# 1. Introduzione

HappyFields si propone di semplificare la pianificazione di match sportivi, al fine di incrementare l’interesse delle persone verso questi e di rendere meno lento il processo di organizzazione.

# 1.1 Object Design Goals

**Riusabilita’:**

Il sistema deve basarsi sulla riusabilita’ attraverso l’uso corretto di ereditarieta’ e design patterns.

**Robustezza:**

Il sistema deve risultare pronto a gestire situazioni impreviste ed errori attraverso la gestione delle eccezioni

**Incapsulamento:**

Il sistema deve garantire l’invisibilita’ degli aspetti implementativi attraverso l’uso di interfacce, rendendo possibile l’utilizzo delle singole componenti come black-box.

# 1.1.2 Object Trade-Off

# 1.2 Linee guida per la documentazione dell’interfaccia

Le linee guida includono una lista di regole da rispettare per la corretta costruzione di interfacce grafiche. Per queste si e’ fatto riferimento alle convenzioni proposte sul sito w3school:

* <https://www.w3schools.com/html/html5_syntax.asp>
* JAVA LO METTO? https://checkstyle.sourceforge.io/sun\_style.html

# 1.3 Definizioni, Acronimi e Abbreviazioni

Di seguito alcune definizioni di termini presenti successivamente nel documento

* **Package:** Raggruppamento di classi o interfacce
* **Design Pattern:** Template riutilizzabili di soluzioni a problemi ricorrenti
* **Interfaccia:** Insieme delle firme alle operazioni offerte dalla classe
* **View:** Nel modello MVC rappresenta cio’ che l’utente ha a disposizione per interagire con le funzioni applicative
* **lowerCamelCase:** La pratica di scrivere frasi in modo tale che, ad eccezione della prima, ogni parola inizi con lettera maiuscola e non abbia spazi con la precedente
* **UpperCamelCase:** La pratica di scrivere frasi in modo tale che, a partire della prima, ogni parola inizi con lettera maiuscola e non abbia spazi con la precedente

# 1.4 Riferimenti

Di seguito una lista di riferimenti ad altri documenti utili durante la lettura

* DA TROVARE ED AGGIUNGERE

# 2. Packages

In questa sezione viene mostrata la suddivisione del sistema in packages, in base a quanto definito nel documento di SDD. Tale suddivisione e’ determinata dalle scelte architetturali e ricalca la struttura convenzionalmente definita da un progetto Maven.

DA AGGIUNGERE MANO MANO IN FASE DI SVILUPPO