

TEMPLATE STRUTTURA RELAZIONE

Nota sulla documentazione da presentare: perché il progetto sia valutato non è sufficiente presentare un codice funzionante. Il codice va corredato da una relazione che non deve essere la mera lista dei diagrammi UML richiesti, ma deve contenere tutte le spiegazioni necessarie ad illustrare come è stata sviluppata la progettazione del sistema e modellata in termini di diagrammi UML.

Quello che segue è lo scheletro di una *possibile documentazione*.

1 Descrizione del problema

Questa sezione deve essere una versione sintetica del documento RASD (Requirements Analysis and Specification Document). In particolare, deve contenere, come minimo, (i) il glossario e (ii) la descrizione e specifica dei requisiti distinguendo tra requisiti funzionali e non funzionali. Va definito e usato in modo consistente un formato per la descrizione (e indicizzazione) dettagliata dei requisiti.

1.1 Analisi e specifica dei requisiti

1.2 Glossario

Definisce termini e acronimi utilizzati nel documento.

2 Progettazione del Sistema

2.1 Diagramma dei casi d'uso

Per ogni caso d'uso ci deve essere almeno una descrizione testuale del caso d'uso. Per i casi d'uso principali (almeno per i casi d'uso base) deve esserci anche la descrizione di un possibile scenario, eventualmente con l'indicazione delle possibili varianti.

2.2 Descrizione degli scenari

Tabella 1- Template per la descrizione di un caso d'uso.

Nome	
Scopo	
Attore/i	
Pre-condizioni	
Trigger	
Descrizione sequenza eventi	
Alternativa/e	
Post-condizioni	

2.3 Diagramma delle classi

In questo stadio ha senso che si riporti e si discuta il diagramma delle classi di progetto. Il diagramma delle classi di programma verrà presentato e discusso nella sezione relativa all'implementazione.

Si ricordi che comunque il diagramma riportato in questa sezione:

1. Deve essere tracciabile con tutte le altre viste.
2. Deve essere tracciabile con il codice nel caso si decida di riportare un unico diagramma delle classi (quindi solo quello di programma).
3. Verrà valutato guardando se i principi chiave della progettazione OO siano stati considerati (classi altamente coese, poco accoppiamento tra classi, uso di gerarchia tra classi, ecc.) e considerando le associazioni tra classi.

2.4 Diagrammi di sequenza

2.5 Diagrammi delle attività

2.6 Macchine di stato

La notazione utilizzata per esprimere le macchine di stato deve essere usata in modo corretto, ad esempio i *call event* devono essere tracciabili con i messaggi dei casi d'uso e i metodi delle classi. Lo stesso si applica per tutte le azioni che sono chiamate di metodi.

2.7 Diagramma dei componenti

3 Implementazione del sistema

3.1 Diagramma delle classi

In questa sezione va riportato il diagramma delle classi di programma.

3.1.1 Discussione dei Design Pattern utilizzati

I design pattern utilizzati non vanno semplicemente elencati, ma anche discussi: va giustificato il loro utilizzo ed eventualmente va specificato quali modifiche del diagramma delle classi abbia implicato il loro utilizzo.

3.2 Gestione dei dati persistenti

In questa sezione va descritta la struttura del database utilizzato per gestire la persistenza dei dati e il DBMS utilizzato.

3.3 Descrizione dell'Interfaccia Grafica

Vanno riportate alcune schermate dell'interfaccia grafica implementata.

Nel caso si sia fatta una validazione dell'input, i controlli effettuati e le loro finalità vanno spiegati in questa sezione.

3.4 Diagramma di deployment

3.5 Specifica e verifica dei vincoli

Vanno elencati i vincoli OCL facendo riferimento al diagramma delle classi definitivo.

Per almeno un piccolo sottoinsieme di vincoli OCL è opportuno mostrare come almeno siano stati mappati in JML.

Visto che l'utilizzo delle librerie per l'implementazione dell'interfaccia grafica richiedono una versione di Java superiore a quella che supporta JML, non è obbligatorio ma opzionale riportare una o più schermate che mostrino il risultato in caso di violazioni di vincoli.

3.6 Descrizione del testing

Vanno documentati i test driver in JUnit e specificato il criterio di copertura utilizzato.

3.7 Note per l'installazione e l'utilizzo

In questa sezione vanno indicati l'ambiente di sviluppo e di compilazione.

Vanno specificate le eventuali credenziali necessarie per l'accesso al database e eventuali account di test già presenti nel sistema.