



UNIVERSITA' DEGLI STUDI DI SALERNO
FACOLTA' DI SCIENZE MM.FF.NN.
CORSO DI LAUREA TRIENNALE IN INFORMATICA
CORSO DI INGEGNERIA DEL SOFTWARE

***Gestione del Portale dedicato al
Dottorato di Ricerca in
Informatica***



platform

ODD - Object Design Document 1.0

18/12/2015

1.0

Coordinatori del progetto

<i>Prof. Andrea De Lucia - Top Manager</i>
<i>Elisa D'Eugenio - Project Manager</i>
<i>Dario Di Nucci - Assistant Manager</i>

Partecipanti

<i>Giuseppe Picciocchi - Matricola 0512101820</i>
<i>Tommaso Minichiello - Matricola 0512102296</i>
<i>Matteo Nardone - Matricola 0512102270</i>
<i>Andrea Fedele - Matricola 0512102418</i>
<i>Armando Ariemma - Matricola 0512101778</i>
<i>Michele Santarsiere - Matricola 0512101766</i>

Revision History

Data	Versione	Descrizione	Autori
18/12/2015	1.0	Prima stesura	Matteo Nardone

Indice

- 1. Introduzione**
- 2. Packages**
- 3. Interfaccia delle classi**

1. Introduzione

1.1. Object Design Trade-offs

Nella fase di object design sorgono diversi compromessi e vi è la necessità di decidere. I trade-offs relativo al progetto PhdPlatform sono:

- **Comprensibilità vs Costi:** un aspetto molto importante è avere un codice comprensibile, anche per chi non ha collaborato ad un progetto; è utile quindi avere classi e metodi chiari e facilmente interpretabili, utilizzando l'indentazione, convenzioni per denominazioni e una documentazione appropriata del codice sorgente. Questa caratteristica comporterà un incremento del tempo per lo sviluppo del progetto;
- **Buy vs Build:** la piattaforma PhdPlatform si suddivide in moduli; è una buona norma cercare di trovare qualche modulo capace di includere la gran parte delle funzionalità, in modo tale da risparmiare tempo e risorse, riutilizzando moduli già esistenti;
- **Funzionalità vs Tempi di consegna:** occorre trovare un buon bilanciamento tra le varie modifiche da apportare alla piattaforma (come nuove funzionalità, aspetti grafici etc) e la data di consegna del progetto; ciò comporta che nell'imminenza della data di consegna vi dovranno essere tutte le funzionalità di base.

1.2. Linee guida per la documentazione delle interfacce

Al fine di rendere più uniforme la comunicazione tra i membri del team di sviluppo, e rendere i programmi più comprensibili, adottiamo specifiche linee guida e convenzioni relative alla documentazione delle interfacce.

Per maggiori dettagli sugli standard di codifica, si rimanda al documento di specifica delle linee guida per il codice Java.

Denominazione:

Le convenzioni sono molto utili nella comprensione del codice sorgente in quanto possono fornire informazioni anche attraverso la semplice funzione di identificatore, ad esempio, se è una costante, un package, o una classe.

Questa sezione fornisce una panoramica sulle convenzione di denominazione. Esse sono descritte singolarmente nella seguente tabella.

Identifier Type	Rules for Naming	Example
Classi	<p>I nomi delle classi dovrebbero essere sostantivi, scritti secondo il sistema Camel: CapitalizedWithInternalWordsA IsoCapitalized</p> <p>I nomi delle classi che modellano una eccezione devono terminare per "Exception":</p> <p>ClassNameEndsWithException</p> <p>I nomi delle classi devono essere semplici e descrittivi. Utilizzare parole intere, evitando acronimi e abbreviazioni, quando possibile.</p>	AddPublicationServlet
Metodi	<p>I metodi devono essere verbi, e devono seguire il sistema Camel, in cui la prima lettera di ogni parola è in maiuscolo, ad eccezione della prima:</p> <p>firstWordLowerCaseButInternal WordsCapitalized()</p>	getCognome() getAllPublication()
Variabili	<p>I nomi delle variabili devono essere brevi ma significativi. La scelta del nome di una variabile deve essere dettata dalla volontà di comunicarne la funzione. I nomi delle variabili deve seguire il sistema Camel, in cui la prima lettera di ogni parola è in maiuscolo, ad eccezione della prima:</p> <p>firstWordLowerCaseButInternal WordsCapitalized</p>	isAdmin idPhdStudent
Costanti	<p>I nomi delle costanti dovrebbero essere maiuscoli, con le parole separate da un</p>	TABLE_CYCLE

	underscore: UPPER_CASE_WITH_UNDERSCORES	
Parametri	I nomi dei parametri devono seguire le convenzioni definite per le variabili e, inoltre, devono iniziare con la lettera "p".	pMessage
Packages	I nomi dei package devono seguire il sistema Camel: firstWordLowerCaseButInternalWordsCapitalized	userManagement

Commenti:

Tutti i commenti devono essere scritti usando le convenzioni javadoc: `/** ... */`. Ogni classe e ogni metodo pubblico devono essere preceduti da un commento che ne descrive lo scopo, garantendo le invarianti, le istruzioni d'uso, e/o gli esempi. Tutti i commenti delle classi devono indicarne l'autore. Tutti i commenti dei metodi devono includere, inoltre, una descrizione degli effetti, delle note algoritmiche, dei parametri e del valore di ritorno.

1.3. Definizioni, acronimi e abbreviazioni

Definizioni:

Trade-off: Situazione in cui si pone una scelta di compromessa da due obiettivi ugualmente desiderabili, ma in conflitto tra di loro.

Packages: Insieme di classi java.

JavaDoc: Strumento che estrae dai commenti di un programma una documentazione dettagliata del codice.

Acronimi:

RAD: Requirement Analysis Document.

SDD: System Design Document.

ODD: Object Design Document.

Abbreviazioni:

1.4. Referenze

- *RAD*;

- *SDD*;

- *Linee guida codice Java*.

2. Packages

Il raggruppamento delle classi del progetto PhdPlatform si è sviluppato cercando di astrarre tutti gli oggetti presenti nella piattaforma; all'interno vi sono sia i manager, sia le servlet che i beans, per ogni oggetto. Di seguito i packages del sistema:

Packages
it.unisa.dottorato.curriculum
it.unisa.dottorato.cycle
it.unisa.dottorato.tutorate
it.unisa.dottorato.account
it.unisa.dottorato.autenticazione
it.unisa.dottorato.curriculumcic
it.unisa.dottorato.exception
it.unisa.dottorato.news
it.unisa.dottorato.phdCourse
it.unisa.dottorato.phdProfile.collaborations
it.unisa.dottorato.phdProfile.missions
it.unisa.dottorato.phdProfile.publications
it.unisa.dottorato.presence
it.unisa.dottorato.utility
it.unisa.integrazione.database
it.unisa.integrazione.database.exception
it.unisa.integrazione.database.utility

2.1. Descrizione dei packages

Package	Description
it.unisa.dottorato.curriculum	Package che contiene il bean, il manager, le servlet per la gestione dei curriculum
it.unisa.dottorato.cycle	Package che contiene il bean, il manager, le servlet per la gestione dei cicli
it.unisa.dottorato.tutorate	Package che contiene le servlet per la gestione dei tutor
it.unisa.dottorato.account	Package che contiene i beans, il manager, le servlet per la gestione degli account, dei docenti e dei dottorandi
it.unisa.dottorato.autenticazione	Package che contiene il manager, le servlet per la gestione dell'autenticazione
it.unisa.dottorato.curriculumcic	Package che contiene il bean, il manager, le servlet per la gestione delle coppie curriculum-ciclo
it.unisa.dottorato.exception	Package che contiene le eccezioni comuni del sistema
it.unisa.dottorato.news	Package che contiene il bean, il manager, le servlet per la gestione delle news
it.unisa.dottorato.phdCourse	Package che contiene i beans, il manager, le servlet per la gestione dei corsi, delle lezioni e dei seminari
it.unisa.dottorato.phdProfile.collaborations	Package che contiene il bean, il manager, le servlet per la gestione delle collaborazioni
it.unisa.dottorato.phdProfile.missions	Package che contiene il bean, il manager, le servlet per la gestione delle missioni
it.unisa.dottorato.phdProfile.publicatio	Package che contiene il bean, il

ns	manager, le servlet per la gestione delle pubblicazioni
it.unisa.dottorato.presence	Package che contiene il bean, il manager, le servlet per la gestione delle presenze
it.unisa.dottorato.utility	Package che contiene classi utili per il sistema
it.unisa.integrazione.database	Package che contiene la classe per la connessione al database
it.unisa.integrazione.database.exception	Package che contiene le classi per gestire le eccezioni sul database
it.unisa.integrazione.database.utility	Package che contiene classi utili per la gestione del database

3. Interfaccia delle classi

In allegato il file di javadoc.zip per l'interfaccia delle classi.