Università degli studi di Salerno

***Corso di Laurea in Informatica***

******

***INGEGNERIA DEL SOFTWARE***

***Object Design Document***

***“UNI-AirLines”***

# Studenti:

##### Nome Matricola

Santoro Mario 0512104850

Marino Raffaele 0512104508

Pastore Matteo 0512104724

Fortunato Angelo 0512104532

*Anno Accademico: 2018/19*

***SOMMARIO***

**1. Introduction…………………………………………………………**

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

1.2 Interface documentation guidelines………………………………………..

1.3 Definitions, acronyms, and abbreviations………………………………

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

**2. Packages……………………………………………………………..**

**3. Class interfaces ………………………………………**

**4. Glossary………………………………………**

**1. Introduction**

**1.1 Object design trade-offs**

Dopo la realizzazione del documento RAD (Requirement Analysis Document) e SDD (System Design Document), abbiamo descritto in linea di massima, quello che sarà il nostro sistema e quindi i nostri obiettivi, tralasciando gli aspetti implementativi.

* *Prestazioni VS Costi*. Considerando il sistema che stiamo realizzando, possiamo dire che il non eccessivo budget a nostra disposizione ci ha consentito di realizzare il prodotto utilizzando materiale open source partendo da zero minimizzando così i costi e rendendo il sistema più che soddisfacente.
* *Interfaccia vs. Tempo di risposta.* Il tempo di risposta tra server e interfaccia sono più che sufficienti (rapidi) a soddisfare le esigenze dei vari utenti collegati al Sistema. Ovviamente maggiore sarà la grandezza del database e maggiore sarà il tempo di risposta e ricerca nel database.
* *Interfaccia vs. Easy-use*. L’interfaccia, grazie all’utilizzo delle form e di una impostazione semplice e intuitiva, permette un uso facile (Easy-Use) della gestione del sistema di database prodotti anche considerevolmente grande così da rendere immediata l’attività anche ai meno esperti col computer.
* *Costi vs. Mantenimento*. Grazie a un uso di materiale open source e l’utilizzo del linguaggio javadoc il sistema può essere facilmente modificato, implementato con nuove funzioni o corretto in presenza di errori.

**1.2 Interface documentation guidelines**

Il sistema è multi-utente (può accedervi chiunque, visitatori, utenti e gestore).

* Al semplice visitatore, il sistema fornisce solamente la consultazione dei voli disponibili nelle tratte scelte oppure di consultare le offerte del sito.
* All’utente (visitatore loggato) il sistema fornisce la possibilità di completare l’acquisto dei biglietti scegliendo classe di viaggio, posto e bagaglio, poi di modificare i dati personali, visualizzare lo storico dei voli acquistati e effettuare il check-in online a partire da 3 giorni prima della partenza.
* Il gestore ha la possibilità di vedere tutti i voli in programma e ha la possibilità di inserire, modificare e/o cancellare i voli.

**1.3 Definitions, acronyms, and abbreviations**

* SDD: Software Design Document;
* ODD: Object Design Document
* RAD: Requirements analysis document.
* BROWSER: Explorer, Chrome, Mozilla.
* WebBrowser: Client (utente che accede al sistema)
* WebServer: Server su cui sono memorizzate le risorse.

**1.4. References**

* Problem Statement
* RAD
* SDD
* https://www.easyjet.com/it
* Object-Oriented Software Engineering Using UML, Patterns, and Java™ Third Edition Bernd Bruegge & Allen H. Dutoit
* http://java.sun.com

**2. Packages**

* Model:
* View :
* Controller: