

**WhatShop**  
**System Design Document**  
**Versione 2.2**



Data: 25/11/2016

Progetto: WhatShop	Versione: 2.2
Documento: System Design Document	Data: 25/11/2016

### Coordinatore del progetto:

Nome	Matricola

### Partecipanti:

Nome	Matricola
D'Avanzo Giuseppe	0512102892
Cerreto Luigi	0512102940
Manganiello Alessandro	0512103246
Barone Gerardo	0512103264

<b>Scritto da:</b>	Giuseppe D'Avanzo, Cerreto Luigi, Manganiello Alessandro
--------------------	--

## Revision History

Data	Versione	Descrizione	Autore
17/11/2016	0.1	Inizio SDD	GDA
18/11/2016	0.5	introduzione e scopo del sistema	LC
20/11/2016	0.8	Obiettivi del design	AM
21/11/2016	1.0	Definizioni, acronimi, abbreviazioni e Riferimenti	GDA, LC
22/11/2016	1.1	Architettura software corrente e proposta	GDA,LC,AM
23/11/2016	1.5	Servizi dei sottosistemi	GDA, LC, AM
24/11/2016	2.0	Glossario e Revisione completa SDD	GDA, LC , AM
25/11/2016	2.2	Completamento SDD	LC, GDA

# Indice

1	INTRODUZIONE .....	4
1.1	Scopo del sistema .....	4
1.2	Obiettivi di design .....	4
1.2.3	Criteri di Manutenzione .....	7
1.2.4	Criteri per l'Utente Finale .....	7
1.3	Definizioni, acronimi e abbreviazioni .....	8
1.4	Riferimenti .....	8
1.5	Panoramica .....	8
2.0	Architettura software corrente .....	9
3.0	Architettura software proposta .....	10
3.1	Overview .....	10
3.2	Decomposizione del sistema .....	12
3.3	Mapping Hardware/Software .....	14
3.4	Gestione dei dati persistenti .....	15
3.5	Controllo degli accessi e della sicurezza .....	15
3.6	Controllo del software globale .....	16
3.7	Condizioni Boundary .....	17
4.	Servizi dei sottosistemi .....	20
4.1	Gestione Utenti .....	20
4.2	Gestione Amministratore .....	21
4.3	Gestione Carrello .....	22
5.0	Glossario .....	24

# **1 INTRODUZIONE**

## **1.1 Scopo del sistema**

Lo scopo del sistema è quello di mettere in vendita a tutti gli utenti oggetti di uso comune di diversi negozi su un unico portale.

L'utente non dovrà più acquistare gli oggetti presso più negozi online ma, attraverso un unico sito.

Questo comporta un risparmio di tempo da parte dell'utente e anche un risparmio da parte del venditore, il quale non avrà la necessità di creare un proprio sito e-commerce ma dovrà inserire solo i propri prodotti all'interno di questa grande piattaforma.

Inoltre l'utilizzo di questa piattaforma da parte di più venditori creerà una vera e propria competizione tra di loro, facendo una vera e propria gara al prezzo più basso. La piattaforma, quindi, avrà sempre prezzi più bassi rispetto a tutti gli altri siti e-commerce in circolazione, permettendo un risparmio notevole all'utente medio.

## **1.2 Obiettivi di design**

Il sistema WhatShop deve poter essere il più efficiente ed intuitivo possibile. Tale efficienza sarà costruita attraverso rapidi tempi di risposta ad ogni genere di input ma anche differenti politiche di tolleranza all'errore (Banna Utente). In più si punterà ad aver una buona manutenibilità attraverso il facile inserimento di nuove funzionalità. Sarà intuitivo attraverso agevoli interfacce. Il sistema quindi sarà accessibile a tutti.

### 1.2.1 Criteri di Performance

<b>Tempo di risposta</b>	WhatShop deve assicurare una risposta rapida alle richieste dell'utente. Una semplice richiesta dell'utente deve essere gestita ed elaborata entro 10 secondi, nel caso di una connessione molto lenta . I tempi di risposta saranno più rapidi in base alla velocità della connessione utilizzata.
<b>Throughput</b>	Il sistema sarà capace di gestire contemporaneamente diversi utenti. Non è possibile avere una stima certa del carico ma, mediamente, si prevede un carico mensile di 100 utenti con acquisti in media di 2 prodotti per utente.
<b>Robustezza</b>	WhatShop deve gestire eventuali input errati senza interrompere il funzionamento dell'intero sistema, attraverso delle notifiche di errore, ad esempio come nome o password errata.
<b>Disponibilità</b>	Whatshop deve essere disponibile all'uso, 24 ore su 24, da parte degli utenti, grazie ad un server sempre attivo.
<b>Tolleranza all'errore</b>	Whatshop deve essere capace di operare durante condizioni d'errore; In modo tale che l'errore non intacchi le funzionalità di un altro.

### **1.2.2 Criteri di Affidabilità**

<b>Robustezza</b>	WhatShop deve gestire eventuali input errati senza interrompere il funzionamento dell'intero sistema, attraverso delle notifiche di errore.
<b>Disponibilità</b>	WhatShop deve essere disponibile all'uso, 24 ore su 24, da parte degli utenti, grazie ad un server sempre attivo.
<b>Tolleranza all'errore</b>	WhatShop deve essere capace di operare durante condizioni d'errore. Ciò sarà reso possibile grazie alle sottoparti in cui è stato implementato il sistema in modo tale che l'errore in un sottosistema non intacchi le funzionalità di un altro.
<b>Sicurezza</b>	L'accesso al sistema è controllato da un sistema di login che permetterà all'utente, se loggato, di effettuare gli acquisti con la massima sicurezza. Il database offre sicurezza dei dati personali come numero di carta di credito o password di accesso

### **1.2.3 Criteri di Manutenzione**

Estendibilità	WhatShop dovrà essere propenso all'inserimento di nuove funzionalità. Quindi, è necessario che il codice scritto sia ben strutturato, dunque, di facile comprensione per evitare ambiguità nella scrittura successiva. Le varie classi sono divise in pacchetti in modo che sià più facile individuare la classe da modificare
Modificabilità	Deve essere possibile effettuare eventuali modifiche al sistema (ad es. in caso di bug). Anche in questo caso, è necessario che il codice sia ben strutturato e di facile comprensione.
Leggibilità	Il codice sarà comodo da leggere grazie ad una accurata indentazione di quest'ultimo. Inoltre, sarà facile da comprendere le diverse parti di cui il codice è composto ci saranno opportuni commenti che ne spiegano il funzionamento e anche grazie ai semplici nomi attribuiti alle variabili.

### **1.2.4 Criteri per l'Utente Finale**

Usabilità	WhatShop rende ogni funzione di semplice uso, garantendo un'ottima esperienza all'utente grazie ad interfacce grafiche intuitive.
-----------	---

### **1.3 Definizioni, acronimi e abbreviazioni**

- WhatShop: Nome del sistema che verrà sviluppato.
- RAD: Requirements Analysis Document.
- DBMS: Database Management System.
- Admin: Amministratore del sistema.
- Carrello: Carrello elettronico contenente i prodotti da acquistare.
- Utente: Utente Registrato o non alla piattaforma.

### **1.4 Riferimenti**

Documento RAD del progetto WhatShop.

### **1.5 Panoramica**

Prima di parlare dell'architettura: è importante fare un accenno alle attività di system design che costituiscono le fondamenta per l'architettura software del sistema.

- Decomposizione del sistema: in cui il sistema viene suddiviso in diversi sottosistemi ognuno dei quali, a sua volta, è caratterizzato da servizi che offre ad altri sottosistemi.
- Mapping Hardware/Software: riguardante la scelta della configurazione hardware del sistema, la comunicazione tra nodi, il come vengano incapsulati i servizi di un sottosistema.
- Gestione dei dati persistenti: nel quale si individuano gli oggetti che devono essere resi persistenti e quale genere di infrastruttura si deve usare per memorizzare tali oggetti.
- Politiche di accesso e Sicurezza: che ci aiuta a rappresentare tramite delle tabelle le operazioni ed informazioni utilizzabili da ogni singolo attore.

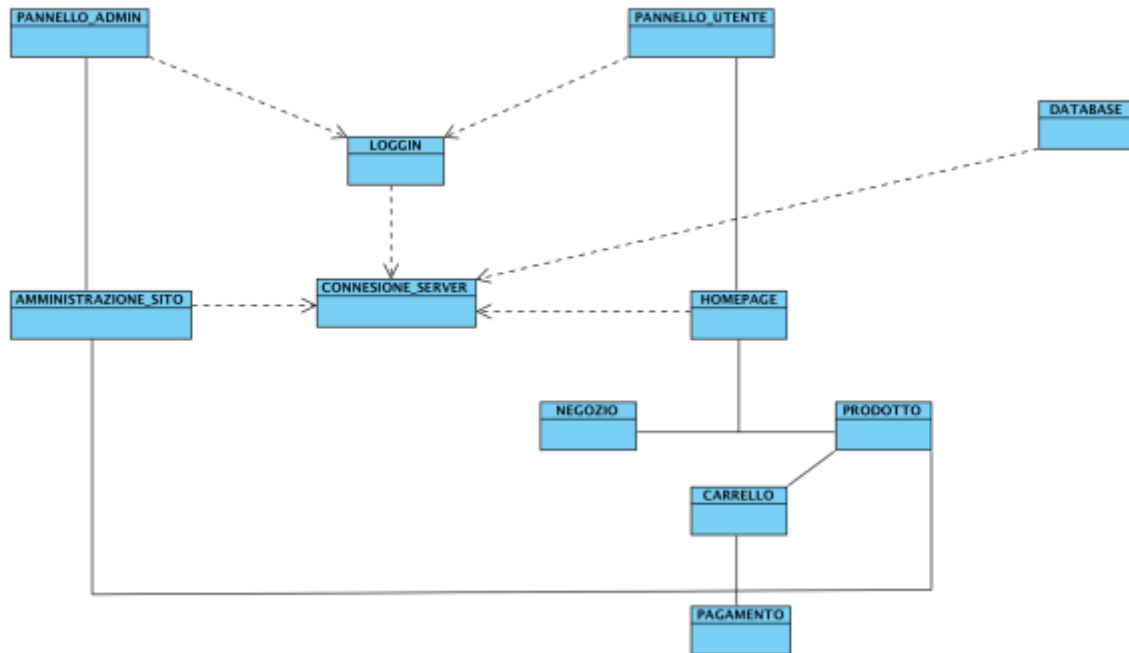


- Controllo del software globale: che ci guida su quali operazioni eseguire ed in che ordine, per garantire il corretto flusso di controllo del sistema.
- Condizioni Boundary: che includono oltre l'avvio e lo shutdown anche la gestione dei fallimenti dovuti all'invecchiamento del sistema, interruzione di corrente o anche a errori di progettazione

## **2.0 Architettura software corrente**

Nella progettazione del sistema i progettisti si sono potuti avvalere della possibilità di confrontarsi con altre piattaforme. La concorrenza attuale è di fascia medio-alta in quanto esistono poche piattaforme che offrono lo stesso servizio di WhatShop, uno dei grandi rivali è il colosso Amazon.

### 3.0 Architettura software proposta



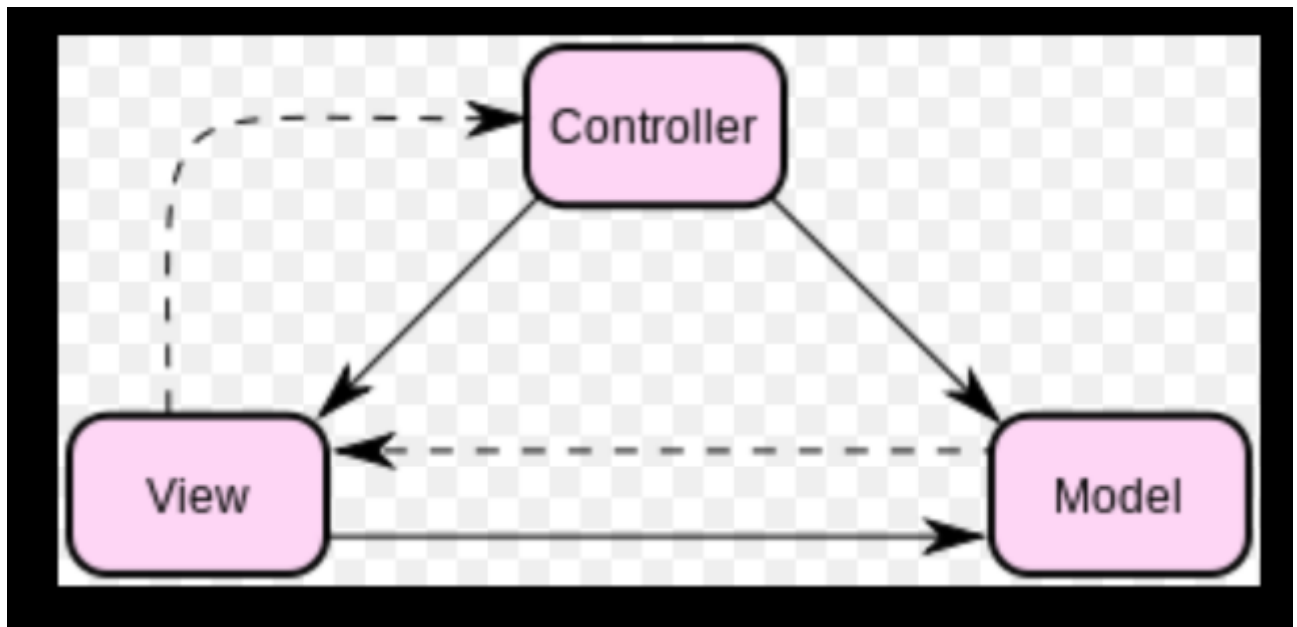
#### 3.1 Overview

L'architettura del sistema WhatShop è di tipo client/server. Il server riceve le richieste da parte del client, e risponde in tempo utile. I motivi di questa scelta sono:

- Portabilità: il sistema potrà essere utilizzato su una varietà di macchine e sistemi operativi, da computer fissi a dispositivi mobili.
- Performance: il client sarà in grado di supportare task interattivi e il server dovrà fornire operazioni CPU-intensive.
- Flessibilità: per ogni tipologia di utente che effettua l'accesso al sistema, vi sarà un'interfaccia grafica apposita, tramite la quale ogni attore potrà eseguire le operazioni che la piattaforma offre.

- Affidabilità: entrambi i componenti client e server devono essere affidabili ed essere in grado di mantenere i propri dati anche in seguito a guasti, quindi deve essere possibile effettuare dei backup periodici al database con cadenza trimestrale.

L'applicazione WhatShop è basata sull'architettura Model View Controller



Questo pattern architetturale è molto usato nella programmazione orientata agli oggetti in quanto permette di separare la logica di presentazione (trasformare i dati ricevuti dal livello di applicazione) dalla logica di business (la parte che rende operativa l'applicazione).

Il modello gestisce la logica e le regole dell'applicazione, oltre ai dati.

Una vista ha il compito di trasmettere in output le informazioni.

Il controller ha il compito di tradurre l'input in comandi per la vista o il modello

### 3.2Decomposizione del sistema

Descrizione di ogni modulo.

View	
Utente Amministratore	Modulo che si occupa di gestire le funzionalità del sistema riservate all'amministratore, come la gestione degli utenti e dei prodotti come il ban di un utente o l'aggiunta, la modifica o la rimozione di un prodotto dalla piattaforma
Utente Visitatore del sito	Modulo che si occupa di gestire le funzionalità del sistema riservate all'utente, come la visualizzazione dei prodotti che offre la piattaforma e l'aggiunta di un prodotto al carrello per poi passare all'acquisto.

Model	
<b>Gestione Utenti</b>	Modulo che permette la registrazione di un nuovo utente al sistema, gestisce l'accesso degli utenti, garantendo l'adeguata assegnazione dei privilegi dovuti (possibilità di acquistare prodotti) e le relative funzionalità.
<b>Gestione Prodotti</b>	Modulo che permette di visualizzare, creare, eliminare o modificare prodotti sulla piattaforma.
<b>Gestione Carrello</b>	Modulo che permette di visualizzare, eliminare, modificare o aumentare la quantità degli oggetti nel carrello elettronico
<b>Gestione Login</b>	Modulo che si occupa della gestione del login degli utenti, controlla se username e password coincidono con quelle esistenti nel database e successivamente permette il login ad un utente.

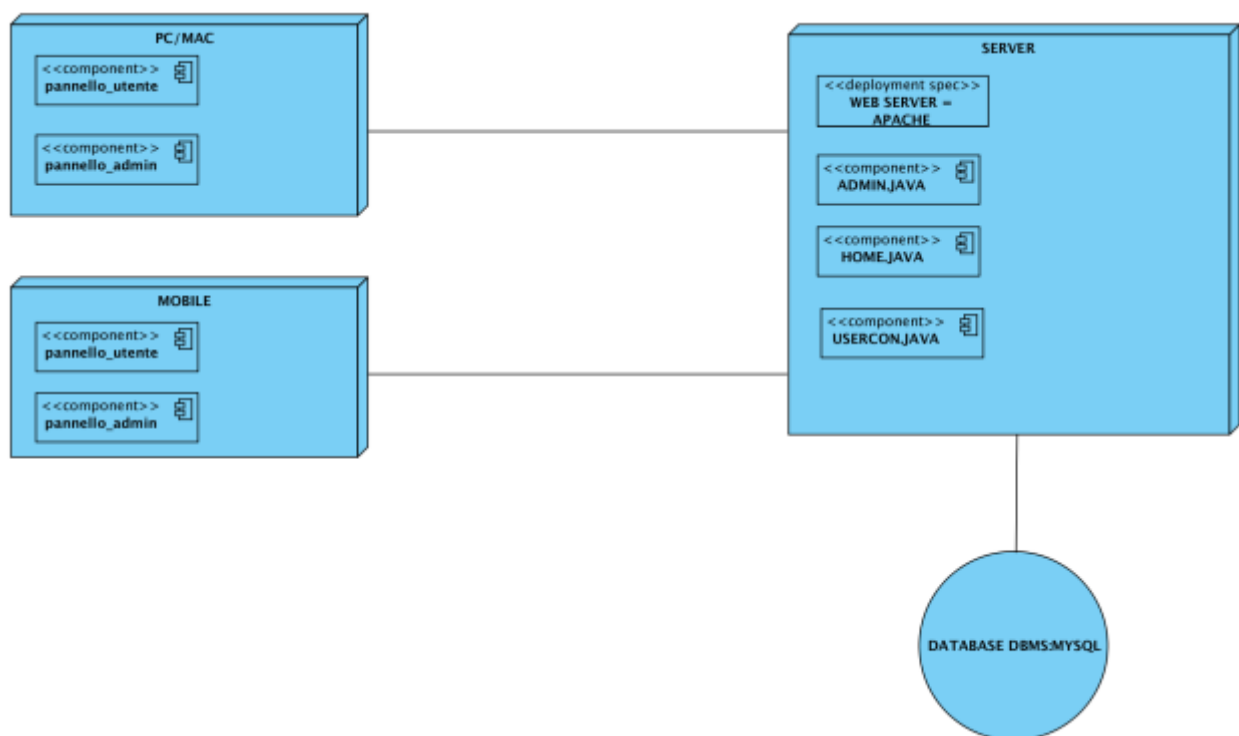
Model	
<b>Dati persistenti</b>	Modulo che si occupa di memorizzare i dati fondamentali degli utenti e dei prodotti in memoria, in modo da poter essere prelevati e modificati in modo corretto.

### 3.3 Mapping Hardware/Software

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 le postazioni client ed il database e le postazioni client dovranno effettuare richieste al server per eseguire le operazioni legate alle loro rispettive funzionalità.

Il database conterrà esclusivamente i dati relativi ai prodotti messi in vendita.

Le postazioni client e il server saranno connesse tramite una rete che utilizzerà il protocollo TCP/IP.



### **Web Server**

Il server utilizzato è Apache Tomcat versione 2.4.17.

### **Interface Layer**

L'utente utilizza il sistema mediante un Browser installato sulla propria macchina (ad es. Opera, Firefox, Chrome).

### **Application Logic Layer**

Il sistema, e quindi le funzionalità, sono implementate in linguaggio HTML. Il codice risultante viene inviato al browser del client.

### **Storage Layer**

Rappresenta il collegamento con il server da parte del sistema e si occupa di tutte le richieste di accesso e modifiche e eliminazione sui dati permanenti presenti nel database.

### **Database Server**

Il DBMS usato è MySQL il quale presenta molte API che permettono l'interazione tra sistema e database.

## ***3.4. Gestione dei dati persistenti***

Ci si ricollega al documento "WhatShop- System Design Document- Gestione Dati Persistenti"

## **3.5 Controllo degli accessi e della sicurezza**

WhatShop è un sistema multiutente, quindi utenti differenti hanno il permesso di eseguire diverse operazioni su vari insiemi di oggetti in contemporanea con altri utenti. Per schematizzare al meglio il controllo degli accessi abbiamo suddiviso per tipologia di utente le azioni consentite, al fine di ottenere una visione più compatta e dettagliata.

## ADMIN

Oggetto	Operazioni
Prodotti	Inserisci Modifica Rimuovi
Utente	Bandisci

## UTENTE

Oggetto	Operazioni
Carrello	Aggiunge prodotti Elimina prodotti Modifica Quantità Acquista prodotti

### ***3.6. Controllo del software globale***

Il controllo del flusso software viene gestito da classi html, java e jsp che interagendo con il client, il quale si interfaccia tramite un web browser, svolgono le varie operazioni. Il server smista ogni nuova richiesta alla classe adeguata, inoltrando poi la risposta al client.



### **3.7. Condizioni Boundary**

Le condizioni limite riguardano l'accensione e lo spegnimento del sistema per quanto riguarda il lato Server. Dal lato Client si riferiscono agli errori di connessione al server.

#### **Scenari**

<b>Nome Scenario</b>	Start Server
<b>Istanze di Attori Partecipanti</b>	Admin
<b>Flusso di Eventi</b>	<ol style="list-style-type: none"><li>1. L'admin decide di voler avviare il sistema e quindi clicca sul pulsante "Avvia".</li><li>2. Il sistema, con le procedure di avvio, attiva i server e i relativi servizi in remoto rendendosi disponibile ad eventuali richieste.</li><li>3. Il sistema notifica il successo della procedura.</li></ol>

<b>Nome Scenario</b>	Shutdown Server
<b>Istanze di Attori Partecipanti</b>	Admin
<b>Flusso di Eventi</b>	<ol style="list-style-type: none"> <li>1. L'admin decide di voler arrestare il sistema e quindi accede alla pagina dedicata e clicca sul pulsante "Arresta". (Arresta server Tomcat o la piattaforma di host).</li> <li>2. Il sistema effettua una scansione per verificare se ci sono ancora richieste in sospeso</li> <li>3. Il sistema porta a termine (se presenti) le richieste in sospeso</li> <li>4. Tramite le procedure di arresto il sistema disattiva tutti i servizi e il server.</li> <li>5. Il sistema notifica il successo della procedura.</li> </ol>

## Casi d'uso

<b>ID</b>	Use Case Startup
<b>Nome Caso Uso</b>	Startup Server
<b>Istanze di Attori Partecipanti</b>	Admin
<b>Condizione di Entrata</b>	L'admin accede al sistema
<b>Flusso di eventi</b>	<p>Utente: L'utente accede al sistema tramite un web browser.</p> <p>Sistema: WhatShop accende il server e attiva i servizi in remoto rendendosi disponibile per le richieste notificando il successo dell'operazione all'utente.</p>

Condizione di Uscita	Il server è attivo e i relativi servizi sono disponibili.
Eccezioni	Errore nello Startup.

ID	Use Case Shutdown
Nome Caso Uso	Shutdown Server
Istanze di Attori Partecipanti	Admin
Condizione di Entrata	L'amministratore accede al sistema.
Flusso di eventi	Utente: L'utente vuole chiudere sistema e preme il pulsante spegni.
Condizione di Uscita	Il server si è spento correttamente
Eccezioni	Errore shutdown.

4. Servizi dei sottosistemi

4.1 Gestione Utenti

Sottosistema	Gestione Utenti
Descrizione	Sottosistema che gestisce la registrazione di un utente, l'autenticazione di tutti gli utenti.
Servizi Offerti	
Servizio	Descrizione
Registra utente	Permette di inserire un nuovo utente nel database.
Login	Permette ad un utente di poter effettuare l'accesso al sistema.
Logout	Permette ad un utente di uscire dal sistema.
Visualizza Utente	Permette di visualizzare a video il profilo di un utente.

## 4.2 Gestione Amministratore

Sottosistema	Gestione Amministratore
Descrizione	Sottosistema che gestisce le operazioni che un amministratore può effettuare sulla piattaforma.
Servizi Offerti	
Servizio	Descrizione
Login	Permette ad un amministratore di poter effettuare l'accesso al sistema.
Logout	Permette ad un amministratore di uscire dal sistema.
Visualizza ordini	Permette di visualizzare a video tutti gli ordini effettuati da un utente.
Aggiungi prodotto	Permette all'amministratore di aggiungere nuovi prodotti nel database
Rimuovi prodotto	Permette all'amministratore di rimuovere prodotti dal database.
Modifica prodotto	Permette all'amministratore di modificare parzialmente prodotti già esistenti nel database.
Banna utente	Permette di bannare un utente dal sistema.

Aggiungi negozio	Permette all'amministratore di aggiungere un nuovo negozio nel database.
Aggiungi Categoria	Permette all'amministratore di aggiungere una nuova categoria nel sito web.

### 4.3 Gestione Carrello

Sottosistema	Gestione Carrello
Descrizione	Sottosistema che gestisce le operazioni che un utente può effettuare all'interno del carrello elettronico
Servizi Offerti	
Servizio	Descrizione
Aggiungi al Carrello	Permette ad un utente di aggiungere prodotti al carrello elettronico.
Rimuovi dal carrello ( Simbolo "X" )	Permette ad un utente di rimuovere un prodotto dal carrello.
Aumenta quantità	Permette ad un utente di aumentare la quantità di un prodotto presente nel carrello.
Aggiorna Totale e quantità	Permette ad un utente di aggiornare la quantità e l'importo totale dell'acquisto.
Paga	Permette ad un utente di pagare e quindi completare l'ordine.

#### 4.4. Gestione Negozio

Sottosistema	Creazione Negozio
Descrizione	Sottosistema che gestisce le operazioni che un utente può effettuare una volta creato il proprio negozio all'interno della piattaforma.
Servizi Offerti	
Servizio	Descrizione
Aggiungi Prodotto	Permette ad un utente di aggiungere prodotti al negozio.
Rimuovi Prodotto	Permette ad un utente di rimuovere un prodotto dal negozio.
Gestione Prodotto	Permette ad un utente di controllare lo stato di un prodotto.

## 5.0 Glossario

**Admin:** Il termine identifica la persona che amministra il sistema, che può essere identificato come una persona che gestisce la configurazione del sistema.

**Utente:** Il termine identifica un utente del sistema che ha la possibilità di creare test esercitativi e valutativi e sottoporli agli studenti.

**Carrello:** Il termine identifica il mezzo mediante il quale l'utente effettua un acquisto.

**RAD:** Documento di Analisi dei Requisiti.

**DBMS:** Database Management System per la gestione di database.

**Database:** Insieme di dati persistenti del sistema.

**Prodotto:** Il termine identifica un prodotto in vendita sulla piattaforma.

**Banna:** Il termine identifica l'operazione mediante il quale un utente viene rimosso dal database della piattaforma, quindi, l'utente bannato non ha più la facoltà di accedere al sito web.