Immagine che contiene testo, Carattere, logo, design

Descrizione generata automaticamente

**Pagina riservata alla copertina del documento.**

**Partecipanti**

|  |  |
| --- | --- |
| **Nome** | **Matricola** |
| Kevin | Paladino |
| Andrea | Landolfi |
| Salvatore | Martire |

**Revision History**

|  |  |  |  |
| --- | --- | --- | --- |
| **Data** | **Versione** | **Descrizione** | **Autore** |
| 20/01/2024 | 1.0 | Prima Stesura Documento | Andrea Landolfi |
| 21/01/2024 | 1.1 | Stilati trade-off e documentation guidline | Andrea Landolfi |

**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**

Nel corso dell'analisi e della progettazione del sistema, sono stati identificati diversi compromessi che devono essere presi in considerazione durante lo sviluppo. In aggiunta, anche nella fase dell’Object Design si presentano vari trade-offs progettuali che sono esaminati in dettaglio in questa sezione del documento:Inizio modulo

Comprensibilità vs Tempo

Il codice deve essere scritto in modo da essere quanto più comprensibile e manutenibile per agevolare modifiche future. Ma visto che il tempo di rilascio del sistema è uno degli obiettivi prioritari che sono stati individuati, a tal proposito non verrà commentata ogni singola riga di codice con descrizioni dettagliate.

Sicurezza vs Efficienza

L'aspetto della sicurezza riveste un ruolo fondamentale nel sistema. Tuttavia, dati i tempi di sviluppo limitati, ci concentreremo sull'implementazione di funzioni di sicurezza basate su username e password.

Response Time vs Hardware

Il tempo di risposta costituisce un elemento cruciale nel sistema, soprattutto per specifiche funzionalità. Tuttavia, tale aspetto sarà influenzato anche dal tipo di hardware su cui il sistema verrà eseguito.

**1.2 Interface documentation guidelines**

Le linee guida includono una lista di regole che gli sviluppatori dovrebbero rispettare durante la progettazione delle interfacce:

- Classi Java e Servlet:

* I nomi dovranno seguire la convenzione UpperCamelCase*.*
* I nomi dovranno corrispondere alle informazioni e/o funzionalità che offre quella classe o Servlet

- Metodi:

* I nomi dei metodi dovranno seguire la convenzione lowerCamelCase.
* I nomi dovranno corrispondere alle informazioni e/o funzionalità che offre quel metodo. Si utilizzerà un verbo più eventualmente aggettivi.
* I nomi dei metodi per ottenere e settare attributi seguiranno la regola di nominazione ‘get[nomeattributo]’ e ‘set[nomeattributo]’.

- Variabili:

* I nomi delle variabili dovranno seguire la convenzione lowerCamelCase.
* Ogni dichiarazione di variabile locale può definire più di una variabile, mentre ogni dichiarazione di variabile di istanza deve definire solo una variabile. Le variabili d’istanza devono essere private.

- Convenzioni:

* Le condizioni d’errore lanciano delle eccezioni e non valori di ritorno.

- Pagine JSP:

Le pagine JSP devono essere conformi allo standard HTML 5. Inoltre, il codice deve utilizzare l'indentazione, per facilitare la lettura, secondo le seguenti regole:

* Un'indentazione consiste in una tabulazione;
* Ogni tag deve avere un'indentazione maggiore del tag che lo contiene;
* Ogni tag di chiusura deve avere lo stesso livello di indentazione del corrispondente tag di apertura;
* I tag di commento devono seguire le stesse regole che si applicano ai tag normali.

- Fogli di stile CSS:

I fogli di stile (CSS) devono seguire le seguenti convenzioni:

* Tutti gli stili devono essere non inline e devono essere collocati in fogli di stile separati.

**1.3 Definizioni, acronimi e abbreviazioni**

|  |  |  |
| --- | --- | --- |
| **Acronimo** | **Abbreviazione** | **Definizione** |
| HTML | Hyper Text Markup Language | Linguaggio di marcatura più usato per siti web. |
| CSS | Cascading Stylee Sheets | Linguaggio usato per definire la formattazione delle pagine web. |
| JAVA | JAVA | Linguaggio di programmazione orientato agli oggetti. |
| SERVLET | SERVLET | Classi java. |
| JSP | Java Server Pages | Una tecnologia Java utilizzata per creare pagine web dinamiche. |

**1.4 References**

RAD : Requirement Analysis Document

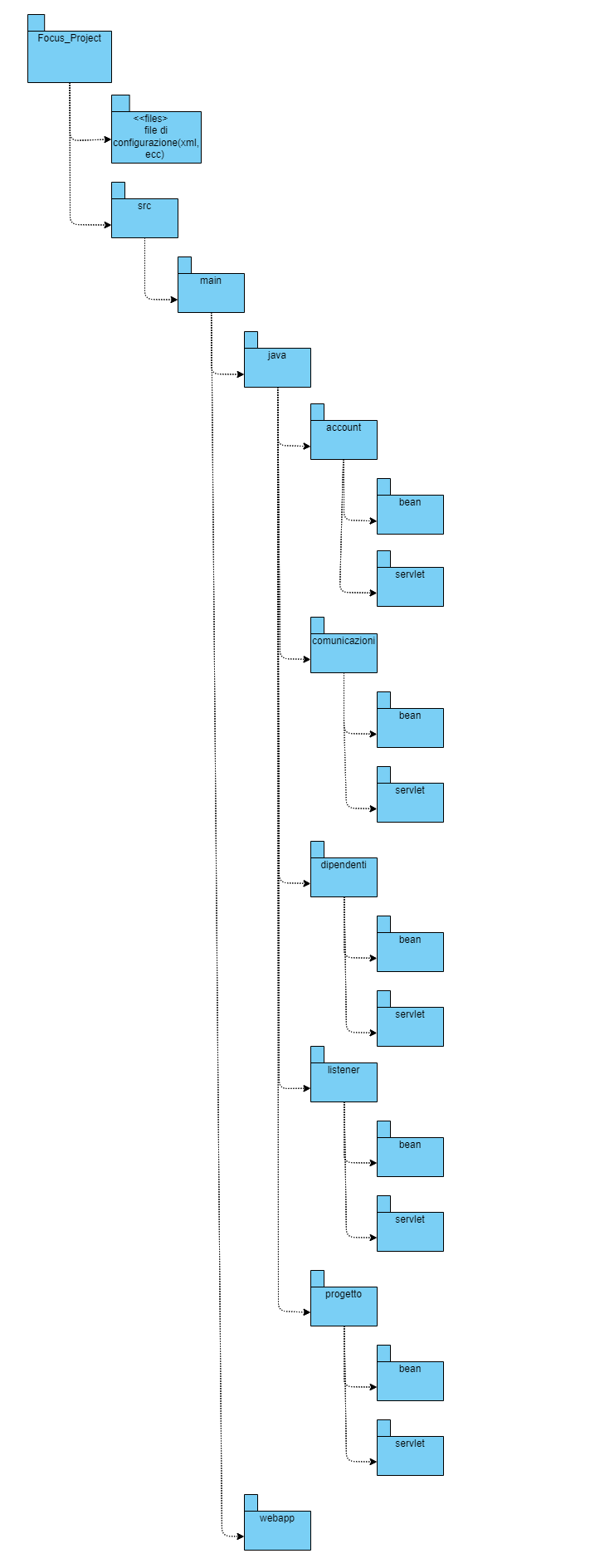
SDD : System Design Document

ODD : Object Design Document

1. **Class interface glossary**

Il sistema è diviso in package nel modo seguente:

**LATO-BACK-END**



.

1. **SERVIZI DEI SOTTOSISTEMI**