Immagine che contiene testo, Carattere, logo, design

Descrizione generata automaticamente

**Pagina riservata alla copertina del documento.**

**INDICE**

[**1.** **INTRODUZIONE**](#_Toc154844412)

1.1 Object design trade-offs…………………………………………………………………………………………………………………………………..

**1.2 Interface documentation giudlines…………………………………………………………………………………………………………………**

**1.3 Defininizioni, acronimi e abbreviazioni…………………………………………………………………………………………………………..**

**1.4 References……………………………………………………………………………………………………………………………………………………….**

[**2.** **PACKAGES**](#_Toc154844417)

[**3.** **CLASS INTERFACE GLOSSARY**](#_Toc154844434)

1. **INTRODUZIONE**

**1.1Object design trade-offs**

Comprensibilità vs Tempo

Il codice deve essere scritto in modo da essere quanto più comprensibile per agevolare modifiche future.

Pertanto, è essenziale fornire opportuni commenti, anche se ciò comporterà una maggiore richiesta di tempo nello sviluppo.

Sicurezza vs Efficienza

La sicurezza rappresenta un elemento importante nel sistema, come

indicato anche nei requisiti non funzionali. Pero’, dati i tempi di sviluppo

limitati, ci limiteremo ad implementare funzioni di sicurezza basati su

username e password.

Response Time vs Hardware

Il tempo di risposta rappresenta un fattore importante nel sistema, in

particolare su determinate funzionalità. Tutto ciò dipenderà però anche dal

tipo di hardware su cui verrà fatto eseguire il sistema.

**1.2 Interface documentation guidelines**

Gli sviluppatori seguiranno le seguenti linee guida per la definizione delle

interfacce :

- Classi Java e Servlet :

o I nomi dovranno iniziare con la lettera maiuscola

o Se il nome contiene più parole, ognuna di esse dovrà iniziare

con una lettera maiuscola

o I nomi dovranno corrispondere alle informazioni e/o

funzionalità che offre quella classe o Servlet

- Metodi :

o I nomi dovranno iniziare con la lettera minuscola

o Se il nome contiene più parole, ognuna di esse dovrà iniziare

con la lettera maiuscola

o I nomi dovranno corrispondere alle informazioni e/o

funzionalità che offre quel metodo. Si utilizzerà un verbo più

eventualmente aggettivi

o I nomi dei metodi per ottenere e settare attributi seguiranno la

regola di nominazione ‘get[nomeattributo]’ e

‘set[nomeattributo]’

- Variabili :

o I nomi delle variabili dovranno iniziare con la lettera minuscola

o Se il nome contiene più parole, ognuna di esse dovrà iniziare

con la lettera maiuscola

**1.3 Definizioni, acronimi e abbreviazioni**

|  |  |  |
| --- | --- | --- |
| **Acronimo** | **Abbreviazione** | **Definizione** |
| UI | User Interface | L’area visuale con cui gli utenti interagiscono per utilizzare il software. |
| DMBS | Database Management System | Un software che gestisce l'organizzazione, lo storage e l'accesso ai dati nel database. |
| HTTPS | HyperText Transfer Protocol over Secure Socket Layer | Un protocollo di comunicazione sicuro utilizzato su reti informatiche. |
| API | Application Programming Interface | Un insieme di regole e strumenti che consente a software diversi di comunicare tra loro. |
| Three-tier | Three-tier | L'architettura Three-tier è un modello di progettazione software che organizza un'applicazione in tre strati distinti: la presentazione, la logica di business e la gestione dei dati. Questi strati operano in modo separato, consentendo una maggiore modularità e facilitando la manutenzione e lo sviluppo delle applicazioni. |
| JSP | Java Server Pages | Una tecnologia Java utilizzata per creare pagine web dinamiche. |
| JDBC | Java Database Connectivity | Una API Java che consente l’accesso e la gestione dei database relazionali. |
| URL | Uniform Resource Locator | L'indirizzo web che identifica una risorsa su Internet. |
| RAD | Requirements Analaysis Document | documento che raccoglie e analizza i requisiti di un progetto software |

**1.3 References**

RAD : Requirement Analysis Document

SDD : System Design Document

ODD : Object Design Document

1. **Class interface glossary**

.

1. **SERVIZI DEI SOTTOSISTEMI**