Project report within Computer Engineering
C, course, 15 points

Web application development with .NET

3-tier architecture

Salle Dhali
Abstract

The reason for performing this project work is to develop a Web application for the Student Union of Mid Sweden University applying the modern and comprehensive Microsoft .NET framework platform architecture. At present, the existing web application is divided into several modules which are built of server-side scripting language technique and an open source database. The customer would like to develop the entire web applications using the Microsoft development tools and technologies in order to determine the possible benefit which could be obtained in terms of cost, maintenance, flexibility and the security perspective issues and also in terms of user friendly interactions options for all the involving partners in an effective way.

The primary aim for the project is to start building a bookstore module for the Students Union that is responsible for selling literature to the students at the University. The module will also be integrated into a database system into which an administrator, a member of staff working in the Student Union, will be able to add a new book when it arrives and also update or delete if necessary later on. In addition to this module application all the book’s details belong to a certain category viewable to the students. The other part of this project work is aiming at finding a pattern similar to the bookstore module in which ordinary users can authenticate them towards a database and be able to add their curriculum vitae data entry and update it at a later stage as required.

Keywords: Human-computer interaction, ASP.NET, .Net, C#, SQL, ADO.NET, N-tier distributed data architecture.
Acknowledgements

I wish to thank my examiner Dr. Ulf Jennehag who has helped me and guided me so as to complete this task. I appreciate the valuable time he has spent for this project. I would also like to thank my tutor Mehdi Raoufi who has spent a great deal of time corresponding me by means of e-mail and telephone. The valuable advice and the instructions delivered by them were very helpful.

I also wish to thank Microsoft for its MSDN (Microsoft Subscriber Network) library that has been very useful in relation to implementing this project. Microsoft has gathered all the technical facts about the development environment and development tools in the Microsoft Web site primarily on MSDN.
# Table of Contents

Abstract ............................................................................................................. ii  
Acknowledgements .......................................................................................... iii  
Table of Contents ............................................................................................. iv  
Terminology ....................................................................................................... vi  

1 Introduction ...................................................................................................... 1  
1.1 Background and problem motivation .......................................................... 2  
1.2 Overall aim .................................................................................................. 3  
1.3 Scope ........................................................................................................... 4  
1.4 Concrete and Verifiable goals ....................................................................... 4  
1.5 Outline .......................................................................................................... 5  

2 Tools .................................................................................................................. 6  
2.1 ASP.NET ....................................................................................................... 7  
2.2 .NET Platform Architecture ......................................................................... 8  
2.3 Language C# ................................................................................................. 10  
2.4 ADO.NET ....................................................................................................... 11  
2.5 Visual Studio.Net .......................................................................................... 12  
2.6 HTML ........................................................................................................... 13  
2.7 XML ............................................................................................................. 13  
2.8 Microsoft SQL Server .................................................................................... 14  

3 Model ................................................................................................................. 15  
3.1 Web forms .................................................................................................... 15  
3.2 User Controls / Menu .................................................................................... 16  
3.3 DataGrid ....................................................................................................... 17  
3.4 Application architecture ............................................................................... 18  
3.5 Database ....................................................................................................... 20  
3.5.1 Creating the data model ........................................................................... 20  
3.5.2 The database schema ............................................................................... 21  
3.5.3 Data Provider ............................................................................................ 22  
3.5.4 Stored Procedures .................................................................................... 26  
3.6 Coding pattern ............................................................................................... 28  
3.6.1 Object insertion method to Database ......................................................... 29  

4 Web Application Design .................................................................................... 32  
4.1 Home ............................................................................................................ 32
# Table of Contents

Salle Dhali

4.2 Admin page ................................................................. 34
4.3 Login page for users .................................................... 39

5 Results ............................................................................. 41

6 Discussion ......................................................................... 43
6.1 Future works.................................................................... 45

7 Conclusions ....................................................................... 47

References ............................................................................ 51

Appendix A: Bio-data access, registration and control validator .... 53
Appendix B: Educational data for CV ...................................... 55
Appendix C: Attribute values before changing .......................... 56
Appendix D: Results after changing records ............................ 57
Appendix E: User Control reuse benefits ................................. 58
Appendix F: Class diagram and code reuse techniques ................ 59
Appendix G: Insertion methods pattern .................................... 61
Appendix H: Database tables, keys and relations ...................... 62
Appendix I: DataGrid control for update purpose ..................... 63
## Terminology

<table>
<thead>
<tr>
<th>Abbreviation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>OS</td>
<td>Operating System. Interface between computer and computer users.</td>
</tr>
<tr>
<td>IIS</td>
<td>Internet Information Server/Services. A server application that Microsoft provides for Internet-based services.</td>
</tr>
<tr>
<td>SQL</td>
<td>Structured Query Language. Query-based languages for data management in a relational database.</td>
</tr>
<tr>
<td>CLR</td>
<td>Common Language Runtime. Execution environment for program code.</td>
</tr>
<tr>
<td>PHP</td>
<td>Hypertext Preprocessor. Script language for web development.</td>
</tr>
<tr>
<td>IL</td>
<td>Intermediate Language. Source code converter.</td>
</tr>
<tr>
<td>JIT</td>
<td>Just In Time. Act as a compiler.</td>
</tr>
<tr>
<td>CSS</td>
<td>Cascading Style Sheet. Document format for color, size, text etc.</td>
</tr>
<tr>
<td>CTS</td>
<td>Common Type System. Control of language during program execution.</td>
</tr>
<tr>
<td>TDS</td>
<td>Tabular Data System. Interface between a database server and client.</td>
</tr>
<tr>
<td>LINQ</td>
<td>Language Integrated Query. Expressions typed language.</td>
</tr>
<tr>
<td>ODBC</td>
<td>Open Database Connectivity. A standard for database access.</td>
</tr>
<tr>
<td>RAD</td>
<td>Rapid Application Development. Software development method.</td>
</tr>
<tr>
<td>URL</td>
<td>Uniform Resource Locator. Shortening a web site address for easier remembering and entering for the human being.</td>
</tr>
<tr>
<td>Term</td>
<td>Definition</td>
</tr>
<tr>
<td>--------</td>
<td>------------</td>
</tr>
<tr>
<td>WWW</td>
<td>World Wide Web or shortly call the Web is a system commonly linked with hypertext documents which allow us accessed to the Internet via a web browser like Internet Explorer.</td>
</tr>
<tr>
<td>DLL</td>
<td>Dynamic Link Library. Microsoft shared library implementation contains code and resource.</td>
</tr>
<tr>
<td>COM</td>
<td>Component Object Model. Binary interface standard for software development allowing dynamic object creation. The term COM can be referred in this report as OLE and ActiveX data object.</td>
</tr>
<tr>
<td>API</td>
<td>Application Programming Interface. Software communication with each other based on the source code.</td>
</tr>
<tr>
<td>jQuery</td>
<td>Java based script library has cross browser capacity and allowing integrating client side script of HTML.</td>
</tr>
<tr>
<td>T-SQL</td>
<td>Transact SQL. Communication between an application and an instance of SQL server sending policy via transact SQL statements.</td>
</tr>
<tr>
<td>IDE</td>
<td>Integrated Development Environment. An application provides tools and debugging opportunity for source code and also gives facilities to the developer building their application using an interactive design environment.</td>
</tr>
<tr>
<td>XML</td>
<td>Extensible Markup language defines a set of rules for encoding a page in such a format which is readable both for man and machine.</td>
</tr>
<tr>
<td>XSD</td>
<td>XML Schema Definition. Defines how formally an element can be described in an XML document. It also represents abstractly an objects character and relation to other objects.</td>
</tr>
</tbody>
</table>
1 Introduction

The main purpose of this thesis is to analyze the newly released server technology ASP.NET (Active server pages) and also to explore many of the different technologies behind the Microsoft .NET platform, not least, the programming language in C# (Pronounced C- sharp). The objective of this thesis has been to be able to apply these technologies to build a Web system that might replace the present application system of the Student Union. The active Web site is created with PHP (Hypertext Preprocessor) and a MYSQL database.

The aim of this work is not only to develop a Web application system, but to ensure its functionality; it will be easy to use, easy to navigate and understand. The aim is also to determine the possibilities to construct some components that would integrate with other modules. The combinations of all the pieces together will thus result in a wide perspective complex system. The different module, with their different functional features will form part of the Union’s entire Web system which will have the ability to the development of an expandable dynamic portal system.

Web application files for the project are stored on a Microsoft Internet Information Server (IIS) that is available on all Windows 2000 server operating systems but will have to be installed separately on the Windows XP Professional edition. Data is stored and retrieved in a Microsoft SQL Server 2000 database. The programming language chosen is C#. The reason behind the choice of Microsoft’s tools and technologies is that the Student Union would like to examine how their present web application able to utilize the benefits relating to the migration to the .NET technology. The cost, maintenance and the effectiveness also will also be evaluated after the new system has been rationalized and integrated.
This report also presents also a brief discussion of a 3-tier data architecture which is preferably suitable in relation to the creation of the new web site. This architecture basically consists of a presentation layer/segment that presents user friendly forms and pages to both the users and the administrator. The business logic layer is responsible for all the data manipulation and works as an active link between the presentation and the third layer which is the data access layer. A component based data objects model is used for accessing data and a SQL (Structured Query Language) server is used for the data storage, retrieval and data manipulation.

1.1 Background and problem motivation
The Student Union has planned to add many new features to their upcoming Web site. Their intention is to include many functional modules such as the news group, discussions forum, voting system, calendar events and also a books management system for an administrator. PHP and MYSQI can perform this task in a sophisticated manner but their reason for migrating to Microsoft technologies is based on the fact that the new ASP.NET is very robust and flexible which allow creating scalable Web sites. In addition they wish to analyze any additional benefits that are offered after this transformation that may be obtained from this new technology. This is a great concept for a small ideal based company in order to take further decision. However, it does require that they both adapt and adopt the new interfaces. The learning process and the capabilities available from the new system resources must be learnt and used for a given period in order to ensure that they satisfy the requirements. This trial and error process can gradually assist the Union staff to make gradual improvements to the functions to meet their future needs.

PHP and the most popular open source database, MySQL are dominant in the market to create dynamic web sites especially for small privately based companies.
The reasons to move from a scripting language associated with HTML and transform it to another platform can lead to minimizing the cost, involve reduced effort to maintain and to minimize the complications associated with building the Web sites. The motivation behind conducting this work is to determine the possibilities for the Union staff to have the ability insert objects in a simple way and to allow them to update and delete a books record. The aim is to give the Unions staff the opportunity to take care and handle some of the functional portions in order to reduce the cost. The hiring of a Webmaster to perform these tasks is beyond their budget means.

1.2 Overall aim

After the Student Union in Sundsvall had made the decision to introduce HCI (Human Computer Interaction) interaction between man and machine they came to the conclusion that the site should be easy to use and that the construction of the site should be user friendly.

The overall goal of this work was to utilize the advantages offered by the Microsoft .NET framework together with its rich contents of Class libraries for building the Web application. The aim was also to explore the possibility of using program codes- and scripts following the same pattern that will be applicable to different parts of the project. To be able to re-use the code and to rewrite it from the Object Orienteering programming concept allowing for the encapsulation of the data was also an aim for this work. The aim is also to design a logical distributed data architecture system in which all involved partners are able to contribute their work and the result is thus the ability to integrate a large complex system.
1.3 Scope

Part of the practical work for developing this web application covers only the functional elements of an administrator who can perform this job in a simple way. The work does not include any graphical design or layout. Resume data entry elements also have their limitations. No account has been taken in relation to any safety measures for the sides since some of the constructed modules did not possess a start date or even if it was possible for them to operate. The data type and value parameter field in relation to different insertion methods such as books record and user details also have the limitations and are not sufficiently optimized. The data fields for those parameters can be added or minimized in accordance with the Union demands. This report does not offer any elaborate comparisons in relation to the technological advantage or disadvantage of both PHP and ASP.NET as well as Microsoft SQL server with MySQL server databases.

1.4 Concrete and Verifiable goals

The concrete goals can be summarized by the following points:

- What types of data fields will be included in the database and how will the database be formed?
- Investigate to find any pattern possibilities for inserting different blocks of data in a certain model for the SQL statements.
- Is there any possibility to reuse and rewrite the same code pattern during the whole programming phase?
- What type of data structure is it preferable to use and will it be beneficial and how effective will it be?
- Exploring the new benefit features providing the new Visual Studio if there are any and does the new programming language C# make the programmer’s life easier or not.
- Investigate the positive and the negative sides on behalf of using the tools and technologies provided by the Microsoft Corporation.
• Check the opportunity to create custom ASP.NET control that helps building the application and will thus improve the overall progress during the application’s lifecycle.

1.5 Outline
Chapter 2 describes the tools required to develop the application. In addition, the .NET compatible tools there are also some other universal tools and techniques and their functions are also discussed here. Chapter 3 describes the methods used to implement and to develop the application. The database model and the programming methodology that integrates the overall system are also describes here. Chapter 4 describes the design pattern and the construction of the different sides for the users and administrators. Chapter 5 describes the results that were produced by using the construction part in chapter 4. Chapter 6 involves the discussion, recommendations, the positive and the negative parts faced during the application building process time and a future working plan that could develop further. Chapter 7 describes the conclusions with additional opinions. The benefit using the .NET platform, different developments tools and the programming techniques from the results are discussed. Appendix A and B illustrates how the data entry for curriculum vitae can be formed. Appendix C and D shows the results after updating some attributes values. Appendix F shows the two commonly used class methods and properties. Appendix G shows the SQL statement especially the Insert method. Appendix I shows the table and the key relations. Appendix I shows the DataGrid bound columns and techniques for update records.
2 Tools

In order to develop the Web application using the technique of ASP.NET with the development environment Microsoft Visual Studio.NET the first version from 2003, also known as Visual Studio 7.0 was used. In this environment, a developer can access the ASP.NET technology for building a Web application. It also permits the choice from two of the popular programming languages like C# and VB and also another twenty languages. Visual Studio also allows including the class files that are available in Microsoft .NET framework for building up an application.

In Visual Studio there is a great feature that allows us to work with different controls named Toolbox tabs. Under these tabs there are many icons that can be added in a Web page by simply dragging those controls to that page or pasting them a code editor by using Visual Studios integrated environment. Both these process will create an instance of a Toolbox item to the project file. There is also access to the Server Explorer or Database Explorer (server management console) for Visual Studio. With the help of this desktop window it is possible to open the database connection, log on to a server that allows a user to work with the tables, columns, rows, Stored Procedures, retrieve and manipulate data from that table etc.

The tools and techniques used are a combination of different types of programming languages especially the new .NET supported language C# to handle the data logic, a set of different techniques including the server side technologies, component based object for data access and data manipulation. Integrated developments tools such as VS and SQL Server Management Studio are also used to develop the Web application.

Sections 2.2 to 2.8 describe the most commonly used tools that were used to create the project and also provide a brief description as to how they work.
2.1 **ASP.NET**

ASP.NET is used to create dynamic web pages and has been developed by Microsoft. ASP.NET is a set of underlying technologies based on .NET Framework. The main advantage of ASP.NET over other technologies in the same genre is that ASP.NET is component and event driven based, is able to offer on a fully fledged programming language instead of the simple scripting language that some competing technologies use [1].

ASP.NET Web pages and other .NET applications are created by using modern and easy object-oriented programming techniques. Each ASP.NET page can contain a number of objects and Web controls. These controls have methods, attributes and events. Each ASP.NET page runs on the web server. After a request for a page is received by a Web client, the server executes the code and sends it to the browser. The page is then sent to the client as an HTML, X-HTML, or XML encoded version.

Figure 2.1 illustrates the method as to how a Web client communicates with a ASP.NET application through the IIS (Internet Information) Web server in .NET environment.

![ASP.NET Architecture](image)

**Figure 2.1:** ASP.NET Architecture [1].

When a browser requests an ASP.NET file, the IIS passes the request to ASP.NET engine on the server, the ASP.NET engine reads the file line
by line, and performs the scripts in the file. Finally, the ASP.NET files are sent back to the browser as plain HTML. When an ASP page is executed, it is firstly compiles and then saved on the server. The next time the page is executed it does not to be recompiled again. The page will appear directly to the client. The advantage of this technique is to access the requested page quickly and thus reduce the server's activity.

There are also very simple and good help functions such as the Control Validator that enables there to be control over what the user types inside a textbox. An attempt to input an incorrect password or an invalid mathematical numbers can generate an error message to the user for correction. A major advantage of ASP.NET files and its connections to the database allows the use of the ASP.NET powerful control features such as the DataGrid [13] or a Data List. These act as a data table where selected information from a single database is shown in a very flexible way and allows the table record to be updated or deleted if necessary.

In ASP.NET it is possible to separate the programming code (C#, VB.NET) and other markup languages such as HTML, XML, and CSS content for a Web page using the "Code Behind" technique. It assists the developer in separating or mixing with different types of language in the object file and provides a significant opportunity to obtain a better understanding and overview of the work [3].

2.2 .NET Platform Architecture

.NET is the core of Microsoft's future operating systems for both private and business owners and enables the creation of computer applications which are very powerful Web based applications. .NET Framework has a rich library of over 4000 classes organized into namespaces that provide a variety of useful functions for everything from file input and output to string manipulation, XML parsing and for Windows Forms control. Each namespace contains types that can be used in the program such as classes, structures, references and inheritance etc. They provide very useful functional building blocks necessary for the development of .NET applications.

1 Displays list of items that are bound to the control
All the namespaces from Microsoft start with *System* or *Microsoft*. An example of a namespace is *System.Data* consisting of the classes that represent the architecture of ADO.NET (ActiveX Data Object for .NET). .NET applications can be console applications, Windows applications with graphical interfaces and Client / Server based Web applications. The Web application can be enriched with the .NET supporting controls of that Web forms for a used friendly presentation that supports most browsers. .NET applications are uses for Windows, Nokia’s and some HTC mobile devices. .Net platform is becoming popular for Server based OS and the corporate Active Directories (AD) by using .NET Directory interfaces and Lightweight Directory Access Protocol (LDAP).

Figure 2.2 illustrates the compile-time and run-time relationship between C# source code files, the .NET Framework class libraries, assemblies and CLR (Common Language Runtime).

![Net architecture diagram](image-url)

*Figure 2.2: .Net architecture [4].*
The most important part of the construction behind .NET framework architecture is its language "Interoperability" options which mean that different languages can be integrated into a system such as TCP / IP, HTTP and other file formats. The IL (Intermediate Language) code produced by C# compiler conforms to the common language type of specification (CTS) interfaces, and the code generated by C# can interact with code that was generated from .NET versions of Visual Basic, Visual C++, or any of more than 20 other CTS-compliant languages. A single application can contain multiple modules that can be written in different .NET languages including C#, VB.Net [4]. Those modules can still refer to each other and communicate as a single unit even though they are separate languages.

2.3 Language C#

C–sharp has been developed by Microsoft engineers. It is a modern, simple, safe type and has been created based on a completely object orienteering programming concept and paradigm. A developer who is familiar with the popular programming languages C / C++ can easily recognize the language. C# uses both the power of C++ and the rapid application development language (RAD) [5].

C# was created from scratch and has been fully developed for the .NET platform. Microsoft began with what worked best in the programming language C / C++ and then added new features that would make C# easier to use. Additional functions that have been added to this language are the exception handling, garbage collection and the code safety features.

The creation of this language has picked the best and left some of the difficulties part of learning a programming language, such as the removal of the macros, templates and multiple inheritance. As is the case for other programming languages, the C# can be used to create a wide variety of applications such as simple traditional Windows client applications, XML based Web services, distributed components, client-server applications, database applications and creating dynamic Web sites.
C# is very similar to the Java programming language developed by SUN, which is now owned by Oracle. Both are strongly typed and class based object oriented language. Both use the semi interpretation or the runtime compilation methods.

C# supports generic methods and types, which increases type safety and performance and the iterators. Via an iterator there can be a rapid access to an element in a container. An iterator acts like a pointer which points elements in the container in a certain way. Language Integrated query (LINQ) expressions are strongly typed queries thus making C# a first-class language construction [6].

### 2.4 ADO.NET

ADO.NET (ActiveX Data Objects for .NET) is a rich set of objects to access data and services. It is part of the base class library included in Microsoft .NET Framework. It is often used by programmers to access and to modify the data stored in a relational database. However it can also access data in non-relational data sources. ADO.NET is sometimes considered as being an evolution of ActiveX Data Objects (ADO) technology [6].

ADO.NET is integrated in .NET Framework and is designed to be used with .NET supported languages such as C# or Visual Basic.NET. System.Data.SqlClient and System.Data.OleDb are two types of namespaces which represent the Microsoft SQL Server and another type of database respectively. For example Oracle or MySQL databases can be connected by using OLEDB data objects.

ADO.NET object model is very rich with their classes, but it is in particular the Dataset that represents the important part of this model. The advantage of working with Dataset is it can store a certain amount of relational data to the local computer as a catch memory and this assists in the manipulation of data without connecting to the database [12].
2.5 Visual Studio.Net

Web application is based and constructed on the development tool Microsoft Visual Studio .NET which is the first version appeared in 2003. Visual Studio is an integrated development environment for building Windows applications, client / server applications, and Web-based applications for creating dynamic web sites, game developments and mobile applications.

Visual Studio .Net IDE (Integrated Development Environment) is also used to create applications using the scripting languages such as JavaScript or VBScript (Visual basic). The advantage is that it is possible to construct a large program that can be integrated with both the script and the programming languages such as C-sharp and Visual Basic.NET. The wide varieties of tools options that are available in VS enable the allocation of different resources such as the file systems management and the database manipulation. The development tools also provide the opportunity to create scalable solutions which are able to integrate with a wide variety of languages.

The latest version of VS 2010 (Visual Studio) has a number of improvements. First of all, it allows for the writing of code against the multiple versions of previously released .NET frameworks and CLR (Common Language Runtime). The interpretation from this is that those who have worked with .Net 2.0 or 3.0 should be able to integrate the system with all the VS versions, i.e. 2005, 2008 etc.

Another handy feature is Intellesence or track which means that it is possible to rewind the program and debug which is of significant assistance in determining the errors in the program lines even though the program has not been run. It also has improved support for jQuery and other client-based script system. F# (pronounced F-sharp) is also built into the new Visual Studio environment, which is a functional programming language.
2.6 HTML

HTML stands for "Hypertext Markup Language" is a markup language that is how a page should be described. It is not a programming language. In fact the language works as varieties of tags that reside inside the text files. These tags correspond to how different parts of the text behave to each other and how the text is displayed to the web client via a Web browser.

2.7 XML

XML stands for "Extensible Markup Language" and is a standard as is HTML developed by the W3C (World Wide Web Consortium). XML is a markup language that defines a set of rules for constructing a document into a format that is both simple, easy and machine readable. The designing goals for XML are on its emphasis on simplicity, generality and usability on the Internet. There is a textual data format with strong support through Unicode languages throughout the world. Although the design of XML focuses on the document, it is widely used to represent arbitrary data structures, such as Web services [8].

Many application programming interfaces (APIs) are available for application developers so that they are able to benefit from both using and processing XML based data. The multiple schedules in a system application in terms of XSD schemes availability offer the extra facility to define the XML-based languages in an easier way.

XML is a platform independent open standard data format and in addition it can use the Web protocol "HTTP" (Hypertext Transport Protocol) to communicate over the Internet. It can be part of viewing data and manipulating the client side data and as well as being used for exchanging data between servers [10].
2.8 Microsoft SQL Server

SQL Server 2000 is Microsoft’s best product among DBMS (Database Management System). Its performance based on its costs has made the SQL Server the most widely used database system among the IT companies that are using the Windows NT/2000 server family. SQL Server is used frequently in the public sector, including managing payroll, absence management, rating etc.

SQL Server is a fully-fledged relational database system for the administration, management, reporting and analyzing of databases. It has a variety of tools to assist developers to connect to a database using an interface, add different jobs, handing out permissions, import / export data, writing a query set (SQL Query) to create the table, test the application or to modify data [9].

The SQL Server is perfectly suited for the Web. For example, XML can be used for data exchange for loosely coupled data systems. It allows the accessing of data using via a Web browser in an easy and secure manner.

SQL Server 2000 provides support for reading, writing and using XML documents. The ADO.NET version 2.6 and its later versions have full support for the use of XML in SQL server-based applications [10].
3 Model

This section depicts how the web application is systematically created by performing different types of procedures. As applications will act as both client and server based roles, building some Web pages that will be shown to the clients is the first consideration. The client is possibly an administrator or an ordinary user.

By dividing the application workflow in different methods this will prove of assistance in arriving at the correct method for performing a particular task. In a Visual Studio environment, creating a new project, choosing language C# and pick ASP.NET Web application results in a new working area. Many techniques are involved for working both methodically and pedagogically in relation to creating the project. The commonly used ASP.NET techniques and model such as the system architecture model, menu model and the database model are discussed in the following section.

3.1 Web forms

ASP.NET Web Forms are part of the ASP.NET technology used by the .NET framework to create programmable Web applications. Web forms consist of pages that ordinary users request (HTTP Request) by clicking their Web browsers, such as IE (Internet Explorer). It then creates a user interface that provides Web applications with their visibility and character.

Web forms are written by a combination of HTML and server control code. When a user requests a page that is valid, it has to be compiled and then executed on the server. The server then generates the HTML code which enables the Web browsers to show the pages for the clients for reviewing. All browsers work with the HTML language and it is platform independent.
With Visual Studio.NET, the developer can create ASP.NET Web Forms with a powerful IDE (Integrated Development Environment). For example, it provides the ability to drag and drop a number of server controls for adding to the Web Form pages or simply calls the .aspx pages. It is easy to set properties, methods and events for the controls of those pages to define the page's behavior, look and character. To write the server code to handle the logic for the page it is possible to use .NET languages such as Visual Basic or C#.

The form's visual part is saved as a file with an .aspx extension or suffix which makes it different to the previous one .Asp file. It consists of two parts, a design view and an HTML view. The design view is the area where the developer can drag and drop various Web controls. Toolbox tabs is available in the Studio environment to obtain those controls. Some of the common Web controls are text boxes, buttons, dropdown menus, list view etc. It is quite easy to select a control object and change properties by using the Windows Properties. Properties can involve appearance, color, visibility, text or size. A method, namely Page Load () initialized to handle the Web page objects event handling procedures. The code that has to be executed each time has been placed in this method.

3.2 User Controls / Menu
An ASP.NET user control works in a similar manner to ASP.NET web forms. It has full capabilities for declaring methods and properties for the web server controls and also allows for the binding of markup languages for the controls. The control can be used to integrate any ASP.NET pages by adding its value to those pages where they will act with each other as a single unit [11].

The difference between the old traditional ASP and ASP.NET are that the old version is written in VBScript and it had very basic level output functionalities. The new ASP.NET can be written in any programming language forming part of the .NET framework, and thus has access to many richer controls and class libraries. ASP.NET User Control helps to
create a rich application that has much more powerful advanced functionalities.

In traditional ASP files are included in order to reuse code. They can contain scripts to perform the HTML code. An example is including an HTML menu that appears on every page and a script block to identify and authorize users. The disadvantage occurs when someone requests an ASP page that would be shown to a client which includes included all the related files as a single continuous side and the package is running in the same context. It means that the application code cannot determine whether a file is to be included or not. Methods and properties are missing in the inclusion files.

In ASP.NET, there is the ability to create a UserControl that is very suitable for developing any customary controller and allows for the reuse code technique for different pages without having to rewrite it. It consists of the simplest form of HTML code that is stored in a separate file. The file is included dynamically when a page is requested. A User Control file is saved with the extension .Ascx, and recorded on each HTML page that uses it. Registration can be conducted as follows:

```csharp
<%@ Register TagPrefix = "control" tagname = "test" src = "user_control.ascx"%>
```

TagPrefix is the prefix used when the UserControl is added to a page, tagnname is an identification name and src is the virtual path for that particular .Ascx user control file.

```csharp
<%@ Register TagPrefix = "control" tagname = "test" src = "user_control.ascx"%>
```

3.3 **DataGrid**

In addition to these two techniques namely Web Forms and User Control, the Web application uses a third and very important control that is the DataGrid control. In Visual Studio from the Toolbox tabs there is a drag and drop option for attaching GridView control to an ASP.NET Web form by simply using a cursor. The GridView can then be connected to a
database located inside the local machine or any remote host by using Server explorer tools. Binding the DataGrid to a specific database enables the database tables to be viewed and tables data to be shown.

DataGrid control also provides also a user interface to the ADO.NET datasets which displays table data in a scrollable list and enables the necessary updates to the data source. In cases where the DataGrid is bound to a data source with a single table containing no relationships, the rows and columns are in a manner similar to that for submitted like a spreadsheet sheet.

The DataGrid control is one of the most useful and flexible controls in the Asp.Net Web Form. As soon as the DataGrid is connected to a valid data source, the controls will automatically create columns and rows based on the structure of the data to the data source. The DataGrid control can be used to display either a single table or hierarchical data [14].

3.4 Application architecture

A conventional distributed Web application usually divides into three separate logical layers, the so called n-tier data architecture. The purpose of choosing this architecture is to separate those layers, in particular the business layer where the classes and the business rules will be corporate with the other two linkage partners. The programming part is following the method of shared code and reusing the code blocks for all the classes that belong to the business layer. Though the code has similarities but it differs in terms of the functionality. The different classes, members and methods in the business logic layer will act separately for both the users and the administrator respectively. The reason for choosing this architecture is relation to an administrator who will add different objects to a database, thus making it obvious for there to be a database server, which in this case is a MS SQL Server 2000 chosen for data storing and data manipulation.
Both the administrators and ordinary users will have access to Web pages for log in functions and data entries for book’s arrival moments will be allowed and as well as update requirements, thus involving the presentation layer at this point. Also, in the middle tier, the classes making the business rules will serve as a bridge between the two other layers. These rules are completely removed from the client. Figure 3.1 illustrates how an n-tier data architecture model is structured and the different parts that are linked to each other for communicating and data manipulation.

![Diagram of a 3-tier architecture](image)

**Figure 3.1:** A 3-layer Client/Server architecture model [10].

The three layers and their properties are briefly discussed in the following:

**DAL**-stands for "Data Access Layer" Accessing data from any data source and the connection to a data source as in this case a SQL Server relational database. The idea behind the whole logical data model is to separate all the corresponding segments and the working process related to those parts. It means that if the data source is changed, altered
or replaced in the data access layer then the two other layers will not be affected.

Database communication in this access layer means two things in particular:

- Retrieve data from the database.
- Save, delete, or update data to the database.

**BLL**-stands for "Business Logic Layer" - this layer is responsible for all the logic and it is in this layer where all the instructions, calculations and the manipulation of data occurs. The data that has to be manipulated and initially retrieved from the data access layer. The business layer has a connection to both the Data Access layer and the Presentation layer and thus acts as a linking bridge between these two layers.

**Web / UI** (User Interface) - is the "Presentation Layer" which is the top layer in the architecture which also support graphical interfaces. This layer is built with Web components such as ASP.NET Web forms, different types of controls and a number of validators. Those Web components are essential to build the user friendly interface. The data to be presented are available through the Business layer.

### 3.5 Database

This section will describe the database for the web application and will firstly provide some information about how the database was designed and how that data can be retrieved from the database. Following this there will be a short discussion about the implementation, including a detailed explanation of the database schema and Stored Procedures.

#### 3.5.1 Creating the data model

The data model used to create the Web application relays in this top to the bottom level model. This means that at the initiation of designing this application, identification of the various tables and data fields is the first requirement. This will eventually assist in paving the way for
building the remainder of the application. As an example, three tables have been identified for this Web application and the different types of field values including integer, string etc. The rows, columns and the keys for holding the table values are briefly discussed in the next section and, in addition, form the construction the data model. These tables are directly coupled to 2 corresponding classes.

The data model specifies how data is to be stored and accessed in a database system. It acts as a “Wayfinding” tool between developer and companies. This abstract model using text and symbols to communicate within an organization thus leads to a flexible and stable application environment [16].

3.5.2 The database schema

In order to design the database schema, it is essential to look at the subsequent requirements contained in the Web application. In this application, the database will require some tables to hold the data values; the first table is named BooksCategory where the administrator can select the category in which the particular book will fall.

BookDetails table is the second table that shows all the books in the database. This table will also serve as a container where admin staff can add new books, update or delete record etc. All the book’s related data will be stored in this contained table.

The third table is intended to be used for user registration purposes. Once they have logged in with proper, eligible codes then they will be redirected to a new ASP.NET page where they can add their CV (Curriculum vitae) data etc.

To be able to identify the tables and Stored Procedures belonging to the database is possible by simply identifying many of the verbs (the methods and properties of class) and nouns (the class itself) for certain user groups such as:

- The first Administrator can add a new book.
The second Administrator has a review of the book.

The third Administrator can remove a book or update.

The fourth Students can register and add their profiles etc.

Below, follows a brief discussion as to how the various database concepts work in the Web application.

### 3.5.3 Data Provider

In .NET framework, the data provider is coupled as a bridge between an application and the data source. The data provider encapsulates all the connections to a database. It contains several objects that represent the key elements of the .NET Framework. The two commonly used providers include the OLEDB (Object Linking and Embedding Database) provider which used to connect a wide variety of databases such as Oracle, MS Access and MySQL.

On the contrary, communication between SQLDB classes with SQL Server occurs via Tabular Data Stream (TDS). TDS is a low level protocol used by SQL Server to manage client and server communication. To use a Microsoft SQL Server imports the following namespaces: `System.Data.SqlClient`

ADO.NET data object is used not only to connect a data source but also to retrieve data from the database in order to work with the manipulation of data. Examples of some of the central objects commonly used in the Web application are the `Connection`, `Command`, `DataSet` and `DataReader`. Figure 3.2 illustrates the central components of ADO.NET data object model.
.NET data provider is one of the core elements behind the ADO.NET data architecture in .NET Framework. The Data Provider assists in not only manipulating the data but also accessing the data in fast, forward only and read only ways and thus resulting in a high performance stream of data access. The following set of components that consist of both the .NET data provider and the DataSet, are briefly discussed in the following, which are the core component behind the ADO.NET architecture.

**SqlConnection** is the part of the data provider used to connect to a specific data source. An example of this is to create an instance of Connection with the name myConnection having different parameters.

```csharp
SqlConnection myConnection = new SqlConnection("server=localhost;database=name;uid=sa,pwd=;Integrated Security=True");
```

The disadvantage associated with this is if someone later changes the password, switches to a different environment, uses another directory to restore databases or the security policy sets to false option, this leads to the whole source code files system having to be recompiled.
The proper solution to this problem is to place the "Connection" string to an AppSettings [Chapter 6] section in the Web application file called Web.config (Web Configuration). It is an XML file that is included during all new ASP.NET applications creation time in a Visual Studio integrated environment. An example is given below:

```xml
<appSettings>
    <Add key = "Connection String" value = "server = data source = localhost; uid = sa; pwd = ; Security=True ">
//Any changes that will be made can be written above in between the Meta tags.
</appSettings>
```

**SqlCommand**

A command contains information that is sent to the server as a query or a procedure. It can be any SQL query that is valid against the data source which it should be linked to. SQL statements always begin with a command, "a word or group of words, describing the action statement will initiate" [15]. A typical command is CREATE DATABASE that describes how to create a database.

SqlCommand can be explicitly set to change a SQL query during the program. It contains the command as a Stored Procedure or SQL command statement and the Connection in which it will be used to perform the execution of that command.

**SqlDataReader**

A DataReader object provides methods and properties that provide a read-only stream of data rows from the data source. It is used to read the results of a database query. Since it is a forward-only iterator then it can access the data source very quickly. This project used DataReader quite frequently because the majority of the data would be read.

The limitation associated with the DataReader object is that it cannot work with the results of a query as a whole. For example, it is not
possible to sort, or scroll to other pages and is the reason why a DataSet is commonly used.

**DataSet**

A DataSet represents a collection of cached data that contains tables, columns and rows. A DataSet object is stored in the memory and works almost as a separate database. It has no connection to the database but uses a DataAdapter which helps to fill the DataSet. It then provides the possibilities to update records towards the data source.

The DataSet contains DataTable and Data Relation classes. The DataTable class represents a database table that is stored in memory. It can be created manually or filled from an existing database. Figure 3.3 illustrates how it works with DataTable in a DataSet object model which enables the creation of relationships between them by using the class DataRelations. In order to filter, sort and search, a DataTable uses the class DataView.

![Diagram of the DataSet object model](image)

**Figure 3.3:** Illustrate a DataSet object model [13].

A DataSet has the important property that it keeps track of the changes with the new value for a specific data that the DataSet contains in its
memory. This means that it can easily update the data source with the new value to be held by the database.

3.5.4 Stored Procedures

SQL "Stored Procedures" are useful for several reasons. Firstly, it assists in avoiding writing SQL statements on the Web Form, thus separating SQL statements from the source file. It assists in calling a query from the front-end. Stored Procedures can be created both in the Enterprise Manager and Query Analyzer or it can be written via T-SQL script.

To be able to write an SQL statement in a separate text file and mark it as a Stored Procedure and thus leads to transparency. It increases readability and in addition the faster response time from the SQL server. The performance also becomes significantly better as a Stored Procedure is a pre compiled line SQL query. Once compiled in a SQL server, this allows residing in the server as a cache and will tremendously reduce the subsequent calling time by speeding up the system. Stored Procedure allows secure data access and assists in the ability not to change the whole application rather than modifying only the procedures.

In Enterprise Manager, the Stored Procedure stores both the user-defined procedures and the system procedures that belong to the database. To create a Stored Procedure in the Enterprise Manager, merely mark the Stored Procedures, then right-click and select New Stored Procedures.

An example of a Stored Procedure named BooksAdd is illustrated in Figure 3.4. This procedure is responsible for inserting books data records.
Web application development with .NET- 3-tier architecture

Salle Dhali

```csharp
/* CREATE A PROCEDURE IN THE DATABASE AND ADDING THE PARAMETER LIST*/
CREATE PROCEDURE booksAdd
(    @BooksName nvarchar(50),
    @ISBNNumber nvarchar(20),
    @AuthorName nvarchar(50),
    @PublishingDate nvarchar(20),
    @Price decimal,
    @BooksImages nvarchar(50),
    @CategoryID int,@BooksID int OUTPUT)
AS
/* THE FIELD VALUES WITH THE SPECIFIC DATA TYPES INSERTION TO THE TABLE*/
INSERT INTO BooksDetails
(    @BooksName,ISBNNumber,AuthorName,PublishingDate,Price,BooksImages,CategoryID)
VALUES
(    @BooksName,
    @ISBNNumber,
    @AuthorName,
    @PublishingDate,
    @Price,
    @BooksImages,
    @CategoryID)

/*RETURN THE IDENTITY VALUE OF THE PRIMARY KEY VALUE*/
SELECT @BooksID = @Identity
RETURN
```

**Figure 3.4:** Inserting a record Stored Procedure scripting method.

The variable declarations with proper data types and the length of the data field are first and the data holding values to that table data. Insert Into represents the database table into which the data is to be inserted. The values are the attributes data fields for that particular data table Booksdetails. The BooksId is the primary key that will help to hold all the values belonging to that table. Identity is the keyword that increments the record values by adding a 1 numerically for a record, for instance a table records (1+2+3+…+n) as the sequential unlimited numbers. It could be performed by programmatically or in the SQL Server Management Studio. This keyword not only keeps track of the numbers but is also used as a primary key value for holding the table data value records related to the number.
A good example of a Stored Procedure T-SQL (Transact SQL) named the BooksAdd can be highlighted according to Figure 3.4. The procedure will handle the books insertion method from an administrator. To add the book's record information and store it in the database is approximately the same as in real life as to entering any sort of similar data records. The purpose of this work is also to enable a student user to enter the student records and be able to insert their resume data entries. Similar techniques can be used in order to write the same type of procedure in order to add bio-data and educational data.

### 3.6 Coding pattern

Without providing a great deal of coding details, here are a few lines about how this project worked with the coding system. The following is a waterfall method algorithm that can be described as follows:

- The first Create a Connection to the database.
- The second Creating a Command object.
- The third Select Command type as a Stored Procedure.
- The fourth Create and delegate parameters.
- The fifth Execute the Command.
- The sixth Closes the Connection

A good example of `GetBooks ()` method that brings all the books belonging to that category from the `BooksCategory` database table:

```csharp
public SqlDataReader GetBooks (int CategoryID)
{
    // Create Instance of Connection and Command Object
    SqlConnection myConnection = new SqlConnection (ConFigureationSettings.AppSettings ["Connection String"]); [chapter 6]

    SqlCommand myCommand = new SqlCommand ("BooksByCategory", myConnection);
    // Rest of the code...
}
```
/ Mark the Command as a stored procedure
myCommand.CommandType = CommandType.StoredProcedure;

// Add parameter to stored procedure
SqlParameter parameterCategoryID = new SqlParameter("@ Category-ID" SqlDbType.Int, 4);
parameterCategoryID.Value = CategoryID;
myCommand.Parameters.Add (parameterCategoryID);

// execute Commands and closes Connection
myConnection.Open ();

SqlDataReader result = myCommand.ExecuteReader (CommandBehavior.CloseConnection);

/* as a command should return a value, they will not be available until
the DataReader is closed. */

// Return the results from the data reader
return result;
}

### 3.6.1 Object insertion method to Database

The project work is generally adding objects to the database. The following code demonstrate how to add users in the database. In this AddUser method it uses three parameters with the same String data type.

```csharp
public String AddUser(string sName, string sEmail, string password)
{
    // Create instance of Connection and the Command Objekt
```
SqlConnection myConnection = new SqlConnection(ConfigurationManager.AppSettings["ConnectionString"]);
SqlCommand myCommand = new SqlCommand("UserAdd", myConnection);

// Mark the Command as a Stored Procedure
myCommand.CommandType = CommandType.StoredProcedure;

// add parameters name, email, password to the Procedure
SqlParameter parameterStudentName = new SqlParameter("@UserName", SqlDbType.NVarChar, 20);
parameterStudentName.Value = sName;
myCommand.Parameters.Add(parameterStudentName);

SqlParameter parameterEmail = new SqlParameter("@UserEmailAddress", SqlDbType.NVarChar, 20);
parameterEmail.Value = sEmail;
myCommand.Parameters.Add(parameterEmail);

SqlParameter parameterPassword = new SqlParameter("@UserPassword", SqlDbType.NVarChar, 20);
parameterPassword.Value = password;
myCommand.Parameters.Add(parameterPassword);

SqlParameter parameterStudentID = new SqlParameter("@UserID", SqlDbType.Int, 4);
parameterStudentID.Direction = ParameterDirection.Output;
myCommand.Parameters.Add(parameterStudentID);

// exception handling
try
Web application development with .NET- 3-tier architecture

Salle Dhali

{
    myConnection.Open(); // open the connection
    myCommand.ExecuteNonQuery(); // returns the number of rows that // affected
    myConnection.Close(); // close afterwards

    // Count StudentID with the help of Output Parameter from the Stored procedure
    int StudentId = (int)parameterStudentID.Value;
    return StudentId.ToString();
}
catch
{
    return String.Empty;
}
4 Web Application Design

In sections 4.1 - 4.3, there is a brief description regarding how the Web application takes on the roles for different users and how it works in a real environment.

4.1 Home

When the Web application is called by the URL (Universal Recourse Locator) address, the start page appears as shown in Figure 4.1. The start page is divided into three segments; the right part is also called the parent part or the mainframe of the windows. The left side is a frame containing a menu list. The menu list items are the category list that will work as a link to the new pages. By clicking a link this will show different books details that belong to those particular categories such as the book name, author name, publishing date and price.

Figure 4.1: Displays the start page
This menu uses an important feature of ASP.NET custom UserControl which helps to show the menu categories by binding an ASP.NET server side DataList control. Categories data are retrieved from the category database table. If there is a click over a menu item, it then fetches all the books belonging to that specific category. Figure 4.2 illustrates how the different books belong to the IT/data books category are presented for the user after clicking that link.

![Books from the category data table](image)

**Figure 4.2:** Showing books from the category data table [17].

It is also possible to click on a certain book image to view a larger scale image and also to obtain detailed information about that book such as the price, the name of the author and the title [chapter 6] as shown in Figure 4.3.
4.2 Admin page

The two link buttons at the top right corner of the start page according to Figure 4.1 are actually for an admin page and a log in page for users such as students or ordinary users.

The idea behind this page construction is that when an administrator clicks on that button he/she/they will be redirected to another Web page which requires authentication. By typing the valid user name and password the administrator has been previously provided with they can be guided to the other page.
The code data has no database connection which means that it is hard coded in the source file named adminLogin.aspx. The following Figure 4.4 shows its appearance.

![Image of adminLogin form]

**Figure 4.4:** Access control Web form for the administrator

If the correct information is not entered then an error message is given as shown in Figure 4.5.

![Image of error message form]

**Figure 4.5:** Showing message for prompting the writing of a correct username or password.
This program control code is written in the source file. But ASP.NET has great validation control for fields with both regular and required expressions. See the last illustration in [Appendix A].

After they have provided the correct username and password to enable them to move to another Web form in which it is possible to enter multiple data entries for the registration of a new book. Figure 4.5 shows the appearance when “Mehdi” as an administrator user name has logged in with the correct password data.

![Image](http://localhost/ExamensArbete/adminInsertBooks.aspx)

**Adminstration site (insert books record)**

<table>
<thead>
<tr>
<th>Book's Name</th>
<th>Computer Network</th>
</tr>
</thead>
<tbody>
<tr>
<td>ISBN Number</td>
<td>1426-14-25</td>
</tr>
<tr>
<td>Author Name</td>
<td>Lars Nilsson</td>
</tr>
<tr>
<td>Publishing Date</td>
<td>1999-10-05 Year/month/day(2001-11-05)</td>
</tr>
<tr>
<td>Price</td>
<td>250</td>
</tr>
<tr>
<td>Image</td>
<td>jan gif</td>
</tr>
<tr>
<td>CategoryID</td>
<td>Humaniora</td>
</tr>
</tbody>
</table>

**Figure 4.5:** The book’s data record insertion process.
A database in reality must also update and delete records. This function could be performed by clicking the button to the right Update / Delete according to Figure 4.5.

Figure 4.6 demonstrates the appearance of the new Web page to an administrator when the update button is pressed. On the left side this grid contains two link buttons; one is for edit a record and the other is for deleting a record.

![Data Grid Control](image)

**Figure 4.6:** Data Grid control displays data for updating or deleting.

By clicking the Edit Command column according to Figure 4.7 the administrator will be able to see two new links buttons which provides them with the authorization to change or remove the displayable records from the database as seen in Figure 4.7. By clicking the cancel button it returns to the original state as shown in Figure 4.6.
Web application development with .NET- 3-tier architecture

Salle Dhali

<table>
<thead>
<tr>
<th></th>
<th><strong>BooksName</strong></th>
<th><strong>IsbnNumber</strong></th>
<th><strong>AuthorName</strong></th>
<th><strong>Price</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>Update</td>
<td>Delete</td>
<td>C++</td>
<td>0124-250-14</td>
<td>Kalle Jönsson</td>
</tr>
<tr>
<td>Edit</td>
<td>Delete</td>
<td>Java</td>
<td>1426-1425</td>
<td>Jan Skanelholm</td>
</tr>
</tbody>
</table>

**Figure 4.7**: Update or delete capabilities for the administrator.

The following paragraph describe how DataGrid control in ASP.NET Web forms take the opportunity to show the table data and by using some ASP.NET binding techniques which enable to be retrieved and to update the affected columns.

Firstly an ASP.NET DataGrid control [14] to the design page according to Figure 4.6 and this is also illustrated in [Appendix I]. In the properties for the object marks AutoGenerateColumns is marked as *false*. If AutoGenerateColumns is set to *true*, the number of columns in the DataGrid control will be created automatically after the number of columns in the data source table and will be bounded to the Grid control. The DataGrid task control can be conducted in the same Windows design view by merely marking the Property Builder which pops up immediately after clicking the control. The Property Builder shows the built in properties for configuring the data source such as Columns list, Paging, Format etc.

The identifying types of columns are the selected columns available from the source database and the Column lists the available columns that are part of the DataGrid control as follows:

- **Bound Column** - bound to a column of data in the data source.
- **Button Column** - can contain Button or LinkButton controls that the user can click on to perform something with a single record.
- **Template Column** - used to create a column that contains a combination of HTML and other Web controls than" Button". It can for example contain a CheckBox control.

The DataGrid control according to Figure 4.6 has designed by selecting four Bound Columns and a Button Link Column.
The first Column contains the *Edit Command Column* and the other four *Bound Columns* are the Item Template columns that enable the data fields to be bound to the data source BooksDetails data table. The data then can be manipulated by using a DataSet.

It is crucial to set the correct logic for *AutoGenerateColumn* is crucial. The BookDetails or other data tables may contain some fields that do not require to be updated or may not posses any viewable significant meaning for other users [Appendix J].

### 4.3 Login page for users

The other button in the first start page according to Figure 4.1 is the Log in function button. In Figure 4.8 the left authentication form is meant for both a registered member and for a new user to be registered. By entering the correct data controlled via a SQL server database, they will be allowed to put their resume data entries.

**Figure 4.8:** Illustrates the forms for registered and for the new users.
In the SQL Server database there is a database table holding the login information with both the Email and Password [Chapter 6]. After authentication and obtaining access control it will allows the users to redirect to another form which in turn offers the possibility to add data to the CV’s ("curriculum vitae"). The Registration button on the right, according to Figure 4.8, is intended for new users. It must be completed on the first entry so that the system can recognize them in the future in order to change their resume profiles. The registration procedure for both the new user and a user who has already been registered in the system and also the database table that holds those values for authentication are illustrated in [Appendix A].
5 Results

The basic requirements and specifications from the student union for their Web site was to create sites that will act in a user friendly way. In chapter 4 from Figure 4.1- Figure 4.8 shows how a user friendly Web system can be created using ASP.NET techniques. The wide varieties of server controls that are available in .NET framework allow developers to create powerful applications by using those controls. It is also possible to create a simple and basic Web page using only the plain HTML in Visual Studio environment. A very well defined and sophisticated Web site can be developed using the new ASP techniques and does not involve the writing of a great deal of code.

The ASP.NET custom UserControl feature allows for the reuse of code in relation to several pages for this application. It also assists in the building of a system that has many different features including footers, headers and in this project using the navigation links for both the administrator and users as well as the menu categories list interface elements for the pages. The UserControl uses classes that are the core behind the OOP concept which allows for better designed and structured code and in general it is easier to support a long term application life cycle. How it works is discussed in section 3.2, and how Web applications pages drag the out the results are illustrated in [Appendix E].

The code reusing capabilities and rewriting similar code blocks such as the add user for the CV data entry module are discussed in section 3.6.1. The insert methods used for both the Student and Book classes follow the same pattern. The class diagram and the other code for adding objects for book detail records are shown in [Appendix F]. The insertion objects are similar for both the methods mentioned in this section.
The different SQL operations such as the Insert, Delete and Update commands can be declared as a Stored Procedure. Those procedures can be applicable to different portions of the data logic layer both in the Books module and the CV data entry module. Insert a books record procedure is discussed in section 3.5.4 and the same pattern model applies to add a new user which is presented in [Appendix G].

The BooksCategory and BooksDetails data tables will show all the books belonging to that category so both the tables had a common key policy that is the Category data Id which holds both tables values. The table formation is illustrated in [Appendix H]. The CategoryId and the UserId and BooksId have Integer data key values. Most of the data fields used in this project are of the Unicode data types.

The program is tested on both a local and also on the web environment. A number of different users equipped with computers have also tested it over the Internet in order to determine faults and also to determine the behavior of the program. The results were quite satisfactory. The project had a certain limitations which generated a certain amount of profitable result. The complements to the various parts for further development and program optimizing appear in Section 6.1 of the future working processing section.

The system architecture is discussed in section 3.4 and the benefit using the distributed web application is dealt with in the conclusions [chapter 7] section. This part also discusses the new programming language C# and its benefit in relation to integrating with the Integrated development environment Visual Studio.NET. The SQL commands creation with the Stored Procedure and the capabilities that VS provides in relation to manipulating the insertion of data values is presented in [Appendix G].

The tools and technologies that are used to create the Web application system such as the C# language, ASP.NET techniques, HTML, XML, Visual Studio.NET, SQL server 2000 for the whole database management are discussed in chapter 2. The results for the updating function for a books record, a single part of the whole application system that uses those tools and technologies can be seen in Figure 4.7 and [Appendix I, D].
6 Discussion

Much has changed in relation to the technical aspect view since the initiation of this project. As previously stated the work environment was Visual Studio 2003, the first released version of ASP.NET and a SQL Server 2000. At that time the development of any Web application system required the installation of an IIS server on the local machine. After the installation it creates a catalog named Inetpub under the C directory. It then enables the creation of Web applications and stores the entire project files such as the .Aspx files, resource files, image file, DLL and the component files such as the C# source code files under the sub directory catalog folder named wwwroot.

The Microsoft Integrated Development Environment Visual Studio 2010 provides the ability to create Web applications without installing the IIS Web server. It automatically creates a Web application under the Visual Studio 2010 project in the C: part of the computer.

At the start of the project a demonstration of this Web application to my tutor in the real environment i.e. via a Web server, a number of problem arose. Firstly, the majority of Web hosting company no longer supports a connection to a SQL Server 2000 environment. Thus the choice of a SQL Server 2005 or 2008 would a better option. However the available Operating System was Windows Server 2003 then it was not possible to install SQL Server 2008 on that environment.

Microsoft has closed all updates and support for both Windows Server 2003 and SQL Server 2000. Therefore, the choice was made to use Windows Server 2008 and the latest development environment Visual Studio 2010. An attempt was made to export and convert the project into the new environment but it proved impossible to create any solution files. It finally worked when a manual copy was made file by file to the new project. It should be noted that old files created by the project such as DLL (Dynamic Linked Library), WebConfiguration file, and resource
files should not be included in the newly developed ASP.NET Web applications.

In 2004, on the first deployment of the application for my tutor Mehdi Raoufi it was possible to choose many companies that had support for both SQL Server and ASP.NET Web hosting in one package which were free for one month. Moreover they had FTP (File Transfer Protocol) access for uploading files. The present situation is different, as not many companies now provide free test drive. My ASP.NET files located on a Web server, ASPspider.info, that is not a profit driven company and is run and operated by Computer Science students around the world. One must be a member to be able to use their services.

My SQL server database file located on another SQL server hosting corporation, Parking.ru, allows only a limited amount of stored media capacity and for a short period of free test driving.

The password in Figure 4.3 is visible for viewing purposes. In a real application, it should not be formed or designed in such a way. The password should be shown as an asterisk mark.

Some Figures in chapter 4 use texts and pictures taken from the Student-litterature’s [17] Web sites. This was meant only for experimental purposes of inserting those text and pictures data to the SQL server database and to retrieve it when necessary. This helps to test the application.

.NET framework 2.0 and ASP.NET 2.0 allow a developer to use the TreeView menu architecture in the VS environment which has proved to be very popular during the last couple of years. The recommendation to the reader is to use this control to bind any data source instead of using DataList control which was chosen to present the menu item. It is also suggested to use ConfigurationManager instead of the previously used configuration Settings keyword [Appendix E].

The following procedure is important in order to insert any new record to any tables belonging to a database after exporting databases from the SQL server 2000 to the 2008. It could be performed in the SQL Server
6.1 Future works

There are many aspects to be considered in the future due to the application’s improvement, performance, efficiency, convenience and appearance. Another part of the project is to add CV related data for the students and part of this code has already been written using C# language. This should prove useful for turning this project into a finished product with the help of these resources that are contained within the project.

As the CV section system gradually builds up by using various ASP.NET Web forms it can generates significant advantages if it becomes possible to construct a standard template on which all information, including personal details, professional working experience, training period and educational background could be accumulated such as in a single document sheet either to print out or to be attached in an e-mail program for sending to a company.

The database table that normal users can connect to has only limited attributes. To maintain the CV data entries there is a requirement for an additional table. By joining those two tables it will be easier to store data for the user and allow them to participate in a discussion forum or a news group etc. Those tables should be normalized prior to working with that as it is firstly necessary to identify a primary / foreign key. Further development of this application can lead to an online shopping book store for the Union. Those table values can be used in a similar manner to a student->user->and customer relational data records. To add some buttons in the Web form and writing some code can enable the students to have the ability to order books via on-line.

In this business layer there is a C# "Security“ file under application data components. This namespace “System.Security.Cryptography“ must be included in the source file. By using this function it enables the passwords to be encrypted for secure use.
It is possible to significantly improve the performance of applications by taking advantage of the caching capability in ASP.NET. There are three types of caching available and they are output caching, partially used in some of the pages. The two other are the fragment caching and data caching.

The application can be improved by using an upload button shown in Figure 4.5 where an administrator can look at the image from its local portion of the hard drive and upload it to the database. In the present situation it is necessary to know what the image file name and extension are such as .Jpeg, .gif, .bmp etc. and write it to the AdminInsert Web form Textfields manually.

The administrator username and password is written in the source file which can be seen in Figure 4.4. This part can be developed by connecting it to a newly developed data table. It could also be possible to link the administrator to the existing data table that is available for the ordinary users. The new Stored Procedure creation and the new data entry in terms of coding for the business logic layer with the details for an administrator can be accomplished using the source code that is attached in section 3.6.1 and [Appendix F]. It is also recommended to use the security method code for encrypting the password data residing in the data table for the administrator. Even if someone is unwilling to create a new table and relay the user data table that is available in the database, it is still possible to develop this component portion for an administrator in order to extend the security.
7 Conclusions

The core element of any web application is the system architecture that will enable those pieces to interact with each other. The functionality of the application and the performance can thus resulting in an excellent system development. The 3-tier data architecture used to create this Web application applying .NET technology definitely has many benefits discussed in the following paragraphs:

The creation of a distributed Web application has become much easier and simpler by using Microsoft .NET technology. This logical design model is segmented into three layers for a single application component, each part or segment can be developed by different types of developers. The system can also be integrated with different programming languages. As the logical process is separated for an application any changes made to a single layer will not affect the two other segments/layers of that application. Moreover, the rapid deployment settings of an application with Assemblies rather than the DLL registry complication have significantly assisted the developers.

This architecture allows the sharing of the codes, services and reusing the components in a distributed computing system where many computer users can take advantage of sharing their works to build more complex systems. This also permits the ability to transform the customer’s demands and particular requirements which could prove of assistance in moving from a simple Web application solution to a multilayered component system. The system offers the benefit of creating scalable solutions which have high performance, flexibility, cost and maintenance and reusability. As it is possible to develop more modules for this system it is felt that this is a suitable choice for the Student Union. Students or developers not acquainted with the C# programming language but who have the VB / Visual C++ programming experience can easily develop the additional parts and particulars on behalf of the Student Union’s requirements.

The negative part of using 3-tier data architecture is that the client side application will be much lighter as the service components are placed on
the middle tier which results in a great deal of network traffic for those systems as components are distributed among many different machines. The advantage of using the architecture outweighs the negative side as the network traffic can be minimized by optimizing the Stored Procedure that can minimize the network round-trip and many more jobs can be placed to the client, the presentation or the User Interface layer.

Visual Studio.NET the most popular development tool from Microsoft allows us to create applications with all the .NET languages. The newer version has an advanced debugging system, database testing and SQL creation. Moreover it allows there to be integration with previous .NET version applications. For instance, this project was created using an early version of .NET framework but it has still proved possible to integrate it with the newer version. Visual Studio package and .NET framework helps to encapsulate all the layers that are involved in 3-tier data architecture.

The language C# has many sophisticated, helpful and handy features that assist a developer to create applications in a very simple way. It makes life easier as an application can be implemented by copying or rewriting similar pieces of code to different source files and this was used in this project. It allows dynamically creating elements in an array, a generic list to insert objects, a well defined sorting and searching algorithm, dictionary method such as key/value type collections and the query type expressions language.

Moreover ASP.NET Web Forms, different types of controls, caching capabilities, working accessibility with different types of heterogeneous data sources both local machine and any remote devices is a great achievement for a distributed computing system. The ASP.NET technique that uses the .NET framework architecture allows the creation of an application by totally using the Object Orienting programming (OOP) concept and paradigm which allows encapsulation, inheritance, properties, methods and reflections etc. OOP allows for the writing of well defined structured code that gives the ability to clear the separation of content such as the business logic rules from the data presentation layer and also supports a long term application life cycle.
There are some great features such as the ASP.NET pages, which after compiling for a first request any subsequent ones for that page minimize the access time. The same is true for the Stored Procedure as it is also pre compiled and resides on the server. .NET based applications can be easily integrated for the next generation of Windows Operative Systems as this framework will be an integrated part of all the upcoming OS from Microsoft. Moreover system vulnerability, caused previously by using Microsoft product, has significantly improved as the company releases security bulletins, updates the products by releasing service packs and by providing a free security essential anti-virus program.

Even though Microsoft .NET platform has many advantages for creating applications the question remains as to whether it is worth migrating the Student Union’s present Web application built using the PHP and MySQL.

The negative parts of using Microsoft products are the cost of using those products. All the products are tightly coupled with each other which mean that all the tools and technologies must be Microsoft compatible for developing an application. The Web server IIS, OS, ASP.NET technologies applying .NET framework, programming language C# and the SQL server must be used in order to take the benefit of using an application under .NET framework. In the long run it can be very costly. Because updating a new product may involve other related products requiring to be updated. Most of the tools used in this Web application are implemented by the Microsoft Corporation and thus lead the developers to creating applications only for the desktop environment. This project is called a Web based application as its behavior is similar to a Windows based application. As the license fee is costly so the deployment for an ASP.NET application with SQL server are expensive.

On the other hand PHP and MYSQL database are fully capable of creating well defined, sophisticated and complex Web applications. The advantages using PHP are its platform independent engine that allows parsing of almost all the OS. PHP is free and works in an elegant fashion with the open source popular MySQL and the popular Apache Web server which are also free. PHP has large open community services including blogs, discussions forums and open source code distributions.
The disadvantage facing mostly PHP is in not supporting the whole OOP concept. PHP tends to execute slowly but buying add on products can optimize the Zend Opcodes, low level binary code runtime execution. PHP and MYSQL Web hosting is relatively very cheaper than ASP.NET hosting.

It is not difficult to transform a simple Web site from PHP to ASP but it requires a great deal of planning and reconsideration for changing to a bigger application as the OOP paradigm must be considered. The PHP scripting language has almost all the functionalities required to build dynamic Web sites. It could still be the choice for the Student Union to develop further on their sites using the same technologies. It might prove necessary in the near future to buy a number of other add on components including the image manipulation, integrated developments tools, native API catch etc. in order to integrate the overall system. The other option for the Union is to choose the Microsoft .NET framework technologies as the ASP.NET is considered to be very powerful, robust, event driven and built entirely using the OOP concept. It is up to the Union to make the decision which technology they will use. In term of cost PHP may the winner but the whole packaging system that Microsoft offers even if money may prove to be the ultimate decider for further development of the Student Union’s current Web sites.
References


Web application development with .NET- 3-tier architecture

Salle Dhali

References
2012-06-11


http://msdn.microsoft.com/en-us/library/y6wb1a0e.aspx
Taken March 17, 2012.

Taken January 25, 2012.

Taken March 8, 2012.


Taken January 29, 2012.

Appendix A: Bio-data access, registration and control validator

<table>
<thead>
<tr>
<th>UserID</th>
<th>UserName</th>
<th>UserEmailAddress</th>
<th>UserPassword</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>salle</td>
<td><a href="mailto:salle@msn.com">salle@msn.com</a></td>
<td>test</td>
</tr>
<tr>
<td>2</td>
<td>jimmy</td>
<td><a href="mailto:jimmy@msn.com">jimmy@msn.com</a></td>
<td>test</td>
</tr>
<tr>
<td>3</td>
<td>lalle</td>
<td><a href="mailto:lalle@msn.com">lalle@msn.com</a></td>
<td>test</td>
</tr>
<tr>
<td>4</td>
<td>Dr.ulf Jannenhag</td>
<td><a href="mailto:ulf@miun.se">ulf@miun.se</a></td>
<td>sundval</td>
</tr>
<tr>
<td>5</td>
<td>rebecca</td>
<td><a href="mailto:rebecca@msn.com">rebecca@msn.com</a></td>
<td>arlov</td>
</tr>
</tbody>
</table>

**UserName:** rebecca

**Email:** rebecca@msn.com

**Password:** •••••

**Confirm Password:** •••••

Resister
ASP.NET required field validator control.
Appendix B: Educational data for CV

[Image of a form with fields for educational data]
Appendix C: Attribute values before changing

![Microsoft SQL Server Management Studio screenshot]

<table>
<thead>
<tr>
<th>BookID</th>
<th>BookName</th>
<th>AuthorName</th>
<th>PublishingDate</th>
<th>Price</th>
<th>BookChanges</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>sall2012</td>
<td>Kalle 1234</td>
<td>14/12/2012</td>
<td>350.00</td>
<td>null</td>
<td>Java Programming</td>
</tr>
<tr>
<td>2</td>
<td>Java</td>
<td>Jan 1234</td>
<td>05/12/2012</td>
<td>500.00</td>
<td>null</td>
<td>Java Programming</td>
</tr>
<tr>
<td>3</td>
<td>sql2012</td>
<td>moto 1234</td>
<td>15/12/2012</td>
<td>0.00</td>
<td>null</td>
<td>Java Programming</td>
</tr>
<tr>
<td>4</td>
<td>sql2012</td>
<td>moto 1234</td>
<td>15/12/2012</td>
<td>0.00</td>
<td>null</td>
<td>Java Programming</td>
</tr>
</tbody>
</table>

...
Appendix D: Results after changing records
Appendix E: User Control reuse benefits
Appendix F: Class diagram and code reuse techniques

```csharp
public string AddBooks(string bName, string isbnNumber, string aName, string pDate, string price, string images, int cid)
{
    // Create Instance of Connection and Command Object
    SqlConnection myConnection = new SqlConnection(ConfigurationManager.AppSettings["ConnectionString"]);
    SqlCommand myCommand = new SqlCommand("BooksAdd", myConnection);
    // Mark the Command as a Stored Procedure
    myCommand.CommandType = CommandType.StoredProcedure;
```
// Add Parameters to the Procedure
SqlParameter parameterBooksName = new SqlParameter("@BooksName",
SqlDbType.NVarChar, 50);
parameterBooksName.Value = bName;
myCommand.Parameters.Add(parameterBooksName);

SqlParameter parameterisbnNumber = new SqlParameter("@IsbnNumber",
SqlDbType.NVarChar, 50);
parameterisbnNumber.Value = isbnNumber;
myCommand.Parameters.Add(parameterisbnNumber);

SqlParameter parameterAuthorName = new SqlParameter("@AuthorName",
SqlDbType.NVarChar, 50);
parameterAuthorName.Value = aName;
myCommand.Parameters.Add(parameterAuthorName);

SqlParameter parameterPublishingDate = new SqlParameter("@PublishingDate",
SqlDbType.NVarChar, 20);
parameterPublishingDate.Value = pDate;
myCommand.Parameters.Add(parameterPublishingDate);

SqlParameter parameterPrice = new SqlParameter("@Price", SqlDbType.Money, 8);
parameterPrice.Value = price;
myCommand.Parameters.Add(parameterPrice);

SqlParameter parameterBooksImage = new SqlParameter("@BooksImages",
SqlDbType.NVarChar, 50);
parameterBooksImage.Value = images;
myCommand.Parameters.Add(parameterBooksImage);

SqlParameter parameterBooksCategory = new SqlParameter("@CategoryID",
SqlDbType.Int, 4);
parameterBooksCategory.Value = cid;
myCommand.Parameters.Add(parameterBooksCategory);

SqlParameter parameterBooksID = new SqlParameter("@BooksID",
SqlDbType.Int, 4);
parameterBooksID.Direction = ParameterDirection.Output;
myCommand.Parameters.Add(parameterBooksID);

try
{
    myConnection.Open();
    myCommand.ExecuteNonQuery();
    myConnection.Close();

    // calculate BooksID, using the Output Parameter from the Stored procedure
    int BooksId = (int)parameterBooksID.Value;

    return BooksId.ToString();
}

catch
{
    return String.Empty;
}
Appendix G: Insertion methods pattern
Appendix H: Database tables, keys and relations
Appendix I: DataGrid control for update purpose

```xml
<asp:datagrid id="DataGrid1" style="z-INDEX: 101; LEFT: 57px; POSITION: absolute; TOP: 71px; runat="server"">
  <columns>
    <asp:BoundColumn HeaderText="BookName" />
    <asp:BoundColumn HeaderText="IsbnNumber" />
    <asp:BoundColumn HeaderText="AuthorName" />
    <asp:BoundColumn HeaderText="Price" />
  </columns>
  <RowTemplate>
    <asp:TextBox id="edit_bookname" runat="server" width="120px" Text='<%# DataBinder.Eval(Container.DataItem, "BookName") %>' /></asp:TextBox>
    <asp:TextBox id="edit_isbnnumber" runat="server" Text='<%# DataBinder.Eval(Container.DataItem, "IsbnNumber") %>' /></asp:TextBox>
    <asp:TextBox id="edit_authorname" runat="server" Text='<%# DataBinder.Eval(Container.DataItem, "AuthorName") %>' /></asp:TextBox>
    <asp:TextBox id="edit_price" runat="server" Text='<%# DataBinder.Eval(Container.DataItem, "Price") %>' /></asp:TextBox>
  </RowTemplate>
</asp:datagrid>
```