

# System Design Document

## Sommario

<b>1. Introduction.....</b>	<b>2</b>
<b>1.1 Purpose of the system .....</b>	<b>2</b>
<b>1.2 Design goals .....</b>	<b>2</b>
<b>1.3 Definition, acronyms and abbreviations .....</b>	<b>3</b>
<b>1.4 References.....</b>	<b>3</b>
<b>2. Current software architecture .....</b>	<b>3</b>
<b>3. Proposed software.....</b>	<b>4</b>
<b>3.1 Subsystem Decomposition .....</b>	<b>4</b>
<b>3.2 Hardware/software mapping .....</b>	<b>4</b>
<b>3.3 Persistent data management.....</b>	<b>6</b>
<b>3.4 Access control and security .....</b>	<b>8</b>
<b>3.5 Global software control.....</b>	<b>9</b>
<b>3.6 Boundary condition.....</b>	<b>10</b>
<b>4. Subsystem services.....</b>	<b>11</b>

# 1. Introduction

## 1.1 Purpose of the system

Lo scopo del progetto TechShop è quello di riutilizzare una piattaforma software per la creazione e gestione del proprio sito e-commerce, contemporaneamente permette all'acquirente di trovare in un unico luogo varie proposte di negozi legati alla tecnologia. La piattaforma è semplice e intuitiva, per utenti inesperti grazie alla guida del sistema.

Per rendere TechShop raggiungibile dal maggior numero di utenti, deve essere disponibile su un server online, visualizzabile su tutti i browser, sia personal computer che smartphone o smart tv....

Gli utenti accederanno alla piattaforma tramite un dominio.

## 1.2 Design goals

Gli obiettivi di progettazione rappresentano le qualità desiderate della piattaforma TechShop e forniscono una serie coerente di criteri che devono essere considerati quando si prendono decisioni di progettazione. Sono stati indentificati i seguenti obiettivi di progettazione.

-Il sistema si appoggia ad un DBMS esterno, sfruttando al massimo le potenzialità di ottimizzazione per il recupero dei dati.

-usando modello MVC: abbiamo 3 diversi tipi

1) **Model**: responsabile del dominio del problema

2) **Control**: responsabile della sequenza di iterazioni con l'utente e le notifiche di visualizzazione delle modifiche nel model. Determina esplicitamente il flusso di controllo.

3) **View**: responsabile della visualizzazione degli oggetti per l'utente del dominio dell'applicazione.

-Linguaggi di programmazioni usati:

**front-end: Html** per la formattazione delle modalità di paginazione o visualizzazione grafica del contenuto, testuale e non, di una pagina web attraverso tag di formattazione, e anche per la supporto di inserimento di script e oggetti esterni;

**JavaScript e JQuery** rendono molto più semplici le operazioni di spostamento e manipolazione dei documenti HTML, gestione degli eventi, animazione e Ajax con un'API di facile utilizzo che funziona su una moltitudine di browser;

**CSS** per la formattazione e di documenti HTML.

**back-end: Java** viene scelto per interoperabilità di interfacciarsi, scambiare informazioni o servizi con altri sistemi o prodotti in maniera più o meno completa e priva di errori, con affidabilità e con ottimizzazione delle risorse.

- **Efficienza:** TechShop risposta all'autorizzazione entro 10 secondi, nel 95% dei casi. Qualora ci siano problemi di connettività, il sistema deve comunque garantire la registrazione in locale del pagamento, in modo da permettere il normale acquisto nei negozi. Supporto di transazioni, con caratteristiche ACID per gli acquisti.
- **Interfaccia intuitiva:** l'interfaccia di TechShop dovrebbe essere semplificata, per permettere all'utente di velocizzare le proprie operazioni. L'interfaccia utente su basa su una finestra principale, che include una barra di campi per accedere con il proprio account (dipende se si è un venditore o cliente), oppure di registrarsi alla piattaforma.
- **Multilingua:** nel sito è disponibile una funzionalità che in base alle impostazioni di sistema traduce nella propria lingua.
- **Robustezza:** tutte le interazioni tra utente e sistema devono avvenire in modo tale che utente non possa inserire dati non validi.
- **Scalabilità:** in termini di utenti contemporaneamente gestiti dal sistema

## 1.3 Definition, acronyms and abbreviations

**SDD: System Design Document.**

**ODD:** Object Design Document. **RAD:** Requirement Analysis Document.

**DB: Database**

## 1.4 References

[PS\\_TechShop.docx](#)

[RAD\\_TechShop.docx](#)

<http://java.sun.com>, <https://jquery.com/> , <https://www.javascript.com/>

[TechShopDB.sql](#)

## 2. Current software architecture

Il progetto TechShop è un progetto di ingegneria re-engineering .

La piattaforma esiste, troviamo un sistema attuale da sostituire, che deve essere riprogettato per uso in un nuovo ambiente, viene modificata l'interfaccia, ed aggiunte nuove funzionalità di supporto per gli utenti.

Nel mercato ci sono delle piattaforme che offrono un servizio simile, sono state inserite nel documento Prepared Statement di TechShop.

### 3. Proposed software

#### 3.1 Subsystem Decomposition

Durante la decomposizione in sottosistemi, dividiamo in sottosistemi più piccoli con una forte coerenza e basso accoppiamento.

#### 3.2 Hardware/software mapping

Durante la decomposizione del sistema di TechShop, dividiamo il sistema in sottosistemi più piccoli. Il sistema viene diviso in tre parti (Layer):

**-UI:** il sottosistema dell'interfaccia utente è diviso per utenti cioè cliente, venditore e gestore Accounting.

L'interfaccia cliente include diversi boundary: di registrazione, di login, di negozi, di categoria, di prodotti, carrello e infine ordine. Nel boundary prodotto è possibile trovare pulsanti per aggiungere un prodotto al carrello e scegliere una quantità. Mentre nel boundary del carrello è possibile trovare pulsanti per modificare una quantità, eliminare un prodotto o infine effettuare l'ordine.

Nell'interfaccia Venditore include diversi boundary: di registrazione, login, categoria, prodotto. Nei vari boundary prodotto e categoria è possibile trovare pulsanti per modificare o eliminare un cliente oppure un venditore.

**-Logic Application:** il sottosistema di logic application è costituito da vari package:

Manager Carrello: contiene la classe carrello che si occupa di effettuare operazione di aggiungere, rimozione, modifica di un prodotto al carrello.

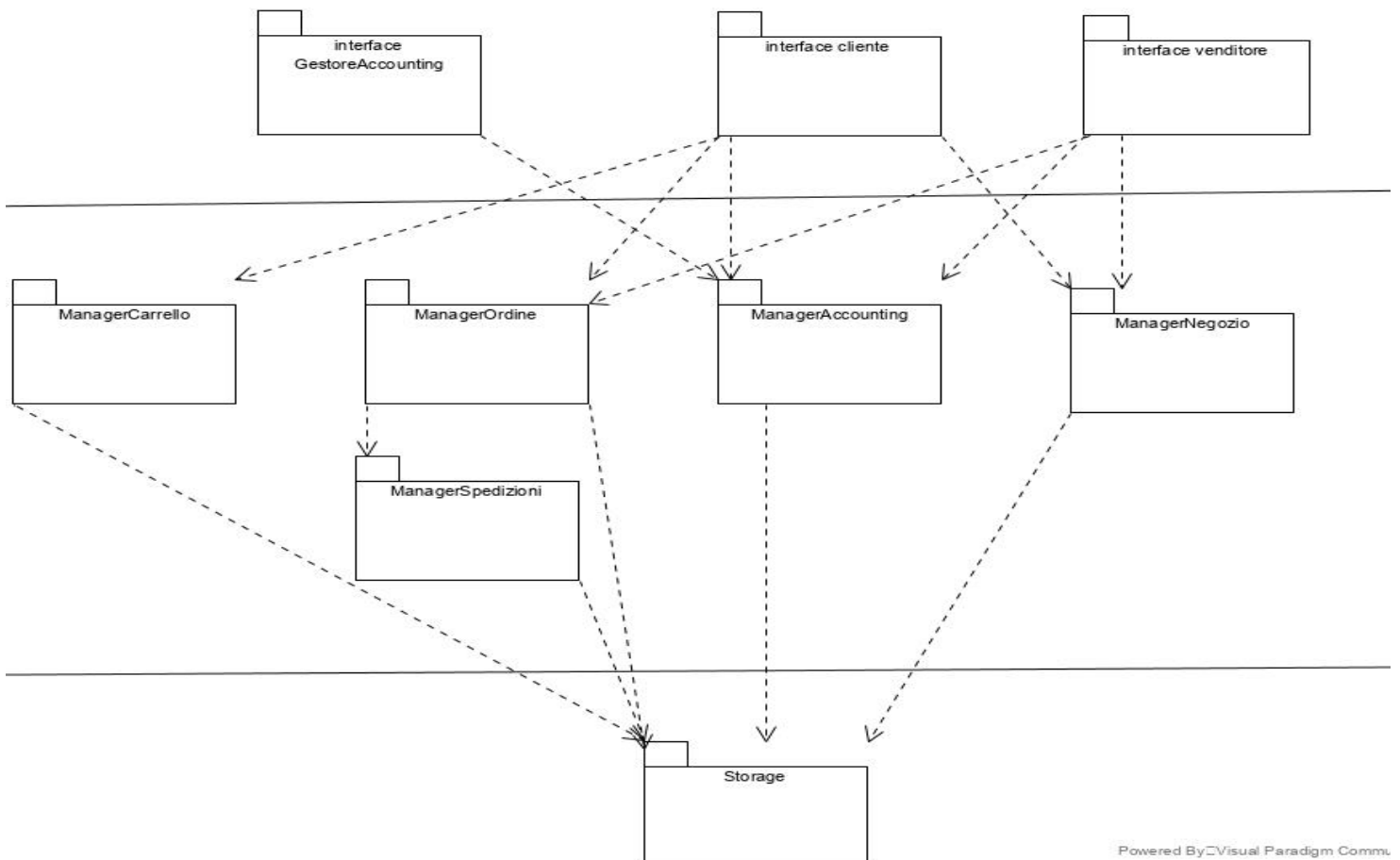
Manager Accounting: contiene la classe cliente, venditore e gestore Accounting che si occupa di recuperare, aggiungere, modificare, controllare, e verificare gli account iscritti alla piattaforma.

Manager Negozio: contiene la classe negozio, categoria e prodotto che si occupa di offrire le operazioni necessarie al venditore (creare negozio, aggiungere prodotti, aggiungi categoria ecc.)

Manager Ordine: contiene la classe fattura che si occupa di offrire un insieme di operazioni in base allo stato dell'ordine (confermato, pronto alla spedizione, annullato)

Manager Spedizioni: contiene la classe gestore delle spedizioni si occupa di offrire operazione quando l'ordine è in transito (spedire, tracciare il pacco)

**-Data Management:** il sottosistema di data Management è costituito da un DBMS capace di memorizzare i dati persistenti e offrire le operazioni di selezione, rimozione, modifica, inserimento e aggiornamento.

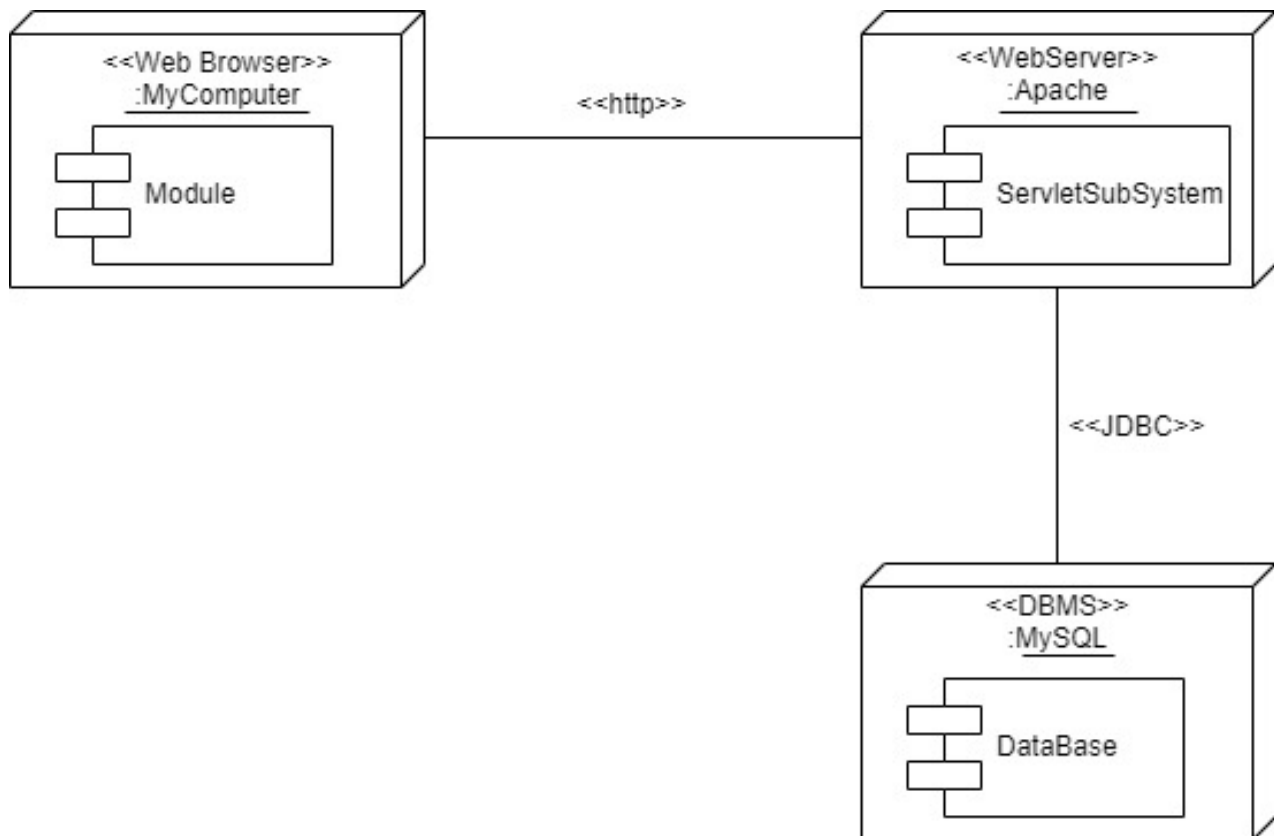


Powered By Visual Paradigm Commu

TechShop è un sito web e ha bisogno di comunicare con altri computer, ma l'utente non avrà percezione di interagire con questi sistemi. Innanzitutto su un computer sarà istanziato un Web Server che contiene tutta la logica applicativa della piattaforma (Servlet), classi Java che si trovano in un container che ricevono, elaborano, e rispondono alla richieste da un web Browser generando una pagina JSP.

Il Web Server Apache presenta alcune funzionalità di default come autenticazione, Virtual Hosting (più siti Web sullo stesso server), multithreading, infine possibilità di fare Url re writing, aliasing, e redirecting.

Su un altro computer ci sarà un DBMS (relazionale) che riceve richieste da parte del Web Server utilizzando il driver JDBC per comunicare le informazioni e i dati da prelevare o inserire in modo persistenti.



### 3.3 Persistent data management

I dati devono essere distribuiti, garantendo che la distribuzione sia trasparente, cioè gli utenti siano in grado di interagire con il sistema come se fosse un unico sistema logico. Ma anche dal punto di vista delle prestazioni per quanto riguarda i metodi di accesso. Inoltre le transazioni devono essere trasparenti, ogni transazione deve garantire integrità del database in tutti gli altri database.

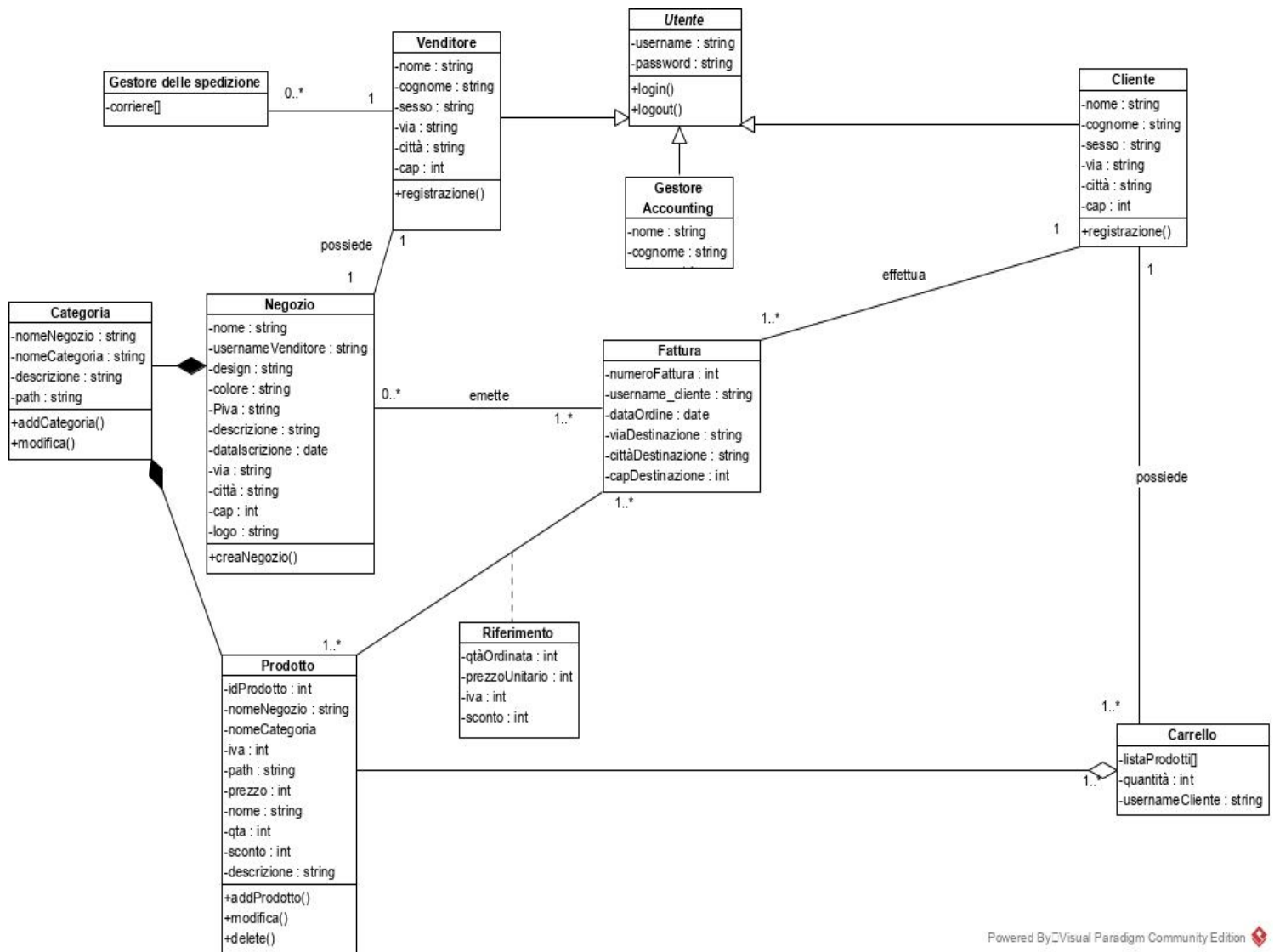
Non sono previste estendibilità del database in immediato futuro.

La stima di frequenza di accesso al database è costante ad ogni operazione da parte di un utente, dal fatto che il sistema sviluppato è basato su letture, scritture e modifiche nel database.

Le dimensioni delle tipiche richieste effettuate sono di estrapolare in media una tupla. Nel peggiore dei casi, la query che preleva tutti i prodotti relativi ad un determinato negozio. La percentuale di questa richiesta è molto ridotta, per mezzo della piattaforma il venditore sarà motivato a suddividere i prodotti del negozio in categorie specifiche. Questo facilita anche da parte del cliente a vedere in dettaglio i prodotti in un tempo ristretto.

Identificazione di oggetti persistenti, TechShop si occupa di una serie di oggetti da essere memorizzati.

Dal Class Diagram sviluppato nel RAD, tutte le classi individuate devono essere rese persistenti.



La posizione del database è nascosta per mezzo della trasparenza di posizione. Usando driver JDBC per la connessione ad un DBMS relazionale, che è un API indipendente dal database, per memorizzare oggetti persistenti in un database relazionale.

La scelta è ricaduta su un DBMS relazionale perché è possibile gestire infiniti insiemi di dati nelle tabelle senza dover tornare indietro e immettere nuovamente i dati una seconda volta, possibilità di creare record separati per ogni tipo di dato da archiviare. Un altro vantaggio è dato dalla flessibilità, una volta che l'utente immette dati in una tabella e li salva nel database, grazie ad una query questi dati potranno essere recuperati in qualsiasi momento.

A discapito di questi vantaggi il database relazionale può richiedere più progettazione e manutenzione. Però opposto sarebbe un database di file flat, ovvero una tabella gigante di singoli record (problema di prestazioni per aggiungere e modificare bisogna immettere nuovamente tutte le informazioni in un nuovo record). Uno svantaggio

concreto che potrebbe portare a scegliere un database orientato agli oggetti, il fatto che gli sviluppatori per un database relazionale dovrebbero ricreare la logica applicativa e il loro sistema.

È anche vero che database relazionali possono essere integrati con quasi tutti i linguaggi di programmazione, molto valido e potente se utilizzato su un web server apache.

Il linguaggio Structured Query Language SQL, linguaggio utilizzato nel nostro software per accedere al DBMS relazionale. Stato scelto per la sua alta compatibilità con questi database, dato che si sono sviluppati insieme.

### 3.4 Access control and security

TechShop è una piattaforma online distribuita su un server, è un sistema multiutente, a diversi attori è consentito visualizzare diversi set di oggetti e invocare diversi tipi di operazioni su di essi. Per documentare i diritti di accesso, disegniamo un matrice controllo d'accessi, che illustra le operazioni consentiti sugli oggetti entità per ciascun attore. In breve la piattaforma può creare specializzazioni di Utenti: Venditore, Cliente, Gestore Accounting e Gestore Spedizione.

I venditori possono creare, aggiungere e modificare categorie e prodotti, visualizzare le fatture, relativo al proprio negozio.

I clienti visualizzano negozi, e i relative categorie e prodotti, possono aggiungere prodotti.

La matrice controlli degli accessi, tuttavia presenta una vista più dettagliata e compatta, consentendo così ad uno sviluppatore di rivedere più schematico gli accessi, per implementarli correttamente.

Lo schema di autenticazione offerto è dato da tipo di utente (venditore, cliente) e dalla coppia nome utente e password, che servono ad identificare l'utente

#### Access Matrix

	<b>Categoria</b>	<b>Negozio</b>	<b>Prodotto</b>	<b>Fattura</b>	<b>Carrello</b>
<b>Gestore Accounting</b>	view()	modifica() view()	view()		
<b>Venditore</b>	addCategoria modifica() view()	creaNegozio modifica()	addProdotto modifica() delete()	view()	
<b>Cliente</b>	view()	view()	view()	view()	view()



			addToCart()		modifica acquista
<b>Gestore Spedizioni</b>				get()	

Quando un'utente effettua il login, tramite i campi username e password inseriti nell'area della piattaforma relativa a quel tipo di utente (venditore, cliente o gestore accounting). Il sistema valida i dati e lo manda nell'home page relativa all'utente.

### 3.5 Global software control

Durante la fase di progettazione teniamo conto del fatto che non tutti gli oggetti hanno il lusso di essere in esecuzione. Esistono tre possibili meccanismi di flusso di controllo: guidato alla procedura, paradigma del flusso di controllo basato su eventi o thread. Nelle sezioni **3.3 Persistent data management** e **3.54 Access control and security**, durante la selezione dei componenti per l'interfaccia e l'archiviazione dei sottosistemi di TechShop, abbiamo effettivamente limitato le alternative per i meccanismi di controllo del flusso la parte dell'organizzazione della piattaforma.

**Il Web Server attende le richieste dal Web Browser.** Con la ricevuta di una richiesta, il Web Server la elabora e la invia all'appropriata servlet o JSP, con conseguente flusso di controllo guidato alla procedura.

Il Web Server alloca un nuovo thread per ogni richiesta, consentendo la gestione parallela delle richieste. Ciò si traduce in un sistema più reattivo consentendo al Web Server di rispondere alle singole richieste Web Browser prima di altre richieste sono stati completamente elaborati e possono aumentare il **throughput** abilitando elaborazione di una richiesta mentre un altro è in attesa che il database risponda.

Lo **svantaggio dei thread** è la loro maggior complessità del sistema derivante dalla sincronizzazione di thread paralleli. Per garantire un design robusto alla concorrenza, si definisce la seguente strategia per affrontare accessi simultanei ai dati condivisi:

## 3.6 Boundary condition

### 3.6.1 Start-up

Il server deve essere sempre acceso altrimenti i client non possono richiedere funzionalità.

Per il primo start-up del sistema "TechShop" è necessario l'avvio di un web server che fornisca il servizio di un Database MySQL per la gestione dei dati persistenti e l'interpretazione ed esecuzione del codice lato server. In seguito, tramite l'interfaccia di Login, sarà possibile autenticarsi tramite opportune credenziali (username e password) che permettono di identificare utente. Una volta effettuato l'accesso, presenterà all'utente la home, dal quale si possono effettuare tutte le operazioni che il sistema fornisce.

### 3.7.2 Start-up (a seguito di un fallimento)

Il sistema può subire guasti dovuti a un elevato numero di accessi contemporaneamente, al sovraccarico del database o a errori di connessione con successivo fallimento.

Nel caso di un errore dovuto alla connessione, il sistema non mantiene nessun dato relativo all'utente. Alla ripresa del funzionamento il sistema riporta l'utente sulla pagina di login per effettuare nuovamente l'accesso alla piattaforma. Nel caso in cui l'utente stesse svolgendo delle operazioni dovrà ripetere l'intera procedura, in quanto non verrà tenuta traccia delle attività non concluse.

### 3.7.3 Terminazione

Ogni client può terminare l'applicazione chiudendo l'applicativo e questo causa la terminazione del sistema con un regolare Logout dal sistema. Al momento del logout tutte le operazioni in corso vengono annullate garantendo così la consistenza dei dati all'interno del sistema.

### 3.7.4 Fallimento

Possono verificarsi diversi casi di fallimento del sistema:

1. Nel caso in cui si verifichi un'interruzione inaspettata dell'alimentazione, al riavvio non sono previsti metodi che ripristinino lo stato del sistema a prima dello spegnimento inaspettato. Tutte le attività non concluse dovranno essere ripetute.
2. Un caso di fallimento potrebbe derivare dal software stesso che causa una chiusura inaspettata dovuta ad errori commessi durante la fase di implementazione, non sono previste politiche correttive.
3. Un altro caso di fallimento potrebbe essere dovuto ad un errore critico nell'hardware, non è prevista alcuna misura correttiva.

## 4. Subsystem services

-**Interface GestoreAccounting:** Interfacce grafiche e oggetti che svolgono il dispatcher verso il ManagerAccounting con cui utente gestore accounting interagirà con il sistema.

-**Interfacce Cliente:** Interfacce grafiche e oggetti che svolgono il dispatcher verso i sottosistemi Manager Carrello, Ordine, Accounting, Negozio.

-**Interfacce venditore:** interfacce grafiche e oggetti che svolgono il dispatcher verso i sottosistemi Manager Ordine, Accounting, Negozio.

-**Manager Accounting:** sottosistema che permette ai vari utenti del sistema di potersi registrare, accedere con le proprie credenziali. Inoltre per utente Gestore Accounting ha la visione completa di tutti gli utenti, abbiamo le funzionalità di cancellazione degli utenti. Si interfaccia con il Database per effettuare le proprie operazioni.

-**Manager Negozio:** sottosistema che permette all'utente cliente di visualizzare i vari negozi, con categorie e prodotti. Invece il venditore, ha la possibilità di creare un negozio, aggiungere categorie e prodotti, modificare le categorie e i prodotti, cancellazione di categorie e prodotti.

-**Manager Ordine:** sottosistema permette all'utente loggato come cliente di completare acquisto confermando ordine e restituendo una fattura. Mentre per il venditore permette di visualizzare tutte le fatture del proprio negozio. Con la facoltà di decidere se spedire ordine privatamente o con una delle ditte delegate con la piattaforma TechShop.

-**Manager Carrello:** sottosistema che permette ad un cliente di gestire il proprio carrello, con operazione di aggiunta di prodotti, scelta delle quantità dei prodotti, scrivere una nota al venditore alla conferma dell'ordine.

-**Manager Spedizioni:** sottosistema che gestisce le spedizioni dei vari ordini, dove il venditore decide a chi affidare la spedizione con dettagli di tempi di consegna. Inoltre fornisce al cliente tracking della propria spedizione.