Università degli Studi di Salerno

Corso di Ingegneria del Software

Football League Manager

System Design Document

Versione 1.0

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	Gerardo Manzo

Revision History

Data	Versione	Descrizione	Autore
11/11/2016	1.0	Aggiunta sezione: 1. Introduction (1.1, 1.2, 1.3, 1.4, 1.5)	Gerardo Manzo
13/11/2016	1.1	Aggiunta sezione: 1. Current software architecture	Christian Giuseppe Nocerino
17/11/2016	1.2	Aggiunta sezione: 1. Proposed software architecture (3.1, 3.2, 3.3)	Giovanni Capaldo
20/11/2016	1.3	Aggiunta sezione: 1. Proposed software architecture (3.4, 3.5)	1. Raffaele Sarubbi 2. Serena Iannuzzi
23/11/2016	1.4	Aggiunta sezione: 1. Subsystem services	1. Gerardo Manzo 2. Giovanni Capaldo

Indice

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.

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 sulla partite del campionato
- Notificare l'inizio delle varie partite all'interno di un campionato

1.2 Design goals

Di seguito vengono elencati i diversi design goals del sistema FLM, estratti dai requisiti non funzionali del Documento di Analisi dei Requisiti ("*RAD_FootballLeagueManager*").

1.2.1. 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.

1.2.2. Security

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

1.2.3. Performance

Il sistema deve rispondere agli input in tempo reale.

1.2.4. Packaging

Il sistema viene sviluppato come applicazione web.

1.3 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.

1.4 References

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

1.5 Overview

Le sezioni successive di questo documento presenteranno l'architettura del sistema software FLM e la sua decomposizione in sottosistemi. Inoltre, verranno indicate le tipologie di utenti, i comportamenti e le funzionalità che il sistema offre ad ognuna di esse. Saranno descritti i requisiti per l'ambiente e le macchine che ospiteranno il sistema, informazioni sulla gestione dei dati persistenti e le politiche di sicurezza adottate dal sistema.

2. Current software architecture

Al momento il centro sportivo non dispone di un sistema informatico per la gestione dei campionati, per cui l'architettura software dev'essere sviluppata ex novo.

3. Proposed software architecture

3.1 Overview

Questa sezione descrive la decomposizione del sistema, l'assegnazione delle varie componenti del sistema alle macchine e le condizioni limite. La gestione dei dati persistenti viene trattata nel documento "*SDD_FootballLeagueManager_PersistentDataManagement*".

3.2 Subsystem decomposition

Il sistema viene suddiviso nei seguenti sottosistemi:

- Gestione Utenti
- Gestione Calciatori
- Gestione Squadre
- Gestione Campionati
- Gestione Partite

Il sottosistema "*Gestione Utenti*" gestisce gli utenti registrati, cioè allenatori e arbitri; il sottosistema "*Gestione Calciatori*" gestisce i giocatori inseriti nel sistema; il sottosistema "*Gestione Squadre*" gestisce le squadre iscritte al sistema; il sottosistema "*Gestione Campionati*" gestisce i campionati; il sottosistema "*Gestione* Partite" gestisce le partite dei vari campionati.

3.3 Hardware/software mapping

Il sistema sarà distribuito su diverse macchine, seguendo un'architettura Client-Server:

- Una macchina per il Server, che si occuperà della logica applicativa
- Una macchina per il DBMS e il Database, che si occuperà della persistenza dei dati
- Più macchine Client, che accederanno ai servizi offerti dal Server

3.4 Persistent data management

La sezione riguardante la gestione della persistenza dei dati viene trattata nel documento "*SDD_FootballLeagueManager_PersistentDataManagement*".

3.5 Access control and security

Il sistema prevede l'accesso di quattro tipologie di utenti: amministratore, arbitro, allenatore e utente ospite. Non è richiesta la registrazione per gli utenti ospiti, ma le uniche operazioni a loro consentite sono la visualizzazione delle squadre, delle classifiche e dei calendari. Le altre tipologie di utenti accederanno al sistema tramite un meccanismo di autenticazione, inserendo username e password fornite al momento della registrazione. Gli utenti registrati, dopo essersi autenticati, verranno reindirizzati alla propria area privata.

Oggetto / Attore	Campionato	Partita	Squadra	Statistica	Giocatore	Allenatore	Arbitro
Amministratore	<<create>> getNome() getNumSquadre() getQuota()	assegnaArbitro() getGiornata() getData() getRisultato()	confermaSquadra() getNomeSquadra() getVittorie() getPareggi() getSconfitte() getGoalFatti() getGoalSubiti() getStatoIscrizione() getRosa()				
Allenatore	iscriviSquadra() getNome() getNumSquadre() getQuota()	getGiornata() getData() getRisultato()	<<create>> getNomeSquadra() getVittorie() getPareggi() getSconfitte() getGoalFatti() getGoalSubiti() getStatoIscrizione() confermaPagamento() aggiungiGiocatore() rimuoviGiocatore() getRosa()	getGoal() getAssist() getCartellino() getMotivazione() getSqualifica()	<<create>> getNome() setNome() getCognome() setCognome()	<<create>> getNome() getCognome()	getNome() getCognome()
Arbitro	getNome() getNumSquadre() getQuota()	getGiornata() getData() getRisultato()	getNomeSquadra() getVittorie() getPareggi() getSconfitte() getGoalFatti() getGoalSubiti() getStatoIscrizione() getRosa()	<<create>> getGoal() setGoal() getAssist() setAssist() getCartellino() setCartellino() getMotivazione() setMotivazione() getSqualifica() setSqualifica()	getNome() getCognome()	getNome() getCognome()	<<create>> getNome() getCognome()
Utente ospite	getNome() getNumSquadre() getQuota()	getGiornata() getData() getRisultato()	getNomeSquadra() getVittorie() getPareggi() getSconfitte() getGoalFatti() getGoalSubiti() getStatoIscrizione() getRosa()	getGoal() getAssist() getCartellino() getMotivazione() getSqualifica()	getNome() getCognome()	getNome() getCognome()	getNome() getCognome()

3.6 Boundary conditions

Avvio del sistema

L'amministratore avvia la macchina sulla quale risiede il server, la macchina sulla quale risiedono il DBMS e il Database e la macchina da utilizzare per backup in caso di malfunzionamento. Il sistema è pronto.

Malfunzionamento del sistema

In caso di malfunzionamento del sistema le informazioni verranno salvate sul server di backup. L'amministratore entro due ore risolve eventuali problemi e riavvia il server.

4. Subsystem services

Il sistema viene suddiviso nei seguenti sottosistemi:

- Gestione Utenti
- Gestione Calciatori
- Gestione Squadre
- Gestione Campionati
- Gestione Partite

Il sottosistema "*Gestione Utenti*" fornisce come servizi la registrazione degli allenatori e degli arbitri e la loro autenticazione.

Il sottosistema "*Gestione Calciatori*" fornisce come servizi l'iscrizione al sistema di un nuovo giocatore e la modifica dei dati ad essi associati.

Il sottosistema "*Gestione Squadre*" fornisce come servizi la creazione di una squadra, la sostituzione di un giocatore della rosa e la visualizzazione della rosa stessa.

Il sottosistema "*Gestione Campionati*" fornisce come servizi l'apertura di un nuovo campionato, la chiusura di un campionato esistente e conclusosi, l'iscrizione di una squadra ad un campionato e la visualizzazione dello stato di un campionato (classifica e calendario).

Il sottosistema "*Gestione Partite*" fornisce come servizi l'assegnazione di un arbitro ad una partita e la registrazione dei referti.

5. 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 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