

**Università degli Studi di Salerno**

**Anno Accademico 2016/2017**

**Corso di Ingegneria del Software**

**Object Design Document**

**V 0.1**

****

Introduzione 4

Scelte di design degli Oggetti 4

Interfaccia vs. Usabilità 4

Tempo di rilascio vs Tolleranza ai fault 4

Sicurezza vs. Efficienza 4

Comprensibilità vs. Tempo 4

Costi vs. Mantenimento 4

Interfaccia vs. Tempo di risposta 4

Package 5

Descrizione dei layer 5

Interface layer 5

Application logic layer 5

Storage layer 6

**Top Manager:**

*Prof. De Lucia Andrea*

**Team di sviluppo:**

|  |  |
| --- | --- |
| Nome e Cognome | Matricola |
| *Stefano Foresta* | 0512100504 |
| *Gennaro Franzese* | 0512100270 |

**Revision History:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Autore** | **Data** | **Descrizione** | **Versione** |
| *Stefano Foresta* | *10/12/16* | *Struttura documento* | *v 0.1* |
| *Gennaro Franzese* | *12/12/16* | *Stesura primo capitolo* | *v 0.2* |
| *Stefano Foresta* | *13/12/16* | *Stesura secondo capitolo* | *v 0.3* |

# Introduzione

Lo scopo dell’Object Design Document è quello di specificare i servizi che ogni sottosistema, che è stato presentato nel System Design Document, offre in termini di classi includendo operazioni, tipi, argomenti e signature in modo da avere una specifica completa. Questo documento serve come base per l’implementazione del progetto software.

## Scelte di design degli Oggetti

### Interfaccia vs. Usabilità

L’interfaccia, grazie all’utilizzo di un’impostazione semplice e intuitiva, permette un uso facile (*Easy-Use*) della gestione del sistema e del relativo database di prodotti, da parte di chi amministra il negozio, e della navigazione all’interno del negozio per la ricerca e il relativo acquisto di prodotti, da parte dei clienti.

### Tempo di rilascio vs Tolleranza ai fault

Una volta creata la struttura principale verrà dedicato tempo in più per la creazione di controllori in grado di gestire ogni eventuale errore della web application e verrà creato un modulo che gestirà i backup (giornalieri/settimanali/mensili) che permetteranno di avere un’istantanea del database che, in caso di guasti, sarà subito disponibile per la rimessa in funzione dell’intera applicazione.

### Sicurezza vs. Efficienza

Verrà implementato un modulo dedicato interamente all’autenticazione degli utenti che permetterà di mantenere privati gli account dei clienti/amministratori e che permetterà, solo dopo effettuata l’autenticazione, l’acquisto di beni/servizi (da parte dei clienti) e la modifica del database (da parte degli amministratori).

### Comprensibilità vs. Tempo

La stesura del codice sarà suddivisa in parti e commentata per rendere il più leggibile possibile lo stesso e permettere a terzi eventuali modifiche strutturali.

### Costi vs. Mantenimento

Grazie a un uso di materiale open source e l’utilizzo di commenti, come descritto al punto precedente, il codice sarà facilmente modificabile (implementazione di nuove funzioni o correzioni di errori) con costi contenuti. Al contrario, l’utilizzo di un’architettura three-tier, avrà dei costi superiori alla media per via della presenza di più macchine su cui girerà la web application e il database utilizzato.

### Interfaccia vs. Tempo di risposta

Il tempo di risposta tra server e interfaccia sono più che sufficienti a soddisfare le

esigenze dei vari clienti che si collegheranno al Sistema per ricercare e/o acquistare beni/servizi. Ovviamente maggiore sarà la grandezza del database e maggiore sarà il tempo di risposta e ricerca nel database che ovviamente sarà indicizzato per abbassare ulteriormente questi tempi.

# Package

Il software ha un’architettura three-tier e i livelli che si distinguono sono i seguenti:

1. **Interface Layer:** oggetti con cui l’utente interagisce (form, button, textbox, ecc.).
2. **Application Logic Layer:** livello che si occupa di gestire moduli che girano su un application server che genera i moduli per i contenuti dinamici e che riceve le richieste e le elabora.
3. **Storage Layer:** questo livello gestisce l’archiviazione persistente dei dati.

I layer sono suddivisi in package ed ognuno di questi contiene degli oggetti che andranno ad implementare le varie funzionalità del sistema che andiamo a sviluppare. I package vengono utilizzati per riunire le classi, logicamente correlate, che forniscono dei servizi simili.

## Descrizione dei layer

### Interface layer

|  |  |
| --- | --- |
| **Nome** | *Interface layer* |
| **Descrizione** | Offre all’utente servizi di interazione uomo-macchina, ovvero click, submit etc. |
| **Moduli presenti** | |
| **Modulo** | **Descrizione** |
| *Interfaccia web* | Questo modulo descrive l’interfaccia grafica con cui l’utente interagisce e che invia al sistema le varie richieste. |

### Application logic layer

|  |  |
| --- | --- |
| **Nome** | *Application logic layer* |
| **Descrizione** | livello che si occupa di gestire moduli che girano su un application server che genera i moduli per i contenuti dinamici e che riceve le richieste e le elabora. |
| **Moduli presenti** | |
| **Modulo** | **Descrizione** |
| *Gestione utente* | Il modulo si occupa della gestione generale dell’utente. Creazione, modifica, gestione, cancellazione. |
| *Gestione magazzino* | Il modulo si occupa della gestione generale del magazzino. |
| *Gestione prodotti* | Il modulo gestisce i prodotti. |
| *Gestione servizi* | Il modulo gestisce i servizi. |
| *Gestione vendite* | Il modulo gestisce le vendite. |
| *Gestione interfaccia* | Il modulo si occupa di ricevere e interpretare gli input. |
| *Gestione database* | Il modulo si occupa della comunicazione con il database. |

### Storage layer

|  |  |
| --- | --- |
| **Nome** | *Storage layer* |
| **Descrizione** | Questo livello gestisce l’archiviazione persistente dei dati. |
| **Moduli presenti** | |
| **Modulo** | **Descrizione** |
| *Database* | Gestisce le richieste di dati in entrata e in uscita. Si occupa del database e della sua gestione. |

## Comunicazione tra package