CSE 681 – Software Modeling and Analysis
Project 2
Web Communication

Learning Objective

Be able to create and document API communication
Better understand web communication formats and presentation

Description

The following *REST get* request will return the json data saved in *team49ers_season2020.json*.

```
https://sports.snoozle.net/search/nfl/searchHandler?
fileType=inline&statType=teamStats&season=2020&teamName=26
```

The stats are in an array (see example results *team49ers_season2020.json*) made up of two Results classes (see below) *visStats* and *homeStats*,

```
Results
statidcode          | character varying(30) |          | extended |
gamecode            | character varying(30) |          | extended |
teamcode            | integer               |          | plain    |
gamedate            | date                  |          | plain    |
rushyds             | integer               |          | plain    |
rushatt             | integer               |          | plain    |
passyds             | integer               |          | plain    |
passatt             | integer               |          | plain    |
passcomp            | integer               |          | plain    |
penalties           | integer               |          | plain    |
penaltyyds          | integer               |          | plain    |
fumbleslost         | integer               |          | plain    |
interceptionsthrown | integer               |          | plain    |
firstdowns          | integer               |          | plain    |
thriddownatt        | integer               |          | plain    |
thirddownconver     | integer               |          | plain    |
fourthdownatt       | integer               |          | plain    |
fourthdownconver    | integer               |          | plain    |
timeposs            | integer               |          | plain    |
score               | integer               |          | plain    |
```

For this simulation, pull all the team's data for the 2020 season by modifying the *get* request value *teamName* (numbers 1-32). Printout all the teams' names, associated team number, and their season records using the score attribute from each game. For extra credit, figure out why the team numbers are ordered the way they are (hint: they are not random).

Functional Requirements

1. Shall communicate with other servers.
2. Shall support printing a JSON object in a user-friendly manner to the console or a file with printing of team name, team number and season record.
3. Shall store JSON response data object.

Non-Functional Requirements

1. Shall be implemented in C# or another OOP language
2. Your system shall be robust and seamlessly handle any unexpected inputs.
3. Your system design should be flexible and clean utilizing OOD principles.
4. Your system should be maintainable by providing adequately detailed module operations and maintenance history as well as function prologue.
5. Your system should be designed with reusability in mind by utilizing modular and cohesive units..

Submission

Your submission should include:

1. Show the model of your classes using a design paradigm (UML, flow chart, etc.) using a design paradigm of your choosing.
2. Show the model of your system including the software and API server using a design paradigm of your choosing.
3. Zip up and submit the structure of your program in Project 2