CIS 3342 Term Project  
Login & Registration

This project will give you experience utilizing all the techniques and concepts learned this semester to create a Web Service (ASMX) that provides cloud-based service to client applications and two client-based applications that use the Web Service. One client will be a Web Application and the other will be a Windows Form Application. This project must be completed with teams of two, but teams must be approved by me before work can begin.

Multi-Tier Architecture

Client-Server systems of the type we have been working on in this course are sometimes referred to as having a Multi-Tier architecture. Software is typically multi-layered where there is a conceptual separation of the software called layers. Software can also be multi-tiered where many different computers are responsible for part of the software’s processing. This type of architecture has 3 or more tiers, where the top tiers are called the Presentation Tier that processes code dealing with the user interface and any associated code that supports it. The last or bottom tier is called the Data Service Tier. It involves the Database Management System (DBMS) and the data it manages. The middle tiers involve what is called the Application or Business Logic that process the code related to the logic of the program. The term project has 4 Tiers:

Tier 1: Presentation Layer (client)

This is implemented as the GUI in the form of HTML and JavaScript processed by the web client.

Tier 2: Presentation Layer (web server/ASP.NET application server)

This is implemented as the GUI in the form of an ASPX WebForm Application, including the ASPX codebehind and other classes that support the GUI.

Tier 3: Business Logic Layer (web server/ASP.NET application server)

This is implemented in the ASPX codebehinds and other application classes that perform the application processing logic, possibly including database processing.

Tier 3: Business Logic Layer (web server/ASP.NET application server)

This is implemented as web services, application classes, and utility classes that are commonly used by Tier 2 programs. It may also include database processing.  
  
Note: the web service and web application both exist on the same server, so there isn’t another tier added to this system. However, in a real-world setting, web services are typically on other web servers, which adds another tier to the system making it a 5-Tier system.

Tier 4: Data Service

The SQL Server DBMS

**Login & Registration Requirements:**

1. **Login**

The opening screen allows an existing user to log-in to the web application, or create a new registration. You application should allow for easier logins by storing and retrieving the user’s LoginID.

If you give your cookie a name, like CIS3342, and set the value to the user’s LoginID, then only one customer will have a cookie at a time. This will produce the following 2 cases when you request all of your application’s cookies:

(a) There are no cookies. In this case, you ask the user to either Login or Register. In the Login and Registration page you will place a checkbox, which asks the user if a cookie should be stored (after a successful login or registration process).

(b) There is exactly 1 cookie. It might or might not be this user, but it’s one of the user’s at your site. A cookie is stored on a computer’s hard-drive, and multiple users may use the computer. In this case you should display the LoginID and request that the user enter the password if this is the correct LoginID. Otherwise, they need to enter the correct LoginID and password.

In the Login page, you will allow the user the option to “Remember me” for quicker logins.  The Registration page will only have the one checkbox, which asks if a cookie should be stored.

Upon successful login or registration, proceed to the application’s main page. However, before doing so you should do 2 things:

(1) Add some code to both the Login and to all of the pages that will prevent someone from simply entering a URL for you a page directly and thus bypass your login. One way to do this is to assign some value to a Session attribute in Login, such as Session(“Login”), and then test Session(“Login”) for nothing (the C# keyword is null) in every subsequent page before its loaded.

(2) Reload the user’s Account and Storage information object from the serialized storage (database) or create an empty account/storage with the User’s Login. This object will contain information about the entire account and storage being used by the user. This object will not contain the actual files, but all the information about the files stored for the user’s account. This object will essentially represent an in RAM data structure of their storage (files stored on the cloud).

**NOTE:** For the Login Project delivery you only have to do (1) above, **not (2)**. You will implement (2) as part of the delivery for Term Project - Part 3.

1. **Registration**

The Registration process will construct a new Account record along with an initial storage capacity. Make the storage capacity small for testing and database purposes. The user must supply a unique LoginID, a password, email address, and any other necessary information. If your email address is the Login, then you require only the email address as the LoginID, but it must still be unique. A checkbox is also displayed, that gives the user the choice of saving the LoginID to a cookie. You will need to use some security measure for storing passwords instead of storing them as plain-text.

1. **Cloud Users & Cloud Admins**

Your application will be used by two different types of users with different sets of capabilities. Cloud users of the application will be users that can store, manage, and retrieve files they stored for their account. Cloud Admins can view user information, update accounts, and view a log of transactions that occurred on the site in the past day or few days. You will not implement their capabilities in this part of the project. You only need to account for two different types of logins and registrations.

1. **Design**

Points will be awarded for various aspects of design which include:

(1) Most effective implementation of the steps to complete the requirements. For example, where the cloud user views and manages their files is implemented using a GridView, or other dynamic data display rather than requiring the user to enter file name in a textbox.

[[(2) Good navigation among pages. Requiring the user to use the Browser Back and Forward buttons is extremely poor design. On each page, the user should have every logical option of moving around the site. The use of User Controls or Master Pages will help organize this navigation.](#SixSteps)](http://cis-iis1.temple.edu/cis342/Lectures/Unit3/TermProject/TermProjectRequirements.htm#SixSteps)   
  
[[(3) Appropriate Client and Server side validation, with precedence being given to server-side since client-side scripting can be disabled.](#SixSteps)](http://cis-iis1.temple.edu/cis342/Lectures/Unit3/TermProject/TermProjectRequirements.htm#SixSteps) You aren’t required to write client-side javascript. However, the validation controls are processed on the client-side.

[[(4) Good placement and sizing of labels and controls. For example, you should not place a control with absolute positioning underneath a dynamically populated list (gridview, dropdownlist, etc.). When the control expands, it may cover the control. Another example is leaving extraneous controls on the screen. This occurs when you do not make proper use of visibility.](#SixSteps)](http://cis-iis1.temple.edu/cis342/Lectures/Unit3/TermProject/TermProjectRequirements.htm#SixSteps)

(5) Clear instructional labels regarding what a person is to do and clear error messages.

(6) Good presentation of the web pages in regards to layout and alignments.

**Extra Credit:**  
It’s possible to obtain extra credit on each of the individual parts of the term project. I will award extra credit for creativity, going above and beyond the scope of the project, and using elements not taught in class. You should see me to find out what qualifies for extra credit. You will not receive extra credit if you give another team work that would allow them to get extra credit.

**Due:**  
See the assignment posting under the Assignments section in BlackBoard.

**Submission:**You need to publish your web application project to the cis-iis2 web server folder TermProject in the root folder of your website, upload the zip file containing the solution with all your code to Blackboard, create a link to the Login/Registration page(s) in your Table of Contents, and provide a URL to your web application’s start page. Also, make sure you have the same version of the solution in your G:\cis3342 folder.

You need to zip the root folder for your solution into a single zip file and submit the assignment in Blackboard. To submit the assignment, you need to click the Assignment’s Title “Term Project - Part 1” to view the submission form and upload the file.

**Make sure you properly submit your assignment and that it works. Programs that don’t run or don’t contain all the necessary files will not be graded. You will need to resubmit the correct files before the late deadline to receive credit.**

Please be sure to save your work periodically as you proceed and also back it up. You may want to store it on your flash drive. If you are going to zip an application in order to store it, BE SURE TO FIRST CLOSE Visual Studio. If you do store information on your flash drive be sure to copy it to a hard drive on your computer before working with the project.