

# Università degli Studi di Salerno

Corso di Ingegneria del Software

## Football League Manager

Requirement Analysis Document

Versione 1.10



Data: 09/11/2016

## Partecipanti

Nome	Matricola
Nocerino Christian Giuseppe	0512103250
Manzo Gerardo	0512102974
Sarubbi Raffaele	0512102934
Capaldo Giovanni	0512102980
Iannuzzi Serena	0512103216

Scritto da	Nocerino Christian Giuseppe
------------	-----------------------------

## Revision History

Data	Versione	Descrizione	Autore
11/10/2016	1.0	Aggiunte sezioni: 1. Introduction (1.1, 1.3, 1.5, 1.6) 2. Proposed system (3.1, 3.2)	Christian Giuseppe Nocerino
12/10/2016	1.1	Aggiunte sezioni: 1. Proposed system (3.3)	1. Giovanni Capaldo 2. Gerardo Manzo
14/10/2016	1.2	Aggiunte sezioni: 1. Proposed system (3.4.1, 3.4.2, 3.4.5) Correzioni	1. Raffaele Sarubbi 2. Serena Iannuzzi
18/10/2016	1.3	Correzioni	Gerardo Manzo
21/10/2016	1.4	Aggiornate sezioni: 1. Introduction (1.1) 2. Proposed system (3.2)	Giovanni Capaldo
24/10/2016	1.5	Aggiornate sezioni: 1. Proposed system (3.4.1, 3.4.2 e 3.4.5)	1. Christian Giuseppe Nocerino 2. Raffaele Sarubbi

27/10/2016	1.6	Aggiornate sezioni: 1. Introduction (1.1) 2. Current system	Giovanni Capaldo
29/10/2016	1.7	Aggiornate sezioni: 1. Proposed system (3.1, 3.2, 3.3 e 3.4)	Gerardo Manzo
01/11/2016	1.8	Aggiornate sezioni: 1. Introduction (1.6)	Serena Iannuzzi
03/11/2016	1.9	Aggiornate sezioni: 1. Proposed system (3.4.2, 3.4.5) Aggiunte sezioni: 1. Proposed system (3.4.3, 3.4.4)	1. Christian Giuseppe Nocerino 2. Gerardo Manzo
09/11/2016	1.10	Aggiunte sezioni: 1. Introduction (1.2, 1.4) 2. Glossary	Serena Iannuzzi Raffaele Sarubbi

# Indice

<b>1. Introduction</b>	<b>5</b>
1.1. Purpose of the system	5
1.2. Scope of the system	5
1.3. Objectives and success criteria of the project	6
1.4. Definitions, acronyms, and abbreviations	6
1.5. References	6
1.6. Overview	6
<b>2. Current System</b>	<b>7</b>
<b>3. Proposed system</b>	<b>7</b>
3.1. Overview	7
3.2. Functional requirements	7
3.3. Nonfunctional requirements	8
3.3.1. Usability	8
3.3.2. Reliability	8
3.3.3. Performance	8
3.3.4. Supportability	8
3.3.5. Implementation	8
3.3.6. Interface	9
3.3.7. Packaging	9
3.3.8. Legal	9

3.4. System models	9
3.4.1. Scenari	9
3.4.2. Use case model	10
3.4.3. Object model	19
3.4.4. Dynamic model	21
3.4.5. User interface navigational paths and screen mock-ups	36
<b>4. Glossary</b>	<b>47</b>

# 1. Introduction

## 1.1. Purpose of the system

Un centro sportivo ha richiesto un sistema informatico per la gestione dei campionati di calcio a 5 organizzati all'interno della propria struttura. Il numero di squadre iscritte varia in base al campionato.

Il sistema viene utilizzato da 3 tipologie di utenti:

- il proprietario del centro che amministra il sistema (amministratore);
- gli allenatori che gestiscono la propria squadra;
- gli arbitri che inseriscono i referti delle partite.

Al momento dell'iscrizione la squadra deve essere composta da 8 giocatori. L'allenatore dovrà pagare la quota d'iscrizione per rendere effettiva l'iscrizione della sua squadra. Durante il campionato è possibile modificare la rosa.

Un calendario definisce l'ordine con il quale le squadre si affrontano. Ad ogni partita ciascuna squadra deve partecipare con almeno 5 giocatori, pena la sconfitta a tavolino. Non è posto alcun limite sul numero di sostituzioni.

Un utente non iscritto al sistema potrà soltanto aggiornarsi sugli sviluppi dei campionati. Non potrà quindi usufruire dei benefici di un utente registrato.

## 1.2. Scope of the system

Il sistema è mirato al miglioramento della gestione delle informazioni, dei servizi e delle altre comuni operazioni inerenti l'ambito gestionale di un campionato.

Il sistema, quindi, si prefigge i seguenti obiettivi:

- Mostrare agli utenti informazioni inerenti al campionato (come quelli riguardanti le partite, le classifiche, ecc.
- Facilitare gli utenti nella registrazione nel campionato
- Consultare informazioni sulle partite del campionato
- Notificare l'inizio delle varie partite all'interno di un campionato

### 1.3. Objectives and success criteria of the project

Il sistema si propone come obiettivo:

- Gestire le partite e la classifica di ogni campionato;
- Tenere traccia dei risultati delle squadre;
- Tenere traccia delle statistiche (goal realizzati, cartellini ricevuti etc...) dei calciatori

### 1.4. Definitions, acronyms, and abbreviations

Definizioni:

- assistman: Nel linguaggio sportivo, giocatore che contribuisce al buon esito di un'azione favorendo un suo compagno di squadra.
- marcatore: Nel linguaggio sportivo, il giocatore che realizza goal per la propria squadra.

### 1.5. References

- B. Bruegge, A.H. Dutoit, *Object Oriented Software Engineering – Using UML, Patterns and Java*, Prentice Hall, 3rd edition, 2009.

### 1.6. Overview

Il sistema offre la possibilità di iscriversi ai campionati organizzati in un centro sportivo. Il proprietario si occupa di aprire un nuovo campionato. Ad inizio campionato gli allenatori possono iscrivere le proprie squadre. Inoltre hanno la possibilità di visualizzare la rosa della propria squadra, il calendario e la classifica del campionato e infine le statistiche di ogni partita. Gli arbitri poi hanno il compito di inserire i referti delle partite, indicando il risultato, il numero di goal e assist di ogni giocatore e eventuali cartellini assegnati a questi ultimi.

Gli utenti non iscritti al sistema possono comunque tenersi aggiornati sull'andamento dei vari campionati.



## **2. Current System**

Al momento il centro sportivo non dispone di un sistema informatico per la gestione dei campionati, per cui il sistema proposto dovrà essere sviluppato ex novo.

## **3. Proposed system**

### **3.1. Overview**

In questa sezione viene fornita una panoramica dei requisiti funzionali e di quelli non funzionali del sistema proposto. Vengono quindi descritti gli scenari più significativi e i casi d'uso.

### **3.2. Functional requirements**

Il sistema deve offrire agli allenatori la possibilità di registrarsi. La registrazione degli arbitri è affidata al proprietario del centro sportivo che ha quindi il ruolo di amministratore.

L'amministratore ha il compito di:

- Creare un campionato e aprirne le iscrizioni (la chiusura avviene in automatico quando è stato raggiunto il numero di squadre prestabilito);
- Iscrivere un arbitro al sistema;
- Confermare l'iscrizione di una squadra.

Gli allenatori devono poter:

- Iscrivere la propria squadra ad un campionato;
- Visualizzare la rosa di ogni squadra;
- Aggiornare i dati di un giocatore;
- Sostituire un giocatore;
- Visualizzare il calendario;
- Visualizzare la classifica del campionato e le statistiche di ogni partita;

Gli arbitri hanno il compito di inserire il referto di ogni partita. Questo include:

- Il risultato della partita, il numero di goal e assist di ogni giocatore;
- I cartellini con relativa motivazione e nel caso di espulsione il numero di giornate di squalifica.

Le sostituzioni effettuate non vengono riportate in quanto non è posto alcun limite sul numero.

### **3.3. Nonfunctional requirements**

#### **3.3.1. Usability**

Le interfacce utente sono composte principalmente da messaggi testuali e/o moduli. Tutte le operazioni sono gestite mediante la compilazione e la sottomissione di tali moduli.

#### **3.3.2. Reliability**

Il sistema deve essere disponibile per la maggior parte del tempo. È previsto un server di backup in caso di malfunzionamento del sistema. Se il sistema risulta fuori servizio, esse dev'essere reso nuovamente disponibile entro 2 ore. In tal caso è comunque possibile usare altri metodi per usufruire dei servizi del sistema (email o moduli cartacei, da recapitare all'amministratore). L'amministratore dovrà registrare eventuali modifiche non appena il sistema ritorna disponibile.

Gli utenti devono registrarsi al sistema. Per eseguire le operazioni è richiesto l'accesso al sistema.

#### **3.3.3. Performance**

Il sistema deve rispondere agli input in tempo reale.

#### **3.3.4. Supportability**

È previsto lo sviluppo di un'app per cellulari che offre la possibilità di consultare le statistiche.

#### **3.3.5. Implementation**

Il sistema deve essere implementato in linguaggio Java.

### **3.3.6. Interface**

*Non applicabile al sistema proposto.*

### **3.3.7. Packaging**

Il sistema viene sviluppato come applicazione web. Potranno quindi usufruire del sistema tutti gli utenti che possiedono una connessione ad Internet e un web browser.

### **3.3.8. Legal**

Non è richiesta alcuna licenza per l'utilizzo del sistema.

## **3.4. System models**

### **3.4.1. Scenari**

#### **Creazione campionato "Holly & Benji"**

Francesco, il proprietario del sistema Football League Manager, ha deciso di organizzare un nuovo campionato dal nome "Holly & Benji". Per crearlo dovrà effettuare il login all'interno del sito inserendo la propria email e password. Una volta entrato, avrà a disposizione una pagina in cui solo lui può accedere; qui potrà svolgere le seguenti operazioni:

1. inserire il nome del campionato
2. decidere la quota di iscrizione
3. decidere il numero delle squadre
4. per ogni partita assegnare un arbitro

#### **Iscrizione squadra "Molossi"**

Jhonny vuole partecipare al campionato "Holly & Benji", per fare ciò dovrà accedere al sistema fornendo le credenziali (e-mail, password). A questo punto entrerà nella pagina dedicata al campionato dove potrà creare e registrare la propria squadra che verrà salvata e confermata da Francesco. Le uniche informazioni relative alla squadra saranno il nome della stessa e i dati anagrafici dei giocatori. Per completare l'iscrizione Jhonny dovrà pagare la quota d'iscrizione.

### Report delle statistiche per la partita Molossi - Tigers

Kumar (arbitro) avrà il compito di pubblicare le statistiche (risultato, goal e assist e cartellini) della partita da lui arbitrata sulla pagina del campionato. Il report avviene attraverso una sezione in cui vi saranno dei campi da compilare con le relative statistiche della partita.

### Fine del campionato "Holly & Benji"

Una volta terminate tutte le partite, alla squadra vincitrice verrà consegnato il premio "Golden Ball" (consiste nell'ammontare di tutte le quote raccolte dai vari partecipanti) che sarà consegnato direttamente da Francesco.

#### 3.4.2. Use case model

<b>ID</b>	<b>UC1</b>
<b>Name</b>	Creazione campionato
<b>Participating actors</b>	Amministratore
<b>Entry condition</b>	Il caso d'uso inizia quando l'amministratore seleziona dal menù l'opzione "Crea nuovo campionato".
<b>Exit condition</b>	Il campionato è stato creato e le iscrizioni sono aperte.
<b>Event flow</b>	<ol style="list-style-type: none"><li>1. L'amministratore inserisce il nome del campionato.</li><li>2. L'amministratore inserisce il numero di squadre che potranno iscriversi.</li><li>3. L'amministratore inserisce la quota di iscrizione.</li><li>4. L'amministratore conferma i dati inseriti.</li></ol>
<b>References</b>	figura 3.4.4.1, figura 3.4.5.2

<b>ID</b>	<b>UC2</b>
<b>Name</b>	Assegnazione arbitro
<b>Participating actors</b>	Amministratore
<b>Entry condition</b>	Il caso d'uso inizia quando l'amministratore seleziona dal menù l'opzione "Assegna arbitro".
<b>Exit condition</b>	L'arbitro è stato assegnato alla partita scelta.
<b>Event flow</b>	<ol style="list-style-type: none"> <li>1. L'amministratore seleziona una partita.</li> <li>2. L'amministratore seleziona un arbitro.</li> <li>3. L'amministratore conferma la scelta.</li> </ol>
<b>References</b>	figura 3.4.4.2, figura 3.4.5.3

<b>ID</b>	<b>UC3</b>
<b>Name</b>	Registrazione allenatore
<b>Participating actors</b>	Allenatore
<b>Entry condition</b>	Il caso d'uso inizia quando l'allenatore seleziona l'opzione "Registrati".
<b>Exit condition</b>	L'allenatore è registrato al sistema.
<b>Event flow</b>	<ol style="list-style-type: none"> <li>1. L'allenatore inserisce una email ed una password.</li> <li>2. L'allenatore inserisce le proprie generalità(Nome,Cognome)</li> <li>3. L'allenatore conferma i dati inseriti.</li> </ol>
<b>References</b>	figura 3.4.4.3, figura 3.4.5.1, figura 3.4.5.4

<b>ID</b>	<b>UC4</b>
<b>Name</b>	Iscrizione squadra ad un campionato
<b>Participating actors</b>	Allenatore
<b>Entry condition</b>	Il caso d'uso inizia quando l'allenatore seleziona l'opzione "Iscrivi una squadra".
<b>Exit condition</b>	La squadra è in attesa di conferma.
<b>Event flow</b>	<ol style="list-style-type: none"> <li>1. L'allenatore sceglie il campionato a cui intende partecipare.</li> <li>2. L'allenatore inserisce il nome della squadra.</li> <li>3. L'allenatore inserisce i dati anagrafici di ogni giocatore (il numero di giocatori varia in base al campionato scelto).</li> <li>4. L'allenatore conferma i dati inseriti.</li> </ol>
<b>References</b>	figura 3.4.4.4, figura 3.4.5.5

<b>ID</b>	<b>UC5</b>
<b>Name</b>	Registrazione arbitro
<b>Participating actors</b>	Amministratore
<b>Entry condition</b>	Il caso d'uso inizia quando l'amministratore seleziona l'opzione "Registra arbitro".
<b>Exit condition</b>	L'arbitro è registrato al sistema.
<b>Event flow</b>	<ol style="list-style-type: none"> <li>1. L'amministratore inserisce una email ed una password per l'arbitro.</li> <li>2. L'amministratore inserisce le generalità dell'arbitro.</li> <li>3. L'allenatore conferma i dati inseriti.</li> </ol>
<b>References</b>	figura 3.4.4.5, figura 3.4.5.6

<b>ID</b>	<b>UC6</b>
<b>Name</b>	Registrazione referto partita
<b>Participating actors</b>	Arbitro
<b>Entry condition</b>	Il caso d'uso inizia quando l'arbitro seleziona l'opzione "Registra referto".
<b>Exit condition</b>	Il referto è registrato sul sistema.
<b>Event flow</b>	<ol style="list-style-type: none"> <li>1. L'arbitro inserisce il risultato della partita.</li> <li>2. L'arbitro inserisce il numero di goal realizzati e gli assist.</li> <li>3. L'arbitro inserisce gli eventuali calciatori ammoniti/espulsi specificandone il motivo e in caso di espulsione la durata della squalifica.</li> </ol>
<b>References</b>	figura 3.4.4.6, figura 3.4.5.7a, figura 3.4.5.7b, figura 3.4.5.7c

<b>ID</b>	<b>UC7</b>
<b>Name</b>	Visualizzazione rosa
<b>Participating actors</b>	Allenatore
<b>Entry condition</b>	Il caso d'uso inizia quando l'allenatore seleziona l'opzione "Visualizza rosa".
<b>Exit condition</b>	La rosa viene mostrata all'allenatore.
<b>Event flow</b>	-
<b>References</b>	figura 3.4.4.7, figura 3.4.5.8

<b>ID</b>	<b>UC8</b>
<b>Name</b>	Modifica giocatore
<b>Participating actors</b>	Allenatore
<b>Entry condition</b>	Il caso d'uso inizia quando l'allenatore seleziona l'opzione "Modifica giocatore".
<b>Exit condition</b>	Le informazioni del giocatore sono state modificate.
<b>Event flow</b>	<ol style="list-style-type: none"> <li>1. L'allenatore seleziona il giocatore di cui intende aggiornare le informazioni.</li> <li>2. L'allenatore inserisce le nuove informazioni del giocatore.</li> <li>3. L'allenatore conferma i dati da inserire.</li> </ol>
<b>References</b>	figura 3.4.4.8, figura 3.4.5.8, figura 3.4.5.9

<b>ID</b>	<b>UC9</b>
<b>Name</b>	Rimuovi giocatore
<b>Participating actors</b>	Allenatore
<b>Entry condition</b>	Il caso d'uso inizia quando l'allenatore seleziona l'opzione "Rimuovi giocatore".
<b>Exit condition</b>	Le informazioni del nuovo giocatore sono state inserite.
<b>Event flow</b>	<ol style="list-style-type: none"> <li>1. L'allenatore seleziona il giocatore che intende rimuovere dalla squadra.</li> <li>2. L'allenatore inserisce le informazioni del nuovo giocatore.</li> <li>3. L'allenatore conferma i dati inseriti.</li> </ol>
<b>References</b>	figura 3.4.4.9, figura 3.4.5.8, figura 3.4.5.10



<b>ID</b>	<b>UC10</b>
<b>Name</b>	Visualizzazione classifica
<b>Participating actors</b>	Allenatore
<b>Entry condition</b>	Il caso d'uso inizia quando l'allenatore seleziona l'opzione "Visualizza classifica".
<b>Exit condition</b>	La classifica viene mostrata all'allenatore.
<b>Event flow</b>	-
<b>References</b>	figura 3.4.4.10, figura 3.4.5.11

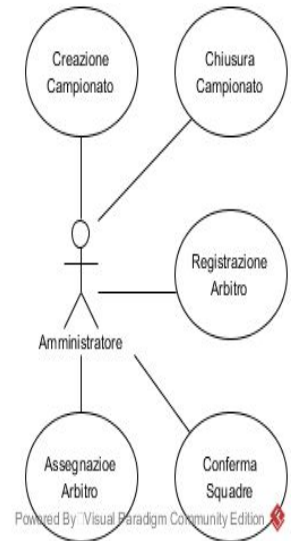
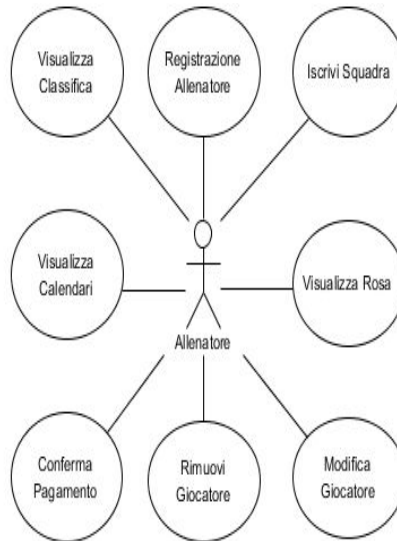
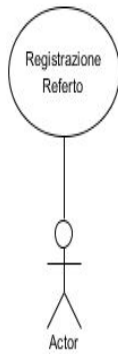
<b>ID</b>	<b>UC11</b>
<b>Name</b>	Visualizzazione calendario
<b>Participating actors</b>	Allenatore
<b>Entry condition</b>	Il caso d'uso inizia quando l'allenatore seleziona l'opzione "Visualizza calendario".
<b>Exit condition</b>	Il calendario viene mostrato all'allenatore.
<b>Event flow</b>	-
<b>References</b>	figura 3.4.4.11, figura 3.4.5.12

<b>ID</b>	<b>UC12</b>
<b>Name</b>	Chiusura campionato
<b>Participating actors</b>	Amministratore
<b>Entry condition</b>	Il caso d'uso inizia quando tutte le partite sono state disputate.
<b>Exit condition</b>	Il campionato è stato eliminato dal sistema.
<b>Event flow</b>	<ol style="list-style-type: none"> <li>1. L'amministratore consegna il premio alla squadra vincitrice.</li> <li>2. L'amministratore elimina il campionato dal sistema.</li> </ol>
<b>References</b>	figura 3.4.4.12, figura 3.4.5.14

<b>ID</b>	<b>UC13</b>
<b>Name</b>	Conferma squadra
<b>Participating actors</b>	Amministratore
<b>Entry condition</b>	Il caso d'uso inizia quando l'amministratore seleziona "Conferma squadra".
<b>Exit condition</b>	La squadra è pronta per l'iscrizione.
<b>Event flow</b>	<ol style="list-style-type: none"> <li>1. L'amministratore sceglie la squadra da confermare.</li> <li>2. L'amministratore conferma la scelta.</li> </ol>
<b>References</b>	figura 3.4.4.13, figura 3.4.5.13

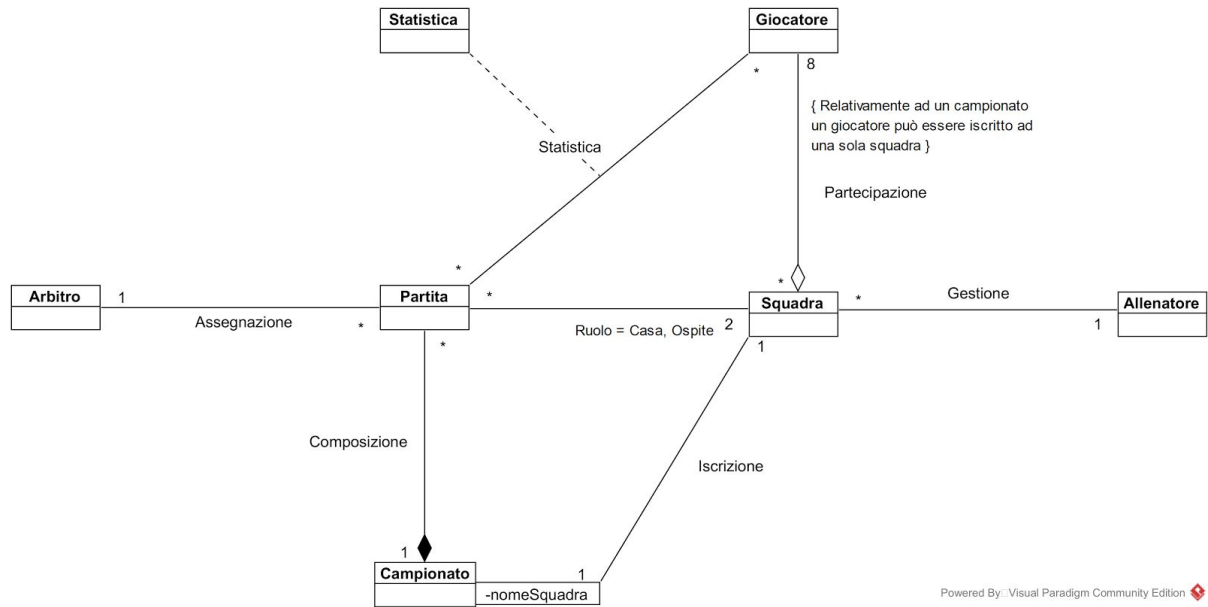
<b>ID</b>	<b>UC14</b>
<b>Name</b>	Conferma pagamento
<b>Participating actors</b>	Allenatore
<b>Entry condition</b>	Il caso d'uso inizia quando l'amministratore ha confermato la squadra e l'allenatore accede al sistema
<b>Exit condition</b>	La squadra è iscritta.
<b>Event flow</b>	<ol style="list-style-type: none"> <li>1. L'allenatore seleziona la tipologia di pagamento.</li> <li>2. L'allenatore inserisce i dati della carta (Numero carta e CVC)</li> <li>3. L'allenatore conferma i dati.</li> </ol>
<b>References</b>	figura 3.4.4.14, figura 3.4.5.15

## Use case diagram



### 3.4.3. Object model

#### Class Diagram



## Class Details

Squadra
-nomeSquadra
-vittorie
-pareggi
-sconfitte
-goalFatti
-goalSubiti
-statolScrizione
+getNomeSquadra()
+setNomeSquadra()
+getVittorie()
+setVittorie()
+getPareggi()
+setPareggi()
+getSconfitte()
+setSconfitte()
+getGoalFatti()
+setGoalFatti()
+getGoalSubiti()
+setGoalSubiti()
+getStatolScrizione()
+confermaSquadra()
+confermaPagamento()
+aggiungiGiocatore()
+rimuoviGiocatore()
+getRosa()

Giocatore
-cognome
-nome
+getCognome()
+setCognome()
+getNome()
+setNome()

Allenatore
-nome
-cognome
-email
-password
+getNome()
+setNome()
+getCognome()
+setCognome()
+getEmail()
+setEmail()
+getPassword()
+setPassword()

Arbitro
-nome
-cognome
-email
-password
+getNome()
+setNome()
+getCognome()
+setCognome()
+getEmail()
+setEmail()
+getPassword()
+setPassword()

Campionato
-nome
-numSquadre
-quota
+getNome()
+setNome()
+getNumSquadre()
+setNumSquadre()
+getQuota()
+setQuota()
+iscriviSquadra()

Partita
-giornata
-data
-risultato
+getGiornata()
+setGiornata()
+getData()
+setData()
+getRisultato()
+setRisultato()
+assegnaArbitro()

Statistica
-goal
-assist
-cartellino
-motivazione
-squalifica
+getGoal()
+setGoal()
+getAssist()
+setAssist()
+getCartellino()
+setCartellino()
+getMotivazione()
+setMotivazione()
+getSqualifica()
+setSqualifica()

### 3.4.4. Dynamic model

#### Sequence Diagrams

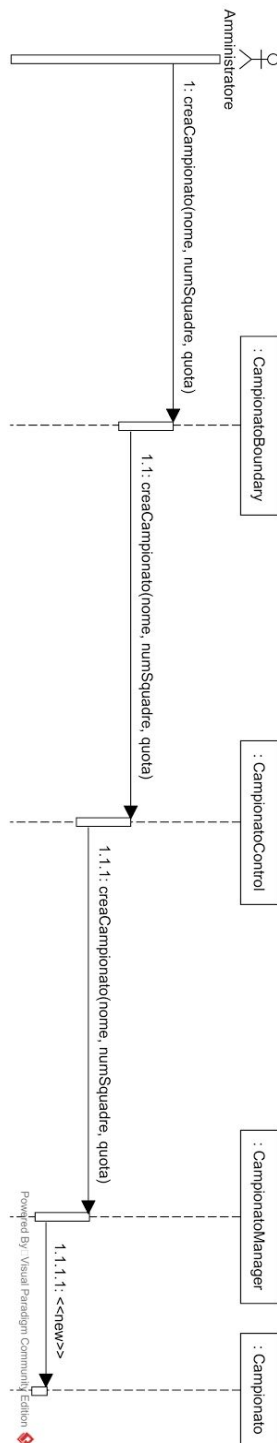


Figura 3.4.4.1

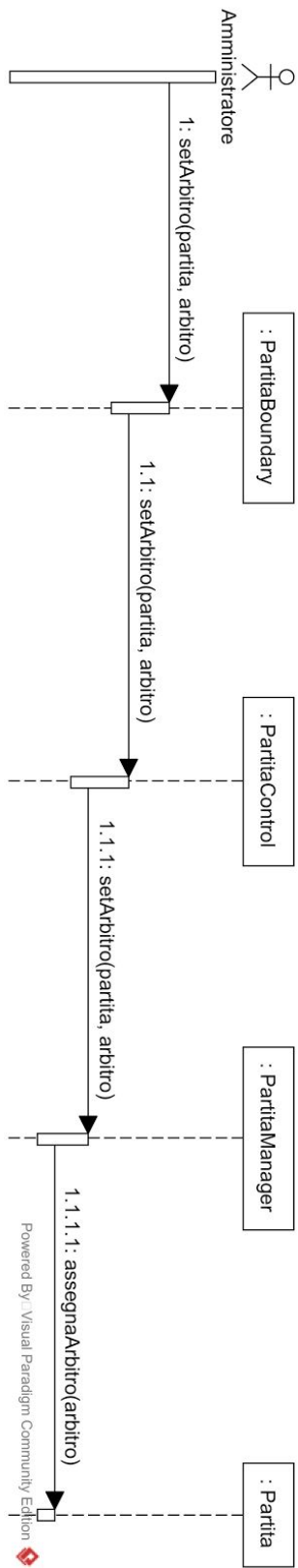


Figura 3.4.4.2



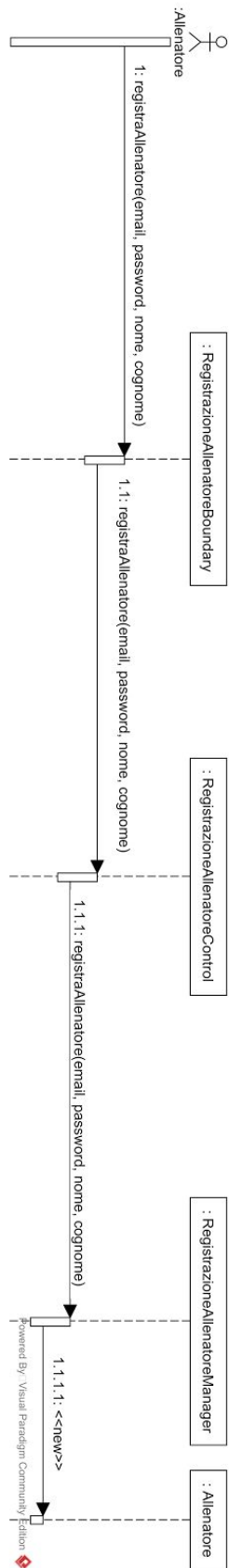


Figura 3.4.4.3

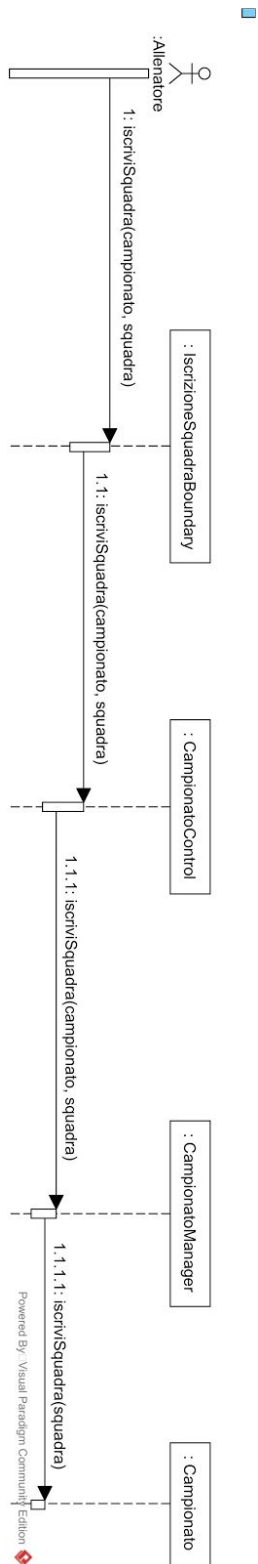


Figura 3.4.4.4

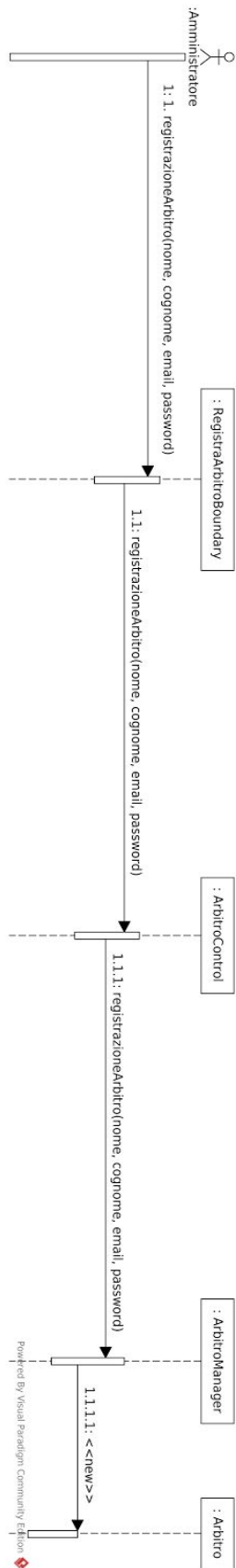


Figura 3.4.4.5

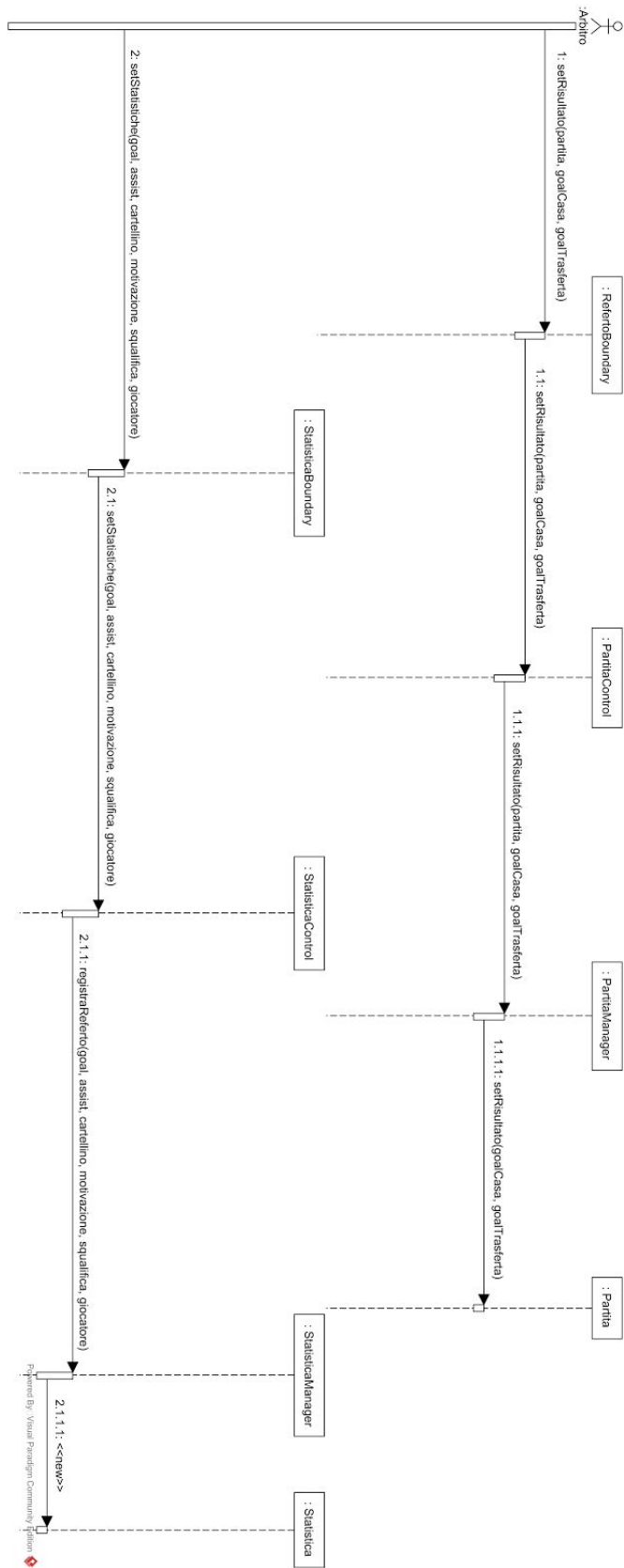


Figura 3.4.4.6

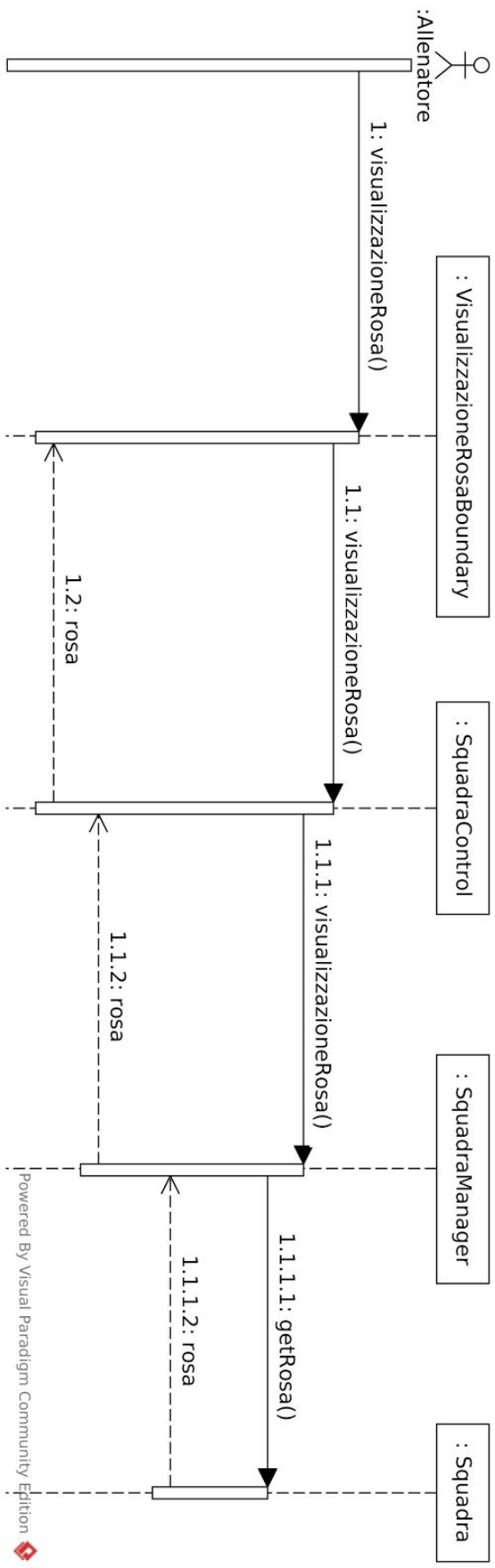
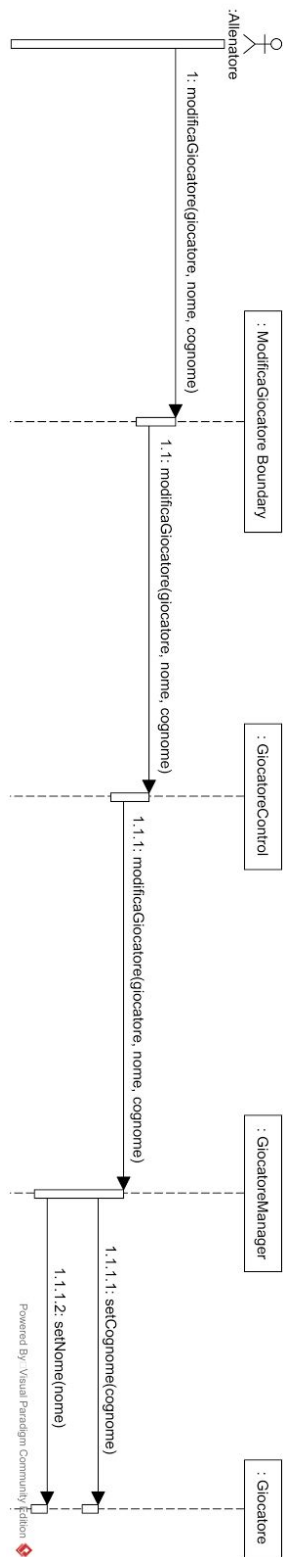


Figura 3.4.4.7



**Figura 3.4.4.8**

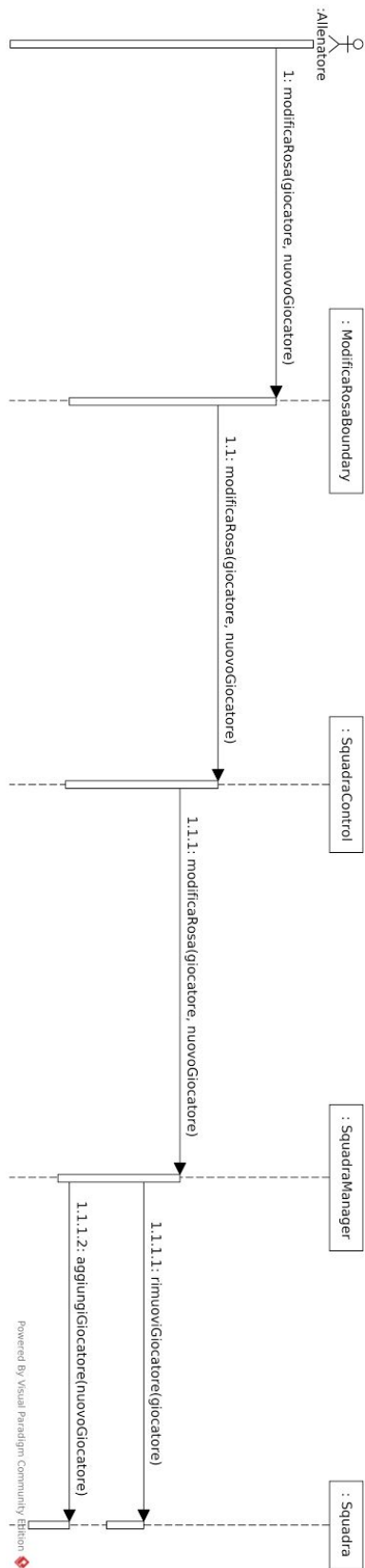


Figura 3.4.4.9

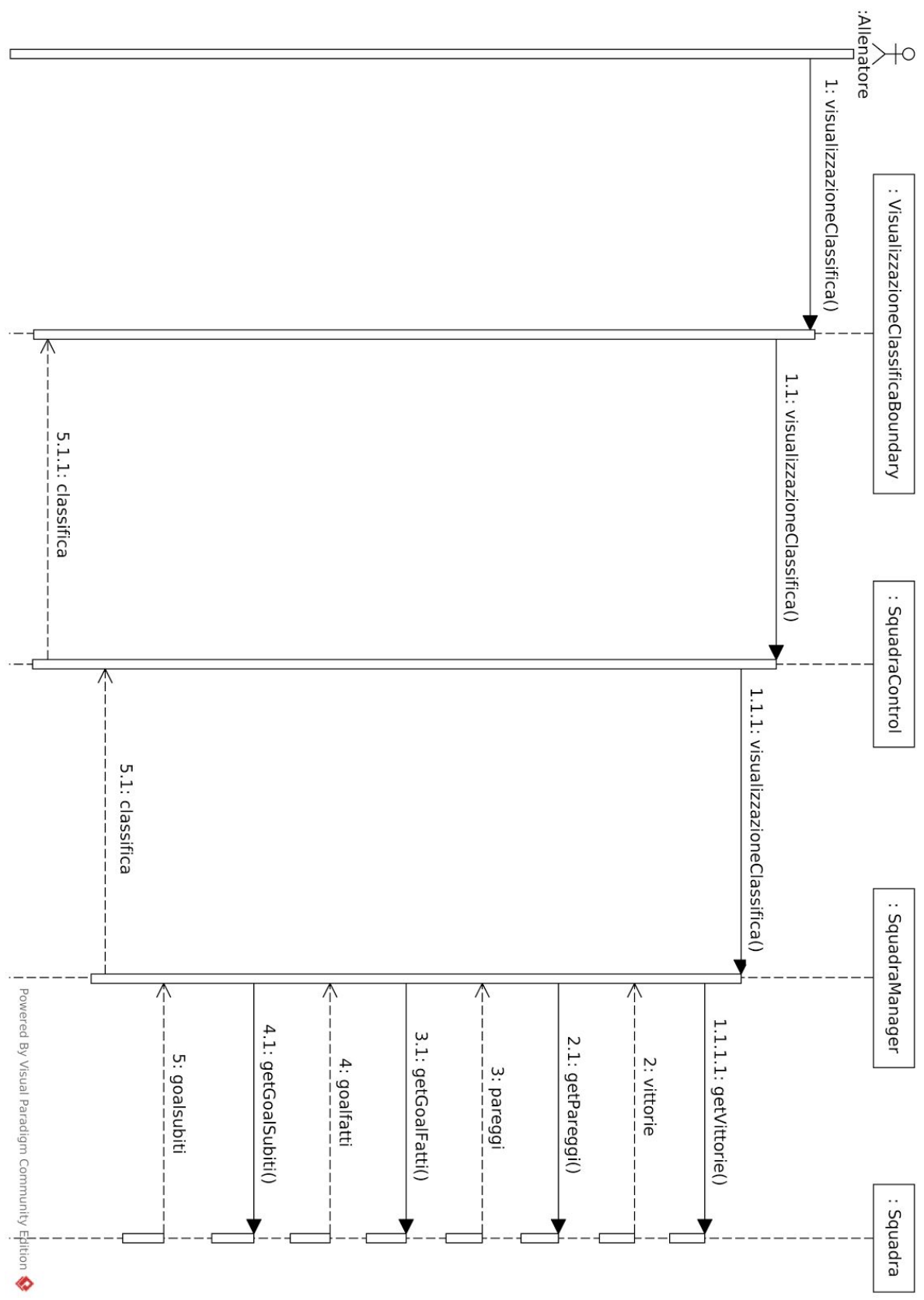


Figura 3.4.4.10



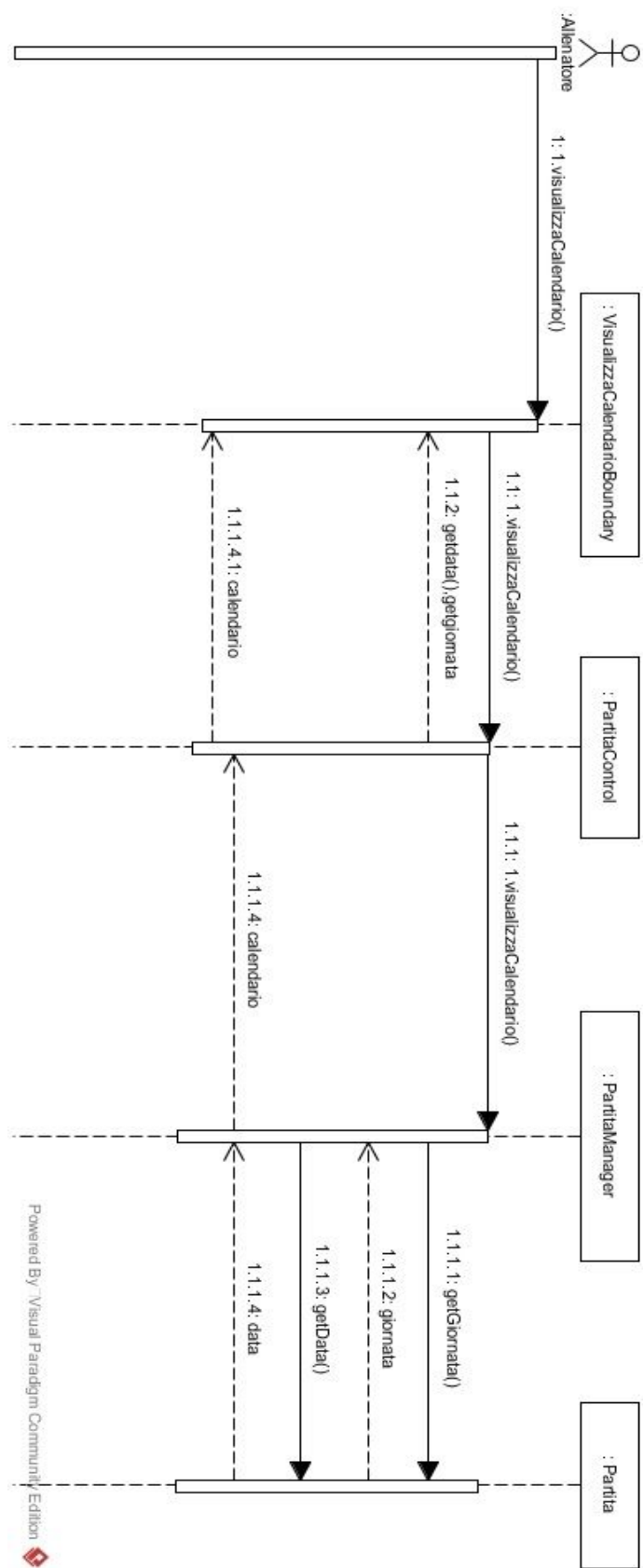


Figura 3.4.4.11

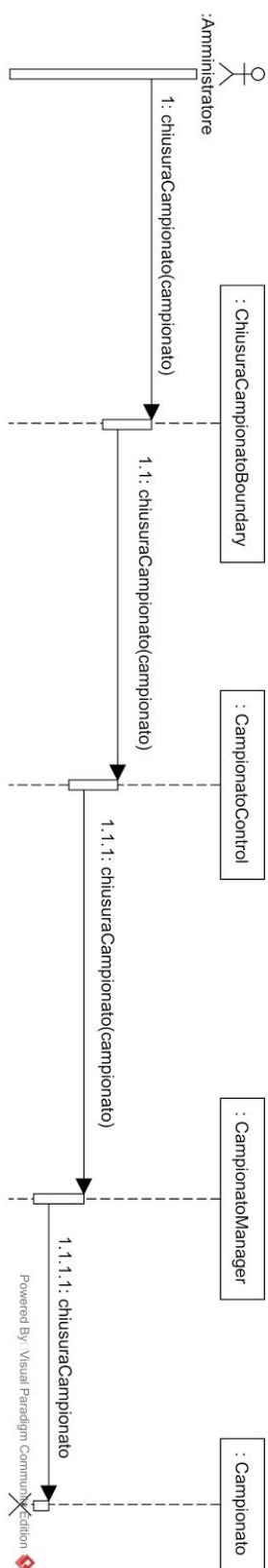


Figura 3.4.4.12

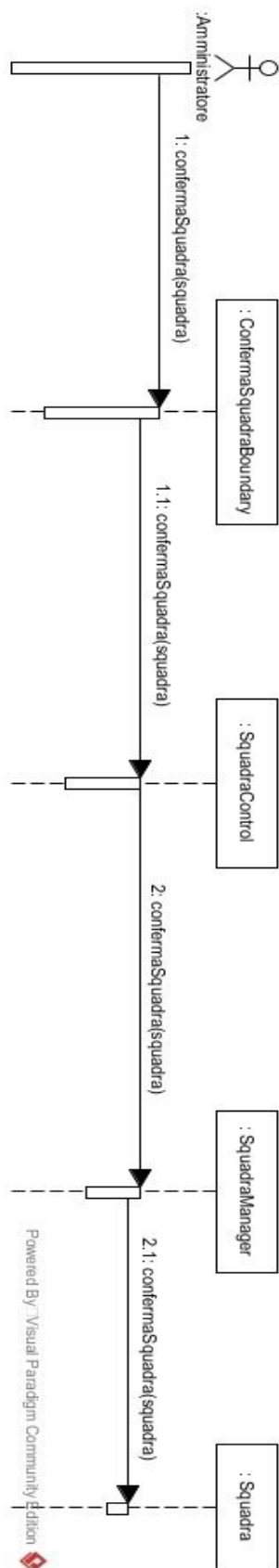


Figura 3.4.4.13

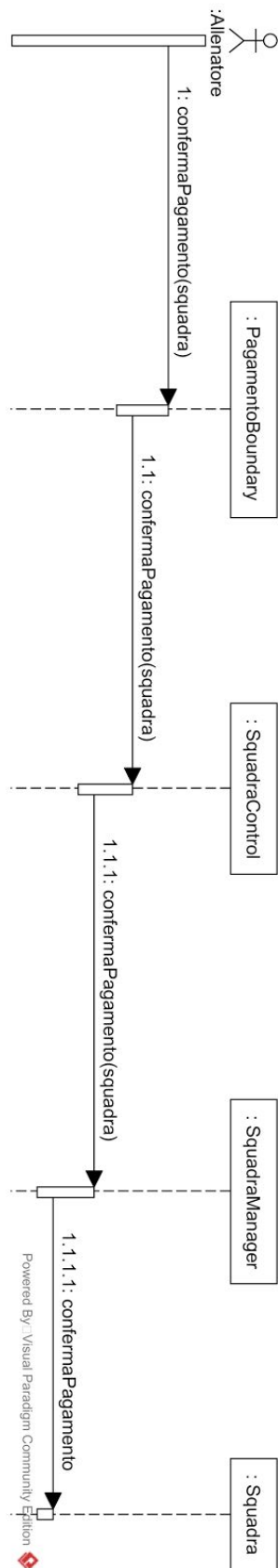
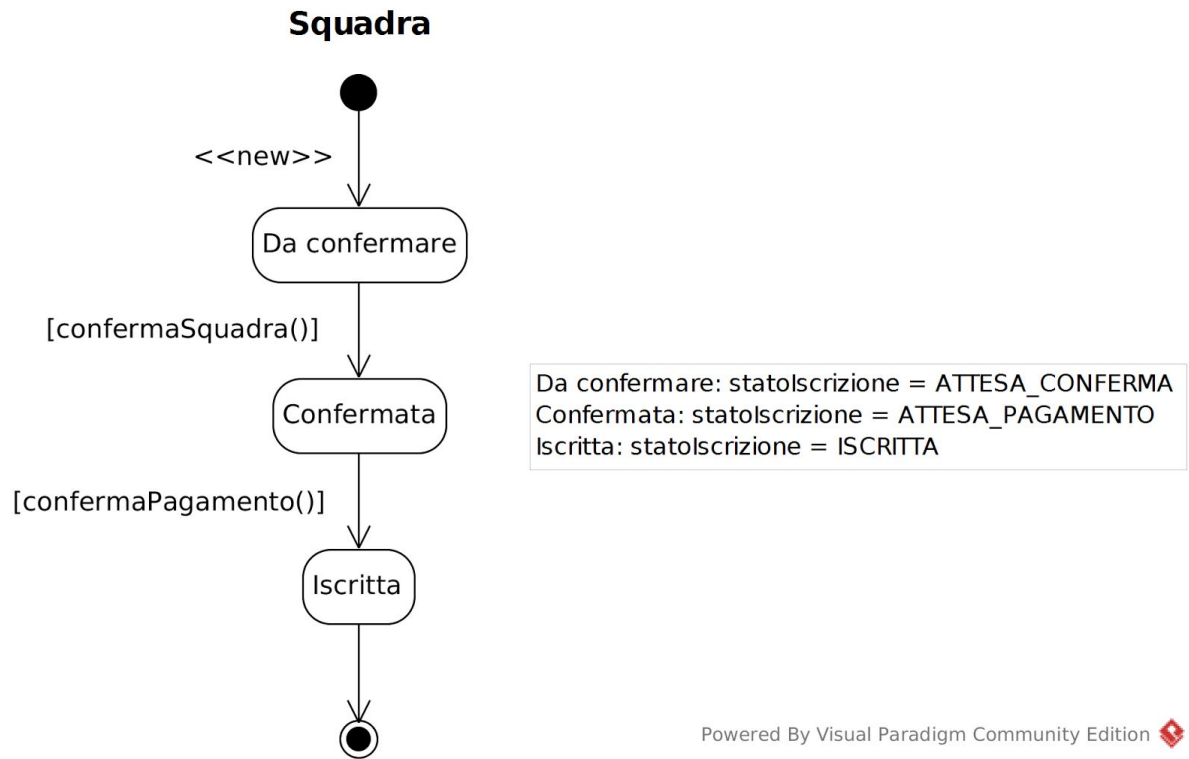


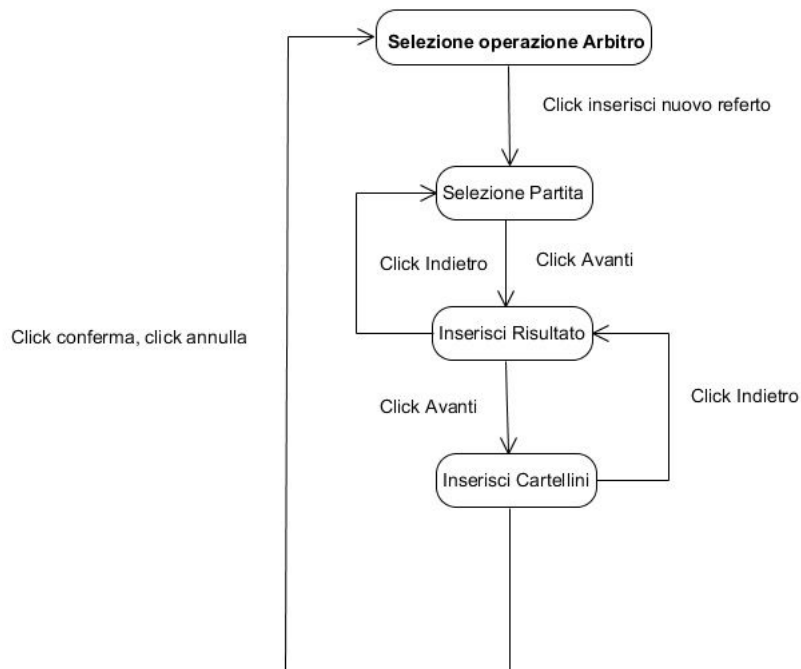
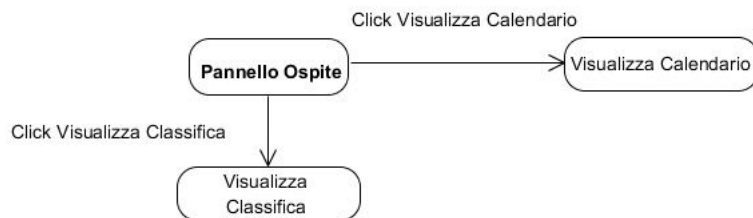
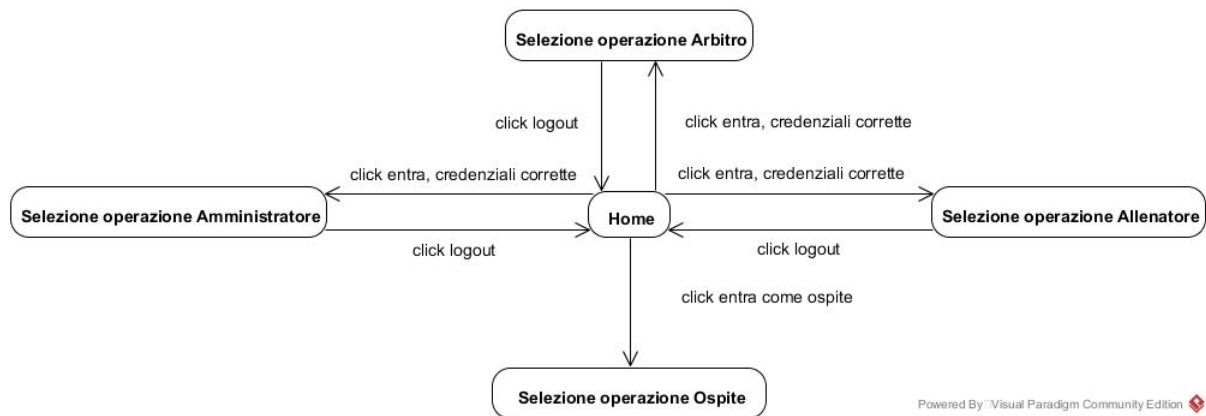
Figura 3.4.4.14

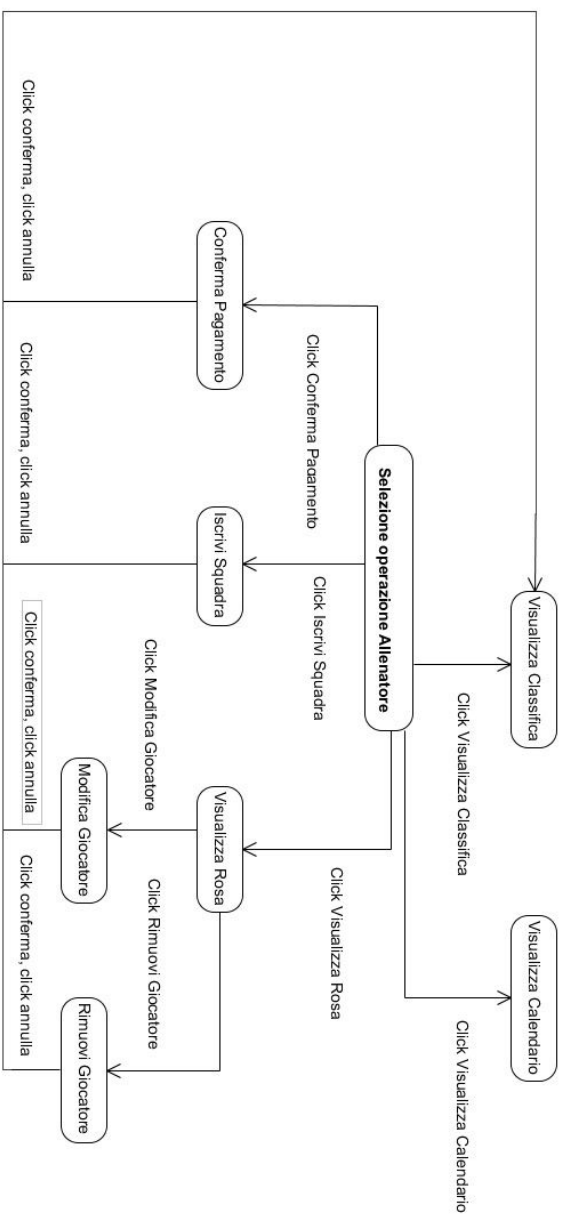
## Activity Diagram

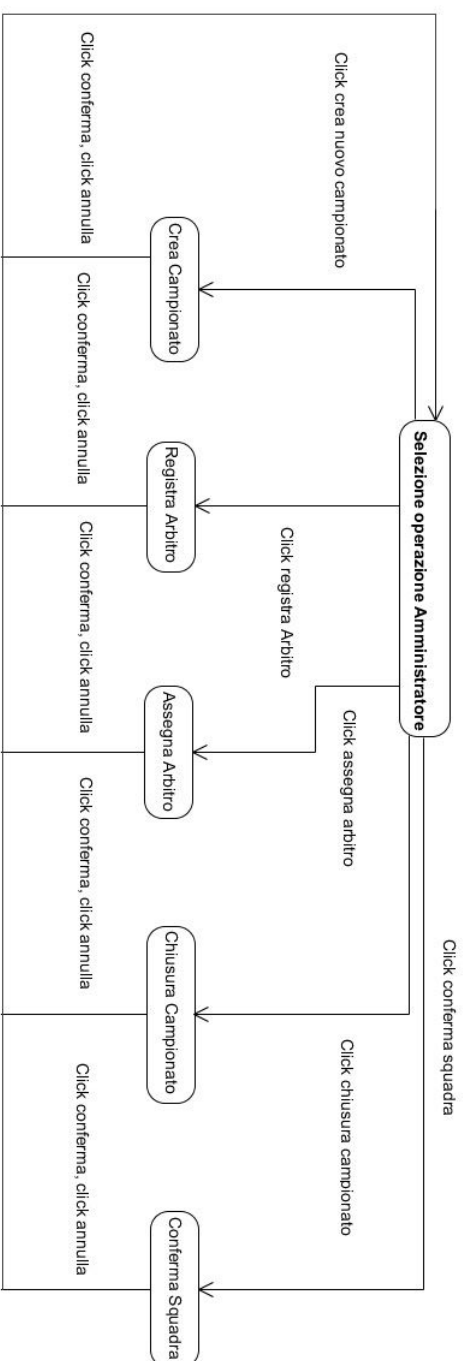


### 3.4.5. User interface navigational paths and screen mock-ups

#### Navigational paths









## Mock-ups



A login form mock-up for FLM. At the top center is a grey box labeled "LOGO". Below it is the text "FLM". The form contains two input fields: "Username" and "Password". Below the "Password" field is a button labeled "Entra". At the bottom, there is a button labeled "Registra nuovo allenatore" and a link labeled "Entra come ospite".

LOGO

FLM

Username

Password

[Entra come ospite](#)

figura 3.4.5.1



A registration form mock-up for FLM. At the top left is a grey box labeled "LOGO". To its right is the text "BENVENUTO FRANCESCO" and a power icon. Below the "LOGO" box is the text "CREAZIONE CAMPIONATO". The form contains three input fields: "Nome", "# squadre" (with a dropdown arrow), and "Quota iscrizione €". At the bottom, there are two buttons: "Conferma" and "Esci".

LOGO BENVENUTO FRANCESCO 

CREAZIONE CAMPIONATO

Nome

# squadre

Quota iscrizione €

figura 3.4.5.2

LOGO

BENVENUTO FRANCESCO

⏻

ASSEGNAZIONE ARBITRO

Arbitro

Kumar

Partita

AAAAA - BBBBB

CCCCC - DDDDD

Conferma

Rifiuta

figura 3.4.5.3

LOGO

BENVENUTO JHONNY

⏻

REGISTRA NUOVO ALLENATORE

Account:

Email

abcdef@uwxyz

Password

xxxxxxx

Generalità:

Nome

Giuseppe

Cognome

Rossi

Conferma

Esci

figura 3.4.5.4

LOGO

BENVENUTO JHONNY

⏻

Iscrizione squadra al campionato

▼

Holly e Benji

Nome squadra

Giacatore n. 1

Ripetuto per ogni giocatore

Nome

Cognome

Conferma

Esci

figura 3.4.5.5

LOGO

BENVENUTO FRANCESCO

⏻

REGISTRA NUOVO ARBITRO

Account:

Email

abcdef@uwx.yz

Password

xxxxxxx

Generalità:

Nome

Giuseppe

Cognome

Rossi

Conferma

Esci

figura 3.4.5.6

LOGO

BENVENUTO KUMAR

⏻

INSERISCI NUOVO REFERTO

SELEZIONA PARTITA ▼

AAAAA - BBBBB  
CCCCC-DDDDD

Avanti

figura 3.4.5.7a

LOGO

BENVENUTO KUMAR

⏻

RISULTATO: 0 - 0

MARCATORE

ASSISTMAN

▼

- ▼

MARCATORE

ASSISTMAN

▼

- ▼

Indietro

Avanti

figura 3.4.5.7b

LOGO

BENVENUTO KUMAR

⏻

CARTELLINI	NO	<div></div>	<div></div>	DESCRIZIONE
<hr/>	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	<div></div>
<hr/>	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	<div></div>
<hr/>	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	<div></div>

Indietro

Conferma

figura 3.4.5.7c

LOGO

BENVENUTO JHONNY

⏻

VISUALIZZA ROSA

Giocatore

<hr/>	<div></div>	<div></div>
<hr/>	<div></div>	<div></div>
<hr/>	<div></div>	<div></div>
<hr/>	<div></div>	<div></div>
<hr/>	<div></div>	<div></div>
<hr/>	<div></div>	<div></div>

figura 3.4.5.8

LOGO

BENVENUTO JHONNY

MODIFICA GIOCATORE

Nome

Giuseppe

Cognome

Rossi

Conferma

Esci

figura 3.4.5.9

LOGO

BENVENUTO JHONNY

ELIMINA GIOCATORE

Eliminazione di G.Rossi  
Inserire i dati del nuovo giocatore:

Nome

Cognome

Conferma

Esci

figura 3.4.5.10

LOGO

BENVENUTO JHONNY




VISUALIZZA CLASSIFICA

Squadre	Pt	G	V	P	S	Gf	Gs
Squadre	0	0	0	0	0	0	0
Squadre	0	0	0	0	0	0	0
Squadre	0	0	0	0	0	0	0
Squadre	0	0	0	0	0	0	0

figura 3.4.5.11

LOGO

BENVENUTO JHONNY



VISUALIZZA CALENDARIO

1° turno

AAAAA - BBBBB	17/11/2016	15:00
CCCCC - DDDDD	17/11/2016	18:00

2° turno

AAAAA - CCCCC	24/11/2016	15:00
BBBBB - DDDDD	24/11/2016	18:00

figura 3.4.5.12

LOGO

BENVENUTO FRANCESCO

⏻

CONFERMA SQUADRA

Xxxxxx ▼

Yyyyyy

Rosa Xxxxxx:

Conferma

Rifiuta

figura 3.4.5.13

LOGO

BENVENUTO FRANCESCO

⏻

CHIUSURA CAMPIONATO

HOLLY & BENJI ▼

Squadra

AAAAA ▼

BBBBB

CCCCC

Montepremio €

500.00

Conferma


Esci

figura 3.4.5.14




LOGO

BENVENUTO JHONNY



PAGAMENTO QUOTA ISCRIZIONE

Quota da pagare: 150€

Tipo di circuito 

Num. carta

CVC

**figura 3.4.5.15**

## 4. Glossary

### A

- Attore: è un'entità fuori dal sistema che deve essere modellata e che interagisce con il sistema.

### B

### C

- Class Diagram: Rappresenta le classi del sistema con i loro attributi e operazioni. Inoltre mostra le relazioni tra le classi (associazioni, relazioni e gerarchie di specializzazione/generalizzazione); può essere utilizzato a diversi livelli di dettaglio (nell'analisi e nel design del sistema).
- Entry condition: E' un vincolo che deve essere soddisfatto quando viene invocata per la prima volta un'operazione.
- Exit condition: E' un vincolo che deve essere soddisfatto quando termina l'operazione.

### D

- Diagram: Rappresentazione grafica di una collezione di elementi del modello. UML supporta i seguenti diagrammi: dei casi d'uso, di sequenza, di collaborazione, di attività, di stato, delle classi, degli oggetti, dei comportamenti e di dispiegamento.

### E

- Extend: Relazione tra un caso d'uso estendente e uno base, che specifica come il comportamento definito dallo use case estendente è incorporato nel comportamento del caso d'uso base. Extends viene usato anche per definire dei casi eccezionali di un caso d'uso.

### F

- FLM: Football League Manager, sistema che permette la creazione e la gestione di un campionato.

G

H

I

- Include: E' un tipo di relazione che serve a fattorizzare una parte del comportamento di un caso d'uso e metterlo in un altro caso d'uso.

L

- Login: La login è un'informazione che serve per accedere ad un sistema, è necessaria ma non sufficiente; ad essa è sempre associata ad una password che è rappresentata da una stringa alfanumerica.
- Logout: Rappresenta l'uscita dal sistema corrente che impone come prossima operazione un nuovo accesso al sistema con un account differente.

M

- Modello: E' un concetto astratto che descrive un sottoinsieme del sistema.
- Mock-ups: Produzione completa dell'interfaccia utente.

N

O

- Object Model: Modello di tipo UML che chiarisce le relazioni tra vari oggetti identificati per lo sviluppo del sistema software. Comprende anche le molteplicità, i versi e gli attributi delle relazioni tra gli oggetti identificati in fase di progettazione.

P

Q

## R

- Requisiti non funzionali: Requisito che specifica proprietà richieste al sistema, come vincoli ambientali e di sviluppo, prestazioni, dipendenze dalla piattaforma, di manutenibilità, estensibilità, sicurezza e affidabilità. Requisito che sancisce vincoli di carattere fisico relativi ai requisiti funzionali.
- Requirements analysis: Insieme di dati utili a chiarire al cliente tutto ciò che concerne le decisioni iniziali, i requisiti, gli obiettivi e i modelli relativi alla progettazione del sistema software da sviluppare.

## S

- Scenario: Un'istanza di un caso d'uso.
- Sequence Diagram: È utilizzato per definire la logica di uno scenario (specifica sequenza di eventi) di un caso d'uso. È uno dei principali input per l'implementazione dello scenario; mostra gli oggetti coinvolti specificando la sequenza temporale dei messaggi che gli oggetti si scambiano.
- Statechart Diagram: È normalmente utilizzato per modellare il ciclo di vita degli oggetti di una singola classe; mostra gli eventi che causano la transizione da uno stato all'altro, le azioni eseguite a fronte di un determinato evento.

## T

## U

- UML: È uno standard usato per modellare software orientato agli oggetti.
- Use Case: Rappresenta un possibile modo di utilizzo del sistema e descrive le interazioni tra esso e gli attori.
- Use Case Diagram: Diagramma che serve a descrivere il comportamento funzionale del sistema da come è percepito dall'utente.
- Utente non registrato: Colui che interagisce con il sistema per visualizzare le partite e le classifiche di un campionato.
- Utente Registrato: Colui che interagisce col sistema per usufruire dei servizi che il sistema offre.

V

Z