

School of Information Technology and Electrical Engineering

INFS3204/7204 – Service Oriented Architecture

Assignment Four (10 Marks)

Due at 23:59 Sunday 22/10/2017

The goal of this assignment is to explore the REST API development. All assignments will have to be developed with **Microsoft Visual Studio** using **C#** as the programming language. No other languages will be accepted. This assignment contributes to 10% of your overall grade. **You must demonstrate this assignment to your lab tutors during your enrolled lab session in week 12. No Demo, No Mark!** It is your own interest to ensure that your codes have been submitted to the BlackBoard before **23:59 Sunday 22/10/2017**. You have only three attempts to submit your codes. **Penalty will be applied for late submission. All submissions need to be in the form of zip/rar file format.**

Preparation

- Before attempting to do this assignment, you should have the basic knowledge of how to use Microsoft Visual Studio 2015 or 2017 to create Web APIs and consume them with Ajax(jQuery), as well as the basic knowledge of programming with C#. Please note that you need to choose .NET Framework 4.6 for your assignments.
- Notes: Please be advised that you need to meet the basic requirements for this assignment, which are mentioned in this document. However, you may try to design and implement your work more interestingly and comprehensively.

Please Note: **The GUI of Prac Four is objective and no marks will be allocated to the layout design of this assignment. But you are responsible for all exception handling in both service and client sides in all required functions. During the demonstration, if your application crashes because of unhandled exception (either service or client side), you will lose marks for that function. For input validation, only the data types listed are accepted, no other types are allowed.**

Task: NBA League Web API

You need to create the player class in your project as follows:

Player

- String **Registration_ID**
- String **Player_name**
- String **Team_name**
- DateTime **Date_of_birth**

You need to provide all the following five Web APIs:

- **GetAllPlayers. (2 Marks)**

Use **RESTful approach** to implement this operation on server side that supports Http **GET** method (1 mark), and use **Ajax** to call this operation from a client side web page (1 mark). This operation reads information about all players from players.txt (Download from Blackboard) and returns a list of players.

All Players

- p1111, Kobe Bryant, Los Angeles Lakers, 1978-08-23
- p2222, Tim Duncan, San Antonio Spurs, 1976-04-25
- p3333, LeBron James, Cleveland Cavaliers, 1984-12-30

- **GetPlayerInfo. (2 Marks)**

Use **RESTful approach** to implement this operation on server side that supports Http **GET** method (1 mark), and use **Ajax** to call this operation on client side (1 mark).

This operation receives a field and a corresponding value of such field, and returns a list of players that satisfy this condition. For example, when the field is ID and the value is p2222, it displays all the players that match the condition (in this case, the information about Tim Duncan).

Note that: **The supported fields should include PlayerID and PlayerFullName (Using Dropdownlist in a web form). Input validation is required, and the input cannot be null. For Name, you need to support both upper and lower cases, and you also need to support partial match.**

Search

Name ▼

- p2222, Tim Duncan, San Antonio Spurs, 1976-04-25

- **DeletePlayer. (2 Marks)**

Use **RESTful approach** to implement this operation on server side that supports Http **DELETE** method (1 mark), and use **Ajax** to call this operation on client side (1 mark).

This operation receives a field and a corresponding value of such field. For example, the field is ID, and the value is p2222. Then, it deletes all the players that match the condition, and save the rest of the players into players.txt. The supported fields should include PlayerID and PlayerName. If the operation fails due to deletion of an inexistent player, it should report a NotFound error.

Note that: **Once you delete a player, the rest players need to be updated on the web page.**

All Players

- p1111, Kobe Bryant, Los Angeles Lakers, 1978-08-23
- p3333, LeBron James, Cleveland Cavaliers, 1984-12-30

Delete

ID ▼

- **PlayerRegistration. (2 Mark)**

Use **RESTful approach** to implement this operation on server side that supports Http **POST** method (1 mark), and use **Ajax** to call this operation on client side (1 mark).

This operation receives the information of a player and creates an instance of class Player, then add this player into existing player list and save into file players.txt. If the ID of the player you want to register already exists, then you need to prompt “This player has already existed in our system!” .

Note that: **The updated player list after this operation needs to be displayed on top of the page. Input validation is needed.**

All Players

- p1111, Kobe Bryant, Los Angeles Lakers, 1978-08-23
- p2222, Tim Duncan, San Antonio Spurs, 1976-04-25
- p3333, LeBron James, Cleveland Cavaliers, 1984-12-30
- p4444, Stephen Curry, Golden State Warriors, 1988-04-16

Player Registration

Registration ID:	<input type="text" value="p4444"/>
First Name:	<input type="text" value="Stephen"/>
Last Name:	<input type="text" value="Curry"/>
Team Name:	<input type="text" value="Golden State Warriors"/>
Date of Birth:	<input type="text" value="1988/04/16"/>
<input type="button" value="Player Registration"/>	

- **PlayerUpdate. (2 Mark)**

Use **RESTful approach** to implement this operation on server side that supports Http **PUT** method (1 mark), and use **Ajax** to call this operation on client side (1 mark).

This operation receives the updated information of a player and save the updated information into file players.txt. If the ID of the player you want to update does not exist, then you need to prompt “This player does not exist in our system!” .

Note that: **The updated player list after this operation needs to be displayed on top of the page. Input validation is needed.**

All Players

- p1111, Kobe Bryant, Los Angeles Lakers, 1978-08-23
- p2222, Tim Duncan, San Antonio Spurs, 1976-04-25
- p3333, LeBron James, Cleveland Cavaliers, 1984-12-30
- p4444, Stephen Curry, Golden State Warriors, 1988-04-16

Player Update

Registration ID:	<input type="text" value="p4444"/>
First Name:	<input type="text" value="Stephen"/>
Last Name:	<input type="text" value="Curry"/>
Team Name:	<input type="text" value="Golden State Warriors"/>
Date of Birth:	<input type="text" value="1988/04/16"/>
<input type="button" value="Player Update"/>	

--- End of Document ---