



**Università degli Studi di Salerno**  
**Anno Accademico 2016/2017**

**Corso di Ingegneria del Software**

**Object Design Document**  
V 1.5



<b>Introduzione.....</b>	<b>4</b>
<i>Scelte di design degli Oggetti.....</i>	<i>4</i>
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
<b>Package .....</b>	<b>5</b>
<i>Descrizione dei layer .....</i>	<i>5</i>
Interface Layer .....	5
Application Logic Layer .....	5
Storage Layer .....	6
<i>Comunicazione tra package.....</i>	<i>6</i>
<b>Class interfaces .....</b>	<b>7</b>
<i>Gestione registrazione .....</i>	<i>7</i>
Creazione account .....	7
<i>Gestione autenticazione .....</i>	<i>7</i>
Login .....	7
Logout.....	7
<i>Gestione utente.....</i>	<i>7</i>
Modifica account .....	7
<i>Gestione prodotti .....</i>	<i>7</i>
Carico prodotto .....	8
Modifica quantità .....	8
Modifica prezzo .....	8
<i>Gestione riparazioni.....</i>	<i>8</i>
Creazione richiesta riparazione .....	8
Accettazione/rifiuto riparazione.....	8
Elaborazione riparazione .....	8
<i>Gestione ordine.....</i>	<i>8</i>
Carrello .....	8
Storico vendite.....	8
<i>Gestione interfaccia .....</i>	<i>9</i>
<b>Class diagram .....</b>	<b>10</b>
<b>Glossario .....</b>	<b>11</b>

## Top Manager:

*Prof. De Lucia Andrea*

## Team di sviluppo:

Nome e Cognome	Matricola
<i>Stefano Foresta</i>	0512100504
<i>Gennaro Franzese</i>	0512100270

## Revision History:

Autore	Data	Descrizione	Versione
<i>Stefano Foresta</i>	<i>10/12/16</i>	<i>Struttura documento</i>	<i>v 0.1</i>
<i>Gennaro Franzese</i>	<i>12/12/16</i>	<i>Stesura primo capitolo</i>	<i>v 0.2</i>
<i>Stefano Foresta</i>	<i>13/12/16</i>	<i>Stesura secondo capitolo</i>	<i>v 0.3</i>
<i>Stefano Foresta</i>	<i>13/12/16</i>	<i>Stesura terzo capitolo, correzioni capitoli precedenti</i>	<i>v 0.6</i>
<i>Gennaro Franzese</i>	<i>13/12/16</i>	<i>Stesura quarto capitolo, revisione documento</i>	<i>v 1.0</i>
<i>Stefano Foresta</i>	<i>13/12/16</i>	<i>Stesura glossario</i>	<i>v 1.0.1</i>
<i>Gennaro Franzese</i>	<i>14/12/16</i>	<i>Inserimento Class Diagram</i>	<i>v 1.2</i>
<i>Stefano Foresta e Gennaro Franzese</i>	<i>19/02/17</i>	<i>Revisione totale documento</i>	<i>v 1.5</i>

## 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 assicurato un minimo di 4 backup giornalieri 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. L'architettura three-tier richiederà tuttavia dei costi di gestione un po' più alti.

#### 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:** livello che gestisce la parte grafica del sistema software
- 2) **Application Logic Layer:** gestisce la parte logica e le relative query che vengono sottomesse allo storage layer.
- 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

Modulo	Descrizione
<i>Interfaccia web</i>	Questo modulo descrive l'interfaccia grafica con cui l'utente interagisce e che interpreta i form sottomessi e quindi di ricevere e interpretare gli input.

#### Application Logic Layer

Modulo	Descrizione
<i>Gestione registrazione</i>	Il modulo consente la registrazione di un cliente sul database quindi la verifica dei dati, la presenza di un cliente con dati uguali e la chiamata al modulo di gestione dati per la memorizzazione nel database.
<i>Gestione autenticazione</i>	Il modulo consente il login e il logout dal sistema e le relative funzioni di esecuzione e controllo.
<i>Gestione utente</i>	Il modulo si occupa della gestione generale dell'utente ossia variabili, dati e chiamate alle query relative ad esso.
<i>Gestione prodotti</i>	Il modulo gestisce i prodotti rappresenta un'entità di magazzino e racchiude nello specifico le funzionalità ad esso relative.
<i>Gestione riparazioni</i>	Il modulo gestisce le riparazioni e quindi i relativi cambi di stato che comportano modifiche anche a livello di interfaccia grafica.
<i>Gestione ordine</i>	Il modulo gestisce gli ordini e coopera con i prodotti e si interfaccia direttamente con l'utente.
<i>Gestione dati</i>	Il modulo si occupa della comunicazione con il database quindi racchiude tutte le query che ogni modulo utilizza.

## Storage Layer

Modulo	Descrizione
<i>Database</i>	Gestisce le richieste di dati in entrata e in uscita. Si occupa del database e della sua gestione.

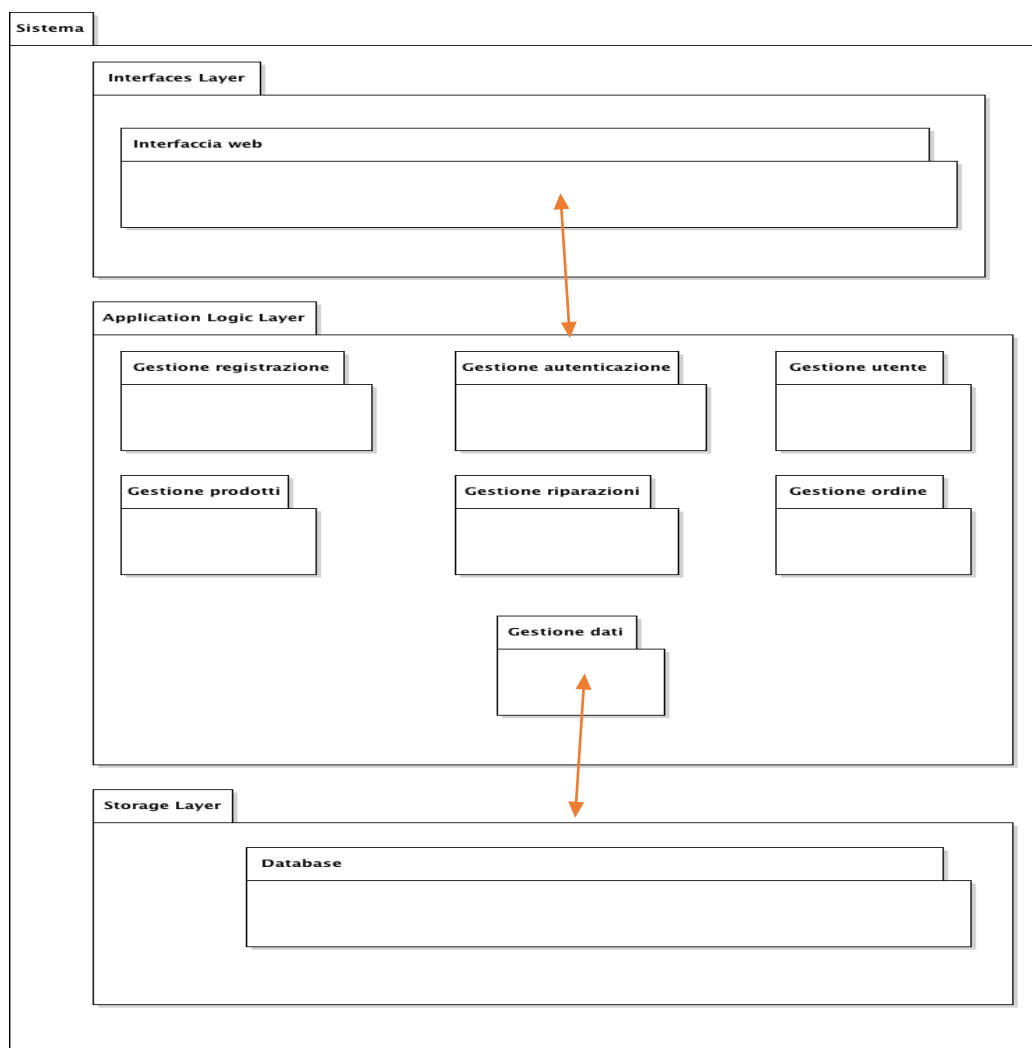
## Comunicazione tra package

Il seguente diagramma mostra la comunicazione tra i vari package che compongono i layer del sistema che sono stati appena descritti. La scelta progettuale è quella di far passare ogni richiesta fatta dalle classi di interfaccia attraverso la parte di Logic Application allo storage.

All'interno dell'Interface Layer è presente il pacchetto per la gestione della web GUI.

Il Logic Application Layer contiene i pacchetti che si occupano della logica applicativa del sistema e della sua gestione.

Infine, attraverso l'interfaccia Storage, la parte relativa al Logic Application accederà allo Storage Layer, per recuperare, aggiornare o eliminare dati persistenti dalla base dati.



## Class interfaces

Ad ogni classe viene assegnato un nome, significativo per il suo scopo e diverso da tutte le altre classi.

### Gestione registrazione

Questo modulo si occupa di gestire la creazione di un account.

#### Creazione account

<b>Dipendenze</b>	Nessuna
<b>Eccezioni</b>	<ul style="list-style-type: none"><li>• Utente già presente nel database;</li><li>• Dati inseriti non validi.</li></ul>

### Gestione autenticazione

Questo modulo si occupa di gestire il login e il logout di ogni utente.

#### Login

<b>Dipendenze</b>	Utenti
<b>Eccezioni</b>	<ul style="list-style-type: none"><li>• Dati non presenti;</li><li>• Dati inseriti non validi.</li></ul>

#### Logout

<b>Dipendenze</b>	Login
<b>Eccezioni</b>	Nessuna

### Gestione utente

Questo modulo si occupa della gestione generale dell'utente dal momento della sua creazione passando per le relative modifiche che possono essere apportate e finendo ad un'eventuale eliminazione dell'account.

#### Modifica account

<b>Dipendenze</b>	Utente
<b>Eccezioni</b>	<ul style="list-style-type: none"><li>• Dati già presenti;</li><li>• Dati inseriti non validi.</li></ul>

### Gestione prodotti

Il modulo gestisce i singoli prodotti che poi sono organizzati nel catalogo

#### Carico prodotto

<b>Dipendenze</b>	Catalogo, Storage
<b>Eccezioni</b>	Nessuna.

#### Modifica quantità

<b>Dipendenze</b>	Prodotti
<b>Eccezioni</b>	Dati inseriti non validi.

#### Modifica prezzo

<b>Dipendenze</b>	Prodotti
<b>Eccezioni</b>	Dati inseriti non validi.

### Gestione riparazioni

Il modulo gestisce le riparazioni e i cambi di stato relativi all'andamento della stessa.

#### Creazione richiesta riparazione

<b>Dipendenze</b>	Utente
<b>Eccezioni</b>	Nessuna

#### Accettazione/rifiuto riparazione

<b>Dipendenze</b>	Utente, riparazione
<b>Eccezioni</b>	Nessuna

#### Elaborazione riparazione

<b>Dipendenze</b>	Utente, riparazione
<b>Eccezioni</b>	Nessuna

### Gestione ordine

Il modulo gestisce le vendite di beni/servizi e si interfaccia direttamente con gli utenti.

#### Carrello

<b>Dipendenze</b>	Storage, categoria, prodotti, servizi.
<b>Eccezioni</b>	Nessuna.

#### Storico vendite

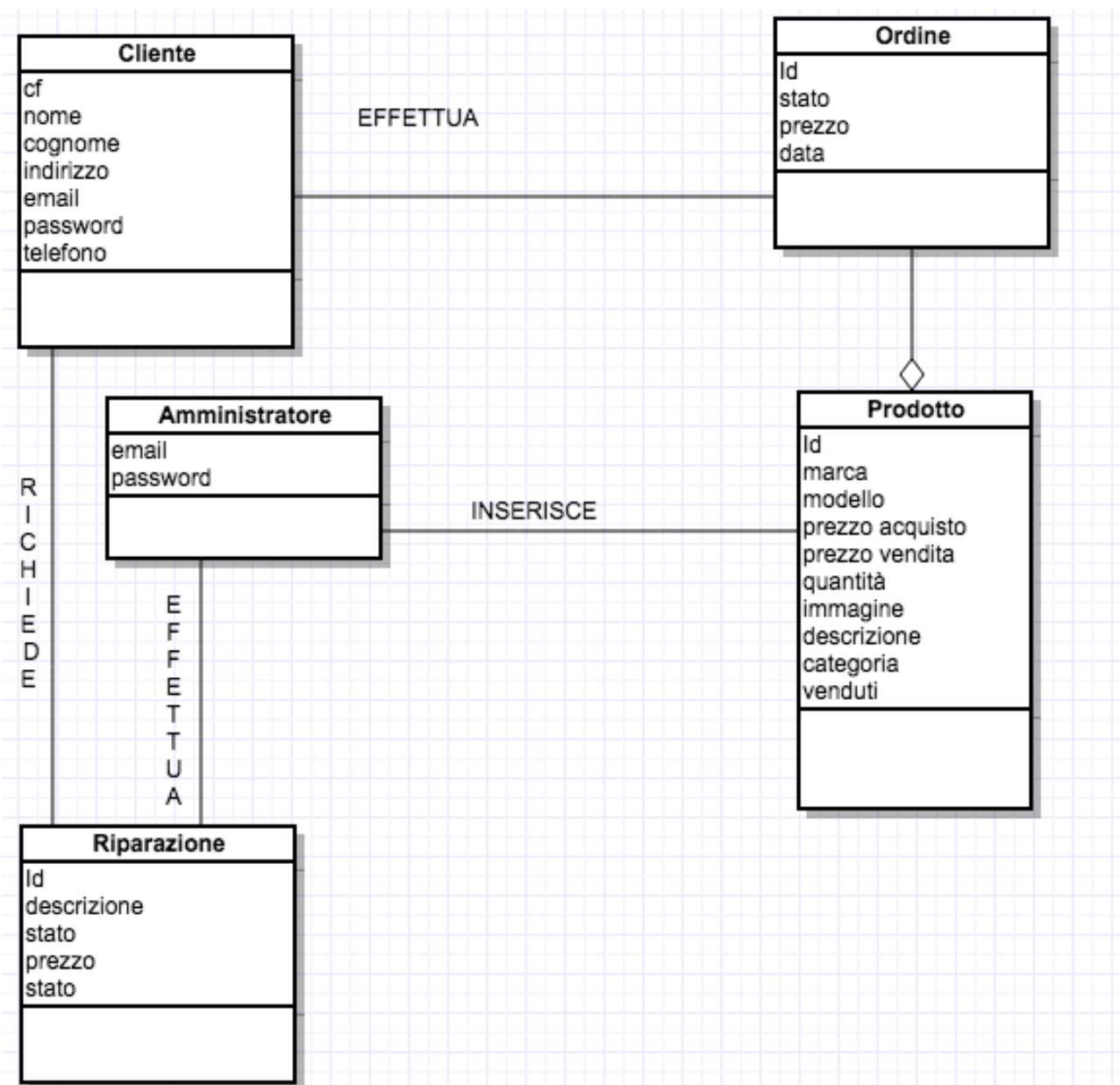
<b>Dipendenze</b>	Storage, clienti.
<b>Eccezioni</b>	Nessuna.



## Gestione interfaccia

Il modulo si occupa di ricevere e interpretare gli input che vengono dal layer interfaccia. Comprendono quindi tutte le varie componenti grafiche che servono per la gestione delle operazioni sulla web application.

# Class diagram



## Glossario

<b>Termine</b>	<b>Descrizione</b>
<i>Easy-Use</i>	Di facile utilizzo.
<i>Package</i>	Collezione di classi, interfacce.