



# SDD

## SYSTEM DESIGN DOCUMENT



# 1 Introduction

## 1.1 Purpose Of The System

Il sistema è stato progettato al fine di migliorare la distribuzione e la pubblicizzazione dei prodotti del negozio di elettronica. Precedentemente alla realizzazione del sistema l'approccio col cliente era esclusivamente fisico, quindi limitato e disorganizzato, con la nascita del sistema questo ambito viene migliorato. Ovviamente c'è un incremento dei guadagni e di conseguenza un'evoluzione del negozio. Tutto ciò rende la gestione delle informazioni, come l'aggiornamento dei prodotti, l'appuntamento con il personale, più veloce e versatile. A differenza delle grandi aziende che mirano ad una clientela più ampia, una piccola realtà come "Tutto Elettronica" può offrire maggiore assistenza ai singoli clienti.

## 1.2 Design Goals

- **Usability:** L'interfaccia grafica permetterà all'utente di poter selezionare in maniera molto semplice le varie funzioni concesse dal sito. Per facilitare la ricerca di uno specifico prodotto nella home è presente una barra di ricerca che permette di visualizzare il prodotto semplicemente e rapidamente.  
La schermata "HOME" del sito permette di:
    - i. visualizzare i prodotti più prenotati;
    - ii. effettuare la Log-In;
    - iii. andare alla sezione per la riparazione dei prodotti.
  - **Performance:** Il software dovrà rispondere velocemente; le risposte dovranno essere fornite in un periodo pari a circa 1 secondo. Il numero di utenti che potranno collegarsi e acquistare prodotti contemporaneamente sarà dato dalla disponibilità del Web Server utilizzato. La latenza massima di attesa, per una risposta, non dovrà superare i 30 secondi.
  - **Reliability:** Il software dovrà essere attivo 24 ore su 24. Inoltre, deve garantire la sicurezza su tutte le operazioni effettuate sia dai gestori sia dai clienti. Nel caso in cui si verificano dei comportamenti anomali, da parte del gestore o del cliente, verranno notificati tramite degli avvisi.
- Supportability:** Il software consisterà in un sistema client-server in cui il server sarà disponibile su ogni tipo di piattaforma e il client potrà collegarsi a tale server mediante un qualsiasi browser. Il software dovrà essere suddiviso in vari moduli per permettere una più facile modifica e aggiornabilità in futuro.



### 1.3 Definizione, acronimi e abbreviazioni

Al momento non sono presenti definizioni, acronimi e abbreviazioni.

### 1.4 Riferimenti

Nella realizzazione del sistema ci siamo confrontati con i più grandi e-commerce sul mercato come Amazon, Ebay.

## 2 PROPOSED SOFTWARE ARCHITECTURE

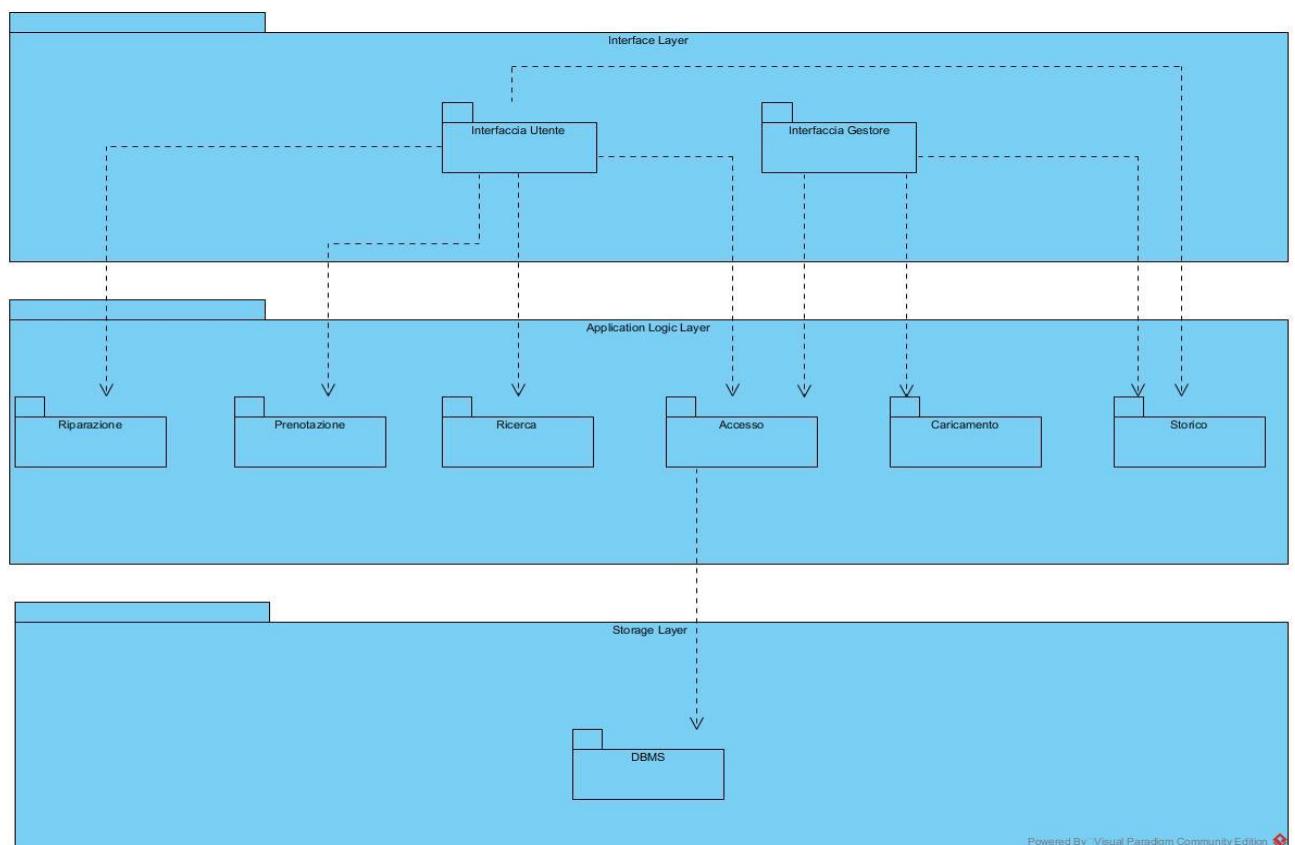
### 2.1 Overview

Si propone un'architettura software ibrida. L'architettura software proposta è un'architettura client-server organizzata mediante una architettura Model-View-Control.

Sul lato client, dell'architettura proposta, è presente la parte View del software.

Sul lato server, invece, è presente la parte Control e la parte Model del software.

### 2.2 Subsystem Decomposition





Utilizziamo un'architettura three-tier. I sottosistemi individuati sono:

<b>Interface Layer</b>	
<b>Interfaccia utente</b>	Fornisce il servizio relativo all'implementazione
<b>Interfaccia Gestore</b>	fornisce il servizio relativo all' implementazione dell' interfaccia vista dai gestori.

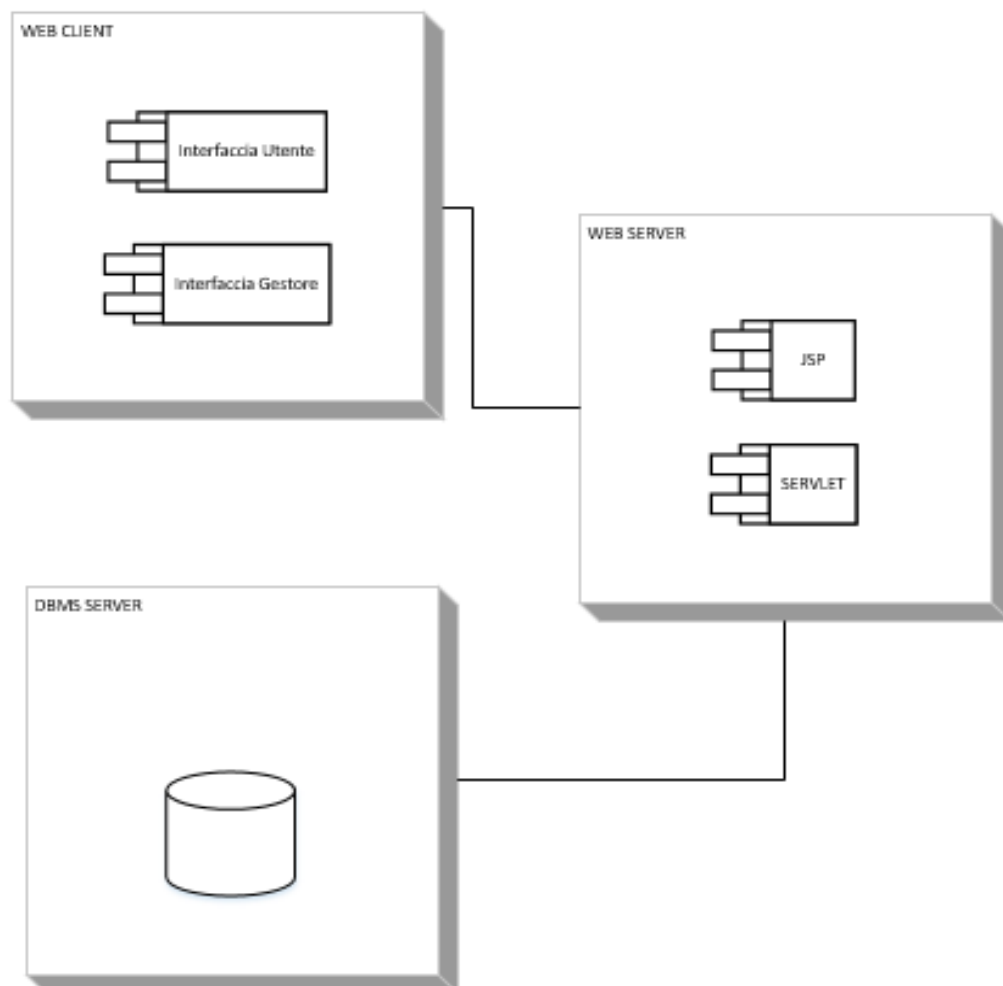
<b>Application Logic Layer</b>	
<b>Accesso</b>	<ul style="list-style-type: none"><li>a. <b>Login:</b> servizio che permette l' autenticazione al sistema.</li><li>b. <b>Registrazione:</b> servizio che permette la registrazione al sistema.</li><li>c. <b>Logout:</b> servizio che permette il logout dal sistema</li></ul>
<b>Ricerca</b>	Fornisce il servizio per la ricerca di un prodotto tramite il nome, la marca, il modello e il codice
<b>Riparazione</b>	Fornisce i servizi inerenti alla prenotazione di una riparazione di un prodotto: <ul style="list-style-type: none"><li>a. <b>Prenota data:</b> servizio che permette al cliente di scegliere una data in cui portare il prodotto in riparazione</li><li>b. <b>Invia dettagli:</b> servizio che permette al cliente di inviare i dettagli sul prodotto malfunzionante.</li></ul>
<b>Prenotazione</b>	Fornisce i servizi inerenti alla prenotazione di uno o più prodotti: <ul style="list-style-type: none"><li>a. <b>Aggiungi al carrello:</b> servizio che permette di aggiungere uno o più prodotti al carrello.</li><li>b. <b>Prenota prodotto:</b> servizio che permette di prenotare i prodotti inseriti nel carrello.</li><li>c. <b>Elimina prodotto:</b> servizio che permette di eliminare uno o più prodotti dal carrello.</li></ul>
<b>Caricamento</b>	Fornisce il servizio per il caricamento di un prodotto, una data o un ruolo utente da parte di un gestore
<b>Storico</b>	Fornisce i servizi inerenti alla visualizzazione delle operazioni effettuate: <ul style="list-style-type: none"><li>a. <b>Storico Cliente:</b> servizio che permette ad un cliente di visualizzare lo storico inerente ai prodotti prenotati o alle riparazioni prenotate.</li><li>b. <b>Storico dei clienti:</b> servizio che permette ad un gestore di visualizzare gli storici dei prodotti acquistati fino a quel momento dagli utenti.</li></ul>

<b>Storage Layer</b>	
<b>DBMS</b>	Fornisce il servizio che si occupa di interfacciare le richieste effettuate, dai vari sottosistemi, al database.

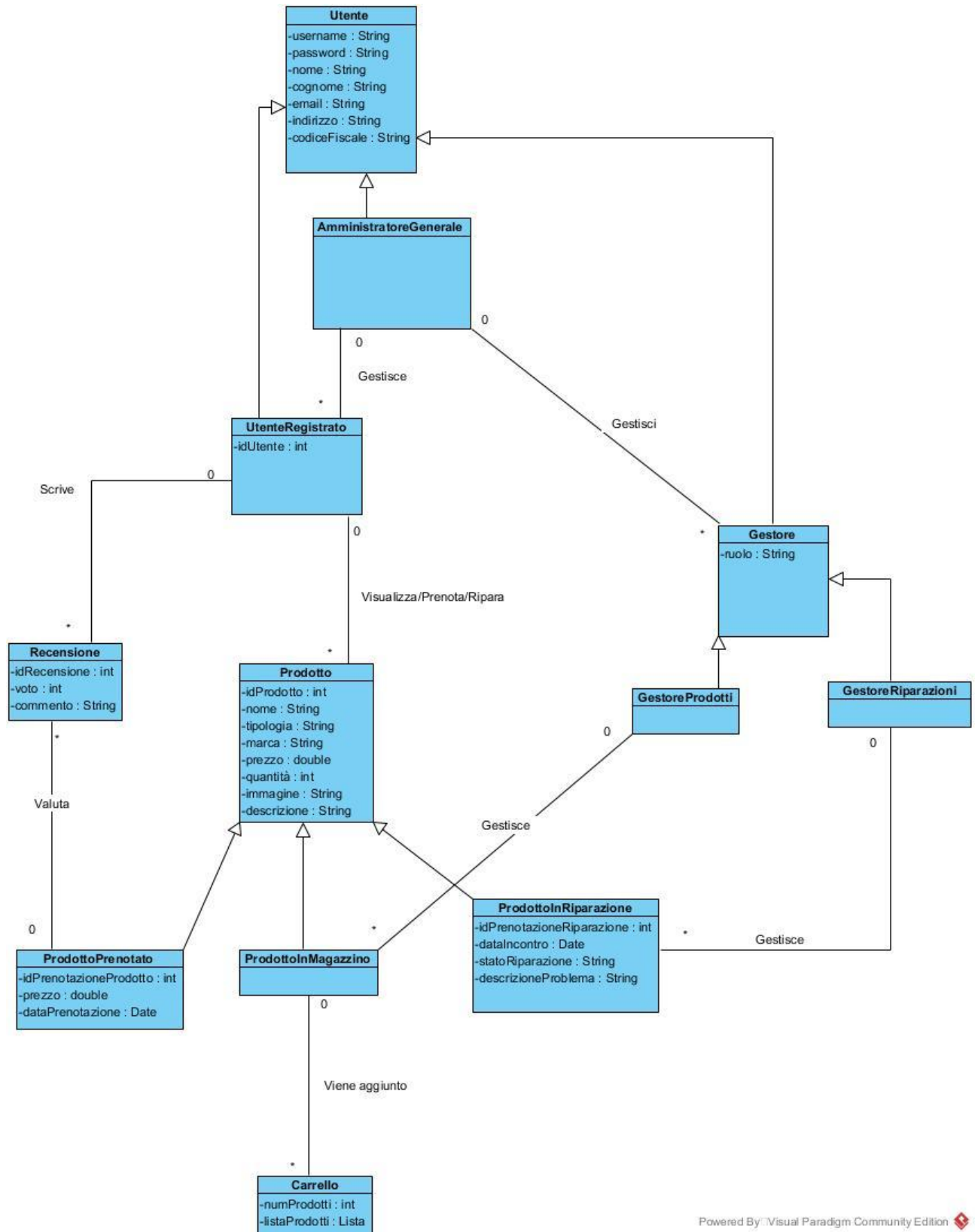
## 2.3 Hardware/Software Mapping



Il seguente diagramma di distribuzione UML illustra la mappatura hardware/software del sistema.

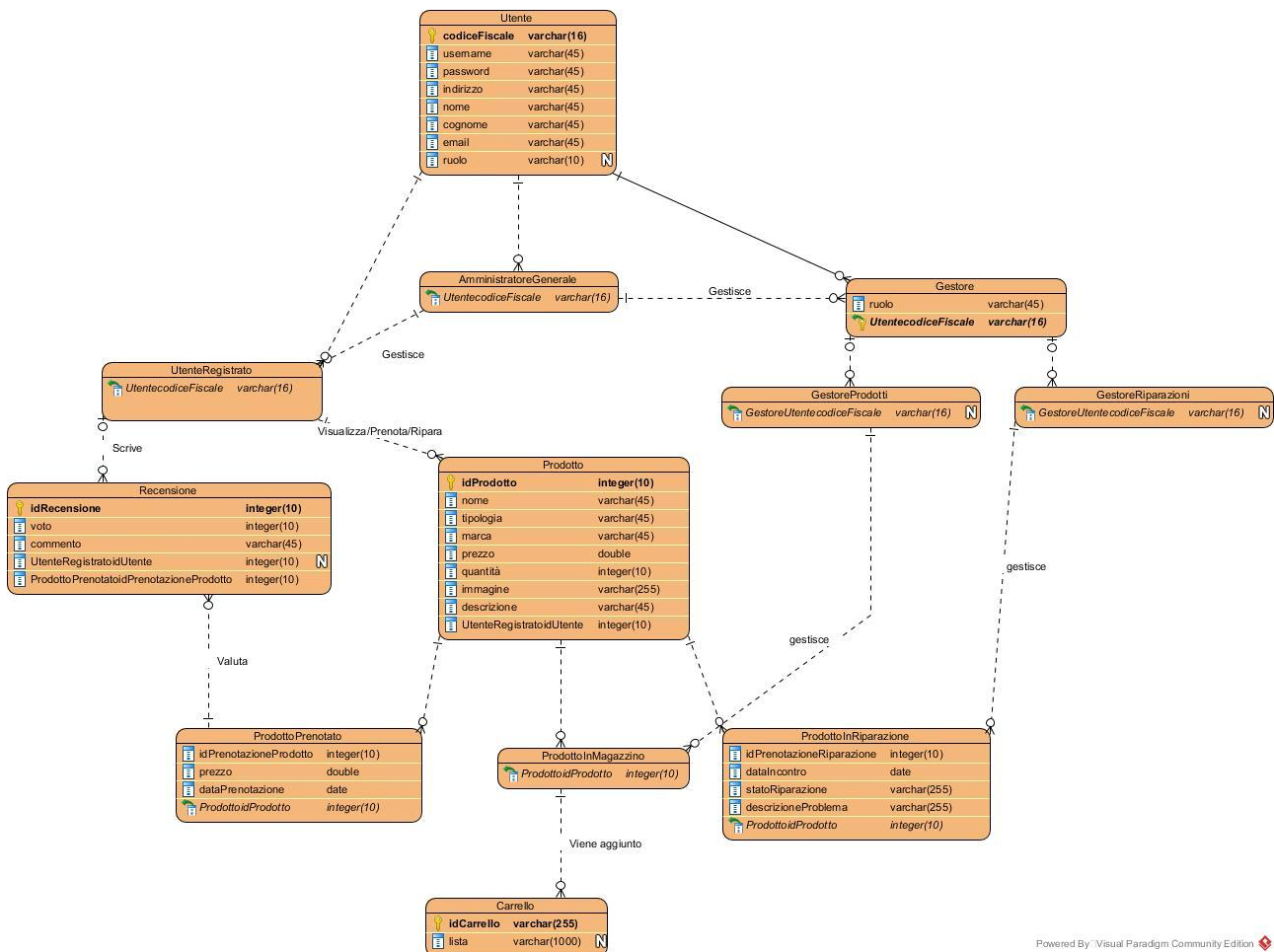


## 2.4 Persistent Data Manager





TERMINE	DESCRIZIONE	SINONIMI
Utente	Persona che si registra al sito utilizzato per la prenotazione dei prodotti online.  Persona che assolve il ruolo di Gestore del sito di vendita di prodotti online.	Cliente Gestore
ProdottoInMagazzino	Prodotto a disposizione per la vendita	
ProdottoInRiparazione	Prodotto consegnato per essere riparato	
ProdottoPrenotato	Prodotto che è stato prenotato da un utente	
Recensione	Commento su un prodotto prenotato effettuato da un utente del sito	
Carrello	Lista dei prodotti selezionati da un utente	





#### UTENTE\_TABLE

Codice Fiscale	Varchar(16), primary key
Username	Varchar(16)
Password	Varchar(16)
Nome	Varchar(16)
Cognome	Varchar(16)
Email	Varchar(30)
Indirizzo	Varchar(30)
Ruolo	Varchar(10)

#### GESTORE\_TABLE

Codice Fiscale	Varchar(16), primary key
Username	Varchar(16)
Password	Varchar(16)
Nome	Varchar(16)
Cognome	Varchar(16)
Email	Varchar(30)
Indirizzo	Varchar(30)
Ruolo	Varchar(30)

#### PRODOTTOPRENOTATO\_TABLE

ID_Prodotto	Int, PrimaryKey
Nome	Varchar(16)
Tipologia	Varchar(16)
Marca	Varchar(16)
Prezzo	numeric(10,2)
Quantità	Int
Immagine	Varchar(100)
Descrizione	Varchar(300)
DataPrenotazione	date





PRODOTTOINRIPARAZIONE\_TABLE

ID_Prodotto	Int, PrimaryKey
Nome	Varchar(16)
Tipologia	Varchar(16)
Marca	Varchar(16)
Prezzo	numeric(10,2)
Quantità	Int
Immagine	Varchar(100)
Descrizione	Varchar(300)
Data_Incontro	Date
StatoRiparazione	Varchar(255)
DescrizioneProblema	Varchar(255)
IDPrenotazioneRiparazione	int

PRODOTTOINMAGAZZINO\_TABLE

ID_Prodotto	Int, PrimaryKey
Nome	Varchar(16)
Tipologia	Varchar(16)
Marca	Varchar(16)
Prezzo	numeric(10,2)
Quantità	Int
Immagine	Varchar(100)
Descrizione	Varchar(300)

CARRELLO\_TABLE

IDCarrello	Int
Lista	Varchar(1000)



Il sistema Tutto Elettronica si avvale dell'uso di un database di tipo relazionale. Questa scelta è stata ponderata data la sicurezza offerta da un DBMS di ultima generazione, assieme ad una maggiore affidabilità e garanzia di coerenza e facilità di gestione, nonché dalla velocità di accesso e trasmissione dei dati.

Un database di tipo relazionale, inoltre, permette la gestione e la memorizzazione permanente di un grosso insieme di dati che devono e/o possono essere acceduti da utenti e applicazioni diverse a una granularità più fine.

Ovviamente, utilizzare un database relazionale significa avere a disposizione circa il triplo dello spazio di memorizzazione richiesto per il corrispondente insieme di dati.

## 2.5 Access Control And Security

ATTORI OGGETTI	Utente non registrato	Utente Registrato	Gestore Prodotti	Gestore Riparazioni
Prodotto Prenotato		Visualizza Recensisci		
Prodotto in magazzino	Visualizza	Visualizza Aggiungi al carrello	Aggiunge Modifica Rimuove	
Prodotto in riparazione		Visualizza Visualizza stato		Modificare stato riparazione Inserire data di fine riparazione
Recensione		Visualizza		
Carrello		Effettua prenotazione Rimuovi prodotto		

Modelliamo gli accessi alle classi con una **Matrice di Accesso**: le colonne rappresentano gli attori del sistema e le righe rappresentano le classi di cui vogliamo controllare l'accesso. Un **Diritto di Accesso** è un'entrata nella matrice degli accessi ed elenca le operazioni che possono essere effettuate dagli attori sulle istanze delle classi.



## 2.6 Global Software Control

Il sistema è un'applicazione Web gestita attraverso un server remoto. L'utente interagisce con le classi del sottosistema UI, dopo l'interazione con queste classi gli eventi vengono creati e inviati a oggetti remoti.

Per quanto riguarda il flusso di controllo esterno (fra sottosistemi), il server sarà sempre in funzione, in attesa di eventuali richieste di servizi da parte delle varie postazioni client; nel caso di più richieste contemporanee, il server utilizzerà la tecnica FIFO (First In First Out).

## 2.7 Boundary Condition

**Start-Up** : descrive come il sistema o una componente è portato da uno stato non inizializzato a uno stato stabile (startup use case).

**ShutDown**: descrive quali risorse sono rilasciate e quali sistemi vengono notificati al momento della terminazione del sistema o di una componente (termination use case).

**Failure** (gestione delle eccezioni): le cause che li provocano possono essere molte (bug, errori, problemi esterni, come l'alimentazione elettrica) e buoni system design devono prevedere fatal failure (failure use case). I casi d'uso eccezionali estendono i casi d'uso più rilevanti.

<b>Nome Use Case:</b>	Start-Server
<b>Condizioni di entrata:</b>	Il ServerAdministrator accede al server
<b>Flusso di eventi:</b>	<ol style="list-style-type: none"><li>1. Dopo aver effettuato correttamente l'accesso viene eseguito dal ServerAdministrator lo start-Up del server</li><li>2. Se Il Server era stato precedentemente chiuso normalmente, il server legge l'elenco dei driver legittimi e la lista dei prodotti in vendita e prenotati. Se il server si è arrestato in modo anomalo, notifica al ServerAdministrator ed esegue un controllo di coerenza ProductDBStore</li></ol>
<b>Condizioni di uscita:</b>	<ul style="list-style-type: none"><li>• Il Server è attivo e riceve una richiesta da parte del Client</li></ul>

ProductDBStore: è responsabile della memorizzazione dei prodotti in vendita e prenotati in un database. Questo sottosistema supporta più driver simultanei. All'avvio, il ProductDBStore rileva se è stato spento correttamente, in caso contrario esegue un controllo sui prodotti in vendita e prenotati e ripara i dati danneggiati se necessario.



<b>Nome Use Case:</b>	ShutDown-Server
<b>Condizioni di entrata:</b>	Il ServerAdministrator esegue l' operazione per arrestare il server
<b>Flusso di eventi:</b>	<ol style="list-style-type: none"><li>1. Viene eseguito dal ServerAdministrator lo ShutDown del server</li><li>2. Se il server si arresta correttamente, vengono rilasciate tutte le risorse allocate, se il server invece si arresta in modo anomalo il ProductDBStore salva tutte le operazioni effettuate dal server prima del failure</li></ol>
<b>Condizioni di uscita:</b>	<ul style="list-style-type: none"><li>• Il Server si arresta</li></ul>

