FIP MANAGEMENT

Relazione per “Programmazione ad Oggetti”

Luca Dal Seno, Francesco Baiocchi

Capitolo 1

1. Analisi

1.1. Requisiti

Il gruppo si pone come obiettivo quello di realizzare un software gestionale per le Federazione Italiana Pallacanestro. L'applicazione sarà in grado di gestire la parte manageriale di campionati, squadre e giocatori; inoltre vi sarà un'ulteriore funzione di gestione di una singola partita e la possibilità di tenere traccia delle statistiche dei giocatori.

1.2. Funzionalità

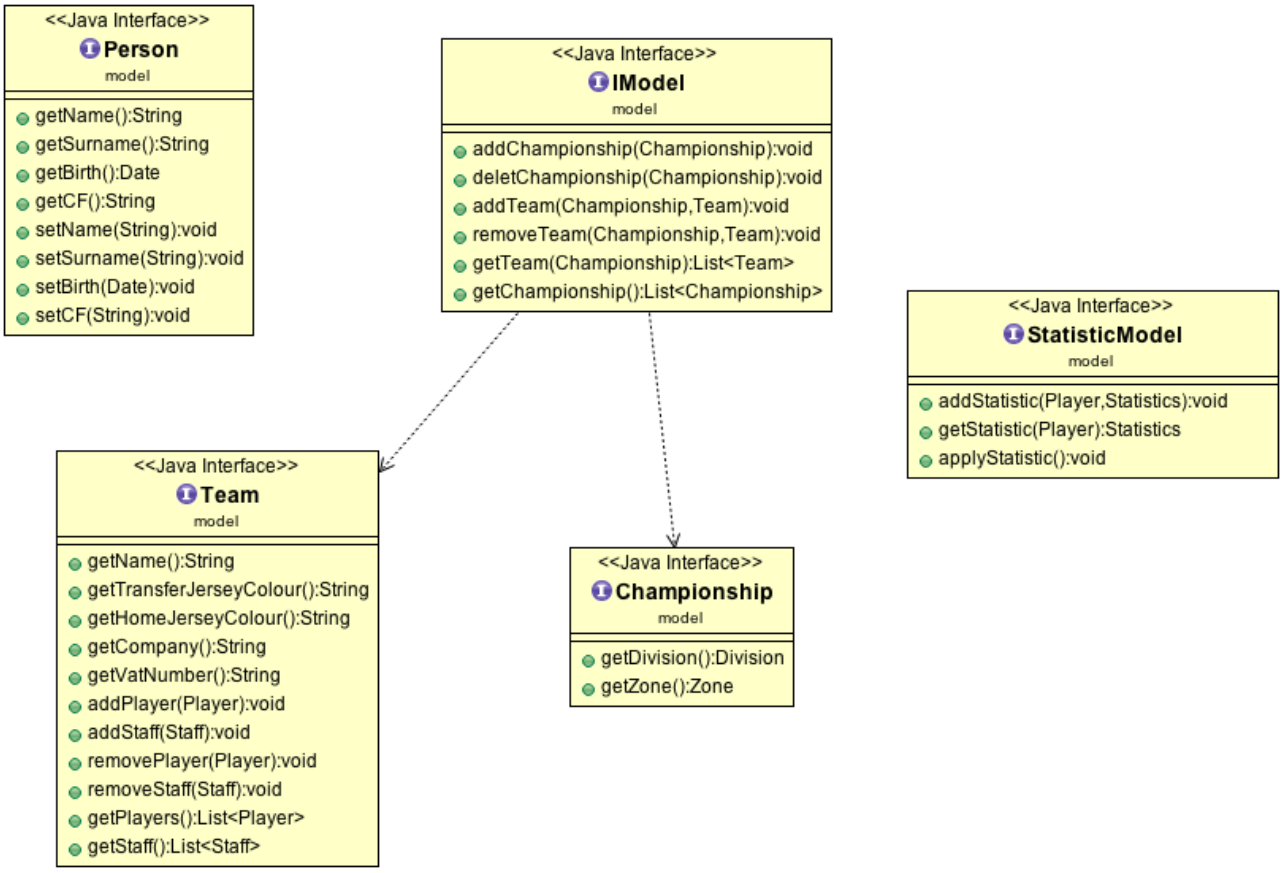
Il software sarà in grado di permettere la creazione di uno o più campionati scelti da un'apposita lista che rispecchia realmente i campionati FIP. Per ogni singolo campionato vi sarà la possibilità di aggiungere o eliminare squadre e per ogni squadra vi sarà la possibilità di aggiungere, eliminare o modificare giocatori e/o staff.
Inoltre sarà in grado di gestire una partita fra due squadre dello stesso campionato: si potrà tenere traccia delle varie statistiche per giocatore ed infine esportarle su un file excel.

Il software esporrà le seguenti funzionalità:

- Aggiungere/eliminare campionati
- Aggiungere/eliminare squadre
- Aggiungere/eliminare giocatori e staff
- Visualizzare/modificare giocatori e staff
- Creare una partita
- Esportare le statistiche della partita su file excel

1.3. Modello del dominio

Il software permette la creazione di più campionati. Ogni campionato sarà composto da più squadre. In ogni squadra vi sarà la possibilità di inserire dei giocatori o dei membri dello staff (specificando anche il ruolo che avranno all'interno di essa). Inoltre la parte delle statistiche deve essere correlata allo specifico giocatore.



Capitolo 2

2. Design

2.1. Architettura

Il pattern scelto alla base dello sviluppo del software è il Model-View-Controller pattern meglio noto come MVC. La scelta di questo pattern deriva dal fatto che esso "divide" lo sviluppo in tre macro sezioni alla quale è possibile lavorare in maniera (quasi del tutto) indipendente: la parte di Model si occupa della gestione e del recupero dei dati utili all'applicazione, la parte di View visualizza tali dati, li interfaccia all'utente e comunica tramite un Observer al controller, la parte di Controller invece esegue i comandi dati dall'utente e modifica la struttura dei dati.
Il pattern Observer infine è il collegamento tra View e Controller.

2.2. Design dettagliato

2.2.1. Model

Il model si basa sulle seguenti entità:

- ***AccessPermission***: enumerazione che distingue i permessi di accesso
- ***Championship***: interfaccia che definisce un campionato
- ***ChampionshipImpl***: classe che modella un campionato
- ***Division***: enumerazione che distingue la divisione di un campionato
- ***IModel***: interfaccia che definisce la classe Model
- ***Model***: classe che si occupa della gestione dei dati
- ***MyTableModel***: classe astratta che definisce una Table
- ***Person***: interfaccia che definisce una persona
- ***PersonImpl***: classe che modella una persona
- ***Player***:classe che modella un giocatore
- ***Staff***: classe che modella un membro dello staff
- ***StatisticModel***: interfaccia che definisce un model per le statistiche

- ***StatisticModelImpl***: classe che si occupa della gestione delle statistiche
- ***Statistics***: classe che modella una statistica
- ***Team***: interfaccia che definisce una squadra
- ***TeamImpl***: classe che modella una squadra
- ***Zone***: enumerazione che distingue la zona di un campionato

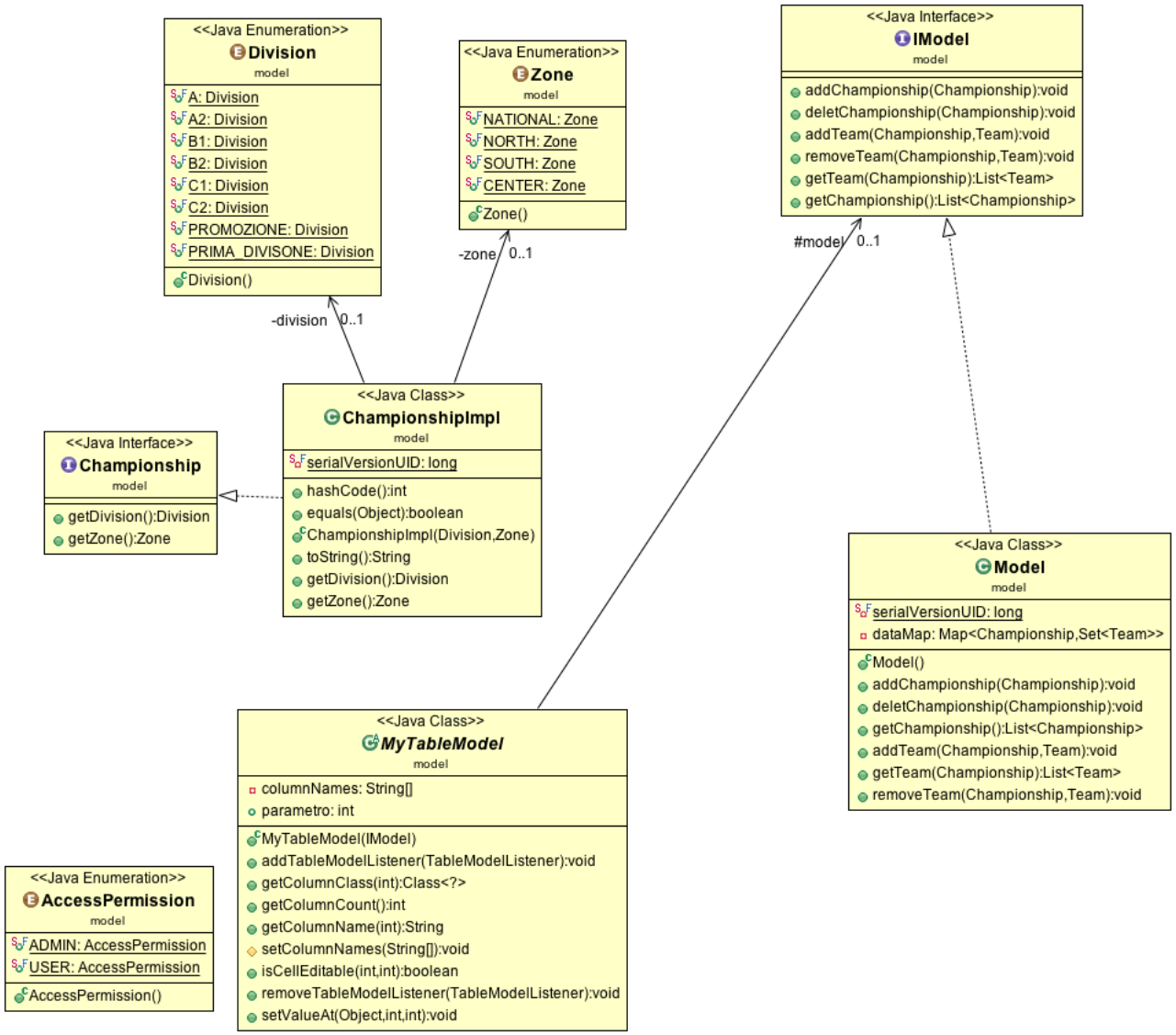


*Figura **Errore. Nel documento non esiste testo dello stile specificato.**-1 Prima parte del model relativa al campionato e alla gestione dati*

Descrizione: la classe Model contiene una mappa per il salvataggio dei dati, avente in chiave un Campionato mentre in value un Set di Squadre associato. Come già anticipato la classe Model contiene i metodi per la gestione dei dati, mentre invece per la visualizzazione dei dati sono state utilizzate delle table che estendono tutte dalla classe MyTableModel (come vedremo più avanti).

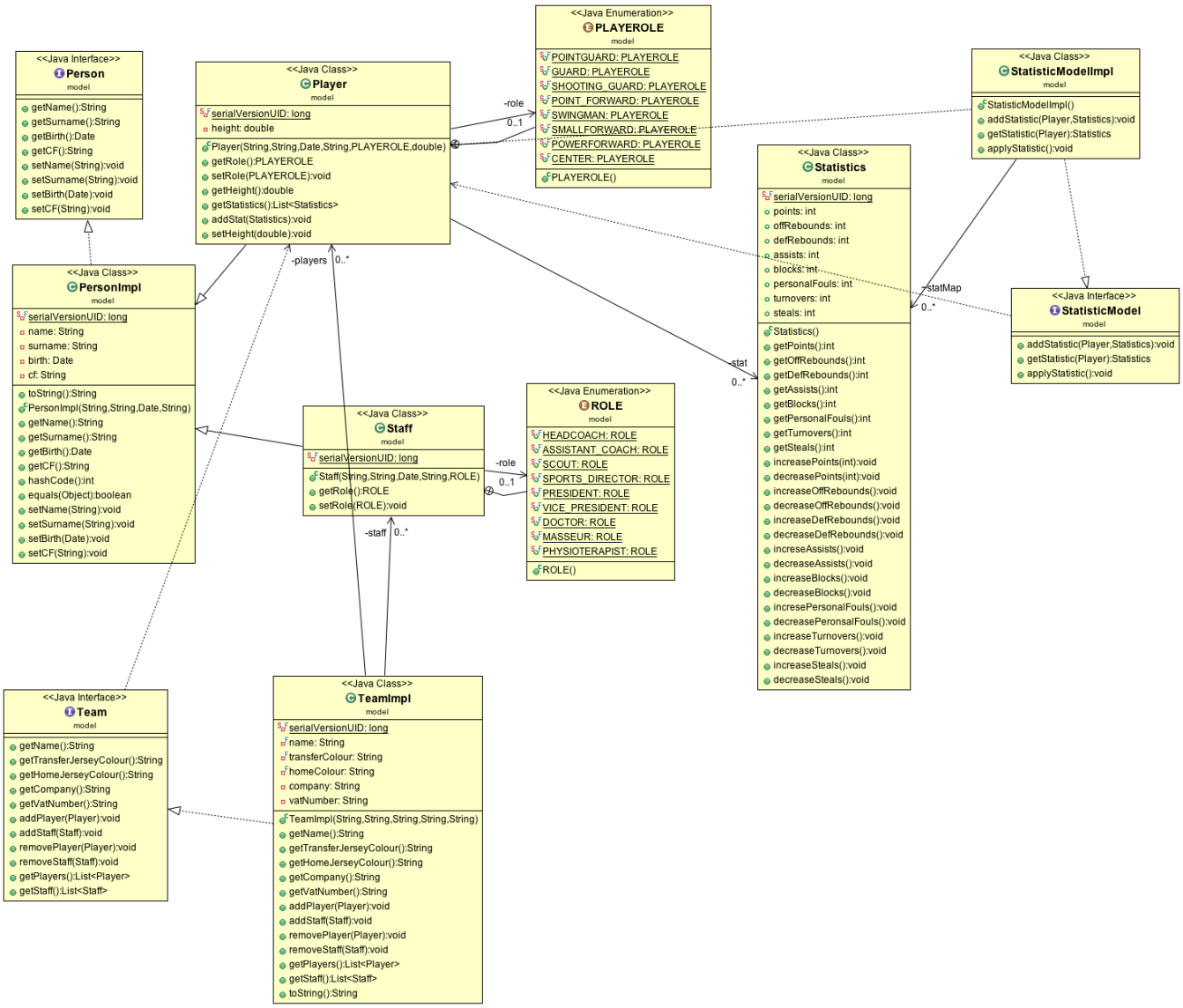


*Figura **Errore. Nel documento non esiste testo dello stile specificato.**-2 Seconda parte del model relativa alla gestione e struttura delle squadre e statistiche*

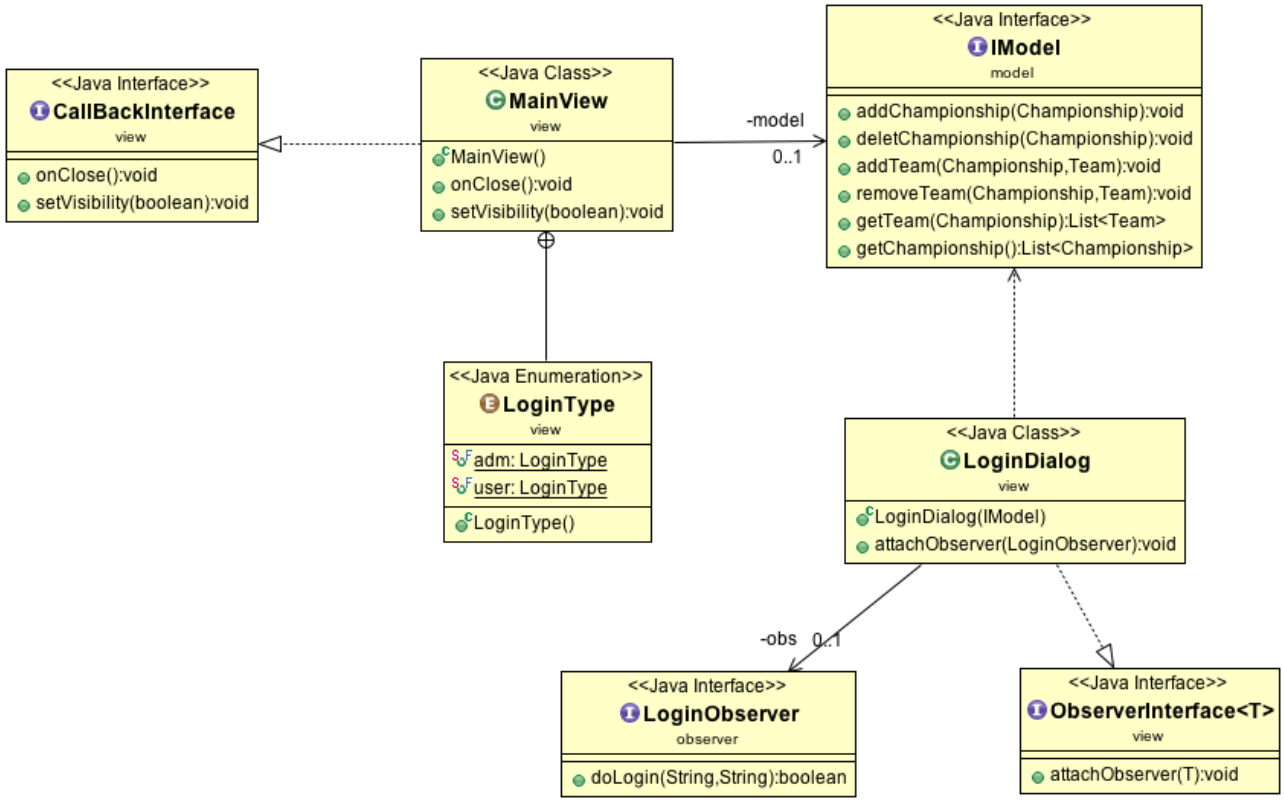
Descrizione: la gestione delle squadre invece è lasciata all'interno della classe Team. Ogni singola squadra possiede due Set: uno per i giocatori e uno per i membri dello staff tecnico. Per quanto riguarda le statistiche si è scelto di implementare un'interfaccia Model a parte per tenere traccia delle statistiche relative ad un determinato giocatore: per fare ciò si è ricorso ad una mappa con in chiave il giocatore e in value un oggetto di tipo Statistics. Come si può vedere dall'UML la classe Statistics presenta tanti campi quante sono le statistiche da tenere traccia.

2.2.2. View

La parte di View della nostra applicazione si compone di 4 pagine con relative Dialog per quanto riguarda la parte manageriale, mentre di una sola pagina per quanto riguarda la parte delle statistiche. Vediamo più nel dettaglio come sono strutturate.

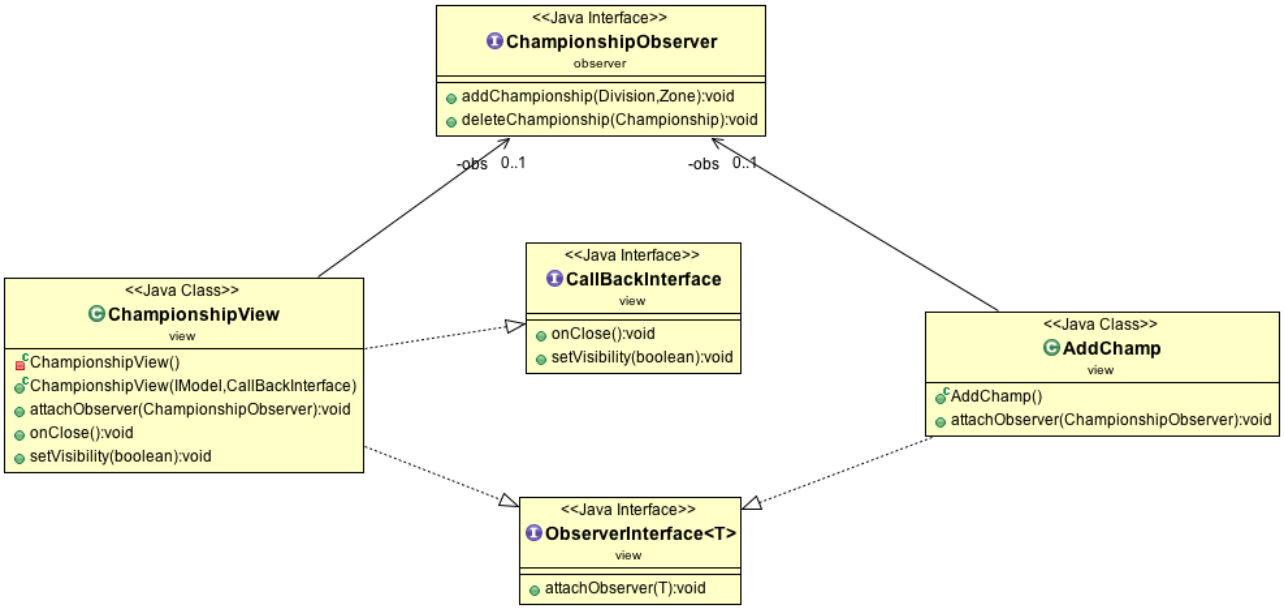
2.2.2.1. MainView

Descrizione: la mainView come suggerisce il nome è la prima View che viene visualizzata dall'utente.
Non appena aperto il programma il model è aggiornato tramite il metodo loading( ): viene caricata la configurazione precedentemente salvata (squadre,campionati ecc).
Presenta due pulsanti: uno per accedere alla parte gestionale e uno per accedere alla parte relativa alla gestione di una partita.
Alla pressione di uno di questi pulsanti si aprirà un JDialog di login per permettere solo agli utenti autenticati di accedervi.



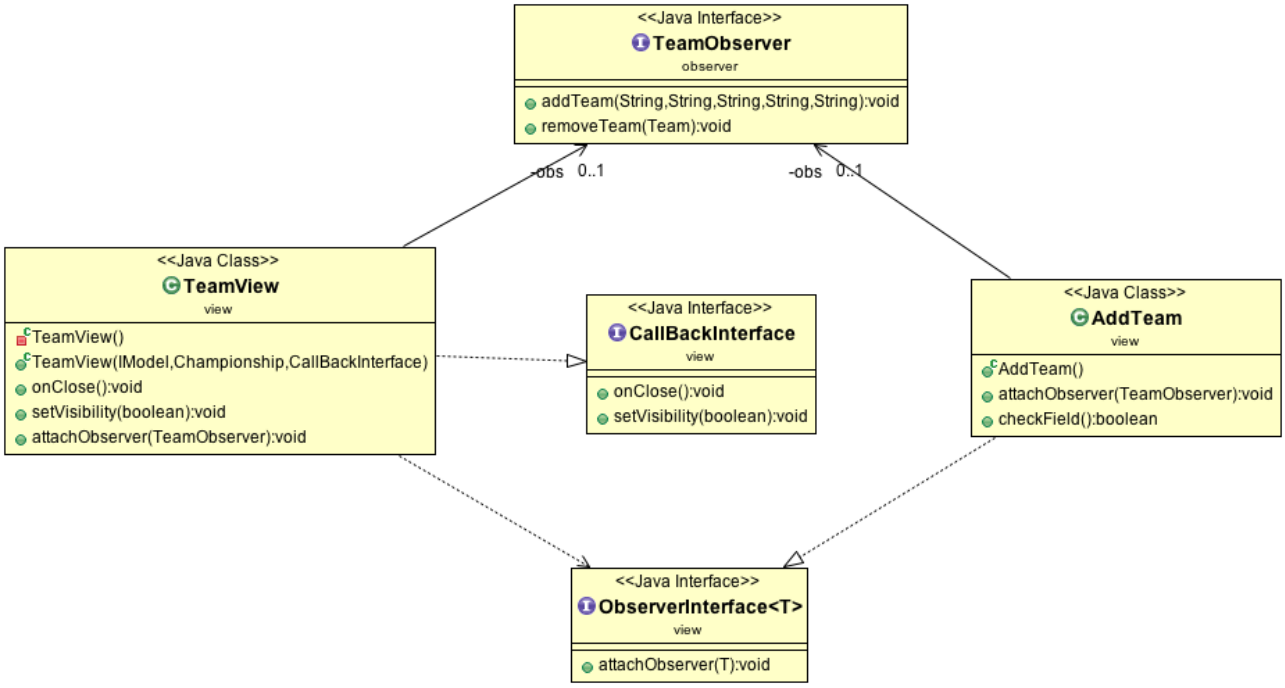
2.2.2.2. ChampionshipView

Descrizione: una volta completata la login della parte manageriale, apparirà la ChampionshipView ossia la pagina di gestione dei campionati. E' composta da una JTable ,dove si trovato tutti i campionati inseriti e da due bottoni: uno permette l'aggiunta di un campionato e uno permette di eliminare il campionato selezionato.
Alla pressione del pulsante per aggiungere un campionato, viene visualizzata la dialog AddChamp che permette di inserire i campi utili a definirlo.



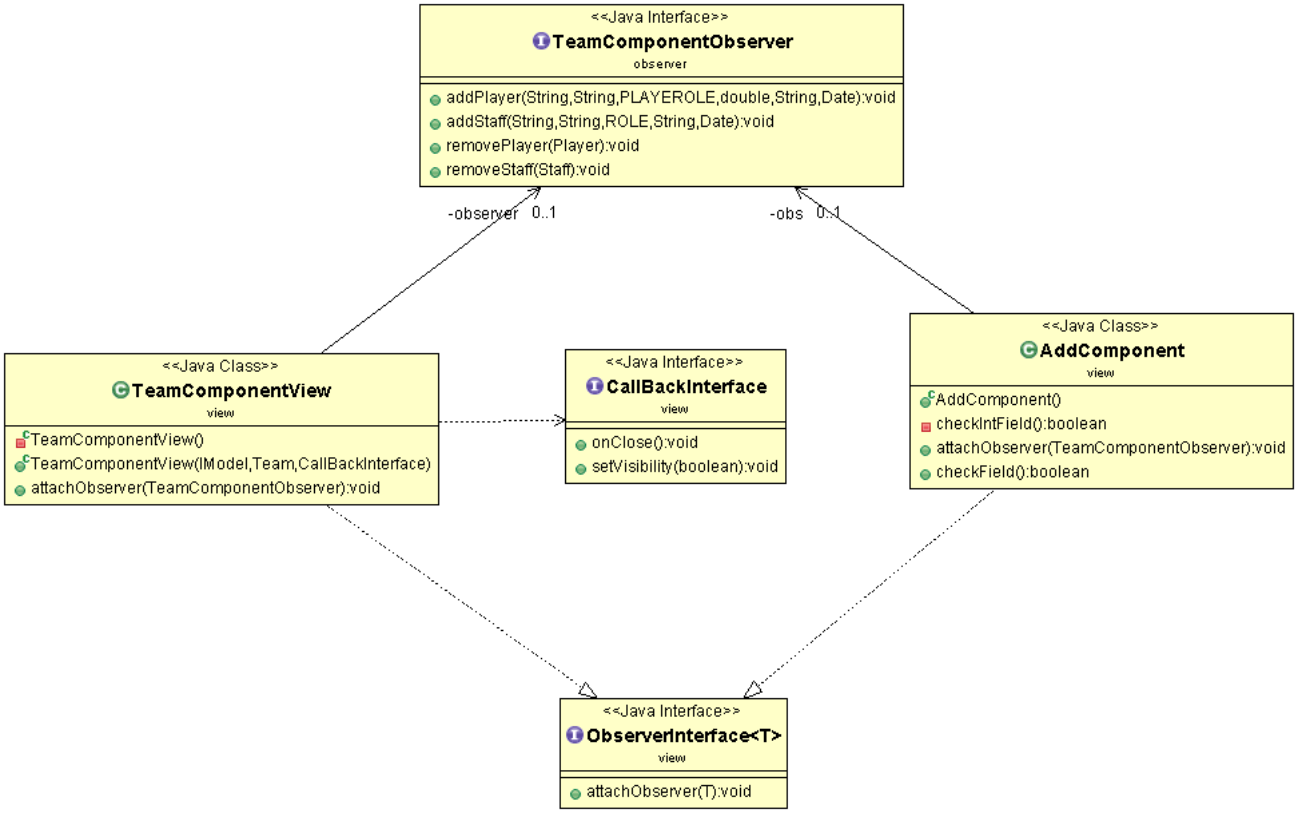
2.2.2.3. TeamView

Descrizione: facendo doppio click su una riga della JTable dei campionati si aprirà la view riguardante le squadre relative a quel campionato.
In maniera analoga alla precedente view, apparirà un JTable contenente le squadre e due pulsanti: uno per l'aggiunta di una squadra e uno per eliminare quella selezionata dal campionato.
Alla pressione del tasto per l'aggiunta viene visualizzata la dialog AddTeam che permette di inserire i campi utili a definire una squadra.



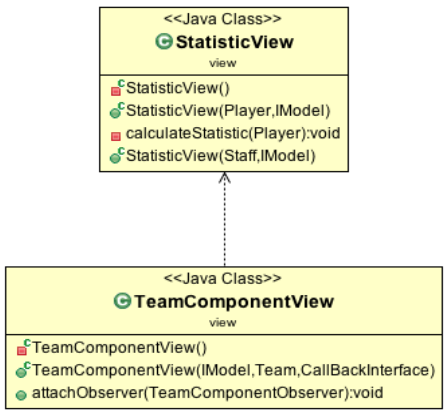
2.2.2.4. TeamComponentView

Descrizione: con un doppio click sulla JTable delle squadre si aprirà la TeamComponentView ossia una view in cui è possibile vedere i giocatori e lo staff facenti parte della squadra selezionata.
Questa volta sono presenti due JTable, una per i giocatori e una per lo staff, ed è possibile aggiungere un componente da un unico pulsante. Mentre per eliminare un membro si è preferito fare la distinzione fra giocatore e staff tecnico.
In maniera analoga alle precedenti View alla pressione del pulsante per l'aggiunta di un membro della squadra viene visualizzata la dialog AddComponent che permette di impostare i campi necessari.



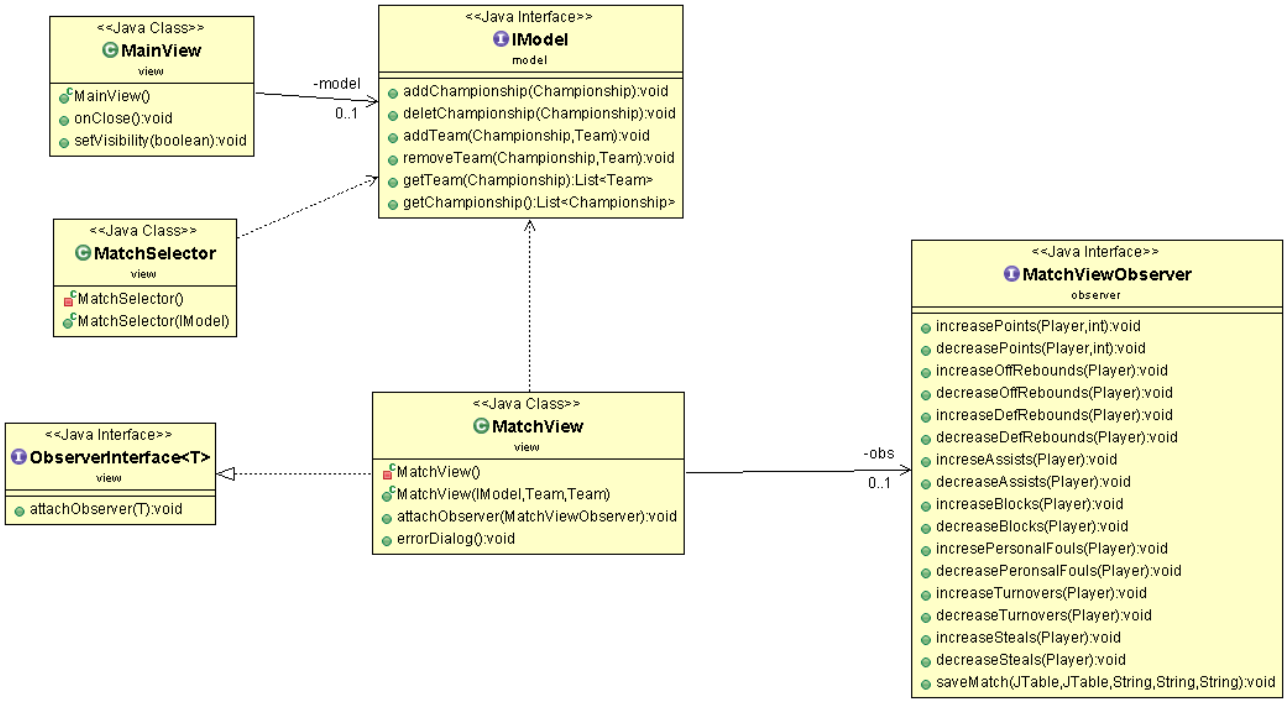
2.2.2.5. StatisticView

Descrizione: con un doppio click su o un giocatore o un membro dello staff si aprirà una StatisticView dove è possibile visualizzare/modificare le informazioni della persona e visualizzare le sue statistiche (qualora fosse un giocatore).
Una volta modificato si potrà confermare tramite un tasto ApplyChanges e si verrà reindirizzato alla JTable dei componenti della squadra con le nuove modifiche effettuate.



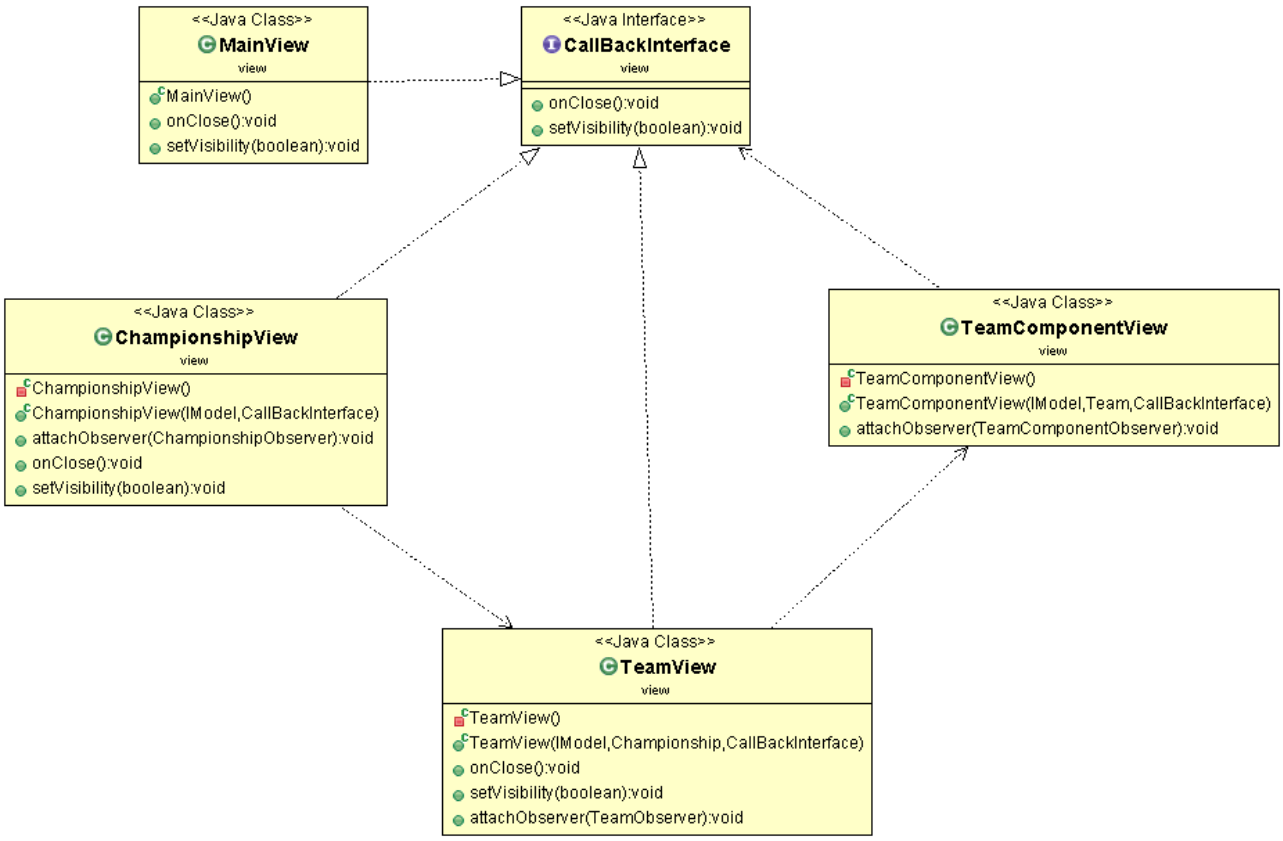
2.2.2.6. MatchView

Descrizione: la MatchView è accedibile dalla mainView cliccando sul pulsante Match. Si aprirà dapprima la diagol MatchSelector che accedendo al model permette di scegliere il campionato di cui si vuole fare disputare la partita e subito dopo le squadre presenti in quel campionato. Una volta selezionate si aprirà la vera e propria MatchView composta da due tabelle, contenente i giocatori delle due squadre, e da una serie di bottoni atti a gestire le statistiche. In fondo invece è possibile esportare la partita in un file excel tramite il pulsante SaveMatch.
Una volta salvato il match verranno salvate anche le statistiche relative ad un giocatore e sommate alle precedenti (visualizzabili cliccando due volte, nella parte gestionale, sul giocatore).
E' facile intuire che questa view deve avere completo accesso al model per poter caricare prima i campionati presenti e poi le squadre presenti a quel campionato.



2.2.2.7. CallBack

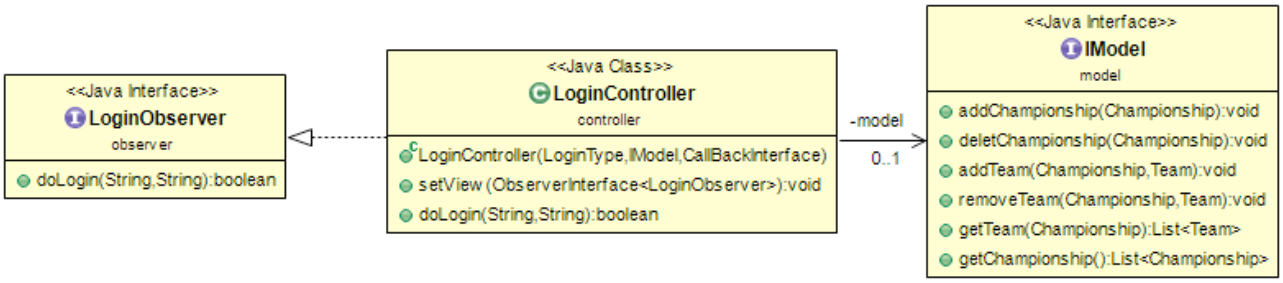
Di particolare attenzione risulta essere il meccanismo di CallBack utilizzato per gestire l'andamento fra le pagine. Tutte le view, ad eccezione di quella per i Match, implementano anche l'interfaccia CallBack che presenta due metodi: onClose( ) e setVisibility( ). Il primo metodo serve a passare come riferimento la view sulla quale si sta operando e una volta premuto un tasto back si tornerà alla precedente view. Il secondo metodo invece è atto e settare la visibilità della view, sulla quale si è, a false.



2.2.3. Controller

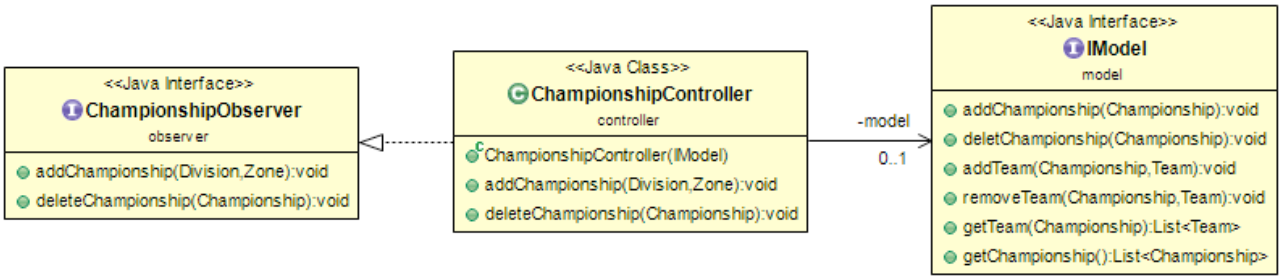
La parte di controller del nostro progetto si compone di svariate classi, inerenti ciascuna a una view diversa, che implementano delle interfacce di observer, andando a effettuare le modifiche a livello del modello.

2.2.3.1. LoginController



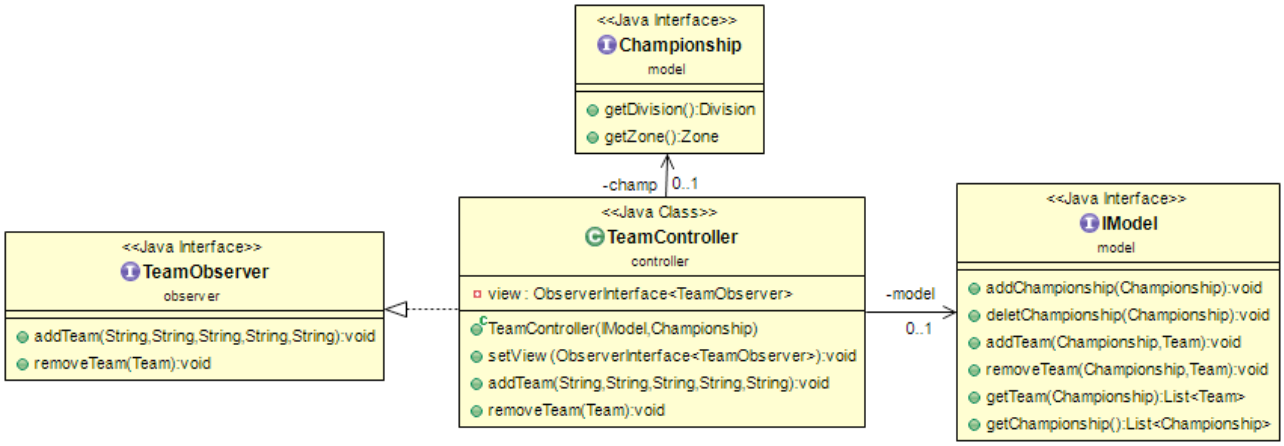
Descrizione: LoginController è la classe di controllo della view di login.
Questa classe implementa l'interfaccia di observer LoginObserver, all'interno della quale sono definiti i metodi. Il metodo "doLogin" si occupa di controllare che le credenziali di accesso siano corrette e, nel caso lo siano, lancia le view corrispondenti all' area alla quale si vuole accedere.

2.2.3.2. ChampionshipController



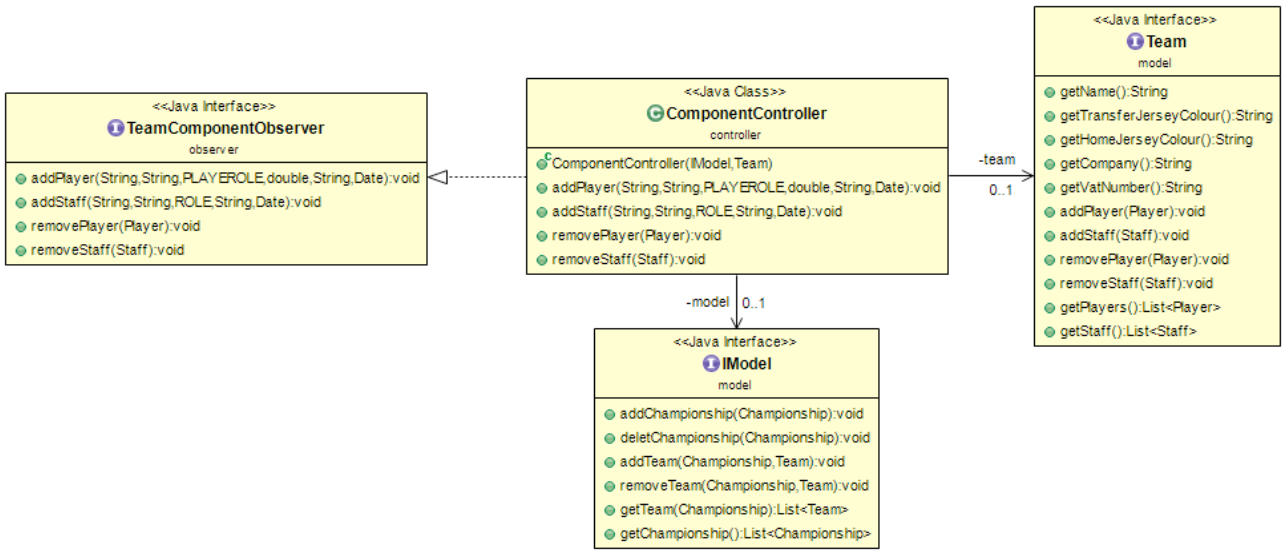
Descrizione : ChampionshipController è la classe di controllo della view ChampionshipView. Questa classe implementa l'interfaccia di observer ChamionshipObserver, in modo da definire l'inserimento dei campionati creati, o la loro cancellazione, all' interno del model.
I metodi principali sono dunque "addChampionship" e "deleteChampionship" i quali si occupano rispettivamente di creare all' interno del modello un campionato, nel primo caso, o di effettuarne la cancellazione, nel secondo.

2.2.3.3. TeamController



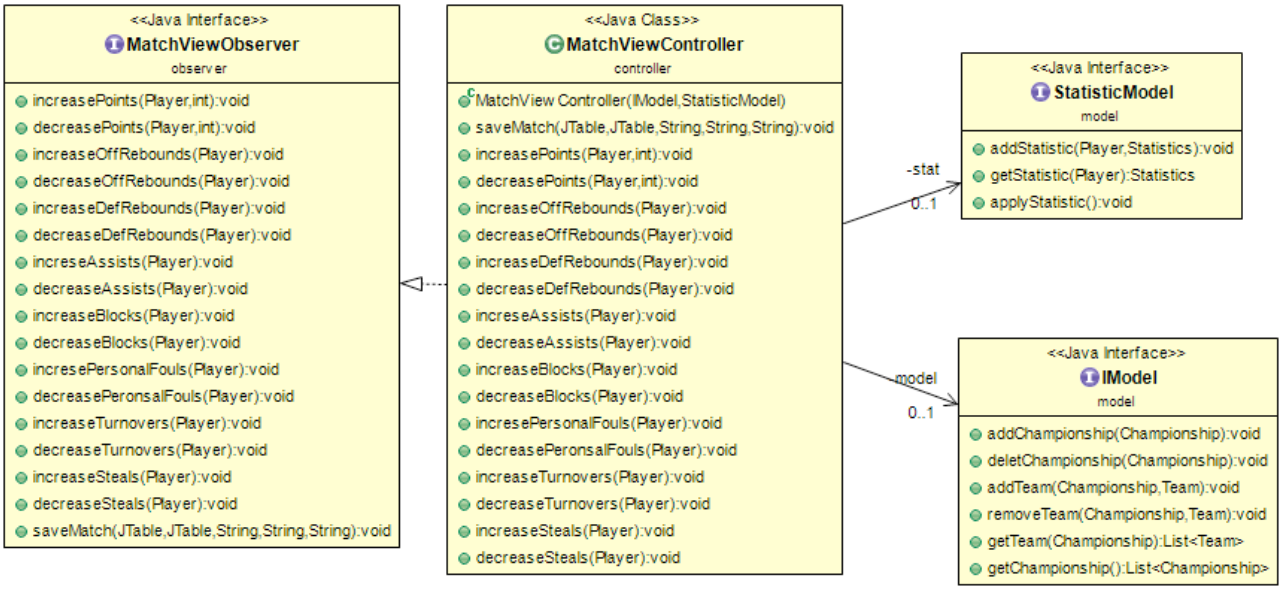
Descrizione: TeamController è la classe di controller utilizzata per gestire l'inserimento o la cancellazione di un nuovo Team all' interno di un Campionato, questa classe implementa i metodi descritti in "TeamObserver". Una volta preso come parametro dal modello il campionato a cui fare riferimento, la classe si occupa con il metodo "addTeam" di aggiungere una squadra e salvarla all'interno del modello, mentre con il metodo "removeTeam" si occupa di eliminarlo dal modello.

2.2.3.4. ComponentController



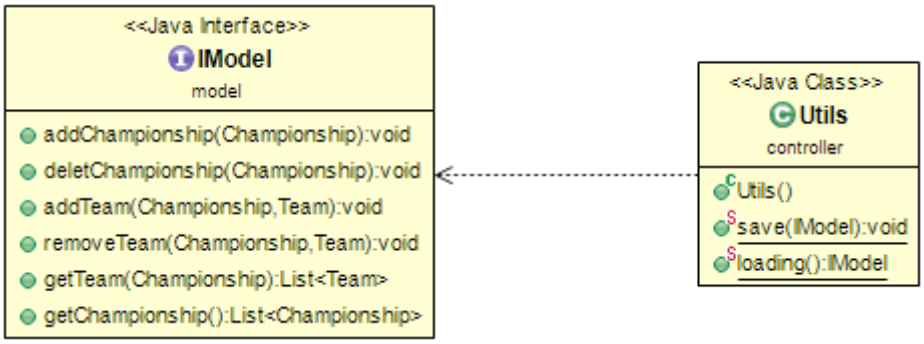
Descrizione: La classe ComponentController è la classe che si occupa, interfacciandosi ai metodi forniti da “TeamComponentObserver”, di prendere dal modello il team di riferimento e di creare e salvare successivamente sempre all’interno del modello il roster di giocatori inseriti e lo staff che compone il Team.
I suoi metodi principali sono “addPlayer” e “addStaff” che aggiungono rispettivamente al roster del team nuovi giocatori e nuovi componenti dello staff, implementa poi i metodi “removePlayer” e “removeStaff” che permettono, se esistenti, la rimozione di giocatori e staff dal modello.

2.2.3.5. MatchViewController



Descrizione: MatchViewController è la classe che si occupa di gestire, interfacciandosi ai metodi forniti da “MatchViewObserver”, una partita e assegnare ad ogni giocatore facente parte del roster di un team le statistiche del match, statistiche che verranno mostrate in contemporanea nella tabella dei team in corrispondenza del giocatore selezionato e salvate all’ interno del modello in modo da potere essere visualizzate nelle statistiche del giocatore stesso. La classe permette inoltre una volta terminato il match di salvare quanto fino a quel momento registrato all’ interno di un file excel.
Il metodo principale di questo controller, oltre naturalmente ai metodi di aggiunta e rimozione delle statistiche, è “saveMatch”. Questo metodo prese in ingresso le tabelle riguardanti i team a confronto si occupa di tutto ciò che riguarda il salvataggio sul file excel, per il quale sono state importate librerie esterne facenti parte del progetto “Apache POI”, definisce quindi la conformazione del file finale e ne gestisce il salvataggio.

2.2.3.6. Utils



Descrizione: Utils è la classe che si occupa esclusivamente della gestione del salvataggio di tutto ciò che viene inserito all'interno del modello e del suo successivo caricamento una volta aperto nuovamente il programma.
Il metodo "save" si occupa preso in ingresso il modello di stamparlo su un file "saveFIP.txt" . Il metodo "loading" invece controlla che tale file esista, se non esiste carica il file contenuto nelle risorse (già predisposto da noi).

Capitolo 3

3. Sviluppo

3.1. Organizzazione in package

- **controller**: contiene i codici sorgenti relativi alla parte di controller
- **exceptions**: contiene le eccezioni da noi sviluppate
- **mainPackage**: contiene solo la classe main che avvia l'applicazione
- **model**: contiene i codici sorgenti relativi alla parte di model e necessari al funzionamento dell'applicazione
- **observer**: contiene tutti gli observer delle view
- **tableModel**: contiene le classi relative alle table delle view
- **view**: contiene i codici sorgenti necessari alla creazione delle gui

3.2. Suddivisione del lavoro

Il lavoro è stato così suddiviso:

- **Model**: Luca Dal Seno
- **View**: Luca Dal Seno/Francesco Baiocchi
- **Statistic View**: Francesco Baiocchi
- **Controller**: Francesco Baiocchi

Subito dopo aver avuto l'approvazione da parte del professore, ci siamo incontrati per la progettazione del software e abbiamo stabilito sia l'entità utili allo sviluppo che la suddivisione delle view in modo da bilanciare il carico di lavoro. Grazie al pattern MVC è stato possibile lavorare rendendo minime le dipendenze fra i membri. BitBucket è stato ampiamente utilizzato, nonostante qualche problema iniziale nel capire il suo funzionamento. Una volta che il software era quasi interamente completato ci siamo incontrati e abbiamo discusso su eventuali migliorie e/o cambiamenti da effettuare.

3.3. Note di sviluppo

Nel realizzare l'interfaccia grafica si è fatto uso del plugin WindowBuilder che ha reso molto più facile e veloce il suo sviluppo. Sono state usate inoltre altre due librerie: Apache Poi per esportare le table delle statistiche sotto forma di file excel e jcalendar per utilizzare un calendario per far scegliere la data di nascita di un giocatore/staff.

Capitolo 4

4. Commenti Finali

4.1. Autovalutazione e lavori futuri

Il gruppo è soddisfatto del lavoro svolto poiché è riuscito ad implementare tutte le funzionalità base che si era preposto. Purtroppo per motivi di tempo è stata accantonata la parte relativa alla gestione delle società e degli sponsor (sarebbe stata implementata in maniera analoga alle altre) , che lasceremo come implementazione futura. Un'altra possibile miglioria sarebbe la gestione e il salvataggio di tutte le partite di un campionato e la creazione di un campionato.

Le difficoltà maggiormente incontrate come già detto sono state quasi tutte relative al funzionamento di bitBucket, per cui all'inizio lo sviluppo è andato un po' a rilento.

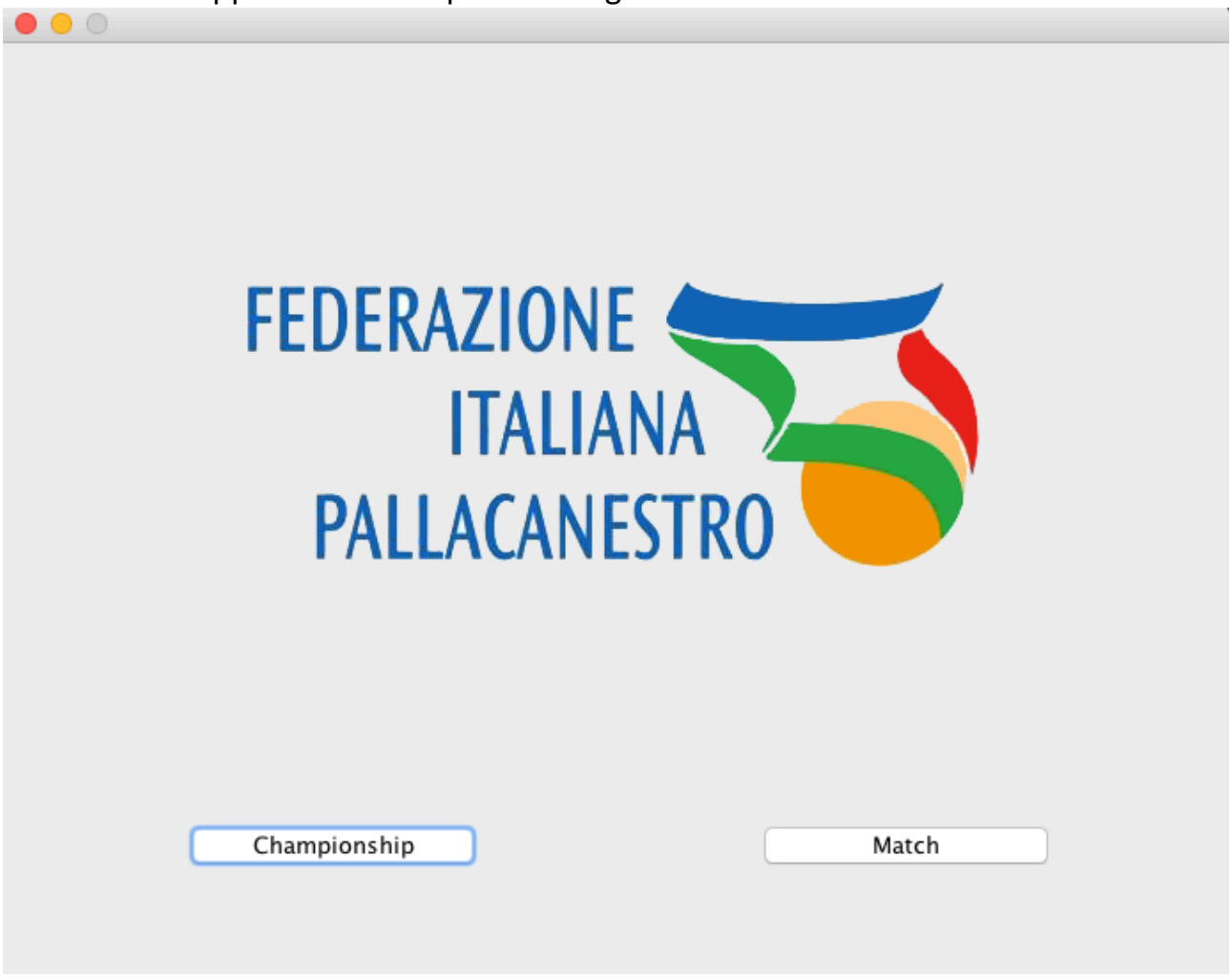
Luca Dal Seno: per quanto mi riguarda era la prima volta che lavoravo ad un progetto in maniera "scollegata" dal proprio collega. E' stato interessante da questo punto di vista affrontare un lavoro del genere, ma al tempo stesso è stato abbastanza faticoso (almeno inizialmente) essendo abituato a lavorare faccia a faccia. Riconosco inoltre che questo è il metodo migliore per portare avanti grossi progetti. Per quanto riguarda la parte di codice mi ritengo soddisfatto di quanto implementato.

Francesco Baiocchi: Dopo le prime difficoltà iniziali nell' organizzazione del lavoro e nell'utilizzo della piattaforma "Bitbucket" il lavoro insieme al mio collega è proceduto in maniera graduale e parallela senza il riscontro di problemi rilevanti. Essendo la prima volta nella quale mi sono confrontato con questo tipo di pattern di programmazione e questo metodo di lavoro posso dire di ritenermi soddisfatto dei risultati ottenuti da parte mia, niente meno che di quelli ottenuti dal mio collega.

Appendice How to

Guida all'utente

All'avvio dell'applicazione comparirà la seguente schermata:



per entrare nella parte relativa alla gestione delle squadre si clicchi su "Championship" e si aprirà una dialog di login le cui credenziali sono:

- User: adm
- Password : adm

Per quanto riguarda la gestione di una partita invece si clicchi su "Match" e si aprirà la stessa dialog di login le cui credenziali questa volta possono essere :

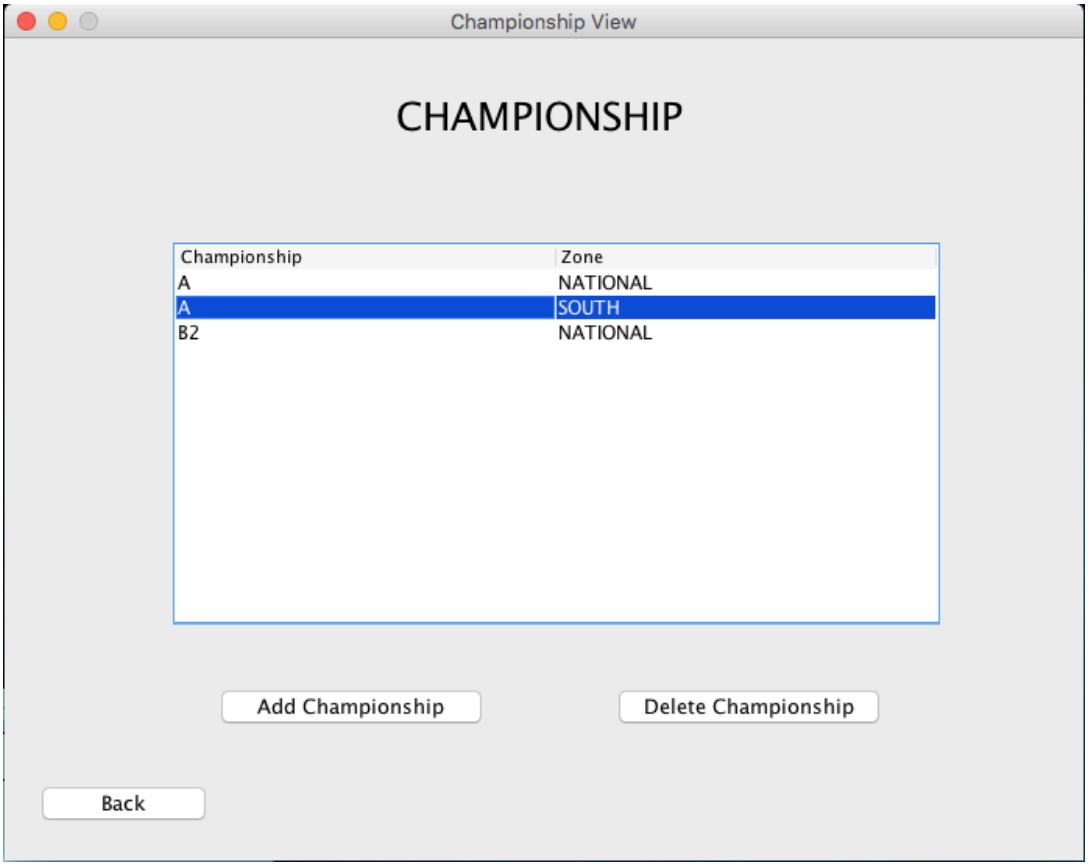
- User: adm

- Password: adm

Oppure

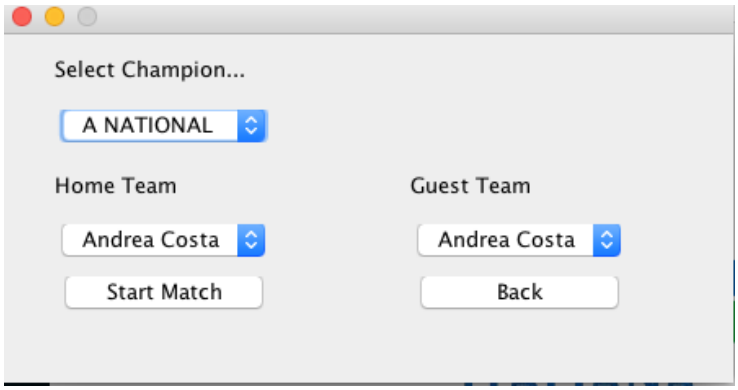
- User: user
- Password: user

In automatico verrà caricato il file di salvataggio (che forniremo alla consegna) e si verrò indirizzati alla pagina dei campionati.

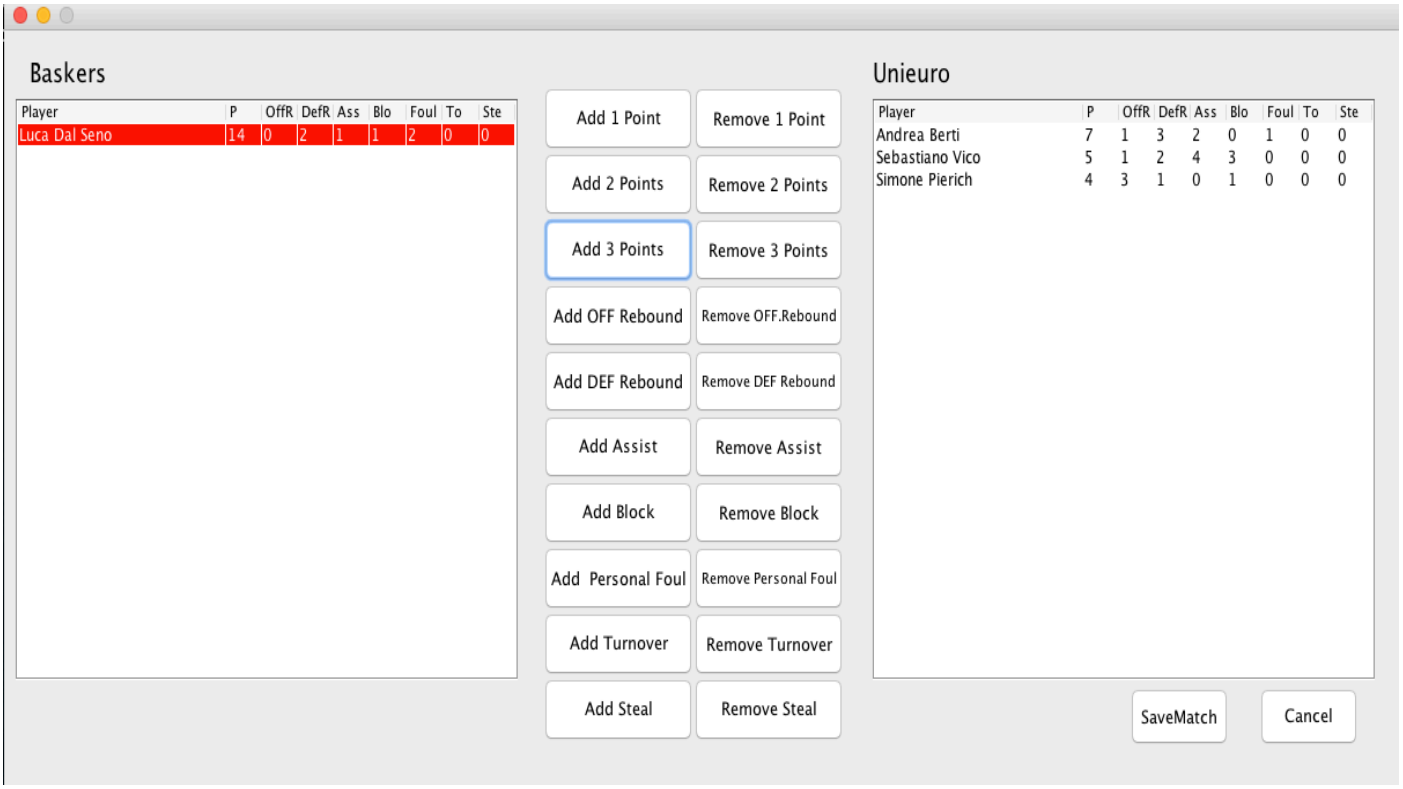


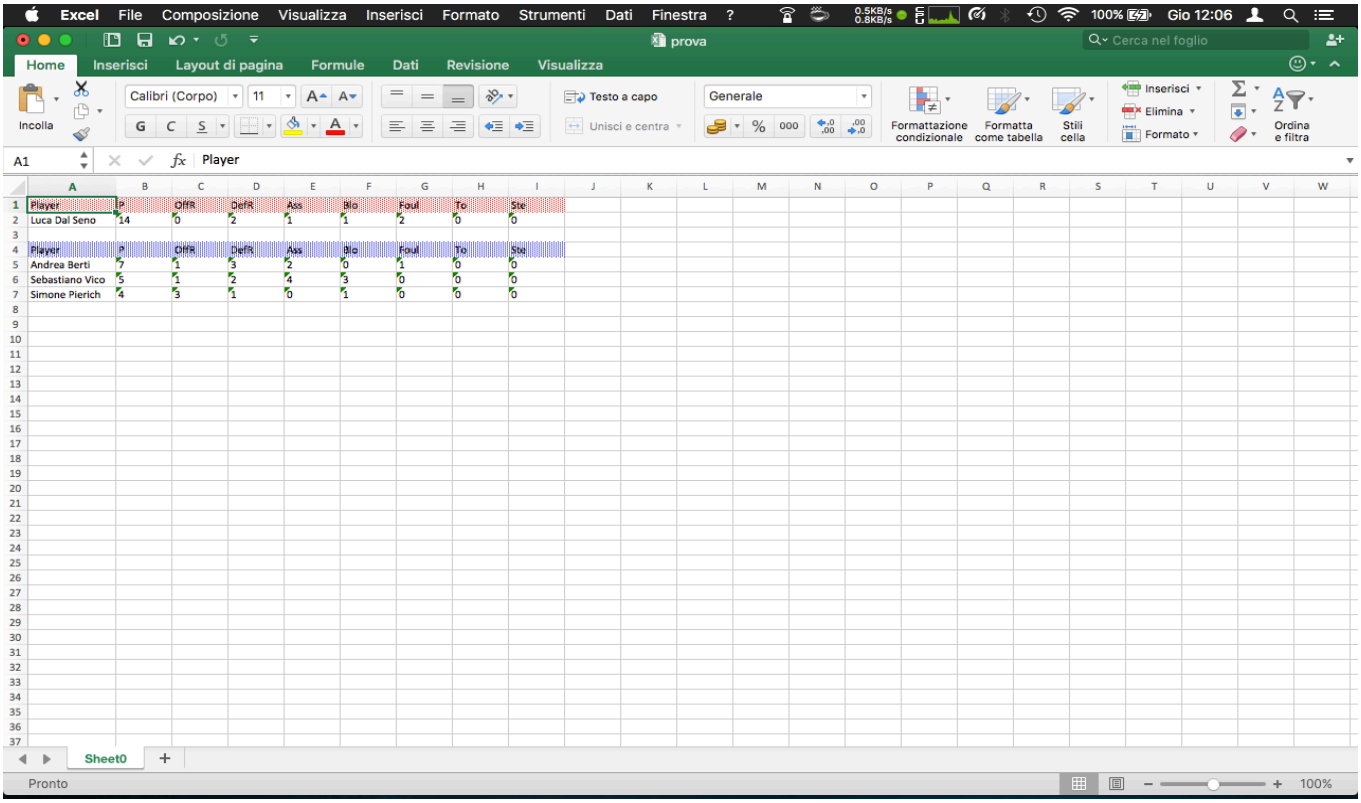
Facendo doppio click su un campionato si aprirà la schermata delle squadre relative al campionato (in modo analogo premendo sulle squadre si aprirà la lista dei giocatori e dello staff mentre premendo su un elemento della squadra si aprirà la possibilità di modificare i suoi campi e/o di visualizzare le statistiche). Per quanto riguarda tutti gli “add” è possibile premerci e compilare tutti i campi che vengono forniti, mentre per il tasto delete è necessario selezionare una riga (in qualsiasi view siamo).

Per quanto riguarda la matchView, una volta effettuato il login si dovrà selezionare prima il campionato, poi le due squadre che si dovranno affrontare.



Infine si aprirà la matchView: è sufficiente selezionare un singolo giocatore e aggiungere la statistica che si vuole aggiungere. Una volta terminato si prema su “Save Match” e si aprirà un path chooser per scegliere dove salvare la tabella excel.




Excel File Composizione Visualizza Inserisci Formato Strumenti Dati Finestra ?
100% Gio 12:06
prova
Cerca nel foglio
Home Inserisci Layout di pagina Formule Dati Revisione Visualizza
Incolla
Calibri (Corpo) 11
Testo a capo
Unisci e centra
Generale
Formattazione condizionale
Formatta come tabella
Stili cella
Inserisci
Elimina
Formato
Ordina e filtra
A1
Player
Player P OffR DefR Ass Blo Foul To Ste
Luca Dal Seno 14 0 2 1 1 2 0 0
Player P OffR DefR Ass Blo Foul To Ste
Andrea Berti 7 1 3 2 0 1 0 0
Sebastiano Vico 5 1 2 4 3 0 0 0
Simone Pierich 4 3 1 0 1 0 0 0
Sheet0
Pronto
100%