



UNIVERSITÀ DI PISA

SCUOLA DI INGEGNERIA
Dipartimento Ingegneria dell'Informazione
Corso LARGE-SCALE AND MULTI-STRUCTURED DATABASES
MSc in ARTIFICIAL INTELLIGENCE AND DATA ENGINEERING

REPORT TASK 2

Designing and Implementing an application
which interacts with a document Database

PRESENTATO DA:

CARUSO Alberto
TUMMINELLI Gianluca
Di NOIA Antonio
CALABRESE Pietro

Anno Accademico
2019/2020

Indice

1	Introduzione	1
2	Requisiti	2
2.1	Requisiti Funzionali	2
2.2	Requisiti non Funzionali	3
3	Analisi del sistema	5
3.1	Identificazione degli attori	5
3.2	Casi d'uso	5
3.3	Diagramma Classi	7
3.4	Dataset	7
3.5	Data Model	8
3.5.1	Gestione Utenti	8
3.5.2	Gestione Squadre e Liste	9
3.5.3	Gestione Calciatori	10
3.5.4	Gestione Statistiche	11
4	Statistiche e Analytics	12
5	Implementazione	13
5.1	Aggiornamento Database	14
6	Aggregazioni	15
6.1	Indici	16
6.1.1	Ricerca utente	17
6.1.2	Ricerca calciatori	18
6.1.3	Generazione statistica grafico a torta	18
7	Conclusioni	19
8	Manuale d'Uso	20

1 Introduzione

Oggi giorno, in ogni campo economico, risulta essere importante l'immediatezza delle transazioni connessa a uno scambio, quasi in tempo reale, dei dati e opinioni tra i componenti dei un team di lavoro. Queste prerogative sono ancora più evidenti nel mondo del calcio, inteso nel suo lato gestionale. Per una società, la tempestività nell'individuazione di un possibile "affare di mercato", risulta essere una fonte di guadagno non solo nel breve termine, ma anche nel lungo periodo. Inoltre la presenza di tutti i membri coinvolti nelle decisioni tecniche e gestionali della società all'interno di un'unica piattaforma, fa sì che tutti possano lavorare indipendentemente dagli altri colleghi senza intaccare il processo decisionale, anzi velocizzano quest'ultimo. Lo scopo di questa applicazione è fornire supporto alle decisioni di mercato di una società di calcio, nel caso specifico fornisce uno strumento di ricerca dei profili dei calciatori militanti anche nelle serie minori e permette di coordinare gli attori coinvolti nelle scelte di mercato tramite la creazione di una lista dei profili da tenere in considerazione. Tutto questo viene supportato da informazioni dettagliate ed accurate che le società trovano a disposizione sulla piattaforma. Le informazioni che si possono ottenere sui calciatori riguardano gran parte della storia calcistica del profilo dell'atleta:

- Le squadre nelle quali ha militato
- Le presenze annuali
- I gol fatti o subiti
- I cartellini ricevuti
- I minuti giocati
- Il valore di mercato anno per anno
- Il ruolo

e molto altro

2 Requisiti

2.1 Requisiti Funzionali

Requisiti per **l'utente generico**:

- Il sistema deve permettere ad un utente già registrato di poter effettuare il login attraverso la propria e-mail e la propria password.
- Se l'utente non è registrato, il sistema deve permettere all'utente di potersi registrare. A tale scopo l'utente deve inserire un insieme di informazioni base che includono una e-mail ed una password personali.
- Il sistema deve permettere ad un utente di poter effettuare il logout per poter uscire dal proprio profilo.

Requisiti per **l'amministratore di sistema**:

- Il sistema deve permettere all'amministratore di sistema di poter aggiornare la parte di database relativa ai calciatori.
- Il sistema deve permettere all'amministratore di sistema di poter modificare le informazioni degli utenti e poter cancellare tali utenti.
- Il sistema può modificare l'utente che occupa la posizione di amministratore di squadra di un dato team.

Requisiti per **l'amministratore di squadra**:

- Il sistema deve permettere all'amministratore di squadra di poter aggiungere/eliminare i profili dell'amministratore delegato, dell'allenatore e dell'osservatore relativi alla squadra di cui esso stesso è l'amministratore.

Requisiti per **l'allenatore**:

- Il sistema deve permettere all'allenatore di poter ricercare i calciatori da inserire nella lista dei calciatori da prendere in considerazione. Tale ricerca deve poter essere effettuata o direttamente tramite il nome ed il cognome del calciatore, oppure tramite una serie di filtri.
- Il sistema deve permettere all'allenatore di poter creare una lista relativa alle caratteristiche dei calciatori da lui preferite. Una volta creata tale lista, l'allenatore deve avere la possibilità di modificarla.

-
- Il sistema deve permettere all'allenatore di poter visualizzare entrambe le liste e di poterle aggiornare, inserendo e/o rimuovendo nuovi calciatori e/o insiemi di caratteristiche.

Requisiti per l'osservatore:

- Il sistema deve permettere all'osservatore di poter ricercare i calciatori da inserire nella lista dei calciatori da prendere in considerazione relativa alla squadra a cui esso appartiene. Tale ricerca deve poter essere effettuata o direttamente tramite il nome ed il cognome del calciatore, oppure tramite una serie di filtri.
- Il sistema deve permettere all'osservatore di poter aggiungere un calciatore ricercato alla lista dei calciatori da prendere in considerazione.
- Il sistema deve permettere all'osservatore di poter visualizzare la lista dei calciatori da prendere in considerazione e aggiornarla aggiungendo un proprio commento con relativo punteggio su di un calciatore da lui visionato.
- Il sistema deve permettere all'osservatore di poter visualizzare la lista dei profili di interesse creata dall'allenatore, tramite la quale potrà consigliare dei calciatori all'allenatore inserendoli nella lista dei calciatori da prendere in considerazione.

Requisiti per l'amministratore delegato:

- Il sistema deve permettere all'amministratore delegato di poter visualizzare la lista dei giocatori di interesse relativa alla squadra a cui esso appartiene. L'amministratore delegato può aggiornare tale lista confermando o negando la possibilità di trattare uno o più giocatori presenti nella lista, eventualmente rimuovendo un calciatore dalla suddetta lista.
- Il sistema deve permettere all'amministratore delegato di poter visualizzare i profili relativi ai calciatori presenti nella lista della squadra a cui esso appartiene.

2.2 Requisiti non Funzionali

Sono stati individuati i seguenti requisiti non funzionali, al fine di rendere le prestazioni del sistema allineate con quelle di una buona operatività:

FRUIBILITÀ: Le operazioni necessarie per completare un compito devono essere facili da capire per un generico utente.

MANUTENIBILITÀ: Il codice deve essere leggibile e modulato per permettere l'aggiunta di nuove caratteristiche e l'individuazione di eventuali errori in modo facile.

SICUREZZA: Il software deve essere usato solo come previsto. Alcuni tipi di requisiti di privacy includono: crittografia dei dati per le tabelle del database. I requisiti di accesso definiscono i tipi / gruppi di account e i loro diritti di accesso. Un esempio di requisiti di accesso potrebbe essere quello di limitare ciascun account a un accesso alla volta o di limitare la distribuzione o l'utilizzo dell'applicazione.

PRESTAZIONE: Il software deve rispondere con minore latenza possibile.

SCALABILITÀ: Il software deve essere in grado di gestire una grande quantità di transazioni

Dall'analisi condotta sui requisiti e sulle funzionalità esposte dal sistema, si è notato che il numero di accessi in lettura risulta essere molto più elevato degli accessi in scrittura, perciò i requisiti di fruibilità e scalabilità, sono stati implementati prediligendo le caratteristiche di disponibilità e partition tolerance, attraverso la manipolazione delle preferenze di read di MongoDB settando la possibilità di effettuare le letture anche sui nodi secondari. Questo fa sì che ci possa essere un qualche tipo di inconsistenza sui dati letti, in parte queste è risolto attraverso la proprietà di MongoDB write concern, che in questo caso viene settata a majority per garantire la scrittura sulla maggior parte dei nodi prima di inviare l'acknowledgement per l'operazione.

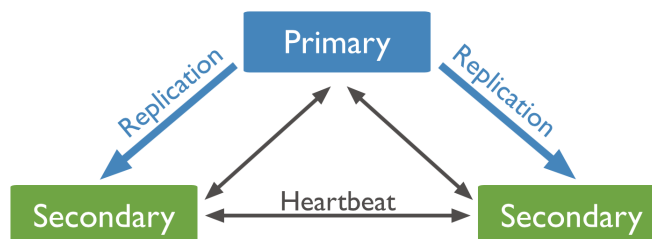


Figura 1: Replica Set

Inoltre per evitare una crescita esponenziale dei dati salvati nel sistema, con informazioni non più valide, è stato previsto che i report degli osservatori vengono automaticamente eliminati dopo un anno di permanenza nel database, in quanto le caratteristiche tecniche di un calciatore vengono ritenute non più valide dopo il passare di un anno.

3 Analisi del sistema

In questa sezione si analizza in modo approfondito le scelte tecniche, i requisiti, le funzionalità e gli attori del progetto.

3.1 Identificazione degli attori

In questa sezione, ci soffermiamo nel definire gli attori che sono stati identificati, e che possono interagire con il sistema.

Durante l'analisi dei componenti del team di gestione della società è stata individuata la squadra di mercato composta da: amministratore delegato, allenatore e osservatore. Questi risultano essere i principali utenti dell'applicazione, a cui vanno aggiunti gli amministratori per la gestione del sistema individuati nelle figure dell'amministratore di squadra e dell'amministratore di sistema.

Allenatore ha la possibilità di ricercare i profili dei calciatori in base a dei filtri come ad esempio per ruolo, per media di gol fatti, per età, ecc. ed aggiungerli alla lista dei giocatori da prendere in considerazione. L'allenatore può stilare una lista di caratteristiche del giocatore di cui ha bisogno. Inoltre, può accettare o meno un giocatore consigliato dall'osservatore.

Osservatore ha anch'esso la possibilità di ricercare i profili dei calciatori utilizzando gli stessi filtri dell'allenatore e inserire atleti all'interno della lista di calciatori da prendere in considerazione. Inoltre, per ogni giocatore presente in lista può stilare una scheda di valutazione finale del calciatore.

Amministratore delegato conferma o nega la possibilità di trattare un giocatore presente nella lista ed ha anche la possibilità di vedere i profili dei calciatori.

Amministratore di squadra gestisce la squadra di mercato concedendo o revocando i diritti di accesso alle rispettive funzionalità.

Amministratore di sistema concede i diritti di accesso agli amministratori di squadra ed aggiorna le informazioni contenute nel sistema.

3.2 Casi d'uso

Inizialmente ci siamo concentrati sul individuare i principali requisiti che il sistema deve soddisfare. Lo scopo principale del sistema è dare la possibilità

alla squadra di mercato di ogni società, di poter preparare una lista di giocatori preferiti, delle liste con le caratteristiche dei giocatori di cui si necessita, gestire le trattative.

Dunque le principali azioni che il sistema eseguirà sono:

- ricerca dei profili dei calciatori da parte della squadra di mercato
- aggiungere i giocatori alla lista dei preferiti
- inserire un report sul giocatore preferito da parte dell'osservatore
- negare o concedere l'autorizzazione a una trattativa da parte dell'amministratore delegato

Oltre a questo, per entrambi gli utenti si definiscono le operazioni basilari quali registrazione al servizio, login e logout. Inoltre si definisce il vincolo che email in atto di registrazione può essere usata per un unico account.

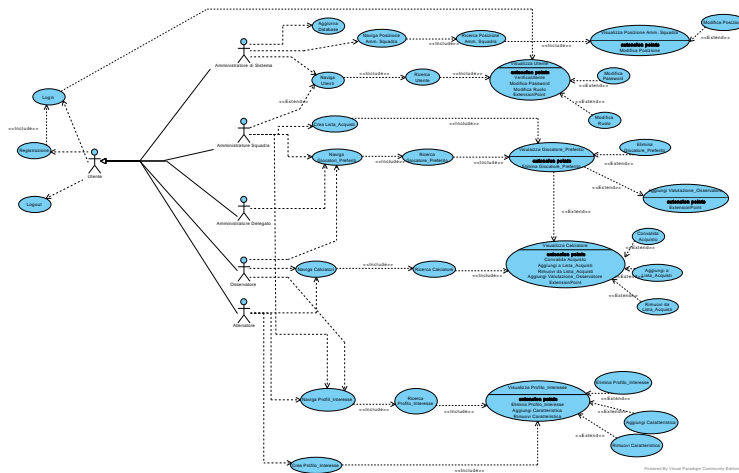


Figura 2: Use Cases Diagram

3.3 Diagramma Classi

Nella Figura3 viene mostrato il diagramma delle classi ottenuto dopo l'analisi dei requisiti e la definizione degli attori. Per favorire la leggibilità, i dettagli e le procedure di ogni singola classe sono state omesse. Infatti, l'obiettivo principale è descrivere, attraverso l'uso delle associazioni, le relazioni che intercorrono tra le classi. Per fare questo è stato adottato il formalismo definito dal UML 2.

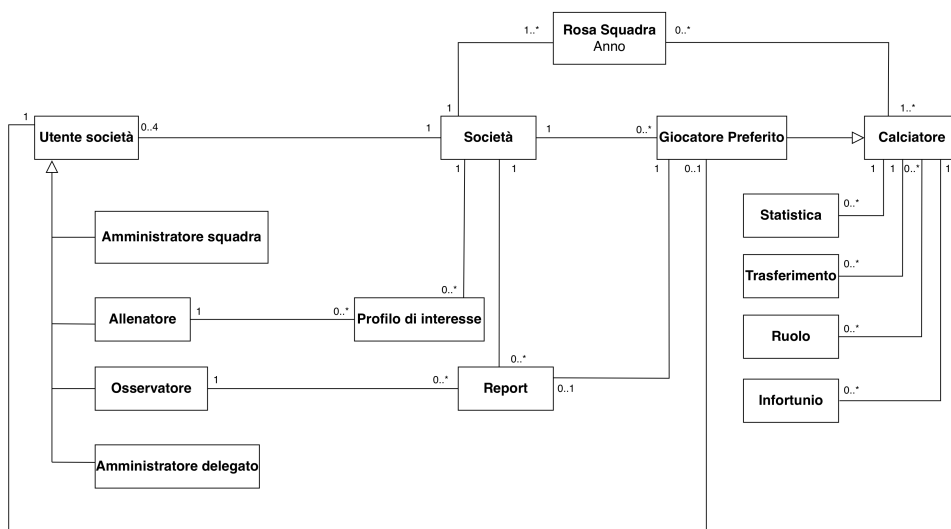


Figura 3: Class Diagram

Da notare l'uso della classe Giocatore Preferito come derivata della classe Calciatore, ciò fa sì che si possano visualizzare le informazioni più rilevanti di un calciatore inserito nella lista dei preferiti, in modo veloce, oltre che a questo possono essere associati il report osservatore ed eventualmente il profilo d'interesse a cui fa riferimento

3.4 Dataset

Il dataset per i dati relativi ai calciatori è stata sviluppata una funzione di scraping per ottenere i dati dal sito specializzato in statistiche calcistiche, www.transfermarkt.it; sono stati presi i dati relativi a tutti i giocatori sia dei

campionati principali, sia delle serie minori di tutti i continenti. Si è ottenuto così un dataset iniziale di circa 300 Mb, che contiene 40833 calciatori, con tutte le loro statistiche.

I dati così ottenuti organizzati in un file JSON, sono stati importati all'interno dell'applicazione e organizzati secondo i Data Model definiti nella sezione seguente.

3.5 Data Model

Visto quanto riportato nelle sezioni precedenti, tenuto conto dei requisiti del sistema e dei dati che si andranno a gestire, si sono progettati i data model come segue.

3.5.1 Gestione Utenti

```
_id: ObjectId("5e499745ba76bc72a219bcd")
nome: "tre"
cognome: "tre"
email: "tre"
password: "tre"
ruolo: "admin"

_id: ObjectId("5e8a1247caaa4e29441f761b")
nome: "aaa"
cognome: "aaa"
email: "aaa"
password: "aaa"
ruolo: "allenatore"
societa: "S06"

_id: ObjectId("5e8a1253caaa4e29441f761c")
nome: "sss"
cognome: "sss"
email: "sss"
password: "sss"
ruolo: "osservatore"
societa: "S06"

_id: ObjectId("5e8a125dcaaa4e29441f761d")
nome: "ddd"
cognome: "ddd"
email: "ddd"
password: "ddd"
ruolo: "amministratore delegato"
societa: "S06"

_id: ObjectId("5e8a1267caaa4e29441f761e")
nome: "fff"
cognome: "fff"
email: "fff"
password: "fff"
ruolo: "amministratore di squadra"
societa: "S06"
```

Figura 4: xml Utenti data model

3.5.2 Gestione Squadre e Liste

```
_id: "506"
nomeSocieta: "Juventus FC"
nazione: "Italia"
lega: "SERIE A"
linkLogo: "https://tmssl.akamaized.net/images/wappen/tiny/506.png?lm=1539869052"
amministratore di squadra: ObjectId("5e8a1267caaa4e29441f761e")
allenatore: ObjectId("5e8a1247caaa4e29441f761b")
osservatore: ObjectId("5e8a1253caaa4e29441f761c")
amministratore delegato: ObjectId("5e8a125dcaaa4e29441f761d")
giocatoriPreferiti: Array
  0: Object
    _id: "40423"
    timeStampAggiunto: 2020-04-14T00:01:53.055+00:00
    linkFoto: "https://tmssl.akamaized.net/images/portrait/header/40423-1471339256.j..."
    nome: "Claudio Bravo"
    nazionalita: "Cile"
    squadra: "Manchester City"
    posizione: "Portiere"
    valoreMercato: 1300000
    dataNascita: 1983-04-13T00:00:00.000+00:00
    propostoDa: "allenatore"
    giudizioAllenatore: 1
    giudizioDirigenza: 2
  1: Object
  2: Object
  listaProfiliDiInteresse: Array
    0: Object
      _id: "1"
      idAllenatore: "5e8a1247caaa4e29441f761b"
      ruolo: "Mediano"
      etaMinima: 15
      etaMassima: 34
      descrizioneCaratteristiche: "Giovane e veloce"
      timeStampAggiunto: 2020-05-27T09:55:36.708+00:00
```

Figura 5: xml Gestione Squadra e Liste data model

All'interno troviamo un array con giocatori preferiti: dove oltre ai dati per la gestione della compravendita sono riportati alcuni dati del giocatore embedded per una più facile fruizione.

3.5.3 Gestione Calciatori

```
_id: "392085"
nome: "Nicolò Zaniolo"
dataNascita: 930873600000
luogoNascita: "Massa"
altezza: 190
nazionalita: "Italia"
piede: "sinistro"
procuratore: "Vigo Global Sport Services Srl"
squadra: "AS Roma"
linkFoto: "https://tmssl.akamaized.net/images/portrait/header/392085-1550050645...."
inRosaDa: 1530403200000
scadenza: 1719705600000
posizionePrincipale: "Trequartista"
valoreAttuale: 50000000
~ altriRuoli: Array
  0: "Centrale"
  1: "Ala destra"
~ trasferimento: Array
  ~ 0: Object
    stagione: "18/19"
    data: 1530403200000
    venditore: "Inter U19"
    acquirente: "Roma"
    valoreDiMercato: 1000000
    riscatto: "4,50 mln €"
  > 1: Object
  > 2: Object
  > 3: Object
  > 4: Object
~ statistiche: Array
  ~ 0: Object
    stagione: "19/20"
    competizione: "Serie A"
    societa: "AS Roma"
    presenze: 18
    puntiPartita: 1.78
    reti: 4
    assist: 1
    ammonizioni: 6
    doppieAmmonizioni: 0
    espulsioni: 0
    minutiGiocati: 1295
  > 1: Object
  > 2: Object
  > 3: Object
  > 4: Object
  > 5: Object
  > 6: Object
  > 7: Object
  > 8: Object
  > 9: Object
  > 10: Object
  > 11: Object
  > 12: Object
  > 13: Object
  > 14: Object
  > 15: Object
  > 16: Object
~ infortunio: Array
  ~ 0: Object
    stagione: "19/20"
    tipoInfortunio: "Rottura del legamento crociato"
    da: 1578873600000
    al: 1598832000000
    giorni: 231
    partitePerse: 23
  > 1: Object
```

Figura 6: xml Gestione Calciatori data model

Nella strutturazione del database si è deciso di inserire i documenti relativi a statistiche, trasferimenti e infortuni all'interno del documento del calciatore. La scelta di inserire questi documenti embedded è stata dettata dalla necessità di dover utilizzare alcuni degli attributi di questi documenti per effettuare delle ricerche dei calciatori in base a dei filtri calcolati a partire dai suddetti attributi. Creare delle apposite collection per uno o più di questi documenti

implicherebbe infatti un eccessivo overhead al momento del calcolo dei filtri di ricerca sopra menzionati in quanto sarebbero necessarie diverse operazioni di join tra documenti appartenenti a collection diverse.

3.5.4 Gestione Statistiche

```
_id: ObjectId("5e59188de46a28f99de1a9af")
calciatore: "40204"
posizionePrincipale: "Portiere"
stagione: "06/07"
competizione: "Premier League"
societa: "Manchester City"
presenze: 1
puntiPartita: 1
reti: 0
assist: undefined
ammonizione: 0
doppieAmmonizioni: 0
espulsioni: 0
minutiGiocati: 90
partiteNoGoal: 1
retiSubite: 0

_id: ObjectId("5e5918dee46a28f99de2bb36")
calciatore: "47082"
posizionePrincipale: "Punta centrale"
stagione: "06/07"
competizione: "Premier League"
societa: "Manchester City"
presenze: 2
puntiPartita: 0
reti: 0
assist: 0
ammonizione: 0
doppieAmmonizioni: 0
espulsioni: 0
minutiGiocati: 19
partiteNoGoal: undefined
retiSubite: undefined
```

Figura 7: xml Gestione Statistiche data model

Per quanto riguarda le statistiche dei calciatori, è stato deciso di creare una collection di documenti così fatti:

La scelta di creare una collection separata per le statistiche è stata presa per velocizzare la funzionalità offerta dall'applicazione di comparazione delle statistiche di un calciatore rispetto agli altri giocatori del suo stesso ruolo e rispetto ai giocatori appartenenti alla propria squadra. Per poter fornire questa funzionalità non è infatti necessario dover eseguire alcuna operazione di join ma solamente operazioni di aggregazione in base a competizione e squadra. Per l'accesso alle statistiche del calciatore preso in considerazione risulta invece più veloce accedere ai documenti embedded mostrati nella sezione precedente così da evitare operazioni di join.

4 Statistiche e Analytics

Nell'applicazione sono implementate e calcolate alcune statistiche e analytics che permetteranno agli allenatori e osservatori di avere più informazioni dettagliate sul calciatore e sulla sua carriera.

Le statistiche che verranno implementate per ciascun giocatore sono:

- Media giorni infortuni (portiere e giocatori di movimento)
- Media goal stagionali (giocatori di movimento)
- Media assist stagionali (giocatori di movimento)
- Media goal subiti (portiere)

Queste statistiche verranno utilizzate inoltre dagli allenatori e dagli osservatori come filtri per una ricerca più dettagliata di giocatori oltre ai filtri di base quali:

- Nome, Cognome
- Campionato
- Squadra
- Posizione
- Valore di mercato
- Altezza
- Stato Contratto
- Età

Bisogna sottolineare che il filtro “Posizione” è usato come filtro principale, poi è stato notato, durante l'analisi, che la ricerca prende come primo valore la posizione del giocatore e che poche volte questa viene effettuata basandosi su altri filtri per fare una prima scrematura del result-set.

Le analytics che verranno implementate nella pagina specifica di un calciatore sono invece le seguenti:

1. Istogramma $NUMERO_{goal} * NUMERO_{calciatori}$ filtrato o per stagione o/e competizione e ruolo e piazzamento in esso del giocatore scelto

-
2. Istogramma $NUMERO_{cartellini} * NUMERO_{calciatori}$
 3. Istogramma $NUMERO_{assist} * NUMERO_{calciatori}$
 4. Istogramma $NUMERO_{goalSubiti} * NUMERO_{calciatori}$
 5. Diagramma a torta del rendimento rispetto alla squadra per i valori ai punti 1, 2, 3, 4

5 Implementazione

In questa sezione, si descrive, in modo più dettagliato, le scelte implementative adottate per lo sviluppo del sistema. Attualmente si tratta di una java app, basata su architettura Client-Server, il prototipo del sistema è stato sviluppato attenendosi all'architettura multi-tier, questo per tenere separati il più possibile il lato Client da quello Server.

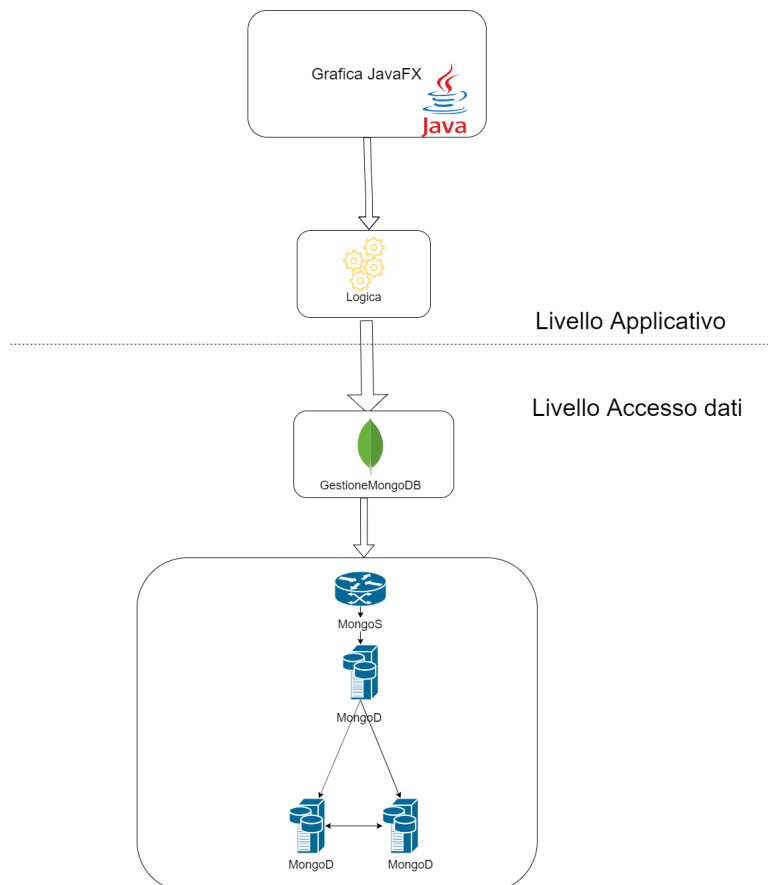


Figura 8: Multi-Tier Architecture Design

Come mostrato nella Figura 8, il lato Client è gestito con java application che si interfaccia con il lato server attraverso l'uso del framework mongoDb-driver-sync. Mentre la grafica viene gestita attraverso il framework JavaFx. Per lo sviluppo della parte di Back End si è scelto di lavorare con dei document database, nel caso specifico MongoDB, vista la buona affidabilità e la grande flessibilità nella gestione dei data model. Per assicurare la consistenza dei dati all'interno del DB, garantendo allo stesso tempo la concorrenza, il sistema è stato realizzato adottando il pattern architetturale MVC. Questo tipo di approccio risulta essere il metodo di controllo della concorrenza comunemente utilizzato dai sistemi di gestione per garantire accesso simultaneo al database e nei linguaggi di programmazione per implementare la memoria transazionale, senza questo ci potremo trovare di fronte a casi di inconsistenza dei dati all'interno del database.

5.1 Aggiornamento Database

In questa sezione è riportata la funzione di aggiornamento del sistema, come è facile intuire, vista la ridondanza introdotta ai dati attraverso l'uso di varie repliche embedded, la gestione dell'operazione è deputata a una Mongo Transaction, che ha cura di aggiornare tutte le repliche prima di terminare l'operazione con successo, facendo sì che i dati siano consistenti. Da notare le **righe da 6** in poi.

```
3 public class aggiornaDatabase extends MongoDataAccess{
4     public static int aggiorna(String calciatore){
5         Document calciatoreDoc = Document.parse(calciatore);
6         String id=calciatoreDoc.getString("_id");
7         ClientSession clientSession=mongoClient.startSession();
8         TransactionBody txnBody = (TransactionBody<String>) () -> {
9             Bson filter = eq("_id", id);
10            Bson filter1 = eq("calciatore", id);
11            collectionCalciatore.deleteMany(clientSession, filter);
12            collectionStatistiche.deleteMany(clientSession, filter1);
13            collectionCalciatore.insertOne(clientSession, calciatoreDoc);
14
15            if(calciatoreDoc.get("statistiche") != null){
16                List<Document> statistiche =
17                    ↳ (List<Document>)calciatoreDoc.get("statistiche");
18                List<Document> auxs=new ArrayList();
19                for(int i=0;i<statistiche.size();i++){
20                    Document aux=new Document();
21                    aux.append("calciatore", calciatoreDoc.getString("_id"));
22                    aux.append("posizionePrincipale",
23                        ↳ calciatoreDoc.getString("posizionePrincipale"));
24                    aux.append("stagione", statistiche.get(i).getString("stagione"));
25                    aux.append("competizione",
26                        ↳ statistiche.get(i).getString("competizione"));
27                    aux.append("societa", statistiche.get(i).getString("societa"));
28                    aux.append("presenze",
29                        ↳ statistiche.get(i).getInteger("presenze").doubleValue());
30                try {
```

```

27         aux.append("puntiPartita",
28             ↳ statistiche.get(i).getInteger("puntiPartita").doubleValue());
29     } catch (Exception e) {
30         aux.append("puntiPartita",
31             ↳ statistiche.get(i).getDouble("puntiPartita"));
32     }
33
34     if(!calciatoreDoc.getString("posizionePrincipale").equals("Portiere")){
35         aux.append("assist",
36             ↳ statistiche.get(i).getInteger("assist").doubleValue());
37         aux.append("reti",
38             ↳ statistiche.get(i).getInteger("reti").doubleValue());
39     } else{
40         aux.append("partiteNoGoal",
41             ↳ statistiche.get(i).getInteger("partiteNoGol").doubleValue());
42         aux.append("retiSubite",
43             ↳ statistiche.get(i).getInteger("retiSubite").doubleValue());
44     }
45
46     aux.append("doppieAmmonizioni",
47         ↳ statistiche.get(i).getInteger("doppieAmmonizioni").doubleValue());
48     aux.append("ammonizione",
49         ↳ statistiche.get(i).getInteger("ammonizioni").doubleValue());
50     aux.append("espulsioni",
51         ↳ statistiche.get(i).getInteger("espulsioni").doubleValue());
52     aux.append("minutiGiocati",
53         ↳ statistiche.get(i).getInteger("minutiGiocati").doubleValue());
54
55     auxs.add(aux);
56     }
57     collectionStatistiche.insertMany(clientSession, auxs);
58 }
59 aggiornaGiocatorePreferito(clientSession, calciatoreDoc);
60
61     return "aggiorna campi!";
62 };
63
64 try {
65     clientSession.withTransaction(txnBody);
66 } catch (RuntimeException e) {
67     e.printStackTrace();
68     return 1;
69 } finally {
70     clientSession.close();
71 }
72
73 return 0;
74 }

```

6 Aggregazioni

All'interno del progetto sono state usate in modo estensivo le aggregazioni per abilitare tutte le funzionalità previste, sia per la ricerca che per la raccolta dei dati per generare i grafici.

```

67 //...
68

```

```

69 List<InformazioniRicercaCalciatore> infos=new ArrayList<>();
70
71     try {
72         MongoClient<Document> cur;
73         if(filtroMatch1.isEmpty() && filtroMatch2.isEmpty())
74             return null;
75         else if(!filtroMatch1.isEmpty() && filtroMatch2.isEmpty()){
76             cur=collectionCalciatore.aggregate(
77                 Arrays.asList(new Document("$match",filtroMatch))
78             ).iterator();
79         } else if(filtroMatch1.isEmpty() && !filtroMatch2.isEmpty()){
80             cur=collectionStatistiche.aggregate(
81                 Arrays.asList(
82                     statisticheGroupSum,
83                     statisticheGroupAvg,
84                     new Document("$match",filtroMatch2),
85                     lookupDoc)
86                 ).allowDiskUse(Boolean.TRUE).iterator();
87             System.out.println(statisticheGroupSum.toJson());
88             System.out.println(statisticheGroupAvg.toJson());
89             System.out.println((new Document("$match",filtroMatch2).toJson()));
90             System.out.println(lookupDoc.toJson());
91         } else{
92             cur=collectionStatistiche.aggregate(
93                 Arrays.asList(
94                     statisticheGroupSum,
95                     statisticheGroupAvg,
96                     new Document("$match",filtroMatch2),
97                     lookupDoc,
98                     new Document("$unwind",unwind),
99                     new Document("$match",filtroMatch1))
100                 ).allowDiskUse(Boolean.TRUE).iterator();
101         }
102         int cont=0;
103
104         //...

```

In particolare è stata riportata l'aggregazione ritenuta più significativa per il progetto, quella che serve la funzione di ricerca dei calciatori in base a dei filtri definiti dall'utente. In base ai filtri definiti, vengono attivate aggregazioni diverse con si vede nelle righe **76** e **80**

6.1 Indici

Un indice in MongoDB è una struttura dati speciale che contiene i dati di pochi campi dei documenti su cui viene creato l'indice. Gli indici migliorano la velocità delle operazioni di ricerca nel database perché invece di cercare l'intero documento, la ricerca viene eseguita sugli indici che contengono solo pochi campi. D'altro canto, avere troppi indici può ostacolare le prestazioni delle operazioni di inserimento, aggiornamento ed eliminazione a causa dello spazio aggiuntivo di scrittura e dati utilizzato dagli indici.

Per i motivi sopra descritti sono stati individuati e successivamente testati, tre indici che intervengono sulle operazioni con maggiore frequenza o che lavorano su collection molto estese, al fine di migliorare i tempi di risposta in lettura del database.

6.1.1 Ricerca utente

Durante l'operazione di scraping sono state individuate circa 5.000 società, e considerando la presenza di quattro utenti per ogni società fa sì che nel database siano presenti circa 20.000 utenti, il che rende l'operazione di login molto onerosa dal punto di vista del tempo d'accesso; per cui si è deciso di inserire un indice su campo "email" che è quello per individuare l'utente al login.

Di seguito sono riportati i risultati dei test effettuati.

```
"executionStats" : {
  "executionSuccess" : true,
  "nReturned" : 1,
  "executionTimeMillis" : 3,
  "totalKeysExamined" : 2005,
  "totalDocsExamined" : 2005,
  "executionStages" : {
    "stage" : "FETCH",
    "filter" : {
      "email" : {
        "$eq" : "aaa"
      }
    }
  },
  "nReturned" : 1,
  "executionTimeMillisEstimate" : 0,
  "works" : 2006,
  "advanced" : 1,
  "needTime" : 2004,
  "needYield" : 0,
  "saveState" : 0,
  "restoreState" : 0,
  "isEOF" : 1,
  "prioritizedIndex" : null
}
```

Figura 9: Ricerca Utente senza indice

```
},
"executionStats" : {
  "executionSuccess" : true,
  "nReturned" : 1,
  "executionTimeMillis" : 0,
  "totalKeysExamined" : 1,
  "totalDocsExamined" : 1,
  "executionStages" : {
    "stage" : "FETCH",
    "nReturned" : 1,
    "executionTimeMillisEstimate" : 0,
    "works" : 2,
    "advanced" : 1,
    "needTime" : 0,
    "needYield" : 0,
    "saveState" : 0,
    "restoreState" : 0,
    "isEOF" : 1,
    "prioritizedIndex" : null
  }
}
```

Figura 10: Ricerca Utente con indice

6.1.2 Ricerca calciatori

Durante l'operazione di scraping sono state individuati circa 40.000 calciatori; dunque per rendere la ricerca di un calciatore da valutare come acquistabile significativamente più veloce, si è creato l'indice composto dai campi "Posizione" e "Valore_Attuale".

Di seguito sono riportati i risultati dei test effettuati.

```
    "executionStats" : {
      "executionSuccess" : true,
      "nReturned" : 1,
      "executionTimeMillis" : 191,
      "totalKeysExamined" : 30370,
      "totalDocsExamined" : 30370,
      "executionStages" : {
        "stage" : "FETCH",
        "filter" : {
          "$and" : [
            {
              "statistiche" : {
                "valoreAttuale" : {
```

Figura 11: Ricerca Giocatore senza indice

```
    "executionStats" : {
      "executionSuccess" : true,
      "nReturned" : 1,
      "executionTimeMillis" : 16,
      "totalKeysExamined" : 3129,
      "totalDocsExamined" : 3129,
      "executionStages" : {
        "stage" : "FETCH",
        "filter" : {
          "$and" : [
```

Figura 12: Ricerca Giocatore con indice

6.1.3 Generazione statistica grafico a torta

Per rendere più agile il calcolo delle analytics, visto che per ogni istanza dell'applicazione, è deputato al lato server e che questo deve essere svolto su un dataset di dimensioni elevate, si è creato sulla collection "Statistiche" l'indice composto dai campi "Società", "Stagione", "Calciatore".

Di seguito sono riportati i risultati dei test effettuati.

```

    },
    "executionStats" : {
      "executionSuccess" : true,
      "nReturned" : 82,
      "executionTimeMillis" : 672,
      "totalKeysExamined" : 458716,
      "totalDocsExamined" : 458716,
      "executionStages" : {
        "stage" : "FETCH",
        "filter" : {
          "$and" : [
            {
              "societa" : {
                "$eq" : "Juventus FC"
              }
            },
            {

```

Figura 13: Statistica grafico a torta senza indice

```

"executionStats" : {
  "executionSuccess" : true,
  "nReturned" : 1,
  "executionTimeMillis" : 16,
  "totalKeysExamined" : 3129,
  "totalDocsExamined" : 3129,
  "executionStages" : {
    "stage" : "FETCH",
    "filter" : {
      "$and" : [

```

Figura 14: Statistica grafico a torta con indice

7 Conclusioni

In questo progetto è stato sviluppato un software per la gestione delle società di calcio, in particolare si è voluto l'attenzione sulla progettazione e l'implementazione del lato Back End, quindi della parte legata al database, visto anche le specifiche richieste dal corso. Ciò non toglie che in caso di una reale implementazione del sistema si dovrebbe procedere ad alcune migliorie in modo da garantire la fruibilità del servizio a un largo numero di utenti, su un più vasto numero di dispositivi, quindi potrebbe essere sviluppata una web app basata su Java EE.

8 Manuale d'Uso

1. Login Page

Iscriviti ora'. There are two input fields: the first is labeled 'Email' and the second is masked with asterisks. Below the input fields is a large blue button labeled 'Login'." data-bbox="181 254 810 638"/>

Login/Registrazione

Non sei registrato? [Iscriviti ora](#)

Email

Login

1.1 IF utente non registrato

1.1.1 L'utente clicca sul link **Iscriviti ora**.

1.1.2 Il sistema caricherà la pagina di Iscrizione e passa al punto 2.1

1.2 IF utente registrato

1.2.1 L'utente inserisce l'E-mail e la password.

1.2.2 L'utente preme il pulsante Login.

1.2.3 Il sistema controlla che l'utente abbia inserito le credenziali corrette.

1.2.4 IF le credenziali sono errate

1.2.4.1 Visualizza pop-up di errore e torna al punto 1.1

1.2.5 IF le credenziali sono corrette

1.2.5.1 IF l'utente è un allenatore

1.2.5.1.1 Il sistema carica la homepage dell'allenatore.

1.2.5.1.2 Il sistema passa al punto 3.1.

1.2.5.2 IF l'utente è un osservatore

1.2.5.2.1 Il sistema carica la homepage dell'osservatore.

1.2.5.2.2 Il sistema passa al punto 4.1.

1.2.5.3 IF l'utente è un amministratore delegato.

1.2.5.3.1 Il sistema carica la homepage dell'amministratore delegato.

1.2.5.3.2 Il sistema passa al punto 5.1.

1.2.5.4 IF l'utente è un amministratore di sistema.

1.2.5.4.1 Il sistema carica l'homepage dell'amministratore di sistema.

1.2.5.4.2 Il sistema passa al punto 6.1

1.2.5.5 IF l'utente è un amministratore di squadra.

1.2.5.5.1 Il sistema carica l'homepage dell'amministratore di squadra.

1.2.5.5.2 Il sistema passa al punto 7.1.

2. Pagina di Iscrizione.

Iscrizione

Nome:

Luigi

Cognome:

Verdi

Email:

luigiverdi@mail.com

Ruolo:

Allenatore

▼

Password:

Iscrivimi

2.1 L'utente inserisce il nome, il cognome, l'E-mail, password e seleziona il suo ruolo.

2.2 L'utente schiaccia il pulsante Iscrivimi.

2.3 IF l'e-mail è già usata da un utente

2.3.1 Il sistema mostra una finestra d'errore.

2.4 IF l'e-mail non è già in uso

2.4.1 Il sistema registra l'utente e ritorna al punto 1.1.

3. Homepage Allenatore



3.1 IF l'utente schiaccia il link [logout](#)

3.1.1 Il sistema effettua il [logout](#) dell'utente e torna al punto 1.1

3.2 IF l'utente clicca sul pulsante [Cerca Giocatore](#)

3.2.1 Il sistema carica la pagina di ricerca giocatore e passa al punto 9.1.1

3.3 IF l'utente clicca sul pulsante [Visualizza lista](#)

3.3.1 Il sistema carica la pagina di visualizzazione della lista dei giocatori da tenere d'occhio e passa al punto 15.1

3.4 IF l'utente clicca sul pulsante [Aggiungi profilo di interesse](#)

3.4.1 Il sistema carica la pagina che permette di aggiungere un profilo di interesse e passa al punto 11.1.

3.5 IF l'utente clicca sul pulsante [Modifica Profilo personale](#)

3.5.1 Il sistema carica la pagina di modifica del proprio profilo e passa al punto 8.1.

4. HomePage Osservatore.



4.1 IF l'utente schiaccia il link logout

4.1.1 Il sistema effettua il logout dell'utente e torna al punto 1.1

4.2 IF l'utente clicca sul pulsante Cerca Giocatore

4.2.1 Il sistema carica la pagina di ricerca giocatore e passa al punto 9.1.1

4.3 IF l'utente clicca sul pulsante Visualizza lista

4.3.1 Il sistema carica la pagina di visualizzazione della lista dei giocatori da tenere d'occhio e passa al punto 15.1

4.4 IF l'utente clicca sul pulsante Visualizza profili di interesse

4.4.1 Il sistema carica la pagina che permette di visualizzare un profilo di interesse del proprio allenatore e passa al punto 12.1.

4.5 IF l'utente clicca sul pulsante Modifica Profilo personale

4.5.1 Il sistema carica la pagina di modifica del proprio profilo e passa al punto 8.1.

5. HomePage Amministratore Delegato.



5.1 IF l'utente schiaccia il link [logout](#)

5.1.1 Il sistema effettua il [logout](#) dell'utente e torna al punto 1.1

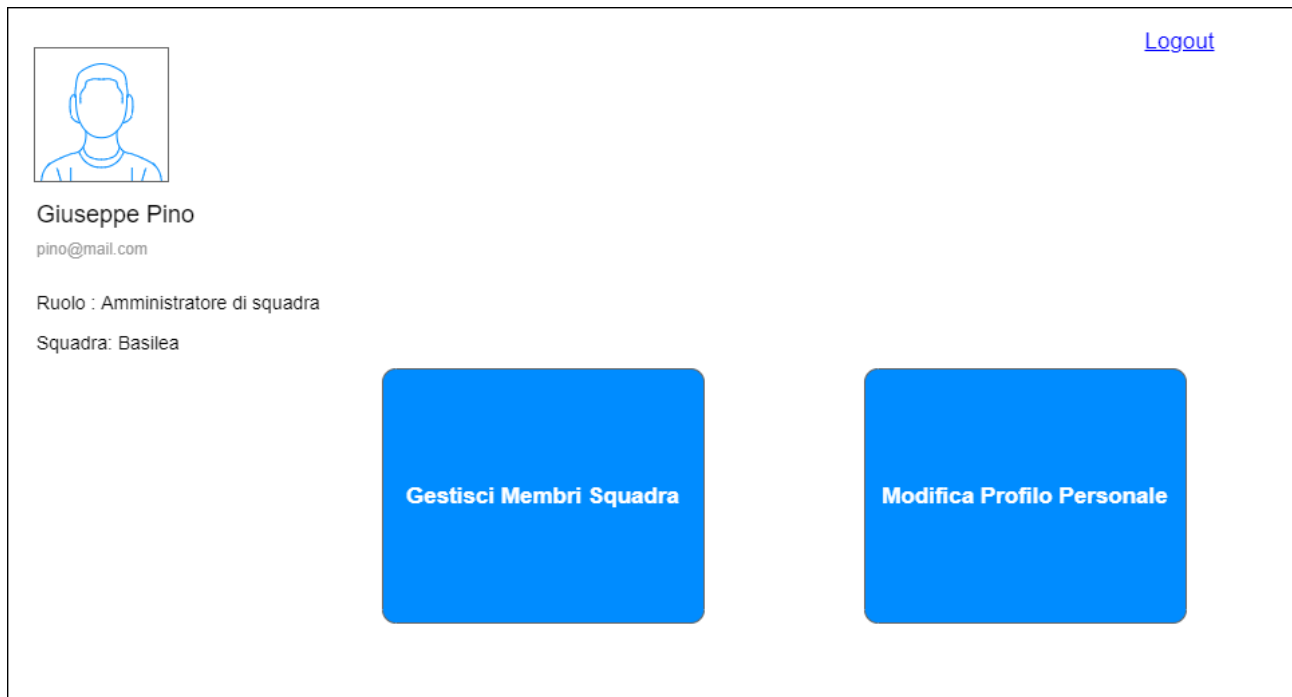
5.2 IF l'utente clicca sul pulsante [Visualizza lista giocatori di interesse](#)

5.2.1 Il sistema carica la pagina di visualizzazione della lista dei giocatori da tenere d'occhio e passa al punto 15.1

5.3 IF l'utente clicca sul pulsante [Modifica Profilo personale](#)

5.3.1 Il sistema carica la pagina di modifica del proprio profilo e passa al punto 8.1.

6. HomePage Amministratore di squadra



6.1 IF l'utente schiaccia il link logout

6.1.1 Il sistema effettua il logout dell'utente e torna al punto 1.1

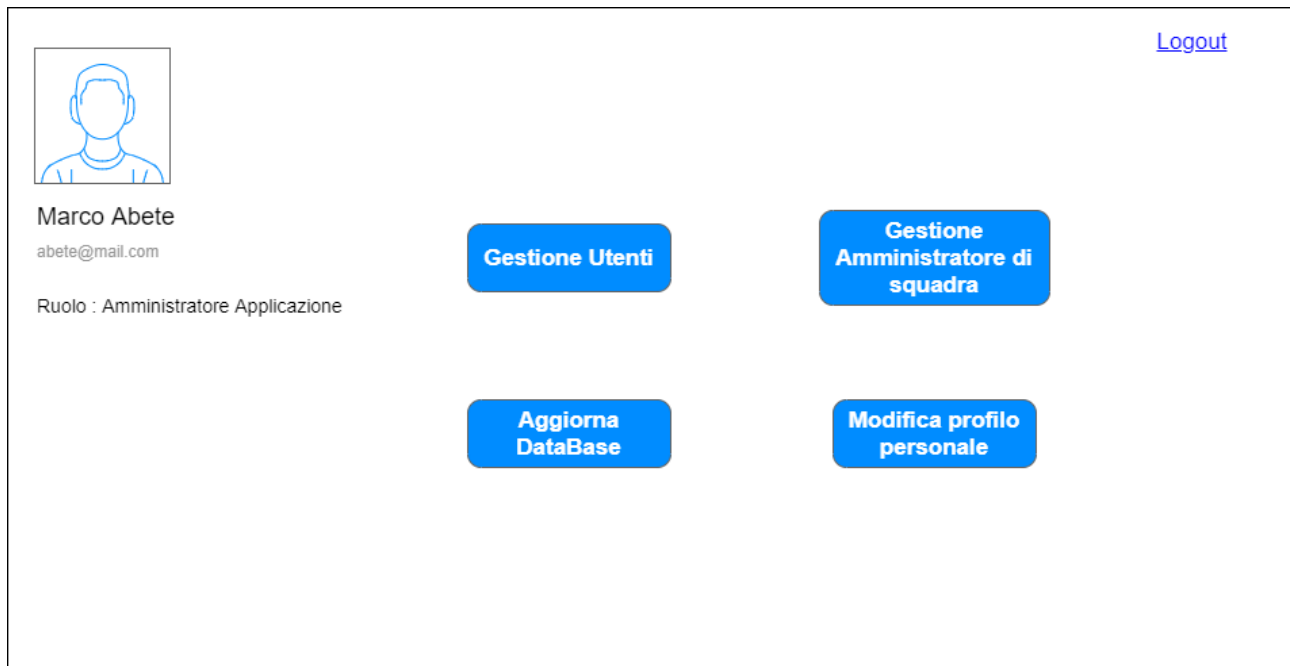
6.2 IF l'utente clicca sul pulsante Gestisci Membri Squadra

6.2.1 Il sistema carica la pagina per la gestione dei membri della squadra e passa al punto 13.1

6.3 IF l'utente clicca sul pulsante Modifica Profilo personale

6.3.1 Il sistema carica la pagina di modifica del proprio profilo e passa al punto 8.1.

7. HomePage Amministratore di sistema



7.1 IF l'utente schiaccia il link logout

7.1.1 Il sistema effettua il logout dell'utente e torna al punto 1.1

7.2 IF l'utente clicca il pulsante Gestione Utenti

7.2.1 Il sistema carica la pagina di gestione utenti e passa al punto 14.1.1

7.3 IF l'utente clicca il pulsante Gestione Amministratore di squadra.

7.3.1 Il sistema carica la pagina di gestione utenti e passa al punto 14.1.2

7.4 IF l'utente clicca il pulsante Aggiorna DataBase

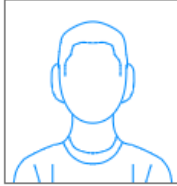

7.4.1 Il sistema aggiorna l'archivio dell'applicazione con le informazioni più aggiornate sui Giocatori.

7.5 IF l'utente clicca sul pulsante Modifica Profilo personale

7.5.1 Il sistema carica la pagina di modifica del proprio profilo e passa al punto 8.1.

8. Pagina modifica profilo personale


[Home](#) [Logout](#)

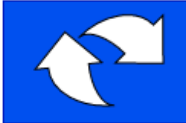


Nome: Andrea Cognome: Rossi

Email: rossi@mail.com

Password: *****





Nuova Email:

Nuova Password:

Conferma nuova Password:

[Torna Indietro](#)

8.1 IF l'utente clicca sul link Home o Torna Indietro

8.1.1 IF l'utente è un allenatore

8.1.1.1 Il sistema carica la homepage dell'allenatore.

8.1.1.2 Il sistema passa al punto 3.1.

8.1.2 IF l'utente è un osservatore

8.1.2.1 Il sistema carica la homepage dell'osservatore.

8.1.2.2 Il sistema passa al punto 4.1.

8.1.3 IF l'utente è un amministratore delegato.

8.1.3.1 Il sistema carica la homepage dell'amministratore delegato.

8.1.3.2 Il sistema passa al punto 5.1.

8.1.4 IF l'utente è un amministratore di sistema.

8.1.4.1 Il sistema carica l'homepage dell'amministratore di sistema.

8.1.4.2 Il sistema passa al punto 6.1

8.1.5 IF l'utente è un amministratore di squadra.

8.1.5.1 Il sistema carica l'homepage dell'amministratore di squadra.

8.1.5.2 Il sistema passa al punto 7.1.

8.2 IF l'utente clicca sul cestino.

8.2.1 il sistema mostra una schermata di conferma.

8.2.2 IF l'utente conferma l'eliminazione

8.2.2.1 il sistema elimina l'utente dall'archivio.

8.2.2.2 il sistema carica la pagina di login.

8.2.2.3 il sistema passa al punto 1.1.

8.2.3 IF il sistema non conferma l'eliminazione.

8.2.3.1 il sistema torna al punto 8.1.

8.3 L'utente inserisce opzionalmente la nuova e-mail, la nuova password e la conferma della password e la nuova foto di profilo.

8.4 L'utente preme il pulsante per cambiare le credenziali.

8.5 IF il campo Nuova E-mail compilato

8.5.1 IF l'e-mail è già utilizzata da un altro utente

8.5.1.2 Il sistema mostra una finestra d'errore

8.5.1.3 Il sistema torna al punto 8.1

8.6 IF i campi Nuova Password e Conferma nuova password sono compilati

8.6.1 IF i campi Nuova Password e Conferma non combaciano.

8.6.1.1 Il sistema mostra una finestra d'errore

8.6.1.2 Il sistema torna al punto 8.1

8.7 IF l'utente ha inserito la nuova immagine.

8.7.1 IF immagine non valida

8.7.1.1 Il sistema mostra una finestra d'errore.

8.7.1.2 Il sistema torna al punto 8.1

8.8 Il sistema aggiorna i dati dell'utente nell'archivio.

8.9 Il sistema torna al punto 8.1

9. Pagina ricerca Giocatore

9.1 Ricerca per nome Giocatore

Ricerca per nome Ricerca per attributo

Nome:

Cognome:

Ricerca

[Torna Indietro](#)

9.1.1 IF l'utente seleziona Ricerca per attributo

9.1.1.1 Il sistema carica la pagina per la ricerca per attributo.

9.1.1.2 Il sistema passa al punto 9.2.1.

9.1.2 IF l'utente seleziona il link Torna indietro.

9.1.2.1 il sistema carica l'homepage dell'utente e passa al punto 1.2.5.1.

9.1.3 L'utente inserisce il nome e il cognome del calciatore da cercare.

9.1.4 IF la lista dei risultati di ricerca è vuota.

9.1.4.1 Il sistema mostra una schermata d'avviso.

9.1.4.2 Il sistema torna al punto 9.1.1

9.1.5 L'utente schiaccia il pulsante Ricerca e passa al punto 10.1.

9.2 Ricerca per attributo.

Ricerca per nome

Ricerca per attributo

Campionato:

Squadra:

Posizione:

Punta Centrale

Resistenza agli infortuni:

Elevata

Disciplina in campo:

Molto disciplinato

Tendenza al goal:

Elevata

Valore di mercato:

Minimo:

100.000

Massimo:

50.000.000

Altezza (cm):

Minimo:

175

Massimo:

190

Stato Contratto:

Tutti

Fascia età:

Minimo:

16

Massimo:

35

Ricerca

[Torna indietro](#)

9.2.1 IF l'utente seleziona Ricerca per attributo

9.2.1.1 Il sistema carica la pagina per la ricerca per attributo.

9.2.1.2 Il sistema passa al punto 9.1.1.

9.2.2 IF l'utente seleziona il link Torna indietro.

9.2.2.1 il sistema carica l'homepage dell'utente e passa al punto 1.2.5.1.

9.2.3 L'utente setta gli attributi con cui vuole effettuare la ricerca di un giocatore.

9.2.4 L'utente schiaccia il pulsante Ricerca.

9.2.5 IF la lista dei risultati di ricerca è vuota.

9.2.5.1 Il sistema mostra una schermata d'avviso.

9.2.5.2 Il sistema torna al punto 9.2.1

9.2.6 L'utente schiaccia il pulsante Ricerca e passa al punto 10.1.

10. Lista risultati di ricerca.

[Home](#)
[Logout](#)

Risultati della ricerca

	Nome	Cognome	Posizione	Squadra	Età	Nazionalità
	Luigi	Rossi	Punta Centrale	Bilbao	23	Italiana
	Giovanni	Verdi	Difensore Centrale	Latina	24	Italiana
	Marco	Bianchi	Punta Centrale	Padova	23	Italiana

[Torna indietro](#)

10.1 IF l'utente clicca il link Home

10.1.1 Il sistema carica l'homepage dell'utente

10.1.2 Il sistema passa al punto 1.2.5.1.

10.2 IF l'utente clicca il link Logout.

10.2.1 Il sistema effettua il logout dell'utente.

10.2.2 Il sistema passa al punto 1.1

10.3 IF L'utente clicca sul link Torna Indietro

10.3.1 Il sistema carica la pagina di ricerca dei giocatori.

10.3.2 Il sistema passa al punto 9.

10.4 L'utente seleziona un giocatore dalla lista

10.5 Il sistema passa al punto 16.1.1.

11. Gestione profili di interesse.

[Home](#) [Logout](#)

Lista Profili ideali

Ruolo

Fascia età:

Descrizione caratteristiche

Aggiungere una descrizione

Aggiungi

Ruolo

Fascia età:

Descrizione caratteristiche

Punta rapida e veloce con un grande senso del goal e della posizione.

Aggiorna

11.1 IF l'utente clicca il link Home

11.1.1 Il sistema carica l'homepage dell'utente

11.1.2 Il sistema passa al punto 1.2.5.1.

11.2 IF l'utente clicca il link Logout.

11.2.1 Il sistema effettua il logout dell'utente.

11.2.2 Il sistema passa al punto 1.1.

11.3 L'utente inserisce il ruolo del giocatore ideale

11.4 L'utente seleziona la fascia d'età del giocatore ideale.

11.5 L'utente aggiunge la descrizione delle caratteristiche del suo giocatore ideale.

11.6 L'utente schiaccia sul pulsante Aggiorna.

11.7 IF i campi hanno dei valori non consistenti

11.7.1 Il sistema mostra una finestra d'errore.

11.7.2 Il sistema torna al punto 11.1

11.8 IF i campi non hanno valori non consistenti

11.8.1 Il sistema aggiorna la lista dei profili ideali.

11.9 L'utente modifica i campi di profili già inseriti

11.10 L'utente clicca il pulsante Aggiorna.

11.11 IF i campi hanno dei valori non consistenti.

11.11.1 Il sistema mostra una finestra d'errore.

11.11.2 Il sistema torna al punto 11.1.

11.12 IF i campi non hanno valori non consistenti

11.12.1 Il sistema aggiorna il profilo ideale.

11.13 Il sistema torna al punto 11.1.

12. Visualizza lista d'interesse (Osservatore)

The screenshot displays a web interface titled "Lista Profili ideali". At the top right, there are two links: "Home" and "Logout". The interface contains two rounded rectangular cards, each representing a player profile. The first card is for a "Difensore centrale" (Central Defender) with an age range of 16 to 42. The description reads: "Giocatore forte fisicamente e preciso nei passaggi." The second card is for a "Punta centrale" (Central Forward) with an age range of 18 to 38. The description reads: "Punta rapida e veloce con un grande senso del goal e della posizione." A vertical scrollbar is visible on the right side of the page.

[Home](#) [Logout](#)

Lista Profili ideali

Ruolo: Difensore centrale

Fascia età: 16 42

Descrizione caratteristiche

Giocatore forte fisicamente e preciso nei passaggi.

Ruolo: Punta centrale

Fascia età: 18 38

Descrizione caratteristiche

Punta rapida e veloce con un grande senso del goal e della posizione.

12.1 L'utente legge i profili da individuare presenti nella lista.

12.2 IF l'utente clicca il link Logout.

12.2.1 Il sistema effettua il logout dell'utente.

12.2.2 Il sistema passa al punto 1.1.

12.3 l'utente clicca il link Home


12.4. Il sistema carica l'homepage dell'utente

12.5. Il sistema passa al punto 1.2.5.1.

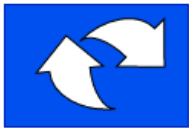
13 Gestione membri della squadra

[Home](#) [Logout](#)

Club: Juve Stabia

Posizione all'interno del club: 

Nome: Michele Cognome: Verdi
Email: verdi@mail.com



Email nuovo allenatore:

[Torna Indietro](#)

13.1 IF l'utente clicca il link Home o Torna Indietro

13.1.1 Il sistema carica l'homepage dell'utente

13.1.2 Il sistema passa al punto 1.2.5.1.

13.2 IF l'utente clicca il link Logout.

13.2.1 Il sistema effettua il logout dell'utente.

13.2.2 Il sistema passa al punto 1.1.

13.3 L'utente seleziona la posizione all'interno del club che intende cercare

13.4 L'utente clicca sul pulsante Cerca.

13.5 Il sistema carica le informazioni di chi occupa la posizione cercata

13.5.1 Se la posizione è vacante il Sistema al posto delle informazioni dell'utente inserisce il seguente messaggio "il posto è vacante."

13.6 L'utente inserisce l'e-mail del nuovo utente per una determinata posizione

13.7 L'utente clicca sul pulsante per aggiungere/modificare l'utente di una posizione all'interno della squadra

13.8 IF l'e-mail non è di un utente registrato OR non corrispondente al ruolo designato OR è già membro

13.8.1 Il sistema apre una finestra d'errore

13.8.2 Il sistema torna al punto 13.1

13.9 IF l'e-mail è di un utente esistente AND svincolato AND del ruolo per cui è stato selezionato.

13.9.1 IF la posizione è vacante

13.9.1.1 Il sistema inserisce l'utente a cui corrisponde l'e-mail nella posizione selezionata.

13.9.2 IF la posizione è già occupata

13.9.2.1 Il sistema sostituisce il vecchio utente con il nuovo utente.

13.10 Il sistema torna al punto 13.1

14.1 Gestione Utenti

[Home](#) [Logout](#)

Gestione Amministratore di squadra

Gestione utenti


Email:

Cerca

Nome: Andrea Cognome: Rossi


Email: rossi@mail.com

Password: 12345



Email:

Password:

 [Torna Indietro](#)

- 14.1.1 IF l'Amministratore di sistema clicca il link Home o Torna Indietro
 - 14.1.1.1 Il sistema carica l'homepage dell'utente
 - 14.1.1.2 Il sistema passa al punto 1.2.5.1.
- 14.1.2 IF l'Amministratore di sistema clicca il link Logout.
 - 14.1.2.1 Il sistema effettua il logout dell'utente.
 - 14.2.2 Il sistema passa al punto 1.1.
- 14.1.3 L'Amministratore di sistema inserisce l'e-mail di un utente da visualizzare.
- 14.1.4 L'Amministratore di sistema clicca sul tasto Cerca
- 14.1.5 IF l'e-mail non appartiene ad un utente registrato.
 - 14.1.5.1 Il sistema mostra una finestra di pop-up di errore.
 - 14.1.5.2 Il sistema torna al punto 14.1
- 14.1.6 IF l'e-mail appartiene ad un utente registrato
 - 14.1.6.1 Il sistema mostra le informazioni dell'utente selezionato
- 14.1.7 L'Amministratore di sistema compila i campi della nuova e-mail e/o nuova password dell'utente
- 14.1.8 L'Amministratore di sistema clicca sul pulsante per cambiare le informazioni dell'utente.
- 14.1.9 IF la nuova e-mail è già usata da un utente registrato
 - 14.1.9.1 Il sistema mostra una finestra di pop-up di errore.
 - 14.1.9.2 Il sistema torna a 14.1
- 14.1.10 IF la nuova e-mail non è già usata da un utente registrato.
 - 14.1.10.1 Il sistema aggiorna i dati dell'utente selezionato all'interno dell'Archivio.
- 14.1.11 IF l'utente clicca sul cestino.
 - 14.1.11.1 il sistema mostra una schermata di conferma.
 - 14.1.11.2 IF l'utente conferma l'eliminazione
 - 14.1.11.2.1 il sistema elimina l'utente dall'archivio.
 - 14.1.11.2.2 il sistema carica la pagina di login.
 - 14.1.11.2.3 il sistema passa al punto 1.1.
 - 14.1.11.3 IF il sistema non conferma l'eliminazione.
 - 14.1.11.3.1 il sistema torna al punto 14.1.1
- 14.1.12 IF l'utente clicca su Gestione Utenti
 - 14.1.12.1 Il sistema carica la pagina gestione utenti e passa al punto 14.2.1

14.2 Gestione Amministratore di squadra

[Home](#) [Logout](#)

Gestione Amministratore di squadra


Gestione utenti

Campionato:

Squadra:

Cerca

Nome: Mario Cognome: Bianchi
Email: mariobianchi@mail.com



Email nuovo amministratore di squadra:

[Torna Indietro](#)

14.2.1 IF l'Amministratore di sistema clicca il link Home o Torna Indietro

14.2.1.1 Il sistema carica l'homepage dell'utente

14.2.1.2 Il sistema passa al punto 1.2.5.1.

14.2.2 IF l'Amministratore di sistema clicca il link Logout.

14.2.2.1 Il sistema effettua il logout dell'utente.

14.2.2.2 Il sistema passa al punto 1.1.

14.2.3 IF l'Amministratore di sistema clicca su Gestione Utenti

14.1.3.1 Il sistema carica la pagina gestione utenti e passa al punto 14.1.1

14.2.4 L'Amministratore di sistema seleziona il campionato

14.2.5 L'Amministratore di sistema seleziona la squadra

14.2.6 L'Amministrazione di sistema clicca sul tasto Cerca

14.2.7 Il sistema carica le informazioni di chi occupa la posizione di Amministratore di squadra della squadra selezionata.

14.2.7.1 Se la posizione è vacante il Sistema al posto delle informazioni dell'utente inserisce il seguente messaggio "il posto è vacante."

14.2.8 L'Amministratore di sistema inserisce l'e-mail del nuovo utente per una determinata posizione

14.2.9 L'Amministratore di sistema clicca sul pulsante per aggiungere/modificare l'amministratore di squadra all'interno della squadra

14.2.10 IF l'e-mail non è di un utente registrato OR non corrispondente al ruolo d'amministratore di squadra OR è già membro

14.2.10.1 Il sistema apre una finestra d'errore

14.2.10.2 Il sistema torna al punto 14.2.1

14.2.10 IF l'e-mail è di un utente esistente AND svincolato AND del ruolo di amministratore di squadra.

14.2.10.1 IF la posizione è vacante

14.2.10.1.1 Il sistema inserisce l'utente a cui corrisponde l'e-mail nella posizione selezionata.

14.2.10.2 IF la posizione di amministratore di squadra è già occupata












14.2.10.2.1 Il sistema sostituisce il vecchio utente con il nuovo utente.

14.2.11 Il sistema torna al punto 14.2.1.


15. Lista giocatori da tenere d'occhio.


[Home](#)
[Logout](#)


Lista Giocatori selezionati

	Nome	Cognome	Posizione	Squadra	Età	ValoreDiMercato	Nazionalità	Giudizio dirigenza	Giudizio allenatore	
	Luigi	Rossi	Punta Centrale	Bilbao	23	2.000.000€	Italiana			
	Giovanni	Verdi	Difensore Centrale	Latina	24	200.000€	Italiana			
	Marco	Bianchi	Punta Centrale	Padova	23	300.000€	Italiana			

Legenda

 : Giocatore approvato

 : Nessun giudizio

 : Giocatore non approvato

[Torna indietro](#)

15.1 IF l'utente clicca il link Home o Torna Indietro

15.1.1 Il sistema carica l'homepage dell'utente

15.1.2 Il sistema passa al punto 1.2.5.1.

15.2 IF l'utente clicca il link Logout.

15.2.1 Il sistema effettua il logout dell'utente.

15.2.2 Il sistema passa al punto 1.1.

15.3 IF l'utente preme l'icona del cestino


15.3 l'utente elimina il calciatore corrispondente dalla lista d'interesse.

15.4 L'utente seleziona un calciatore dalla lista d'interesse

15.4.1 Il sistema carica la scheda del giocatore corrispondente

15.4.2 Il sistema passa al punto 16.2.1

16.1 Scheda giocatore non in lista

Informazioni principali	Storico statistiche stagionali	Cronologia trasferimenti	Grafico statistiche	Cronologia Infortuni	Scheda osservatore
<div></div> <div>Luigi Verdi<div>Valore Di Mercato: 100.000 €</div></div> <div><p>Data di nascita: 20/10/2000</p><p>Luogo di nascita: Roma</p><p>Nazionalità : italiana</p><p>Altezza: 176 cm</p><p>Ruolo: Punta Centrale</p><p>Squadra Attuale : Cremonese</p><p>Campionato: Serie B</p><p>Presenze Stagione Attuale: 22</p><p>Minuti giocati Stagione Attuale: 1900</p><p>Goal Stagione Attuale: 15</p><p>Cartellini Gialli Stagione Attuale: 2</p><p>Cartellini Rossi Stagione Attuale: 0</p><p>Procuratore: Famiglia</p></div> <div><div>Torna ai risultati di ricerca</div><div>Aggiungi alla lista dei giocatori di interesse</div></div>					

Informazioni principali

Storico statistiche stagionali

Cronologia trasferimenti

Grafico statistiche

Cronologia Infortuni

Scheda osservatore




Luigi Verdi

Stagione	Competizione	Società	Presenze	Pap	Goal	Assist	Ammonizioni	Doppie Ammonizioni	Espulsioni	Minuti Giocati
19/20	SerieB	Cremonese	22	1,6	15	2	2	0	0	1900'
19/20	Coppa Italia	Cremonese	3	2	2	1	1	0	0	270'
18/19	SerieB	Cremonese	34	1,8	21	2	3	0	0	3000'
18/19	Coppa Italia	Cremonese	1	0	1	0	0	0	0	90'

Torna ai risultati di ricerca

Aggiungi alla lista dei giocatori di interesse

Informazioni principali	Storico statistiche stagionali	Cronologia trasferimenti	Grafico statistiche	Scheda osservatore	Scheda osservatore												
<div>  <p>Luigi Verdi</p> </div> <div> <table> <tr> <th>Stagione</th><th>Data</th><th>Venditore</th><th>Acquirente</th><th>Valore di mercato</th><th>Riscatto</th></tr> <tr> <td>18/19</td><td>17/07/2018</td><td>Latina Calcio</td><td>Cremonese</td><td>100.000€</td><td>110.000€</td></tr> </table> </div> <div> <div>Torna ai risultati di ricerca</div> <div>Aggiungi alla lista dei giocatori di interesse</div> </div>						Stagione	Data	Venditore	Acquirente	Valore di mercato	Riscatto	18/19	17/07/2018	Latina Calcio	Cremonese	100.000€	110.000€
Stagione	Data	Venditore	Acquirente	Valore di mercato	Riscatto												
18/19	17/07/2018	Latina Calcio	Cremonese	100.000€	110.000€												

Informazioni
principali

Storico statistiche
stagionali

Cronologia
trasferimenti

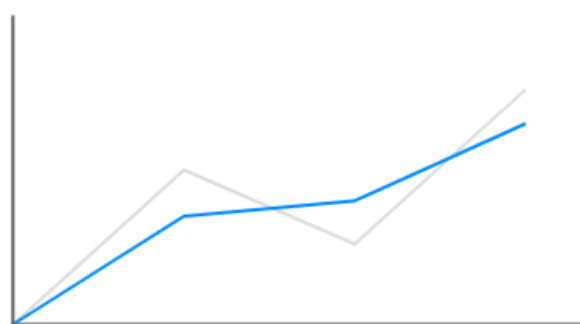
**Grafico
statistiche**

Cronologia
Infortuni

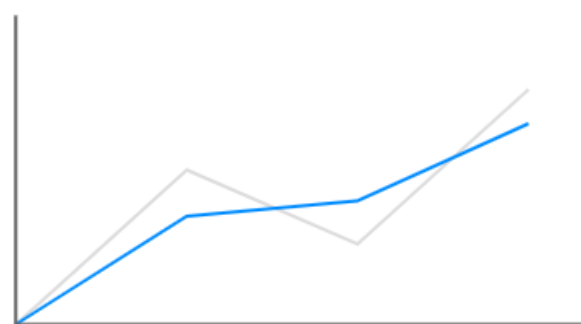
Scheda
osservatore



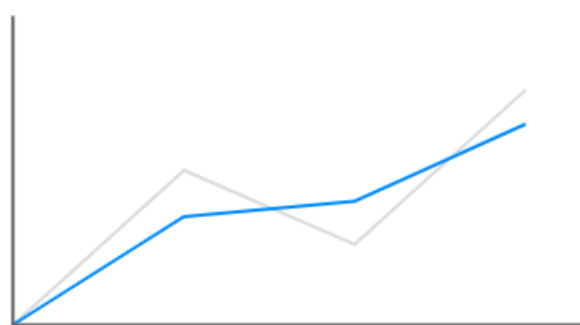
Luigi Verdi



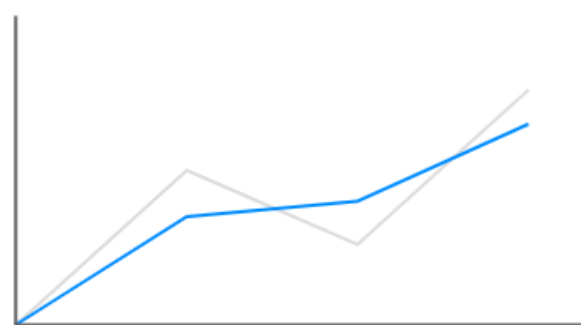
Goal fatti a stagione



Minuti giocati a partita



Assist Fatti a Stagione



Cartellini a stagione

**Torna ai risultati di
ricerca**

**Aggiungi alla lista dei
giocatori di interesse**

Informazioni principali


Storico statistiche stagionali

Cronologia trasferimenti

Grafico statistiche

Cronologia Infortuni

Scheda osservatore



Luigi Verdi

Stagione	Infortunio	da	fino al	Giorni	Partite Perse	Società
19/20	Risentimento muscolare	18/dic/2019	27/dic/2019	9	2	Cremonese
19/20	Influenza	18/nov/2019	22/nov/2019	4	1	Cremonese
18/19	Problemi alla coscia	14/gen/2019	16/gen/2019	2	1	Cremonese
18/19	Problema muscolare	18/set/2018	26/set/2018	7	1	Cremonese

Torna ai risultati di ricerca

Aggiungi alla lista dei giocatori di interesse

Informazioni
principali

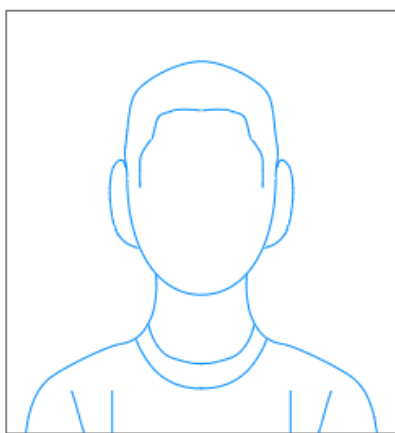
Storico statistiche
stagionali

Cronologia
trasferimenti

Grafico
statistiche

Cronologia
Infrotuni

Scheda
osservatore



Luigi Verdi

Scheda Valutazione finale



Invia Relazione

**Torna ai risultati di
ricerca**

**Aggiungi alla lista dei
giocatori di interesse**

16.1.1 IF l'utente preme il pulsante Torna ai risultati di ricerca

16.1.1.1 Il sistema carica la pagina dei risultati di ricerca

16.1.1.2 Il sistema passa al punto 10.1.

16.1.2 IF l'utente clicca il menu di selezione.

16.1.2.1 Il sistema cambia la pagina della scheda del giocatore.

16.1.3 IF l'utente preme il tasto Aggiungi alla lista dei giocatori di interesse.

16.1.3.1 Il sistema aggiunge alla lista di interesse il giocatore attuale.

16.1.4 IF l'utente è un osservatore

16.1.4.1 L'utente nella Scheda osservatore scrive una relazione finale.


16.1.4.2 L'utente seleziona un punteggio da 1 a 5.

16.1.4.2 L'utente preme il pulsante Invia Relazione.


16.1.4.3 Il sistema aggiunge al giocatore una relazione.

16.1.5 Il sistema torna al punto 16.1.1.

16.2 Scheda Giocatore in lista

Informazioni principali	Storico statistiche stagionali	Cronologia trasferimenti	Grafico statistiche	Cronologia Infortuni	Scheda osservatore
<div></div> <div>Luigi Verdi<div>Valore Di Mercato: 100.000 €</div></div> <div><p>Data di nascita: 20/10/2000</p><p>Luogo di nascita: Roma</p><p>Nazionalità : italiana</p><p>Altezza: 176 cm</p><p>Ruolo: Punta Centrale</p><p>Squadra Attuale : Cremonese</p><p>Campionato: Serie B</p><p>Presenze Stagione Attuale: 22</p><p>Minuti giocati Stagione Attuale: 1900</p><p>Goal Stagione Attuale: 15</p><p>Cartellini Gialli Stagione Attuale: 2</p><p>Cartellini Rossi Stagione Attuale: 0</p><p>Procuratore: Famiglia</p></div> <div><div>Approva Giocatore</div><div>Nessun giudizio</div><div>Rifiuta Giocatore</div><div>Torna ai risultati di ricerca</div></div>					

Informazioni principali	Storico statistiche stagionali	Cronologia trasferimenti	Grafico statistiche	Cronologia Infortuni	Scheda osservatore
-------------------------	--------------------------------	--------------------------	---------------------	----------------------	--------------------



Luigi Verdi

Stagione	Competizione	Società	Presenze	Pap	Goal	Assist	Ammonizioni	Doppie Ammonizioni	Espulsioni	Minuti Giocati
19/20	SerieB	Cremonese	22	1,6	15	2	2	0	0	1900'
19/20	Coppa Italia	Cremonese	3	2	2	1	1	0	0	270'
18/19	SerieB	Cremonese	34	1,8	21	2	3	0	0	3000'
18/19	Coppa Italia	Cremonese	1	0	1	0	0	0	0	90'

Approva Giocatore

Nessun giudizio

Rifiuta Giocatore

Torna ai risultati di ricerca

Informazioni principali	Storico statistiche stagionali	Cronologia trasferimenti	Grafico statistiche	Cronologia Infortuni	Scheda osservatore
-------------------------	--------------------------------	--------------------------	---------------------	----------------------	--------------------



Luigi Verdi


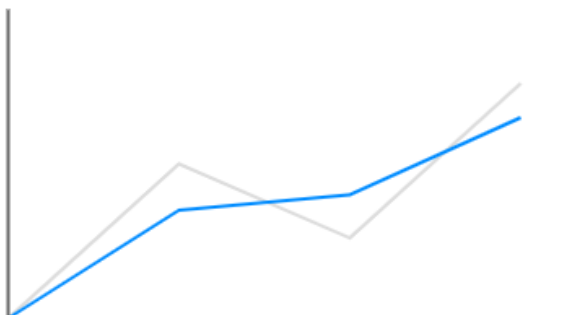
Stagione	Data	Venditore	Acquirente	Valore di mercato	Riscatto
18/19	17/07/2018	Latina Calcio	Cremonese	100.000€	110.000€

Approva Giocatore

Nessun Giudizio

Rifiuta Giocatore

Torna ai risultati di ricerca

Informazioni principali	Storico statistiche stagionali	Cronologia trasferimenti	Grafico statistiche	Cronologia Infortuni	Scheda osservatore
<div data-bbox="193 389 357 551">  </div> <div data-bbox="193 562 347 600"> <p>Luigi Verdi</p> </div> <div data-bbox="185 674 762 1039">  <p>Goal fatti a stagione</p> </div> <div data-bbox="837 674 1415 1039">  <p>Minuti giocati a partita</p> </div> <div data-bbox="185 1144 762 1509">  <p>Assist Fatti a Stagione</p> </div> <div data-bbox="837 1144 1415 1509">  <p>Cartellini a stagione</p> </div> <div data-bbox="209 1588 523 1733"> <p>Approva Giocatore</p> </div> <div data-bbox="632 1588 946 1733"> <p>Nessun Giudizio</p> </div> <div data-bbox="1058 1588 1372 1733"> <p>Rifiuta Giocatore</p> </div> <div data-bbox="632 1783 946 1928"> <p>Torna ai risultati di ricerca</p> </div>					

Informazioni principali	Storico statistiche stagionali	Cronologia trasferimenti	Grafico statistiche	Cronologia Infortuni	Scheda osservatore
-------------------------	--------------------------------	--------------------------	---------------------	----------------------	--------------------



Luigi Verdi

Stagione	Infortunio	da	fino al	Giorni	Partite Perse	Società
19/20	Risentimento muscolare	18/dic/2019	27/dic/2019	9	2	Cremonese
19/20	Influenza	18/nov/2019	22/nov/2019	4	1	Cremonese
18/19	Problemi alla coscia	14/gen/2019	16/gen/2019	2	1	Cremonese
18/19	Problema muscolare	18/set/2018	26/set/2018	7	1	Cremonese

Approva Giocatore

Nessun Giudizio

Rifiuta Giocatore

Torna ai risultati di ricerca

Informazioni
principali

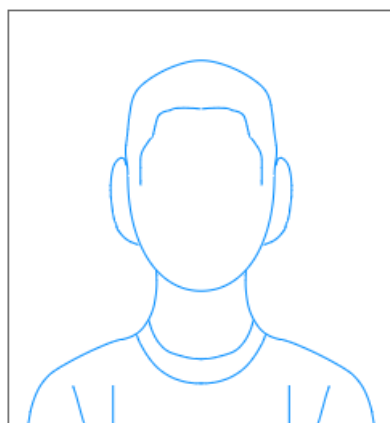
Storico statistiche
stagionali

Cronologia
trasferimenti

Grafico
statistiche

Cronologia
Infortuni

Scheda
osservatore



Luigi Verdi

Scheda Valutazione finale



Approva Giocatore

Nessun Giudizio

Rifiuta Giocatore

**Torna ai risultati di
ricerca**

16.2.1 IF l'utente preme il pulsante Torna ai risultati di ricerca

16.2.1.1 Il sistema carica la pagina dei giocatori selezionati

16.2.1.2 Il sistema passa al punto 15.1.

16.2.2 IF l'utente clicca il menu di selezione.

16.2.2.1 Il sistema cambia la pagina della scheda del giocatore.

16.2.3 IF l'utente preme il tasto Approva giocatore OR Nessun Giudizio OR Rifiuta Giocatore.

16.2.3.1 Il sistema aggiorna lo stato del giocatore nella lista dei giocatori selezionati.

16.2.5 Il sistema torna al punto 16.2.1.