**Mobile Applications Development**

**H.Dip in Software Development**

Project

Soccer Player & Quote Reader

**Edivagner Ribeiro**

**Student Number: G00411275**

**email: G00411275@gmit.ie**

Contents

[Introduction 3](#_Toc123478237)

[Innovations 4](#_Toc123478238)

[Settings Page Input 4](#_Toc123478239)

[List-ID and History 5](#_Toc123478240)

[Home Page Layout 7](#_Toc123478241)

[Home Page with Player data 10](#_Toc123478242)

# Introduction

In the project presented, we have all the basic functions implemented according to the description for this assignment. The mobile application reads soccer player data, flag data, and quotes from internet resources.

The query is made through the settings page, if no age range is selected, the app displays the complete list of players available in the database.

If only one of the values is presented, the app returns the search according to the age limits of the search.

To improve the user experience, we optimize the layout for data entry and the layout of results in the table. We have implemented functions on the settings page to check the country ID list and another function to check the search history.

On the home page, we have the option to sort the player's table by first name and surname. When a valid search is performed, however, it does not show results, we have an appropriate message on the screen instead of the table.

# Innovations

## Settings Page Input

The application presents better care in the graphical presentation of the data entry fields on the Settings page, with Inputs Floating Labels. “Floating labels are labels that animate or “float” up when the input is selected. Add the floating attribute to an ion-label to use.”

|  |  |
| --- | --- |
| **\G00411275\src\pages\settings\settings.html**  <ion-list>      <ion-item>        <ion-label floating>Enter Country ID</ion-label>        <ion-input type="number" [(ngModel)]="countryID"></ion-input>      </ion-item>      <br>      <h4>Player Age:</h4>      <ion-item>        <ion-label floating>Minimum</ion-label>        <ion-input type="number" [(ngModel)]="minAge"></ion-input>      </ion-item>      <ion-item>        <ion-label floating>Maximum</ion-label>        <ion-input type="number" [(ngModel)]="maxAge"></ion-input>      </ion-item>    </ion-list> | ***Graphical user interface, text, application  Description automatically generated***  ***Graphical user interface, application  Description automatically generated*** |

## List-ID and History

On the Settings page we add the button to list all the countries in the database with the following query:

getCountries() {

    var url = `https://app.sportdataapi.com/api/v1/soccer/countries?apikey=${this.apiSportKey}`;

    return this.http.get(url);

  }

To present the table with the countries, IDs, and respective flags, we had to update the country code information to uppercase.

In the flags database, we did not get an answer for 5 countries, so we have a special condition for queries in another database.

|  |
| --- |
| **\G00411275\src\pages\settings\settings.ts**  for (let i = 0; i < this.countryData.length; i++) {          if (this.countryData[i].country\_code != null && this.countryData[i].country\_code != undefined) {            if (this.countryData[i].country\_code == "xk") {              this.countryData[i].flag = "https://flagcdn.com/64x48/xk.png";            } else if (this.countryData[i].country\_code == "en") {              this.countryData[i].flag = "https://flagcdn.com/64x48/gb-eng.png";            } else if (this.countryData[i].country\_code == "n") {              this.countryData[i].flag = "https://flagcdn.com/64x48/gb-nir.png";            } else if (this.countryData[i].country\_code == "w") {              this.countryData[i].flag = "https://flagcdn.com/64x48/gb-wls.png";            } else if (this.countryData[i].country\_code == "s") {              this.countryData[i].flag = "https://flagcdn.com/64x48/gb-sct.png";            } else {              this.countryData[i].country\_code = this.countryData[i].country\_code.toUpperCase();              this.countryData[i].flag = "https://flagsapi.com/" + this.countryData[i].country\_code + "/shiny/64.png";            }          }        } |

|  |  |  |
| --- | --- | --- |
| Code | Country ID | Name |
| EN | 42 | Englang |
| N | 88 | Northen Ireland |
| S | 106 | Scotland |
| XK | 69 | Kosovo |
| W | 129 | Wales |

The HISTORY button displays the list of the last queries made by the user. For this functionality, all searches are stored in memory and updated with each query during the session.

|  |  |
| --- | --- |
| allCountryID() {      this.showID = !this.showID;      this.storage.get("countryData")        .then((val\_ID) => {          if (!val\_ID) {            this.navCtrl.pop();          }          else {            const length\_Keys = Object.keys(val\_ID).length;            this.val\_ID\_dataArray = val\_ID.splice(8, length\_Keys);          }        });    } | getHistorySearch() {      this.showHistory = !this.showHistory;      this.storage.get("settingsHistory")        .then((val\_Search) => {          if (!val\_Search) {            this.navCtrl.pop();          }          else {            this.val\_SearchArray = val\_Search;          }        });    } |
| Graphical user interface, text, application  Description automatically generated | Graphical user interface  Description automatically generated with medium confidence |

## Home Page Layout

On the Home Page we implemented a smooth and well-distributed interface of the elements. The flag and player list are hidden by a \*ngIf condition.

|  |  |
| --- | --- |
| **\G00411275\src\pages\home\home.html**  <ion-header>    <ion-navbar>      <ion-buttons left>        <button ion-button icon-only (click)="clickSettings()">          <ion-icon name="settings"></ion-icon>        </button>      </ion-buttons>      <ion-title>G00411275</ion-title>    </ion-navbar>  </ion-header>  <ion-content padding text-wrap>    <ion-list>      <ion-item>        <h1><b>Quote from {{author}} about:</b></h1>        <ul>          <li \*ngFor="let t of tags">{{ t }}</li>        </ul>        <h2><b>{{content}}</b></h2>      </ion-item>      <ion-item>        {{messageHome}}      </ion-item>      <ion-item \*ngIf=flagStatus>        <ion-thumbnail><img src="{{ flagImage }}" alt=" Flag {{myCountry\_Name}}" /></ion-thumbnail>      </ion-item>    </ion-list>    <page-players \*ngIf=flagStatus></page-players>  </ion-content> | Graphical user interface, text, application  Description automatically generated |

When the user performs a valid search, the CountryID and name are updated on the screen, the flag is displayed and the list of players corresponding to the search is presented in a well-formatted table.

|  |  |
| --- | --- |
| Graphical user interface, application  Description automatically generated | Graphical user interface, text, application  Description automatically generated |

If the user performs a search from a valid country, however, the age range does not match any results, a message is displayed on the Home Page at the location of the table.

  <div class="not-match" \*ngIf="list\_length == 0">

    <p>

      Your search did not match any documents!

    </p>

    <img src="../../assets/imgs/empty-folder.png" width="75px" />

  </div>

|  |  |
| --- | --- |
| Graphical user interface, text, application  Description automatically generated | Graphical user interface, application  Description automatically generated |

## Home Page with Player data

The player's table can be manipulated dynamically by the user. It is possible to sort the list by First Name or Surname, as well as delete some values from the table that is displayed on the screen with the icon in the Action column.

|  |  |
| --- | --- |
| **\G00411275\src\pages\players\ players.html**  <!------------------ HEADER ----------------------->    <ion-row class="header-row" \*ngIf="list\_length > 0">      <ion-col size="6" size-md="4" size-lg="2" tappable (click)="sortBy('firstname')">        First Name        <ion-icon name="arrow-down" \*ngIf="sortDirection == 1  && sortKey == 'firstname'"></ion-icon>        <ion-icon name="arrow-up" \*ngIf="sortDirection == 2  && sortKey == 'firstname'"></ion-icon>      </ion-col>      <ion-col size="6" tappable (click)="sortBy('lastname')">        Surname        <ion-icon name="arrow-down" \*ngIf="sortDirection == 1 && sortKey == 'lastname'"></ion-icon>        <ion-icon name="arrow-up" \*ngIf="sortDirection == 2  && sortKey == 'lastname'"></ion-icon>      </ion-col>      <ion-col size="2">        Age      </ion-col>      <ion-col size="2">        Height      </ion-col>      <ion-col size="2" \*ngIf="bulkEdit">        Action      </ion-col>    </ion-row> | **\G00411275\src\pages\players\ players.ts**   sortBy(key: any) {      this.sortKey = key;      this.sortDirection++;      this.sort();    }    sort() {      if (this.sortDirection == 1) {        this.dataArray = this.dataArray.sort((a, b) => {          const valA = a[this.sortKey];          const valB = b[this.sortKey];          return valA.localeCompare(valB);        });      } else if (this.sortDirection == 2) {        this.dataArray = this.dataArray.sort((a, b) => {          const valA = a[this.sortKey];          const valB = b[this.sortKey];          return valB.localeCompare(valA);        });      } else {        this.dataArray = this.dataArray.sort((n1, n2) => {          if (n1.player\_id > n2.player\_id) {            return 1;          }          if (n1.player\_id < n2.player\_id) {            return -1;          }          return 0;        });        this.sortDirection = 0;        this.sortKey = null;      }    }    removeRow(index: number) {      this.dataArray.splice(index, 1);    } |

|  |  |
| --- | --- |
| Graphical user interface, text, application  Description automatically generated | Sorted by First Name  Table  Description automatically generated  Sorted by First Name  Table  Description automatically generated |
| Sorted by Surname  Table  Description automatically generated | Sorted by Surname  Table  Description automatically generated |