



SDD

System Design Document

Versione	1.0
Data	19/12/2018
Destinatario	Prof. Andrea De Lucia
	Giovanni Vassalluzzo Luca Giaffreda Alfonso Luciani
Presentato da	Roberto De Luca



Revision History

Data	Versione	Descrizione	Autori
19/12/2018	1.0	Prima stesura	Giovanni Vassalluzzo
18/01/2019	1.0.1	Modifica dei Component	Luca Giaffreda
19/01/2019	1.0.2	Modifica schema relazionale	Roberto De Luca
20/01/2019	1.0.3	Modifica Design Goal e inserimento di component	Alfonso Luciani



Sommario

Revision History	
1. Introduzione	
1.1 Obiettivo del sistema	
1.2 Design Goal	
1.3 Definizioni, acronimi e abbreviazioni	
1.4 Riferimenti	
1.5 Overview	
2. Architettura del sistema corrente	
3. Architettura del sistema proposto	
3.1 Overview	
3.2 Decomposizione in sottosistemi	
3.3 Mapping hardware/software	
3.4 Gestione dati persistenti	
3.5 Controlli accesso e sicurezza	
3.6 Controllo flusso globale del sistema	
3.7 Condizioni limite	
4. Glossario	

1. Introduzione

1.1 Obiettivo del sistema

Si vuole modellare un E-Commerce per la gestione di un negozio di abbigliamento per la vendita dei prodotti online. Il sito si occupa della distribuzione via internet di capi d'abbigliamento maschili e femminili, cercando di far trovare all'acquirente quello che sta cercando in modo veloce e semplice. Il sito, attraverso una base di dati, sarà in grado di gestire gli acquisti in modo da facilitare il lavoro dei dipendenti, in particolare il sito sarà in grado di: affidare l'acquisto a un corriere e comunicare all'acquirente un numero di tracking, comunicare al magazziniere uno o più prodotti che stanno per terminare, l'aggiunta di offerte o aumenti di prezzi da parte di un gestore marketing, far controllare al proprietario statistiche sui prodotti venduti e gestire la parte contabile del negozio

Il sito è rivolto alla massa, in particolare a persone di età compresa tra i 18 e 30 anni.

Il nostro obiettivo è però quello di aumentare questo range di età e prendere la fiducia di chi (in particolare over 30) vive con la paura e con falsi miti dello shop online.

1.2 Design Goal

PRIORITA'	DESIGN GOAL	DESCRIZIONE
Alta	Sicurezza	Il sistema deve garantire un elevato livello di sicurezza utilizzando un protocollo sicuro e criptando le password poiché esso ha il compito di gestire e memorizzare i dati sensibili di una grande quantità di utenti. Inoltre, siccome il sistema si deve occupare di tenere traccia di una grande quantità di beni materiali memorizzandoli all'interno di un database relazionale, esso deve essere invulnerabile ad attacchi di tipo SQL Injection. Tali obiettivi sono indispensabili per perseguire l'obiettivo principale dell'e-commerce, ovvero ricavare profitto.
Media	Bassi costi di sviluppo e mantenimento	Per ottimizzare il più possibile i profitti lo sviluppo e il mantenimento del sistema devono richiedere un costo abbastanza contenuto. Ciò si ottiene selezionando componenti open-source o gratuiti e dimensionando in modo appropriato l'hardware del sistema.
Alta	Estendibilità	Il sistema deve gestire un e-commerce di piccole dimensioni ma in forte crescita, quindi non è da escludere la possibilità di aggiungere nuove funzionalità in futuro o di modificarne quelle esistenti.
Media	Disponibilità	Il sistema deve garantire il funzionamento dalle prime ore della mattina fino a tarda sera con una particolare attenzione verso gli orari in cui i potenziali acquirenti passano maggior tempo sul web.



1.4 Riferimenti

Il presente SDD fa riferimento al progetto Dress Shop; per maggiori dettagli sui requisiti ai quali si fa riferimento, o per una panoramica più ad alto livello di questo progetto, consultare il RAD. Per orientarsi nella stesura dell'SDD, il team si è servito sia del libro di testo consigliato durante il corso (Bruegge) sia delle slides e del materiale didattico fornito dal docente sulla piattaforma e-learning (M-3 System Design), sia dell'aiuto dei tutor messi a disposizione, ma soprattutto delle lezioni e delle spiegazioni del docente.

1.5 Organizzazione del contenuto

Il presente documento illustra la struttura richiesta per la piattaforma Dress Shop. Si definiscono i dettagli dell'architettura del sistema proposto: si tratterà della decomposizione in sottosistemi, delle componenti hardware/software off-the-shelf che verranno utilizzate ai fini della progettazione e di come verranno gestiti i dati persistenti.

La sezione riguardante l'architettura del sistema proposto si conclude con i dettagli su sicurezza e gestione degli accessi e sulla gestione delle condizioni limite. Il documento prosegue con una sezione che illustra i dettagli della decomposizione in sottosistemi spiegando, in modo più dettagliato, i servizi che ogni sottosistema offre agli altri sottosistemi. Il documento si conclude infine con un glossario

2. Architettura del sistema corrente

Non sono presenti sistemi precedenti a questo.

Tuttavia, trovare architetture simili per un e-commerce è davvero semplice, sul web girano ormai migliaia di siti riguardanti lo shop-online(ASOS, Zalando, ecc.) il nostro intento è quello di rendere il sito molto semplice all'uso dell'utente e arrivare il prima possibile a DARE quello che il cliente VUOLE e che sta cercando.

3. Architettura del sistema proposto

3.1 Sintesi della sezione

Per realizzare il sistema, proponiamo un'applicazione web: tipologia software che, a nostro parere, calza perfettamente con l'idea proposta. L'obiettivo che la piattaforma si pone, come già ampiamente discusso nell'introduzione del presente documento e nei documenti precedenti, è la creazione di un E-Commerce per la gestione di un negozio di abbigliamento per la vendita dei prodotti online.

L'architettura che verrà utilizzata nell'implementazione della piattaforma è la diffusissima architettura Client-Server: un server gestirà tutto ciò che riguarda i dati persistenti e la loro accessibilità da parte degli utenti; mentre la controparte client avrà il compito di presentare i vari contenuti in modo semplice ed efficace agli utenti finali.

Il sistema è diviso in tre layer: Storage (gestione dati persistenti), Presentation (gestione interfaccia grafica), Controller (gestione logica di sistema): i tre layer sono organizzati secondo il noto pattern “model – view – controller”. Ad un livello di dettaglio maggiore, invece, è possibile identificare ben 11 sottosistemi, ognuno dei quali si occupa di una funzione specifica e ben definita, in modo da favorire una alta coesione ed un basso accoppiamento.

3.2 Decomposizione in sottosistemi

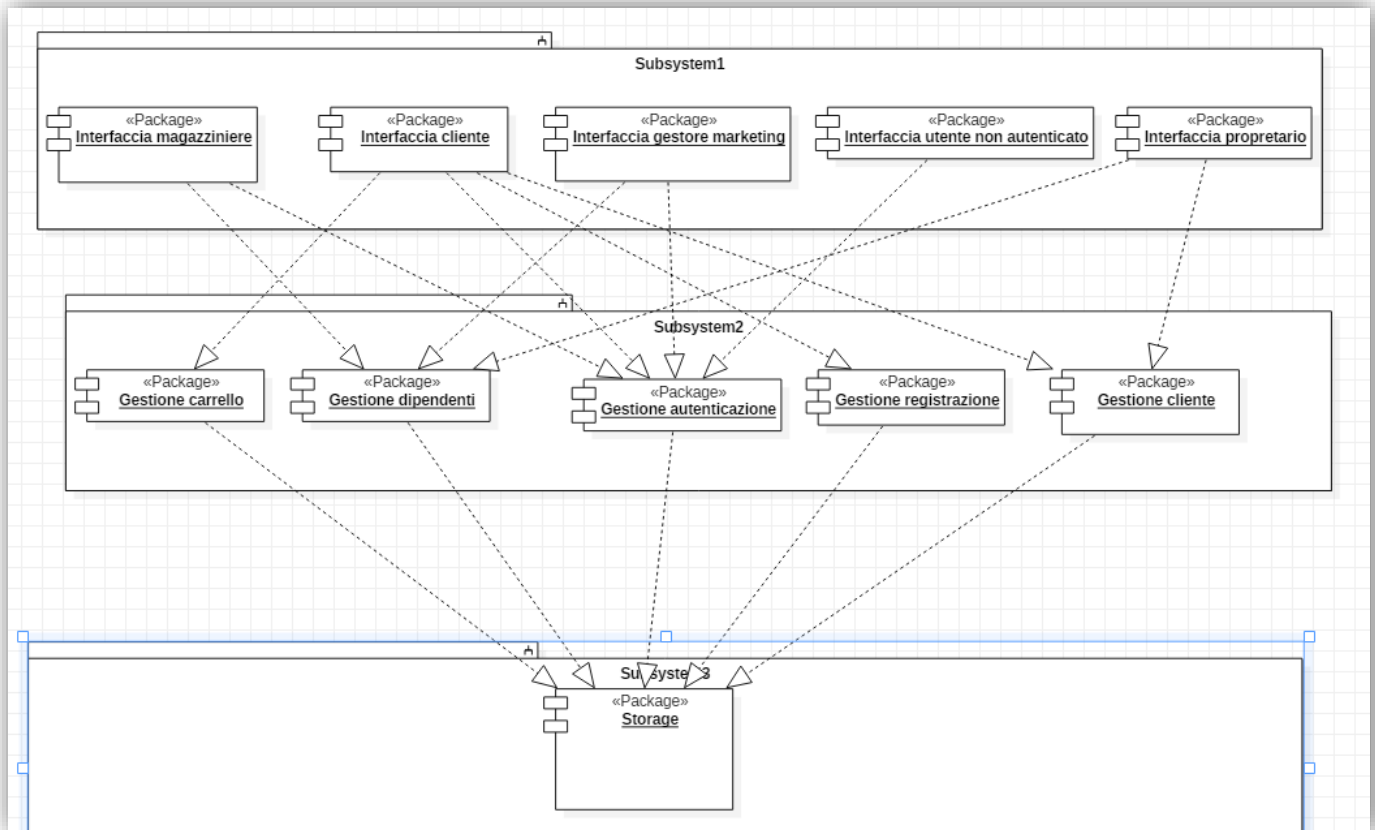
3.2.1 Decomposizione in layer

La decomposizione prevista per il sistema è composta da tre layer che si occupano di gestirne aspetti e funzionalità differenti:

- Presentation (View): raccoglie e gestisce l'interfaccia grafica e gli eventi generati dall'utente;
- Controller (Controller): si occupa della gestione della logica del sistema, accettando l'input dall'utente e convertendolo in comandi per il modello e/o la vista.
- Storage (Model): si occupa della gestione e dello scambio dei dati tra i sottosistemi, indipendentemente dall'interfaccia utente;



Dopo un'analisi funzionale abbiamo deciso di dividere il sistema nel seguente modo dato che abbiamo bisogno di un'alta coesione (dato che le classi del sottosistema effettuano operazioni simili) e di un accoppiamento basso (perché le varie modifiche ad un sottosistema non ne modifichino gli altri). La suddivisione è stata effettuata secondo la “Decomposizione del Sottosistema in Strati” e cercando di rispettare le sue euristiche. Ci sono tre livelli uno che gestisce la visualizzazione dei dati di un utente, uno che gestisce le operazioni e uno che immagazzina i dati.



Primo livello

- Interfaccia magazzino: Sottosistema che gestisce l'interfaccia di un magazzino, in questo caso saranno disponibili le varie operazioni che può effettuare un magazzino
- Interfaccia cliente: Sottosistema che gestisce l'interfaccia di un cliente, in questo caso il cliente può usufruire di tutte le azioni offerte per un cliente
- Interfaccia utente non autenticato: Sottosistema che gestisce l'interfaccia di un utente non autenticato, in questo caso si suppone che sia un cliente e quindi può fare solo alcune delle operazioni che può fare il cliente finché non effettuerà l'autenticazione.
- Interfaccia gestore marketing: Sottosistema che gestisce l'interfaccia di un gestore marketing, in questo caso si visualizzeranno le varie operazioni che può svolgere il gestore marketing
- Interfaccia proprietario: Sottosistema che gestisce l'interfacci del proprietario del negozio, in questo caso si potranno vedere tutte le operazioni che può fare e le informazioni relative ai dipendenti.



Secondo livello

- Gestione Carrello: Sottosistema che gestisce gli acquisti, permette di svolgere le operazioni di acquisto e le eventuali conseguenze
- Gestione autenticazione: Sottosistema che gestisce le regole di autenticazione degli utenti e dell'accesso alle varie funzionalità
- Gestione autenticazione: Sottosistema che gestisce le regole per la registrazione degli utenti
- Gestione clienti: Sottosistema che gestisce il profilo di un cliente che può cambiare i dati che riguardano lo stesso e le modalità di pagamento
- Gestione dipendenti: Sottosistema che gestisce le informazioni del negozio, dei prodotti e dei dipendenti e permette di modificarle

Il terzo livello è un singolo sottosistema che gestisce ed immagazzina i dati personali

3.3 Mapping hardware/software

La struttura hardware proposta è costituita da un server centrale e dalle postazioni utenti (computer portatili o fissi), dotati di sistemi operativi che possono essere diversi fra loro.

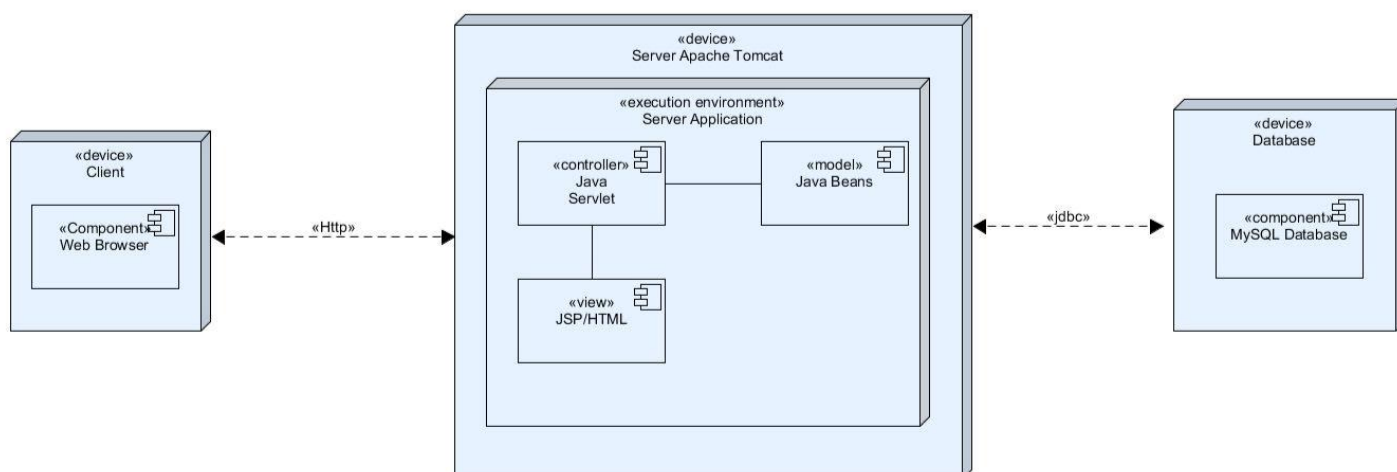
All'application server, in cui risiedono gli oggetti di tipo control ed entity, si collegano le postazioni client e lo storage; da una parte, la comunicazione tra client e server avverrà attraverso il protocollo HTTP, dall'altra, quella tra server e database si servirà di JDBC. Sarà il server stesso a garantire un'adeguata ripartizione delle operazioni consentite ai vari client, distinte in base alla modalità di autenticazione e considerando anche il caso in cui essa non avvenga.

In fase di accesso i dati sottomessi dall'utente saranno, infatti, inviati allo storage allo scopo di verificarne l'esattezza.

Al fine di assicurare una completa visione del sistema a "run time" è proposto un unico deployment diagram, mentre la comprensione della struttura statica a "compile time" potrà avvenire tramite i component diagram.

3.3.1 Deployment diagram

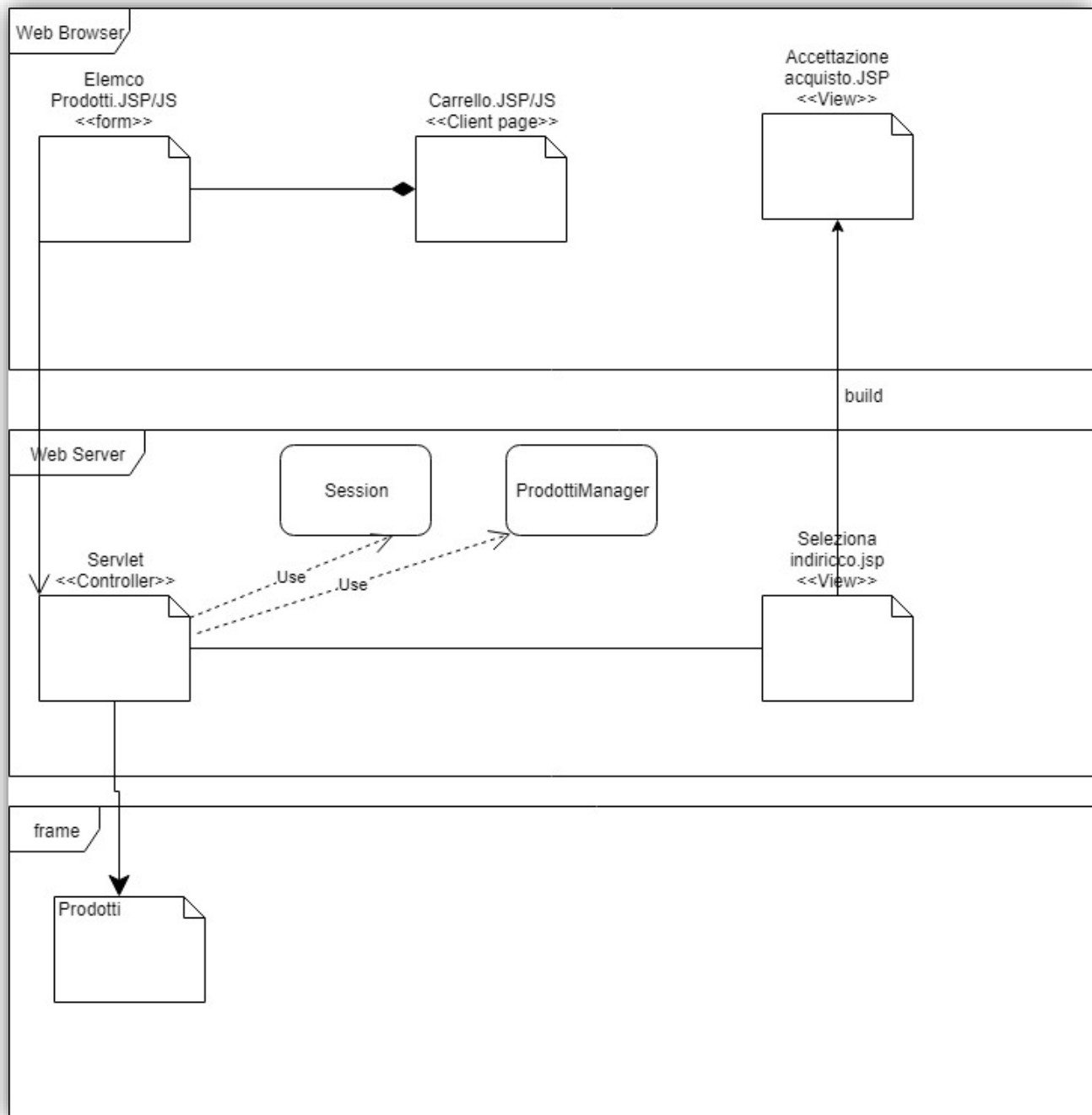
Il Deployment Diagram proposto descrive il sistema in termini di risorse hardware con le relative relazioni. Come mostrato dallo schema sottostante, gli utenti potranno interagire con ETM Platform tramite il browser su qualunque dispositivo semplicemente collegandosi all'indirizzo web della piattaforma. Sarà compito poi del Controller interpretare gli eventi generati dall'utente, richiedere e prelevare le opportune risorse dal database ed inviare le informazioni richieste dagli utenti tramite il protocollo http, il quale si collegherà al Database tramite JDBC.



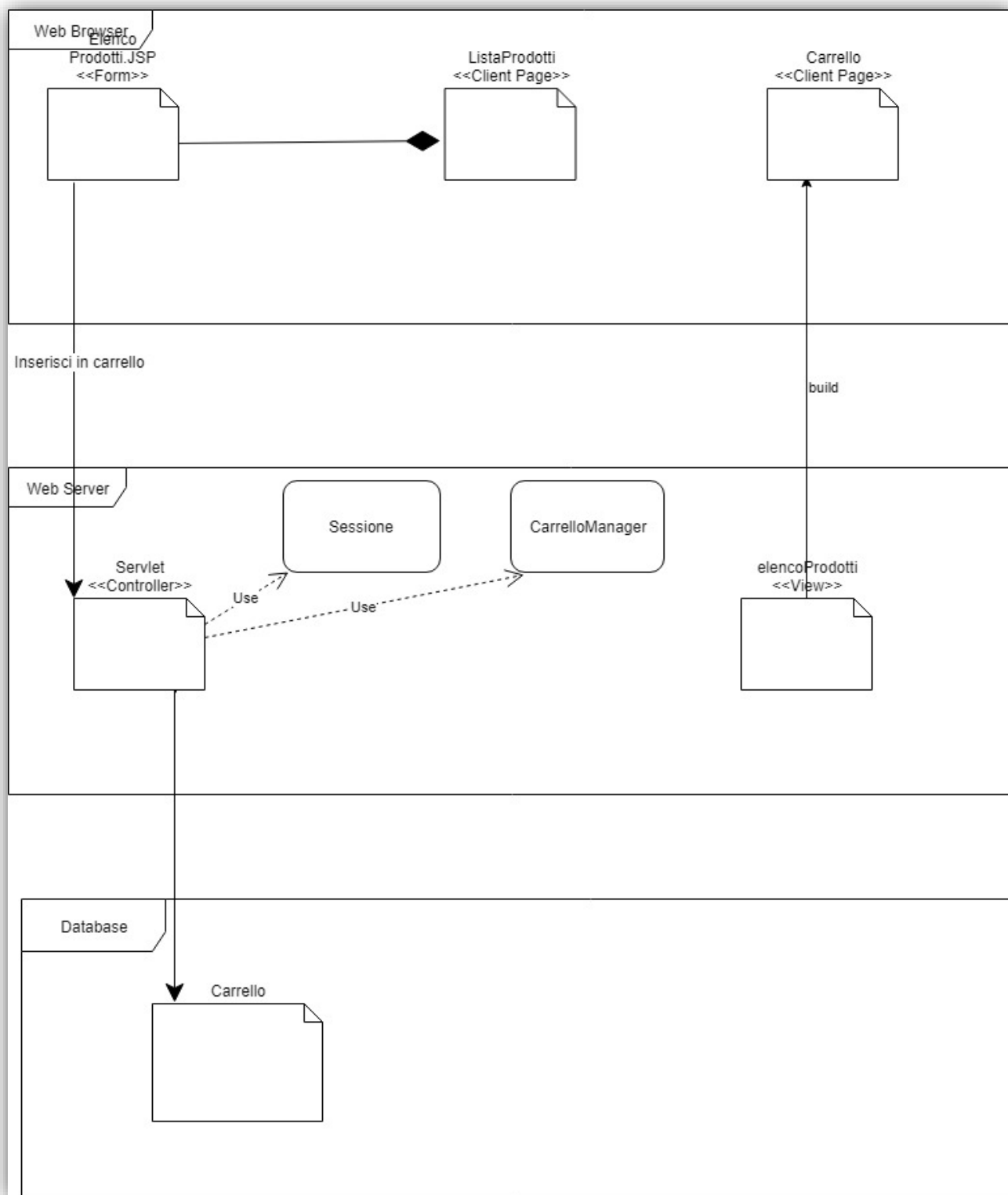
3.3.2 Component diagrams

Di seguito sono, invece, sono elencati i component diagrams, schematizzando la struttura statica del sistema.

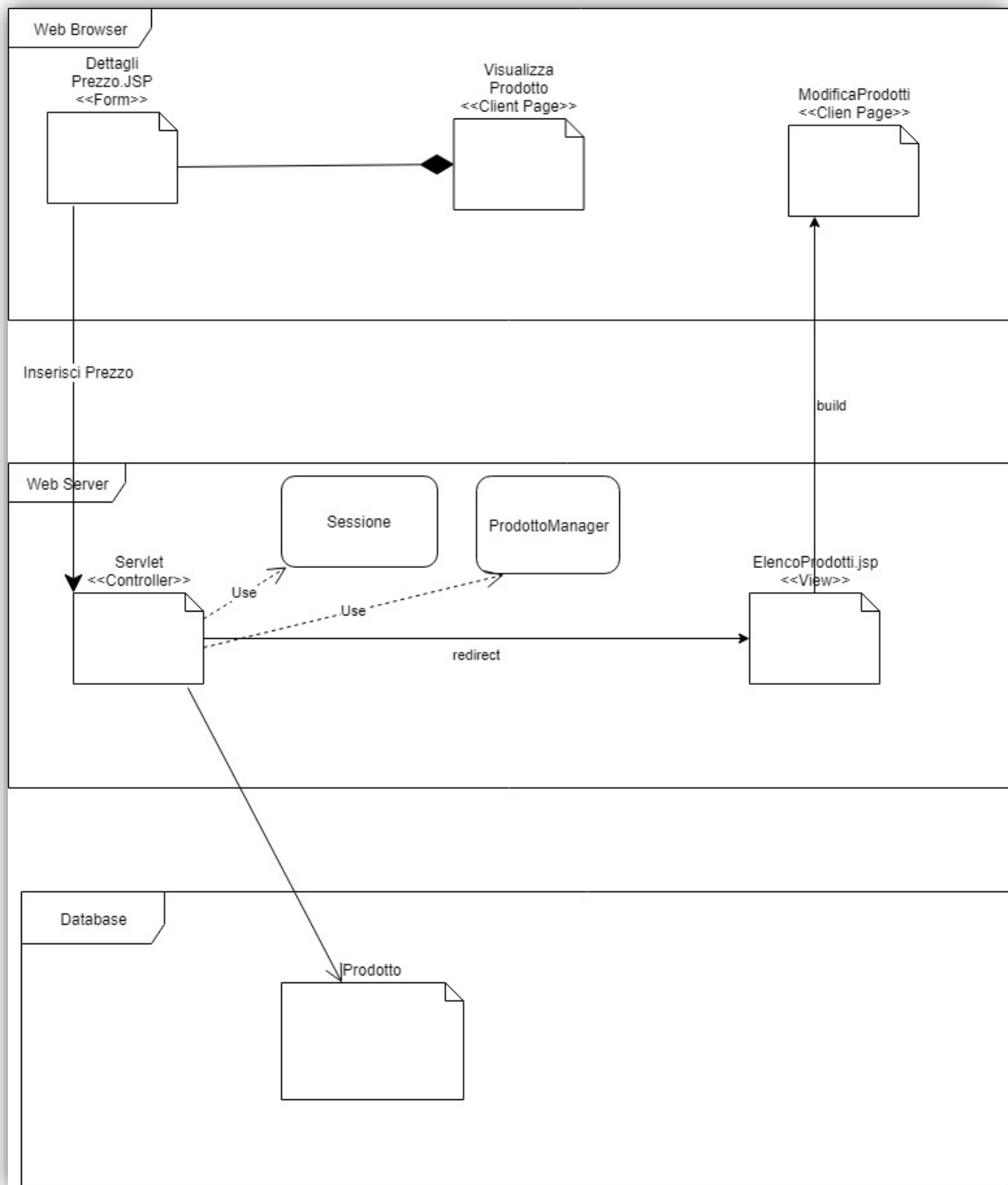
Acquisto:



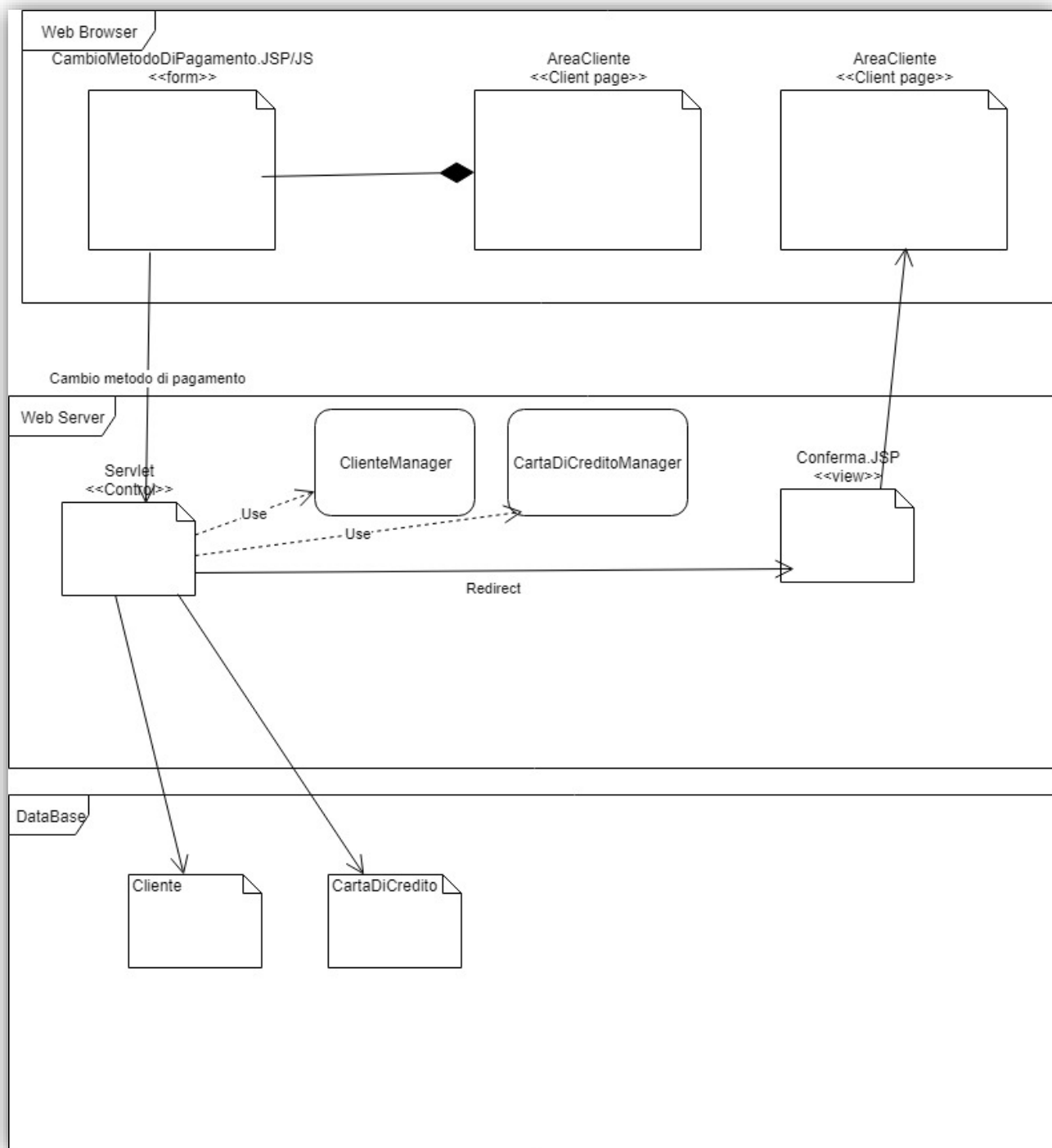
Inserimento in carrello:



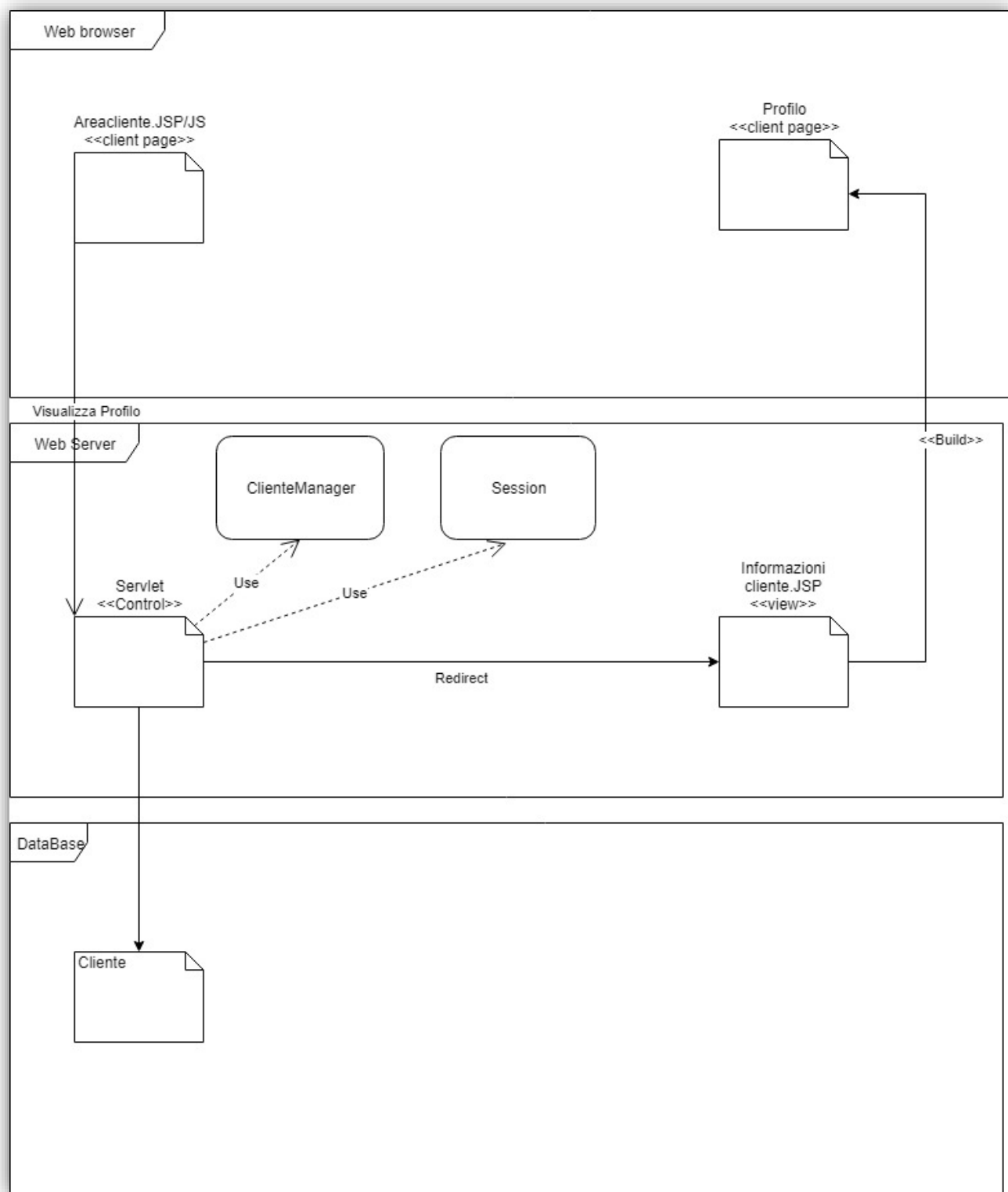
Inserimento prezzo:



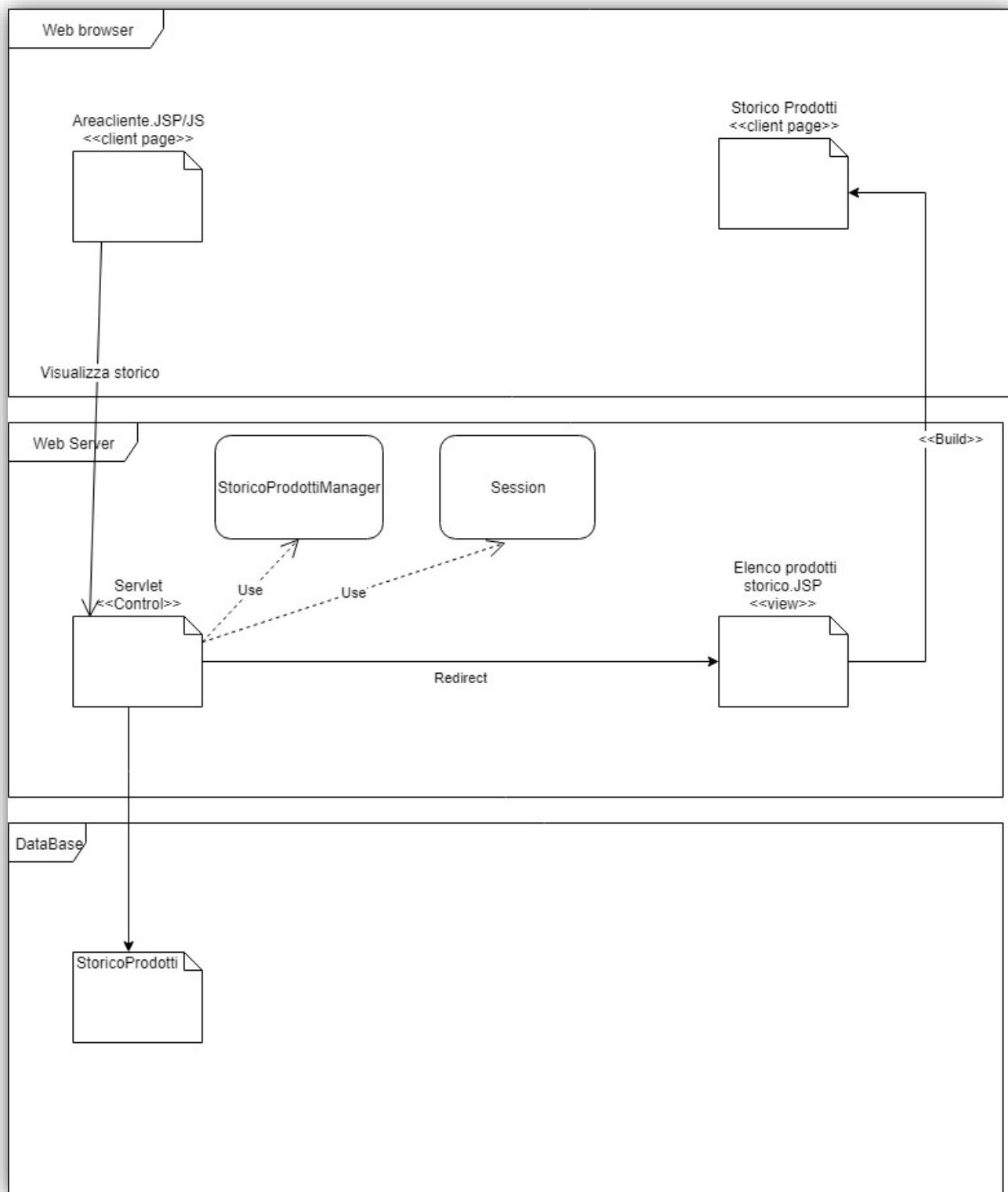
Cambio metodo di pagamento:



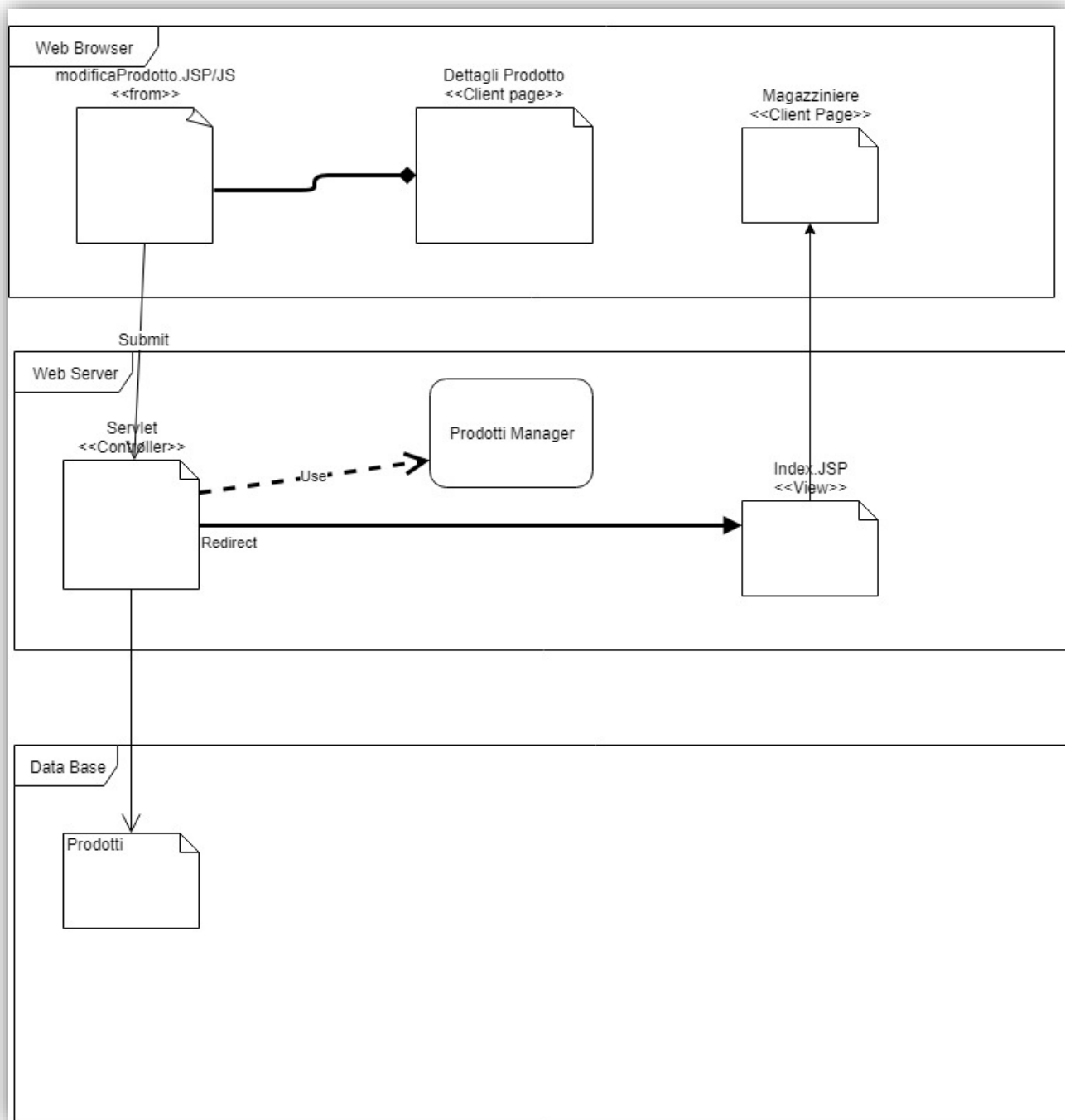
Visualizza profilo:



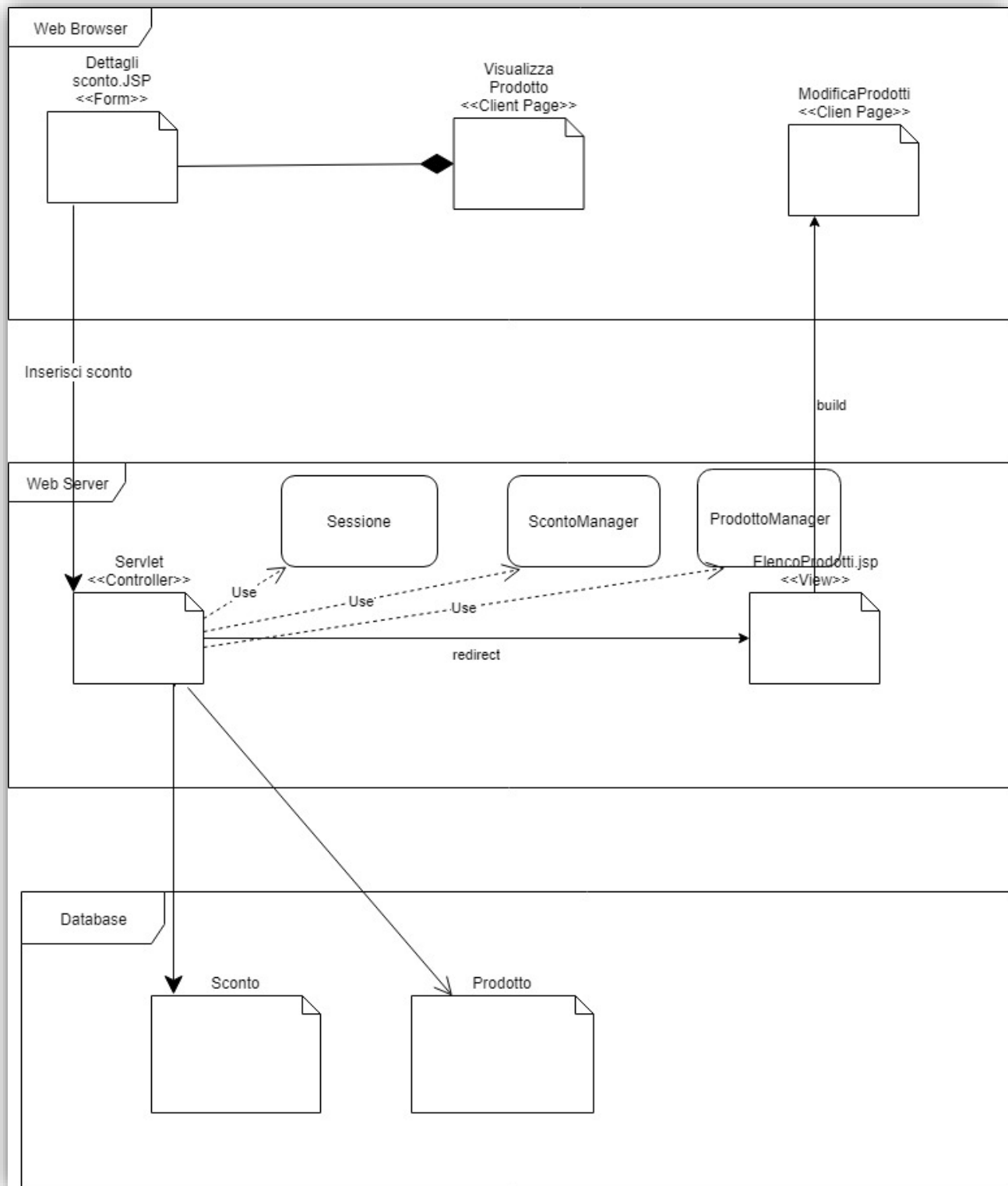
Visualizza storico:



Modifica prodotto:



Inserisci sconto:

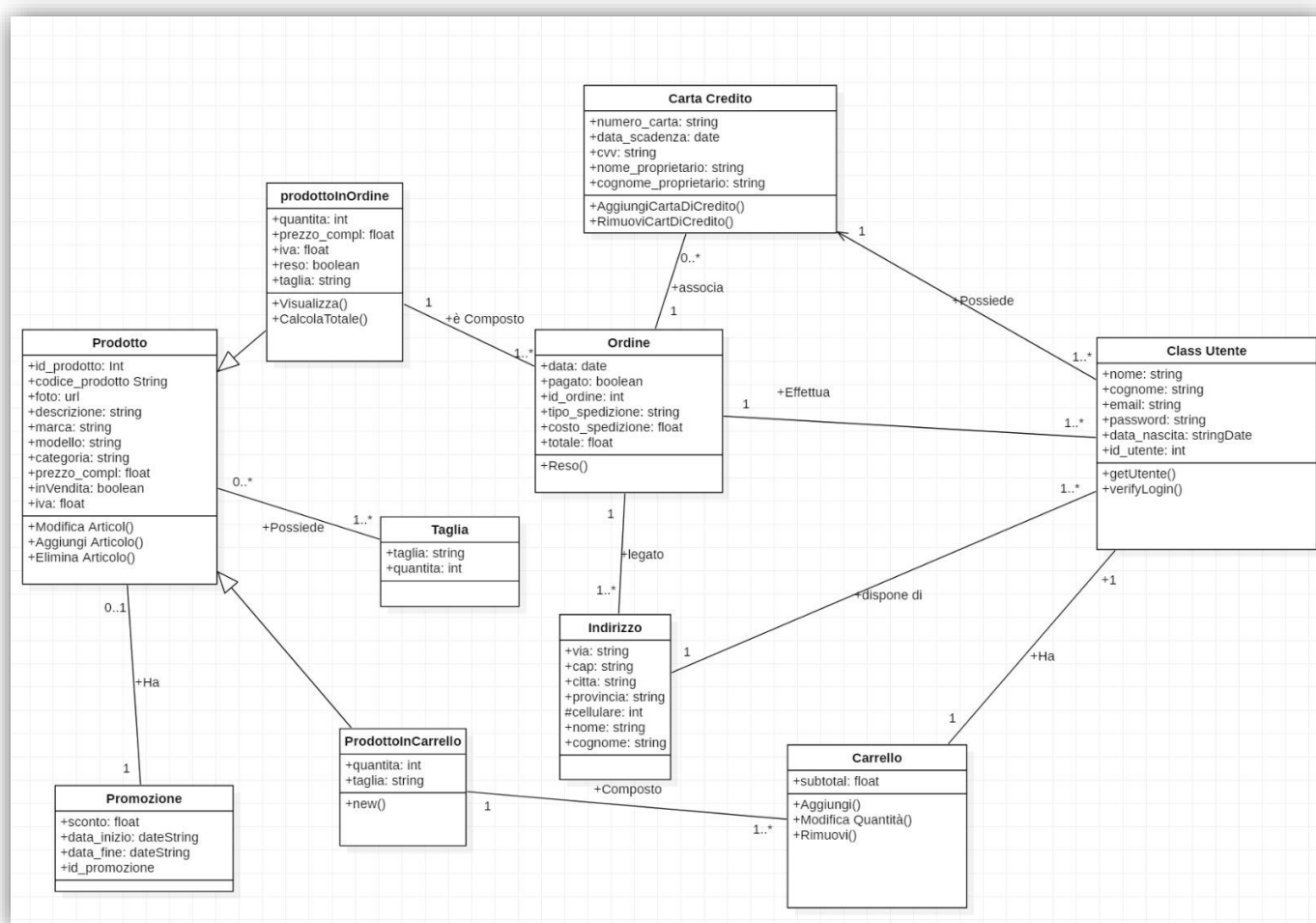


3.4 Gestione dati persistenti

3.4.1 Modello Concettuale

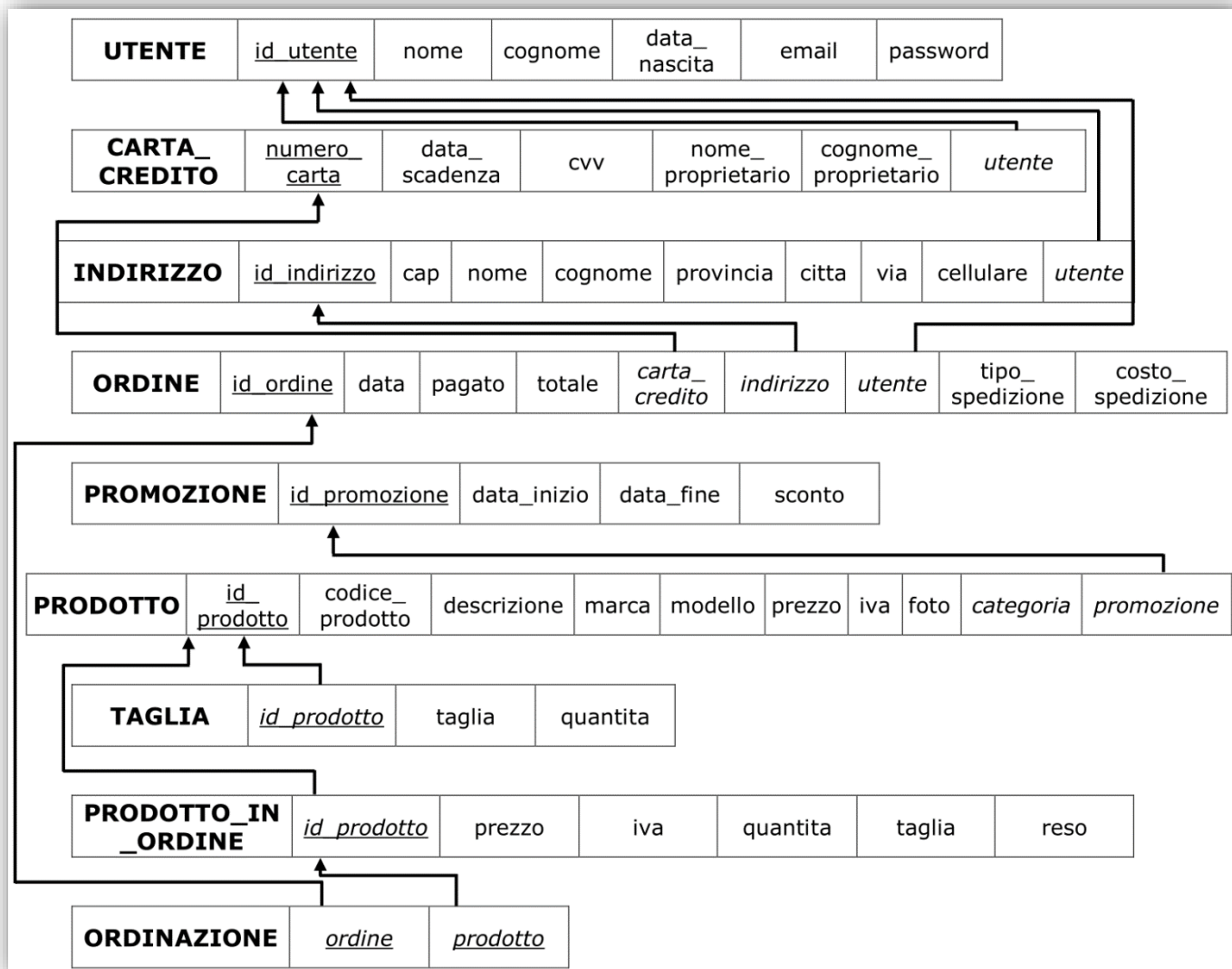
La gestione dei dati persistenti è un fattore chiave nella realizzazione di DressShop E-Commerce: esso avrà infatti un impatto significativo sull'efficienza del sistema, in tutti i suoi aspetti.

Innanzitutto, vista la mole di dati prevista, la gestione dei dati persistenti sarà effettuata tramite un DBMS di tipo relazionale. In una fase iniziale, il team ha sviluppato un primo modello per la gestione dei dati, tenendo conto soltanto delle effettive relazioni logiche tra i dati e trascurando completamente le considerazioni sull'efficienza. Il risultato della prima fase è il seguente class diagram



3.4.2 Schema relazionale

Dopo aver analizzato, invece, i dati da un unto di vista dell'ottimizzazione del sistema rispetto ai design goal prefissati si è giunti al seguente schema relazionale che rappresenta fedelmente l'organizzazione dei dati nel database.



3.5 Controlli accesso e sicurezza

La gestione del controllo dell'accesso prevede principalmente un meccanismo di autenticazione, il login, che prevede l'inserimento di indirizzo e-mail e password per l'autenticazione. Qualora un utente non loggato tenti di accedere ad una zona che richieda l'autenticazione, gli verrà notificato di star tentando di accedere ad un'area riservata a soli utenti registrati e gli verrà richiesto di effettuare il login. Qualora un utente loggato tenti di accedere ad una zona a lui non riservata gli verrà impedito l'accesso e gli verrà notificata l'impossibilità di accedere.

Il tipo Utente: si divide in:

- Utente non registrato: sottotipo di Utente, può inoltre registrarsi, può vedere il catalogo ma senza fare acquisti
- Utente registrato: sottotipo di Utente che può: modificare il proprio profilo, effettuare il login/logout, vedere il catalogo. Tale sottocategoria di Utente si divide in:
 - Cliente: Effettuare un acquisto, effettuare un reso, tracciare la posizione del proprio acquisto, inserire/eliminare prodotti nel carrello, visualizzare il carrello, visualizzare il proprio storico acquisti, modificare le modalità di pagamento (aggiungere o eliminare carte di credito)
 - Magazziniere: può: controllare il numero di scorte di un determinato prodotto, modificare il numero di prodotti nel database,
 - Gestore Marketing: può scegliere un prezzo quando arriva un prodotto, , effettuare sconti promozionali, cambiare prezzo ai prodotti.
 - Proprietario: può visualizzare statistiche di vendite, controllare il guadagno, aggiungere o rimuovere dipendenti, controllare i dati di tutti gli utenti .

<u>OGGETTO</u> <u>ATTORE</u>	Utente	Carta di Credito	Prodotto	Carrello	Ordine	Promozione
Utente non registrato	RegistraAccount()	Ndr	Ndr	Ndr	Ndr	Ndr
Cliente	login() modificaProfiloutente() ,logout()	AggiungiCC() RimuoviCC()	Acquista();	AggiungiAlCrll() ModificaQntCrll() RimuoviDalCrll() VisualizzaCrll()	getStorico();	Ndr
Magazziniere	login() modificaProfiloutente() logout()	Ndr	VediQnt() modificaQnt() aggiungiProdotto()	Ndr	Ndr	Ndr
Gestore Marketing	login() modificaProfiloutente() getUtente(),logout()	Ndr	AggiungiPrezzo() ModificaPrezzo()	Ndr	Ndr	AggiungiPromozione()
Proprietario	login() modificaProfiloutente() logout(), getListaUtenti, eliminaUtente()	Ndr	vStoricoVendite()	Ndr	Ndr	ndr

3.6 Controllo flusso globale del sistema

Il controllo del flusso globale del sistema sarà gestito attraverso la procedura “event-driven”.

3.7 Condizioni limite

3.7.1 Start-up e configurazione

Il sistema lato server parte con l'avvio del database server e successivamente del web server che provvederà alla creazione della connessione con il database.

Completata questa procedura il lato client sarà inizializzato ogni qual volta un utente inserisce l'URL del sito nel browser.

3.7.2 Terminazione

Il sistema lato server, dopo l'avvio, rimarrà sempre attivo (a meno di guasti).

Il sistema lato client termina alla chiusura del browser con il conseguente rilascio delle informazioni temporanee della sessione utente, mentre i dati persistenti saranno mantenuti nel database.

3.7.3 Fallimento

Il fallimento del sistema può avvenire in condizioni eccezionali dovuti a problemi hardware o software. Nel caso in cui il fallimento avvenisse nel web server verrà effettuato un ripristino della configurazione del server e un riavvio, il fallimento non causerà perdita di dati in quanto il database è situato in un server esterno.

Una possibile perdita di dati potrebbe avvenire in caso di guasti alla memoria del database, per questo motivo verranno effettuati backup periodici dei dati. Nel caso in cui il web server non riuscisse a comunicare con il database, gli utenti che ne hanno richiesto l'utilizzo saranno avvisati.

Potrebbero, inoltre, verificarsi fallimenti dovuti a bug nel codice del sistema, questa situazione prevede tre casi:

- Se il fallimento avviene nel client, il sistema continua a funzionare regolarmente;
- Se il fallimento avviene nel web server, il sistema non sarà utilizzabile;
- Se il fallimento avviene nel database server, il sistema sarà utilizzabile solo per eventuali operazioni che non prevedono l'accesso ai dati persistenti. Per tutti le altre operazioni il web server comunicherà il problema agli utenti;



4. Glossario

E-commerce: negozio online

Component Diagram: è un [diagramma](#) che ha lo scopo di rappresentare la struttura interna del [sistema software](#) modellato in termini dei suoi [componenti](#) principali e delle relazioni fra di essi.

DBMS: Database Management System, software progettato per la creazione e la manipolazione di database.

Design Goal: descrive una qualità che i progettisti dovrebbero ottimizzare.

Diagramma ER: diagramma entità-relazione.

Login: procedura attraverso la quale un utente registrato può accedere alla piattaforma. Richiede l'inserimento della e-mail universitaria e di una password.

Logout: procedura attraverso la quale un utente registrato può disconnettersi dalla piattaforma.

ODD: object design document.

RAD: Requirements Analysis Document, documento che tratta nel dettaglio l'analisi dei requisiti.

SDD: System Design Document, documento che tratta nel dettaglio della progettazione del sistema, e dei suoi obiettivi