# Project 2 (Web-based LINQ DB)

Problem Statement

Niantic Corporation wants to keep track of all information regarding the people playing their newest Pokémon game. You will make a Windows Forms application to accomplish this. The data required will be accessible from a URL that will return the contents of a database in XML format. Your application will retrieve this data, deserialize it, and store it in C# collections. Your application will provide an interface which allows a user to perform LINQ queries on the data.

The DataBase Layout

Player information is stored in a table called Player. Each player has a unique identification number, a first name, a nickname and city location. This game is pay-to-play, so a player's record stores whether that player is currently paid up or not. A second table (Pokemon) stores the information for each type of card in the game. Each Pokémon character has a name, a unique identification number along with its attack and defense values. A third table (Ownership) ties the first two tables together. It maintains records of which Pokemon types that player owns. (A player will likely own more than one of a type).. While Player and Pokémon listings are unique, one player may own multiple Pokémon, as the following schema indicates.
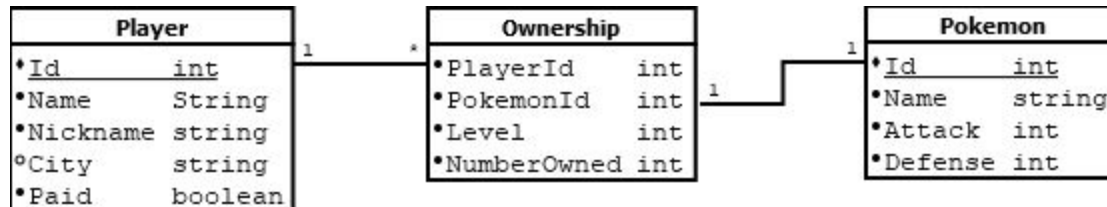
Each Player has zero or more Ownership records. Each Ownership record maps to one Pokemon record. Underlined fields indicate primary keys.

Requirements:

The application must have the following capabilities.

1. Clearly display the complete contents of each of the Player, Ownership, and Pokemon collections.
2. Provide a ComboBox where the user can select a city name. Use that name to query and display all player nicknames from that city. Break up the display into users who are paid up, and those who are not (LINQ orderby).
3. Provide a TextBox where the user can input an attack value. Use that value to query and display all Pokémon characters who have attack values equal or greater than it. Sort the display in decreasing attack value order.
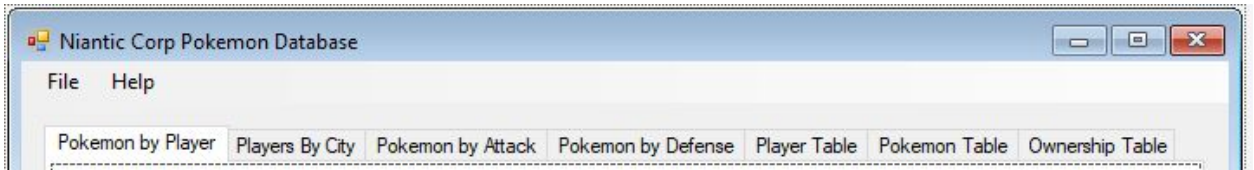
4. Do the same as Part 3, but with defense values instead.

Other requirements
- Must be a Windows Forms application
- Must download and deserialize data using a WebRequest (on startup)
- Must have a menu strip with File,Exit and Help,About. The about box must show your name.
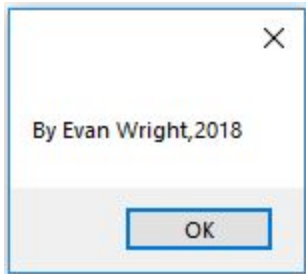
- Must have Tab Control with a tab for each of the user functions listed above (see below)

- Must use LINQ to perform queries on the data

Loading the Data

The data is accessible from the following URL.

http://mrwrightteacher.net/NianticCorp/PokemonDB.php

Open this URL in a browser, then view the page source. You will see the XML formatting.

Your application will download the data via a **WebRequest**, then deserialize it into a C# class so LINQ queries can be performed on it. A C# class into which the XML data can be deserialized is provided for you. This file is named XmlMappings.cs is located in the project resources. **Add this project to your file, then change the namespace in the file to match that of your project.**

The root object in the mappings is the PokemonDB object. In your class add one of these objects to deserialze the XML into.
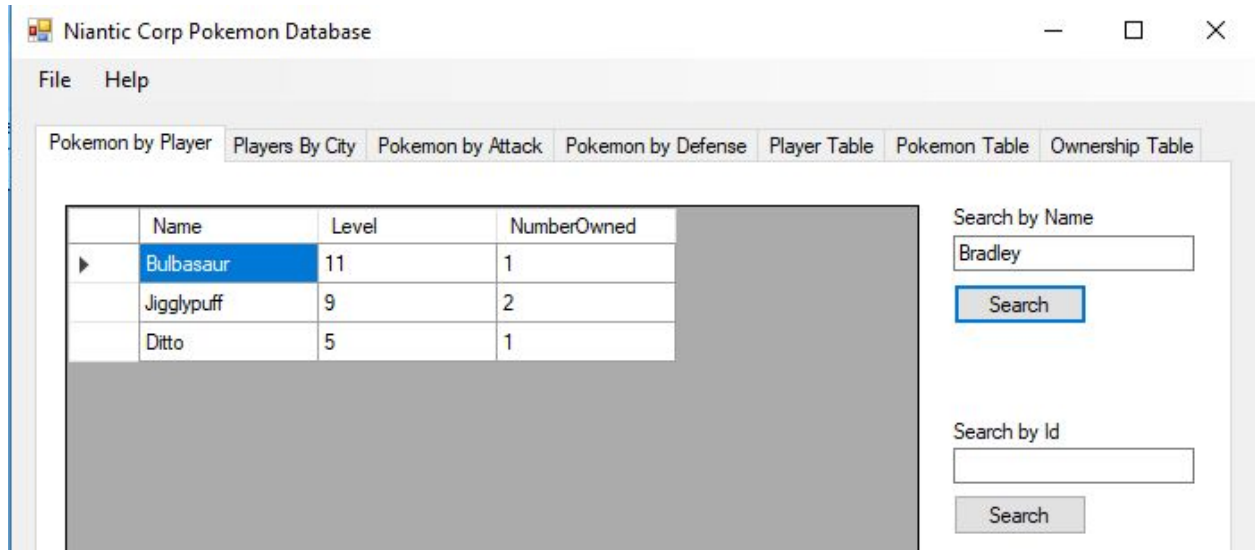
```
PokemonDB pokemonDB;
```

Note: This is exactly like the MovieDB project we did in week 13. Refer to that project as an example.
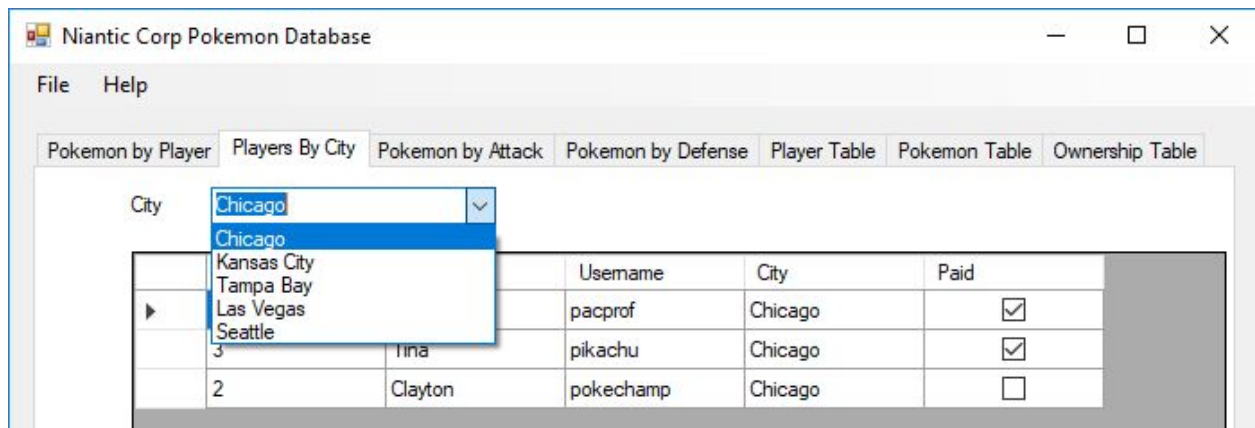
Deliverables: Zip your solution folder and submit it via the link in black board

Examples

Example of the search by Name / Id tab

Example of the Players by City tab

Example of filtering by attack/defense

An example of the tables (Player/Pokemon/Ownership)