



# Università degli Studi di Salerno

Corso di Ingegneria del Software



**Negozio-Online**  
**SDD – System Design Document**  
**Versione 1.8**

**Partecipanti:**

Nome	Matricola
Gaglione Rino	0512102476
Capobianco Marco	0512102462

**Revision History**

Data	Versione	Descrizione	Autore
16/11/2015	1.0	Introduzione	Gaglione Rino
17/11/2015	1.1	Component Diagram	Gaglione Rino
18/11/2015	1.2	Component Diagram	Gaglione Rino
19/11/2015	1.3	Deployment Diagram	Gaglione Rino
21/11/2015	1.4	Gestione Dati persistenti	Gaglione Rino
27/11/2015	1.5	Controllo degli accessi e sicurezza/Matrice di Accesso	Gaglione Rino
1/12/2015	1.6	Controllo software globale	Gaglione Rino
2/12/2015	1.7	Boundary condition	Gaglione Rino
3/12/2015	1.8	Sottoinsieme di Servizio	Gaglione Rino

# Indice

## **1.Introduzione**

- 1.1 Scopo del sistema
- 1.2 Obiettivi del design
- 1.3 Definizioni, acronimi e abbreviazioni
- 1.4 Riferimenti
- 1.5 Overview

## **2. Architettura software corrente**

## **3. Architettura software proposta**

- 3.1 Overview (Visione d'insieme)
- 3.2 Decomposizione in sottoinsiemi
- 3.3 Hardware/software mapping
- 3.4 Gestione dei dati persistenti
- 3.5 Controllo degli accessi e sicurezza
- 3.6 Controllo software globale
- 3.7 Boundary condition

## **4. Servizi dei sottoinsiemi**

# 1. Introduzione

## 1.1 Scopo del Sistema

Lo scopo del sistema "Negozio Online" è sviluppare un'applicazione web-based\* in grado di:

1. Offrire un supporto telematico per la gestione del magazzino.
2. Offrire un'interfaccia semplice, chiara ed intuitiva che stimoli l'utente ad usufruire del servizio.
3. Offrire un'area per l'assistenza immediata al cliente.
4. Offrire la possibilità di ordinare prodotti.

## 1.2 Obbiettivi del design

- Criteri di Performance:

1. Throughput: Il sistema sarà capace di servire contemporaneamente un numero diverso di utenti dipendente dalle capacità di elaborazione della macchina (server) sulla quale verrà installato. Ciò sarà favorito anche da una grande velocità di connessione.
2. Tempo di risposta: il sistema deve garantire tempi di risposta decisamente brevi o quantomeno notificare nel minor tempo possibile eventuali indisponibilità del sistema stesso. Una richiesta di un utente deve essere soddisfatta entro 10 secondi. Gli accessi in lettura e scrittura alla base di dati del sistema saranno comunque eseguiti tanto più rapidamente quando maggiore è la velocità di elaborazione della macchina (server) sulla quale il sistema verrà installato e la comunicazione client/server.
3. Memorizzazione: per quanto riguarda la memorizzazione dei dati non è possibile fare una stima assoluta.

- Criteri di Affidabilità:

4. Sicurezza: L'accesso al sistema sarà controllato da un apposito sistema di autenticazione, che permetterà ad ogni categoria di utenti di eseguire il proprio lavoro senza intaccare o modificare quello altrui.
5. Robustezza: Il software avvertirà l'utente qualora sbagli o ometta l'inserimento

di un campo, spiegando dettagliatamente cosa ha provocato un errore nel sistema.

- Criteri di Manutenzione:

1. Estendibilità: E' consentito, in quanto è possibile aggiungere in futuro, nuove funzionalità al sistema (ex:come estendere il campo d'azione al di fuori del territorio nazionale) oppure creare nuove classi, con l'estensione di quelle già esistenti. A tal proposito il codice dovrà essere di facile comprensione per non creare confusione durante l'aggiunta di nuove funzionalità.
2. Modificabilità: Deve essere possibile intervenire sul codice esistente per correggere eventuali bugs, o implementare nuove funzionalità aggiuntive. Per ottenere questo requisito, anche in questo caso bisogna garantire che il codice sia leggibile per facilitare le modifiche o l'aggiunta di nuove funzionalità.
3. Portabilità: Il sistema potrà essere utilizzato da un qualsiasi computer che abbia la possibilità eseguire correttamente un browser web.

- Criteri per L'utente finale:

4. Usabilità: Le interfacce mostrate all'utente saranno intuitive ed auto esplicative. Ogni pulsante avrà un'etichetta che ne specifica la funzione, le caselle di testo presenteranno del testo illustrativo per aiutare l'utente nell'immissione dei dati.

### 1.3 Definizioni, acronimi e abbreviazioni

#### **Acronimi**

*SDD:* System Design Document

*DB:* Database

#### **Definizioni**

*Web Browser:* Software che permette la navigazione sul web

*Web Server:* Server che permette al client di visualizzare le pagine web attraverso il web browser

## 1.4 Riferimenti

Bernd Bruegge & Allen H. Dutoit, Object-Oriented Software Engineering: Using UML, Patterns and Java, (3rd edition), Prentice- Hall, 2003.

## 1.5 Overview

Il sistema proposto aiuterà gli utenti dei vari settori nella gestione del negozio, quindi agevolerà le entrate e le uscite dal magazzino, proponendo un sistema facile e veloce. Il Cliente invece sarà agevolato da un'interfaccia semplice, chiara ed intuitiva. Al termine del progetto, il sistema vorrà migliorare la rapidità e l'efficienza con cui verranno eseguite le operazioni, supportando ogni membro nel modo più utile possibile.

## 2. Architettura software corrente

Il sistema da noi proposto è implementato dal nulla, in quanto nessun sistema esiste in precedenza.

## 3. Architettura software proposta

### 3.1 Overview

L'architettura del sistema è di tipo client/server. Il server riceve le richieste da parte dei client e risponde in tempo utile. Tale architettura favorisce l'affidabilità del sistema, in particolare la robustezza e la sicurezza, e inoltre facilita la progettazione e l'implementazione. Il software client in genere è di limitata complessità, limitandosi normalmente ad operare come interfaccia verso il server.

### 3.2 Decomposizione in sottoinsiemi

La decomposizione in sottosistemi permette di ridurre la complessità del dominio della soluzione.

Per realizzare il sistema è stato utilizzato lo stile architetturale “Three-tier” in versione Client/Server. Abbiamo fatto questa scelta in base alle tecnologie utilizzate: infatti, essendo il sistema web-based, tale stile architetturale risponde ampiamente a tutte le caratteristiche richieste e rispetta tutti i requisiti non funzionali. L’architettura a tre strati indica una particolare architettura software di tipo multi-tier per l’esecuzione di un ‘applicazione web-based che comporta la suddivisione del sistema, in tre diversi moduli o strati dedicati all’interfaccia utente, alla logica applicativa e alla gestione dei dati persistenti.

In particolare i tre strati si occupano di differenti funzionalità del sistema di seguito descritte:

- ***L'interface layer*** che rappresenta l'interfaccia grafica e in generale i boundary object come le form e i link che permettono agli utenti di interagire con il sistema.

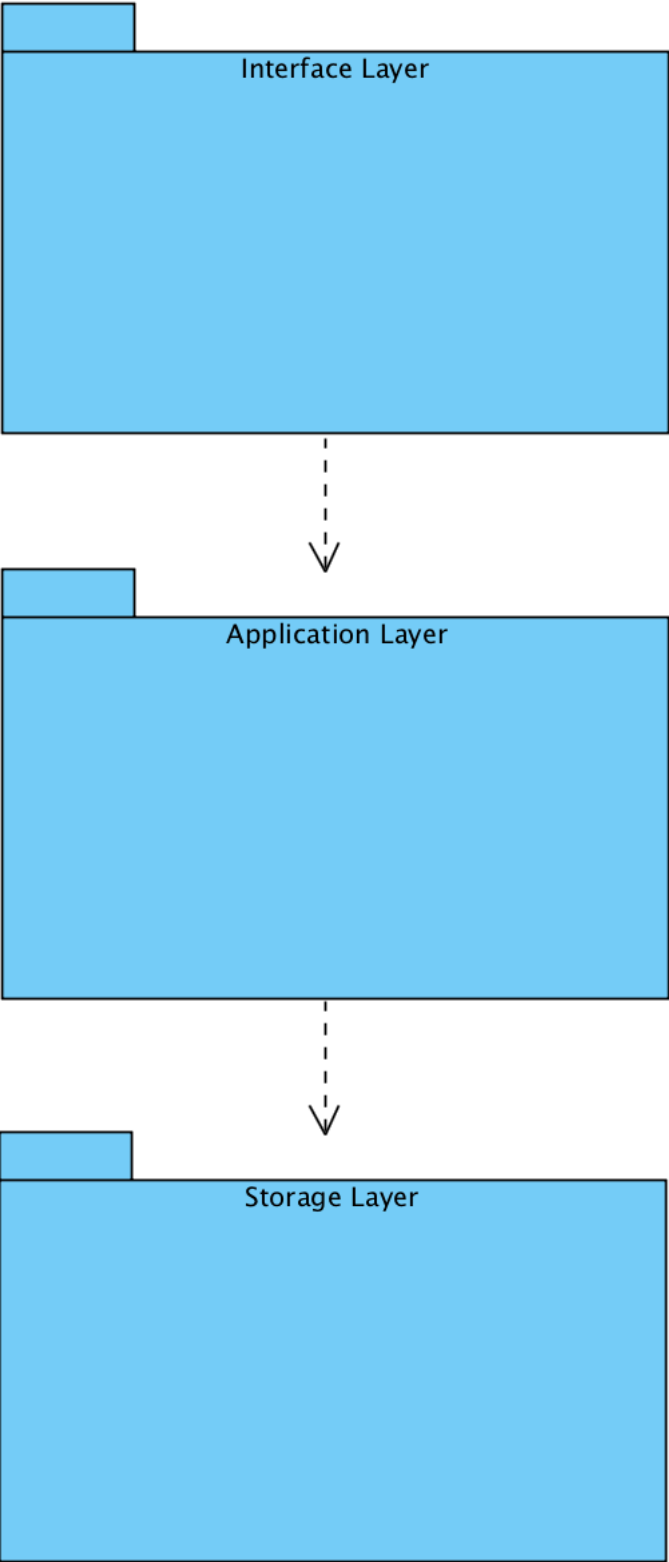
Nell'architettura client-server ricopre il ruolo di client, che richiede pagine al server.

Utilizza un browser che interpreta codice HTML e script.

- ***L'application logic layer*** che ha il computo di eseguire le elaborazioni, generando le risposte da inviare al client. In molti casi, ha bisogno di interrogare il database e, per farlo, si interfaccia con il storage layer.

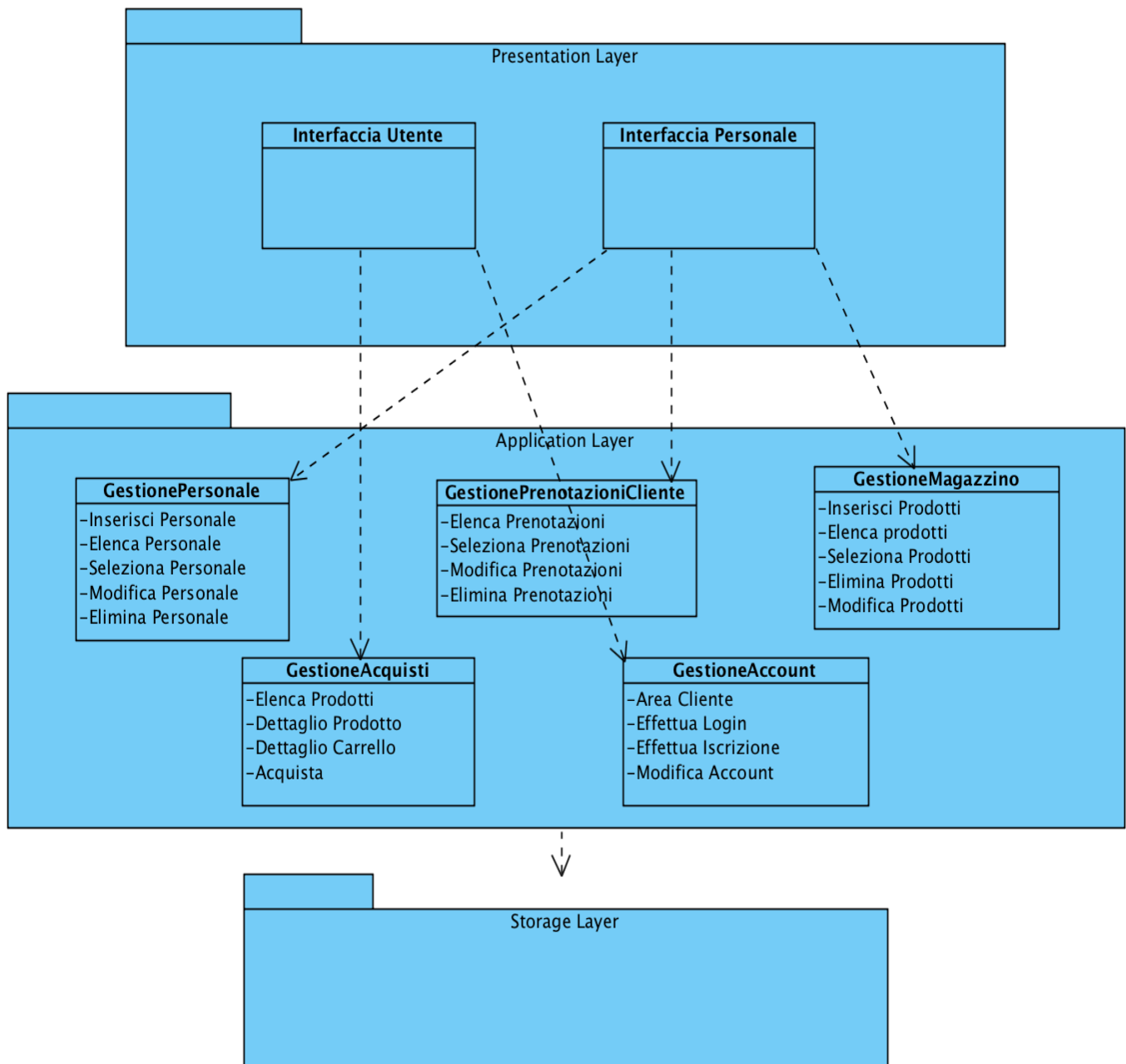
- ***Lo storage layer*** che si occupa della gestione dei dati. Riceve interrogazioni dell'application logic layer e risponde in tempo connettendosi al DBMS ed eseguendo le query.

Il client interagisce con interfaccia web che comunica con un web server a cui inoltra le richieste. L'interface layer comunica con l'application logic layer, il quale esegue le operazioni richieste e si avvale dell'ausilio dello storage layer. Quest'ultimo utilizza le componenti di MySQL per comunicare con il DBMS. Tutto ciò che avviene sempre considerando il fatto che si tratta di un'architettura di tipo client-server.





Ora vediamo nello specifico le funzionalità e gli obiettivi di ogni singolo strato implementato nel sistema.



Nello stile architetture del “Negozio-Online” il presentation Layer è caratterizzato dalle interfacce grafiche del cliente e del personale quali (Amministratore, Capo Magazziniere, Magazziniere ).

La **logica applicativa** contiene 5 sottoinsiemi ognuno di essi è legata all'interfaccia del sottosistema presente allo strato superiore in base alle funzionalità a cui egli stessi possono accedere.

**Lo Storage Layer** conterrà la base di dati dove verranno resi persistenti i dati, e comunicherà con l'application logic attraverso il framework JDBC.

### **Gestione Personale**

Modulo di gestione del personale permette all'amministratore di poter inserire , modificare ed eliminare il personale.

### **Gestione Prenotazioni Cliente**

Modulo di gestione delle prenotazioni dei clienti, permette di consultare le prenotazioni effettuate dai clienti, di apportare delle modifiche e di cancellarle.

### **Gestione Magazzino**

Modulo di gestione del Magazzino, permette l'inserimento dei prodotti acquistati dai fornitori, consultare le scorte disponibili, modificare e cancellare dati relativi ai prodotti.

### **Gestione Acquisto**

Modulo di gestione degli acquisti, riguarda la consultazione dei prodotti disponibili, l'inserimento dei prodotti al carrello ed eventuali modifiche e l'acquisto dei prodotti.

### **Gestione Account**

La gestione dell'Account, riguarda la parte del sistema che permette di visualizzare ed apportare modifiche al proprio account, di iscrizione ed accesso al sistema, e di chiedere eventualmente assistenza.

### 3.3 Mapping Hardware/Software

Il sistema che sarà realizzato si basa su un'architettura Web-based.

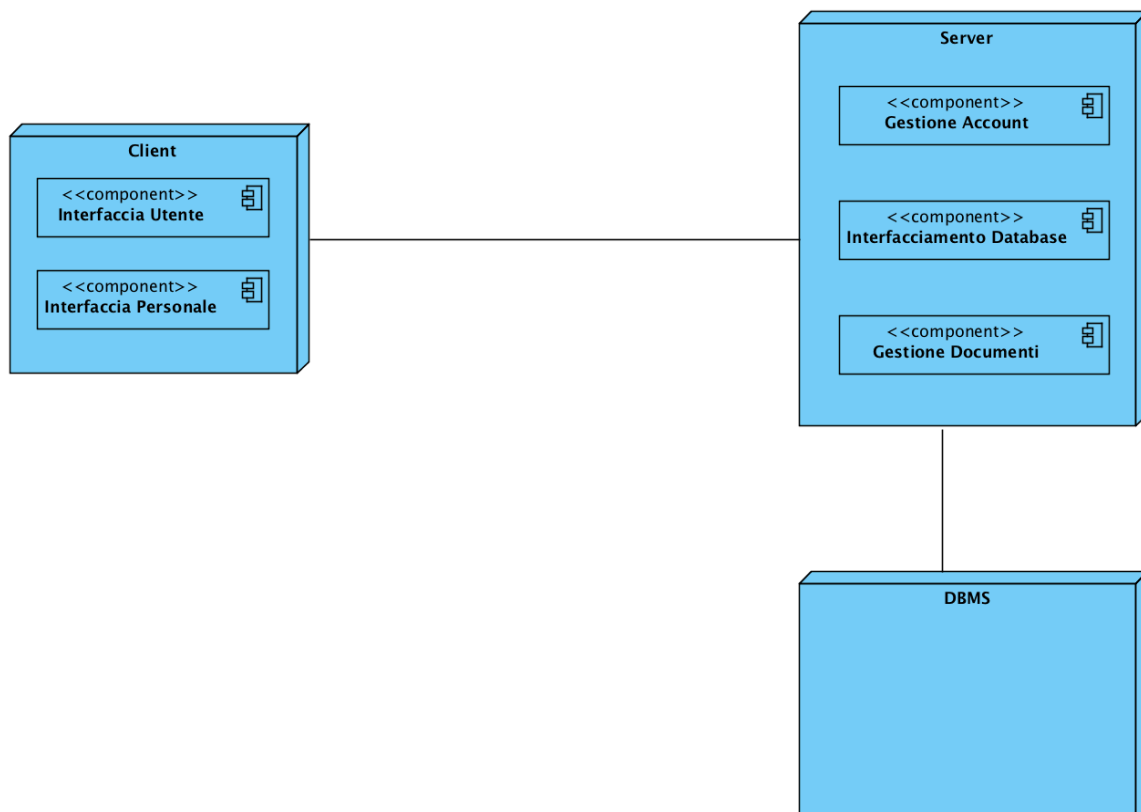
La struttura hardware proposta è costituita da un Server centrale e dalle postazioni utenti che possono essere costituite da computer portatili o fissi nei quali possono girare sistemi operativi diversi fra loro. Al server si collegano i client e il database. I client dovranno effettuare richieste al server per eseguire operazioni legate alle loro rispettive funzionalità. Il DMBS conterrà sia i dati relativi agli account, sia i dati relativi alle prenotazioni, prodotti, credenziali del personale. Il cliente ed il server saranno connesse tramite una rete che utilizzerà il protocollo TCP/IP.

**Protocollo Richiesto:** HTTP

**Memorizzazione Dati:** DMBS MySQL

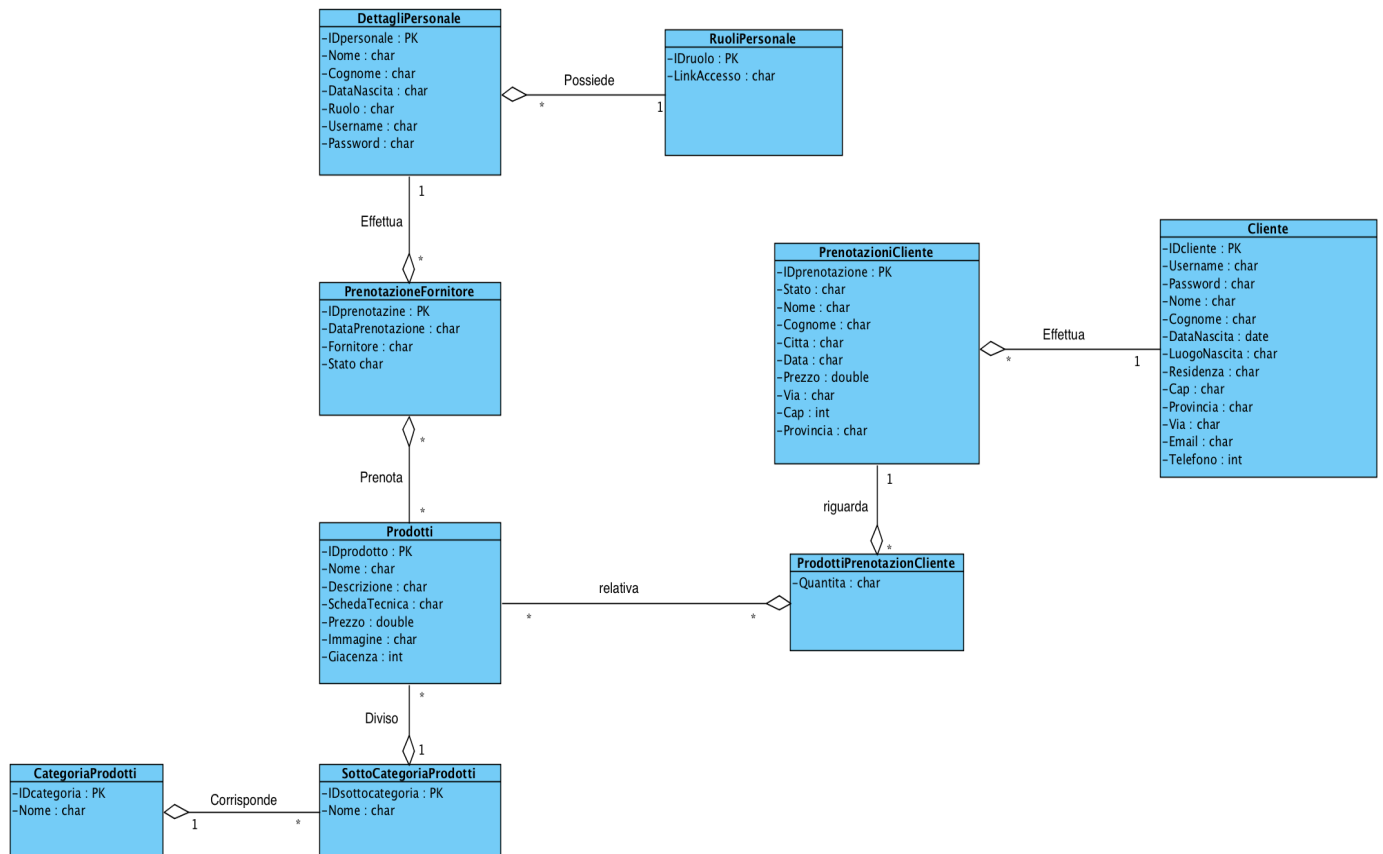
**WebServer:** Apache

**Linguaggi di programmazione utilizzati:** PHP, Html, CSS, Javascript, Ajax.



### 3.4 Gestione dei dati persistenti

#### Class Diagram



Memorizzazione dati: è stato preferito l'utilizzo di un DataBase relazionale per la memorizzazione dati, poiché serve un ampio spazio di memorizzazione (i dati sono molti). In più i DB consentono di ottenere un veloce tempo di risposta (query di ricerca), garantiscono una gestione concorrente (multiutente).

- Le credenziali per il personale saranno conservati in un database così da impedire l'accesso agli utenti non autorizzati.
- Le varie categorie di prodotti saranno memorizzati nel database.
- Scorte: in funzione di una buona gestione del magazzino è necessario tenere traccia dei consumi avvenuti durante gli acquisti effettuati dai clienti.
- I clienti che si registrano al sito sono memorizzati nel database, così da impedire l'accesso agli utenti non autorizzati.
- Prenotazione: è necessario avere un elenco delle prenotazioni ricevute in modo da gestire nel migliore dei modi l'ubicazione dei colli.

### 3.5 Controllo d'accesso

All'interno del sistema saranno controllati gli accessi degli utenti tramite un meccanismo di login. Ogni tipo di utente potrà accedere al sito....

- L'amministratore accede al sito tramite la sua pagina di login.
- Il cliente potrà accedere al sito per effettuare le prenotazioni, elenco dettagli, elenco carrello, dettagli ordini. Nel caso di modifica o cancellazione potrà accedere alla sua area utente.
- Il capo Magazziniere avrà diritto di visualizzare le schermate relative alla consultazione delle scorte.
- Il Magazziniere è il coordinatore delle prenotazioni e avrà la possibilità di accedere alle informazioni che riguardano le prenotazioni in modo tale da poter gestire al meglio le spedizioni da consegnare al corriere.

Il sistema prevede quattro figure di utenti: Clienti, Amministratore, Capo Magazziniere, Magazziniere. Ogni attore può accedere alle diverse funzionalità del sistema ma con diritti di accesso regolamentati sulla base delle differenti tipologie di utenza. Per documentare i diritti di accesso e per tenerne traccia all'interno del sistema usufruiamo di una tabella di controllo di accesso che descrive le varie operazioni permesse agli attori sui diversi oggetti. Le colonne della matrice rappresentano gli oggetti del sistema mentre le righe rappresentano gli attori su cui sono regolamentati gli accessi.

Oggetti Attori	Effettua Login	Elenca Prodotti	Controllo Prenotazioni	Inserisci Prodotti
Amministratore	X	X	X	
Cliente	X	X	X	
Capo Magazziniere	X	X		X
Magazziniere	X		X	

### 3.6 Flusso di controllo globale

Il controllo del flusso software è gestito da classi php che interagiscono con il client. Quando il client ha una richiesta, una classe preposta si prende in carico di controllare e inviarle alle classi apposite e svolgere l'operazione. Terminata l'operazione, la risposta verrà inoltrata al client, questo perché il software è gestito da un “event – driven control” che utilizza javascript. Il server attende le risposte di un client che si interfaccia tramite webBrowser, e una volta ricevuta una richiesta, la smista alla classe php preposta.

### 3.7 Condizione di limite(Boundary conditions)

Le Boundary conditions di installazione, start up e shut down saranno così definite:

Installazione: richiede lato server la presenza del DBMS e del Webserver con il quale si accederà a determinati file inseriti in un particolare percorso di file. Lato client occorre la connessione ad internet e un browser compatibile con PHP, Javascript, AJAX, Html e CSS.

Start-up: comporta l'avvio del DBMS e del Webserver.

Shut-down: viene interrotto il servizio fornito dal DBMS e dal Webserver.

#### Inizializzazione :

Il sistema lato server parte nel momento in cui è lanciato il web server dopo di che è sempre up in quanto sarà attivo 24h su 24 ed una volta attivato non è più stoppato. Il sistema lato client è inizializzato ogni volta che un utente o il personale accede al portale (start di sessione) inserendo l'URL del sito nel browser.

Allo start-up del cliente o del personale si presenta un interfaccia intuitiva di login in cui l'utente andrà ad inserire nome e password per accedere alle funzionalità del sistema. Nel momento in cui l'utente effettua il submit della form, i sottosistemi, in particolare quello di autenticazione, accederanno al database per trarre informazioni e verificare il login. Una volta che l'operazione ha avuto successo è possibile accedere a un numero limitato di operazioni dove ogni utente avrà pieno accesso alle funzionalità disponibili per il ruolo che ricopre.

#### Terminazione:

Il sistema lato server non può terminare ( a meno di guasti ). Il sistema lato client è terminato alla chiusura del browser ed in quell'istante vengono rilasciate le informazioni temporanee quali variabili di sessione.

### Fallimento :

Il sistema lato server può fallire solo a causa di condizioni eccezionali quali mancanza di elettricità o guasti all'hardware (hard disk danneggiato ecc., che danneggiano permanentemente il sistema) o in caso di crash di sistema (attacchi al server dall'esterno). Il sistema lato client può fallire a causa di guasti temporanei (hardware o software), o per la caduta della linea telefonica, ma ciò non influisce in alcun modo sul lato server (che resterà comunque stabile), neanche se ci fosse stata una connessione attiva al DB, visto che il DBMS gestisce le transazioni.

Per recuperare da un fallimento basta riaccendere il web server, ma ovviamente, nel frattempo, i client non possono usufruire nel sistema visto che il server è down.

Essendo distribuito, potrebbe verificarsi un congestionamento della rete che provocherebbe un notevole rallentamento delle interazioni.

### Eccezioni :

Il web server può cadere e non essere agibile per un po' di tempo. Al sistema lato client apparirà il messaggio "Http 404", ossia che la pagina non è disponibile.

## **4    Sottosistema di servizio**

### **GestionePersonale**

Application Layer	Presentation Layer	Storage Layer
InserisciPersonale	GUIInserisciPersonale	//
ElencaPersonale	GUIElenca Personale	//
SelezionaPersonale	GUISelezionaPersonale	//
EliminaPersonale	GUIEliminaPersonale	//
ModificaPersonale	GUIModificaPersonale	//



### GestionePrenotazioniCliente

Application Layer	Presentation Layer	Storage Layer
ElencaPrenotazioni	GUIElencaPrenotazioni	//
SelezionaPrenotazioni	GUISelezionaPrenotazioni	//
EliminaPrenotazioni	GUIEliminaPrenotazioni	//
ModificaPrenotazioni	GUIModificaPrenotazioni	//

### GestioneMagazzino

Application Layer	Presentation Layer	Storage Layer
ElencaProdotti	GUIElencaProdotti	//
SelezionaProdotti	GUISelezionaProdotti	//
EliminaProdotti	GUIEliminaProdotti	//
ModificaProdotti	GUIModificaProdotti	//
InserisciProdotti	GUIInserisciProdotti	//

### GestioneAccount

Application Layer	Presentation Layer	Storage Layer
EffettuaLogin	GUIEffettuaLogin	//
EffettuaIscrizione	GUIEffettuaIscrizione	//
ModificaAccount	GUIModificaAccount	//
EliminaAccount	GUIEliminaAccount	//
AreaCliente	GUIAreaCliente	//
RecuperaPassword	GUIRecuperaPassword	//
ChiediAssistenza	GUICHiediAssistenza	//
dettagliAccount	GUIDettagliAccount	//

**GestioneAcquisto**

Application Layer	Presentation Layer	Storage Layer
ElencaProdotti	GUIElencaProdotti	//
DettagliProdotto	GUIDettagliProdotto	//
DettagliCarrello	GUIDettagliCarrello	//
RicercaProdotti	GUIRicercaProdotti	//
EliminaProdotto	GUIEliminaProdotto	//
Acquista	GUIAcquista	//
ConfermaAcquisto	GUIConfermaAcquisto	//
AggiungiAlCarrello	GUIAggiungiAlCarrello	//
ModificaQuantità	GUIModificaQuantità	//