



UNIVERSITÀ
DEGLI STUDI
FIRENZE

UNIVERSITÀ DEGLI STUDI DI FIRENZE

DIPARTIMENTO DI INGEGNERIA DELL'INFORMAZIONE

Architettura ed Implementazione di un sistema per il gioco del Fantacalcio

Autori:

Barbieri Andrea
Cappini Niccolò
Zanni Niccolò

N° Matricola:
7077249

Corso principale:

Ingegneria del Software

Docente corso:

Enrico Vicario

Indice

1	Introduzione generale	2
1.1	Statement	2
1.1.1	Actors	2
1.2	Architettura	2
1.2.1	Dipendenze	2
2	Strumenti utilizzati	2

Elenco delle figure

1 Introduzione generale

Se volete
yappare per
allungarla
fate pure

1.1 Statement

Il sistema progettato vuole ricreare una applicazione desktop per il gioco del Fantacalcio. I partecipanti al gioco sono in grado di entrare, creare e gestire le leghe ed i relativi fanta team, in particolare sono in grado di schierare una formazione per la prossima partita da giocare. Il sistema inoltre permette a varie testate giornalistiche di registrarsi e di dare i voti ai calciatori.

1.1.1 Actors

Il sistema prevede tre tipi di utenti distinti ognuno con capacità diverse:

- **FantaUser:** l'utente base dell'applicazione è in grado di entrare in leghe già esistenti o di crearne nuove compete con gli altri giocatori schierando la formazione migliore ogni giornata
- **Admin:** è un utente base che però è l'admin di una lega. Si occupa di generare il calendario, assegnare o rimuovere i vari calciatori ad i team e di calcolare i risultati delle partite alla fine della giornata
- **Newspaper:** è la testata giornalistica, si occupa di assegnare i voti ad i calciatori ogni giornata del campionato

1.2 Architettura

Il sistema è stato sviluppato in moduli ciascuno dei quali si occupa di compiti specifici. Ciò permette di separare le responsabilità rendendo il codice più organizzato e semplice da mantenere e da estendere.

Aggiungere
Figura, pro-
babilmente
quella ge-
nerale non
specifica va
creata

- **Business Logic:** contiene le classi che si occupano della logica di business. Tra queste ci sono i services che gestiscono le operazioni che i vari attori sono in grado di compiere e le classi che si occupano di gestire le transazioni con **JPA**
- **Domain Model:** contiene le entità annotate dell'applicazione
- **Repositories:** contiene le implementazione concreta dei repositories che si interfacciano con il database

Andre cor-
reggimi se
sbaglio su
Jpa. Inoltre
le interfacce
dei reposi-
tory vanno
bene qui o è
meglio spo-
starle?

1.2.1 Dipendenze

L'applicazione utilizza delle dipendenze esterne per la persistenza dei dati e per l'implementazione dei test.

- **JPA**
- **Junit**
- **Mockito**
- **h2Database**
- **Java Swing**

Aggiungi
descrizioni
e cose se
mancano

2 Strumenti utilizzati

Il codice è stato scritto in Java utilizzando **IntelliJ Idea** ed **Eclipse**. La **GUI** è stata scritta utilizzando **Java Swing** e **Window Builder** Per la costruzione degli **UML** è stato utilizzato **StarUML** mentre per il versionamento del codice è stato utilizzato **Github**. Inoltre sono stati usati vari **LLMS** come aiuto nella scrittura del codice, tra questi **Copilot**, **ChatGpt** e **Gemini**.

Andre ag-
giungi la de-
scrizione di
cosa può fare

Riferimenti bibliografici

[1]

Cosa ci mettiamo libro del vicario su jpa? poi?