



Progetto TecWeb A.A. 2017/2018
Riccardo Damiani 1125327

<http://tecweb2016.studenti.math.unipd.it/rdamiani>

Indice

1. Abstract -----	3
2. Struttura -----	4
2.1 File System -----	4
2.2 Architettura -----	4
2.3 Documentazione -----	5
3. Diagramma delle classi -----	6
4. Database -----	7
4.1 Struttura -----	7
4.2 Popolamento -----	7
5. Features -----	8
5.1 Header per la paginazione -----	8
5.2 Ordinamento colonne -----	9
5.3 Barra di ricerca -----	9
5.4 Calendario -----	10
5.5 Formazione -----	11
5.6 Radar Chart -----	12
5.7 Commentary -----	13
6. Accessibilità -----	14
6.1 Struttura e validazione -----	14
6.2 Lingua -----	14
6.3 Meta Tag -----	14
6.4 Immagini -----	14
6.5 Breadcrumb -----	15
6.6 Screen Reader -----	15
6.7 Scripting -----	15
7. Usabilità -----	16
7.1 Struttura grafica -----	16
7.2 Responsive design -----	17
8. Test -----	18
8.1 Desktop -----	18
8.2 Mobile -----	19

1. Abstract

Il sito proposto come progetto del corso di tecnologie web consiste in una rassegna di informazioni riguardanti la stagione calcistica 2015/2016 dei principali campionati europei. Sono presenti dati relativi alle varie partite disputate, alle squadre partecipanti ed a tutti i giocatori ed ai membri appartenenti all'organico delle varie società calcistiche.

Il sito ha principalmente uno scopo informativo, riportando tutte le informazioni riguardanti i campionati, perciò non vi sarà alcuna particolare interazione con l'utente, che potrà così principalmente visualizzare le pagine e leggerne le informazioni riportate, fatta esclusione per la possibilità di usare la barra di ricerca per cercare squadre o giocatori (oppure tramite il calendario nella pagina delle partite od ordinando le tabelle nelle altre pagine se l'interesse dell'utente dovesse essere quello di trovare un particolare elemento informativo).

L'utente potrà registrarsi con le sue credenziali e specificare la sua squadra preferita, che apparirà a destra della barra di navigazione una volta portata a termine l'autenticazione.

2. Struttura

2.1 File System

La struttura dei file è stata organizzata in maniera gerarchica ad albero in cui alla radice si trova la home page, e via discendendo verso le foglie sono annidate delle cartelle che corrispondono a grandi linee ad ogni pagina del sito, accessibile tramite il file “index.php” contenuto in ogni cartella rappresentante una pagina.

Nel percorso radice ci sono inoltre le cartelle “js”, che contiene i pochi script scritti utilizzando javascript, “css”, che contiene il foglio di stile onnicomprensivo che viene linkato da ogni pagina, “img”, che contiene tutte le immagini e le icone presenti nel sito (ho deciso di tenere una copia in locale di ogni immagine per non essere costretti ad essere online per far funzionare il tutto), “fonts”, che contiene i font custom utilizzati dal sito ed infine “templates” che contiene gli script e le classi comuni a diverse pagine del sito.

2.2 Architettura

In fase di progettazione e di sviluppo si è cercato di utilizzare le funzionalità di *PHP* che dalla versione 5 permette di aderire agli standard di programmazione Object Oriented. Non è stato possibile programmare interamente ad oggetti ed è stato utilizzato in parte il paradigma procedurale nei punti del codice in cui gli script si alternano a codice *html*.

Le classi principali utilizzate dal sito e strutturate in modo tale da essere estese ed implementate hanno dei metodi astratti che ne definiscono il comportamento specifico, oltre al costruttore delle classi derivate che ha il compito di assemblare le varie componenti utilizzando per lo più i metodi della classe base, sono:

- ***DBConnection***: Classe che utilizza il design pattern *singleton*, rappresenta la connessione con il database e ne veicola l’interazione con le pagine, i form, gli script e le tabelle. Utilizza internamente *mysqli* e possiede un metodo statico (*DBConnection::console_log*) utile in fase di debugging che permette di stampare dei messaggi sulla console del browser.
- <<*abstract*>> ***Table***: Classe che rappresenta una tabella ed il suo contenuto, utilizza un *DOM* internamente per facilitare la ricerca dei nodi e la loro gestione, possiede un *ID* che identifica la tabella e permette di effettuare operazioni specifiche come l’ordinamento (implementato da *DBTable*) o la paginazione utilizzando la *query string* presente nell’URL della pagina.
- <<*abstract*>> ***DBTable***: Versione evoluta di *Table*, che al costo di un overhead leggermente maggiore espone allo sviluppatore un’interfaccia più complessa e completa della sua classe base. Permette di

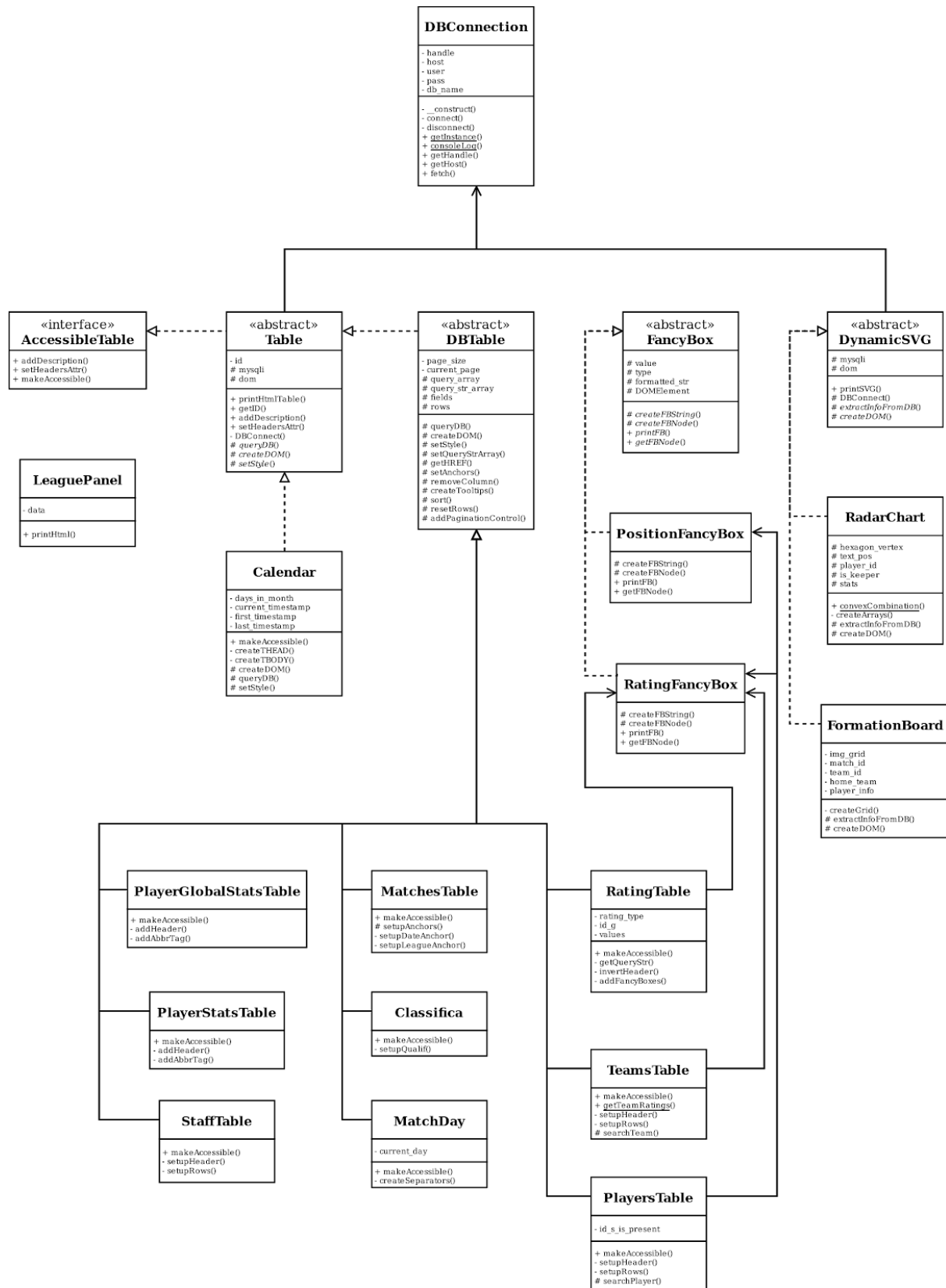
aggiungere un header con dei controlli per la paginazione dei contenuti della tabella (solitamente di grandi dimensioni) e di ordinare le righe della tabella secondo l'ordine stabilito dall'utente cliccando sulle intestazioni delle colonne, che al bisogno possono anche avere dei tooltip (per esplicitare gli acronimi). Sia *Table* che *DBTable* implementano l'interfaccia *AccessibleTable* che impone l'implementazione di alcuni metodi che rendono le tabelle più accessibili.

- <<abstract>> **FancyBox**: Porzione di testo racchiusa all'interno di un tag *span* che cambia colore in base al contenuto rappresentato, viene implementata da *PositionFancyBox*, che rappresenta i ruoli dei giocatori in campo (si presuppone che l'utente abbia dimestichezza con i formalismi di FIFA, in particolare gli acronimi che rappresentano i ruoli, che possono essere facilmente trovati online) e da *RatingFancyBox* che rappresenta la valutazione di una certa abilità del giocatore.
- <<abstract>> **DynamicSVG**: Classe base delle immagini vettoriali in formato svg costruite dinamicamente a partire dai dati presenti nel database, in particolare *FormationBoard* che disegna la posizione dei giocatori in campo a partire dalle coordinate presenti per ogni giocatore nella tabella *Formazione* all'interno della base di dati, e *RadarChart* che mostra all'utente un grafico in cui le principali abilità di ogni giocatore sono raggruppate in aree.

2.3 Documentazione

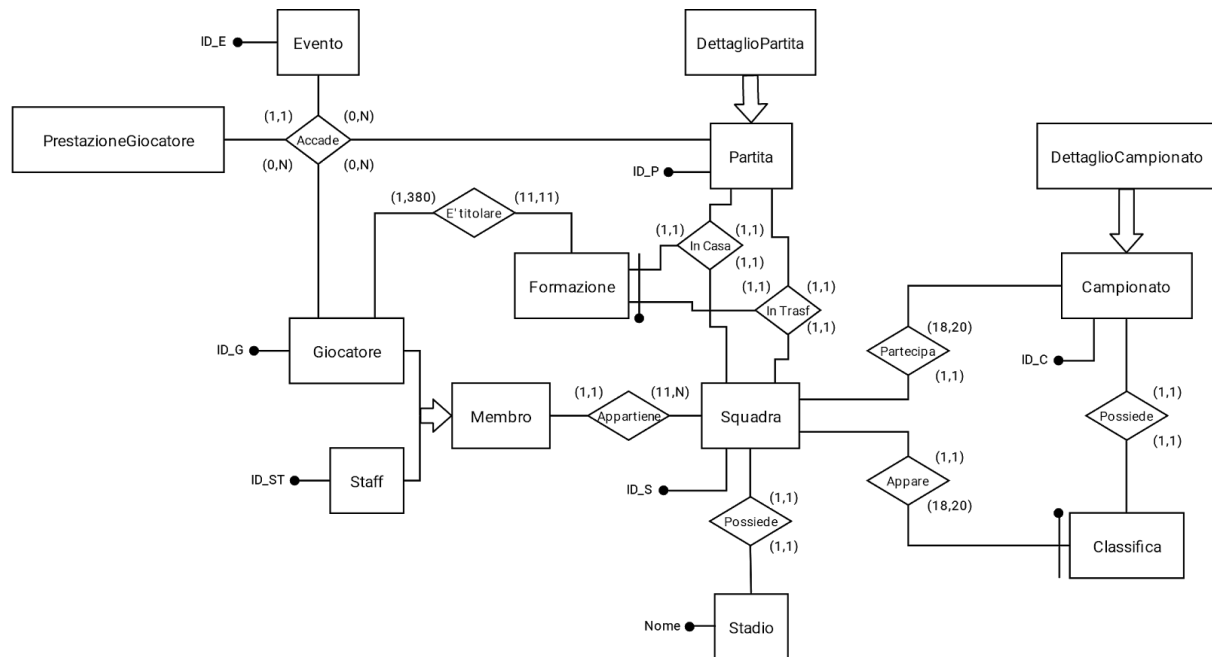
Non sono riuscito a commentare tutto il codice, ma ho descritto i passaggi fondamentali delle procedure più complesse, e ho commentato utilizzando la sintassi di *doxygen* le classi fondamentali, come *DBConnection* e *DBTable*.

3. Diagramma delle classi



4. Database

4.1 Struttura



Il database e la presentazione dei dati tramite l'interfaccia web realizzata dal sito sono stati sinergici fin dall'inizio, uno ha influenzato lo sviluppo dell'altro vicendevolmente. E' stato possibile realizzare e popolare la base di dati grazie alla creazione di vari *crawler* scritti utilizzando *python* ed il framework *scrapy*¹. Il database contiene esclusivamente dati relativi alla stagione calcistica 2015/2016 dei 6 maggiori campionati europei (Prima divisione di Inghilterra, Francia, Germania, Italia, Olanda e Spagna). Le tabelle più sostanziose sono *Evento*, *Giocatore* e *Partita* che contengono rispettivamente 194727, 2860 e 2132 records. La struttura con i vari constraint e gli attributi delle relazioni rappresentate dalle tabelle della base di dati non è descritta in dettaglio in questa sede, ma è riportata in cima alla pagina uno schema Entity-Relationship per avere un'idea generale della sua architettura. Nonostante il database sia il cuore del progetto vorrei spostare il focus più sulla presentazione dei dati che sulla loro organizzazione interna.

4.2 Popolamento

I dati sono stati estrapolati principalmente dal maggiore portale di informazioni calcistiche *Transfermarkt*² (per le informazioni riguardanti le partite) e dal sito web della *EA Sports*³ (per le valutazioni dei giocatori

¹ <https://scrapy.org>






















² <https://www.transfermarkt.it>

³ <https://www.easports.com/it/fifa/ultimate-team/fut/database>

e per i volti) e rimaneggiate in maniera tale da prestarsi allo scopo del progetto. La struttura sia del database che delle varie pagine del sito è stata concepita in modo tale da permettere la sua estensione futura, con l'aggiunta di nuove informazioni (ulteriori stagioni) e nuove sezioni (commenti e notizie), come per esempio il calciomercato, che per motivi di tempo non è stato possibile prendere in considerazione durante lo sviluppo.

5. Features

5.1 Header per la paginazione

Show 50 ▼ 100 to 150 of 2860 First Prev 2 3 4 Next Last						
TOT	Nome		Ruolo_Preferito		Età	
  	Bernardo Silva				24 	
  	Virgil van Dijk				27 	
  	Marquinhos				24 	
  	Javier Martinez		 		29 	

Grazie all'utilizzo della libreria *php-sql-parser*⁴ mi è stato possibile modificare la query salvata all'interno di *DBTable* andando ad aggiungere all'array associativo contenente tutte le componenti strutturate della query l'istruzione *LIMIT* a cui sono associati due valori, offset e base, che permettono la paginazione direttamente utilizzando *mysql*. Come è possibile notare nell'immagine in alto è stato aggiunto ad alcune tabelle molto grandi un nuovo header contenente alcuni elementi delle form come un dropdown button che permette (tramite query string) di informare il server sul numero di righe che si desidera visualizzare all'interno della tabella, e più a destra dei submit button che permettono (sempre tramite query string) di muoversi all'interno delle varie pagine in cui sono stati organizzate le migliaia di record che andrebbero altrimenti ad invadere lo spazio della pagina a discapito dell'accessibilità e dell'usabilità.

⁴ <https://code.google.com/archive/p/php-sql-parser>

5.2 Ordinamento colonne

Squadra	Punti 1 ▼	P_Vinte 2 ▼	P_Pareggiate	P_Perse	G_Fatti	G_Subiti	Diff_Reti 3 ▼
Leicester City	81	23	12	3	68	36	32
Arsenal	71	20	11	7	65	36	29
Tottenham Hotspur	70	19	13	6	69	35	34
Manchester City	66	19	9	10	71	41	30

Sempre attraverso la libreria *php-sql-parser* e l'array associativo salvato all'interno della variabile *DBTable::query_array*, è possibile modificare la query *mysql* aggiungendo l'istruzione *ORDER BY* che permette di ordinare tutte le tabelle (per cui abbia senso ordinare i risultati) in base alle colonne selezionate cliccando sull'intestazione dell'header, che contiene un *anchor* che aggiunge delle variabili nella *query string* che istruiscono il server su come costruire la tabella a seguito di un necessario refresh della pagina. Le stesse funzionalità si potevano alternativamente implementare utilizzando script client side, ma ho preferito farlo tramite *PHP*, per motivi di coerenza con i principi di accessibilità riportati nella sezione corrispondente.

5.3 Barra di ricerca



All'interno della navbar, nella parte destra prima dell'icona per il login, se si è all'interno della pagina delle squadre o di quella dei giocatori, è presente una barra di ricerca che permette di inserire il nome della squadra o del giocatore che si stanno cercando ed essa, se presente nel database, verrà successivamente visualizzata all'interno della tabella corrispondente.

5.4 Calendario

<< 2015-08 >>						
Lun	Mar	Mer	Gio	Ven	Sab	Dom
27	28	29	30	31	01	02
03	04	05	06	07	08	09
10	11	12	13	14	15	16
17	18	19	20	21	22	23
24	25	26	27	28	29	30
31	01	02	03	04	05	06

Nella pagina che permette di cercare le partite ho pensato di aggiungere, oltre alla tabella con possibilità di paginazione contenente tutte le partite situata in basso, anche un calendario collegato ad una tabella contenente tutte le partite che si sono svolte nel giorno selezionato fosse utile e più intuitiva e accelerasse i tempi necessari all'utente per individuare la giornata di campionato contenente la partita a cui è interessato. Come si può vedere dallo screenshot ogni cella della tabella che rappresenta una pagina del calendario ha diverse colorazioni, che rispecchiano l'appartenenza del giorno al mese corrente o a quello precedente/successivo, la presenza o meno di partite (che la rende selezionabile o meno) ed il giorno corrente selezionato.

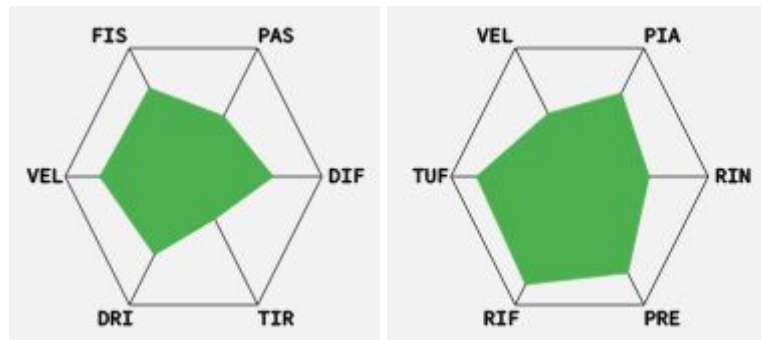
5.5 Formazione



Per mostrare i dati presenti nel database relativi alla formazione di ogni partita, la classe *FormationBoard* a partire dall'immagine vettoriale di un campo da calcio posiziona con precisione ogni giocatore (all'interno di una griglia 9x11 non visibile all'utente), rappresentato da una maglia (bianca se in casa e nera se in trasferta, caso che è possibile vedere in figura) contenente il suo numero di maglia che permette di associarlo al suo nome presente al di sotto dell'immagine nella pagina della partita selezionata.

Per rendere il codice del file svg creato dalla classe *FormationBoard* snello e leggibile ho utilizzato i tag *symbol* contenuti all'interno di *defs* situato subito dopo l'apertura del nodo radice, che descrivono (sono degli svg annidati) le icone delle maglie bianche/nere e che possono essere richiamati grazie al tag *use* tramite il loro *id* univoco, rendendo tutto più compatto e di più facile gestione, oltre che elegante.


5.6 Radar Chart



Tramite la classe *RadarChart* anch'essa derivata da *DynamicSVG* come per la formazione vengono estratti i dati dal database, questa volta relativi alle abilità dei giocatori, e mostrati dinamicamente (i.e. per ogni pagina) all'utente visitatore. Ogni angolo dell'esagono contiene uno dei sei gruppi di abilità che rappresentano le capacità calcistiche dell'atleta in questione (secondo le stime fatte dalla *EA Sport* che si basano su statistiche aggiornate di anno in anno, nel caso particolare del progetto ricordo che fanno riferimento all'anno 2015/2016). Il poligono irregolare inscritto nell'esagono va a distendersi in base al valore di queste abilità, più si avvicina a 99 più i vertici dei due poligoni tenderanno a sovrapporsi.

Per posizionare i vertici del poligono irregolare ho utilizzato la formula della "Combinazione Convessa" a due punti, che permette di posizionare il vertice del poligono a partire da un coefficiente *lambda* che determina a che distanza si trova il punto dagli estremi del raggio appartenente alla circonferenza che circonda l'esagono, la funzione statica *RadarChart::convexCombination(\$first_point,\$second_point,\$lambda)* permette di fare questo calcolo e restituisce il punto dove posizionare il vertice.

5.7 Commentary

		9' Goal! E' Konstantin Rausch a segnare per la sua squadra. Il risultato è ora di 0-1, l'arbitro fischia e la palla è di nuovo al centro! Goal
		9' Che passaggio di fino! Assist da parte di Konstantin Rausch Assist
11'		Julian Draxler ci prova dalla distanza, ma dovrà aggiustare la mira se vuole fare male al portiere al prossimo tentativo! Tiro fuori
12'		Joel Matip prova un tiro in porta, ma la sua precisione lascia a desiderare, tentativo fuori dallo specchio della porta. Tiro fuori
18'		Palla alta servita in area proveniente da un corner a destra, a calciare è Johannes Geis. Calcio d'angolo
		19' Fallo di Aytac Sulu che afferra vistosamente la maglia del suo avversario. A subire il fallo è stato Klaas Jan Huntelaar. Fallo

In questa sezione della pagina relativa ad una delle partite disponibili vengono presentati tutti gli eventi salienti della partita, descritti a partire dalle informazioni presenti nella tabella *Evento* del database. Gli eventi riguardano goal (con tipo di goal, risultato aggiornato, marcatore e assistman) falli con tipologia di cartellino annessa e i giocatori coinvolti, calci d'angolo, calci di punizione, tiri in porta e fuori (e tipo di tiro, dalla rovesciata al tiro sul palo o la traversa). Particolare attenzione è stata rivolta anche al layout, che è stato realizzato dopo numerosi tentativi fallimentari con metodi diversi, alla fine è stato possibile posizionare l'icona al centro grazie agli utilissimi strumenti messi a disposizione da *flexbox*. La sezione di testo dell'evento riporta il minuto (più eventuali minuti di recupero) una descrizione simpatica che fa uso del gergo calcistico dell'evento considerato, e la tipologia dell'evento stesso, è posizionata a sinistra della sua icona se l'evento riguarda la formazione di casa e a destra se quella coinvolta è quella in trasferta.

6. Accessibilità

6.1 Struttura e Validazione

Nella progettazione del sito ho cercato di aderire il più possibile alle raccomandazioni W3C, il codice è stato validato utilizzando la piattaforma *Unicorn*⁵. Nonostante il contenuto e la natura del mio progetto non si prestasse molto, soprattutto per la presenza di molte tabelle, ad una estensiva revisione dell'accessibilità, ho cercato di renderlo accessibile alla maggior parte delle utenze, con disabilità di vari generi. Ho deciso di utilizzare HTML5 e CSS3 per cercare di realizzare un prodotto aggiornato e di apprendere tecniche al passo con i tempi e che non fossero inutili in futuro. Ho evitato anche per questo l'uso di tag e attributi deprecati, oltre che per gli ovvi motivi. La struttura del codice rispetta la separazione tra contenuto e presentazione, tutta la parte di presentazione risiede in un file CSS separato e non è mai dichiarata inline, HTML è utilizzato solo per i contenuti delle varie pagine e non è mai utilizzato per presentare le informazioni.

6.2 Lingua

Il sito è stato scritto interamente in italiano, perciò il tag *html* di apertura che contiene l'attributo *lang* fa riferimento alla lingua italiana, attributo che è stato utilizzato anche nei tag che contengono testo espresso in altre lingue (la lingua inglese nel titolo dei form ad esempio).

6.3 Meta Tag

Ogni pagina contiene all'interno del tag *head* i meta-tag *title*, che attribuisce un titolo alla pagina, *description*, in cui è presente una breve descrizione, *author*, contenente l'autore del sito, *charset*, che indica il tipo di codifica del testo, ed infine *viewport*, che permette una corretta visualizzazione responsiva della pagina suggerendo al browser di renderizzare alla stessa larghezza dello schermo.

6.4 Immagini

Ogni immagine ha un testo alternativo contenuto nell'attributo *alt*, che viene mostrato all'utente nel caso in cui non fosse possibile caricare l'immagine, o per gli screen reader. Le uniche immagini sprovviste di questo attributo sono quelle decorative, senza contenuti informativi, le immagini inoltre non sono mai state utilizzate per riportare testo, quindi il contenuto informativo è sempre accessibile, anche nel caso in cui dovesse fallire il caricamento delle immagini.

⁵ <https://validator.w3.org/unicorn>

6.5 Breadcrumb

Tutte le pagine hanno in alto sotto la barra di navigazione un menu *breadcrumb*, utile a migliorare l'esperienza utente permettendogli di tenere traccia del suo percorso all'interno del sito web.

Nelle pagine dove si è rivelato utile e necessario, è stato posizionato un *anchor* "torna su" che rimanda ad inizio pagina senza costringere gli utenti ad usare lo scrolling laterale.

6.6 Screen Reader

Ogni campo presente nei vari form è corredato da una etichetta *label* oppure *aria-label* (presente anche sui button, per non rovinare l'estetica del sito senza inficiare sull'accessibilità) e le varie voci sono raggruppate all'interno dei tag *fieldset*, inoltre per rendere il sito accessibile solo tramite tastiera sono stati utilizzati i tag *tabindex* ed *autofocus*.

Per motivi di spazio orizzontale alcuni header delle colonne sono stati ridotti a degli acronimi, quando ciò si è rivelato necessario sono però sempre stati affiancati da tooltip che appaiono con l'hover del puntatore del mouse, e dai tag *abbr* per facilitare la comprensione delle abbreviazioni agli utenti che utilizzano gli screen readers.

Nel tentativo di rendere le tabelle attraversabili in maniera più o meno agevole per gli screen reader, ho ritenuto utile aggiungere gli attributi *aria-describedby* per descrivere il contenuto delle tabelle e *headers* per collegare tramite l'*id* ogni riga al suo *header* di riferimento.

6.7 Scripting

Infine ci tengo a precisare che l'utilizzo di script client side è stato ridotto all'osso a favore della programmazione server side, anche quando sarebbe stato più agevole utilizzare javascript (per esempio per realizzare il calendario) con l'intenzione di renderlo non funzionale ma solo puramente estetico, nonostante a parer mio i computer moderni siano più inclini ad avere problemi di instabilità della rete domestica piuttosto che di efficienza computazionale che giustificherebbe la riduzione dell'utilizzo della capacità di calcolo del client a favore di quella del server.

7. Usabilità

Per quanto riguarda l'usabilità in fase di realizzazione del progetto mi sono posto sempre in relazione alle cosiddette *5W* derivanti dal giornalismo: (in seguito aggiungere sezione estetica e responsive-mobile)

1. **Who:** Il target del sito sono curiosi e appassionati di calcio in cerca di informazioni specifiche, o solamente di farsi un giro in cerca di curiosità nascoste, per questo ho cercato di rendere le pagine e la grande mole di dati a disposizione coerenti e coese, in modo tale che fossero esplorabili agilmente senza perdere l'orientamento e l'interesse.
2. **What:** Il contenuto del sito sono dati e statistiche appartenenti al mondo del calcio, ciò è subito evidente ai visitatori per via dei colori che richiamano l'erba ed il pallone, e per il chiaro simbolismo che suggerisce l'argomento trattato.
3. **Where:** Sia le immagini presenti nella *home page* che il nome stesso del sito suggeriscono che il contesto di interesse delle manifestazioni sportive è esclusivamente quello europeo, se vogliamo interpretare "where" come luogo fisico dove collocare i contenuti del sito, se invece pensiamo a dove possano essere fisicamente presenti i vari fruitori del servizio, per esempio in treno, o seduti sul divano, è stata pensata anche una versione mobile delle pagine web, raggiungibili dallo smartphone o per mezzo di un tablet, non necessariamente davanti al monitor di un computer desktop.
4. **When:** Considerato il contenuto ricreativo e hobbistico del sito, ho ipotizzato un'affluenza principalmente negli orari non lavorativi, e nei weekend. Quindi ho cercato di rendere il sito più accattivante e meno noioso possibile, aggiungendo molte immagini, colori accesi, volti per i giocatori e simboli per le squadre, per fare in modo che gli utenti non perdessero in fretta interesse se messi di fronte solo ed esclusivamente ai dati, statici e poco vitali.
5. **Why:** Il motivo per cui un visitatore potrebbe essere interessato al mio sito web è principalmente per reperire informazioni su una determinata squadra, o un determinato giocatore, per questo motivo ho cercato di rendere il più semplice possibile trovare delle informazioni specifiche o determinati elementi, mostrando i dati con una strategia top-down, dal generale al particolare, ed organizzando le tabelle in maniera sistematica, dotandole di controlli particolari che permettono di estrarre informazioni in maniera più agevole.

7.1 Struttura Grafica

Il lato estetico è stato pensato in modo tale da aderire al *material design* che va molto in voga oggi, specialmente in ambito mobile. E' uno stile molto asciutto e asettico, dai contorni squadrati, colori opachi e molto in risalto, che danno all'utente una sensazione di ordine, di solidità e di organizzazione delle informazioni. I colori principali sono il verde, il grigio ed il bianco, che sono stati utilizzati in tutte le pagine del sito. La navbar sempre a disposizione dell'utente che lo segue

durante lo scrolling, il footer dello stesso colore distintivo, che racchiudono i contenuti delle pagine, danno un senso di contenimento e di circoscrizione alle informazioni. Ogni tabella segue lo styling generale scelto per l'estetica del progetto, con il testo degli header bianco su sfondo verde, come per la barra di navigazione. Per le icone sparse per il sito, ed in particolare quelle relative agli eventi presenti nella pagina delle partite, si è scelto sempre il material design, che trovo elegante ed asciutto, ed in linea con il resto dei contenuti. Le icone sono state prese da *icons8*⁶.

7.2 Responsive Design

Fin dall'inizio della progettazione estetica tramite fogli di stile si sono scelte le strategie che permettessero di rendere le pagine il più *responsive* possibile, per rendere in seguito più semplice l'integrazione della parte mobile, utilizzando un unico *spreadsheet* contenente delle *mediaquery* che considerassero diverse dimensioni dello schermo e redistribuissero gli elementi all'interno delle pagine. Ho fatto un uso massiccio del modulo di layout denominato *flexbox*⁷, che permette di organizzare in maniera flessibile e scalabile (sia da un punto di vista delle dimensioni degli elementi che delle dimensioni degli spazi vuoti) flussi di elementi che possono essere *wrapped*, cioè andare a capo, sia in direzione verticale che orizzontale. L'ho trovato molto utile, per non dire fondamentale, nel realizzare pagine responsive, sia desktop che mobile.

⁶ <https://icons8.it>

⁷ <https://css-tricks.com/snippets/css/a-guide-to-flexbox>

8. Test

Per la fase più estensiva e sistematica di testing mi sono appoggiato al servizio *BrowserStack*⁸ che permette di debuggare il proprio sito su una moltitudine di browser, device, e sistemi operativi differenti. Riporto in seguito un riassunto delle considerazioni che ho potuto fare sulle piattaforme più diffuse utilizzate per accedere al web.

8.1 Desktop

8.1.1 Windows 10

Edge 17:

- Le ombre dietro alcune tabelle non sono visibili
- Una volta loggati l'icona dell'utente viene visualizzata come se fosse inline-block e questo fa occupare più spazio del necessario agli elementi della navbar
- Per il resto tutto funzionante

Chrome 68:

- Tutto perfettamente funzionante

Firefox 61:

- Tutto perfettamente funzionante

8.1.2 OpenSUSE Tumbleweed (piattaforma di sviluppo)

Firefox Quantum:

- Tutto perfettamente funzionante

Chromium:

- Tutto perfettamente funzionante

8.1.3 OS X 10.9 Mavericks

Safari 7.1:

- Flexbox non supportata, difficile lettura delle informazioni in alcune sezioni del sito

⁸ <https://www.browserstack.com>

8.2 Mobile

8.2.1 Windows Phone

IE Mobile (Nokia Lumia 930):

- Versione mobile funzionante e accessibile come previsto

8.2.2 Apple IOS

Safari (I-Phone 6S):

- Contenuto che causa overflow orizzontalmente e allarga le dimensioni della pagina, problema dovuto alla scorretta implementazione di Flexbox da parte del browser

Firefox (I-Phone 6S):

- Stesso problema presente in safari

Chrome (I-Phone 6S):

- Stesso problema presente in firefox

8.2.3 Android (e custom rom derivate)

Chrome (Nexus 6):

- Versione mobile funzionante e accessibile come previsto

Firefox (Nexus 6):

- Versione mobile funzionante e accessibile come previsto

Chrome (Oneplus 5T):

- Versione mobile funzionante e accessibile come previsto

Firefox (Oneplus 5T):

- Versione mobile funzionante e accessibile come previsto