**MOBILE FOOTBALL APP REQUIREMENTS DOCUMENT**

**FUNCTIONAL REQUIREMENTS**

* The system should provide an http client to communicate with the API’s endpoints.
  + It should be able to perform GET http calls to the available endpoints using the parameters that were provided by the user interaction.
    - It should be able to get the response from the ***/countries*** endpoint.
      * It should be able to send the **search** parameter to filter the countries.
    - It should be able to get the response from the ***/leagues*** endpoint.
      * It should be able to send the country’s **code** and the **season** parameters to get the available leagues and cups related to those filters.
    - It should be able to get the response from the ***/teams*** endpoint.
      * It should be able to send the league’s **code** and the **season** parameters to get the teams available for the specific league and season.
      * It also should be able to send the **team’s** **id** parameter to get a specific team.
    - It should be able to get the response from the **/fixtures** endpoint.
      * It should be able to send the **league’s** **id** and **season** parameters to get the fixtures available for the specific league and season.
      * Also it should be able to send the **team’s id** parameter to get the fixture related to the specific team.
    - It should be able to get the response from the ***/fixtures/events*** endpoint.
      * It should be able to send the **fixture’s id** to get all the events related to that fixture
    - It should be able to get the response from the ***/standings*** endpoint.
      * It should be able to send the **league’s id** and **season** parameters to the corresponding standings.
    - It should be able to get the response from the ***/squads*** endpoint.
      * It should be able to send the **team’s id** parameters to get its squad
    - It should be able to get the response from the ***/players*** endpoint.
      * It should be able to send the **player’s id** to get the player’s details
* The system should be able to parse the JSON responses from the different endpoints into objects that can be used.
  + It should be able to parse the response from the ***/countries*** endpoint.
  + It should be able to parse the response from the ***/leagues*** endpoint.
  + It should be able to parse the response from the ***/teams*** endpoint.
  + It should be able to parse the response from the ***/fixtures*** endpoint.
  + It should be able to parse the response from the ***/fixtures/events*** endpoint.
  + It should be able to parse the response from the ***/standings*** endpoint.
  + It should be able to parse the response from the ***/squads*** endpoint.
  + It should be able to parse the response from the ***/players*** endpoint.
* As a user, I want to be able to see a grid split into 3 columns with all the countries (with their names and flags) that are part of the FIFA.
  + So, I can select any of those countries.
* As a user, I want to be able to filter countries by tipping their names in a search bar at the top of the screen.
* As a user, I want to be able to select one country and see a grid split into 2 columns with all the available leagues and cups (with their names and logos) related to that country.
  + So, I can choose any of the available leagues/cups related to the previous selected country and see the available seasons for that tournament.
  + The screen should include a title at the top with the selected country’s name.
* As a user, I want to be able to filter leagues by tipping their names in a search bar at the top of the screen
* As a user, I want to be able to see a list with all the available seasons related to the previous selected tournament.
  + So, I can select any of them.
  + The screen should include a title at the top with the selected league’s name.
* As a user, I want to be able to see a screen divided into 3 sections/tabs for teams, fixtures and standings.
  + I should be able to scroll horizontally across the different tabs.
  + I should be able to tap on any of the tabs and see its section.
  + For the team section, I want to be able to see a grid split into 2 columns with all the available teams related to the previous selected tournament and season.
    - So I can select any of them a see its details.
    - It should display the team’s name and logo.
  + For the fixture section, I want to be able to see a list with the latest matches.
    - It should display the last 10 fixtures for the current week.
    - If there are not fixtures available for the current week, it should the display the last 10 available matches.
  + For the standing section, I want to be able to see a table for each of the available standing groups.
    - The tables should be ordered by the team’s rank in descending order (from the 1st team to the latest team in the ranking).
    - At the top of the table it should display the league’s name and season, including the group’s description. Below it should be divided by columns.
      * It should have a column for the clubs.
      * It should have a column for the played fixtures.
      * It should have a column for the won fixtures.
      * It should have a column for the draw fixtures.
      * It should have a column for the loose fixtures.
      * It should have a column for the points.
    - Each row of the table should contain the club information for each of the columns (club – played – wins – draws – loses - points).
      * It should include a status indicator according to the team status in the ranking (up – same - down).
      * The club section should be able to be pressed to see the team’s details.
    - At the bottom it should display a glossary.
* As a user, I want to able to see the team details after selecting a team from the team’s section.
  + It should include the team’s stadium image (if it’s available) at the top of the screen.
  + Also it should include a card with the team’s information (address, foundation year, capacity, city, country).
  + At the bottom it should display 2 buttons.
    - One of them to see the team’s squad.
    - And the other to see the fixture related the specific team.
* As a user I want to be able to see the squad for a specific team.
  + So I can select any of the team’s players
  + It should be displayed as a list separated by the player’s role (goalkeeper – defender – midfielder - attacker).
* As a user I want to be able to see the player’s details.
  + It should show its team’s logo
  + A player’s photo (if it’s available)
  + The player’s information. Such as, weight, height, age, first name, last name, birth date, place, nationality and other available details.
* As a user, I want to able to see the events related to a certain fixture.
  + It should be displayed as a list.
    - At the top it should display the results
    - Each event should be shown with the following information
      * The player’s name and team
      * The minute were the event happened
      * An icon to represent the event (goal – substitution – yellow card – red card)

**NON-FUNCTIONAL REQUIREMENTS**

* To build this app, it must be done using the Flutter framework
* The system should be split into 3 different layers.
  + Data Layer
    - It should be responsible to provide the interface with the API and provide an http client to communicate with it
    - It should provide the required models for the app
  + Domain Layer
    - It should be the responsible to parse the models from the data layer into the required domain models
    - It should provide the corresponding repositories for each domain model
    - It should provide a cache implementation
      * to save the information that was already requested
      * to reduce the amount of calls to the API endpoints
      * the cache save the information during a session and temporary saving for the information that needs to be updated in a short time
  + Presentation/Application Layer
    - This layer should be responsible to provide the UI components that are required for the application
      * The UI components like texts, cells, grids, buttons, tables should be independent from the models so it can work with any of them
      * It should provide the connections and the navigation logic to transition from one screen to another
    - It should provide view models for the main screens in the app
* The application should use the BLoC pattern (<https://bloclibrary.dev/#/architecture> ) to manage the application resources and also the navigation events.
* The icons of the app would be provided by a third library called font\_
* awesome\_flutter