# Spring Data Exam – 1 August 2021

Player Finder

In recent months, the world has been once again conquered by the football magic. Unfortunately, our national team was again not among the best teams, which clearly tells us that we need to find better new players. A small team from SoftUni starts working on the staff selection and invites you to be involved in the program development. The idea of it is to receive a huge amount of information about the players in different formats - JSON or XML. Retrieve the information and record only those that meet certain criteria in the database. Also, when requesting certain data, the program should display the necessary information.

1. Functionality Overview

The application should be able to easily **import** hard-formatted data and **support functionality** for also **exporting** the imported data. The application is called – **Player Finder**.

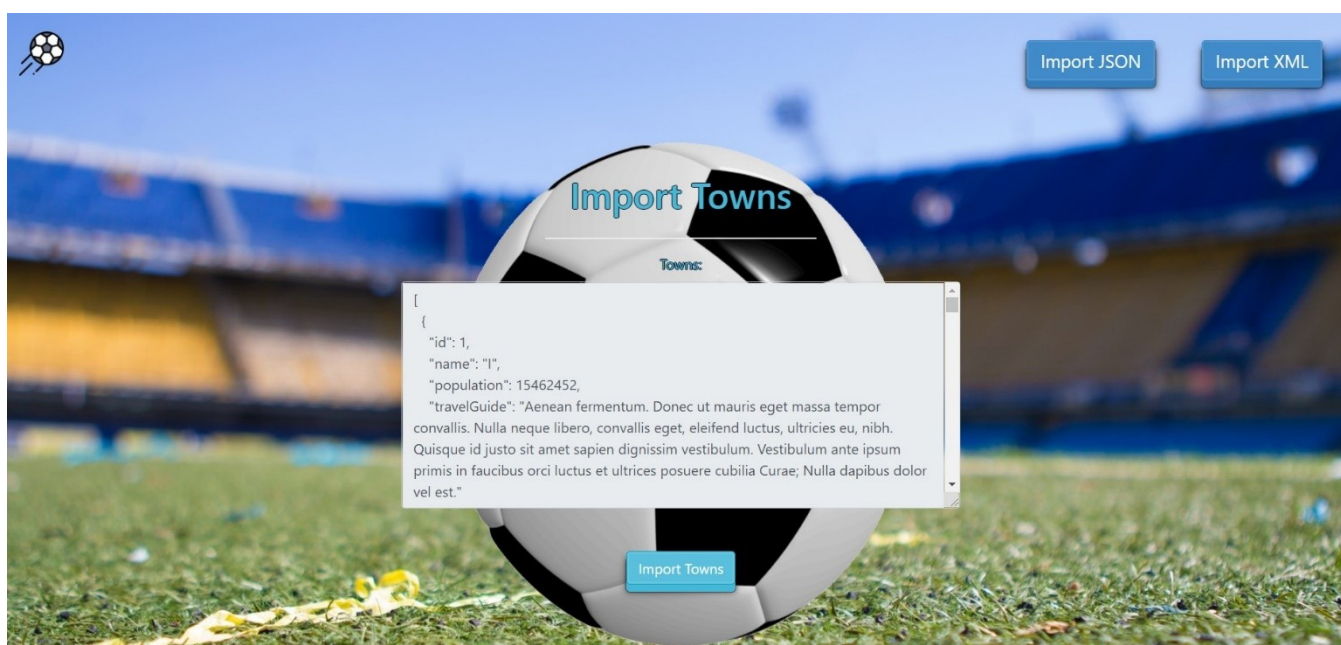Look at the pictures below to see what must happen:

- The home page before importing anything:

Follow us:
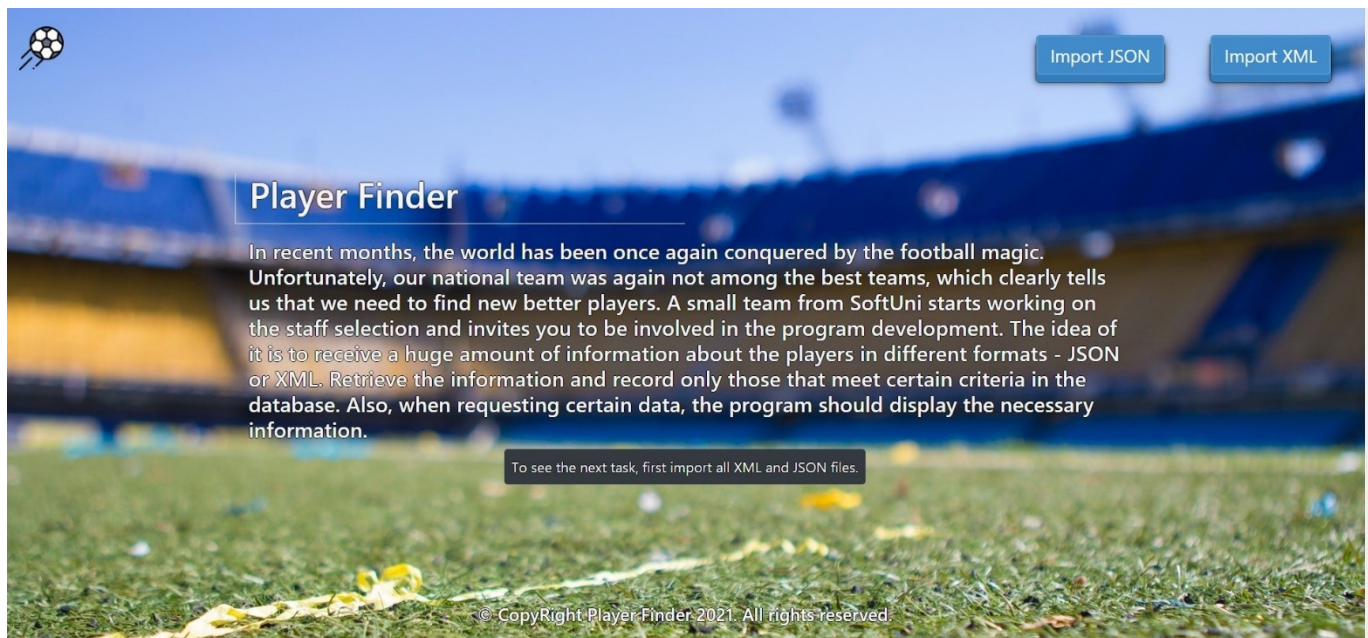
- The import JSON page before importing anything:

- Import the towns first:
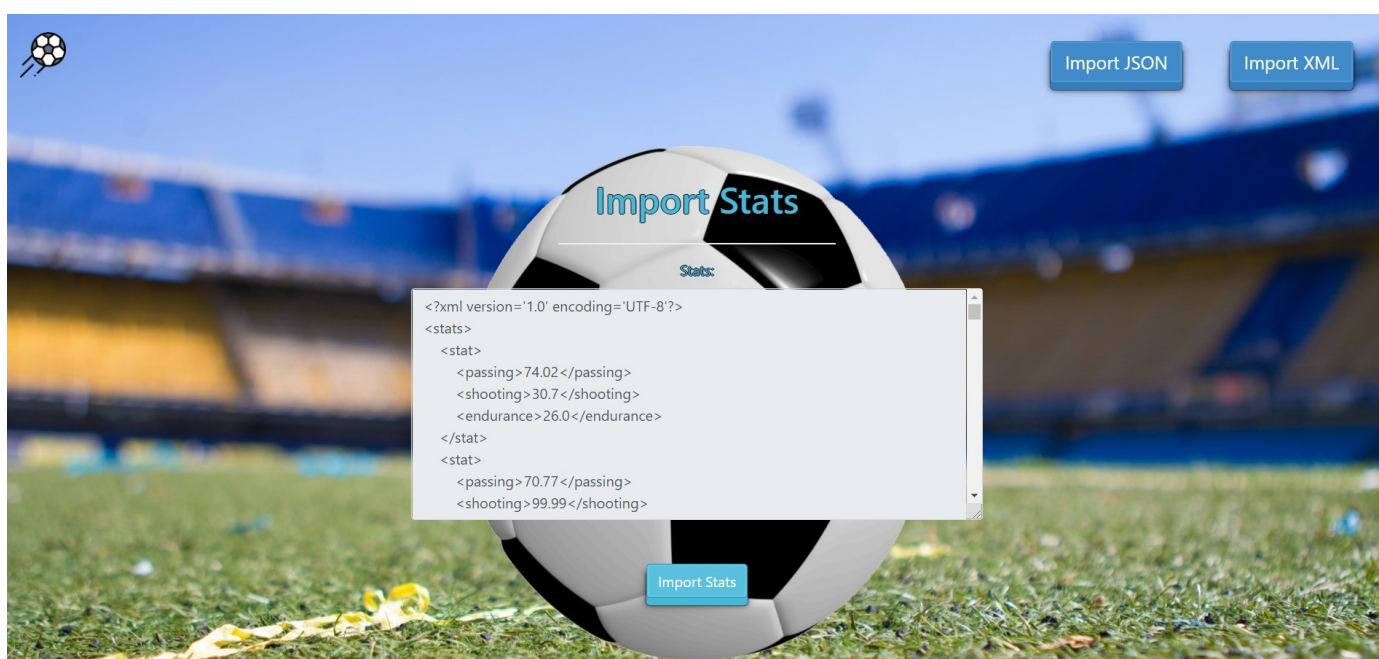
- Import the teams second:

- The import JSON page after importing both files:
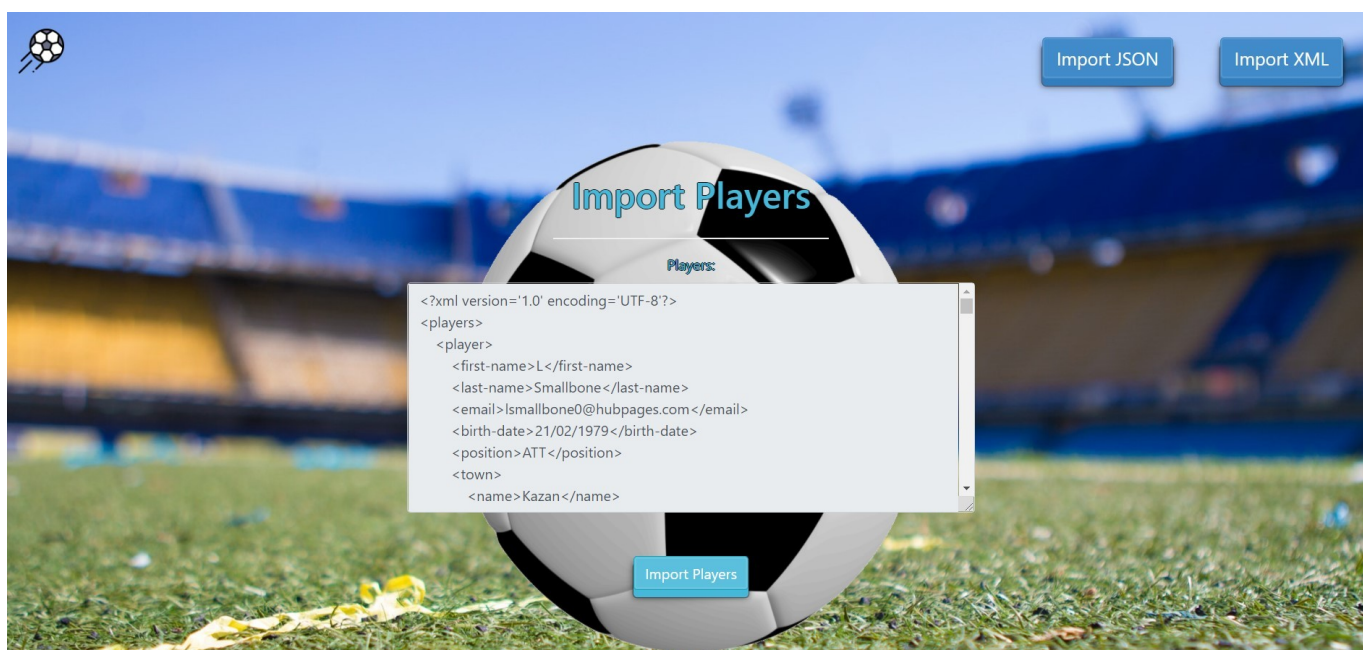
- The import XML page before importing the given data:
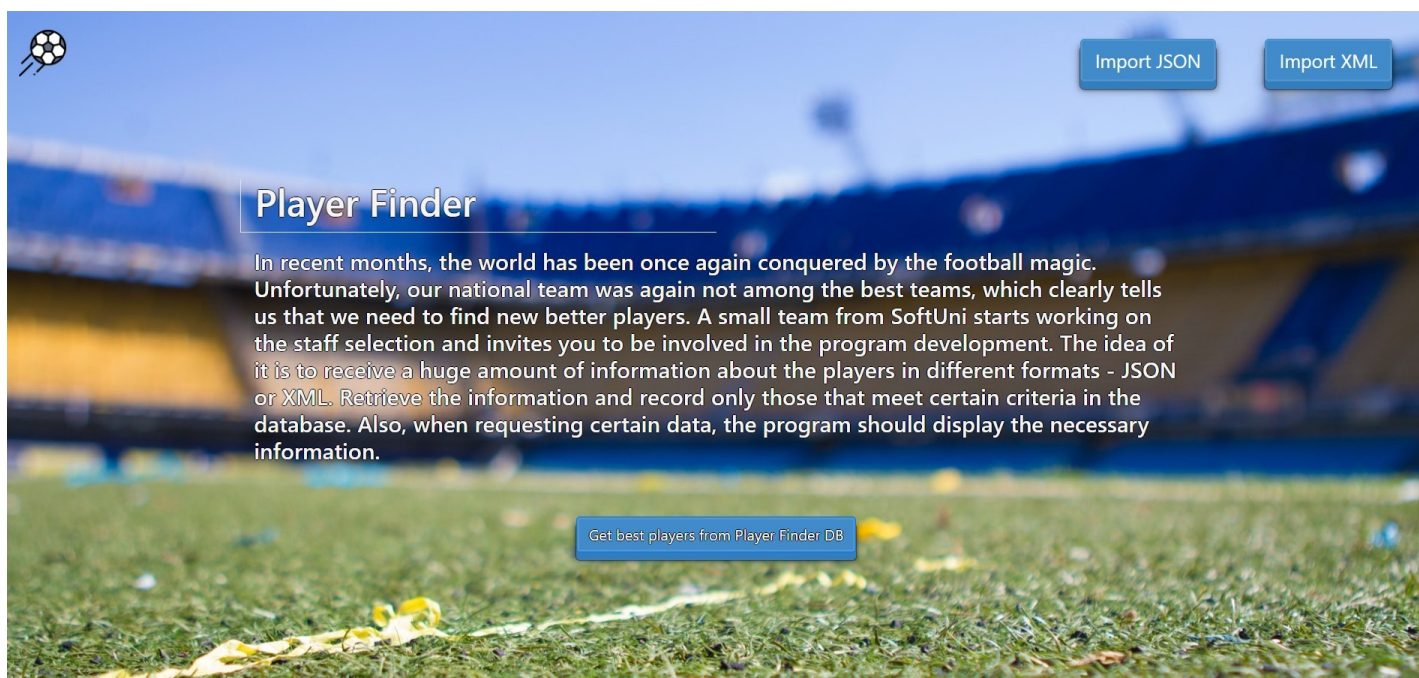

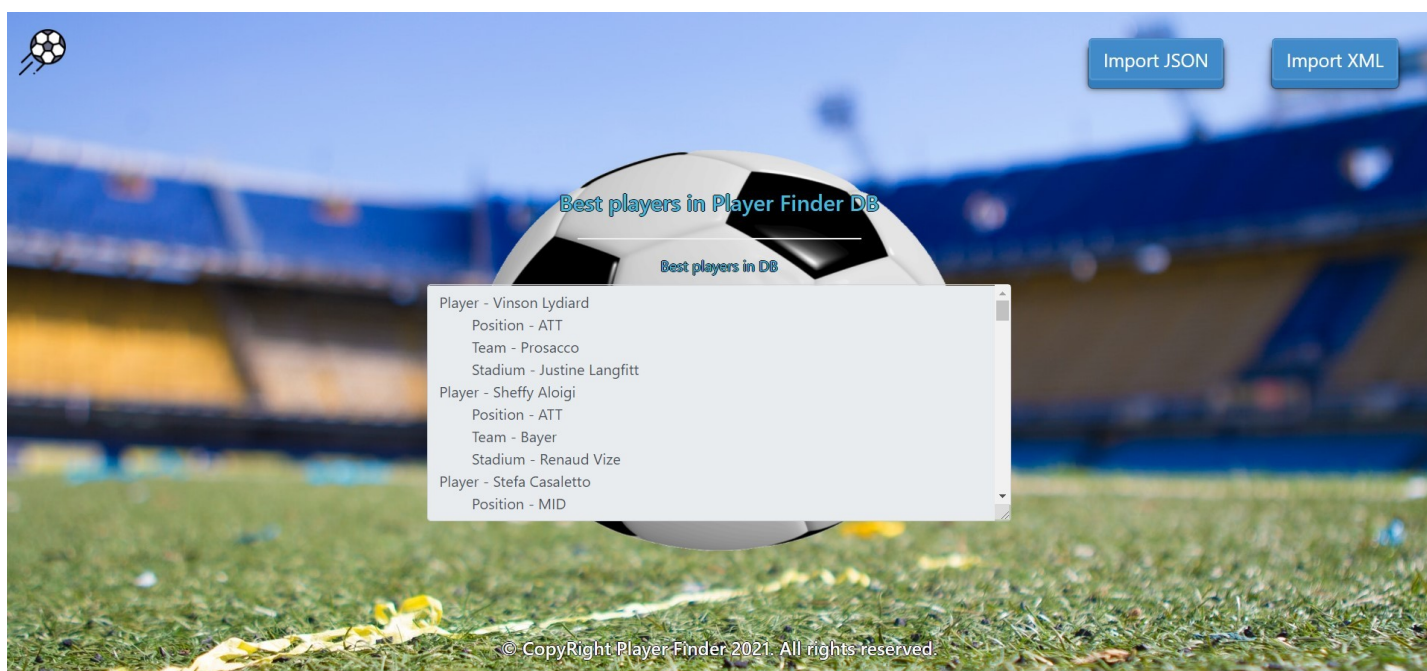- Import the stats data:

- Import the players data:

- The import XML page after importing the data:

- The home page after the data is imported:

- Export best players in Player Finder DB:

2. Project Skeleton Overview

You will be given a **skeleton**, containing a **certain architecture (MVC)** with **several classes**, some of which are completely empty. The **Skeleton** will include the **files** with which you will **seed** the **database**.

3. Model Definition

There are 4 main models that the **Player Finder database** application should contain in its functionality.

Design them in the **most appropriate** way, considering the following **data constraints**:

Town

- **id** – accepts **integer** values, a **primary identification field, an auto incremented field**.
- **name** – accepts **char sequences** as values where their character length value **higher than or equal to 2.** The values are **unique in the database**.
- **population** – accepts **number** values (must be a positive number), 0 as a value is **exclusive**.
- **travel guide – a long and detailed description** of all known places with a character length value **higher than or equal to 10.**

Team

- **id** – accepts **integer** values, a **primary identification field, an auto incremented field**.
- **name** – accepts **char sequences** as values where their character length value **higher than or equal to 3.** The values are **unique in the database**.
- **stadium name** – accepts **char sequences** as values where their character length value **higher than or equal to 3.**
- **fan base** – accepts **number** values that are **more** than or **equal** to **1000**.
- **history** – **a long and detailed description** of team's history with a character length value **higher than or equal to 10.**
 - **Note:** The teams table has relation with the towns table.

Stat

- **id** – accepts **integer** values, a **primary identification field, an auto incremented field**.
- **shooting – a floating point** number**.** The value must be **positive (larger than 0).**
- **passing – a floating point** number**.** The value must be **positive (larger than 0).**
- **endurance – a floating point** number**.** The value must be **positive (larger than 0).**

Player

- **id** – accepts **integer** values, a **primary identification field, an auto incremented field**.
- **first name** – accepts **char sequences** as values where their character length value **higher than 2.**
- **last name** – accepts **char sequences** as values where their character length value **higher than 2.**
- **email** – accepts valid **email addresses** (must contains '@' and '.' – a dot). The values are **unique in the database**.
- **birth date** – a date in the **"dd/MM/yyyy"** format.
- **position** – one of the following – ATT, MID, DEF.
  - **Note:** The players table has relations with the towns, teams and stats tables.

NOTES:

- Name the entities and their class members **exactly** in the **format stated** above.
- All fields are **NOT NULL** unless explicitly stated to be nullable.
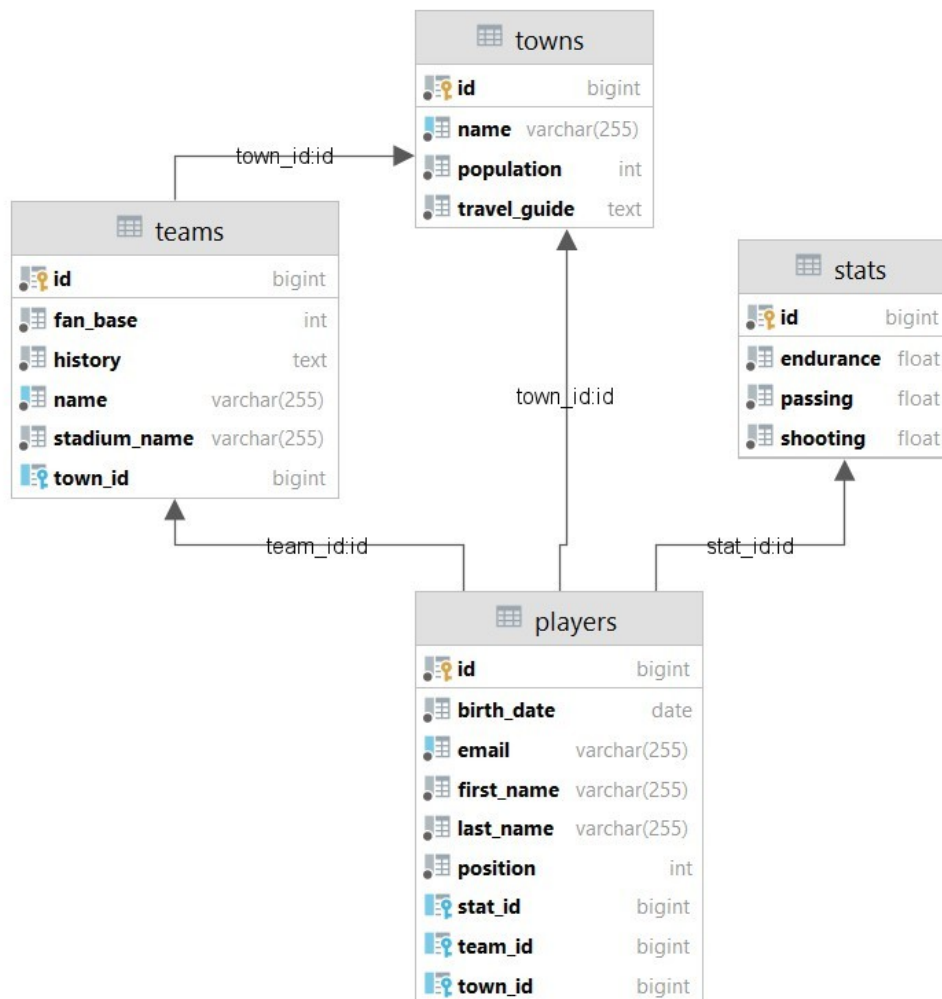
4. Data Import

Use the provided files to populate the database with data. Import all the information from those files into the database.

You are not allowed to modify the provided files.

ANY INCORRECT data should be **ignored** and a message:

"**Invalid {Town / Team / Stat / Player}**" should be printed.

When the import is finished:

"**Successfully imported {Town / Team / Stat / Player} {name - population / teamName - fanBase / passing-shooting-endurance / firstName - lastName - position}**"

Constraint: If the town name already exists in the DB return "Invalid Town"

JSON Import

<table>
<tr><td align="center">**Towns (towns.json)**</td></tr>
</table>

```json
[
  {
    "name": "I",
    "population": 15462452,
    "travelGuide": "Aenean fermentum. Donec ut mauris eget massa tempor convallis.
Nulla neque libero, convallis eget, eleifend luctus, ultricies eu, nibh. Quisque id
justo sit amet sapien dignissim vestibulum. Vestibulum ante ipsum primis in faucibus
orci luctus et ultrices posuere cubilia Curae; Nulla dapibus dolor vel est."
  },
  {
    "name": "Moscow",
    "population": 12195221,
    "travelGuide": "Suspendisse potenti. Cras in purus eu magna vulputate luctus.
Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus
mus. Vivamus vestibulum sagittis sapien. Cum sociis natoque penatibus et magnis dis
parturient montes, nascetur ridiculus mus. Etiam vel augue. Vestibulum rutrum rutrum
neque. Aenean auctor gravida sem. Praesent id massa id nisl venenatis lacinia.
Aenean sit amet justo."
  },
  {
    "name": "London",
    "population": 9126366,
    "travelGuide": "Vestibulum ac est lacinia nisi venenatis tristique."
  },
  {
    "name": "Saint Petersburg",
    "population": -5383890,
    "travelGuide": "Nullam varius. Nulla facilisi. Cras non velit nec nisi vulputate
nonummy. Maecenas tincidunt lacus at velit. Vivamus vel nulla eget eros elementum
pellentesque. Quisque porta volutpat erat. Quisque erat eros, viverra eget, congue
eget, semper rutrum, nulla."
  },
  {
    "name": "Berlin",
    "population": 3748148,
    "travelGuide": "Short."
  },

...
```

Invalid Town
Successfully imported Town Moscow - 12195221
Successfully imported Town London - 9126366
Invalid Town
Invalid Town
…

Constraint:

- **If the team's name already exists in the DB return "Invalid Team".**
- **The provided town names will always be valid.**

<table>
<tr><td align="center">**Teams (teams.json)**</td></tr>
</table>

```json
[
  {
    "name": "Sc",
    "stadiumName": "Tera",
    "fanBase": 317723,
    "history": "Cras pellentesque volutpat dui. Maecenas tristique, est et tempus semper, est quam pharetra magna, ac consequat metus sapien ut nunc. Vestibulum ante ipsum primis in faucibus orci luctus et ultrices posuere cubilia Curae; Mauris viverra diam vitae quam. Suspendisse potenti. Nullam porttitor lacus at turpis. Donec posuere metus vitae ipsum.",
    "townName": "Dallas"
  },
  {
    "name": "Rowe",
    "stadiumName": "Vladamir Paskell",
    "fanBase": 333624,
    "history": "Phasellus in felis. Donec semper sapien a libero. Nam dui. Proin leo odio, porttitor id, consequat in, consequat ut, nulla. Sed accumsan felis. Ut at dolor quis odio consequat varius. Integer ac leo. Pellentesque ultrices mattis odio. Donec vitae nisi.",
    "townName": "Minsk"
  },
  {
    "name": "Marquardt",
    "stadiumName": "M",
    "fanBase": 6211243,
    "history": "Fusce consequat. Nulla nisl.",
    "townName": "Hamburg"
  },
  {
    "name": "Moore",
    "stadiumName": "Paolo Kilpatrick",
    "fanBase": 200,
    "history": "Nulla mollis molestie lorem. Quisque ut erat. Curabitur gravida nisi at nibh. In hac habitasse platea dictumst. Aliquam augue quam, sollicitudin vitae, consectetuer eget, rutrum at, lorem.",
    "townName": "Moscow"
  },
  {
    "name": "Powlowski",
    "stadiumName": "Hussein Jozaitis",
    "fanBase": 9122364,
    "history": "Etiam.",
    "townName": "London"
  },
  ...
```

Invalid Team
Successfully imported Team Rowe - 333624
Invalid Team
Invalid Team

Invalid Team …

XML Import

Your new colleagues have prepared some XML data for you to import.

Constraint:

- **If the stat (a combination of the values of the passing, shooting and endurance) already exists in the DB return "Invalid Stat".**
- **The values for the ID column must start from 1.**
  - **The first valid Stat from the stats.xml must have an ID of 1.**

Stats (stats.xml)
<pre>...
<stat>
<passing>5.75</passing>
<shooting>77.64</shooting>
<endurance>45.09</endurance>
</stat>
<stat>
<passing>35.75</passing>
<shooting>-95.64</shooting>
<endurance>31.08</endurance>
</stat>
<stat>
<passing>12.33</passing>
<shooting>16.74</shooting>
<endurance>-33.09</endurance>
</stat>
<stat>
<passing>22.33</passing>
<shooting>44.89</shooting>
<endurance>-32.11</endurance>
</stat>
<stat>
<passing>40.80</passing>
<shooting>80.80</shooting>
<endurance>-33.09</endurance>
</stat>
</stats></pre>
… Successfully imported Stat 77.64 - 5.75 - 45.09 Invalid Stat Invalid Stat Invalid Stat Invalid Stat

- o **Note: This is the end of the file and the output.**
- o **All stats are rounded to second decimal point**

Players (Players.xml)

Constraint:

- • **If the player's email already exists in the DB return "Invalid Player".**
- • **The provided town and team names will always be valid.**
- • **The Stat id referenced to the valid Stat id.**

Players (players.xml)

```xml
<?xml version='1.0' encoding='UTF-8'?>
<players>
    <player>
        <first-name>L</first-name>
        <last-name>Smallbone</last-name>
        <email>lsmallbone0@hubpages.com</email>
        <birth-date>21/02/1979</birth-date>
        <position>ATT</position>
        <town>
            <name>Kazan</name>
        </town>
        <team>
            <name>McGlynn</name>
        </team>
        <stat>
            <id>53</id>
        </stat>
    </player>
    <player>
        <first-name>Talbert</first-name>
        <last-name>Doherty</last-name>
        <email>tdoherty1@toplist.cz</email>
        <birth-date>05/09/1984</birth-date>
        <position>DEF</position>
        <town>
            <name>San Diego</name>
        </town>
        <team>
            <name>Johnson Group</name>
        </team>
        <stat>
            <id>86</id>
        </stat>
    </player>
    <player>
        <first-name>Regina</first-name>
        <last-name>B</last-name>
        <email>rbyers2@weebly.com</email>
        <birth-date>22/06/1999</birth-date>
        <position>ATT</position>
```

```xml
        <town>
            <name>Paris Minsk</name>
        </town>
        <team>
            <name>Conroy</name>
        </team>
        <stat>
            <id>17</id>
        </stat>
    </player>
    <player>
        <first-name>Adair</first-name>
        <last-name>Dewicke</last-name>
        <email>adewicke3@mail.ru</email>
        <birth-date>11/11/1995</birth-date>
        <position>MID</position>
        <town>
            <name>Philadelphia</name>
        </town>
        <team>
            <name>United D</name>
        </team>
        <stat>
            <id>4</id>
        </stat>
    </player>
    <player>
        <first-name>Herman</first-name>
        <last-name>Adelberg</last-name>
        <email>invalidemail</email>
        <birth-date>29/06/2004</birth-date>
        <position>MID</position>
        <town>
            <name>Cologne</name>
        </town>
        <team>
            <name>Rodriguez</name>
        </team>
        <stat>
            <id>10</id>
        </stat>
    </player>
...
```

Invalid Player
Successfully imported Player Talbert Doherty - DEF
Invalid Player
Successfully imported Player Adair Dewicke - MID
Invalid Player
. . .

5. Data Export

Get ready to export the data you have imported in the previous task. Here you will have some complex database querying. Export the data in the formats specified below.

Export The Best Players from Data Base.

- **Order Them by Shooting in Desc Order, Then by Passing in Desc Order, Then by Endurance Desc Order and Finally Then by Player Last Name.**
- Extract from the database, the **first name**, **last name**, **position**, **team name** and the **name** of the **stadium** of the team.
- Select only players with birth date **after** 01-01-1995 and **before** 01-01-2003
- Return the information in this format:

```
"Player - {firstName} {lastName}
    Position - {position name}
    Team - {team name}
    Stadium - {stadium name}
. . . "
```