ODD

1. Introduzione
   1. Trade off

/\*testo\*/

* 1. Linee Guida

Per lo sviluppo saranno utilizzate le seguenti convenzioni

* + - Gli oggetti e i metodo saranno nominati a seconda della convenzione CummelCase
    - Gli oggetti entita saranno chiamati semplicemente con il nome dell'entita
    - Gli oggetti saranno nominati a seconda dell' entita di cui si occupano e del loro ruolo nell' applicazione per esempio TorneoService, UtenteController ecc...
    - le pagine html in thydeconposizionemeleaf saranno segnate completamente in minuscolo
    - i nomi dei package sono in minuscolo
  1. Definizione,Acronimi, abbrevazioni Acronimi:
     + RAD: Requirements Analysis Document
     + SDD: System Design Document
     + ODD: Object Design Document
  2. Reference
     + R.A.D.
     + S.D.D.
  3. Convenzioni di tecnologia( nome temporaneo)

La tecnologia che stiamo utilizzando porta alcune convenzioni dovute alla logica enterprise.

* 1. Design Pattern

i Design Pattern che abbiamo utilizzato sono:

Inversion of control

L'inversion of control è un pattern per cui un compon/ente di livello applicativo riceve il controllo da un componente appartenente a una libreria riusabile. Questo lo fa tramite un altro pattern cioè dependency Injection che tramite utilizzo di metadati istruisce i containter.

MVC

1. Package

MVC è un pattern che divide il codice in tre parti specifiche cioè il Model View e Controller . Il model gestisce i dati, la view si occupa della visualizzazione dei dati e i controller riceve i comandi dell'utente attraverso le view e li attua modificando lo stato del model e delle view

la deconposizione in package è la seguente:

<eventuale disegno>

il package gottaBattleEmAll , è un package in cui sono contenuti altri package che riguardano le classi Service, Repository, Controller, Configuration(config), Entity , si trova anche un file di startup per lo start del sistema

il package resources è composto dalle pagine html e dai package dei fragment e dal package static, rappresentano le view del sistema/\*da sistemare\*/

il package fragment contiene le sottopagine html che possibile ripetere su piu pagine il package static contiene le risorse statiche per le pagine html

il package repository contiene tutte le interfacce per la creazione delle repository per le operazioni di CRUD per ogni singola entità

il package controller contiene tutti gli oggetti controller che si occupano delle richieste Http

da parte dell'utente

il package configuration (per convenzione si chiamera config) contiene le configurazioni di sistema per operazioni che vengono fatte durante lo startup o la creazione di determinati Bean

il package entity conterrà le entità del sistema con eventuale mapping in sql tramite le annotazioni di spring

* 1. View gottaBattleEmAll
  2. View ...

ecc...

1. Interfacce di classe

//TODO inserire OCL