NET initiative and will understand the terminology, features, and services offered by the .NET Framework, the software platform for .NET applications.

1.2. Introduction

In June 2000, Microsoft announced the .NET initiative - a major shift in the technical direction for Microsoft and a major shift for those engaged in developing software based on Microsoft tools and technologies.

The label .NET has been applied to three distinct entities. They are:

- A vision of how software will evolve to take advantage of the Internet and encompass the increasing variety of computing devices that are joining the PC in customers' offices, pockets, and homes.
- A software platform to help developers build such applications and also to address some long-time shortcomings of Windows.
- An application-hosting business that will deliver applications as services over the Internet.

1.3. The Vision

The Web has evolved a long way from browsing static Hypertext Markup Language (HTML) pages. Today, users can download music, participate in auctions, buy items online, and even talk to their family face-to-face over the Internet. Even businesses are not behind. They have been implementing business-to-business (B2B) and business-to-consumer (B2C) applications that communicate over the Internet.

Microsoft believes that the Internet will evolve from a collection of isolated Web sites and applications into a general “communication bus” for distributed applications. Individual parts of the distributed application could be running on different hardware and software platforms. The computing devices include your desktop systems as well as mobile devices such as cellular phones, Pocket PCs, personal digital assistants (PDAs), and so on. Even household appliances such as microwaves and dishwashers will participate in this communication over the Internet.

1.4. Web Services

To be fair, this vision of anytime, anywhere, any-device computing is also shared by many other software companies, such as IBM and Hewlett-Packard, and many respected computer scientists around the world. A key technology enabler for this distributed computing model is Web services. A *Web service* can be defined as a service that can be accessed programmatically over the Web. Companies can make their business applications available as Web services. These Web services, for example, can be used to integrate applications within various divisions of the same company. The Web services can also be used to automate communication over the Internet between two companies.

To be able to develop distributed applications that transcend geographical, hardware, and OS boundaries, Web services need to be based on universally accepted standards. Table 1.1 lays out the foundation elements of Web services.

| <i>Standard</i> | <i>Purpose</i> |
|--|--|
| Internet | Ubiquitous communication |
| Extensible Markup Language (XML) | Universal data format |
| Simple Object Access Protocol (SOAP) | Communication protocol |
| Web Services Description Language (WSDL) | Describe the semantics of the methods available on a Web service |
| Universal Description, Discovery, and Integration (UDDI) | Publish and find Web services |

Table 1.1. Web Service Foundation

1.5. The Platform

Figure 1.1 shows an overview of the .NET platform. The central component of the .NET platform is the .NET Framework. This consists of a runtime environment called the common language runtime and a set of supporting libraries. The runtime environment controls the installation, loading, and execution of .NET applications. The libraries provide code for common programming tasks, thus increasing developer productivity. The libraries also provide a layer over many OS APIs, providing an isolation from OS dependencies.

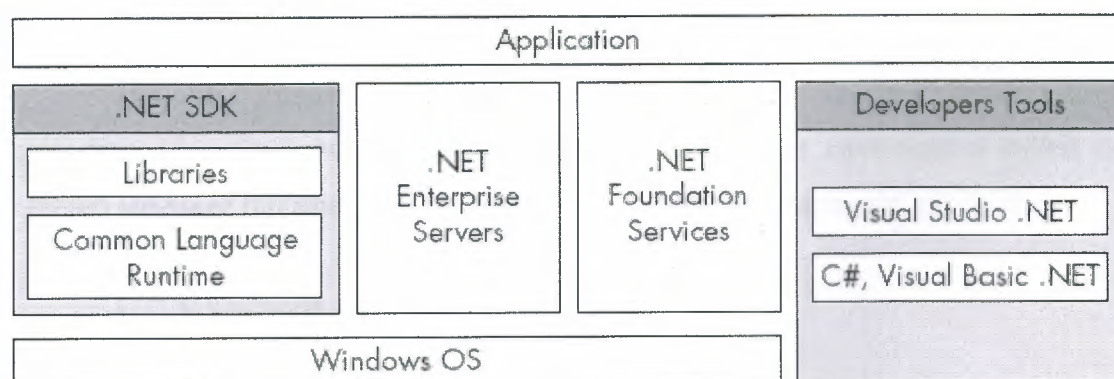


Figure 1.1. The .Net Platform

1.5.1. The .NET Framework 9

Most enterprise applications and Web services require back-end servers to perform operations such as storing data, exchanging messages via e-mail, and so on. Microsoft's family of .NET servers such as SQL Server, Exchange Server, and so on, can be used to obtain such functionality. The family also includes some special servers that provide a higher level of integration and aggregation of Web services. BizTalk

Server and Commerce Server, application frameworks for e-commerce, fall under this category.

The .NET platform also includes a set of developer tools such as Visual Studio .NET and programming languages such as Visual Basic .NET and C# (pronounced *C sharp*).

In developing applications, developers can also take advantage of the foundation services offered by Microsoft or other software vendors. We take a look at a few important foundation services in a later section of this chapter.

Finally, the Windows operating system is at the base of the .NET platform. Operating systems such as Windows NT, Windows 2000, and Windows XP do not come preinstalled with the .NET Framework. However, one can install the framework separately by downloading it from Microsoft's Web site. Windows .NET and the newer releases of the Windows operating system are expected to ship with elements of the .NET vision.

1.6. The .Net Framework

The .NET Framework is a high-productivity, standards-based, multilanguage application execution environment. It consists of a runtime environment called the common language runtime and a set of libraries to boost programmers' productivity.

1.7. Common Language Runtime

The common language runtime, or just the runtime, is Microsoft's implementation of the ECMA CLI specification. When a .NET application is run, the common language runtime manages the loading and execution of the code and provides many runtime services to the application.

If you have been developing your code under COM, you will be surprised by the simplicity the .NET model offers. Forget dealing with GUIDs, CLSIDs, PROGIDs, IDL, type-libraries, apartments, server registration, AddRef(), Release(), and so on. They all have been replaced by a simpler model of programming.

It would not be fair to say that COM is dead. The basic tenet of COM, the ability for applications to communicate across hardware and programming language boundaries, is still present in .NET. In particular, the first release of the .NET Framework still depends on COM+ to provide enterprise services such as transaction and queuing. However, the COM infrastructure has certainly been replaced under .NET.

Besides providing a simpler model of communication, the .NET runtime provides many other services to simplify programming and to develop robust and secure applications. Any code that targets the .NET runtime is called the *managed code*—it can take advantage of the services offered by the .NET runtime. All other code, including existing legacy code, is called the *unmanaged code*. Although, the common language runtime can execute the unmanaged code, the code cannot avail the services provided by the common language runtime. Let's examine some services provided by the common language runtime.

1.8. Simplified Deployment

In the simplest case, the directory hierarchy of an application can be copied to a machine and the application can be executed. There is no need to modify any registry entry. This is referred to as *XCOPY deployment*.

The framework also solves the “DLL hell” problem. A common problem with Windows is that upgrading a dynamic-link library (DLL) routinely breaks an already installed application. Under .NET, two versions of an application can execute side-by-side without breaking any application.

1.9. Hardware Independence

When a .NET application is built, the code is stored in a language called Microsoft Intermediate Language (MSIL). When the application is executed, the runtime performs a just-in-time (JIT) compilation to convert the MSIL code to machine instructions. This makes a .NET application run on any CPU type, as long as a JIT compiler is available for the CPU. Moreover, the JIT compiler can perform hardware-specific optimizations, boosting execution performance.

1.10. Automatic Memory Management

When writing managed code, developers need not worry about memory deallocation issues. The runtime automatically frees any unused memory using a mechanism called *garbage collection*. Not only does this simplify programming, but it also makes the application more robust; as programmers sometimes simply forget to free previously allocated memory.

1.11. Cross-Language Integration

The .NET Framework defines a specification called the Common Language Specification (CLS). Among other things, the CLS defines a set of data types that is intended to work across all .NET-compliant programming languages. If these data types are used, the runtime provides seamless integration between applications developed in different programming languages. The integration is so seamless that a type defined in one language can be inherited in another language. Even exceptions can be thrown from one language and caught in another language.

1.12. Metadata Driven

An application developed for .NET contains complete information about the types it implements, the methods each type defines, the parameters for each method, and so on. The presence of such metadata eliminates the need for COM-style IDL and type libraries. This also makes it possible to keep the Windows registry clean.

1.13. Enhanced Security

.NET defines a permission-based security model that offers fine-grained control over which assembly can access what resource on the local machine. The security becomes especially important when users access code over the Internet. The runtime prevents the executions of any unauthorized code.

1.14. Interoperability

The runtime provides the functionality to integrate with legacy COM servers. The runtime also provides the ability to invoke any unmanaged code or Windows APIs (although such an application may not be portable to other platforms).

1.15. Class Libraries

The .NET Framework also provides hundreds of types (classes, interfaces, structures, etc.) that not only enable programmatic access to the features of the common language runtime, but also provide a number of useful highlevel services to help developers boost their productivity. These types are collectively referred to as the .NET Framework Class Library.

The .NET Framework Class Library can roughly be broken down into four groups—the Base Class Library (BCL), ADO.NET and XML, Windows Forms, and ASP.NET.

The BCL implements the set of functionality that is shared by all the applications targeting the .NET Framework. It defines and implements all the core data types (e.g., string, integer, float, etc.) used by every application.

ADO.NET is the successor to an earlier data access technology called Active Data Object (ADO). ADO.NET provides a set of classes to access and manipulate data. The data is typically obtained from a database and can be converted to XML for better remote manipulation.

Windows Forms (often called WinForms) provide features for developing standard Windows desktop applications. They provide a rich, unified set of controls and drawing functions for all languages, effectively wrapping Windows user interface (UI) APIs in such a way that developers rarely would need to access the Windows APIs directly.

ASP.NET is the successor to a Web-request processing technology called Active Server Pages (ASP). ASP.NET adds two significant enhancements to ASP:

1. It simplifies the process of developing Web services.
2. It provides a model of developing a Web browser-based UI called Web Forms. Controls on the Web Forms run on the server but the UI is displayed on the client browser. This takes lots of coordination and behind-the-scenes activity. However, the end result is Web interfaces that look and behave very much like WinForms interfaces. Moreover, the Web interfaces can deal with a broad range of browsers such as Microsoft Internet Explorer as well as less capable browsers such as the ones found on wireless palmtop devices. WebForms will render themselves appropriately on the target device.

1.16. Development Tools

A productive set of tools is critical to developer success on a new platform like .NET. Microsoft offers many development tools to build Web services as well as traditional Windows applications.

1.17. Programming Languages

.NET offers an improved ball game for programmers. Features such as automatic memory management make it unnecessary for programmers to deal with these issues. The .NET programming model encourages object-oriented programming.

To simplify programming under .NET and to exploit the capability of the .NET Framework to its fullest extent, Microsoft has introduced a new programming language called C#, which offers the simplicity of Visual Basic and the flexibility of C++. C# borrows most of its constructs directly from C++, making C++ and Java programmers feel right at home. More information about the origins of C# can be found in an interview with Anders Hejlsberg [Osb-00], the chief architect for C#.

Microsoft has also introduced Visual Basic .NET, an upgrade to its popular Visual Basic programming language. Visual Basic .NET adds object-oriented features as well as provides full access to .NET platform features. The new feature list of Visual Basic.NET can be found in [Pat-01a] and [Pat-01b].

Microsoft has also extended C++ to develop code for .NET. This extension is referred to as Managed Extension for C++.

Finally, .NET provides an open standard for developing language compilers that target .NET. Many independent software vendors are providing their own programming language support for .NET.

1.18. The .NET Framework SDK

The .NET Framework SDK contains documentation, tools, C# and Visual Basic .NET compilers, and samples for developers to write, build, test, and deploy .NET applications. The SDK also includes the .NET Framework as a redistributable package. The SDK can be downloaded free of charge from Microsoft's Web site. Remember to read the licensing agreement when you download the SDK.

1.19. Visual Studio .NET

Visual Studio .NET is the next generation of Microsoft's popular multilanguage development tool, built especially for .NET. Visual Studio .NET helps you build as well as consume Web services and .NET applications quickly. It supports C#, Visual Basic .NET, and C++. Any other .NET programming language can be easily integrated into Visual Studio .NET. The Integrated Development Editor (IDE) contains many features, such as IntelliSense, to boost programmers' productivity. Visual Studio .NET will likely remain the most popular choice for developing .NET applications.

1.20. Foundation Services

Microsoft also envisions that providing a compelling user experience to consumers is important for the success of Internet as a communication bus. To this effect, Microsoft plans to release some foundation or building block Web services. Software vendors can leverage against these foundation services. With time, Microsoft intends to release more such foundation services.

The first set of foundation services are being released as a product called Microsoft .NET My Services. Table 1.2 lists these services.

| <i>Name</i> | <i>Description</i> |
|--------------------------|---|
| .NET Address | Billing, shipping, and other addresses |
| .NET Profile | Name, picture, and so on |
| .NET Contacts | Electronic address book |
| .NET Location | Electronic and geographical locations |
| .NET Alerts | Send and/or electronic notifications |
| .NET Inbox | E-mail and voice-mail storage |
| .NET Calendar | Appointment management |
| .NET Documents | Users can store, share, and back up important files |
| .NET ApplicationSettings | Application settings |
| .NET FavoriteWebSites | List of favorite Web sites |
| .NET Wallet | Credit card information, coupons, receipts, and so on |
| .NET Devices | Settings for various personal devices |
| .NET Services | List of services provided |
| .NET Usage | Usage report for the preceding services |

Table 1.2. .Net Foundation Services

1.21. What Does It All Mean?

Microsoft's .NET initiative impacts consumers, businesses, software vendors, and developers.

Consumers will be the biggest beneficiaries of .NET and the foundation services. As the data is stored on the Web, they will be able to access documents and other personal information anytime, anywhere, from any smart device.

For businesses, implementing applications using Web services solves many of today's B2B and B2C integration challenges.

By making their business applications available as Web services, or by providing innovative Web services, software vendors may be able to find newer modes of revenue. Microsoft itself is gravitating toward providing a subscription-based model for its services, thereby ensuring a monthly source of revenue.

The .NET Framework helps developers write robust, secure, Internet-enabled code in a language of their choice. The rich set of class libraries provided by the .NET Framework, as well as new features in Visual Studio .NET will boost developer productivity.

CHAPTER 2: ASP.NET

2. Introduction to ASP.NET

ASP.NET is the .NET framework layer that handles Web requests for specific types of files, namely those with .aspx and .ascx extensions. The ASP.NET engine provides a robust object model for creating dynamic content and is loosely integrated into the .NET framework. This integration makes it easy to change the implementation when the .NET framework migrates to platforms other than Windows.

ASP.NET is the latest version of Microsoft's Active Server Pages technology (ASP).

- WWW, HTML, XML and the basics of building Web pages
- Scripting languages like JavaScript or VBScript
- The basics of server side scripting like ASP or PHP

2.1. What Is ASP.NET?

ASP.NET is a server side scripting technology that enables scripts (embedded in web pages) to be executed by an Internet server.

- ASP.NET is a Microsoft Technology
- ASP stands for Active Server Pages
- ASP.NET is a program that runs inside IIS
- IIS (Internet Information Services) is Microsoft's Internet server
- IIS comes as a free component with Windows servers
- IIS is also a part of Windows 2000 and XP Professional

2.2. What is an ASP.NET File?

- An ASP.NET file is just the same as an HTML file
- An ASP.NET file can contain HTML, XML, and scripts
- Scripts in an ASP.NET file are executed on the server
- An ASP.NET file has the file extension ".aspx"

2.3. Why Do You Need ASP.NET?

The first computer languages were little more than mnemonics substituting for raw machine code instructions, but as computers became more complex, each new language generation has supported an increasing level of abstraction. Visual Basic, for example, abstracted user interface design and construction into simple drag-and-drop operations. For the first time, you could create a working Windows application with very little effort.

Similarly, when Web programming first became widespread, there were few tools to help programmers write Web applications. To create a Web application, you started by writing low-level socket communications code. Over the years, the abstraction level has increased for Web programming as well. ASP.NET is the latest (and arguably the best) of these abstractions, because it lets you work

almost exclusively with rich high-level classes and objects rather than directly with raw data. Without ASP.NET, building a Web application is a chore. With ASP.NET, building a Web application is similar to building a Win32 application.

2.4. ASP.NET is Not ASP

ASP.NET is the next generation ASP, but it's not an upgraded version of ASP. ASP.NET is an entirely new technology for server-side scripting. It was written from the ground up and is not backward compatible with classic ASP.

ASP.NET is the major part of the Microsoft's .NET Framework

Unlike classic ASP, ASP.NET uses .NET languages. Therefore, you have access to the full power of any .NET assembly or class in exactly the same way as you do from VB.NET. In this sense, ASP.NET is similar to early compiled CGI programs, but with CGI, a separate copy of the program had to be loaded and executed for each request. ASP.NET code exists in multithreaded JIT compiled DLL assemblies, which can be loaded on demand. Once loaded, the ASP.NET DLLs can service multiple requests from a single in-memory copy.

ASP.NET supports all the .NET languages (currently C#, C++, VB.NET, and JScript, but there are well over 20 different languages in development for .NET), so you will eventually be able to write Web applications in your choice of almost any modern programming language. JavaServer pages support only Java, but because Java now has a wide support base, that's not much of a limitation. Classic ASP supports several scripting language versions (although in practice, VBScript and JScript are by far the most prevalent). The scripting languages let you extend ASP's basic functionality by writing DLLs in any COM-compliant language. Cold-Fusion uses ColdFusion Markup Language (CFML) tags, which have a powerful but limited set of capabilities; however, you can extend CFML with custom programming.

Microsoft was able to draw on millions of hours of developer experience with classic ASP, so in addition to huge increases in speed and power, ASP.NET provides substantial development improvements, like seamless server-to-client debugging, automatic validation of form data, and a programming model very similar to that of a Windows application.

In classic ASP, the server handed off file requests that ended in .asp to the ASP engine, an Internet Server Application Programming Interface (ISAPI) ASP DLL. Because there's a difference in the file extension (.asp versus .aspx, .ascx, and .asmx) for classic ASP and ASP.NET files, respectively, you can have both running on the same server simultaneously.

2.5. Framework for Processing HTTP Requests

Microsoft's Web server, Internet Information Server (IIS), handles HTTP requests by handing the request off to the appropriate module based on the type of file requested. Note that the IIS responds with one of only a few possible actions when it receives a request:

Respond with the file's contents The server locates and reads the requested file's contents and then streams the contents back to the requester. The server responds in this manner to .htm and .html file requests, as well as to all requests that have no associated application type—for example, EXE files.

Respond by handing off the request The server hands off requests for files that end in .asp to the classic ASP processor, and files that end in .aspx, .ascx, or .asmx to the ASP.NET processor.

Respond with an error IIS responds with a customizable error message when a requested file does not exist or when an error occurs during processing.

2.6. Comparison of Classic ASP and ASP.NET

| CLASSIC ASP | ASP.NET | DESCRIPTION |
|---|--|---|
| Intercept client requests for files with an .asp extension. | Intercept client requests for files with the .aspx extension. | Provides the ability to create content "on-the-fly"—dynamic content. |
| Write server-side script in one of a small number of languages. Script languages are interpreted at runtime. | Write server-side code in any .NET language. .NET languages are compiled, not interpreted. | Compiled code is faster. The development environments and debug facilities are more powerful. |
| Extend ASP scripting functionality with COM objects. | Use any of the .NET System classes or call existing COM objects. | Provides the ability to extend ASP capabilities by writing custom code. |
| All processing happens <i>after</i> the server passes control to the ASP engine. Cannot take advantage of ISAPI services. | You can write code to intercept requests <i>before</i> the ASP engine takes control. You can write ISAPI services within the .NET framework. | Sometimes, you want to respond to a request <i>before</i> the ASP engine parses the request. You can do that in .NET, but not with classic ASP. |
| Code and HTML are usually mixed in-line within a page. | Code may be placed in-line in ASP.NET pages, but is usually separated from the HTML in "code-behind" files. | The .NET code-behind pages provide a cleaner separation of display and logic code and also simplify code reuse. |
| Developer responsible for implementing ways to maintain state data between pages. | Web Forms and Web Form controls act much like classic VB forms and controls, with properties and methods for retrieving and setting values. | While both classic ASP and ASP.NET render output in HTML, ASP.NET introduces ViewState, a scheme that automatically maintains the state of controls on a page across round trips to the server. Web Forms, Web Form controls, and ViewState simplify development and eliminate much of the gap between programming Web applications and stand-alone Windows applications. |
| Process submitted HTML form fields. | Process and validate submitted form fields. | Provides the ability to gather user input. Automatic validation takes much of the grunt work out of programming pages that require user input. |

2.7. Accepts Requests

All ASP.NET pages work essentially the same way. A client application makes an HTTP request to a Web server using a URL. The Web server hands off the request to the ASP.NET processor, which parses the URL and all data sent by the client into

collections of named values. ASP.NET exposes these values as properties of an object called the `HttpRequest` object, which is a member of the `System.Net` assembly. An assembly is a collection of classes. Although an assembly *can* be a DLL, it may consist of more than one DLL. Conversely, a single DLL may contain more than one assembly. For now, think of an assembly as a group of related classes.

When a browser, or more properly a *user agent*, makes a request, it sends a string containing type and version information along with the request. You can retrieve the `HTTP_USER_AGENT` string via the `HttpRequest` object. For example, the following code fragment retrieves several items from the user agent and writes them back to the client. An ASP.NET Web Form Page object exposes the `Http-Request` with the shorter (and familiar to ASP).

```
Response.Write("UserAgent=" & Request.UserAgent & "<br>")
```

```
Response.Write("UserHostAddress=" & Request.UserHostAddress & "<br>")
```

```
Response.Write("UserHostName=" & Request.UserHostName & "<br>")
```

2.8.Builds Responses

Just as ASP.NET abstracts incoming data in the `HttpRequest` object, it provides a way to respond to the request via the `HttpResponse` object. Abstracting responses in this manner has been so successful that you'll find you need to know almost nothing about HTTP itself to use the `HttpRequest` and `HttpResponse` objects.

CHAPTER 3 : ADO.NET

3.1. How Does ADO.NET Work?

ADO.NET base classes enable you to manipulate data from many data sources, such as SQL Server, Exchange, and Active Directory. ADO.NET leverages .NET data providers to connect to a database, execute commands, and retrieve results.

The ADO.NET object model exposes very flexible components, which in turn expose their own properties and methods, and recognize events. In this chapter, you'll explore the objects of the ADO.NET object model and the role of each object in establishing a connection to a database and manipulating its tables.

3.2. Is OLE DB Dead?

Not quite. Although you can still use OLE DB data providers with ADO.NET, you should try to use the managed .NET data providers whenever possible. If you use native OLE DB, your .NET code will suffer because it's forced to go through the COM interoperability layer in order to get to OLE DB. This leads to performance degradation. Native .NET providers, such as the System.Data.SqlClient library, skip the OLE DB layer entirely, making their calls directly to the native API of the database server.

However, this doesn't mean that you should avoid the OLE DB .NET data providers completely. If you are using anything other than SQL Server 7 or 2000, you might not have another choice. Although you will experience performance gains with the SQL Server .NET data provider, the OLE DB .NET data provider compares favorably against the traditional ADO/OLE DB providers that you used with ADO 2.x. So don't hold back from migrating your non-managed applications to the .NET Framework for performance concerns. In addition, there are other compelling reasons for using the OLE DB .NET providers. Many OLE DB providers are very mature and support a great deal more functionality than you would get from the newer SQL Server .NET data provider, which exposes only a subset of this full functionality. In addition, OLE DB is still the way to go for universal data access across disparate data sources. In fact, the SQL Server distributed process relies on OLE DB to manage joins across heterogeneous data sources.

Another caveat to the SQL Server .NET data provider is that it is tightly coupled to its data source. Although this enhances performance, it is somewhat limiting in terms of portability to other data sources. When you use the OLE DB providers, you can change the connection string on the fly, using declarative code such as COM+ constructor strings. This loose coupling enables you to easily port your application from an SQL Server back-end to an Oracle back-end without recompiling any of your code, just by swapping out the connection string in your COM+ catalog.

Keep in mind, the only native OLE DB provider types that are supported with ADO.NET are SQLOLEDB for SQL Server, MSDAORA for Oracle, and Microsoft.Jet.OLEDB.4 for the Microsoft Jet engine. If you are so inclined, you can write your own .NET data providers for any data source by inheriting from the System.Data namespace.

At this time, the .NET Framework ships with only the SQL Server .NET data provider for data access within the .NET runtime. Microsoft expects the support for .NET data providers and the number of .NET data providers to increase significantly. (In fact, the ODBC.NET data provider is available for download on Microsoft's website.) A major design goal of ADO.NET is to synergize the native and managed interfaces, advancing both models in tandem.

You can find the ADO.NET objects within the System.Data namespace. When you create a new VB .NET project, a reference to the System.Data namespace will be automatically added for you, as you can see in Figure 6.1.

To use ADO.NET, reference the System.Data namespace.

To comfortably use the ADO.NET objects in an application, you should use the Imports statement. By doing so, you can declare ADO.NET variables without having to fully qualify them. You could type the following Imports statement at the top of your solution:

```
Imports System.Data.SqlClient
```

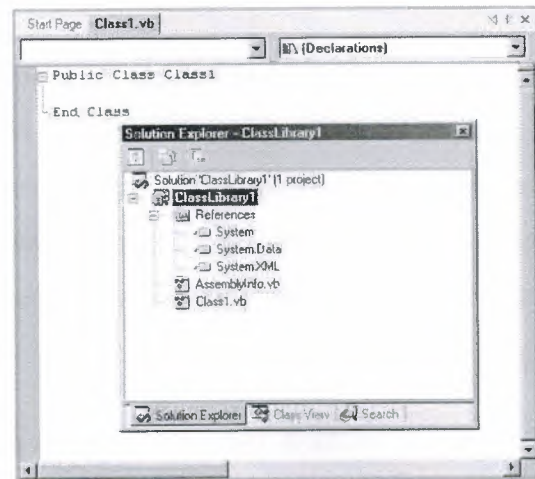
After this, you can work with the SqlConnection ADO.NET objects without having to fully qualify the class names. If you want to dimension the SqlDataAdapter, you would type the following short declaration:

```
Dim dsMyAdapter as New SqlDataAdapter
```

Otherwise, you would have to type the full namespace, as in:

```
Dim dsMyAdapter as New System.Data.SqlClient.SqlDataAdapter
```

Alternately, you can use the visual database tools to automatically generate your ADO.NET code for you. As you saw in Chapter 3, "The Visual Database Tools," the various wizards that come with VS .NET provide the easiest way to work with the ADO.NET objects. Nevertheless, before you use these tools to build production systems, you should understand how ADO.NET works program-matically. In this chapter, we don't focus too much on the visual database tools, but instead concentrate on the code behind the tools. By understanding how to program against the ADO.NET object model, you will have more power and flexibility with your data access code.



3.3. Using the ADO.NET Object Model

You can think of ADO.NET as being composed of two major parts: .NET data providers and data storage. Respectively, these fall under the connected and disconnected models for data access and presentation. *.NET data providers*, or *managed providers*, interact natively with the database. Managed providers are quite similar to the OLE DB providers or ODBC drivers that you most likely have worked with in the past.

The .NET data provider classes are optimized for fast, read-only and forward-only retrieval of data. The managed providers talk to the database by using a fast data stream (similar to a file stream). This is the quickest way to pull read-only data off the wire, because you minimize buffering and memory overhead.

If you need to work with connections, transactions, or locks, you would use the managed providers, not the DataSet. The DataSet is completely disconnected from the database and has no knowledge of transactions, locks, or anything else that interacts with the database.

Five core objects form the foundation of the ADO.NET object model, as you see listed in Table 6.1. Microsoft moves as much of the provider model as possible into the managed space. The Connection, Command, DataReader, and DataAdapter belong to the .NET data provider, whereas the DataSet is part of the disconnected data storage mechanism.

| OBJECT | DESCRIPTION |
|-------------|--|
| Connection | Creates a connection to your data source |
| Command | Provides access to commands to execute against your data source |
| DataReader | Provides a read-only, forward-only stream containing your data |
| DataSet | Provides an in-memory representation of your data source(s) |
| DataAdapter | Serves as an ambassador between your DataSet and data source, proving the mapping instructions between the two |

3.4. The Connection Object

Both the `SqlConnection` and `OleDbConnection` namespaces inherit from the `IDbConnection` object. The `Connection` object establishes a connection to a database, which is then used to execute commands against the database or retrieve a `DataReader`. You use the `SqlConnection` object when you are working with SQL Server, and the `OleDbConnection` for all other data sources. The `ConnectionString` property is the most important property of the `Connection` object. This string uses name-value pairs to specify the database you want to connect to. To establish a connection through a `Connection` object, call its `Open()` method. When you no longer need the connection, call the `Close()` method to close it. To find out whether a `Connection` object is open, use its `State` property.

3.5. Connecting to a Database

The first step to using ADO.NET is to connect to a data source, such as a database. Using the `Connection` object, you tell ADO.NET which database you want to contact, supply your username and password (so that the DBMS can grant you access to the database and set the appropriate privileges), and, possibly, set more options. The `Connection` object is your gateway to the database, and all the operations you perform against the database must go through this gateway. The `Connection` object encapsulates all the functionality of a data link and has the same properties. Unlike data links, however, `Connection` objects can be accessed from within your VB .NET code. They expose a number of properties and methods that enable you to manipulate your connection from within your code.

Let's experiment with creating a connection to the Northwind database. Create a new Windows Application solution and place a command button on the Form; name it **Connect to Northwind**. Add the Imports statement for the `System.Data.SqlClient` name at the top of the form module. Now you can declare a `Connection` object with the following statement:

```
Dim connNorthwind As New SqlConnection()
```

As soon as you type the period after `SqlConnection`, you will see a list with all the objects exposed by the `SqlConnection` component, and you can select the one you want with the arrow keys. Declare the `connNorthwind` object in the button's click event.

3.6. The `ConnectionString` Property

The `ConnectionString` property is a long string with several attributes separated by semicolons. Add the following line to your buttons click event to set the connection:

```
connNorthwind.ConnectionString="data source=(local);"& "initial  
catalog=Northwind;integrated security=SSPI;"
```

Replace the data source value with the name of your SQL Server, or keep the local setting if you are running SQL Server on the same machine. If you aren't using Windows NT integrated security, then set your user ID and password like so:

```
connNorthwind.ConnectionString="data source=(local);"& "initial  
catalog=Northwind; user ID=sa;password=xxx"
```

TIP Some of the names in the connection string also go by aliases. You can use `Server` instead of `data source` to specify your SQL Server. Instead of `initial catalog`, you can specify `database`.

Those of you who have worked with ADO 2.x might notice something missing from the connection string: the provider value. Because you are using the `SqlClient` namespace and the .NET Framework, you do not need to specify an OLE DB provider. If you were using the `OleDb` namespace, then you would specify your provider name-value pair, such as `Provider=SQLOLEDB`

3.7. Overloading the Connection Object Constructor

One of the nice things about the .NET Framework is that it supports constructor arguments by using overloaded constructors. You might find this useful for creating your ADO.NET objects, such as your database Connection. As a shortcut, instead of using the `ConnectionString` property, you can pass the string right into the constructor, as such:

```
Dim connNorthwind as New SqlConnection _
```

```
("data source=localhost; initial catalog=Northwind; user ID=sa;password=xxx")
```

Or you could overload the constructor of the connection string by using the following:

```
Dim myConnectionString As String = "data source=localhost; initial  
catalog=Northwind; user ID=sa;password=xxx"
```

3.8. The Open () Method

After you have specified the `ConnectionString` property of the `Connection` object, you must call the `Open()` method to establish a connection to the database. You must first specify the `ConnectionString` property and then call the `Open()` method without any arguments, as shown here (`connNorthwind` is the name of a `Connection` object):

```
connNorthwind.Open()
```

3.9. The Close () Method

Use the `Connection` object's `Close()` method to close an open connection. Connection pooling provides the ability to improve your performance by reusing a connection from the pool if an appropriate one is available. The `OleDbConnection` object will automatically pool your connections for you. If you have connection pooling enabled, the connection is not actually released, but remains alive in memory and can be used again later. Any pending transactions are rolled back.

3.10. The DataAdapter Object

The `DataAdapter` represents a completely new concept within Microsoft's data access architecture. The `DataAdapter` gives you the full reign to coordinate between your in-memory data representation and your permanent data storage source. In the OLE DB/ADO architecture, all this happened behind the scenes, preventing you from specifying how you wanted your synchronization to occur.

The `DataAdapter` object works as the ambassador between your data and data-access mechanism. Its methods give you a way to retrieve and store data from the data source and the `DataSet` object. This way, the `DataSet` object can be completely agnostic of its data source.

The DataAdapter also understands how to translate *deltagrams*, which are the DataSet changes made by a user, back to the data source.

The DataAdapter implicitly works with Connection objects as well, via the Command object's interface. Besides explicitly working with a Connection object, this is the only other way you can work with the Connection object.

The DataAdapter object is very "polite," always cleaning up after itself. When you create the Connection object implicitly through the DataAdapter, the DataAdapter will check the status of the connection. If it's already open, it will go ahead and use the existing open connection. However, if it's closed, it will quickly open and close the connection when it's done with it, courteously restoring the connection back to the way the DataAdapter found it.

3.11.Updating a Data Source from a DataSet by Using the DataAdapter

The DataAdapter uses the Update() method to perform the relevant SQL action commands against the data source from the deltagram in the DataSet.

3.12. Insert Commands by Using the DataAdapter Object with Parameters

```
Dim strSelectCustomers As String = "SELECT * FROM Customers ORDER BY CustomerID"
Dim strConnString As String = "data source=(local);" & _
```

"initial catalog=Northwind;integrated security=SSPI;" ' We can't use the implicit connection created by the ' DataSet since our update command requires a ' connection object in its constructor, rather than a ' connection string

```
Dim connNorthwind As New SqlConnection(strConnString) ' String to update the customer
record - it helps to ' specify this in advance so the CommandBuilder doesn't ' affect our performance at
runtime Dim strInsertCommand As String = _
```

```
"INSERT INTO Customers(CustomerID,CompanyName) VALUES (@CustomerID,
@CompanyName)"
```

```

Dim daCustomers As New SqlDataAdapter() Dim dsCustomers As New DataSet() Dim
cmdSelectCustomer As SqlCommand = New SqlCommand _

(strSelectCustomers, connNorthwind) Dim cmdInsertCustomer As New
SqlCommand(strInsertCommand, connNorthwind) daCustomers.SelectCommand = cmdSelectCustomer
daCustomers.InsertCommand = cmdInsertCustomer connNorthwind.Open()

daCustomers.Fill(dsCustomers, "dtCustomerTable") cmdInsertCustomer.Parameters.Add _

(New SqlParameter _

("@CustomerID", SqlDbType.NChar, 5)).Value = "ARHAN"
cmdInsertCustomer.Parameters.Add _

(New SqlParameter _ ("@CompanyName", SqlDbType.VarChar, 40)).Value = "Amanda
Aman Apak Merkez Inc." cmdInsertCustomer.ExecuteNonQuery() connNorthwind.Close()

```

In summary, you work with a DataAdapter by using the following steps:

1. INSTANTIATE YOUR DATAADAPTER OBJECT.
2. SPECIFY THE SQL STATEMENT OR STORED PROCEDURE FOR THE SELECTCOMMAND OBJECT. THIS IS THE ONLY COMMAND OBJECT THAT THE DATAADAPTER REQUIRES.
3. SPECIFY THE APPROPRIATE CONNECTION STRING FOR THE SELECTCOMMAND'S CONNECTION OBJECT.
4. SPECIFY THE SQL STATEMENTS OR STORED PROCEDURES FOR THE INSERTCOMMAND, UPDATECOMMAND, AND DELETECOMMAND OBJECTS. ALTERNATELY, YOU COULD USE THE COMMANDBUILDER TO DYNAMICALLY MAP YOUR ACTIONS AT RUNTIME. THIS STEP IS NOT REQUIRED.
5. CALL THE FILL() METHOD TO POPULATE THE DATASET WITH THE RESULTS FROM THE SELECTCOMMAND OBJECT.
6. IF YOU USED STEP 4, CALL THE APPROPRIATE EXECUTE() METHOD TO EXECUTE YOUR COMMAND OBJECTS AGAINST YOUR DATA SOURCE

3.13. The DataReader Object

The DataReader object is a fast mechanism for retrieving forward-only read-only streams of data. The SQL Server .NET provider have completely optimized this mechanism, so use it as often as you can for fast performance of read-only data. Unlike ADO RecordSets, which force you to load more in memory than you actually need, the DataReader is a toned-down, slender data stream, using only the necessary parts of the ADO.NET Framework. You can think of it as analogous to the server-side, read-only, forward-only cursor that you used in native OLE DB/ADO. Because of this server-side connection, you should use the DataReader cautiously, closing it as soon as you are finished with it. Otherwise, you will tie up your Connection object, allowing no other operations to execute against it (except for the Close() method, of course).

As we mentioned earlier, you can create a DataReader object by using the ExecuteReader() method of the Command object. You would use DataReader objects when you need fast retrieval of read-only data, such as populating combo-box lists.

3.13.1. Creating the DataReader Object

```
Dim strCustomerSelect as String = "SELECT * from Customers"
```

```
Dim cmdCustomers as New SqlCommand(strCustomerSelect, connNorthwind)
```

```
Dim drCustomers as SqlDataReader
```

```
connNorthwind.Open()
```

```
drCustomers = cmdCustomers.ExecuteReader()
```

3.14. The DataSet Object

There will come a time when the DataReader is not sufficient for your data manipulation needs. If you ever need to update your data, or store relational or hierarchical data, look no further than the DataSet object. Because the DataReader navigation mechanism is linear, you have no way of traversing between relational or hierarchical data structures. The DataSet provides a liberated way of navigating through

both relational and hierarchical data, by using array-like indexing and tree walking, respectively.

Unlike the managed provider objects, the DataSet object and friends do not diverge between the OleDb and SqlClient .NET namespaces. You declare a DataSet object the same way regardless of which .NET data provider you are using:

```
Dim dsCustomer as DataSet
```

Realize that DataSets stand alone. A DataSet is not a part of the managed data providers and knows nothing of its data source. The DataSet has no clue about transactions, connections, or even a database. Because the DataSet is data source agnostic, it needs something to get the data to it. This is where the DataAdapter comes into play. Although the DataAdapter is not a part of the DataSet, it understands how to communicate with the DataSet in order to populate the DataSet with data.

There are three main ways to populate a DataSet:

- ◆ After establishing a connection to the database, you prepare the DataAdapter object, which will retrieve your results from your database as XML. You can use the DataAdapter to fill your DataSet.

- ◆ You can read an XML document into your DataSet. The .NET Framework provides an XMLDataDocument namespace, which is modeled parallel to the ADO.NET Framework. You will explore this namespace in Chapter 7.

- ◆ You can use DataTables to build your DataSet in memory without the use of XML files or a data source of any kind. You will explore this option in the section “Updating Your Database by Using DataSets” later in this chapter.

Lets work with retrieving data from the Northwind database. First, you must prepare the DataSet object, which can be instantiated with the following statement:

```
Dim dsCustomers As New DataSet()
```

Assuming you’ve prepared your DataAdapter object, all you would have to call is the Fill() method.

3.14.1. Creating the DataSet Object

```
Dim strSelectCustomers As String = "SELECT * FROM Customers ORDER BY CustomerID"  
Dim strConnString As String = "data source=(local);" & _
```

```
"initial catalog=Northwind;integrated security=SSPI;" Dim daCustomers As New  
SqlDataAdapter(strSelectCustomers, strConnString) Dim dsCustomers As New DataSet() Dim  
connNorthwind As New SqlConnection(strConnString)
```

```
daCustomers.Fill(dsCustomers, "dtCustomerTable") MsgBox(dsCustomers.GetXml, , "Results  
of Customer DataSet in XML")
```

In summary, all you have to do is execute the following steps to commit updates to your DataSet:

1. INSTANTIATE YOUR DATASET AND DATAADAPTER OBJECTS.
2. FILL YOUR DATASET OBJECT FROM THE DATAADAPTER OBJECT.
3. MANIPULATE YOUR DATASET BY USING THE DATAROW OBJECTS.
4. CALL THE ACCEPTCHANGES() METHOD OF THE DATASET, DATATABLE, OR DATAROW OBJECT TO COMMIT YOUR CHANGES TO YOUR DATASET.

CHAPTER 4 : PROJECT

4.The Details of Project

If the details of project and the functions which are done in project is needed to explain step by step , explaining the written ASP.Net pages 's functions is more suitable than explaining the written codes.The project includes 53 pages,Microsoft Access Database,config files,css pages and pictures.If we examine the files and their functions one by one:

4.1.The Files Which Are Used :

- **main.css** : The main.css files which is at the bottom of style file includes the functions and colors which are used.Css pages which includes `<link href="style/main.css" rel ="stylesheet" type ="text/css">` code in pages provides to give table background colours,writing,colours,and firms(shapes).Qualities of forms (shapes)are given by the means of the quality which is written "class" in codes.

- **web.config** : If someone does not want to expose to the problem of Turkish letters in ASP.Net pages,web.config file which is defined Turkish letters should be in the main file.If we don't have this file,we don't have any alternative ağırt from defined Turkish letters which is in pages of hosting firm's server.(Most of server is defined as UTF-8 and the problem of Turkish letters is often come out.)

- **top.aspx** : This file included in pages has necessary links which has university information logo and fast communication.Links which are shown as site administer introduction (entrance)are changeable.(include file).

- **bottom.asp** : Again,bottom.aspx pages are included all pages like top.aspx pages.There are links which provide staff's log in and log out.(include file)

- **default.aspx** : It is the main page of system. It gives preknowledge to visitors about information system and how to work.It has also functions of direction due to links.

- **login.aspx** : It is the pages of user's log in.The user can be either a student or a teacher.The user can log in using his own number and password.The password is valid during the user's online and it is also valid until closing the page and clicking the log out. Due to this password the user can see all details about himself and

the user can do the whole procedure. The teacher can see the lesson which he gives that term. Also, if the term is suitable, he can prepare the lesson Schedule. The teacher has the permission of seeing or changing the details of students. If the student log in, he can see his lessons, and marks. If the term is suitable, he can select the lessons whichever he wants. (important: while logging in the button of logging in should be clicked with Mouse.)

- **login1.aspx** : It is the page which is done administrative logging in. Administrative staff's passwords which belong to every faculty are defined. The staff selects the suitable faculty and password. After doing this, he has all rights about the faculty.

- **liste_admin.aspx(*)**: This page is only used by the system administrator. The password which belong to the administration are determined in liste_admin.aspx page. The passwords which the administration use are saved, changed or deleted in this page. (liste_admin.aspx). Lots of passwords can be given just for one faculty.

- **liste_donem.aspx(*)** : This page is only used by the system administrator, too. There is information of education in this page. This page is determined between whichever term it is or whether it belongs to fall or spring term. You can enter new terms. The registered terms can be changed or deleted.

- **liste_gun.aspx(*)** : The System administrator can use this page, too. There are lots of registrations about the days which the university has lessons while preparing the lesson Schedules, these days are kept in mind.

- **liste_saat.aspx**: the system administrator can use this page, too. Lesson duration is saved as hour :minute.

- **logout.aspx** : Logging out is done in logout.aspx page. Students, teachers, administrative staff and system administrator use this file to log out and they are directed to the main page.

- **liste_fakulte.aspx** : Even if the main page of the system is default.aspx, there are beginning of the system definitions and its branches in this page. Liste_fakulte.aspx determine the faculties and campuses of university by taking from the database. The departments which belong to faculties and the teachers of that faculties can be found. If logging in of system administrator is done, a new faculty can be added or the faculty which is already saved can be changed or deleted.

- **liste_bolum.aspx** : There are lots of selected departments which belong to a faculty and there are also lots of information about these departments. The classes which belong to the departments or the lessons which are given in that department are found in liste_bolum.aspx page. If logging in of system administrator or logging in of the stuff of that faculty is done, a new department can be added to the new faculty and departments can be changed or deleted.

- **liste_ders.aspx** : It is the list of the department lessons. It shows that which term the lessons are given and it also shows the credits of lessons.

- **liste_sinif.aspx** : The classes in liste_bolum.aspx page which belong to the selected department are shown in that page. The class list can also be found in that page.

- **liste_ogrenci.aspx** : The list which belongs to the class is shown in liste_ogrenci.aspx page. The student numbers, names, surnames and lesson schedule can be found in this page. If the student log in, he can see his own details. But he can see other student's details. If the system administrator log in or the staff log in, the student in list can be deleted or a new student can be added.

- **liste_ogretmen.aspx** : It lists the teachers of faculty. The teacher's code, name and position are determined. The teacher can see only his own details and information like the student list page in liste_ogretmen.aspx. If the system administrator or the administrative staff which belongs to the faculty logs in, the teacher can be deleted or a new teacher can be added to this faculty.

- **detay_ogrenci.aspx** : Information about students is shown in detay_ogrenci.aspx. Lesson Marks, lesson schedules are directed in this page.

- **detay_ogretmen.aspx** : Information about the teacher is shown in detay_ogrenci.aspx. The lessons which are taken and lesson Schedule are directed in this page.

- **ekle_ogretmen.aspx** : A new teacher is defined to the faculty in this page. Information about the teacher is registered in this page (only, the system administrator or the administrative faculty can reach the information)

- **guncelle_ogrenci.aspx** : The student which is registered in advanced can be changed in this page. (Only the system administrator or the administrative faculty can reach the information)

- **guncelle_ogretmen.aspx** : The student which is registered in advanced can be changed in this page.(only the system administrator or the administrative faculty can reach the information)

- **ogretmen_ders.aspx** : The lessons which the teacher gives in that term are listed and determined in this page. The lessons can not be determined by the teachers. The system administrator or faculty does it. The users can see the list and check the students which take the lessons.

- **ogretmen_ders_ogrenci.aspx** : The students who attend the lesson which the teacher gives are shown in this page. If there is not any logging in, the students's marks and their attendance can be observed, changed or added in the page of the student's number and name.

- **ogretmen_program.aspx** : The teacher can see the lesson Schedule. If the term is suitable (the lesson registration is containing), the teacher can prepare a new Schedule and he can change the lesson Schedule and he can change the lesson Schedule which is done before.

- **ogrenci_ders.aspx** : The lessons which the student take is shown in this page. If the term is suitable like the page of **ogretmen_program.aspx**, the student can choose the lessons in this page. He can choose the lesson according to the teacher who give the lesson or the Schedule which the teacher prepare. If the lesson registration is containing, he can choose the lesson he wants and he can change.

- **ogrenci_program.aspx** : The automatic lesson Schedule is prepared according to the student who choose the lessons. Because, that lesson which is given by the teacher is determined which day it will be or what time it will be. According to this criteria, the unit plan of the student is shown in this page.

- **ogrenci_ders_program.aspx** : The Marks and attendance of the students are shown in this page.

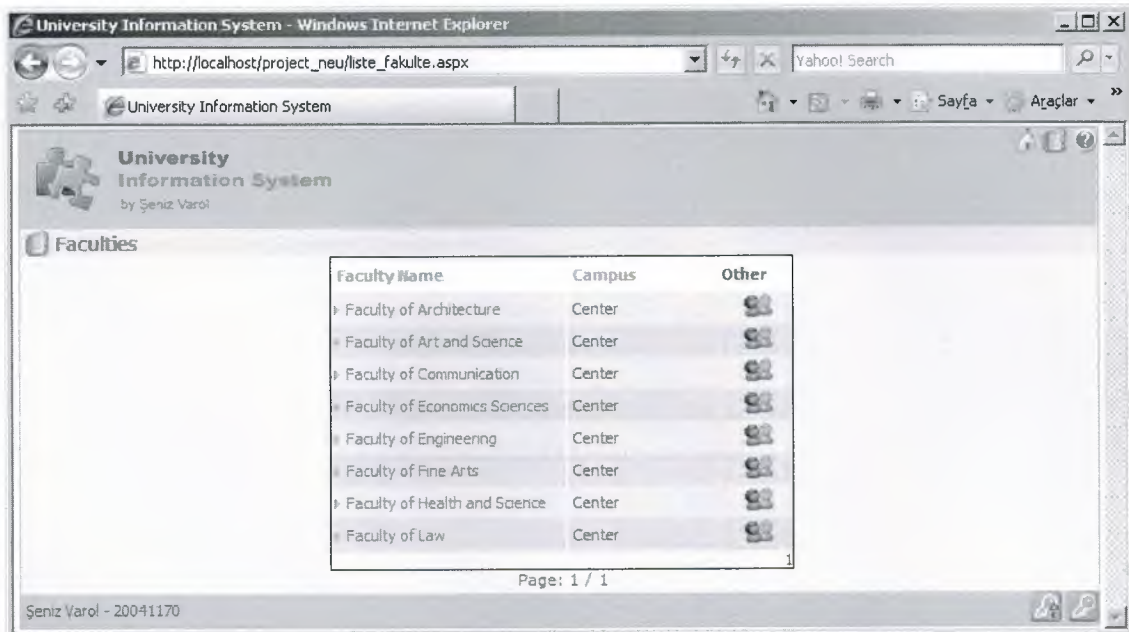
- **help.aspx** : It is a helper page to get information about the university information system.

- **hata.aspx** : It is an error page. When the user try to log in the page which is not permitted, he received an error message.

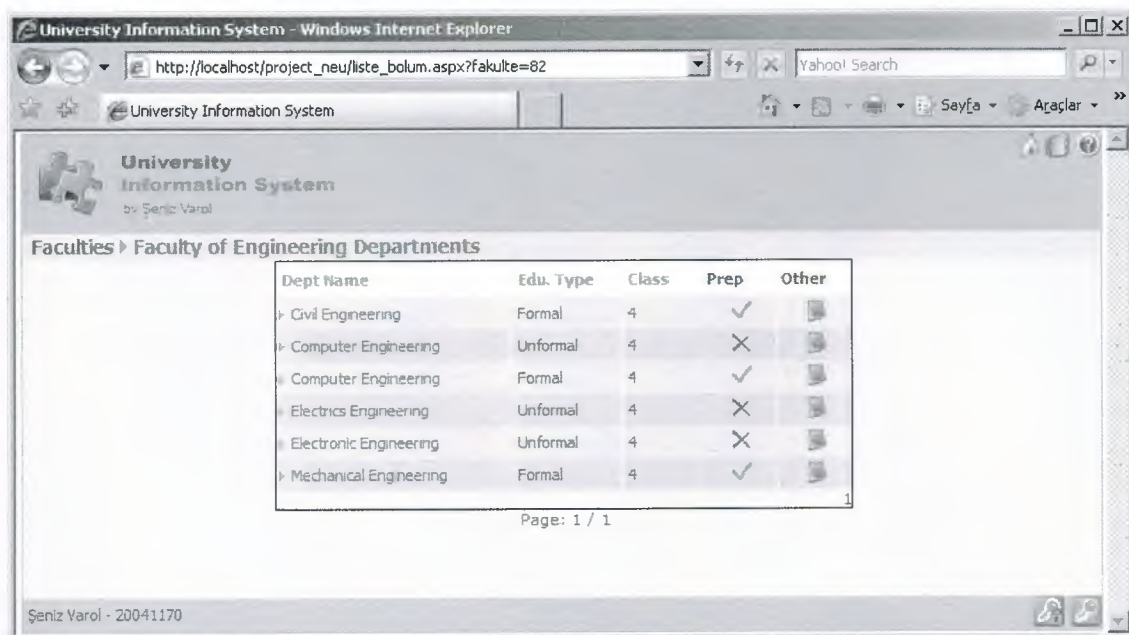
4.2. Overview of Project

Every user/visitor can log in these pages about faculties ,

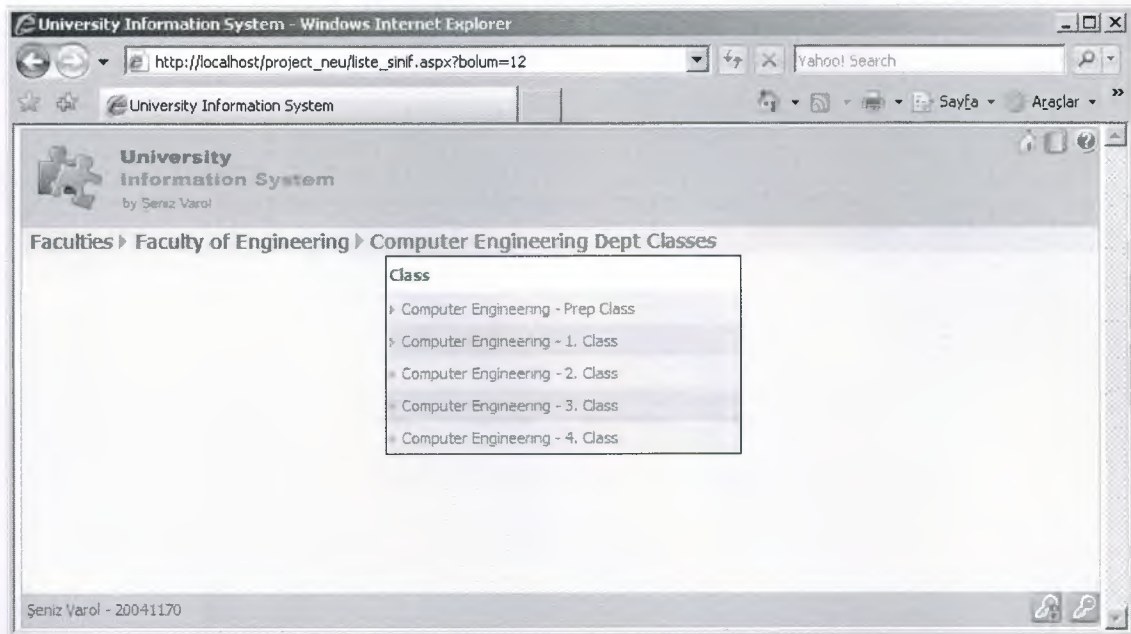
It shows the list of liste_fakulte.aspx in below ;



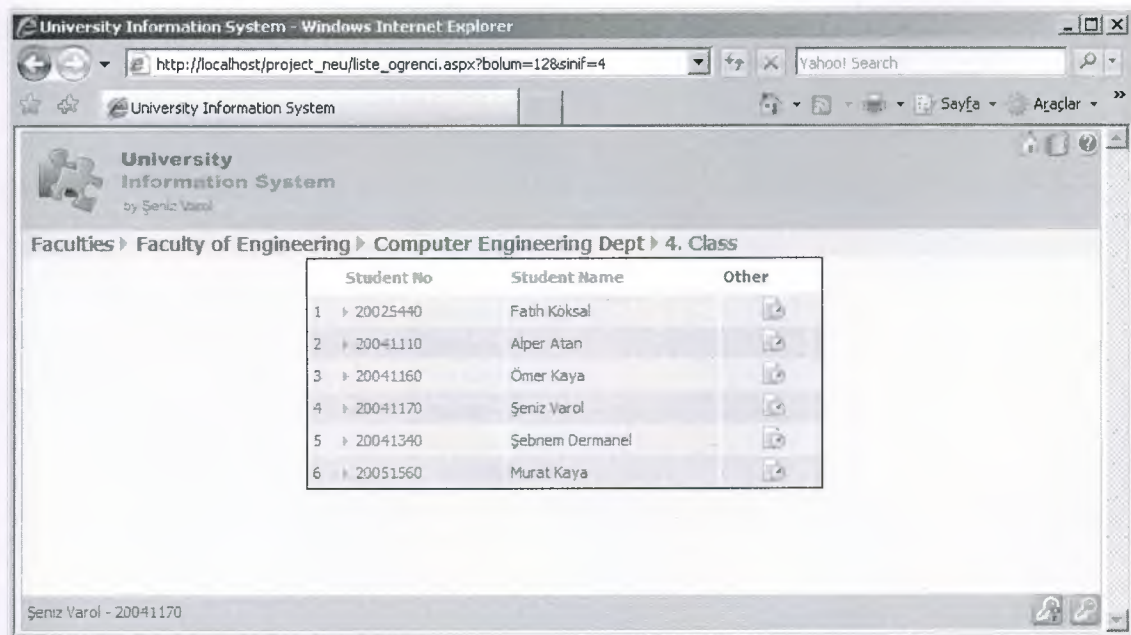
This page lists selected faculty of departments



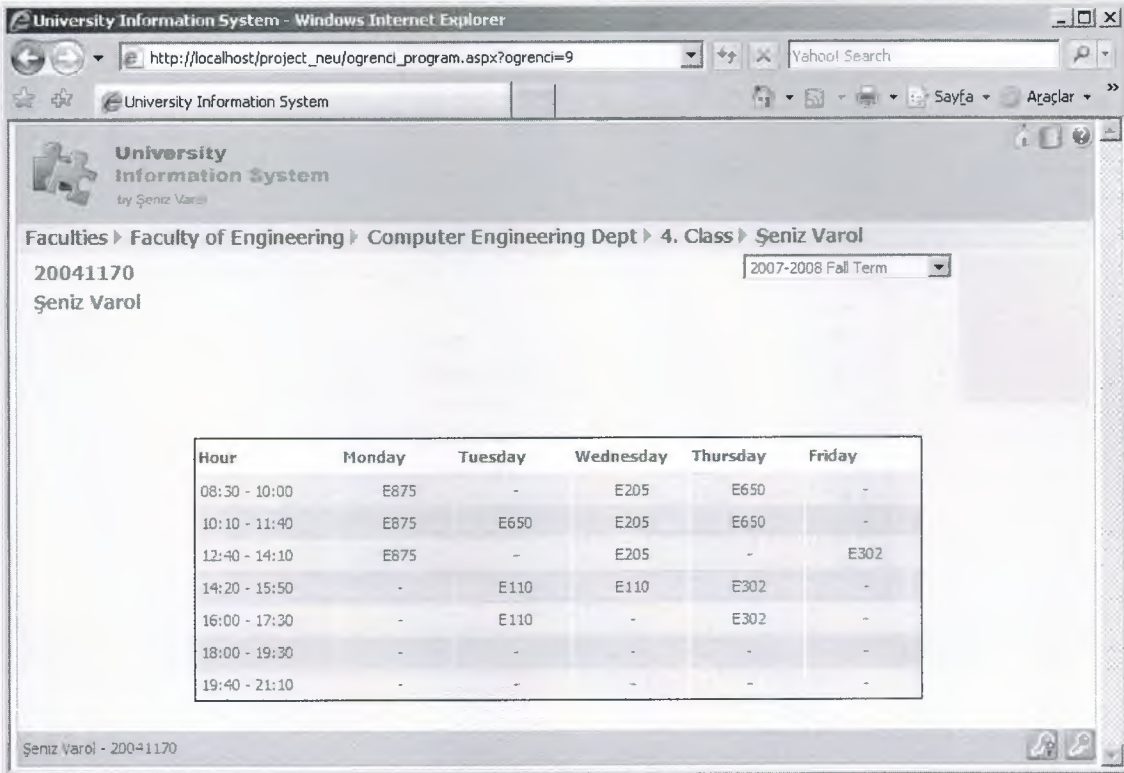
The List of class ;



The List of class ;



Visitors can not see the details of students and they can only see lesson schedule of selected term.



University Information System - Windows Internet Explorer

http://localhost/project_neu/ogrenci_program.aspx?ogrenci=9

University Information System

Faculties > Faculty of Engineering > Computer Engineering Dept > 4. Class > Şeniz Varol

20041170

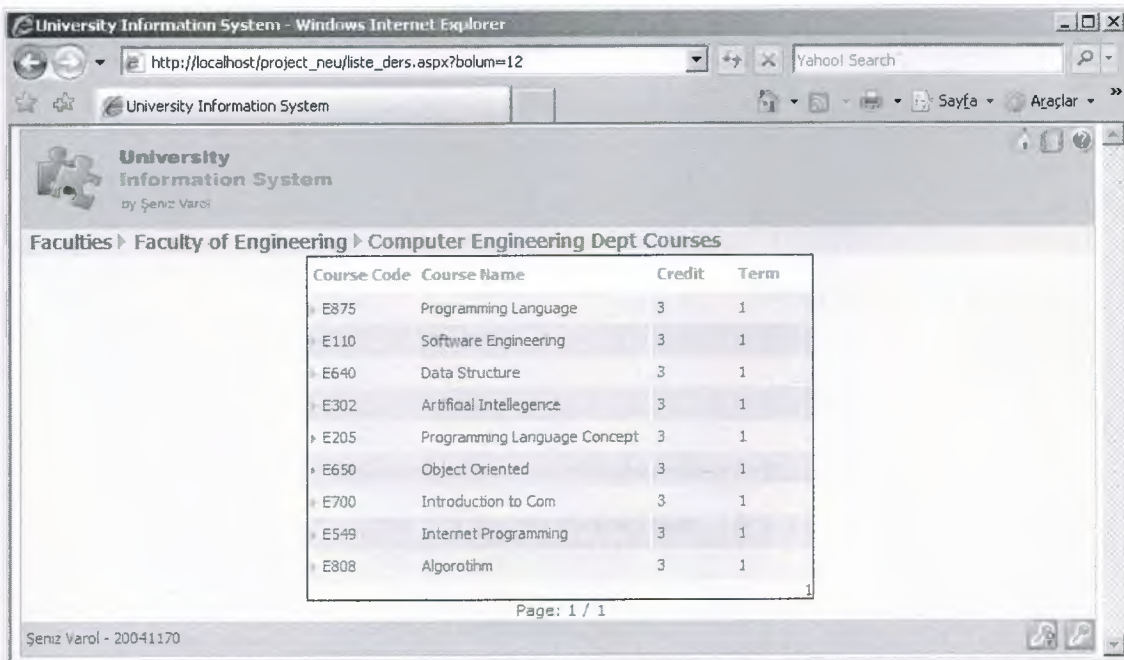
Şeniz Varol

2007-2008 Fall Term

| Hour | Monday | Tuesday | Wednesday | Thursday | Friday |
|---------------|--------|---------|-----------|----------|--------|
| 08:30 - 10:00 | E875 | - | E205 | E650 | - |
| 10:10 - 11:40 | E875 | E650 | E205 | E650 | - |
| 12:40 - 14:10 | E875 | - | E205 | - | E302 |
| 14:20 - 15:50 | - | E110 | E110 | E302 | - |
| 16:00 - 17:30 | - | E110 | - | E302 | - |
| 18:00 - 19:30 | - | - | - | - | - |
| 19:40 - 21:10 | - | - | - | - | - |

Şeniz Varol - 20041170

It is the list of the department courses. It shows that which term the lessons are given and also shows the credits of lessons.



University Information System - Windows Internet Explorer

http://localhost/project_neu/liste_ders.aspx?bolum=12

University Information System

Faculties > Faculty of Engineering > Computer Engineering Dept Courses

| Course Code | Course Name | Credit | Term |
|-------------|------------------------------|--------|------|
| E875 | Programming Language | 3 | 1 |
| E110 | Software Engineering | 3 | 1 |
| E640 | Data Structure | 3 | 1 |
| E302 | Artificial Intelligence | 3 | 1 |
| E205 | Programming Language Concept | 3 | 1 |
| E650 | Object Oriented | 3 | 1 |
| E700 | Introduction to Com | 3 | 1 |
| E549 | Internet Programming | 3 | 1 |
| E808 | Algorithm | 3 | 1 |

Page: 1 / 1

Şeniz Varol - 20041170

It is the list of Lecturers of Faculty

The screenshot shows a web browser window titled "University Information System - Windows Internet Explorer". The address bar displays "http://localhost/project_neu/liste_ogretmen.aspx?fakulte=82". The page content includes a header for "University Information System" by Şeniz Varol. Below this, a breadcrumb trail reads "Faculties > Faculty of Engineering Lecturers". A table lists eight lecturers with their IDs, names, duties, and other information. The table is as follows:

| | Lecturer Id | Lecturer Name | Duty | Other |
|---|-------------|---------------|------------|-------|
| 1 | 12121 | Sabiha Kara | Major | |
| 2 | 13133 | Huseyin Çiçek | Lecturer | |
| 3 | 19544 | Tony Doyle | Researcher | |
| 4 | 22221 | Oya Gezer | Lecturer | |
| 5 | 54555 | Hasan Kemal | Lecturer | |
| 6 | 56776 | Ahmed Seyid | Researcher | |
| 7 | 67766 | Ivan Nelson | Researcher | |
| 8 | 89888 | Kwun Lui | Lecturer | |

Page: 1 / 1

Şeniz Varol - 20041170

The lessons which the teacher gives in that term are listed and determined in this page.

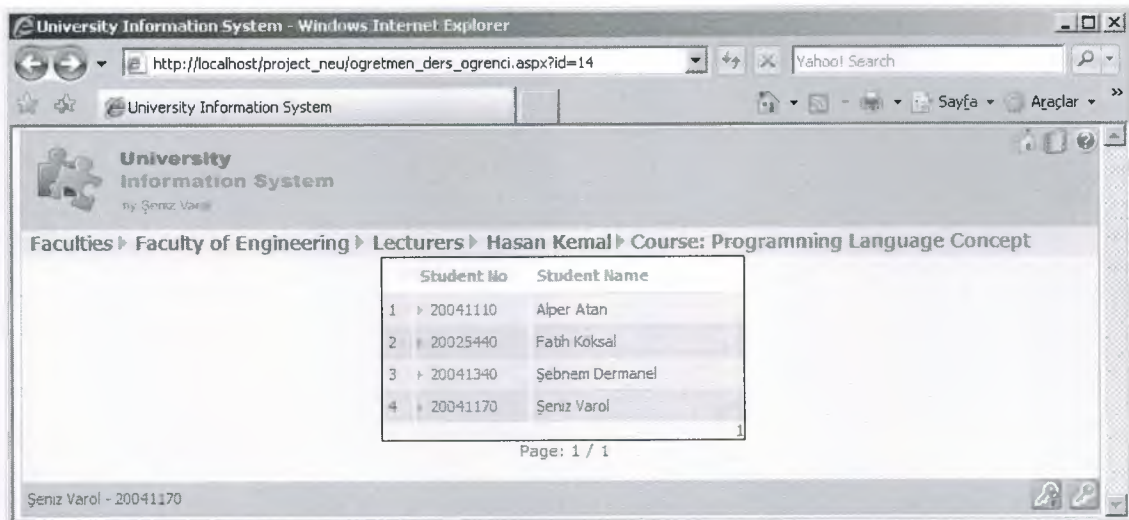
The screenshot shows a web browser window titled "University Information System - Windows Internet Explorer". The address bar displays "http://localhost/project_neu/ogretmen_ders.aspx?ogretmen=7". The page content includes a header for "University Information System" by Şeniz Varol. Below this, a breadcrumb trail reads "Faculties > Faculty of Engineering > Lecturers > Hasan Kemal". The page displays the name "Hasan Kemal" and title "Lecturer" for ID "54555". A dropdown menu shows "2007-2008 Fall Term". A portrait photo of Hasan Kemal is shown. Below this, a table lists the courses taught by the lecturer. The table is as follows:

| Course Code | Course Name | Other |
|-------------|------------------------------|-------|
| E205 | Programming Language Concept | |
| E549 | Internet Programming | |
| E650 | Object Oriented | |
| E808 | Algorithm | |

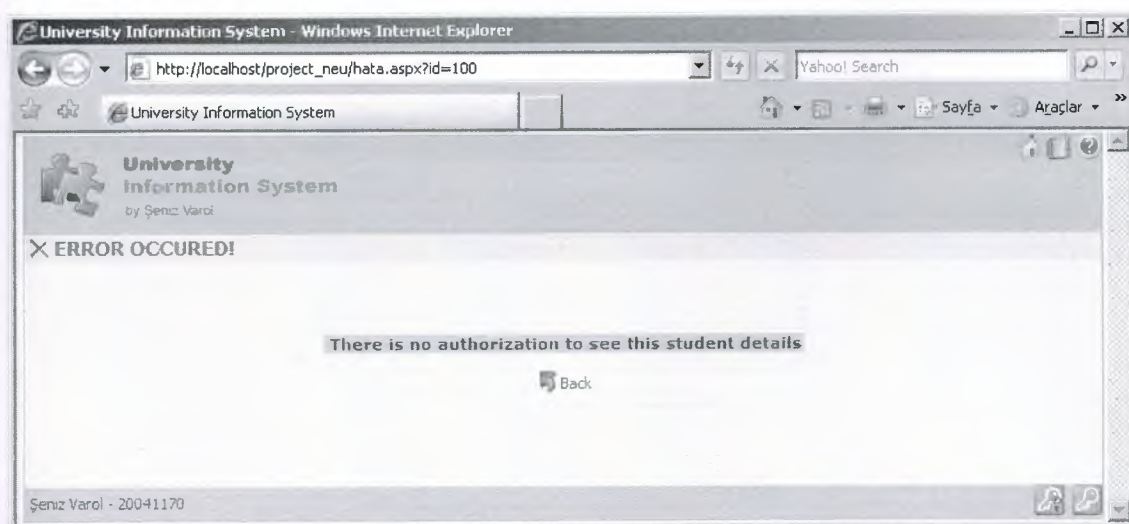
Page: 1 / 1

Şeniz Varol - 20041170

The students who attend the lesson which the teacher gives are shown in this page
ex:List of students who attend the Programming Language Concept lesson which
Hasan Kemal gives.



Visitor and users can not see any details of student or teacher. There is an error message from system as shown in below.



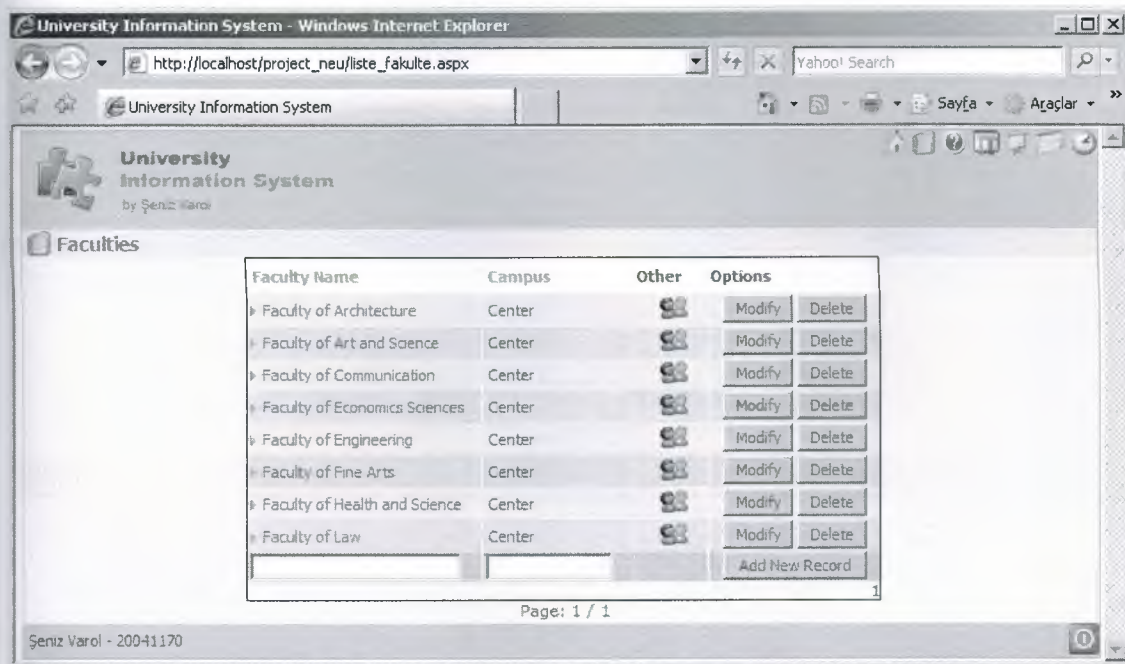
It is the pages of user's log in. The user can be either a student or a teacher. The user can log in using his own number and password. The password is valid during the user's online and it is also valid until closing the page and clicking the log out

The screenshot shows a web browser window titled "University Information System - Windows Internet Explorer". The address bar displays "http://localhost/project_neu/login.aspx". The page header includes the site name "University Information System" and a navigation menu with "Sayfa" and "Araçlar". The main content area features a "User Login" section with a form titled "USER LOGIN". The form contains two input fields for "Number" and "Password", and two radio buttons for "Student" (selected) and "Lecturer". A submit button with a right-pointing arrow is at the bottom right of the form. The footer of the page shows "Şeniz Varol - 20041170".

It is the page which is done administrative logging in. Administrative staff's passwords which belong to every faculty are defined. The staff selects the suitable faculty and password. Administrative Staff should log in with this password = 'project2007' it does not need to select the exact faculty

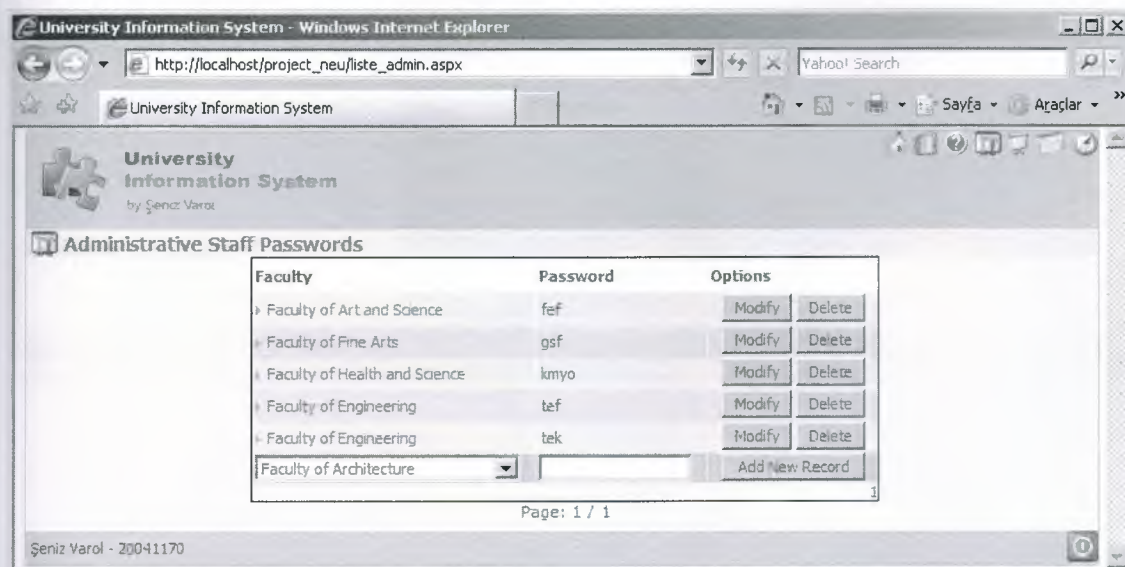
The screenshot shows a web browser window titled "University Information System - Windows Internet Explorer". The address bar displays "http://localhost/project_neu/login1.aspx". The page header includes the site name "University Information System" and a navigation menu with "Sayfa" and "Araçlar". The main content area features an "Administrative Staff Login" section with a form titled "ADMINISTRATIVE STAFF LOGIN". The form contains a dropdown menu for "Faculty" (currently showing "Faculty of Architecture") and a text input field for "Password". A submit button with a right-pointing arrow is at the bottom right of the form. The footer of the page shows "Şeniz Varol - 20041170".

If the system administrator logs in, the page seems like the following page. New icons (defined icons, admin, term, day, hour) are attached in the page of top.aspx. Also, login links in the page of bottom.aspx are exchanged with logging at. New faculty's logging can be deleted, updated and added by the means of the page of faculty list.



System Tools .:

Password of every faculty staff can be determined



Determine the terms.

University Information System - Windows Internet Explorer

http://localhost/project_neu/liste_donem.aspx

University Information System

University Information System
by Şeniz Varol

Education Terms

| Term Year | Semester | State | Default | Options |
|-------------|----------|--------------------------|---------|----------------|
| 2008 - 2009 | Fall | Term Not Started | X | Modify Delete |
| 2008 - 2009 | Spring | Term Not Started | X | Modify Delete |
| 2007 - 2008 | Fall | Term Course Registration | ✓ | Modify Delete |
| 2007 - 2008 | Spring | Term Not Started | X | Modify Delete |
| 2006 - 2007 | Fall | Term Ended | X | Modify Delete |
| 2006 - 2007 | Spring | Term Ended | X | Modify Delete |
| 2005 - 2006 | Fall | Term Ended | X | Modify Delete |
| 2005 - 2006 | Spring | Term Ended | X | Modify Delete |
| 2007 - 2008 | Fall | Term Not Started | | Add New Record |

Page: 1 / 1

Şeniz Varol - 20041170

Determine the lesson days.

University Information System - Windows Internet Explorer

http://localhost/project_neu/liste_gun.aspx

University Information System

University Information System
by Şeniz Varol

Course Days

| Day | Options |
|-----------|----------------|
| Monday | Modify Delete |
| Tuesday | Modify Delete |
| Wednesday | Modify Delete |
| Thursday | Modify Delete |
| Friday | Modify Delete |
| Monday | Add New Record |

Şeniz Varol - 20041170

Determine the Lesson duration

University Information System - Windows Internet Explorer

http://localhost/project_neu/liste_saati.aspx

University Information System

by Şeniz Varol

Daily Course Hours

| Course Hour | Edu. Type | Options |
|---------------|-----------|---------------|
| 08:30 - 10:00 | Formal | Modify Delete |
| 10:10 - 11:40 | Formal | Modify Delete |
| 12:40 - 14:10 | Formal | Modify Delete |
| 14:20 - 15:50 | Formal | Modify Delete |
| 16:00 - 17:30 | Formal | Modify Delete |
| 18:00 - 19:30 | Unformal | Modify Delete |
| 19:40 - 21:10 | Unformal | Modify Delete |

00 : 00 - 00 : 00 Formal Add New Record

Page: 1 / 1

Şeniz Varol - 20041170

Other utilities;

If logging in of System Administor is done,a new faculty can be added or the faculty which is already saved can be changed or deleted.

University Information System - Windows Internet Explorer

http://localhost/project_neu/liste_fakulte.aspx

University Information System

by Şeniz Varol

Faculties

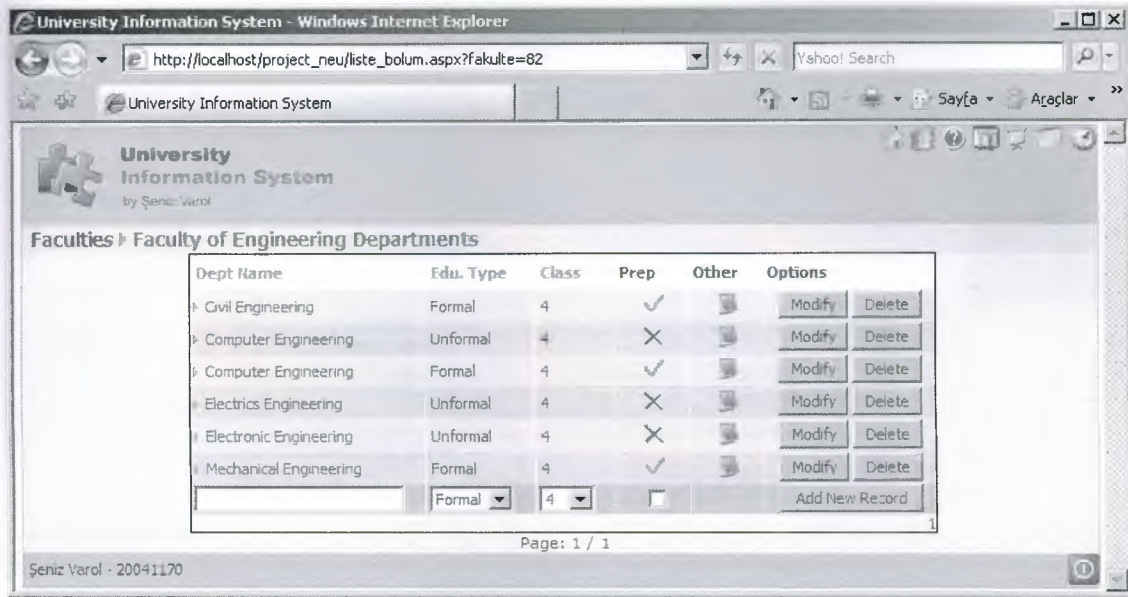
| Faculty Name | Campus | Other | Options |
|-------------------------------|--------|-------|---------------|
| Faculty of Architecture | Center | | Modify Delete |
| Faculty of Art and Science | Center | | Modify Delete |
| Faculty of Communication | Center | | Modify Delete |
| Faculty of Economics Sciences | Center | | Update Cancel |
| Faculty of Engineering | Center | | Modify Delete |
| Faculty of Fine Arts | Center | | Modify Delete |
| Faculty of Health and Science | Center | | Modify Delete |
| Faculty of Law | Center | | Modify Delete |

Add New Record

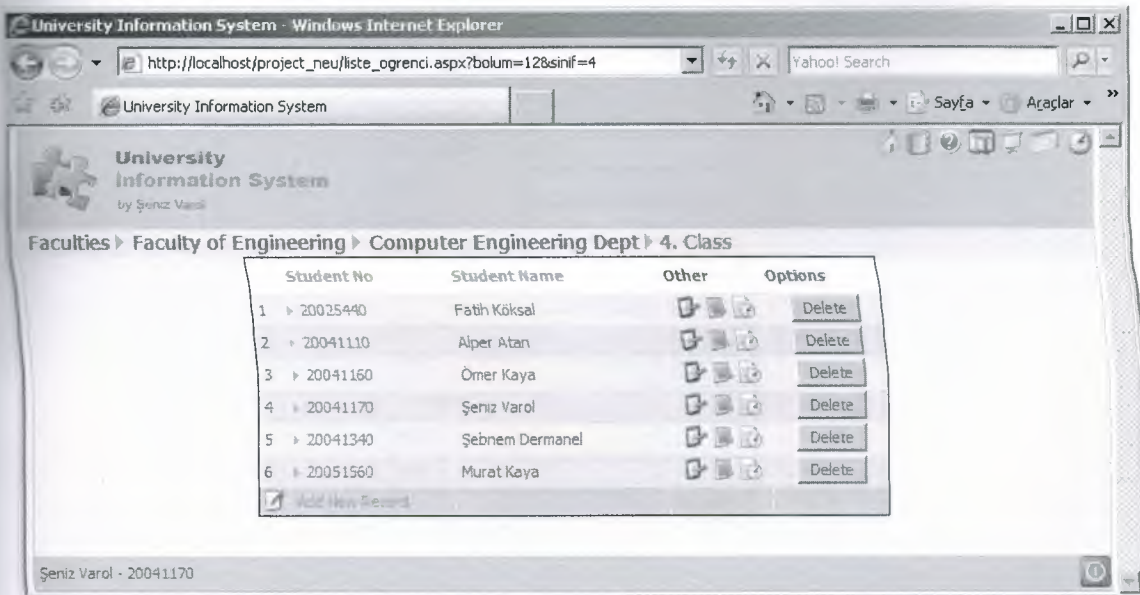
Page: 1 / 1

Şeniz Varol - 20041170

If logging in of system administer or logging in of the staff of that faculty is done, a new department can be added to the new faculty and the departments can be changed or deleted.



The List of the class can be changed and the new students can be added or deleted.



The Details of Students

University Information System - Windows Internet Explorer


http://localhost/project_neu/detay_ogrenci.aspx?ogrenci=9

University Information System

University Information System
by Şeniz Varol

Faculties ▶ Faculty of Engineering ▶ Computer Engineering Dept ▶ 4. Class ▶ Şeniz Varol

20041170
Şeniz Varol



PERSONEL INFO

| | | | |
|-------|---------------|------------|------------------------|
| Sex | Male | Birth Date | 26.01.1982 |
| Phone | (532) 4553322 | Mail | senizvarol@hotmail.com |

EDUCATION INFO

Marks List Courses Course Program

Şeniz Varol - 20041170

Add New Student

University Information System - Windows Internet Explorer

http://localhost/project_neu/ekle_ogrenci.aspx?bolu=12&sinif=4

University Information System

University Information System
by Şeniz Varol

Faculties ▶ Faculty of Engineering ▶ Computer Engineering Dept ▶ 4. Class ▶ Add New Student

ADD NEW STUDENT

Number

Password

Password (Again)

Name

Surname

Sex ☒ Male ☐ Female

Birth Date

Phone

Mail

Add

Şeniz Varol - 20041170

Update the information of students on student details page.

University Information System - Windows Internet Explorer

http://localhost/project_neu/guncelle_ogrenci.aspx?ogrenci=9

University Information System

Faculties ▶ Faculty of Engineering ▶ Computer Engineering Dept ▶ 4. Class ▶ Şeniz Varol

UPDATE STUDENT INFORMATIONS

Number: 20041170
Password: 1
Name: Şeniz
Surname: Varol
Sex: ☒ Male ☐ Female
Birth Date: 26.01.1982
Phone: (532) 4553322
Mail: senizvarol@hotmail.com

Update

Şeniz Varol - 20041170

Add new record to the selected department and determine the
Information of lesson

University Information System - Windows Internet Explorer

http://localhost/project_neu/liste_ders.aspx?bolum=12

University Information System

Faculties ▶ Faculty of Engineering ▶ Computer Engineering Dept Courses

| Course Code | Course Name | Credit | Term | Options |
|-------------|------------------------------|--------|------|----------------|
| E875 | Programming Language | 3 | 1 | Modify Delete |
| E110 | Software Engineering | 3 | 1 | Modify Delete |
| E640 | Data Structure | 3 | 1 | Modify Delete |
| E302 | Artificial Intelligence | 3 | 1 | Modify Delete |
| E205 | Programming Language Concept | 3 | 1 | Modify Delete |
| E650 | Object Oriented | 3 | 1 | Modify Delete |
| E700 | Introduction to Com | 3 | 1 | Modify Delete |
| E549 | Internet Programming | 3 | 1 | Modify Delete |
| E806 | Algorithm | 3 | 1 | Modify Delete |
| | | 4 | 1 | Add New Record |

Page: 1 / 1

Şeniz Varol - 20041170

Determine lecturers's information in faculty









University Information System - Windows Internet Explorer

http://localhost/project_neu/liste_ogretmen.aspx?fakulte=82

University Information System

University Information System
by Şeniz Varol

Faculties ▶ Faculty of Engineering Lecturers

| | Lecturer Id | Lecturer Name | Duty | Other | Options |
|-----|-------------|---------------|------------|--|---------|
| 1 ▶ | 12121 | Sabiha Kara | Major |  | Delete |
| 2 ▶ | 13133 | Hüseyin Çiçek | Lecturer |  | Delete |
| 3 ▶ | 19544 | Tony Doyle | Researcher |  | Delete |
| 4 ▶ | 22221 | Oya Gezer | Lecturer |  | Delete |
| 5 ▶ | 54555 | Hasan Kemal | Lecturer |  | Delete |
| 6 ▶ | 56776 | Ahmed Seyid | Researcher |  | Delete |
| 7 ▶ | 67766 | Ivan Nelson | Researcher |  | Delete |
| 8 ▶ | 89888 | Kwun Lui | Lecturer |  | Delete |

Page: 1 / 1

Şeniz Varol - 20041170

Details of Lecturers' information

University Information System - Windows Internet Explorer


http://localhost/project_neu/detay_ogretmen.aspx?ogretmen=7

University Information System

University Information System
by Şeniz Varol

Faculties ▶ Faculty of Engineering ▶ Lecturers ▶ Hasan Kemal

54555
Hasan Kemal
Lecturer



PERSONEL INFORMATION

| | | | |
|-------|---------------|------------|---------------|
| Sex | Male | Birth Date | 10.10.1974 |
| Phone | (533) 4432211 | Mail | hasan@bil.org |

WORK INFORMATIONS

Courses Course Program

Şeniz Varol - 20041170

Add a new Lecturer

University Information System - Windows Internet Explorer

http://localhost/project_neu/ekle_ogretmen.aspx?fakulte=82

University Information System

University Information System
by Şeniz Varol

Faculties ▶ Faculty of Engineering ▶ Lecturers ▶ Add New Lecturer

ADD NEW LECTURER

| | |
|------------------|--|
| Code | <input type="text"/> |
| Password | <input type="password"/> |
| Password (Again) | <input type="password"/> |
| Name | <input type="text"/> |
| Surname | <input type="text"/> |
| Sex | <input checked="" type="radio"/> Male <input type="radio"/> Female |
| Duty | <input type="text" value="Lecturer"/> |
| Birth Date | <input type="text"/> |
| Phone | <input type="text"/> |
| Mail | <input type="text"/> |

Add

Şeniz Varol - 20041170

Update lecturer's information

University Information System - Windows Internet Explorer

http://localhost/project_neu/guncelle_ogretmen.aspx?ogretmen=7

University Information System

University Information System
by Şeniz Varol

Faculties ▶ Faculty of Engineering ▶ Lecturers ▶ Hasan Kemal

UPDATE LECTURER INFORMATIONS

| | |
|------------|--|
| Code | 54555 |
| Password | 55 |
| Name | Hasan |
| Surname | Kemal |
| Sex | <input checked="" type="radio"/> Male <input type="radio"/> Female |
| Unvan | <input type="text" value="Lecturer"/> |
| Birth Date | 10. 10. 1974 |
| Phone | (533) 4432211 |
| Mail | hasan@bil.org |

Update

Şeniz Varol - 20041170

The lesson which depends on faculty can be given by lecturer

University Information System - Windows Internet Explorer

http://localhost/project_neu/ogretmen_ders.aspx?ogretmen=7

University Information System

Faculties ▶ Faculty of Engineering ▶ Lecturers ▶ Hasan Kemal

54555

Hasan Kemal

Lecturer

2007-2008 Fall Term

| Course Code | Course Name | Other | Options |
|-------------------|------------------------------|-------|---------|
| E205 | Programming Language Concept | | Delete |
| E549 | Internet Programming | | Delete |
| E650 | Object Oriented | | Delete |
| E808 | Algorotihm | | Delete |
| E808 - Algorotihm | | | Add |

Page: 1 / 1

Şeniz Varol - 20041170

The lesson schedule of lecturer

University Information System - Windows Internet Explorer

http://localhost/project_neu/ogretmen_program.aspx?ogretmen=7

University Information System

Faculties ▶ Faculty of Engineering ▶ Lecturers ▶ Hasan Kemal

54555

Hasan Kemal

Lecturer

2007-2008 Fall Term

| Hour | Monday | Tuesday | Wednesday | Thursay | Friday |
|---------------|--------|---------|-----------|---------|--------|
| 08:30 - 10:00 | E808 | E549 | E205 | E650 | E808 |
| 10:10 - 11:40 | E808 | E650 | E205 | E650 | - |
| 12:40 - 14:10 | E549 | - | E205 | - | - |
| 14:20 - 15:50 | E549 | - | - | - | - |
| 16:00 - 17:30 | - | - | - | - | - |
| 18:00 - 19:30 | - | - | - | - | - |
| 19:40 - 21:10 | - | - | - | - | - |

Create New Program

Şeniz Varol - 20041170

Prepare a new schedule of lecturer

University Information System - Windows Internet Explorer

http://localhost/project_neu/ogretmen_program.aspx?ogretmen=7

University Information System

Faculties > Faculty of Engineering > Lecturers > Hasan Kemal

54555

Hasan Kemal

Lecturer

2007-2008 Fall Term

| Hour | Monday | Tuesday | Wednesday | Thursday | Friday |
|---------------|--------|---------|-----------|----------|--------|
| 08:30 - 10:00 | E608 | E549 | E205 | E650 | E808 |
| 10:10 - 11:40 | E808 | E650 | E205 | E650 | - |
| 12:40 - 14:10 | E549 | - | E205 | - | - |
| 14:20 - 15:50 | E549 | E808 | - | - | - |
| 16:00 - 17:30 | - | E549 | - | - | - |
| 18:00 - 19:30 | - | E650 | - | - | - |
| 19:40 - 21:10 | - | E205 | - | - | - |

Ok Cancel

Şeniz Varol - 20041170

It shows the lessons which are given in the faculty of student.

University Information System - Windows Internet Explorer

http://localhost/project_neu/ogrenci_ders.aspx?ogrenci=9

University Information System

Faculties > Faculty of Engineering > Computer Engineering Dept > 4. Class > Şeniz Varol

20041170

Şeniz Varol

2007-2008 Fall Term

| Course Code | Course Name | Course Info |
|-------------|------------------------------|---|
| E110 | Software Engineering | Term: 2007-Fall Credit: 3 Lecturer: Oya Gezer |
| E205 | Programming Language Concept | Term: 2007-Fall Credit: 3 Lecturer: Hasan Kemal |
| E302 | Artificial Intelligence | Term: 2007-Fall Credit: 3 Lecturer: Oya Gezer |
| E650 | Object Oriented | Term: 2007-Fall Credit: 3 Lecturer: Hasan Kemal |
| E875 | Programming Language | Term: 2007-Fall Credit: 3 Lecturer: Oya Gezer |

New Course Registration

Şeniz Varol - 20041170

Choose the lesson of student:

The student chooses according to the teacher which gives and determines the lesson. Apart from the lesson, credit, teacher, days and hours are shown.

University Information System - Windows Internet Explorer

http://localhost/project_neu/ogrenci_ders.aspx?ogrenci=9


University Information System

University Information System
by Şeniz Varol

Faculties ▶ Faculty of Engineering ▶ Computer Engineering Dept ▶ 4. Class ▶ Şeniz Varol

20041170
Şeniz Varol

2007-2008 Fall Term



| Course Code | Course Name | Ders Bilgileri | Course Hours |
|--|------------------------------|---|---|
| <input checked="" type="checkbox"/> ▶ E110 | Software Engineering | Term: 2007-Fall Credit: 3 Lecturer: Oya Gezer | Tuesday - (14:20 - 15:50) Tuesday - (16:00 - 17:30) Wednesday - (14:20 - 15:50) |
| <input checked="" type="checkbox"/> ▶ E205 | Programming Language Concept | Term: 2007-Fall Credit: 3 Lecturer: Hasan Kemal | Wednesday - (08:30 - 10:00) Wednesday - (10:10 - 11:40) Wednesday - (12:40 - 14:10) |
| <input checked="" type="checkbox"/> ▶ E302 | Artificial Intelligence | Term: 2007-Fall Credit: 3 Lecturer: Oya Gezer | Thursday - (14:20 - 15:50) Thursday - (16:00 - 17:30) Friday - (12:40 - 14:10) |
| <input type="checkbox"/> ▶ E302 | Artificial Intelligence | Term: 2007-Fall Credit: 3 Lecturer: Hüseyin Çiçek | Wednesday - (10:10 - 11:40) Thursday - (08:30 - 10:00) Thursday - (10:10 - 11:40) |
| <input type="checkbox"/> ▶ E549 | Internet Programming | Term: 2007-Fall Credit: 3 Lecturer: Hasan Kemal | Monday - (12:40 - 14:10) Monday - (14:20 - 15:50) Tuesday - (08:30 - 10:00) |
| <input type="checkbox"/> ▶ E640 | Data Structure | Term: 2007-Fall Credit: 3 Lecturer: Hüseyin Çiçek | Monday - (14:20 - 15:50) Monday - (16:00 - 17:30) Friday - (10:10 - 11:40) |
| <input checked="" type="checkbox"/> ▶ E650 | Object Oriented | Term: 2007-Fall Credit: 3 Lecturer: Hasan Kemal | Tuesday - (10:10 - 11:40) Thursday - (08:30 - 10:00) Thursday - (10:10 - 11:40) |
| <input type="checkbox"/> ▶ E808 | Algorithm | Term: 2007-Fall Credit: 3 Lecturer: Hasan Kemal | Monday - (08:30 - 10:00) Monday - (10:10 - 11:40) Friday - (08:30 - 10:00) |
| <input type="checkbox"/> ▶ E808 | Algorithm | Term: 2007-Fall Credit: 3 Lecturer: Hüseyin Çiçek | Tuesday - (08:30 - 10:00) Tuesday - (10:10 - 11:40) Wednesday - (08:30 - 10:00) |
| <input checked="" type="checkbox"/> ▶ E875 | Programming Language | Term: 2007-Fall Credit: 3 Lecturer: Oya Gezer | Monday - (08:30 - 10:00) Monday - (10:10 - 11:40) Monday - (12:40 - 14:10) |

Ok Cancel

Şeniz Varol - 20041170

The lesson schedule is prepared according to the lessons which the students chooses.

The lessons schedule of the student depends on the days and hours which the teacher determines.

The screenshot shows a web browser window titled "University Information System - Windows Internet Explorer". The address bar displays "http://localhost/project_neu/ogrenci_program.aspx?ogrenci=9". The page header includes "University Information System" and "by Şeniz Varol". The breadcrumb trail is "Faculties > Faculty of Engineering > Computer Engineering Dept > 4. Class > Şeniz Varol". The student ID "20041170" and name "Şeniz Varol" are shown. A dropdown menu is set to "2007-2008 Fall Term". A small portrait of Şeniz Varol is visible on the right. The main content is a table showing the lesson schedule:

| Hour | Monday | Tuesday | Wednesday | Thursday | Friday |
|---------------|--------|---------|-----------|----------|--------|
| 08:30 - 10:00 | E875 | - | E205 | E650 | - |
| 10:10 - 11:40 | E875 | E650 | E205 | E650 | - |
| 12:40 - 14:10 | E875 | - | E205 | - | E302 |
| 14:20 - 15:50 | - | E110 | E110 | E302 | - |
| 16:00 - 17:30 | - | E110 | - | E302 | - |
| 18:00 - 19:30 | - | - | - | - | - |
| 19:40 - 21:10 | - | - | - | - | - |

The footer shows "Şeniz Varol - 20041170".

The teacher can see the student which take his lesson and he can give the marks

By the means of this system.

The screenshot shows a web browser window titled "University Information System - Windows Internet Explorer". The address bar displays "http://localhost/project_neu/ogretmen_ders_ogrenci.aspx?id=14". The page header includes "University Information System" and "by Şeniz Varol". The breadcrumb trail is "Faculties > Faculty of Engineering > Lecturers > Hasan Kemal > Course: Programming Language Concept". The main content is a table showing student marks:

| Student No | Student Name | 1.Mterm | 2.Mterm | Final | Attendance | Average | State | Options |
|--------------|-----------------|---------|---------|-------|------------|---------|--------------|---------|
| 1 > 20041110 | Alper Atan | 60 | 50 | 60 | 30 | 58 | Absent | Modify |
| 2 > 20025440 | Fatih Koksai | 70 | 20 | 20 | | 32 | Unsuccessful | Modify |
| 3 > 20041340 | Şebnem Dermanel | 60 | 80 | 70 | | 70 | Successful | Modify |
| 4 > 20041170 | Şeniz Varol | 80 | 90 | 80 | | 82 | Successful | Modify |

The footer shows "Page: 1 / 1" and "Şeniz Varol - 20041170".

Midterms, Final's marks and attendance are shown in the same page.

University Information System - Windows Internet Explorer

http://localhost/project_neu/ogretmen_ders_ogrenci.aspx?id=14

University Information System

University Information System by Şeniz Varol

Faculties ▶ Faculty of Engineering ▶ Lecturers ▶ Hasan Kemal ▶ Course: Programming Language Concept

| Student No | Student Name | 1.Mterm | 2.Mterm | Final | Attendance | Average | State | Options |
|--------------|-----------------|---------|---------|-------|-------------------------------------|---------|--------------|---------------|
| 1 ▶ 20041110 | Alper Atan | 60 | 50 | 60 | 30 | 58 | Absent | Modify |
| 2 ▶ 20025440 | Fatih Köksal | 70 | 20 | 20 | | 32 | Unsuccessful | Modify |
| 3 ▶ 20041340 | Şebnem Dermanel | 60 | 80 | 70 | | 70 | Successful | Modify |
| 4 ▶ 20041170 | Şeniz Varol | 80 | 90 | 80 | <input checked="" type="checkbox"/> | 82 | Successful | Update Cancel |

Page: 1 / 1

Şeniz Varol - 20041170

The students can see the marks like the following;

University Information System - Windows Internet Explorer

http://localhost/project_neu/ogrenci_ders_ogretmen.aspx?ogrenci=9

University Information System

University Information System by Şeniz Varol

Faculties ▶ Faculty of Engineering ▶ Computer Engineering Dept ▶ 4. Class ▶ Şeniz Varol

2007-2008 Fall Term Course Marks

2007-2008 Fall Term

| Course Code | Course Name | Course Info | 1.Mterm | 2.Mterm | Final | Attendance | Average | State |
|-------------|------------------------------|---|---------|---------|-------|------------|---------|--------------|
| 1 ▶ E110 | Software Engineering | Lecturer: Oya Gezer Term: 2007-Fall Credit: 3 | 70 | 50 | | | 30 | |
| 2 ▶ E205 | Programming Language Concept | Lecturer: Hasan Kemal Term: 2007-Fall Credit: 3 | 80 | 90 | 80 | | 82 | Successful |
| 3 ▶ E302 | Artificial Intelligence | Lecturer: Oya Gezer Term: 2007-Fall Credit: 3 | 55 | 70 | | | 31 | |
| 4 ▶ E650 | Object Oriented | Lecturer: Hasan Kemal Term: 2007-Fall Credit: 3 | 70 | 60 | 80 | | 72 | Successful |
| 5 ▶ E875 | Programming Language | Lecturer: Oya Gezer Term: 2007-Fall Credit: 3 | 50 | 40 | 40 | | 42 | Unsuccessful |

Şeniz Varol - 20041170

4.3. Database

| DERS_TANIM |
|----------------|
| Ders_Id |
| Ders_Kod |
| Ders_Ad |
| Kredi |
| Donem |
| Bolum_Id |

| DONEM_OGRETMEN_DERS_GUN_SAAAT |
|--|
| Donem_Ogretmen_Ders_Gun_Saat_Id |
| Donem_Ogretmen_Ders_Id |
| Gun_Id |
| Saat_Id |

| DONEM_OGRETMEN_DERS |
|-------------------------------|
| Donem_Ogretmen_Ders_Id |
| Donem_Id |
| Ogretmen_Kayit |
| Ders_Id |

| OGRENCI_TANIM |
|----------------------|
| Ogrenci_Kayit |
| Ogrenci_No |
| Sifre |
| Ad |
| Soyad |
| Cinsiyet |
| Dog_T |
| Tel |
| Mail |
| Resim |
| Donem |
| Bolum_Id |

| OGRETMEN_TANIM |
|-----------------------|
| Ogretmen_Kayit |
| Ogretmen_Kod |
| Sifre |
| Ad |
| Soyad |
| Cinsiyet |
| Unvan |
| Dog_T |
| Tel |
| Mail |
| Resim |
| Fakulte_Id |

| DONEM_OGRENCI_DERS |
|------------------------------|
| Donem_Ogrenci_Ders_Id |
| Donem_Ogretmen_Ders_Id |
| Ogrenci_Kayit |
| Kka |
| Vize1 |
| Vize2 |
| Final |
| Devam |
| Devam_Durum |
| Ortalama |
| Durum |

| DONEM_TANIM |
|-----------------|
| Donem_Id |
| Basla_Donem_Ad |
| Yariyil |
| Durum |
| Varsayilan |

| BOLUM_TANIM |
|-----------------|
| Bolum_Id |
| Bolum_Ad |
| Ogretim |
| Sinif |
| Hazirlık |
| Fakulte_Id |

| GUN_TANIM |
|---------------|
| Gun_Id |
| Gun |

| ADMIN_TANIM |
|-----------------|
| Admin_Id |
| Sifre |
| Fakulte_Id |

| FAKULTE_TANIM |
|-------------------|
| Fakulte_Id |
| Fakulte_Ad |
| Kampus |

| SAAT_TANIM |
|----------------|
| Saat_Id |
| Saat |
| Ogretim |

REFERENCES

[W3C-01] Gudgin, Martin, et al., "SOAP Version 1.2," W3C Working Draft, July 2001.

www.w3.org/TR/2001/WD-soap12-20010709/

[Osb-00] Osborn, John, "Deep Inside C#: An Interview with Microsoft Chief Architect Anders Hejlsberg," July 2000.

windows.oreilly.com/news/hejlsberg_0800.html

[Pat-01a] Pattison, Ted, "Basic Instincts: New Features in Visual Basic .NET," MSDN Magazine, May 2001.

msdn.microsoft.com/msdnmag/issues/01/05/instincts/instincts0105.asp

[Pat-01b] Pattison, Ted, "Basic Instincts: Exploiting New Language Features in Visual Basic .NET, Part 2," MSDN Magazine, August 2001.

msdn.microsoft.com/msdnmag/issues/01/08/Instincts/Instincts0108.asp

Mastering™ Visual Basic® .NET

Database Programming Evangelos Petroustos

Active Server Pages Basic Guide – Dr.Hakkı Öcal Haziran 2000,IMG BILISIM PRESS.

APPENDIX

default.aspx

```
<!-- University Information System - Şeniz Varol 20041170 -->

<html>
<link href="style/main.css" rel="stylesheet" type="text/css">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-9">
</head>
<body>
<TABLE width="770" height="100%" align="center" cellpadding="0" cellspacing="0"
class="backtable">
<TR><TD height="70" class="color-footer"><!--#include file="include/top.aspx"--
></TD></TR>
<TR><TD height="20" class="color-row1"><asp:Label id="lblFakulte" runat="server"
CssClass="txt-Title" Text=' University Information System Home Page' /></TD></TR>
<TR><TD align="center" valign="top">
<br><br>
<table class="formtable" width="400">
<tr class="color-title">
<td class="txt-title">
 What is University Information System?
</td>
</tr>
<tr>
<td align="center"><br>
<table width="345"><tr>
<td> 
The university information system which has been new yet is
a system which keeps the whole information about a university's
characteristics and presents them to use.This system which is
```


stable unit of university or includes the functions which need to be updated every education term provides an opportunity for most of functions which are often done and it also provides an opportunity to use it handier.

A staff in a university provides the interaction between a teacher and a student. This system makes the functions of association easy and due to this system the information is taken fast and without interruption. </td>

</tr></table>

</td>

</tr>

</table>

<table class="formtable" width="400">

<tr class="color-title">

<td class="txt-title">

 How to Use System?

</td>

</tr>

<tr>

<td align="center">

<table width="365"><tr>

<td valign="baseline"></td>

<td>Information About the Structure of System Defined as Shape

and Explanation Linked Help

Topic with Details.

</td></tr>

<tr>

<td valign="baseline"></td>

<td> It is a system which keeps the information which is beginning from the biggest Faculty to the student's details.If it is revised step by step,the branches of the faculty and the details of information are needed to be observed.</td>

</tr></table>

</td>

</tr>

</table>

</TD></TR>

<TR><TD height="20" class="color-footer"><!--#include file="include/bottom.aspx"--></TD></TR>

</TABLE>

</body>

</html>

hata.aspx

<!-- University Information System - Şeniz Varol 20041170 -->

<%@ Page Language="VB" Debug="true" codePage="28599" %>

<script runat="server">

Sub Page_Load(Sender as Object, E as EventArgs)

Dim Id as integer

Dim mesaj as string

Id = request.QueryString("id")

Select Case Id

Case 100 : mesaj = "There is no authorization to see this student details"

Case 200 : mesaj = "There is no authorization to see this lecturer details"

Case 300 : mesaj = "Only system administrator can access this page"

```

Case else : mesaj = "An error was occurred"
End Select
Hata.Text = mesaj
End Sub
</script>
<html>
<link href="style/main.css" rel="stylesheet" type="text/css">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-9">
</head>
<body>
<TABLE width="770" height="100%" align="center" cellpadding="0" cellspacing="0"
class="backtable">
<TR><TD height="70" class="color-footer"><!--#include file="include/top.aspx"--
></TD></TR>
<TR><TD height="20" class="color-row1"><asp:Label id="lblFakulte" runat="server"
CssClass="txt-Title" Text=' ERROR OCCURED!' /></TD></TR>
<TR><TD align="center" valign="top">
<br><br><br><br>
<% if session("Admin") = "OK" Then
response.Write("ERROR<br><br>")
End if %>
<asp:Label id="Hata" runat="server" CssClass="txt-warning" />
<br><br>
<a href="#" onClick="javascript: history.go(-1);" class="txt-item"> Back</a>
</TD></TR>
<TR><TD height="20" class="color-footer"><!--#include file="include/bottom.aspx"--
></TD></TR>
</TABLE>
</body>
</html>

```


help.aspx

```
<!-- University Information System - Şeniz Varol 20041170 -->

<html>
<link href="style/main.css" rel="stylesheet" type="text/css">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-9">
</head>
<body>
<TABLE width="770" height="100%" align="center" cellpadding="0" cellspacing="0"
class="backtable">
<TR><TD height="70" class="color-footer"><!--#include file="include/top.aspx"--
></TD></TR>
<TR><TD height="20" class="color-row1"><asp:Label id="lblHelp" runat="server"
CssClass="txt-Title" Text=' University Information System Yardım' /></TD></TR>
<TR><TD align="center">
<br><br>
<table width="650" cellpadding="7" cellspacing="1">
  <tr class="color-row1">
    <td width="190" valign="baseline"><strong>· main.css</strong></td>
    <td>The main.css files which is at the bottom of style file includes the functions
and colors which are used.Css pages which includes &lt;link href= "style/main.css" rel
="stylesheet" type ="text/css&gt; code in pages provides to give table background
colours,writing,colours,and firms(shapes).Quailities of forms (shapes)are given by the
means of the quality which is written "class"in codes.<br></td>
  </tr>
  <tr class="color-row2">
    <td valign="baseline"><strong>· web.config</strong></td>
    <td>If someone does not want to expose to the problem of Turkish letters in
ASP.Net pages,web.config file which is defined Turkish letters should be in the main
file.If we don't have this file,we don't have any alternative ağırt from defined Turkish
```

```

<td valign="baseline"><strong> · login1.aspx</strong></td>
<td>It is the page which is done administrative logging in. Administrative staff's
passwords which belong to every faculty are defined. The staff selects the suitable
faculty and password. After doing this, he has all rights about the faculty.<br></td>
</tr>
<tr class="color-row2">
<td valign="baseline"><strong> · liste_admin.aspx(*)</strong></td>
<td>This page is only used by the system administrator. The password which
belong to the administration are determined in liste_admin.aspx page. The passwords
which the administration use are saved, changed or deleted in this
page.(liste_admin.aspx). Lots of passwords can be given just for one faculty.<br></td>
</tr>
<tr class="color-row1">
<td valign="baseline"><strong> · liste_donem.aspx(*)</strong></td>
<td>This page is only used by the system administrator, too. There is information of
education in this page. This page is determined between whichever term it is or
whether it belongs to fall or spring term. You can enter new terms. The registered terms
can be changed or deleted.<br></td>
</tr>
<tr class="color-row2">
<td valign="baseline"><strong> · liste_gun.aspx(*)</strong></td>
<td>The System administrator can use this page, too. There are lots of registrations
about the days which the university has lessons while preparing the lesson Schedules ,
these days are kept in mind.<br></td>
</tr>
<tr class="color-row1">
<td valign="baseline"><strong> · liste_saat.aspx(*)</strong></td>
<td>The system administrator can use this page, too. Lesson duration is saved as
hour :minute.<br></td>
</tr>
<tr class="color-row2">
<td valign="baseline"><strong> · logout.aspx</strong></td>

```

<td>Logging out is done in logout.aspx page.Students,teachers,administrative staff and system administer use this file to logo ut and they are directed to the main page.
</td>

</tr>

<tr class="color-row1">

<td valign="baseline">· liste_fakulte.aspx</td>

<td>Even if the main page of the system is default.aspx ,there are beginning of the system definitions and its branches in this page.Liste_fakulte.aspx determine the faculties and campuses of university by taking from the database.The departments which belong to faculties and the teachers of that faculties can be found.If logging in of system administer is done , a new faculty can be added or the faculty which is already saved can be changed or deleted.</td>

</tr>

<tr class="color-row2">

<td valign="baseline">· liste_bolum.aspx</td>

<td>There are lots of selected departments which belong to a faculty and there are also lots of information about these departments.The classes which belong to the departments or the lessons which are given in that department are found in liste_bolum.aspx page.If logging in of system administer or logging in of the stuff of that faculty is done, a new department can be added to the new faculty and te departments can be changed or deleted.</td>

</tr>

<tr class="color-row1">

<td valign="baseline">· liste_ders.aspx</td>

<td>It is the list of the department lessons.It shows that which term the lessons are given and it also shows the credits of lessons.</td>

</tr>

<tr class="color-row2">

<td valign="baseline">· liste_sinif.aspx</td>

<td>The classes in liste_bolum.aspx page which belong to the selected department are shown in that page.The class list can also be found in that page.</td>

</tr>

<tr class="color-row1">

<td valign="baseline">· liste_ogrenci.aspx</td>

<td>The list which belongs to the class is shown in liste_ogrenci.aspx page. The student numbers , names,surnames and lesson scheldule can be found in this page.If the student log in,he can see his own details.But he can see other student's details.If the system administer log in or the staff log in,the student in list can be deleted or a new student can be added.</td>

</tr>

<tr class="color-row2">

<td valign="baseline">· liste_ogretmen.aspx</td>

<td>It lists the teachers of faculty .The teacher's code ,name and position are determined.The teacher can see only his own details and information like the student list page in liste_ogretmen.aspx.If the system administer or the administrative staff which belongs to the faculty logs in,the techer can be deleted or a new teacher can be added to this faculty.</td>

</tr>

<tr class="color-row1">

<td valign="baseline">· detay_ogrenci.aspx</td>

<td>Information about students is shown in detay_ogrenci.aspx.Lesson Marks,lesson schedules are directed in this page.</td>

</tr>

<tr class="color-row2">

<td valign="baseline">· detay_ogretmen.aspx</td>

<td>Information about the teacher is shownin detay_ogrenci.aspx .The lessons which are taken and lesson Schedule are directed in this page.</td>

</tr>

<tr class="color-row1">

<td valign="baseline">· ekle_ogrenci.aspx</td>

<td>A new student is defined to the class in this page .Information about the teacher is registered in this page (only,the system administor or the administrative faculty can reach the information)
</td>

</tr>

<tr class="color-row2">

<td valign="baseline">· ekle_ogretmen.aspx</td>

<td>A new teacher is defined to the faculty in this page .Information about the teacher is registered in this page (only,the system administor or the administrative faculty can reach the information)
</td>

</tr>

<tr class="color-row1">

<td valign="baseline">· guncelle_ogrenci.aspx</td>

<td>The student which is registered in advanced can be changed in this page.(Only the system administor or the administrative faculty can reach the infrmation)
</td>

</tr>

<tr class="color-row2">

<td valign="baseline">· guncelle_ogretmen.aspx</td>

<td>The teacher which is registered in advenced can be changed in this page.(only th system administor or the administrative faculty can reach the information)
</td>

</tr>

<tr class="color-row1">

<td valign="baseline">· ogretmen_ders.aspx</td>

<td>The lessons which the teacher siyes in that term are listd and determined in this page.The lessons can not be determined by the teachers.The system administer or faculty does it.The users can see the list and check the students which take the lessons.
</td>

</tr>

<tr class="color-row2">

<td valign="baseline">· ogretmen_ders_ogrenci.aspx</td>

<td>The students who attend the lesson which the teacher gives are shown in this page.If there is not any logging in,the students's marks and their attendance can be obseved,changed or added in the page of the student's number and name.</td>

</tr>

<tr class="color-row1">

<td valign="baseline">· ogretmen_program.aspx</td>

<td>The teacher can see the lesson Schedule.If the term is suitable(the lesson registration is contaioning),the teacher can prepare a new Schedule and he can change

the lesson Schedule and he can change the lesson Schedule which is done before.
</td>

</tr>

<tr class="color-row2">

<td valign="baseline">ogrenci_ders.aspx</td>

<td>The lessons which the student take is shown in this page.If the term is suitable like the page of ogretmen_program.aspx,the student can choose the lessons in this page.He can choose the lesson according to the teacher who give the lesson or the Schedule which the teacher prepare.If the lesson registration is containing,he can choose the lesson he wants and he can change.</td>

</tr>

<tr class="color-row1">

<td valign="baseline">ogrenci_program.aspx</td>

<td>The automatic lesson Schedule is prepared according to the student who choose the lessons.Because,that lesson which is given by the teacher is determined which day it will be or what time it will be .According to this criteria,the unit plan of the student is shown in this page.
</td>

</tr>

<tr class="color-row2">

<td valign="baseline">ogrenci_ders_program.aspx</td>

<td>The Marks and attendance of the students are shown in this page.
</td>

</tr>

<tr class="color-row1">

<td valign="baseline">help.aspx</td>

<td>It is a hepler page to get information about the university information system.
</td>

</tr>

<tr class="color-row2">

<td valign="baseline">hata.aspx</td>

<td>It is an error page.When the user try to log in the page which is not permitted , he received an error message.
</td>

</tr>

</table>


```

</TD></TR>
<TR><TD height="20" class="color-footer"><!--#include file="include/bottom.aspx"-->
</TD></TR>
</TABLE>
</body>
</html>

```

detay_ogrenci.aspx

```

<!-- University Information System - Şeniz Varol 20041170 -->

<%@ Page Language="VB" Debug="true" codePage="28599" %>
<%@ Import Namespace="System.Data.OleDb" %>

<script runat="server">
Dim bag as new OleDbConnection("Provider=Microsoft.Jet.OLEDB.4.0; DATA
Source=" & Server.MapPath("db/veri.mdb"))

Sub Page_Load(Sender as Object, E as EventArgs)
Dim Id as integer
Id = request.QueryString("ogrenci")

bag.open()
Dim infoCommand as new OleDbCommand("Select
FAKULTE_TANIM.Fakulte_Id,Fakulte_Ad,Bolum_Ad,OGRENCI_TANIM.* From
FAKULTE_TANIM,BOLUM_TANIM,OGRENCI_TANIM where Ogrenci_Kayit= "&
Id &" and BOLUM_TANIM.Bolum_Id = OGRENCI_TANIM.Bolum_Id and
FAKULTE_TANIM.Fakulte_Id = BOLUM_TANIM.Fakulte_Id" , bag )
Dim infoReader As OleDbDataReader = infoCommand.ExecuteReader()
Dim Sinif As String
Dim SinifNo as integer
infoReader.Read()
If infoReader.GetInt16(13).ToString = 0 Then

```

```

Sinif = "Prep Class"
SinifNo = 0
Else
Sinif = Cint(infoReader.GetInt16(13).ToString/2) & ". Class"
SinifNo = Cint(infoReader.GetInt16(13).ToString/2)
End If
lblNo.text = infoReader.GetString(4)
lblAdSoyad.text = infoReader.GetString(6) & " " & infoReader.GetString(7)
lblCinsiyet.text = Yazi(infoReader.GetString(8))
lblDog_T.text = Yazi(infoReader.GetString(9))
lblTel.text = Yazi(infoReader.GetString(10))
lblMail.text = Yazi(infoReader.GetString(11))
lblFakulte.Text = "<a href='liste_fakulte.aspx' class='txt-Title'>Faculties</a> <img
src='MyIcon/go.gif'> " + "<a href='liste_bolum.aspx?fakulte=" +
infoReader.GetInt32(0).ToString() + "' class='txt-Title'>" + infoReader.GetString(1) +
"</a> <img src='MyIcon/go.gif'> " + "<a href='liste_sinif.aspx?bolum=" +
infoReader.GetInt32(14).ToString() + "' class='txt-Title'>" + infoReader.GetString(2) +
" Dept </a> <img src='MyIcon/go.gif'> <a href='liste_ogrenci.aspx?bolum=" +
infoReader.GetInt32(14).ToString() + "&sinif=" + SinifNo.ToString + "' class='txt-
title'>" + Sinif + "</a>" + " <img src='MyIcon/go.gif'> " & infoReader.GetString(6) & "
"& infoReader.GetString(7)
picture.ImageUrl="Picture1/" & infoReader.GetString(4) & ".jpg"

```

'Admin Kontrolleri

```

If Not (session("Admin") = "OK" or Session("Idare_Id") =
infoReader.GetInt32(0).ToString() or Session("Ogretmen") = "OK" or
Session("Ogrenci_Kayit") = infoReader.GetInt32(3).ToString() ) Then
Response.Redirect("hata.aspx?id=100")
Else If Session("Ogretmen") = "OK" or Session("Ogrenci_Kayit") =
infoReader.GetInt32(3).ToString() Then
linkUpdate.visible = "false"
End If

```

```

infoReader.close()

```

```
bag.close()
```

```
linkUpdate.NavigateUrl = "guncelle_ogrenci.aspx?ogrenci=" & Id.ToString  
linkNotlar.NavigateUrl = "ogrenci_ders_ogretmen.aspx?ogrenci=" & Id.ToString  
linkNotlar1.NavigateUrl = "ogrenci_ders_ogretmen.aspx?ogrenci=" & Id.ToString  
linkDersler.NavigateUrl = "ogrenci_ders.aspx?ogrenci=" & Id.ToString  
linkDersler1.NavigateUrl = "ogrenci_ders.aspx?ogrenci=" & Id.ToString  
linkProgram.NavigateUrl = "ogrenci_program.aspx?ogrenci=" & Id.ToString  
linkProgram1.NavigateUrl = "ogrenci_program.aspx?ogrenci=" & Id.ToString
```

```
End Sub
```

```
Function Yazi(txt as string) as String
```

```
If txt = "" Then
```

```
return "No Comment"
```

```
Else
```

```
return txt
```

```
End If
```

```
End Function
```

```
</script>
```

```
<html>
```

```
<link href="style/main.css" rel="stylesheet" type="text/css">
```

```
<head>
```

```
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-9">
```

```
</head>
```

```
<body>
```

```
<TABLE width="770" height="100%" align="center" cellpadding="0" cellspacing="0"  
class="backtable">
```

```
<TR><TD height="70" class="color-footer"><!--#include file="include/top.aspx"--  
></TD></TR>
```

```
<TR><TD height="20" class="color-row1"><asp:label id="lblFakulte" CssClass="txt-  
Title" runat="server" /></TD></TR>
```

```
<TR><TD align="center" valign="top">
```



```

<table width="100%">
<tr>
<td valign="top">
<table>
<tr><td><asp:Label id="lblNo" runat="server" CssClass="txt-Title" /></td></tr>
<tr><td><asp:Label id="lblAdSoyad" runat="server" CssClass="txt-Title" /></td></tr>
</table>
<td width="90" height="110" valign="top" class="formtable"><asp:image ID="picture"
Width="90" Height="110" runat="server" /></td>
</tr>
</table>
<br>
<table class="formtable">
<tr class="color-title">
<td colspan="4">
<table cellpadding="0" cellspacing="0" width="100%">
<tr>
<td class="txt-title"> PERSONEL
INFO</td>
<td align="right"><asp:hyperlink ID="linkUpdate" ImageUrl="MyIcon/update.gif"
Title="Update Info" runat="server" /></td>
</tr>
</table>
</td>
</tr>
<tr>
<td colspan="2">
<tr>
<td width="100">
<strong>Sex</strong></td>
<td width="120"><asp:Label id="lblCinsiyet" runat="server" /></td>
<td width="100"> <strong>Birth
Date</strong></td>
<td width="120"><asp:Label id="lblDog_T" runat="server" /></td>
</tr>

```

```

<tr>
<td> <strong>Phone</strong></td>
<td><asp:Label id="lblTel" runat="server" /></td>
<td> <strong>Mail</strong></td>
<td><asp:Label id="lblMail" runat="server" /></td>
</tr>
</table>
<br>
<table class="formtable">
<tr class="color-title">
<td colspan="6" class="txt-title">
EDUCATION INFO</td>
</tr>
<tr>
<td width="15">
<asp:hyperlink ID="linkNotlar" ImageUrl="MyIcon/settings.gif"
Title="Marks List" runat="server" />
</td>
<td width="126">
<asp:hyperlink ID="linkNotlar1" Text="Marks List" runat="server"
CssClass="txt-item" />
</td>
<td width="15">
<asp:hyperlink ID="linkDersler" ImageUrl="MyIcon/notepad.gif"
Title="Courses" runat="server" />
</td>
<td width="126">
<asp:hyperlink ID="linkDersler1" Text="Courses" runat="server"
CssClass="txt-item" />
</td>
<td width="15">
<asp:hyperlink ID="linkProgram" ImageUrl="MyIcon/documents.gif"
Title="Course Program" runat="server" />
</td>

```

```

        <td width="126">
            <asp:hyperlink ID="linkProgram1" Text="Course Program"
runat="server" CssClass="txt-item" />
        </td>
    </tr>
</table>

</TD></TR>
<TR><TD height="20" class="color-footer"><!--#include file="include/bottom.aspx"-->
</TD></TR>
</TABLE>
</body>
</html>

```

detay_ogretmen.aspx

```

<!-- University Information System - Şeniz Varol 20041170 -->

<%@ Page Language="VB" Debug="true" codePage="28599" %>
<%@ import Namespace="System.Data.OleDb" %>

<script runat="server">
    Dim bag as new OleDbConnection("Provider=Microsoft.Jet.OLEDB.4.0; DATA
    Source=" & Server.MapPath("db/veri.mdb"))

    Sub Page_Load(Sender as Object, E as EventArgs)
        Dim Id as string
        Id = request.QueryString("ogretmen")

        bag.open()
        Dim infoCommand as new OleDbCommand("Select
        Fakulte_Ad,OGRETMEN_TANIM.* From

```



```
FAKULTE_TANIM,OGRETMEN_TANIM where Ogretmen_Kayit= "& Id &" and
FAKULTE_TANIM.Fakulte_Id = OGRETMEN_TANIM.Fakulte_Id" , bag )
Dim infoReader As OleDbDataReader = infoCommand.ExecuteReader()
infoReader.Read()
```

```
lblId.text = infoReader.GetString(2)
lblAdSoyad.text = infoReader.GetString(4) & " " & infoReader.GetString(5)
lblUnvan.text = infoReader.GetString(7)
lblCinsiyet.text = Yazi(infoReader.GetString(6))
lblDog_T.text = Yazi(infoReader.GetString(8))
lblTel.text = Yazi(infoReader.GetString(9))
lblMail.text = Yazi(infoReader.GetString(10))
picture.ImageUrl="Picture2/" & infoReader.GetString(2) & ".jpg"
lblFakulte.Text = "<a href='liste_fakulte.aspx' class='txt-Title'>Faculties</a> <img
src='MyIcon/go.gif'> <a href='liste_bolum.aspx?fakulte=" +
infoReader.GetInt32(12).ToString + "' class='txt-title'>" + infoReader.GetString(0)
+"</a> <img src='MyIcon/go.gif'> <a href='liste_ogretmen.aspx?fakulte=" +
infoReader.GetInt32(12).ToString + "' class='txt-title'>Lecturers</a> <img
src='MyIcon/go.gif'> " & infoReader.GetString(4) & " " & infoReader.GetString(5)
```

'Admin Kontrolleri

```
If Not (Session("Admin") = "OK" or Session("Idare_Id") =
infoReader.GetInt32(12).ToString() or Session("Ogretmen_Kayit") =
infoReader.GetInt32(1).ToString() ) Then
Response.Redirect("hata.aspx?id=200")
Else If Session("Ogrenci") = "OK" or Session("Ogretmen_Kayit") =
infoReader.GetInt32(1).ToString() Then
linkUpdate.visible = "false"
End If
```

```
infoReader.close()
```

```
bag.close()
```

```
linkUpdate.NavigateUrl = "guncelle_ogretmen.aspx?ogretmen=" & Id.ToString
```

```

linkDersler.NavigateUrl = "ogretmen_ders.aspx?ogretmen=" & Id.ToString
linkDersler1.NavigateUrl = "ogretmen_ders.aspx?ogretmen=" & Id.ToString
linkProgram.NavigateUrl = "ogretmen_program.aspx?ogretmen=" & Id.ToString
linkProgram1.NavigateUrl = "ogretmen_program.aspx?ogretmen=" & Id.ToString

```

```
End Sub
```

```
Function Yazı(txt as string) as String
```

```
If txt = "" Then
```

```
return "-"
```

```
Else
```

```
return txt
```

```
End If
```

```
End Function
```

```
</script>
```

```
<html>
```

```
<link href="style/main.css" rel="stylesheet" type="text/css">
```

```
<head>
```

```
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-9">
```

```
</head>
```

```
<body>
```

```
<TABLE width="770" height="100%" align="center" cellpadding="0" cellspacing="0"
class="backtable">
```

```
<TR><TD height="70" class="color-footer"><!--#include file="include/top.aspx"--
></TD></TR>
```

```
<TR><TD height="20" class="color-row1"><asp:label id="lblFakulte" CssClass="txt-
Title" runat="server" /></TD></TR>
```

```
<TR><TD align="center" valign="top">
```

```
<table width="100%">
```

```
<tr>
```

```
<td valign="top">
```

```
<table>
```

```

<tr><td><asp:Label id="lblId" runat="server" CssClass="txt-Title" /></td></tr>
<tr><td><asp:Label id="lblAdSoyad" runat="server" CssClass="txt-Title" /></td></tr>
<tr><td><asp:Label id="lblUnvan" runat="server" CssClass="txt-Title" /></td></tr>
</table>
<td width="90" height="110" valign="top" class="formtable"><asp:image ID="picture"
Width="90" Height="110" runat="server" /></td>
</tr>
</table>
<br>
<table class="formtable">
<tr class="color-title">
<td colspan="4">
<table cellpadding="0" cellspacing="0" width="100%">
<tr>
<td class="txt-title"> PERSONEL
INFORMATION</td>
<td align="right"><asp:hyperlink ID="linkUpdate" ImageUrl="MyIcon/update.gif"
Title="Bilgileri Güncelle" runat="server" /></td>
</tr>
</table>
</td>
</tr>
<tr>
<td>
<tr>
<td width="100">
<strong>Sex</strong></td>
<td width="120"><asp:Label id="lblCinsiyet" runat="server" /></td>
<td width="100"> <strong>Birth
Date</strong></td>
<td width="120"><asp:Label id="lblDog_T" runat="server" /></td>
</tr>
<tr>
<td> <strong>Phone</strong></td>
<td><asp:Label id="lblTel" runat="server" /></td>
<td> <strong>Mail</strong></td>

```



```

        <td><asp:Label id="lblMail" runat="server" /></td>
    </tr>
</table>
<br>
<table class="formtable">
<tr class="color-title">
<td colspan="8" class="txt-title">
WORK INFORMATION</td>
</tr>

        <tr>
            <td width="21">
                <asp:hyperlink ID="linkDersler" ImageUrl="MyIcon/notepad.gif"
Title="Courses" runat="server" />
            </td>
            <td width="200">
                <asp:hyperlink ID="linkDersler1" Text="Courses" runat="server"
CssClass="txt-item" />
            </td>
            <td width="21">
                <asp:hyperlink ID="linkProgram" ImageUrl="MyIcon/documents.gif"
Title="Course Program" runat="server" />
            </td>
            <td width="200">
                <asp:hyperlink ID="linkProgram1" Text="Course Program"
runat="server" CssClass="txt-item" />
            </td>
        </tr>
</table>

</TD></TR>
<TR><TD height="20" class="color-footer"><!--#include file="include/bottom.aspx"-->
</TD></TR>
</TABLE>
</body>

```

</html>

ekle_ogrenci.aspx

<!-- University Information System - Şeniz Varol 20041170 -->

<%@ Page Language="VB" Debug="true" codePage="28599" %>

<%@ import Namespace="System.Data.OleDb" %>

<script runat="server">

Dim bag as new OleDbConnection("Provider=Microsoft.Jet.OLEDB.4.0; DATA
Source=" & Server.MapPath("db/veri.mdb"))

Sub Page_Load(Sender as Object, E as EventArgs)

Dim Bolum , Sinif as integer

Dim Sinif_Ad as String

Bolum = request.QueryString("bolum")

Sinif = request.QueryString("sinif")

If Sinif = 0 Then

Sinif_Ad = "Prep Class"

Else

Sinif_Ad = Sinif.ToString & ". Class"

End If

bag.open()

Dim infoCommand as new OleDbCommand("Select

BOLUM_TANIM.Fakulte_Id,Fakulte_Ad,Bolum_Ad From

FAKULTE_TANIM,BOLUM_TANIM where Bolum_Id= "& Bolum &" and

BOLUM_TANIM.Fakulte_Id = FAKULTE_TANIM.Fakulte_Id" , bag)

Dim infoReader As OleDbDataReader = infoCommand.ExecuteReader()

infoReader.Read()

lblFakulte.Text = "Faculties <img

src='MyIcon/go.gif'> " + "<a href='liste_bolum.aspx?fakulte=" +

infoReader.GetInt32(0).ToString() + "' class='txt-Title'>" + infoReader.GetString(1) +

" " + "<a href='liste_sinif.aspx?bolum=" +

```
Bolum.ToString + " class='txt-Title'" + infoReader.GetString(2) + " Dept </a> <img  
src='MyIcon/go.gif'> <a href='liste_ogrenci.aspx?bolum=" + Bolum.ToString +  
"&sinif=" + Sinif.ToString + " class='txt-title'" + Sinif_Ad + "</a>" + " <img  
src='MyIcon/go.gif'> Add New Student"
```

```
'Admin Kontrolleri
```

```
If Not (session("Admin") = "OK" or Session("Idare_Id") =
```

```
infoReader.GetInt32(0).ToString() ) Then
```

```
Response.Redirect("hata.aspx?id=100")
```

```
End If
```

```
infoReader.close()
```

```
bag.close()
```

```
End Sub
```

```
Sub Ekle (obj as object, e as eventargs)
```

```
Dim Ogrenci_No,Sifre,Ad,Soyad,Cinsiyet,Dog_T,Tel,Mail,Resim as string
```

```
Dim Donem,Bolum as integer
```

```
Ogrenci_No = newOgrenci_No.text
```

```
Sifre = newSifre.text
```

```
Ad = newAd.text
```

```
Soyad = newSoyad.text
```

```
Cinsiyet = newCinsiyet.SelectedItem.text
```

```
Dog_T = newDog_T.text
```

```
Tel = newTel.text
```

```
Mail = newMail.text
```

```
Resim = newOgrenci_No.text & ".jpg"
```

```
Donem = Cint(request.QueryString("sinif")*2)
```

```
Bolum = request.QueryString("bolum")
```

```
bag.Open()
```



```

Dim db_komut As New OleDbCommand( "Insert INTO OGRENCI_TANIM (
Ogrenci_No, Sifre , Ad, Soyad , Cinsiyet , Dog_T , Tel , Mail , Resim , Donem ,
Bolum_Id ) Values( '"& Ogrenci_No &"','& Sifre &"','& Ad &"','& Soyad &"','&
Cinsiyet &"','& Dog_T &"','& Tel &"','& Mail &"','& Resim &"','& Donem
&"','& Bolum &"' )", bag )
On Error Resume Next
db_komut.ExecuteNonQuery()

```

```

if not err.Number = 0 then
lblDurum.text = "<B>Error No: </B>"& err.Number & " - <B>Description: </B>"&
err.Description
else
lblDurum.text = "<B>RECORD ADDED!</B>"
end if
bag.Close()

```

End Sub

</script>

<html>

<link href="style/main.css" rel="stylesheet" type="text/css">

<head>

<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-9">

</head>

<body>

<TABLE width="770" height="100%" align="center" cellpadding="0" cellspacing="0"
class="backtable">

<TR><TD height="70" class="color-footer"><!--#include file="include/top.aspx"--
></TD></TR>

<TR><TD height="20" class="color-row1"><asp:Label id="lblFakulte" runat="server"
CssClass="txt-Title" /></TD></TR>

<TR><TD align="center" valign="top">

<form runat="server" ID="Form1">

<table class="formtable">

```

<tr class="color-title">
<td colspan="2" class="txt-title">ADD NEW STUDENT</td>
</tr>

<tr>
<td width="100" valign="baseline">Number</td>
<td width="145"><asp:TextBox id="newOgrenci_No" Width="140"
MaxLength="15" runat="server" />
<asp:RequiredFieldValidator id="NoReqVal"
ControlToValidate="newOgrenci_No"
ErrorMessage="<br>Enter Student Number"
Display="Dynamic"
runat="server" />
</td>
</tr>
<tr>
<td valign="baseline">Password</td>
<td><asp:TextBox TextMode="Password" Width="140" id="newSifre"
MaxLength="20" runat="server" />
<asp:RequiredFieldValidator id="passwdReqVal"
ControlToValidate="newSifre"
ErrorMessage="<br>Enter Password"
Display="Dynamic"
runat="server" />
</td>
</tr>
<tr>
<td valign="baseline">Password (Again)</td>
<td><asp:TextBox TextMode="Password" Width="140" id="newSifre1"
MaxLength="20" runat="server" />
<asp:RequiredFieldValidator id="passwdReqVal1"
ControlToValidate="newSifre1"
ErrorMessage="<br>Re Enter Password"
Display="Dynamic"
runat="server" />

```

```

<asp:CompareValidator id="CompareValidator1"
    ControlToValidate="newSifre1" ControlToCompare="newSifre"
    ErrorMessage="<br>Check Passwords"
    Display="Dynamic"
    runat=server />
        </td>
    </tr>
    <tr>
        <td valign="baseline">Name</td>
        <td><asp:TextBox id="newAd" Width="140" MaxLength="20"
runat="server" />
        <asp:RequiredFieldValidator id="nameReqVal"
            ControlToValidate="newAd"
            ErrorMessage="<br>Enter Name"
            Display="Dynamic"
            runat=server />
                </td>
            </tr>
            <tr>
                <td valign="baseline">Surname</td>
                <td><asp:TextBox id="newSoyad" Width="140" MaxLength="20"
runat="server" />
                <asp:RequiredFieldValidator id="surnameReqVal"
                    ControlToValidate="newSoyad"
                    ErrorMessage="<br>Enter Surname"
                    Display="Dynamic"
                    runat=server />
                        </td>
                    </tr>
                    <tr>
                        <td>Sex</td>
                        <td><asp:radiobuttonlist id="newCinsiyet"
RepeatDirection="Horizontal" runat="server">
                            <asp:listitem Text="Male" selected />

```



```

        <asp:listitem Text="Female" />
    </asp:radiobuttonlist></td>
</tr>
<tr>
    <td valign="baseline">Birth Date</td>
    <td><asp:TextBox id="newDog_T" Width="140" runat="server" />
    <asp:RegularExpressionValidator id="dateRegexVal"
        ControlToValidate="newDog_T"
        ErrorMessage="<br>Date Must Be DD.MM.YYYY"
        ValidationExpression="[0-9]{2}\.[0-9]{2}\.[1-2]{1}[0-9]{3}"
        Display="Dynamic"
        runat=server />
    </td>
</tr>
<tr>
    <td valign="baseline">Phone</td>
    <td><asp:TextBox id="newTel" Width="140" runat="server" />
    <asp:RegularExpressionValidator id="phoneRegexVal"
        ControlToValidate="newTel"
        ErrorMessage="<br>Phone Must Be (XXX) XXXXXXXX"
        ValidationExpression="(^x\s*[0-9]{5}$)|(^(\([1-9][0-9]{2}\)\s)?[1-9]{1}[0-9]{6}(\s*x\s*[0-9]{5})?$)"
        Display="Dynamic"
        runat=server />
    </td>
</tr>
<tr>
    <td valign="baseline">Mail</td>
    <td><asp:TextBox id="newMail" Width="140" MaxLength="30"
runat="server" />
    <asp:RegularExpressionValidator id="emailRegexVal"
        ControlToValidate="newMail"
        ErrorMessage="<br>Enter a Valid Email"
        Display="Dynamic"

```

```

ValidationExpression="^[\\w-]+@[\\w-]+\\. (com|net|org|edu|mil|edu.tr|com.tr)$"
runat=server />
        </td>
    </tr>
    <tr>
        <td align="right" colspan="2"><asp:Button id="buton" runat="server"
Text="Add" Width="60" OnClick="ekle" /></td>
    </tr>
</table>
</form>
<asp:label id="lblDurum" runat="server" Font-Name="verdana" Font-Size="8pt" />
</TD></TR>
<TR><TD height="20" class="color-footer"><!--#include file="include/bottom.aspx"-->
</TD></TR>
</TABLE>
</body>
</html>

```

login.aspx

```

<!-- University Information System - Şeniz Varol 20041170 -->

<%@ Page Language="VB" Debug="true" codePage="28599" %>
<%@ import Namespace="System.Data.OleDb" %>
<script runat="server">
Dim bag as new OleDbConnection("Provider=Microsoft.Jet.OLEDB.4.0; DATA
Source=" & Server.MapPath("db/veri.mdb"))

Sub Giris(Sender as Object, e As System.Web.UI.ImageClickEventArgs)
    bag.open()

    IF kullanici_tip.SelectedItem.Value = "tip1" THEN
        Dim infoCommand as new OleDbCommand("Select Sifre , Ogrenci_Kayit From
        OGRENCI_TANIM Where Ogrenci_No = '" & Kullanici_No.Text & "'", bag )

```

```

Dim infoReader As OleDbDataReader = infoCommand.ExecuteReader()
If infoReader.Read() Then
    If Sifre.Text = infoReader.GetString(0) Then
        Session("Ogrenci") = "OK"
        Session("Ogrenci_Kayit") = infoReader.GetInt32(1).ToString
        Response.Redirect("detay_ogrenci.aspx?ogrenci=" & infoReader.GetInt32(1).ToString)
    Else
        lblDurum.Text = "Wrong Number of Password"
    End If
Else
    lblDurum.Text = "Wrong Number of Password"
End If
infoReader.Close()
ELSEIF kullanici_tip.SelectedItem.Value = "tip2" THEN
    Dim infoCommand as new OleDbCommand("Select Sifre , Ogretmen_Kayit From
    OGRETMEN_TANIM Where Ogretmen_Kod = '" & Kullanici_No.Text & "'", bag )
    Dim infoReader As OleDbDataReader = infoCommand.ExecuteReader()
    If infoReader.Read() Then
        If Sifre.Text = infoReader.GetString(0) Then
            Session("Ogretmen") = "OK"
            Session("Ogretmen_Kayit") = infoReader.GetInt32(1).ToString
            Response.Redirect("detay_ogretmen.aspx?ogretmen=" &
            infoReader.GetInt32(1).ToString)
        Else
            lblDurum.Text = "Wrong Number of Password"
        End If
    Else
        lblDurum.Text = "Wrong Number of Password"
    End If
    infoReader.Close()
END IF

bag.close()
End Sub

```



```

</script>
<html>
<link href="style/main.css" rel="stylesheet" type="text/css">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-9">
</head>
<body>
<TABLE width="770" height="100%" align="center" cellpadding="0" cellspacing="0"
class="backtable">
<TR><TD height="70" class="color-footer"><!--#include file="include/top.aspx"--
></TD></TR>
<TR><TD height="20" class="color-row1"><asp:Label id="lblFakulte" runat="server"
CssClass="txt-Title" Text=' User Login'
/></TD></TR>
<TR><TD align="center" valign="top">
<form runat="server" ID="Form1">

<table class="formtable">
<tr class="color-title">
<td colspan="2" class="txt-title">USER LOGIN</td>
</tr>
<tr>
<td width="100" valign="baseline">Number</td>
<td width="145"><asp:TextBox id="Kullanici_No" Width="140"
MaxLength="15" runat="server" />
<asp:RequiredFieldValidator id="NoReqVal"
ControlToValidate="Kullanici_No"
ErrorMessage="<br>Enter User Number"
Display="Dynamic"
runat="server" />
</td>
</tr>
<tr>
<td valign="baseline">Password</td>

```

```

        <td><asp:TextBox TextMode="Password" Width="140" id="Sifre"
MaxLength="20" runat="server" />
        <asp:RequiredFieldValidator id="passwdReqVal"
        ControlToValidate="Sifre"
        ErrorMessage="<br>Enter Password"
        Display="Dynamic"
        runat="server" />
    </td>
</tr>
<tr>
<td colspan="2">
        <asp:radiobuttonlist ID="kullanici_tip" runat="server"
RepeatDirection="Horizontal">
            <asp:listitem Text="Student" Value="tip1" Selected />
            <asp:listitem Text="Lecturer" Value="tip2" />
        </asp:radiobuttonlist>
    </td>
</tr>
<tr>
        <td align="right" colspan="2"><asp:imagebutton
ImageUrl="MyIcon/ok.gif" ID="buton" runat="server" Title="Login" OnClick="Giris"
/></td>
    </tr>
</table>
</form>
<asp:label id="lblDurum" runat="server" Text=" " Font-Name="verdana" Font-
Size="8pt" />
</TD></TR>
<TR><TD height="20" class="color-footer"><!--#include file="include/bottom.aspx"--
></TD></TR>
</TABLE>
</body>
</html>

```

liste_admin.aspx

<!-- University Information System by Seniz Varol 20041170 -->

<%@ Page Language="Vb" CodePage="28599" debug="true" buffer="false" %>

<%@ import Namespace="System.Data" %>

<%@ import Namespace="System.Data.OleDb" %>

<script language="Vb" runat="server">

Dim SortField as String

Dim bag as new OleDbConnection("Provider=Microsoft.Jet.OLEDB.4.0; Data
Source=" & Server.MapPath("db/veri.mdb"))

' Database Veri Tanimlamalari

Sub veribagla

Dim adaptor as new OleDbDataAdapter ("Select Admin_Id ,
FAKULTE_TANIM.Fakulte_Id , Fakulte_Ad , Sifre From ADMIN_TANIM ,
FAKULTE_TANIM Where FAKULTE_TANIM.Fakulte_Id =
ADMIN_TANIM.Fakulte_Id" , bag)

Dim ds as new DataSet()

adaptor.fill(ds,"ADMIN_TANIM")

DataGrid1.DataSource = ds

End Sub

' Sayfa Yuklendiginde Yapilacaklar

Sub Page_Load(Sender as Object, E as EventArgs)

'Admin Tanimlamalari

If Not Session("Admin") = "OK" Then

response.Redirect("hata.aspx?id=300")

End If

lblGuncelle.Text=""

veribagla

If not IsPostBack then

DataGrid1.DataBind()

sayfa_no.Text = "Page: 1 /"

toplamsayfa.Text = DataGrid1.PageCount

End If

End Sub

' Sayfalama Fonksiyonu

Sub sayfala(Sender as Object, e as DataGridPageChangedEventArgs)

DataGrid1.CurrentPageIndex = e.NewPageIndex

DataGrid1.EditItemIndex = -1

DataGrid1.DataBind()

sayfa_no.Text = "Page: " & DataGrid1.CurrentPageIndex + 1 & " /"

toplamsayfa.Text = DataGrid1.PageCount

End Sub

' Secilen Satirin Duzenlenmesi

Sub Duzenle(Sender as Object, e as DataGridCommandEventArgs)

DataGrid1.EditItemIndex = e.Item.ItemIndex

DataGrid1.DataBind()

End Sub

' Secilen Satirdaki Verilerin Guncellenmesi

Sub Guncelle(sender As Object, e As DataGridCommandEventArgs)

Dim Fakulte1 As DropDownList = e.Item.Cells(1).Controls(1)

Dim A_Sifre1 As TextBox = e.Item.Cells(2).Controls(1)

Dim A_Id as integer = e.Item.Cells(0).text

Dim A_Sifre as String

Dim Fakulte as integer

A_Sifre = A_Sifre1.text

Fakulte = Fakulte1.SelectedItem.Value

```

bag.Open()
Dim db_update as new OleDbCommand("Update ADMIN_TANIM Set Sifre=""&
A_Sifre &"" , Fakulte_Id=""& Fakulte &"" Where Admin_Id=""& A_Id &"" , bag )
db_update.ExecuteNonQuery
bag.Close()
lblGuncelle.Text="<b>RECORD UPDATED!</b>"
DataGrid1.EditItemIndex = -1
veribagla
DataGrid1.DataBind()
End Sub

```

```

' Secilen Satirdan Vazgecilmesi
Sub Vazgec(Sender as Object, e as DataGridCommandEventArgs)
DataGrid1.EditItemIndex = -1
DataGrid1.DataBind()
End Sub

```

```

' Secilen Satirin Silinmesi
Sub Sil(Sender as Object, e as DataGridCommandEventArgs)
Dim A_Id as integer = e.Item.Cells(0).text
bag.Open()
Dim db_delete as new OleDbCommand("Delete From ADMIN_TANIM Where
Admin_Id=""& A_Id &"" , bag )
db_delete.ExecuteNonQuery
bag.Close()
lblGuncelle.Text="<b>RECORD DELETED!</b>"
DataGrid1.EditItemIndex = -1
veribagla
DataGrid1.DataBind()
End Sub

```

```

' Yeni Kayit Eklenmesi
Sub Ekle(Sender as Object, e as DataGridCommandEventArgs)
If e.CommandName = "Insert" Then

```

```

Dim Sifre As String = CType(e.Item.FindControl("newSifre"), TextBox).Text
Dim Fakulte As String = CType(e.Item.FindControl("newFakulte"),
DropDownList).SelectedItem.Value

```

```

If Sifre="" Then

```

```

    lblGuncelle.text="-Enter Password-<br>"

```

```

    Exit Sub

```

```

End If

```

```

    bag.Open()

```

```

    Dim db_insert as new OleDbCommand("Insert Into ADMIN_TANIM ( Sifre ,
Fakulte_Id ) values ('& Sifre &' , '& Fakulte &')", bag )

```

```

    db_insert.ExecuteNonQuery

```

```

    bag.Close()

```

```

    lblGuncelle.Text="<b>RECORD ADDED!</b>"

```

```

    DataGrid1.EditItemIndex = -1

```

```

    veribagla

```

```

    DataGrid1.DataBind()

```

```

End If

```

```

End Sub

```

```

Sub Olustur(Sender As Object, e As DataGridItemEventArgs)

```

```

    Select Case e.Item.ItemType

```

```

        Case ListItemType.Item, ListItemType.AlternatingItem

```

```

            Dim myDeleteButton As Button

```

```

            myDeleteButton = e.Item.FindControl("btnSil")

```

```

            myDeleteButton.Attributes.Add("onclick","return confirm('Are you sure you want to
delete this record?');")

```

```

        Case ListItemType.EditItem

```

```

            Dim myCancelButton As Button

```

```

            myCancelButton = e.Item.FindControl("btnVazgec")

```

```

            myCancelButton.Attributes.Add("onclick","history.go(-1);")

```

```

        End Select

```

```

    End Sub

```



```
' Admin Kontrolleri
```

```
Sub Ayarlar(sender as object, e as DataGridItemEventArgs)
```

```
If Not session("Admin") = "OK" Then
```

```
e.Item.Cells(0).visible = "true"
```

```
End If
```

```
End Sub
```

```
Function GetFakulte() as DataTable
```

```
Dim Adapter as new OleDbDataAdapter("SELECT Fakulte_Id , Fakulte_Ad FROM  
FAKULTE_TANIM Order By Fakulte_Ad", bag)
```

```
Dim ds as new DataSet()
```

```
Dim myTable as new DataTable("FAKULTE_TANIM")
```

```
myTable.Columns.Add("Fakulte_Id" , System.Type.GetType("System.Int32"))
```

```
myTable.Columns.Add("Fakulte_Ad" , System.Type.GetType("System.String"))
```

```
ds.Tables.Add(myTable)
```

```
Adapter.Fill(ds,"FAKULTE_TANIM")
```

```
return myTable
```

```
End Function
```

```
Function GetFakulteIndex(Fakulte as Integer) as Integer
```

```
Dim iLoop as Integer
```

```
Dim dt as DataTable = GetFakulte()
```

```
For iLoop = 0 to dt.Rows.Count - 1
```

```
If Fakulte = dt.Rows(iLoop)("Fakulte_Id") then
```

```
Return iLoop
```

```
End If
```

```
Next iLoop
```

```
End Function
```

```
</script>
```

```
<html>
```

```

<link href="style/main.css" rel="stylesheet" type="text/css">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-9; charset=iso-
8859-9">
</head>
<body>
<TABLE width="770" height="100%" align="center" cellpadding="0" cellspacing="0"
class="backtable">
<TR><TD height="70" class="color-footer"><!--#include file="include/top.aspx"--
></TD></TR>
<TR><TD height="20" class="color-row1"><asp:Label id="lblFakulte" runat="server"
CssClass="txt-Title" Text=' Administrative Staff Passwords' /></TD></TR>
<TR><TD align="center" valign="top">
<form runat="server">
<asp:DataGrid
    id="DataGrid1"
    runat="server"
    BorderColor="black"
    ItemStyle-Height="23"
    GridLines="vertical"
    Cellpadding="0"
    Cellspacing="0"
    HeaderStyle-CssClass="color-title"
    FooterStyle-CssClass="color-footer"
    HeaderStyle-Height="25"
    ItemStyle-CssClass="color-row1"
    AlternatingItemStyle-CssClass="color-row2"
    EditItemStyle-CssClass="color-edit"
    HeaderStyle-Font-Bold="true"
    ShowFooter="true"
    AutoGenerateColumns="false"
    OnItemCreated="Olustur"
    OnItemCommand="Ekle"

```

```

OnItemDataBound="Ayarlar"
AllowSorting="true"
AllowPaging="true"
PageSize="10"
PagerStyle-CssClass="tableyazi"
PagerStyle-Mode="NumericPages"
PagerStyle-HorizontalAlign="Right"
OnPageIndexChanged="sayfala"
OnEditCommand="Duzenle"
OnUpdateCommand="Guncelle"
OnCancelCommand="Vazgec"
OnDeleteCommand="Sil" >
<Columns>
    <asp:BoundColumn DataField="Admin_Id" ReadOnly="true" Visible="false"
/>
    <asp:templatecolumn HeaderText="Faculty" HeaderStyle-Width="200" >
        <itemtemplate>
            <asp:image ImageUrl="MyIcon/triangle.gif" runat="server" />
            <asp:hyperlink NavigateUrl='<%#
"liste_bolum.aspx?fakulte="&Container.DataItem("Fakulte_Id") %>' runat="server"
Text='<%# Container.DataItem("Fakulte_Ad") %>' Title='<%#
Container.DataItem("Fakulte_Ad")&" Bölümleri" %>' CssClass="txt-item" />
        </itemtemplate>
        <edititemtemplate>
            <asp:dropdownlist Id="Fakulte" Width="190" runat="server"
DataSource="<%# GetFakulte() %>"
DataValueField="Fakulte_Id"
DataTextField="Fakulte_Ad"
SelectedIndex='<%#
GetFakulteIndex(Container.DataItem("Fakulte_Id")) %>' />
        </edititemtemplate>
        <FooterTemplate>
            <asp:dropdownlist Id="newFakulte" Width="190" runat="server"
DataSource="<%# GetFakulte() %>"

```

```

        DataValueField="Fakulte_Id"
        DataTextField="Fakulte_Ad" />
    </FooterTemplate>
</asp:templatecolumn>
<asp:templatecolumn HeaderText="Password" HeaderStyle-Width="120" >
    <itemtemplate>
        <asp:Label runat="server" Text='<%# Container.DataItem("Sifre") %>'
/>
    </itemtemplate>
    <edititemtemplate>
        <asp:TextBox Id="Sifre" Width="110" runat="server" Text='<%#
Container.DataItem("Sifre") %>' MaxLength="20" />
        <asp:RequiredFieldValidator Runat="server" ControlToValidate="Sifre"
Display="Dynamic" ErrorMessage="*" />
    </edititemtemplate>
    <FooterTemplate>
        <asp:TextBox id="newSifre" Width="110" runat="server"
MaxLength="20" />
    </FooterTemplate>
</asp:templatecolumn>
<asp:templatecolumn HeaderText="Options" HeaderStyle-Width="120"
ItemStyle-HorizontalAlign="center" FooterStyle-HorizontalAlign="center">
    <itemtemplate>
        <asp:button runat="server" Text="Modify" CommandName="Edit"
Width="50" />
        <asp:button runat="server" ID="btnSil" Text="Delete"
CommandName="Delete" Width="50" />
    </itemtemplate>
    <edititemtemplate>
        <asp:button runat="server" Text="Update" CommandName="Update"
Width="50" />
        <asp:button runat="server" ID="btnVazgec" Text="Cancel"
CommandName="Cancel" Width="50" />
    </edititemtemplate>

```



```

        <FooterTemplate>
            <asp:button runat="server" Text="Add New Record"
CommandName="Insert" Width="103" />
        </FooterTemplate>
    </asp:templatecolumn>
</Columns>
</asp:DataGrid>
<asp:Label id="sayfa_no" runat="server" Font-Name="verdana" Font-Size="8pt" />
<asp:Label id="toplamsayfa" runat="server" Font-Name="verdana" Font-Size="8pt" />
<br>
<asp:Label id="lblGuncelle" runat="server" Font-Name="verdana" Font-Size="8pt" />
</form>
</TD></TR>
<TR><TD height="20" class="color-footer"><!--#include file="include/bottom.aspx"--
></TD></TR>
</TABLE>
</body>
</html>

```

CONCLUSION

In briefly, This project is created to see the details of the teachers' and students' details of each faculty of a university. Also this project provides new technology .Net. So ,the whole information is reached fast and in a security way without interruption.

REFERENCES

[W3C-01] Gudgin, Martin, et al., "SOAP Version 1.2," W3C Working Draft, July 2001.
www.w3.org/TR/2001/WD-soap12-20010709/

[Os0-00] Osborn, John, "Deep Inside C#: An Interview with Microsoft Chief Architect Anders

Hejlsberg," July 2000.

windows.oreilly.com/news/hejlsberg_0800.html

[Pat-01a] Pattison, Ted, "Basic Instincts: New Features in Visual Basic .NET," *MSDN Magazine*,

May 2001.

msdn.microsoft.com/msdnmag/issues/01/05/instincts/instincts0105.asp

[Pat-01b] Pattison, Ted, "Basic Instincts: Exploiting New Language Features in Visual Basic .NET,

Part 2," *MSDN Magazine*, August 2001.

msdn.microsoft.com/msdnmag/issues/01/08/Instincts/Instincts0108.asp

Mastering™ Visual Basic® .NET

Database Programming

Evangelos Petroustos; Asli Bilgin