# Purpose

Presentation tier (a web client application front-end),

Logic tier (a WCF web service that will provide an interface to the web client application for calling the web service’s methods)

Data tier (a SQL Server database) will respond to web service’s requests for data.

The web site will be hosted using an IIS web server.

A technical report will document the architecture and functionality of this distributed system and you will be required to demonstrate your application in class.

# Brief

Task:

Designing and implementing a web service for the WattsALoan finance company.

The application will be used by travelling financial advisors who visit the homes of potential clients who are interested in securing a personal loan.

The advisors will use laptops to connect to a 4G broadband network that will provide connectivity to the web application.

The application will allow financial advisors to create and modify client data, allocate loans and record client payments.

Managers will be able to create and modify new employees and create new types of loan.

The script to create the SQL Server database, used as a data source for this application, can be downloaded from Moodle.

Demonstrate your application either before or during the first week of the semester break.

# Tasks

Download the script (WattsALoan.SQL) from Moodle that will be used to create the SQL Server database and its stored procedures.

All of the required tables and the stored procedures that the web service is required to invoke will be created by running this script.

The only change that may be made to the existing database is to add and modify its records i.e. do not make any structural changes to the tables or modifications to the database’s stored procedures.

You may re-run this script to re-create the database and its stored procedures any time you need a ‘clean’ database to work on during development and testing.

The web service will use this database as its primary source of data when responding to requests from the web application front-end.

The web service will connect to the database when required using a connection string.

You may assume that Windows authentication will be used to connect to the WattsALoan database.

Create a new folder C:\inetpub\WattsALoan on your machine that will be used as the physical location of the web site you will create in step 4. Copy the files iisstart.htm and iis-85.png from C:\inetpub\wwwroot to the new folder. These files will display the default web page in step 4.

Within IIS, stop all web sites (important). Now create a new website called WattsALoan and ensure that it is associated with an application pool of the same name, that it uses port 80 and that the physical address is the same as the new folder created in step 3.

Start the web site and browse to it from within IIS. The default web page should be displayed if you have configured your web site correctly.

In Visual Studio, create a WCF web service called WattsALoanService that will service requests from a web client front-end that you will create in step 6.

Set the location for the service to the folder you created in step 3.

The web service should present an interface to the web client that will allow the stored procedures in the WattsALoan database to be invoked on demand.

The parameters required by the each stored procedure will be passed to the web service from the web client front-end.

In Visual Studio, create a new empty ASP.NET web application called WattsALoanClient that will be used as the web client front-end for the web service created in step 5.

The web client front-end will allow two groups of users, managers and advisors, to perform a variety of administrative tasks.

Use the Site Map template and TreeView control to allow managers and advisors to select and perform their role-orientated tasks on web pages, as described below.

These controls should appropriately group together the tasks performed by managers and advisors.

Managers need to be able to add and modify employee records and to create new types of loan.

Advisors need to create new customer records, update existing customer details and create loan allocations. They also need to record payments against loans.

This functionality will be provided by remotely invoking the appropriate stored procedure contained within the WattsALoan database. For example, clicking on a button called ‘Add New Employee’ on a web form invokes the stored procedure to insert a new employee into the WattsALoan database.

The web service that you will create (see step 5) needs to access these tables and stored procedures using a SQLClient connection string to the WattsALoan database.

Database security is to be implemented on the two types of users who will access this system i.e. managers and advisors (these users should be created using the Computer Management utility).

Each type of user should then be assigned a SQL Server login (create two logins - one for each type of user).

Both managers and advisors should only be mapped to the WattsALoan database with a db\_datareader role.

Managers should in addition be given permission to execute only those stored procedures needed to perform the tasks listed step 7.

Similarly, advisors should be given permission to execute only the stored procedures associated with their role.

Your application should include a suitable logo and you should select appropriate CSS styles and apply them to the controls on your web pages.

The web application should incorporate validation controls with suitable error messages.

Required field and regular expression validators should be used where appropriate to achieve this task.

The website should convey a corporate look and feel that should be consistent from one web page to the next.

In your report (see Assignment Deliverables section overleaf for details), create a heading called Application Architecture that documents the tiered architecture of your application and the form of communication used between each tier.

This should be a detailed, technical description that draws upon the theory presented in the lectures.

This section of the report should also include a diagram that illustrates each tier and the communication protocols employed.

# Assignment Deliverables

A class demonstration of your application that will be assessed using the ‘In Class Demo Marking Scheme’ in the Assignment One section of my Moodle site.

A word-processed System Implementation Report that contains:

An overview (executive summary)

Application architecture (described in Task 10)

A description of the application’s functionality

The code required to achieve this functionality

Screen dumps of each web form in your application