Application System for File Uploading using ASP.NET

Ali Al-Farsi
Student, Computer Engineering
Caledonian College of
Engineering Muscat, Oman

Ali Al-Mahruqi
Senior Lecturer
Caledonian College of
Engineering Muscat, Oman

Jayavrinda Vrindavanam
Senior Lecturer
Caledonian College of
Engineering Muscat, Oman

ABSTRACT

Web services have been a fast evolving field fuelled by the ever-changing customer expectations. As new technologies emerge, investigating and introducing new tools can optimize the time and improves the methods of building web-oriented enterprise applications and web services. With every such advancements, the interface used for inserting new records has to be updated to allow the user to upload a file from their computer and extra steps must be taken to display or provide means of downloading record associated with binary data, which is challenging. In this context, in this paper, a fully functional application is displayed and associated it with many types and size of data to upload by using ASP.NET. The paper reviews different techniques for storing data and explore how to enable users to upload a file from their computer and have it saved on the web server's file system. This would provide the basis for easy ways of storing and retrieving data directly within the database which involves a bit more code than working with data through the file system. Additionally, in order for a user to view or download data they must be presented with a URL to that data. If the data resides on the web server's file system, the URL would be a straightforward process. And if the data is stored in the database, a web page needs to be created and retrieve and return the data from the database. Further, if the data is stored on the file system, the demand and network congestion between the database server and web server will be less than when the data is stored directly within the database.

General Terms

Active Server Pages (ASP.NET), C Sharp (Programming language C#)

Keywords

File upload control in ASP.NET 2.0, Upload files using C# language in ASP.NET by Microsoft Visual Studio 2012.

1. INTRODUCTION

Nowadays, modern technology offers multitude of applications and systems which can be used to communicate with each other. From the perspective of this study, a lot of technologies can be used to allow the user to interact with the application system and to get a reliable connection between the system application and the end-user platform. With every such advancements, the interface used for inserting new records has to be updated to allow the user to upload a file from their computer and extra steps must be taken to display or provide means of downloading record associated with binary data, which is challenging. In this context, in this paper, a fully functional application is displayed and associated it with many types and size of data to upload by using ASP.NET. The interface is used to insert new records in order to be updated and that will allow the user to upload a file from their computers with any devices. The use of binary data bring into the fore, its own challenge. It must be decided how the binary data will be stored in the application system. The home page in ASP.NET acts as an interface for the application system that uploads the files and inserts the records which has to be updated with the system. It gives a privilege for the user to upload the file from their personal computer system and provide a download link to download that files which is uploaded.

The paper reviews different techniques for storing data and explore how to enable users to upload a file from their computer and have it saved on the web server's file system. This would provide the basis for easy ways of storing and retrieving data directly within the database which involves a bit more code than working with data through the file system. Additionally, in order for a user to view or download data they must be presented with a URL to that data. After a review of the literature, the paper explains the framework of the proposed system. Thereafter, the design and the results etc. are discussed and finally conclusion.

2. LITERATURE REVIEW

The paper has undertaken a review of the literature relating to the study. Bill Even and Reuters [1] applied Microsoft ASP.NET 2.0v and Microsoft Visual Web Developer 2005 Express Edition application system that deals with '**FileUplaod'** sever which is managed by Microsoft ASP.NET 2.0. Since Microsoft ASP.NET introduced version 1.0, there was built-in ability of building Web Applications that had the competence to upload files to the hosting server. This method was already done by using the '**File Field'** HTML server control. Bill Evjen has written an MSDN article on the way to use the control in ASP.NET application effects. In this research, the authors have used limited amount of size to upload only one file at the time, therefore the limitation of this research consider the size and the type of file that has been chosen by the user, compared to the proposed idea which presents more advantages increasing the size limitation from 2MB to 100MB and above.. Another research [2], introduced File Upload Control with ASP.NET and AJAX. It has been created by using ASP.NET and uses **File Upload** control to upload the files on server-side. In such a way of using AJAX for Scott Hanselman with **FileUpload** control, the author found out that none value in **FileUpload** control **[HttpFileCollection]**, therefore the **FileUpload** control that requires a full postback to upload the file.

The '**FileUpload'** control won't work within an '**Update Panel'** control for uploading the files using Asynchronous Post-back. In case of security decreed by browser which doesn't allow for the JavaScript to access directly to the file in the user system, Synchronous Post-back can be used to upload any file. ASP.NET AJAX allows the two sides of Synch & Asynch Post-back. The author present the file upload control which uses AJAX technology to allow the two sides of synchronization and asynchronization that allow to access directly to the file system of the user, the limitation of this research is that, the update panel control does not work with file upload control, therefore it acts as the advantage that will avoid the use of update panel. In a similar research, DevExpress [3], organization introduced **FileUpload** control. They developed and designed an Upload Control **[ASPxUploadControl]** that allows the users to upload their files in the web application in an easy way. When a user needs to place the control onto the webpage, end-user can select the files and browse for certain file in their system to upload it to the server by entering the file's path into the field of textbox or using the Open File dialog. The markup code introduced in the study demonstrates how to add the **'ASPxUploadControl'** to any project for the purpose of uploading the file to the FTP server and also to display Add, Remove & Upload buttons. It also enables the bulletin of progress panel; therefore users can automatically be notified as uploading is in progress during the uploading time. There are two types of modes, the standard and advance mode which allows more flexibility to the users. It also includes the file validation which allows the uploading of more than one file at a time. As that is the case, this idea includes more efficient tools that improve the uploading progress.

Harald Kirschner [4] presented '**FancyUpload'** – that meets Ajax technology, which has improved the upload file system by '**MooTool'** that make the application system works in all new browsers. '**FancyUpload'** support multi-file selection with menu and queued bar progress for the upload. It's created by CSS and XHTML web pages. José Rolando Guay Paz [5], developed a new concept of Uploading Files called "RadUpload", it's designed in order to upload the files from the client PC to the server. Many options can be configured by this RadUpload to make the uploading file more flexible and easier for the user to use. In addition, it provides a powerful alternative for the old input controls. The 'RadUpload' enables the application for file upload, to enable this control, should be registered for the module in the *web. config* file. There are 3 different ways to enable 'RadUpload' in the application, manually by adding the *httpModule* model in the section of *the web. Config*. RadUpload provides to the user a wizard to accept the files to be uploaded, and it just takes a few steps to configure it. The *RadStyleSheetManager* & *RadScriptManager* helps the developers to manage the CSS resource & JavaScript files by minimizing and combining them to be requested from the server.

Dino Esposito [6], has designed an application by using ASP .NET for Uploading Files. In this application system they used the *HtmlInputFile* control tools. HTML tool is used in this application to upload the files to the web server by using the browser. The *HtmlInputFile* control consists of a textbox and a browse button. Users can select the files from their local PC, then clicks and select the files to be uploaded to the server. While this process occurs, the selected files will be uploaded to the server. [7] On the server - side, the uploaded file will be posted into an object of type *HttpPostedFile* and it waits until it is processed. The *HttpPostedFile* object support methods and properties that receive the information on an individual file to read & save the file. Also, the author mentioned another property called *Input Stream* of the *HttpPostedFile* to read the posted file/data before the processing. It allows limiting the file types to upload it to the Web server.

[8] From the review, it's clear that the studies are quite similar to each other and on a broader comparison, the study [1], present an application system that has a limitation on the file size that is uploaded by using this technique. The maximum size for each uploaded file on the server using the **FileUpload** control is only 4MB. Therefore, it may limit uploaded file to the specified amount of 4MB for each file and it is not possible to upload more than this size. One of the ability of .NET is, it provides a way to handle the limitation on file uploaded. It can change the value of default settings in place of the limited uploaded file size per time. Therefore, the way to change size by the **'web.config.comments'** or on the application's ' web**. config'** file. Further [3] the other study used a different codes to maximaize the size to be uploaded from the user side which makes it unlike the other introduced application system that used the **ValidationSettings. MaxFileSize** this characterizes used to identify the allowed file size for each uploaded file. Also, the **ValidationSettings. AllowedFileExtensions** this characterizes makes the specific file extensions can be used to upload to the server.

Eventually, among all research and different systems reviewed, we observe that, the system as per the study [1], proposed a system that is similar to the proposed idea in some aspects, since it utilizes upload file control and use it as function to upload the files to the database SQL or the other way to store it in local file system. The concepts and ideas [9] look very similar even though the application system has implemented File Upload control by using different technologies.

3. SYSTEM FRAMEWORK

3.1 FLOWCHART

The flow chart of the proposed system is shown under:

3.1.1 Login and Upload the file to the server

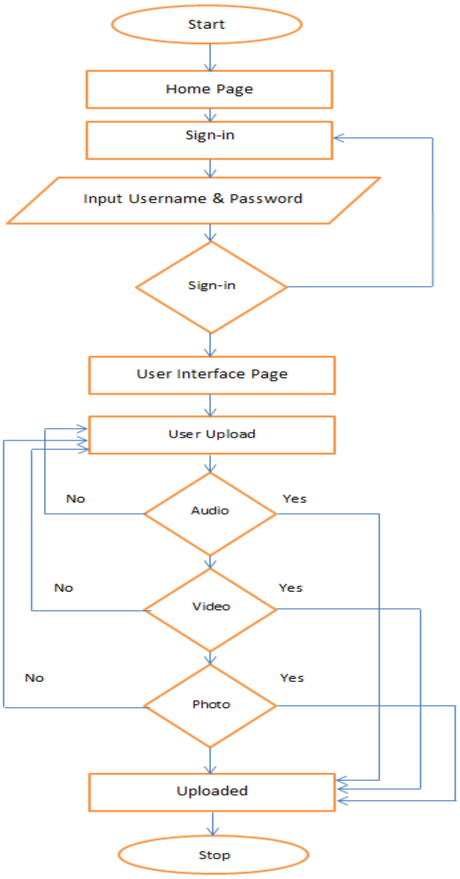


Fig1: Login & Upload file to the server

3.1.2 Algorithm of the proposed system

i. Launch the ASP.NET program
ii. Read "home.aspx"
iii. Read "login.aspx"
iv. Read "Username & Password"
v. If username and password are correct go to "default.aspx" as the user name account.
vi. If the users upload any type of "audio", "video" and "picture" then store the file in the path for specific folder in the account.
vii. Else show massage failed.

3.1.3 The process of the queue for each file to be uploaded

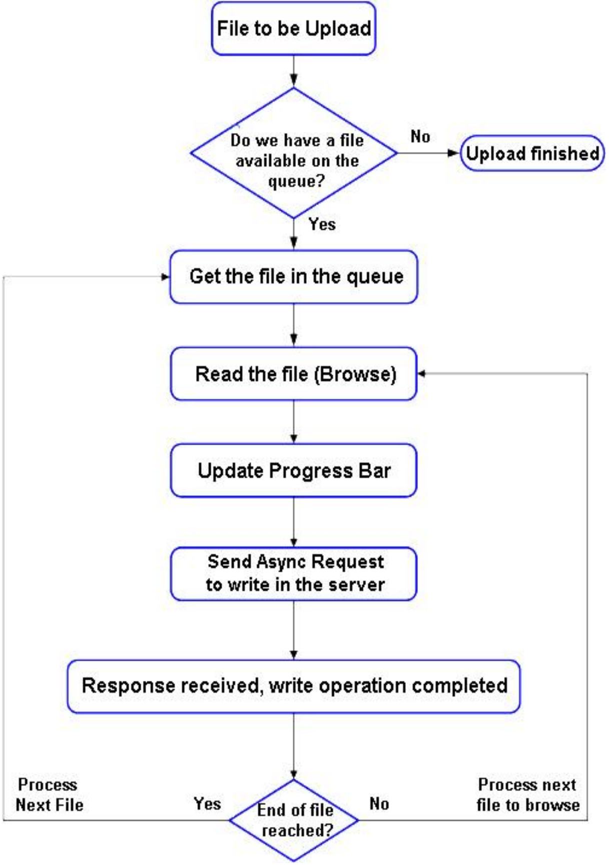

File to be Upload
Do we have a file available on the queue?
No
Upload finished
Yes
Get the file in the queue
Read the file (Browse)
Update Progress Bar
Send Async Request to write in the server
Response received, write operation completed
Process Next File
Yes
End of file reached?
No
Process next file to browse

Fig 2: Process of the queue for each file to be uploaded

The following is the algorithm used for the process flow as per Fig.2.

3.1.4 Algorithm

i. Read the file to be uploaded
ii. If a file is in the queue, then stores the file to the folder which the account has been created.
iii. Else show upload is busy.
iv. Browse for the file in the computer directory.
v. Send a request to send the file to certain folder.
vi. If finished upload, then go to stop

4. DESIGN

This application system includes only the system design which is designed to upload the files to the local host and encrypt the data which is uploaded. This file will be decrypted since the user account download the file from his/her account directly, which provide the file to be open or modified by any other users within the local system.

4.1 E-R DIAGRAM

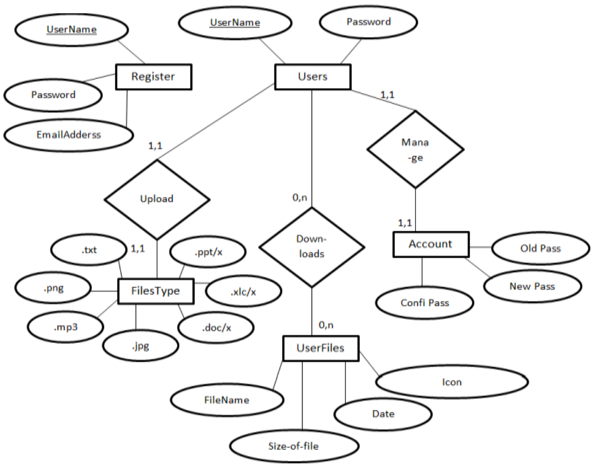


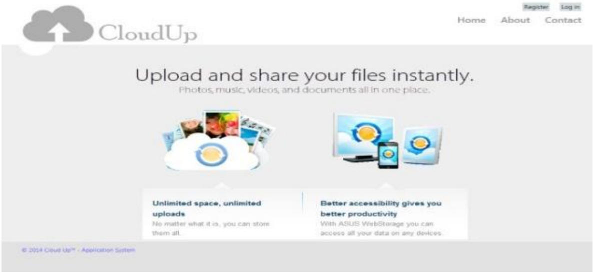
Fig 3: E-R Digram

The proposed system can upload different types of files as mentioned in ER-Diagram.

4.2 APPLICATION SYSTEM DESIGN

4.2.1 Home page

When a user launches the application system, the page as shown below will appear as home page for Cloud Up application system. The home page consists of many buttons in navigation bar; also consist of a banner with a logo for the application system. It has a user friendly interface with a “Register” button which is responsible to link to another page for the purpose of making a new registration of users. A login button will also be provided for registered users to login directly to the system. The moment the user logs in, a new folder will be created dynamically with the name of the log in user. The pages of the site are controlled by “Site Master” in order to structure the order of the pages so as to allow flexibility to the users.



Upload and share your files instantly.
Photos, music, videos, and documents all in one place.

Unlimited space, unlimited uploads
No matter what it is, you can store them all.

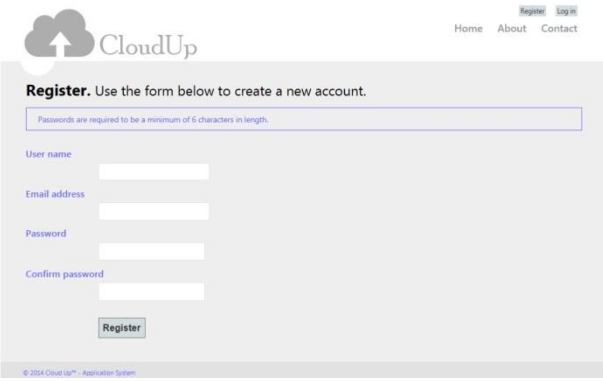
Better accessibility gives you better productivity
With ASUS WebStorage you can access all your data on any devices.

© 2014 Cloud Up™ - Application System

Fig 4: Cloud Up home page

4.2.2 Registration page

This page takes users to register their new account, and ask for the user to fill the field to complete the registration. This step will make a new user account in the database list that stored in an ASP.NET WSAT database and gives the credentials to access the user account page.



Register. Use the form below to create a new account.

Passwords are required to be a minimum of 6 characters in length.

User name

Email address

Password

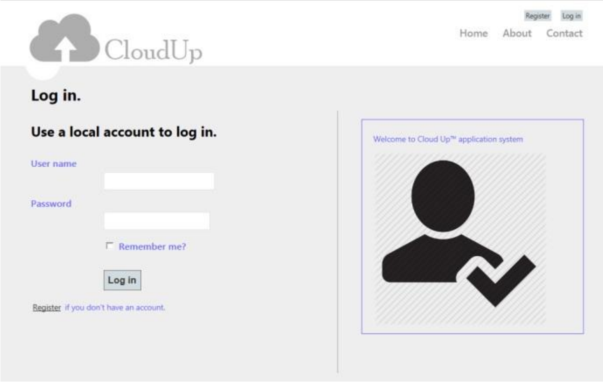
Confirm password

© 2014 Cloud Up™ - Application System

Fig 5: Registration page

4.2.3 Login page

After the user has successful registered to the application system, the details will be stored in the database. The registered user will be able to login after been recognized by the system. After a successful login attempt, the user will have an interface to upload the file to the system.



Log in.

Use a local account to log in.

User name

Password

□ Remember me?

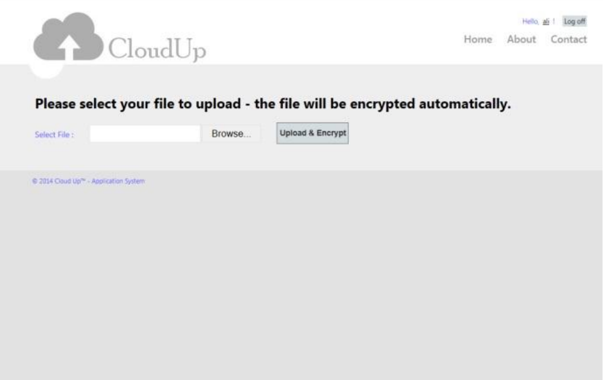
Register if you don't have an account.

Welcome to Cloud Up™ application system

Fig 6: Login page

4.2.4 User account interface

After the user account has been logged in, the interface will appear. The interface will consist of a “Browse” button that will enable the logged in user to browse and upload any type of files with no limitation in size. A folder will be created after the upload of the files is completed. The folder can be reused for the future uploads. The interface will also allow the user to log off and direct them to the home page.



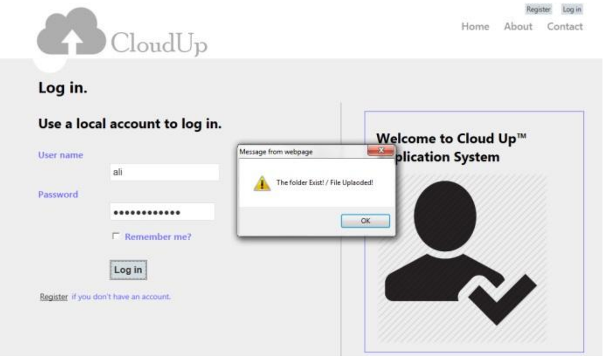
Please select your file to upload - the file will be encrypted automatically.

Select File :

© 2014 Cloud Up™ - Application System

Fig 7: User account interface

Here the user account can log in to the account, since the log in is successful, then, it will directly take the user to the user account interface page and the folder will be created to the account by the username account as indicated in below ,fig. 7.



Log in.

Use a local account to log in.

User name: ali

Password: ••••••••••••

□ Remember me?

Register if you don't have an account.

Message from webpage

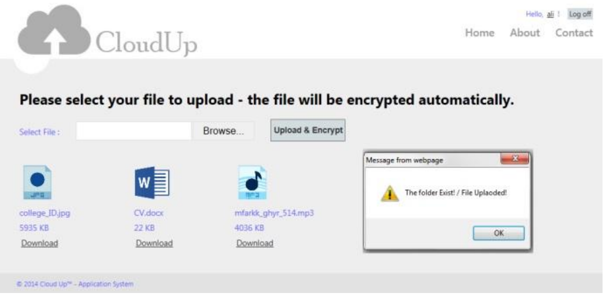
The folder Exist! / File Uplaoded!

Welcome to Cloud Up™ plication System

Fig 7: Login Page

4.3 User account upload file

Since the user has been logged in, then interface of user account will be created as well as separate folder will be created for the registered user account. The file will be uploaded within a seconds due to the local host network, therefore the file can be located to the private path of the user account "username".



Please select your file to upload - the file will be encrypted automatically.

Select File :

college_ID.jpg
5935 KB
Download

CV.docx
22 KB
Download

mfarkk_ghyr_514.mp3
4036 KB
Download

Message from webpage

The folder Exist! / File Uplaoded!

© 2014 Cloud Up™ - Application System

Fig 8: File uploaded

5. RESULTS

The design of any application system should go through the implementation and testing processes. The outcome of this application system will be based on the accomplishment of the aim and objectives. As for the purpose of this research, the success is measured in terms of the system functionality. Since the system was developed for the purpose of uploading files application, the system has shown a fruitful result of encrypting the files of users and provides maximum security.

6. CONCLUSION

This research has been accomplished and covered the intended objectives. The research has reviewed the existing systems of different Uploading application system with different technologies and structures. Essentially, for each Uploading File application system should have a browser and support the .NET technology on their system. A lot of technologies can be used to allow the user to interact with the application system and to get a reliable connection between the system application and the end-user platform (Example. Internet). ASP.NET technology is supported by many languages and it has more speed and several modes of the security. To summarize, the proposed application system, allows user to use this technology in many systems and different browsers that are based on ASP.NET technology.

The system has developed to store the data and allow a user to upload binary content through a web page which is associated with the application data model that can be stored in one of two places: on the web server's file system with a reference to the file stored in the database; or directly within the database itself. The system can also associate the uploaded data with a record in the database as well as how to display the binary data alongside its text data fields.

The FileUpload Web control provides a mechanism for users to send a file from their computer to the web server. The FileUpload control renders as an <input> element whose type attribute is set to fail, which browsers display as a textbox with a Browse button. Clicking the Browse button brings up a dialog box from which allow the user to select a file. When the form is posted back, the selected file's contents are sent along with the postback. On the server-side, the information about the uploaded file is accessible through the FileUpload control's properties. The FileUpload control provides a variety of properties for working with the uploaded data. For instance, the "Has File" property indicates whether a file was uploaded by the user, while the "File Bytes provides access to the uploaded binary data as an array of bytes. The Click event handler starts by ensuring that a file has been uploaded. If a file has been uploaded, the Label shows the name of the uploaded file, its size in bytes, and its content-type.

7. FUTURE WORK

The existing system allows users to upload only one file at a time, as for the future work, we are intending to research on the possibilities of adding new row that can upload multiple file at a time. Also it will consider the use of Drag/Drop technology that will enable the users to drag and drop a file of any size from one location to another in order to simplify and provide flexibility to the users. Since the system is accessed vial local host, the authors are working to upload the stem to the webserver with extra security features of encryption.

8. REFERENCES

[1] Bill Evjen, Jay Glynn, Christian Nagel, Karli Watson, Morgan Skinner, 2010. *Professional C# 4.0 and .NET 4*. 4th ed. US: John Wiley & Sons, 2010.

[2] AjaxFileUpload Sample . 2013. *AjaxFileUpload Sample .* [ONLINE] Available at: http://www.asp.net/ajaxlibrary/ajaxcontroltoolkitsamplesite/ajaxfileupload/ajaxfileupload.aspx.

[3] Example - ASP.NET AJAX File Upload Control Demo | DevExpress . 2013. *Example - ASP.NET AJAX File Upload Control Demo | DevExpress* . [ONLINE] Available at: http://demos.devexpress.com/ASPxFileManagerAndUploadDemos/UploadControl/Example.aspx.

[4] Single File Button » FancyUpload - Swiff meets Ajax Showcase » digitarald:Harald Kirschner. 2008. *Single File Button » FancyUpload - Swiff meets Ajax Showcase » digitarald:Harald Kirschner*. [ONLINE] Available at: http://digitarald.de/project/fancyupload/3-0/showcase/single-file-button/

[5] Jose Rolando Guay Paz, Dr., 2013. *Beginning ASP.NET MVC 4*. 1st ed. UK: Apress; 1 edition (June 26, 2013).

[6] Dino Esposito, Dr., 2009. *Programming Microsoft ASP .NET 2.0 Core Reference*. 3rd ed. Washinton, US: WP Publishers & Distributors Pvt Limited.

[7] Cristian Darie, et al, 2011. *Build Your Own ASP.NET 4 Website Using C# & VB*. 4th ed. Australia: SitePoint Pty. Ltd. pp. 2-4

[8] Dykstra, T., 2013. *ASP.NET Web Deployment using Visual Studio*. 2nd ed. United States of America: Microsoft Corporation.

[9] Daniel Cazzulino et al, Beginning C# Web Applicatios with Visual Studio .NET. Chicago: Wrox Press, 2002