



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

**Corso di Ingegneria del Software**

**System Design Document**  
V 1.4

**ITECH**

## 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>Gennaro Franzese</i>	<i>21/11/2016</i>	<i>Struttura documento</i>	<i>v 0.4</i>
<i>Stefano Foresta</i>	<i>21/11/2016</i>	<i>Stesura generale del documento</i>	<i>v 1.0</i>
<i>Stefano Foresta</i>	<i>22/11/2016</i>	<i>Stesura capitolo "Architettura del sistema proposto"</i>	<i>v 1.1</i>
<i>Gennaro Franzese</i>	<i>23/11/2016</i>	<i>Stesura capitolo "Gestione del controllo globale"</i>	<i>v 1.2</i>
<i>Gennaro Franzese</i>	<i>24/11/2016</i>	<i>View sottosistemi</i>	<i>v 1.2.1</i>
<i>Stefano Foresta</i>	<i>25/11/2016</i>	<i>Stesura capitolo "Gestione dei dati persistenti"</i>	<i>v 1.4</i>

## Sommario

<b>Introduzione.....</b>	<b>4</b>
<b>Scopo del sistema.....</b>	<b>4</b>
<b>Design goals .....</b>	<b>4</b>
Criteri di prestazione .....	4
Criteri di affidabilità.....	4
Criteri di manutenzione .....	5
<b>Overview.....</b>	<b>6</b>
<b>Architettura del sistema .....</b>	<b>6</b>
<b>Architettura del sistema proposto .....</b>	<b>6</b>
<b>Overview.....</b>	<b>6</b>
<b>Decomposizione in sottosistemi .....</b>	<b>7</b>
Interface Layer.....	7
Application Logic Layer .....	7
Storage Layer .....	7
View sottosistemi .....	8
<b>Hardware/Software mapping.....</b>	<b>9</b>
<b>Gestione dei dati persistenti.....</b>	<b>9</b>
Schema EER .....	9
Dizionario dei dati.....	9
Relazioni.....	9
Effettua: .....	9
Compie: .....	10
Diretto a: .....	10
Contiene: .....	10
Fornito da: .....	10
Effettua: .....	10
Entità.....	10
Dipendente:.....	10
Spedizione: .....	11
Fornitore: .....	11
Ordine: .....	11
Cliente: .....	11
Riparazione: .....	12
Prodotto: .....	12
Vendita: .....	12
<b>Controllo degli accessi e sicurezza .....</b>	<b>12</b>
Controllo degli accessi .....	12
Sicurezza .....	14
<b>Gestione del controllo globale.....</b>	<b>14</b>
<b>Boundary conditions .....</b>	<b>14</b>
Configuration .....	14
Exception handling .....	15
<b>Servizi sottosistemi .....</b>	<b>16</b>
Interface layer.....	16
Application logic layer .....	16
<b>Glossario .....</b>	<b>17</b>

# Introduzione

## Scopo del sistema

Il sistema, automatizzando la maggior parte delle operazioni che si svolgono in un normale negozio di informatica, renderà più fluido e leggero il carico di lavoro di chi è addetto all'amministrazione del suddetto negozio. Le operazioni di carico/scarico merci, di acquisto/vendita prodotti e di gestione documentazione (fatture in entrata/uscita) saranno gestite dal sistema che si verrà a creare e quindi i dati verranno conservati in un database relazionale creato ad-hoc e la gestione sarà affidata ad una web application disegnata su misura per il cliente.

## Design goals

Il sistema sarà progettato con un'interfaccia user-friendly in modo da agevolare l'utilizzo a persone che non hanno molta dimestichezza con le web-application di questo genere. Il software sarà ideato in modo da gestire al meglio le risorse del sistema evitando fastidiosi "crash". Sarà un prodotto utile e semplice nell'interazione che permetterà ai propri utenti di trovarsi davanti un software facile da utilizzare. La gestione del database e le relative operazioni saranno completamente gestibili dall'interfaccia di gestione web che sarà creata ad-hoc secondo le richieste del cliente e attraverso l'invio di numerosi feedback per ogni operazione il proprietario sarà agevolato nel cercare di commettere meno errori possibili. Dal lato client l'interfaccia sarà il più semplice possibile per permettere una ricerca veloce ed intuitiva dei prodotti e per agevolare l'acquisto di un bene/servizio in maniera semplice e con una serie finita di passi.

## Criteri di prestazione

Criteri di design	Definizione
<i>Tempo di risposta</i>	Il sistema deve essere efficiente, deve garantire una risposta in breve tempo, massimo 15 secondi, senza che si verifichino errori.
<i>Throughput</i>	Il sistema riesce a soddisfare massimo 50 task contemporaneamente in un tempo prefissato di un minuto.
<i>Memoria</i>	La memoria verrà gestita dal sistema il quale oltre a gestirne la memoria fisica (RAM), gestisce anche la memoria secondaria (Hard Disk).

## Criteri di affidabilità

Criteri di design	Definizione
<i>Robustezza</i>	Il sistema deve prevedere e gestire eventuali input errati senza bloccare il funzionamento del sistema quindi verrà fatta molta attenzione durante la fase di sviluppo per lo sviluppo della gestione degli errori. Il sistema ha poi un layer-application che serve ad elaborare i dati inviati dalle varie interfacce.

<b><i>Sicurezza</i></b>	Per sicurezza, oltre a intendere la privacy del cliente, si intende l'inalterabilità dei dati dovuta da modifiche sbagliate, quindi in caso di modifiche, il sistema tramite interfaccia GUI, manderà un messaggio di conferma a colui che utilizza il software.
<b><i>Tolleranza crash</i></b>	Questo è un aspetto fondamentale, il sistema non può operare in situazioni di errore, pertanto devono essere previste queste situazioni. Al verificarsi di un crash verrà salvato il lavoro in corso in file temporanei ed il sistema software terminerà la sua esecuzione riavviandosi e cercando di mantenere inalterate le impostazioni. Nel caso in cui ci siano errori che non permettono il riavvio del sistema deve essere prevista una macchina di supporto che rimanga attiva per tutto il tempo del ripristino mantenendo attive le funzionalità di base.
<b><i>Affidabilità</i></b>	Il sistema oltre alla gestione degli errori deve garantire il funzionamento 24h su 24h tranne nei tempi relativi alla manutenzione ordinaria che da contratto prevede un tempo massimo di stop di 12h.

#### Criteri di manutenzione

<b>Criteri di design</b>	<b>Definizione</b>
<b><i>Estendibilità</i></b>	Ai fini di uno sviluppo futuro deve essere possibile aggiungere, in un secondo momento, altre funzionalità al sistema. Per rendere possibile questo, la piattaforma verrà programmata nel modo più semplice possibile cercando di dividere al meglio le funzionalità in modo da poter, in futuro, apportare modifiche anche a singole parti e non per forza stravolgendo l'intero sistema.
<b><i>Leggibilità</i></b>	Il codice deve essere chiaro, quindi i nomi delle variabili e dei metodi che verranno utilizzati devono essere coerenti al contesto d'uso, dove è strutturato, per rendere più veloce una modifica o una estensione del sistema anche da parte di terzi che si avvicinano al progetto in un secondo momento.

## Overview

Il presente documento SDD (System Design Document) si occupa di:

- 1) Definire gli obiettivi di design del progetto (identificando quali caratteristiche di qualità dovrebbero essere ottimizzate);
- 2) Decomporre il sistema in vari sottosistemi più piccoli che sono di più facile realizzazione;
- 3) Selezionare le strategie per costruire il sistema:
  - a) Strategia Hardware/Software;
  - b) Strategie che si riferiscono alla gestione dei dati persistenti;
  - c) Il flusso di controllo globale;
  - d) Le politiche di controllo degli accessi;
  - e) La gestione delle condizioni limite;

L'output del system design è un modello del sistema che include quindi la decomposizione dello stesso in sottosistemi con la relativa descrizione per ognuna delle strategie utilizzate.

## Architettura del sistema

L'architettura attualmente proposta non va a sostituire nessuna architettura esistente, il sistema è alla sua prima versione.

## Architettura del sistema proposto

### Overview

Il software è composto seguendo lo schema architetturale client/server

- **Client:** il browser preferito del cliente.
- **Server:** le pagine PHP che implementano l'intero sistema.

Sono possibili tre configurazioni:

- **Single-Tier:** tutto il sistema è distribuito su una sola macchina che funzionerà da client e da server.
- **Two-Tier:** il sistema è distribuito ed eseguito su due macchine differenti; una macchina gestirà tutto il lato server e conterrà anche il database mentre l'altra macchina eseguirà il client.
- **Three-Tier:** a differenza della configurazione Two-Tier il database verrà gestito su una macchina separata.

La configurazione scelta è quella three-tier, così eventuali modifiche ad uno dei tre moduli non comporterà il blocco del sistema e la relativa modifica dei blocchi rimanenti rispetto a quello aggiornato. Tale configurazione è basata su tre layer:

- 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.

## Decomposizione in sottosistemi

I layer previsti sono: *Interface Layer*, *Application Logic Layer*, *Storage Layer*. Di seguito riportiamo, per ogni modulo, le componenti da cui è composto, che rappresentano nel dettaglio le operazioni di cui si occupa e i dati su cui operano.

### Interface Layer

Modulo	Descrizione
<i>Interfaccia web</i>	Questo modulo descrive l'interfaccia grafica con cui l'utente interagisce e che invia al sistema le varie richieste.

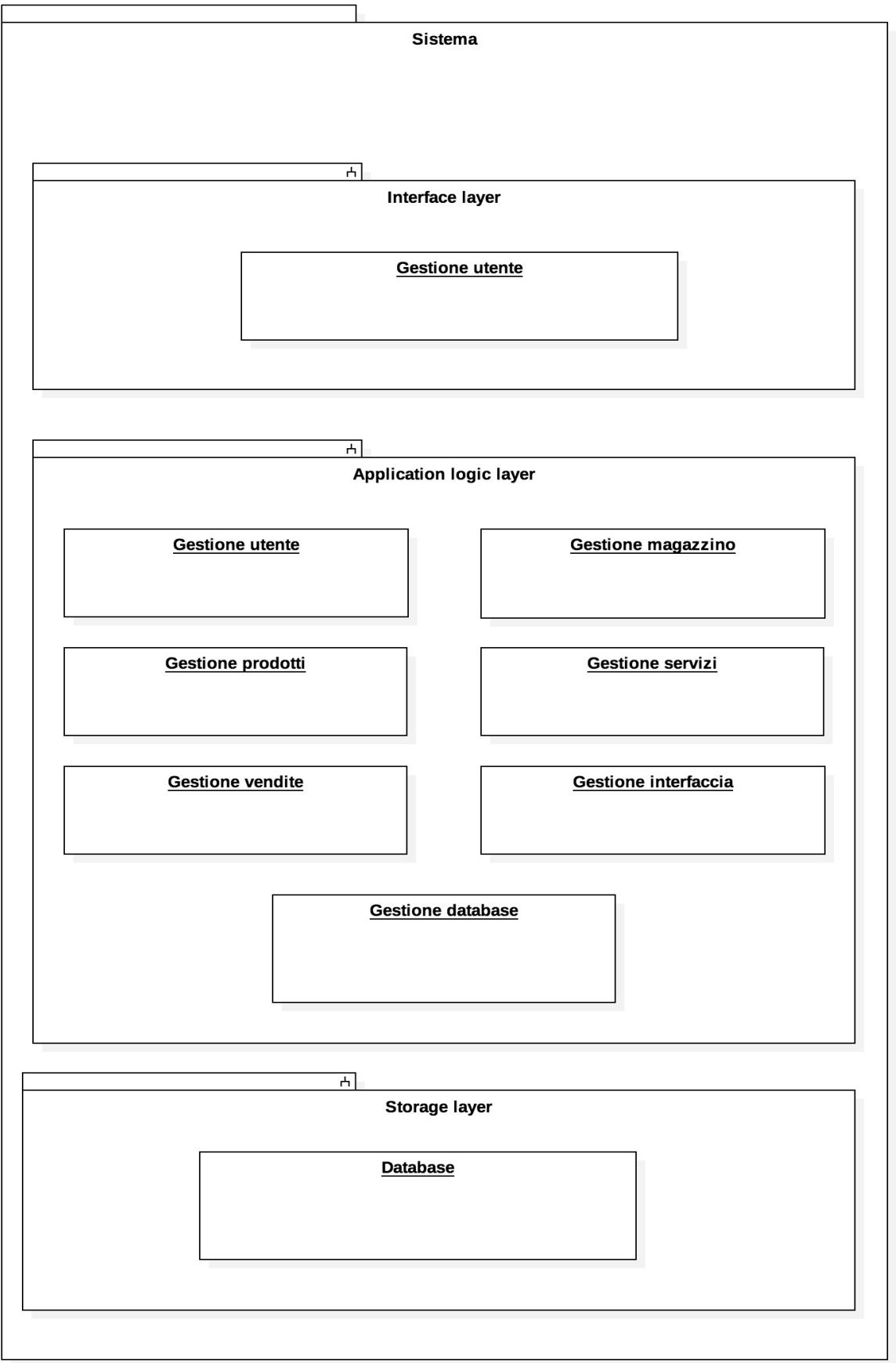
### Application Logic Layer

Modulo	Descrizione
<i>Gestione utente</i>	Il modulo si occupa della gestione generale dell'utente. Creazione, modifica, gestione, cancellazione.
<i>Gestione magazzino</i>	Il modulo si occupa della gestione generale del magazzino.
<i>Gestione prodotti</i>	Il modulo gestisce i prodotti.
<i>Gestione servizi</i>	Il modulo gestisce i servizi.
<i>Gestione vendite</i>	Il modulo gestisce le vendite.
<i>Gestione dati</i>	Il modulo si occupa della comunicazione con il database.
<i>Gestione interfaccia</i>	Il modulo si occupa di ricevere e interpretare gli input.

### 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.

View sottosistemi





## Hardware/Software mapping

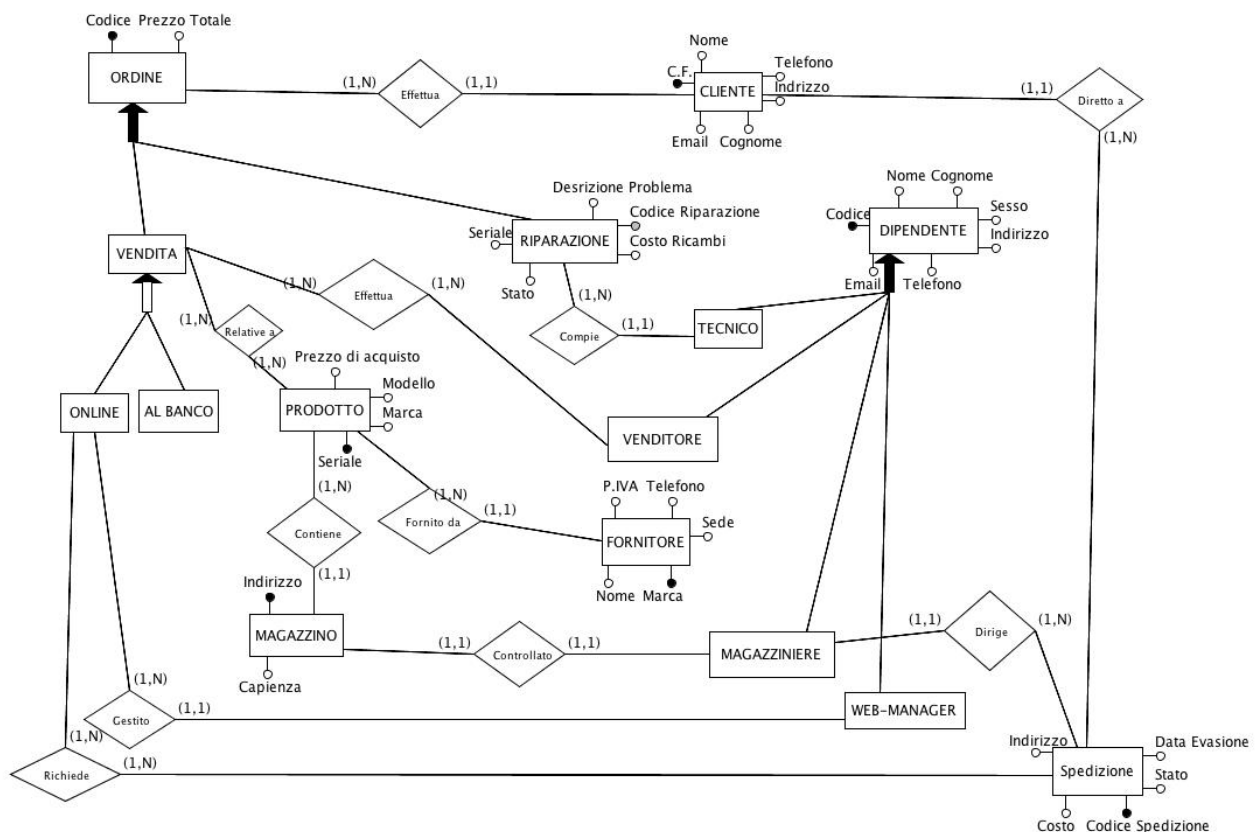
Il software utilizzerà un'architettura Three-Tier in configurazione Client-Server-Database:

- **Client:** si intende la macchina utilizzata dall'utente e dove gira il browser che quest'ultimo utilizza per navigare sul sito.
- **Server:** si intende la macchina su cui gira l'applicazione: genera i moduli per i contenuti dinamici e riceve ed elabora le richieste.
- **Database:** si intende la macchina su cui il database è installato e che gestisce quindi le richieste di dati sia in input che in output.

## Gestione dei dati persistenti

La scelta dei dati persistenti è stata fatta analizzando tutti gli oggetti rilevanti che in caso di chiusura controllata/incontrollata dovranno essere conservati.

## Schema EER



## Dizionario dei dati

### Relazioni

#### Effettua:

<b>Chiavi Esterne</b>	<i>Relazione tra venditore e ordine</i>
<u>Codice Univoco</u>	Chiave esterna codice del dipendente

<u>Codice</u>	Chiave esterna codice dell'ordine
---------------	-----------------------------------

Compie:

<b>Chiavi Esterne</b>	<i>Relazione tra tecnico e riparazione</i>
<u>Codice Univoco</u>	Chiave esterna codice del dipendente
<u>Codice Riparazione</u>	Chiave esterna codice della riparazione

Diretto a:

<b>Chiavi Esterne</b>	<i>Relazione tra spedizione e cliente</i>
<u>Codice Spedizione</u>	Chiave esterna codice di spedizione
<u>C.F.</u>	Chiave esterna del cliente

Contiene:

<b>Chiavi Esterne</b>	<i>Relazione tra prodotto e magazzino</i>
<u>Seriale</u>	Chiave esterna seriale prodotto
<u>Indirizzo</u>	Chiave esterna indirizzo del magazzino

Fornito da:

<b>Chiavi Esterne</b>	<i>Relazione tra prodotto e fornitore</i>
<u>Seriale</u>	Chiave esterna seriale prodotto
<u>Marca</u>	Chiave esterna marca del fornitore

Effettua:

<b>Chiavi esterne</b>	<i>Relazione tra il cliente e l'ordine</i>
<u>C.F.</u>	Chiave esterna C.F. del cliente
<u>Codice</u>	Chiave esterna codice di ordine

Entità

Dipendente:

Identifica un dipendente all'interno del database.

<b><i>Campo</i></b>	<b><i>Ruolo</i></b>
<u>Codice Univoco</u>	Contiene la matricola del dipendente
Nome	Contiene il nome del dipendente
Cognome	Contiene il cognome del dipendente
Sesso	Contiene il prezzo della spedizione
Indirizzo	Contiene l'indirizzo del dipendente
Telefono	Contiene il numero telefonico del dipendente
Email	Contiene l'indirizzo email del dipendente
Ruolo	Specializzazione del dipendente:

	“Tecnico” “Magazziniere” “WebManager” “Venditore”
--	--

#### Spedizione:

Contiene i dati relativi alla spedizione.

<b><i>Campo</i></b>	<b><i>Ruolo</i></b>
Codice Spedizione	Codice di tracciamento della spedizione e lo identifica
Indirizzo	Contiene l’indirizzo del cliente a cui va fatta la spedizione
Data Spedizione	Contiene la data in cui viene effettuata la spedizione
Costo	Contiene il prezzo della spedizione
Codice ordine	Contiene il codice dell’ordine per cui va fatta la spedizione
Stato	Contiene lo stato della spedizione: “Evaso” “Non evaso”

#### Fornitore:

Identifica un fornitore.

<b><i>Campo</i></b>	<b><i>Ruolo</i></b>
Marca	Codice la marca del prodotto venduto venduti dal fornitore e lo identifica.
Telefono	Contiene Numero telefonico del fornitore
Nome	Contiene Nome del fornitore
Sede	Contiene l’indirizzo del fornitore
Partita iva	Contiene il codice di partita iva

#### Ordine:

Identifica un ordine.

<b><i>Campo</i></b>	<b><i>Ruolo</i></b>
Codice	Contiene Codice alfanumerico che identifica l’ordine
Prezzo totale	Contiene Prezzo finale dell’ordine

#### Cliente:

Identifica un cliente all’interno del database.

<b><i>Campo</i></b>	<b><i>Ruolo</i></b>
C.F	Codice alfanumerico che identifica il cliente

Telefono	Contiene Numero telefonico del cliente
Nome	Contiene Nome del cliente
Cognome	Contiene Cognome del cliente

#### Riparazione:

Identifica una riparazione all'interno del database.

<i><b>Campo</b></i>	<i><b>Ruolo</b></i>
Codice_riparazione	Contiene Codice alfanumerico che identifica la riparazione
Prezzo_totale	Contiene il prezzo della spedizione
Descrizione_problema	Contiene la descrizione del problema da risolvere
Costo_ricambi	Contiene il costo dei pezzi di ricambi usati per la riparazione
Stato	Contiene lo stato della riparazione "In lavorazione" "Riparato" "Non riparabile"
Seriale	Contiene il seriale del prodotto da riparare

#### Prodotto:

Identifica un prodotto all'interno del database.

<i><b>Campo</b></i>	<i><b>Ruolo</b></i>
Seriale	Codice alfanumerico che identifica univocamente il prodotto
Marca	Contiene la marca del prodotto
Modello	Contiene il modello specifico del prodotto
Prezzo di acquisto	Contiene il prezzo di acquisto del prodotto

#### Vendita:

Identifica una vendita all'interno del database.

<i><b>Campo</b></i>	<i><b>Ruolo</b></i>
Codice	Codice univoco che identifica la vendita
Prezzo totale	Prezzo della vendita

### Controllo degli accessi e sicurezza

#### Controllo degli accessi

Il sistema prevede due tipi di attori:

- **Admin:** è l'amministratore del sistema ed ha accesso a tutte le componenti. È unico e può inserire nuovi utenti, assegnando ad ognuno di loro compiti specifici e l'accesso a uno o più parti del programma (es. se l'admin assume un dipendente che si occuperà solo del

magazzino, esso dovrà avere accesso alla sezione relativa al magazzino sia in termini di database e sia in termini di moduli).

- **Utente:** è colui che si registra sulla piattaforma e usufruisce dei servizi che essa propone. Può acquistare beni e/o servizi, comunicare con gli admin.

Gli oggetti che vengono gestiti direttamente dagli attori sono:

- *Ordine*
- *Dipendente*
- *Spedizione*
- *Fornitore*
- *Cliente*
- *Riparazione*
- *Prodotto*
- *Vendita*

<i><b>Attore/Oggetto</b></i>	<i><b>Admin</b></i>	<i><b>Utente</b></i>
<i>Ordine</i>	- <i>M</i> <i>V</i> <i>E</i>	<i>C</i> <i>M</i> <i>V</i> <i>E</i>
<i>Dipendente</i>	<i>C</i> <i>M</i> <i>V</i> <i>E</i>	- - <i>V</i> -
<i>Spedizione</i>	<i>C</i> <i>M</i> <i>V</i> <i>E</i>	- - <i>V</i> -
<i>Fornitore</i>	<i>C</i> <i>M</i> <i>V</i> <i>E</i>	- - <i>V</i> -
<i>Cliente</i>	<i>C</i> <i>M</i> <i>V</i> <i>E</i>	- - - -
<i>Riparazione</i>	- <i>M</i> <i>V</i> <i>E</i>	<i>C</i> <i>M</i> <i>V</i> <i>E</i>
<i>Prodotto</i>	<i>C</i> <i>M</i> <i>V</i> <i>E</i>	- - <i>V</i> -
<i>Vendita</i>	- <i>M</i> <i>V</i> <i>E</i>	<i>C</i> <i>M</i> <i>V</i> <i>E</i>

<b>C</b>	<i>Crea</i>
<b>M</b>	<i>Modifica</i>
<b>V</b>	<i>Visualizza</i>
<b>E</b>	<i>Elimina</i>

## Sicurezza

Il software assicura la protezione dei dati tramite un controllo degli accessi al sistema.

Ogni utente può accedervi attraverso l'username e la password che è stata scelta in fase di registrazione e in nessun altro modo.

Ogni accesso al sistema sarà registrato in un file di log che permetterà all'amministratore di avere sempre il controllo sulla situazione degli accessi.

Il software, inoltre, garantisce che non sia possibile accedere all'area utente e/o acquistare prodotti senza che l'utente sia registrato.

Nel caso vengano inseriti dati sbagliati durante la fase di accesso alla piattaforma, il sistema deve avvertire l'utente dell'errore commesso e mostrare nuovamente la schermata di accesso.

La sicurezza del profilo utente è fortemente legata alla scelta della password (una password che non richiama parole di uso comune, che ha al suo interno lettere maiuscole e minuscole e che è caratterizzata dalla presenza di caratteri numerici sarà più sicura di un nome proprio o di un nome che richiama il linguaggio di uso comune).

I canoni che UserID e Password devono rispettare sono i seguenti:

- *UserID*: deve avere una lunghezza compresa fra un minimo di 5 ed un massimo di 15 caratteri.
- *Password*: deve avere una lunghezza compresa fra un minimo di 8 ed un massimo di 20 caratteri, non può contenere caratteri speciali e deve essere diversa dall'UserID.

## Gestione del controllo globale

Il flusso di controllo globale del sistema software in questione è gestito da un **“Event-driven control”**.

Con questo tipo di controllo il flusso del programma è largamente determinato dal verificarsi di eventi esterni infatti il flusso di eventi principale una volta ricevuto un input lo trasferisce al relativo oggetto che manda in esecuzione un sottosistema.

È stata fatta questa determinata scelta perché tale meccanismo di flusso è particolarmente flessibile ed adatto alle interfacce utenti e soprattutto sfrutta appieno le potenzialità del linguaggio di programmazione usato per sviluppare questo sistema.

## Boundary conditions

### Configuration

Di seguito sono riportate le fasi in cui gli oggetti vengono creati, resi persistenti o distrutti.

- *Ordine*
  - *Viene creato ogni volta che un cliente procede all'acquisto di beni e/o servizi;*
  - *Sono memorizzati sul database e vengono associati ai clienti;*
  - *Vengono distrutti nel caso in cui l'ordine è errato o viene annullato dal cliente.*
- *Dipendente*

- *Viene creato dall'admin nel caso in cui l'organico aziendale viene ampliato;*
- *È memorizzato nel database sia durante il periodo lavorativo e sia in caso di licenziamento;*
- *Può essere distrutto dall'admin.*
- *Spedizione*
  - *Si crea una volta che un ordine viene confermato;*
  - *È memorizzato nel database e viene associato ad un cliente e ad un ordine;*
  - *Viene distrutto nel caso in cui l'ordine rientra per motivi particolari o nel caso in cui i dati sono errati.*
- *Fornitore*
  - *Viene creato dall'admin;*
  - *È memorizzato in modo permanente nel database;*
  - *Viene distrutto nel caso in cui non ci sono più operazioni commerciali con esso.*
- *Cliente*
  - *Viene creato nel momento in cui qualcuno effettua la registrazione sulla piattaforma;*
  - *È memorizzato in modo permanente sul database e identifica univocamente un cliente collegandolo a tutti gli ordini che ha effettuato;*
  - *Viene distrutto nel caso in cui un utente viene bannato dall'admin oppure lui stesso si elimina dal sito;*
- *Riparazione*
  - *Viene creata dal cliente nel momento in cui è richiesta;*
  - *È memorizzata nel database in modo permanente ed è collegata al cliente che l'ha richiesta;*
  - *Può essere distrutta dall'admin.*
- *Prodotto*
  - *Viene creato dall'admin e poi aggiornato in termini di prezzi e quantità;*
  - *È memorizzato nel database in modo permanente e viene costantemente aggiornato in base alle variazioni della quantità o a eventuali aumenti/diminuzioni di prezzo;*
  - *Può essere distrutto dall'admin nel caso in cui il prodotto viene eliminato dal catalogo mentre se non è più disponibile in magazzino esso viene comunque conservato fino a che non viene aggiornata la quantità.*
- *Vendita*
  - *Viene creata dal cliente nel momento in cui finalizza l'acquisto di un bene e/o servizio;*
  - *È memorizzato nel database ed è collegato al cliente;*
  - *Può essere distrutto dal cliente nel caso in cui annulli l'acquisto o dall'admin se vengono riscontrati problemi.*

## Exception handling

- *Connessione al database fallita:* Errore del sistema durante la connessione al database, questo comporta l'impossibilità di accedere ai dati e di inserire dati. In questo caso, il sistema, riprova la connessione e nel caso in cui non riesce a connettersi al database comunicherà l'errore all'admin e tenterà la connessione ad un database di backup.
- *Interruzione inaspettata del sistema:* essa può accadere per vari motivi come ad esempio un blackout elettrico (gestito attraverso l'installazione di un gruppo di continuità) oppure per un errore di sistema che lo costringe al riavvio. Questo tipo di problematiche sono gestite singolarmente ma comunicano all'admin ogni errore che si presenta.

## Servizi sottosistemi

Di seguito riportiamo i servizi forniti da ognuno dei sottosistemi in termini di operazioni

### Interface layer

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

### Application logic layer

<b>Nome</b>	<i>Application logic layer</i>
<b>Descrizione</b>	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.
<b>Moduli presenti</b>	
<b>Modulo</b>	<b>Descrizione</b>
<i>Gestione utente</i>	Il modulo si occupa della gestione generale dell'utente. Creazione, modifica, gestione, cancellazione.
<i>Gestione magazzino</i>	Il modulo si occupa della gestione generale del magazzino.
<i>Gestione prodotti</i>	Il modulo gestisce i prodotti.
<i>Gestione servizi</i>	Il modulo gestisce i servizi.
<i>Gestione vendite</i>	Il modulo gestisce le vendite.
<i>Gestione interfaccia</i>	Il modulo si occupa di ricevere e interpretare gli input.
<i>Gestione database</i>	Il modulo si occupa della comunicazione con il database.

<b>Nome</b>	<i>Storage layer</i>
<b>Descrizione</b>	Questo livello gestisce l'archiviazione persistente dei dati.
<b>Moduli presenti</b>	
<b>Modulo</b>	<b>Descrizione</b>
<i>Database</i>	Gestisce le richieste di dati in entrata e in uscita. Si occupa del database e della sua gestione.



## Glossario

<b>Termine</b>	<b>Descrizione</b>
<i>Sistema</i>	Si intende il sistema che verrà sviluppato.
<i>Architettura</i>	Struttura del sistema.
<i>Attori</i>	Gli utenti tipo che utilizzeranno il sistema.
<i>Affidabilità</i>	L'attitudine del sistema a svolgere la funzione richiesta in determinate condizioni per un dato intervallo di tempo.
<i>UserID</i>	Nome identificativo dell'utente
<i>Estensibilità</i>	Capacità di estensione del sistema.
<i>Client</i>	Chi accede ai componenti e ai servizi di un server.
<i>Server</i>	Chi offre componenti e servizi.
<i>Boundary conditions</i>	Condizioni di uscita/errore/inizializzazione/chiusura
<i>Event-driven control</i>	Controllore di eventi