Worksheet 8 – Authentication and Authorization System

Topics:

- Web Services
- Database
- Login/Password authentication system
- Methods of storage of passwords: hash and salt
- Authorization and access to resources
- Authentication using a digital certificate as an alternative method

©2020: { rui.ferreira,marisa.maximiano,ricardo.p.gomes,nuno.reis }@ipleiria.pt

1. Objective and Projects

The purpose of this worksheet is to implement a small system with a user authentication service. For this, two applications will be used: "AuthService" and "WindowsClient".

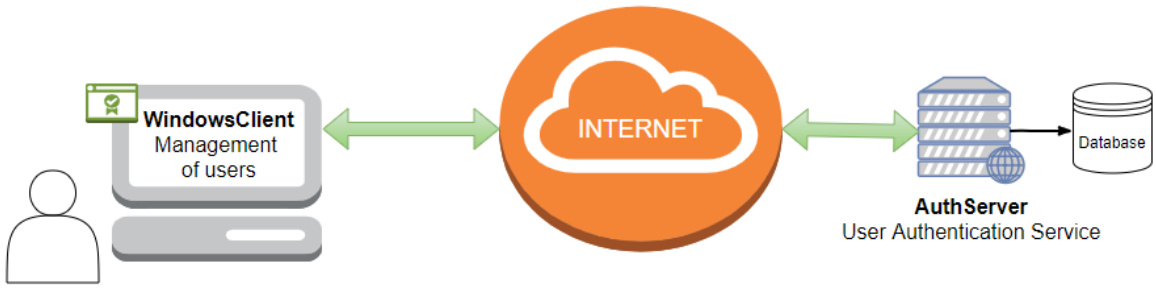

Figure 1 – Applications to implement in this worksheet

The exercises proposed below are intended to show how Web Services ("simple") can be used to provide information and create clients that "consume" these services. To start the implementation let's know what code provided:

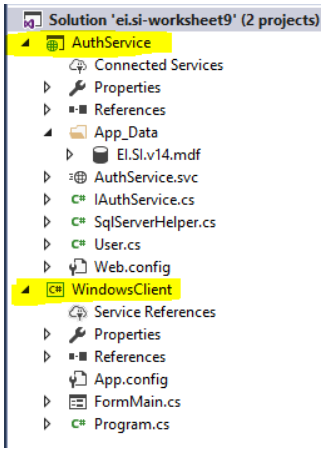

Figure 2 - *Solution* and projects to be used

1.1. AuthService

The projet “AuthService” has the following structure:

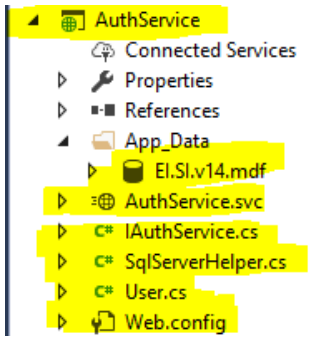

Figure 3 – AuthService project

Note the following files:

- EI.SI.v14.mdf – database to use in user management system;
- IAuthService.cs e AuthService.svc.cs – interface and implementation of the service;
- SqlServerHelper.cs – code to access to the database (table: “Users”);
- User.cs – class that maps a row of the "Users" table.

WindowsClient

The projet “WindowsClient” has the following structure:

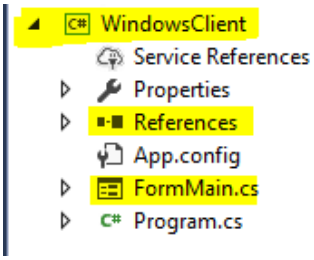

Figure 4 - WindowsClient project

Note the following files:

- Service References – folder to store references to "AuthService" Webservice;
- FormMain.cs – Main form:

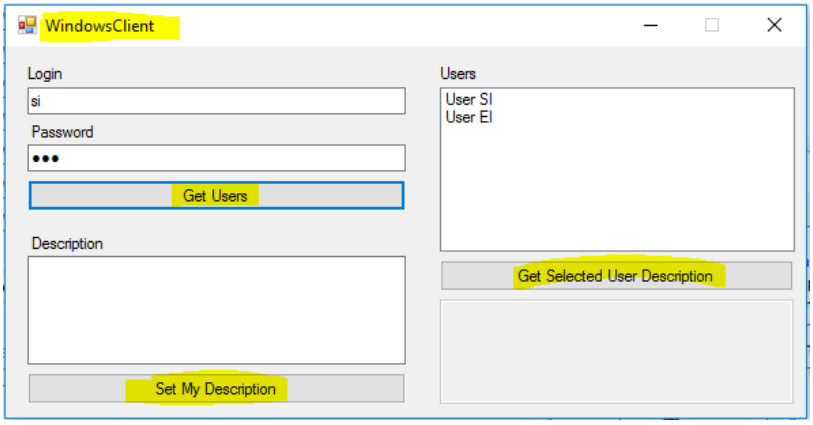

Figure 5 – WindowsClient "FormMain"

On the left side buttons to call to services that will require authentication and on the right side the authentication is optional.

The actions to implement are **all** using the Web service:

- The "Get Users" button retrieves the list of users and places it in the "Users" list (at right);
- The "Set My Description" button retrieves the text from the textbox above the button and updates the user information in the database;
- The "Get Selected User Description" button gets the user information selected in the user list (above the button) and places it in the textbox below the button.

2. Authentication Service

The authentication concept can be defined as "checking the credentials of the connection attempt". If these credentials, which can be of several types (eg: login/password), are valid in the system then we have a valid authentication. Following are some exercises where this and other concepts are present.

1. Analyze the existing code in each of the available projects, especially in the previously presented files.
2. In the "AuthService" project" double-click the database file to verify that it is "loaded" and can be accessed through Server Explorer (View -> Server Explorer).
3. Run the Solution and, using the "WCF Test Client", test the "VerifyAcessToBD" method by verifying the return value, thus also confirming access to the database.

   Note: check in the "Web.Config" file, within the <configuration> section, if the <connectionStrings> section has the appropriate "ConnectionString" and that it is being used correctly in the code.

4. In the "WindowsClient" project:

   a) Add the reference "ServiceReference_AuthService" to the previous service:

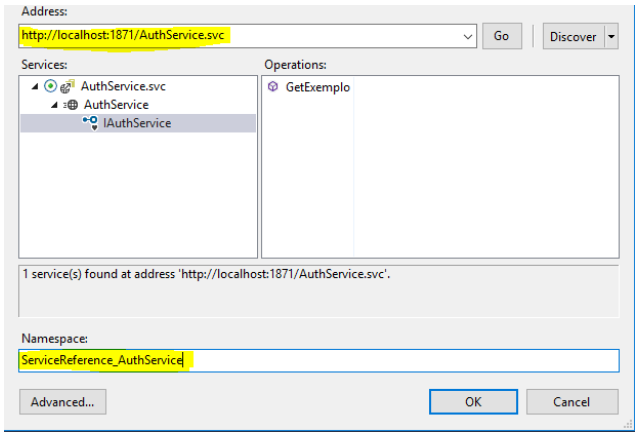

Figure 6 – Reference to the Webservice

   b) To test the Webservice first, use the "Get Selected User Description" button to invoke the "VerifyAccessToBD" method of the service and place the value returned in the "txtDescription" textbox.

   Note: If you want to start only the "WindowsClient" application, place the project as "Startup Project".

5. In the "AuthService" project:

   a) Using the appropriate files, implement the services that allow you to fulfill the requirements of "WindowsClient", that are:

```
string GetUserDescription(string login);
string SetUserDescription(string login, string password, string description);
User[] GetUsers(string login, string password);
```

   Please note the following notes:

      i. The method names and their input and output parameters are just suggestions for use in the lesson, so any others can be used;

      ii. To access the database, use the code provided and already included in the project;

      iii. Change only the "cmd.CommandText = sql" line of the "UserExists" method of the "SqlServerHelper" class, so that authentication is done with login and password (these parameters are already defined in the method).

   b) Using the next information, choose the appropriate settings for the service:

```
/// see this site to understand the InstanceContextMode:
http://www.codeproject.com/KB/WCF/WCFInstance.aspx

/// use this approach instead the Application and Session Variables and choose:
[ServiceBehavior(InstanceContextMode = InstanceContextMode.PerCall)]

[ServiceBehavior(InstanceContextMode = InstanceContextMode.PerSession)]

[ServiceBehavior(InstanceContextMode = InstanceContextMode.Single)]
```

c) Test the previous services and, using the interface "WCF Test Client", check that the connection to the database is being made properly.

6. In the "WindowsClient" project:

   a) Update the Webservice settings to make the services available;

   b) Implement and test all the features of FormMain, as explained in Chapter 2.

7. In the "AuthService" project and in regarding the way the passwords are used/stored:

   a) Make it the result of the SHA256 hash algorithm to be stored in the database and not the clean text password.

   Note: use the Convert.ToBase64String (byte []) method.

8. In the "WindowsClient" project:

   a) Test the application and verify that everything is working properly. If not, change what is necessary to regain access to the service.

9. **[extra]** In the "AuthService" project:

   a) Implement a method that best protects the passwords that are stored in the database against dictionary attacks.

   Note: search for the technique called "salt".

10. **[extra]** In the "WindowsClient" project:

    a) Re-test the application and make sure everything is working correctly. If not, change what is necessary to regain access to the service.

3. Authorization service

Another important concept is authorization, that is, "verification of permission to attempt to connect." This notion sometimes goes missing, but "authorization occurs after successful authentication". So, in the coming exercises you will be asked to implement a small scenario where this concept is present. To authorization concept make sense we will divide the users into three groups (roles): "Admins", "Users", "Guests", where each of these groups has different permissions on the information that is returned from the database.

1. In the database:
    a) Add to the table "Users" plus 2 users with the following logins: "Admin" and "Guest";
2. In the "AuthService" project:
    a) Make the changes to the project and the database to the service that returns the list of users is only available to users belonging to the group "Admins";
    b) **[extra]** Make the update of the "Description" field of users only available to users who belong to the "Users" group and also to the "Admins" group.
3. In the "WindowsClient" project:
    a) Test the applications and make sure everything is working properly. Otherwise change what is necessary so that the service only works for users who are authorized.

4. Authentication using a Digital Certificate

1. In the database, to allow for digital certificate authentication, to the following:

    a) Associate manually in the database the identifier for the certificate of each user.

    Note: use the "thumbprint" property for the association (look out for capitalization and whitespaces);

2. On the “AuthService” project:

    a) Implement a new method (“User[] GetUsersByCertificate(…)”) to allow for digital certificate authentication:

        i. Update the “SqlServerHelper” class to reflect the new database structure;

        ii. Don’t forget to validate the certificates every time they are used.

3. On the “WindowsClient” project:

    a) Implement the button “Choose Certificate (CC/Store)” to allow for the user to choose the certificate to use:

        i. Use just the digital certificates with the purpose of authentication and selected from the Store

        ii. Use another personal certificate if you can’t use the Citizens Card;

        iii. Save the digital certificate to be used in future actions.

    b) Implement the button “Get Users (Certificate)” to use the new authentication method of the “AuthService”.

    Note: don’t forget to update the service reference