# INFS3204/7204 Practical 3 – WCF Web Services

The goal of this practical is to explore WCF web services. All practicals will have to be developed with Microsoft Visual Studio 2010/2012 using **C#** as the programming language. No other languages will be accepted. This practical contributes to 10% of your overall grade. **You must present this practical to your lab tutor during your scheduled lab session in week 9 (16/09/2013 to 20/09/2013). No early or late demonstration is allowed.**

This practical is divided into 4 tasks:

- Creating a WCF Web Service: Australian Postcode Validation (**0.5 Mark**)
- Creating a WCF Web Service: Managing people and their job information (**6.5 Marks**)
- Defining two custom types (user-defined class) in the above WCF web service (**1.5 Mark**)
- Creating an ASP.NET Web Application that invokes the above WCF web services (**1.5 Marks**)

## Preparation

Before attempting to do this practical, you should have a basic knowledge of how to use Microsoft Visual Studio 2010/2012 to create ASP.NET Web Applications and WCF Web Services, as well as a basic knowledge of C# programming. Please note that you need to choose .NET Framework 4 or later for your practicals. You also need to have Windows 7 installed on your computer.

**Notes:** Please be advised that you need to meet the basic requirements for this practical, which are mentioned in this document. However, you may decide to have more attractive and comprehensive design and implementation. This increases your chance to be selected as one of the students who will get the opportunity for a demonstration in week 12.

## Task 1: Create an 'Australian Postcode Validation' WCF Web Service (0.5 Mark)

Create a **WCF** web service that is designed for validation of Australian postcodes. This web service should have one operation called **PostcodeValidation()**, which receives two values (**int** postcode and **string** state) and returns a **Boolean** value (true if postcode matches with the state, and false otherwise).

The Australian states and their corresponding postcodes are as follows:

| State | Postcode |
|---|---|
| **NSW** | 2000 – 2599, 2619 – 2898, 2921 – 2999 |
| **ACT** | 2600 – 2618, 2900 – 2920 |
| **VIC** | 3000 – 3999 |
| **QLD** | 4000 – 4999 |
| **SA** | 5000 – 5799 |
| **WA** | 6000 – 6797 |
| **TAS** | 7000 – 7799 |
| **NT** | 0800 - 0899 |

## Task 2 & 3: Create a 'People and Job Information Management' WCF Web Service (8 Marks)

Create a new **WCF** web service and define the following **two** custom types (user-defined classes) inside it (**Please note that each class should have the specified fields with the specified data types**):

1. **Person** (**1 Mark**)
    a. **String** firstName
    b. **String** lastName
    c. **DateTime** dateOfBirth
    d. **String** email
    e. **String** streetAddress (i.e. unit number/street number and street name)
    f. **String** suburb
    g. **String** state
    h. **Int** postcode
    i. **Job** job (**Please note** that this field should be of type **Job**)

2. **Job** (**0.5 Mark**)
    a. **Int** positionNumber (assume that it is **unique** for each job)
    b. **String** positionTitle
    c. **String** positionDescription
    d. **String** companyName

This WCF web service should have the following **three** operations (**Please note that each operation should only receive the specified parameters with the specified data types**):

1. **SaveInfo()** (**1.5 Marks**):

   This operation receives person's information (i.e. **String** firstName, **String** lastName, **DateTime** dateOfBirth, **String** email, **String** streetAddress, **String** suburb, **String** state, **Int** postcode, **Job** job) and stores it in **two separate** text files as explained below:

   - **'Person.txt'**: In this file, you should keep firstName, lastName, dateOfBirth, email, streetAddress, suburb, state, postcode and job.**positionNumber**
   - **'Job.txt'**: In this file, you should keep job.positionNumber, job.positionTitle, job.positionDescription and job.companyName

   For each file, you may have your own text structure for separating records and fields from each other. In order to save and retrieve information from files, you should make use of FileStream, StreamWriter and StreamReader objects, or others from System.IO namespace.

   This operation returns a **Boolean** value, which indicates whether the information has been saved successfully or not.

2. **GetJobInfo()** (**2 Marks**):

   This operation receives **String** firstName and **String** lastName and performs a search through 'Person.txt' and 'Job.txt' files in order to retrieve job information of the person. If a person with the specified firstName and lastName exists in the 'Person.txt', the operation returns an object of type **Job**, which has been created by the person's job information from the 'Job.txt' file. Otherwise, the operation returns **null**.

3. **GetColleagues()** (**3 Marks**):

   This operation receives **String** firstName and **String** lastName and calls the **GetJobInfo()** operation in order to retrieve job information of the person. It will then performs a search through the 'Person.txt' and

'Job.txt' files and returns a **Person[]/<Person>** (array or list) of all his/her colleagues (i.e. people who work for the **same company**). If no one with the specified firstName and lastName exists, the operation returns **null**.

## Task 4: Create an ASP.NET Web Application (1.5 Marks)

Create an ASP.NET web application and **invoke the two WCF web services (0.5 Mark)** that you have created in previous tasks. Your web application should have the following pages (**0.5 Mark** for creating the required web pages. You should also put some links/buttons/tabs/… for **navigating** through different pages):

1. **Save Information** page:

   In this page, you should put input fields of your own choice for all the specified fields in Task 2 (**8** fields for **Person** and **4** fields for the **Job**), in order to get a person's and his/her job information. You are also required to have a **Save button** on this page so that when user clicks on it, the web application calls the **SaveInfo()** operation of the WCF web service in order to save the information. If the operation returns true, '**Successfully saved**' message should be displayed. Otherwise, '**Unable to save**' message will be displayed.

   You should also do the following **validation checks (0.5 Mark)** on the input fields of this page and display an appropriate error message for each of them, when necessary:

   - **Required field validation**: firstName, lastName, positionNumber and companyName are required fields.
   - **Email Validation**: should be entered in a valid format.
   - **Position Number**: should be an integer.
   - **Postcode Validation**: invoke the **Australian Postcode Validation WCF web service** to check if the entered postcode and state match with each other.

2. **Search Person's Job** page:

   You are required to have the following elements on this page:

   - A **Textbox**: for first name input
   - A **Textbox**: for last name input
   - A Job Search **Button**: When user clicks on this button, the web application calls the **GetJobInfo()** operation of the WCF web service to retrieve job information of the person.
   - A **Label/Textbox**: if the **GetJobInfo()** operation returns null, a '**Not found**' message should be displayed. Otherwise, the person's job information will be displayed here.

3. **Search Colleagues** page:

   You are required to have the following elements on this page:

   - A **Textbox**: for first name input
   - A **Textbox**: for last name input
   - A Colleague Search **Button**: When user clicks on this button, the web application calls the **GetColleagues()** operation of the WCF web service to retrieve **all** the person's colleagues' information (**all** the 9 fields for type **Person**).
   - A **Multiline Textbox**: if the **GetColleagues()** operation returns null, a '**Not found**' message should be displayed. Otherwise, all the person's colleagues' information will be displayed here.