CIS 3342 Term Project – Part 2  
Web Service

All web services must have the same directory path on cis-iis2 (cis-iis2.temple.edu/Spring2017/3342\_nnn/TermProjectWS/).   
  
**CloudService.asmx** with the codebehind class name **CloudService**

**Important: anyone over the Internet can access your Web Service’s Web Methods. Therefore, all of the methods must verify the person’s authenticity through a use of a verification token (a private username/password, or some unique value assigned to all applications that use the Web Service).**

**Web Service Interface Specifications:**

1. **Create a Web Method to add a file for a cloud user that receives an account, file, and verification token (see above or item 8).**This method will be used to store a file for a cloud user. It should return true when the file was added successfully or false when an issue occurs. This method should not allow the user to add files once they reached their storage capacity quota. Note: don’t over think the storage quota as it’s simply a numerical value that represents the total storage in bytes a user is allowed to store.This means the total size in bytes for all files cannot exceed this number. Each file that gets uploaded has file information like its file size. Also, the verification token is used to verify the caller of the Web Method (see item 8).
2. **Create Web Methods to delete and update a file for a cloud user.**These methods will allow the cloud user to manage their cloud storage.
3. **Create Web Methods to add, modify, delete, and get account information.**These methods will be used to retrieve a cloud user’s account information including the storage quota and perform operations on accounts like increasing storage capacity. The cloud user can update their account information and change their password, but not delete their account or increase their storage capacity. Increasing storage capacity, resetting passwords, and deleting accounts are all cloud admin operations. Note: don’t over think the storage quota as it’s simply a numerical value that represents the total storage in bytes a user is allowed to store.This means the total size in bytes for all files cannot exceed this number.
4. **Create a Web Method to get a cloud user’s storage.**This method must return storage and file information. This method will not return the actual files; it should only return information regarding the files including image icons that represent the file type for each file. Hint: think of their storage like a shopping cart where the products in the shopping cart are the files currently stored by the cloud user.
5. **Create a Web Method to get recent transactions for a single user and a time period to view the history.**This method will be used by cloud admins to view a log history of operations that occurred for a particular cloud user. This method should allow the admin to get all transactions for a given user or transactions that occurred over a certain period of time like the past day or two.
6. **Create the remaining Web Methods for performing all other operations that deal with the database.**
7. **All database operations should be implemented using Stored Procedures for security purposes.**
8. **All Web Methods must verify the caller of the Web Method through a use of a verification token (a private username/password or some unique value assigned to all applications that use the Web Service).**This protection is necessary because anyone can call your Web Methods over the Internet. Therefore, we need to build in some simple protection that the Web Method will check before executing the code inside the method. When the method is invoked without the correct verification token, then the Web Method should not execute its code and immediately exit the method.
9. **Create a Web Service Test Page**

Create a simple ASPX page that demonstrates your Web Service’s Web Methods work properly. The page must be created to allow the grader to input some of the required values to ensure the methods work. You don’t need a working application to test the Web Methods.

1. **Use Component-Based Software Design**
2. **Design and Implement a Good Data Model.**

**Due:**  
See project posting under the Assignments section of Blackboard.

**Submission:**You need to publish the Web Application project containing your Web Service and the Web Service Test Page to the cis-iis2 Web server folder TermProjectWS, upload zip file containing the solution with all your code to Blackboard, and provide the necessary URLs for your HomePage located in the root of your Web with links to all projects, and the URL to your Web Service Test Page (simple ASPX page used to test your Web Methods).   
  
You need to zip the root folder for your Visual Studio 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 2” to view the submission form, upload the zip file containing the Visual Studio Solution, and necessary URLs.

**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 and marked late.**

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.