

S.D.D.

System Design Document

El Tamborito 

Team:

NOME	MATRICOLA
Raffaele Casaburi	0512103796
Giuseppe Ranieri	0512103616
Antonio Piacentini	0512103552

Sommario

Sommario

1. Introduzione	3
1.1 Scopo del sistema	3
1.2 Design goals	3
1.2.1 Performance	3
1.2.2 Affidabilità	4
1.2.3 Manutenzione.....	5
1.2.4 End User.....	6
1.3 Definizioni, acronimi e abbreviazioni	6
1.4 Riferimenti	6
1.5 Panoramica	6
2. Architettura del software corrente	8
3. Architettura del software proposto	8
3.1 Panoramica	8
3.2 Decomposizione del sistema	9
3.2.1 Gestore	11
3.2.2 Utente Registrato	12
3.2.3 Utente non registrato	13
3.3 Mapping Hardware/Software.....	14
3.4 Gestione dei dati persistenti	17
3.5 controllo degli accessi e sicurezza	19
3.6 Controllo software.....	20
3.7 Condizione boundary.....	20
3.7.1 Avvio del sistema.....	20
3.7.2 Terminazione del sistema.....	20
3.7.3 Fallimento del sistema.....	20
4. Servizi dei sottosistemi	24
4.1 Gestione Profilo	24
4.2 Gestione Profili	24
4.3 Gestione prodotti	25
4.4 Gestione Acquisti.....	25
4.5 Gestione Carrello	25
5. Glossario	26

1. Introduzione

1.1 Scopo del sistema

“El Tamborito” è una piattaforma attraverso la quale vengono resi disponibili all’acquisto online strumenti musicali e oggetti relativi al mondo della musica. Nel sito è disponibile una descrizione storica e tecnica dello strumento visualizzato ed insieme ad esse il suo prezzo.

1.2 Design goals

Il sistema fornisce un’interfaccia grafica user friendly, facile ed intuitiva, con bottoni e metodi di navigazione accessibili a chiunque. Di conseguenza le informazioni presenti nella home page possono guidare l’utente in altre pagine anche se quest’ultimo non ha competenze specifiche nell’utilizzo dell’applicazione. Il sistema proposto rispetterà i criteri di design elencati di seguito.

1.2.1 Performance

Tempo di risposta	La piattaforma deve garantire un tempo di risposta minimo sia per quanto riguarda le operazioni che compie il gestore, sia per quanto riguarda le operazioni compiute dall’utente, tuttavia essendo un sistema web la connessione incide su questo criterio.
Throughput	Anche i picchi di carico (parliamo 200 utenti simultaneamente collegati) devono essere gestiti dal sistema senza rallentamenti, garantendo fluidità e una latenza molto bassa. Il sistema, deve garantire consistenza in tutte le operazioni che includono anche chiamate al database, specialmente nei momenti di maggiore carico.
Memoria	Il sistema utilizza un database relazionale per memorizzare tutti i dati il carico di dati non rappresenterà un problema per le performance del sistema.

1.2.2 Affidabilità

Robustezza	Il sistema deve essere robusto, capace di poter mantenere dati anche in caso di guasti di qualsiasi tipo, come guasti elettrici, hardware, attacchi informatici, problemi legati al browser.
Disponibilità	Il sistema dovrà essere disponibile 24/7 grazie ad un server costantemente attivo
Tolleranza all'errore	Il sistema deve essere tollerante agli errori, cioè anche in caso di errore deve proseguire con la sua esecuzione senza che l'errore possa compromettere il normale utilizzo del sistema.
Sicurezza	<p>Gli accessi al sistema avvengono tramite un'operazione di routing che controlla se l'utente che sta operando sul sistema ha i permessi adeguati per usufruire dei servizi forniti (Es. un utente non registrato non può effettuare acquisti).</p> <p>Quindi il fulcro della sicurezza del sistema è il login, il quale permette di riconoscere un utente registrato da uno non registrato e permette, altrettanto, di riconoscere l'amministratore e quindi di farlo accedere per ottenere i servizi appositi offerti per la modalità di gestione. Le password sono codificate in modo da non comparire in nessuno schema interno al sistema.</p>

1.2.3 Manutenzione

Estensibilità	La progettazione del sistema sarà condotta in modo da agevolare la facile introduzione di nuove funzionalità utilizzando django (una moderna tecnologia per lo sviluppo web), CSS3, Bootstrap per sviluppare il template, JQuery.
Modificabilità	Deve essere possibile intervenire sul codice esistente per correggere eventuali difetti o implementare nuove funzionalità. Bisogna garantire che il codice sia leggibile da programmatori che non lo hanno implementato.
Leggibilità	Il codice sarà ben strutturato per semplificare eventuali interventi su di esso.
Tracciabilità dei requisiti	Grazie alla tracciabilità dei requisiti, sarà possibile effettuare le modifiche necessarie al corretto funzionamento del sistema, valutando correttamente i costi e i rischi che le modifiche porteranno.

1.2.4 End User

Usabilità	El Tamborito rende ogni funzione di semplice uso garantendo un'ottima esperienza di navigazione all'utente grazie ad interfacce grafiche user friendly.
------------------	---

1.3 Definizioni, acronimi e abbreviazioni

- El Tamborito: Nome del sistema che verrà sviluppato.
- Gestore: attore del sistema che gestisce il sistema di vendita.
- Utente non registrato: utente non registrato al sistema che può effettuare visualizzazione ricerche.
- Utente Registrato: utente registrato che può effettuare operazioni di acquisto.
- DBMS: Database Managment System.
- Django: framework web usato per l'implementazione

1.4 Riferimenti

- Bern Bruegge, Allen H. Dutoit, Object-Oriented Software Engineering - Using UML, Patterns, and JAVA, 3rd edition.

1.5 Panoramica

Fin ora sono stati introdotti gli obiettivi di design e sono stati forniti i riferimenti ai materiali usati per la composizione del documento. In seguito, nella sezione sistema software corrente, verrà descritto un sistema software simile.

La sezione sistema software proposto documenta il system design del nuovo sistema. In questa sezione vengono descritti i seguenti elementi:

- Decomposizione in sottosistemi: il sistema viene suddiviso in diversi

sottosistemi. Ricordiamo che un sottosistema è formato da un insieme di classi, associazioni, operazioni e vincoli che sono in relazione tra di loro. Ogni sottosistema è caratterizzato dai servizi che offre agli altri sottosistemi.

- Mapping hardware/software: in questa sezione vengono prese decisioni riguardo le piattaforme hardware su cui il sistema dovrà girare, una volta decise le piattaforme è necessario mappare le componenti su di esse.
- Gestione dei dati persistenti: descrive i dati persistenti che vengono memorizzarli dal sistema ed il tipo di infrastruttura usata per memorizzarli.
- Controllo degli Accessi e Sicurezza: descrive il modello degli utenti del sistema in termini di una matrice degli accessi. Questa descrive le informazioni a cui i singoli attori possono accedere e quali sono le operazioni che questi hanno la possibilità di effettuare.
- Controllo Globale del Software: descrive il modo in cui è implementato il controllo globale del software e come si sincronizzano i sottosistemi.
- Condizioni di boundary: in questo paragrafo vengono descritte le condizioni limite del sistema come start-up e shutdown e la gestione dei fallimenti del sistema.

La sezione servizi dei sottosistemi descrive in termini di operazioni quali sono i servizi forniti da ciascun sottosistema.

L'ultima parte del documento è costituita dal glossario che si occupa di elencare una serie di termini e fornire la relativa spiegazione in maniera tale da fornire supporto a coloro che leggono il documento.

2.Architettura del software corrente

Ricercando sul web, in data 10/07/2018, abbiamo trovato qualche sistema simile a quello che stiamo progettando, tra questi troviamo mercatinomusicale.net, musicstore.it i quali sono accumunati da una vastissima gamma di strumenti musicali e di prezzi molto competitivi.

3.Architettura del software proposto

3.1Panoramica

L'architettura del sistema è di tipo client/server di tipo thin-client, tale scelta comporta molti vantaggi tra i quali portabilità, infatti il sistema potrà essere utilizzato su macchine e sistemi operativi differenti, performance, scalabilità, il server sarà in grado di gestire un elevato numero di client connessi allo stesso momento, Flessibilità, per ogni tipologia di utente che effettua l'accesso (gestore compreso) vi sarà un'interfaccia apposita tramite la quale ogni attore potrà eseguire le operazioni ad esso riservate, ed infine, affidabilità, il sistema deve essere affidabile e in grado di mantenere i propri dati anche in seguito a guasti, quindi deve essere possibile effettuare backup periodici.

Tutto lo stack si basa sul design pattern Model-View-Template, una specializzazione del Model-View-Control, che estende i vantaggi del primo con un sistema di templating che offre la possibilità di trattare i template come artefatti di programmazione Object Oriented: esso infatti consente di estendere un template che può essere visto come uno skeleton, limitando al minimo il bisogno di scrivere codice, andando a minimizzare i tempi di sviluppo e favorendo un maggior riutilizzo.

L'utilizzo di tale modello comporta vantaggi anche nella suddivisione del lavoro tra i vari membri del team che si occupano di interventi di manutenzione e implementazione, inoltre, facilita la comprensione del codice a chi non ha inizialmente lavorato con questo sistema.

3.2 Decomposizione del sistema

Per rendere il sistema più facile da progettare e aumentare la manutenzione decomporremo l'intero sistema in una serie di sottosistemi al fine di rendere la progettazione delle funzionalità molto più semplice così come una passibile aggiunta di funzionalità ancora non definite.

Per effettuare questa decomposizione dobbiamo ridurre al minimo la dipendenza fra i sottosistemi, quindi puntare ad un basso coupling, e di conseguenza rendere minima la coesione all'interno dei sottosistemi, ovvero fare in modo che le classi non abbiano una forte dipendenza.

Il sistema è stato diviso secondo un'architettura Three tier: Presentation Layer, composto da interfacce grafiche e in particolare dai boundary object come le form, Application Layer, composto invece dagli oggetti che si occupano della gestione del controllo e di notificare i cambiamenti del presentation layer, Storage Layer si occupa della memorizzazione dei dati persistenti e del loro recupero dal database (uso delle query).

Presentation Layer

-Sottosistema Utente non registrato: include tutte le interfacce grafiche a cui il cliente può accedere per visualizzare informazione relative ai prodotti, e alla registrazione.

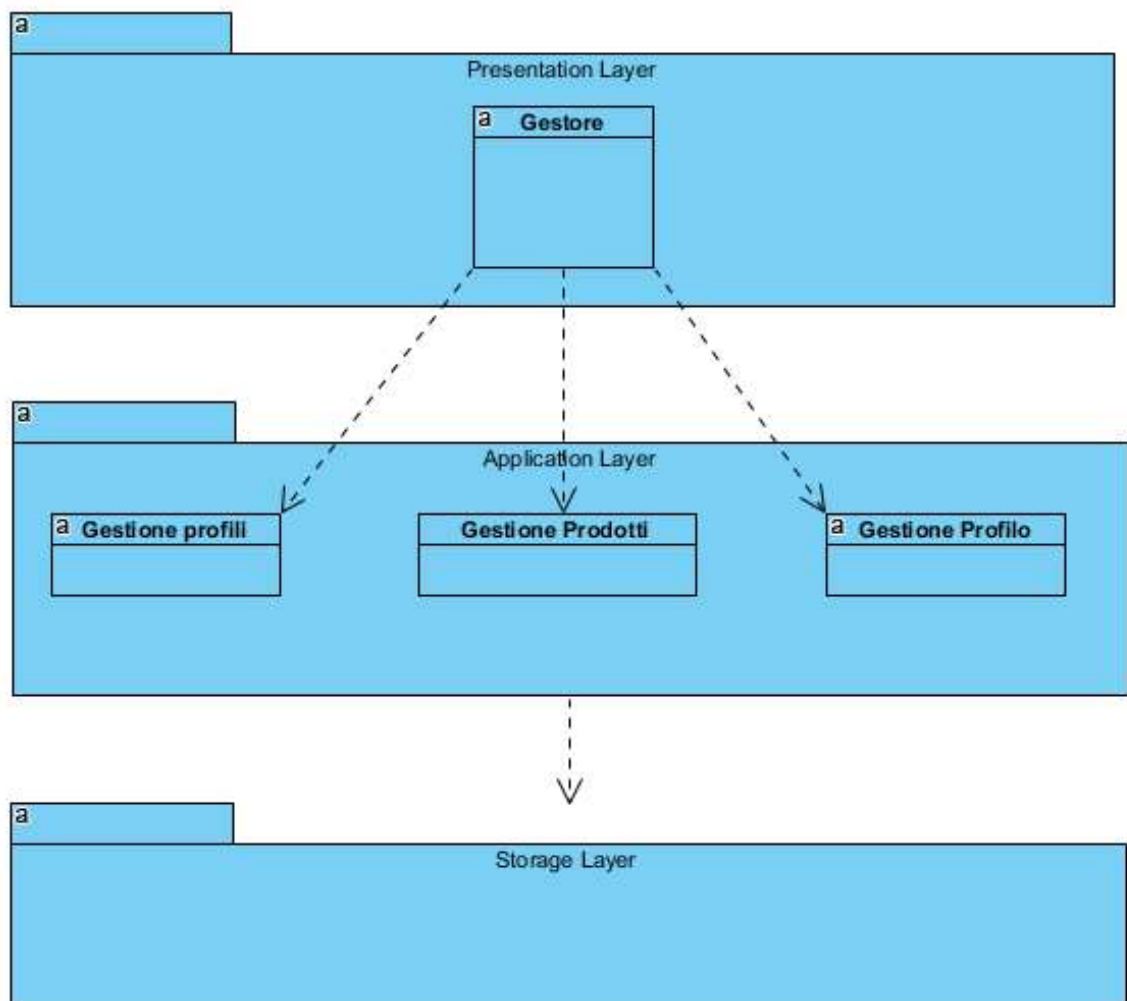
-Sottosistema Utente Registrato: include tutte le interfacce di utente non registrato ma in più sono presenti form per completare l'acquisto e per effettuare il logout

-Sottosistema Gestore: include un'interfaccia semplice dalla quale è possibile aggiungere, rimuovere, modificare i prodotti e rimuovere i clienti, inoltre tramite questo sottosistema è possibile visualizzare gli incassi e le statistiche relative ad essi.

Application Layer

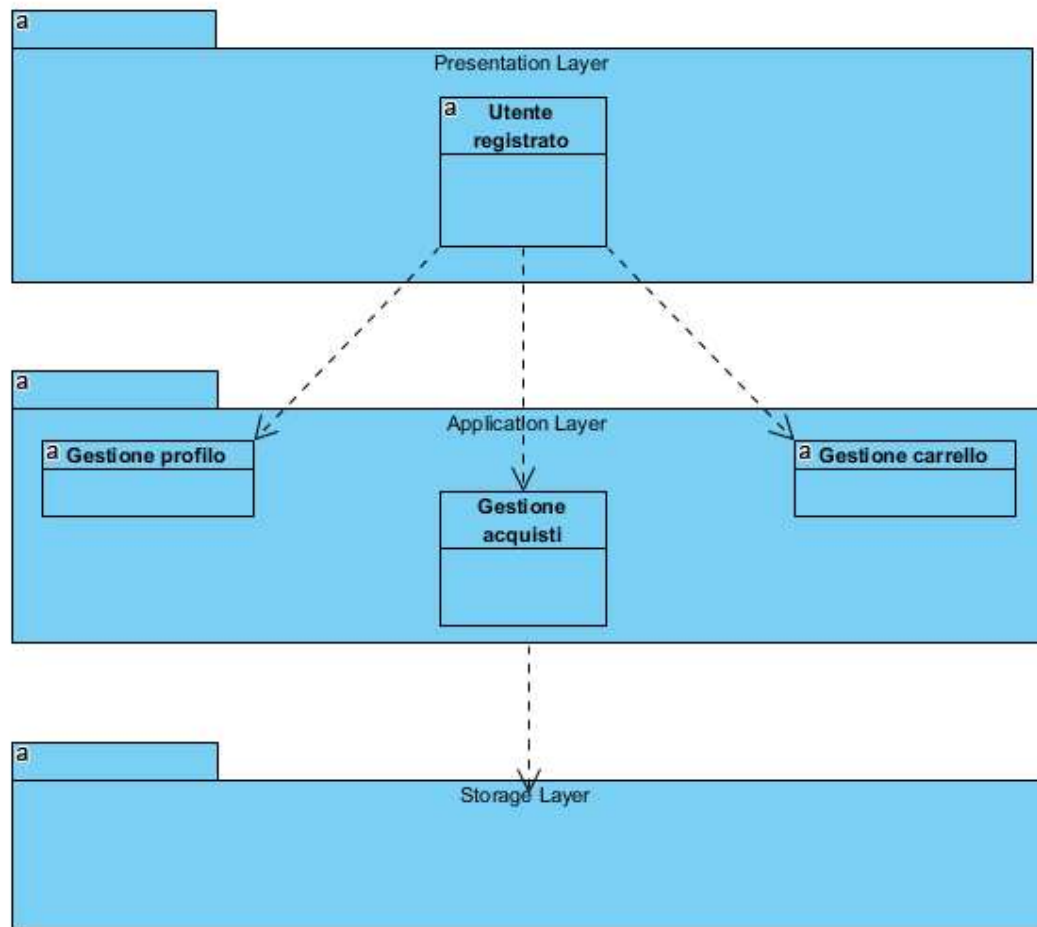
- Sottosistema Gestione acquisti: include le funzionalità necessarie per terminare un acquisto
- Sottosistema Gestione Profilo: permette di gestire il profilo di un utente registrato, fornisce servizi come la modifica della password e della mail, anche per log in e registrazione è usato questo sottosistema.
- Sottosistema Gestione prodotti: permette di gestire i prodotti fisicamente disponibili alla vendita e quindi modificarne il prezzo, caratteristiche e aggiungere o rimuovere prodotti.
- Sottosistema Gestione profili: permette di rimuovere utenti registrati alla piattaforma
- Sottosistema Gestione Carrello: gestisce i dati per mantenere il carrello attivo e aggiornato durante tutta la sessione di navigazione.

3.2.1 Gestore



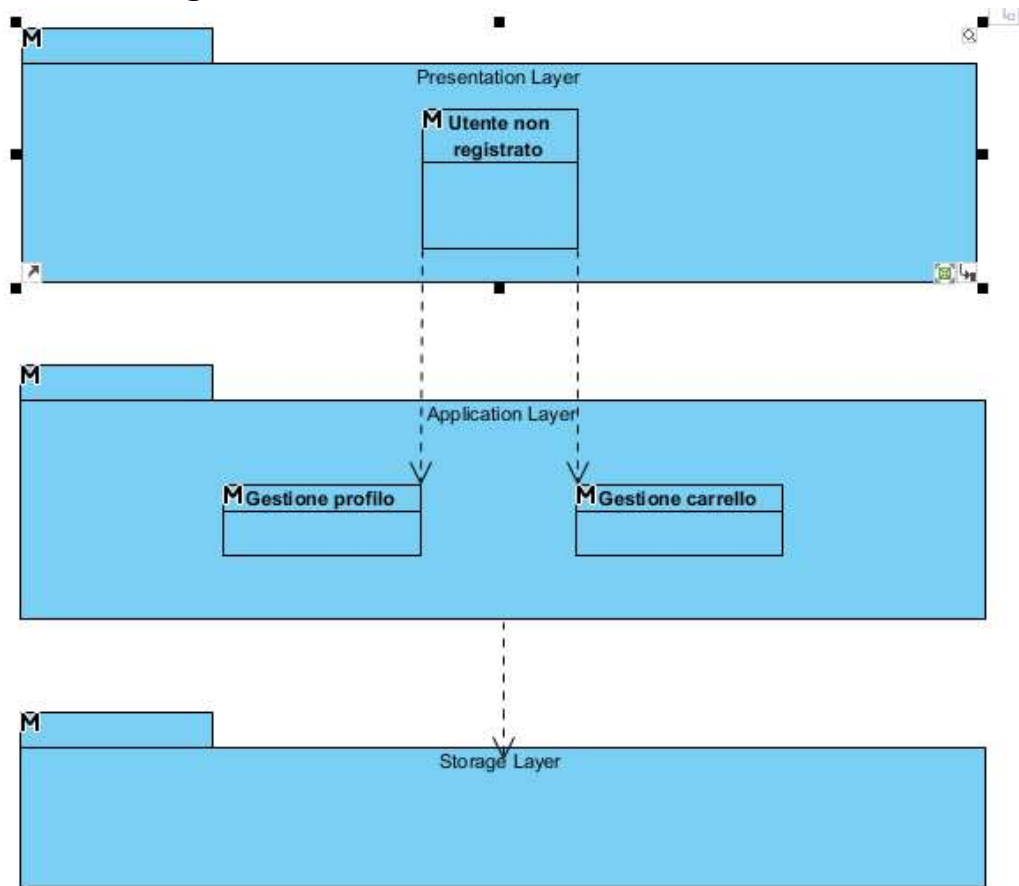
Il gestore accede a gestione profili per ottenere il servizio di rimozione del profilo di un utente mentre l'accesso a gestione prodotti è dato dal fatto che il gestore ha necessità di modificare aggiungere o rimuovere i prodotti. Il gestore inoltre può anche solo visualizzare la lista dei suoi utenti e il profilo di un singolo utente, mentre accede al sottosistema gestione profilo dato che necessita anch'esso di un log in e un log out.

3.2.2 Utente Registrato



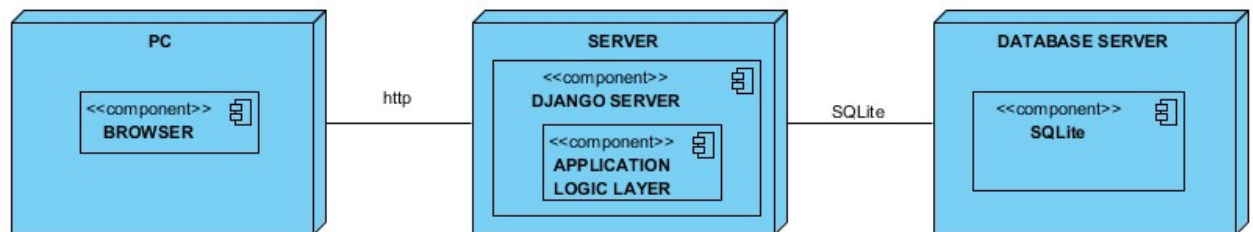
L'utente registrato può accedere alla gestione del suo profilo per modificare i suoi dati, alla gestione del carrello per aggiungere prodotti al carrello e alla gestione degli acquisti dato che è l'unica tipologia di cliente a cui è permesso effettuare acquisti sulla piattaforma.

3.2.3 Utente non registrato



L'utente non registrato ha la possibilità di registrarsi e quindi accedere ai servizi offerti da gestione profilo. L'utente non registrato può tuttavia anche salvare nel carrello alcuni prodotti che gli interessano.

3.3 Mapping Hardware/Software



Web server: viene utilizzato il framework Python Django.

Interface layer: l'utente deve utilizzare un browser di navigazione.

Application Logic Layer: Il sistema è implementato in linguaggio python. Il codice python genera html e django template language e il codice risultante viene inviato al browser del client.

Database server: il DBMS usato è SQLite il quale presenta API che permettono l'interazione tra sistema e database.

È bene fare focalizzare l'attenzione su una descrizione di cosa avviene con il framework django quando viene processata una richiesta dell'utente.

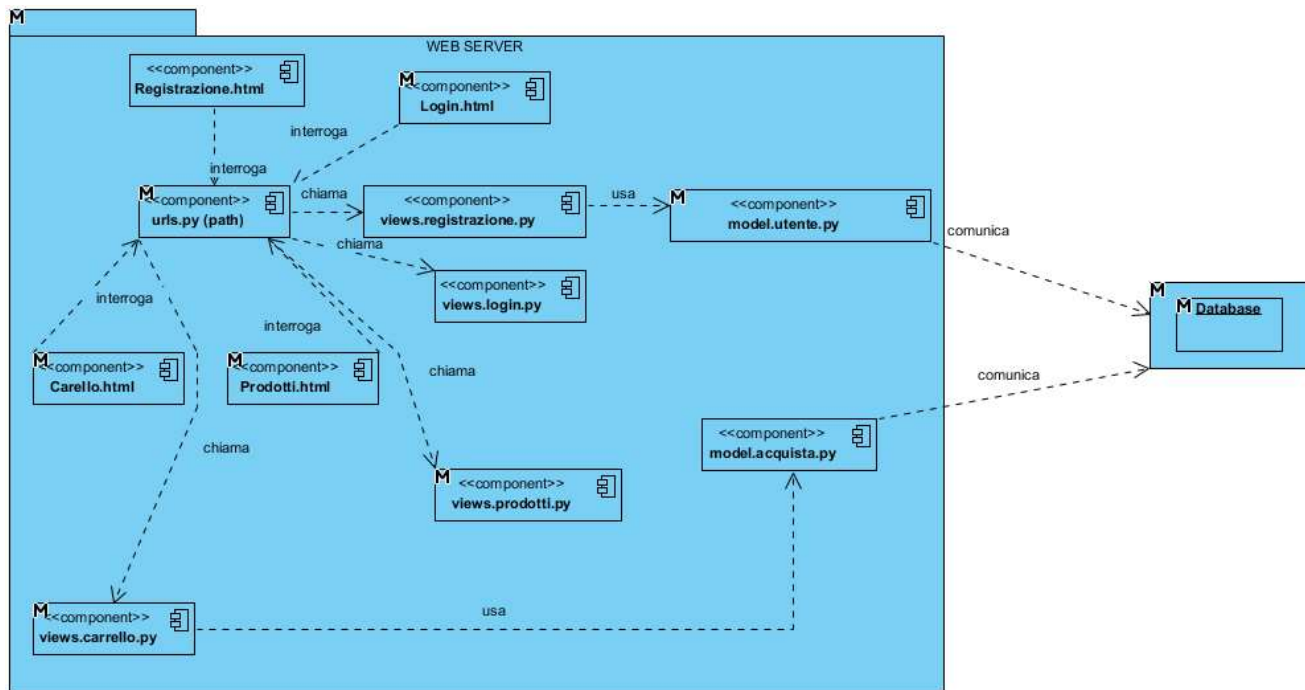
Per progettare gli URL per un'app, si crea un modulo Python chiamato informalmente un URLconf (configurazione URL). Questo modulo è puro codice Python ed è un mapping tra le espressioni del percorso URL alle funzioni Python (ovvero le nostre views).

Django fornisce un modo per tradurre gli URL in base alla lingua in cui è settato il browser del client.

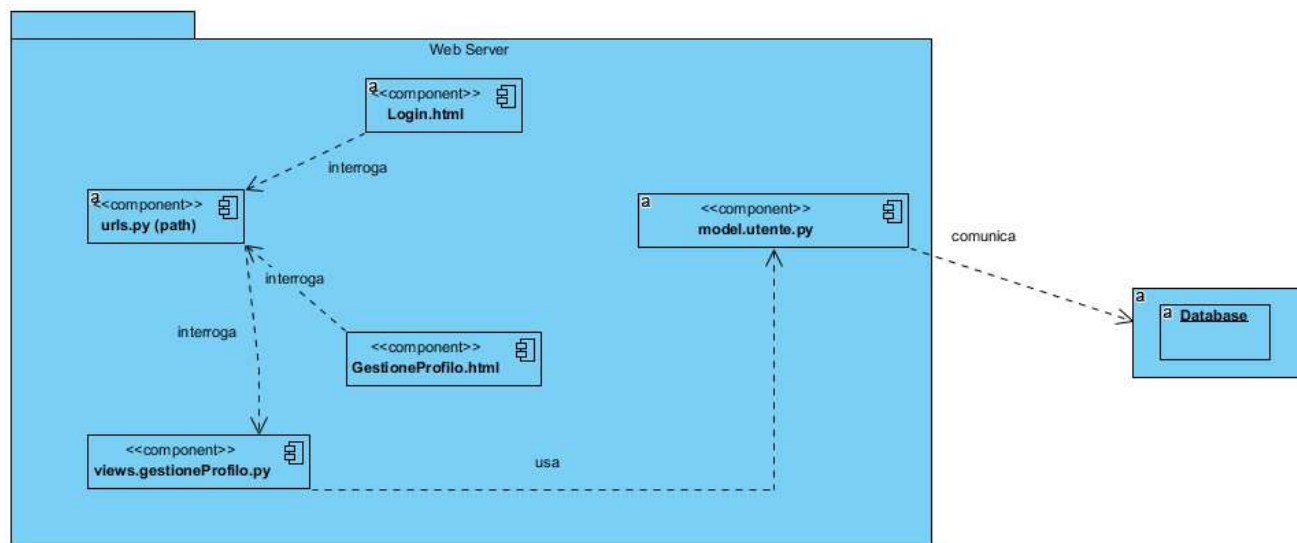
Tuttavia le richieste di un utente/gestore vengono processate in questo modo:

1. Django determina il modulo URLconf di root da utilizzare.
2. Django carica il modulo Python e cerca la variabile `urlpatterns`. Questo dovrebbe essere un elenco di istanze di tipo `django.urls.path()`.
3. Django scorre tutti i pattern in elenco e si ferma appena trova la corrispondenza desiderata.
4. Una volta trovato l'URL che corrisponde django importa e chiama la view, la quale è una semplice funzione Python, ovviamente alla view viene passata un'istanza di `HttpRequest`.
5. Se non ci sono URL che corrispondono oppure vi è un'eccezione c'è un apposito gestore degli errori creato da django che viene invocato.

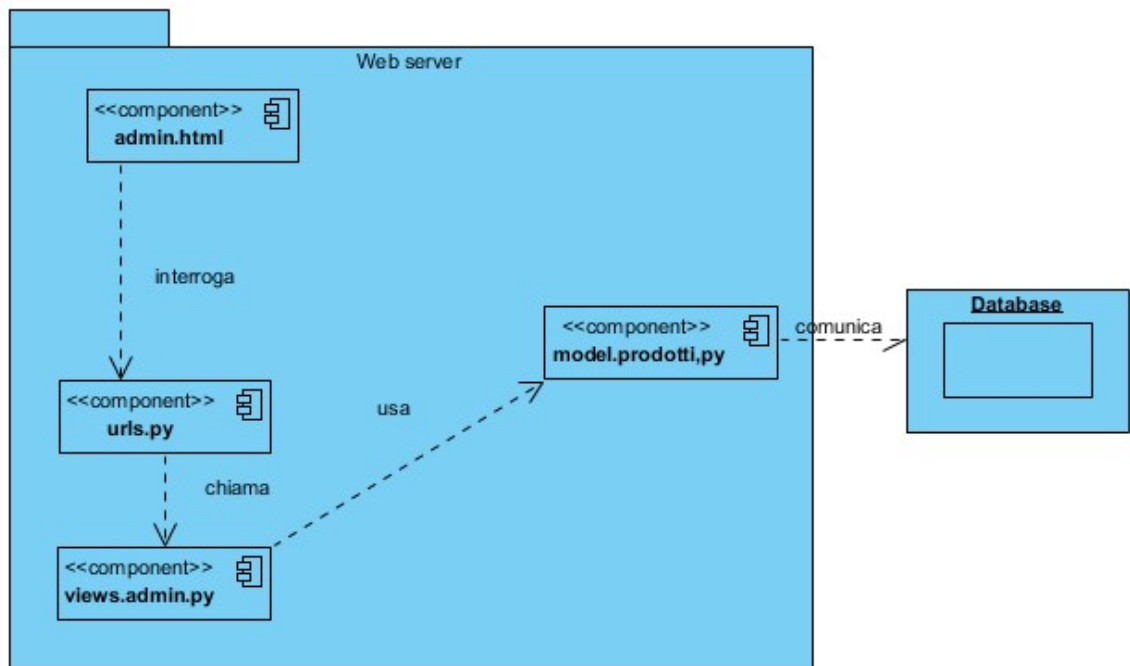
Gestione acquisti



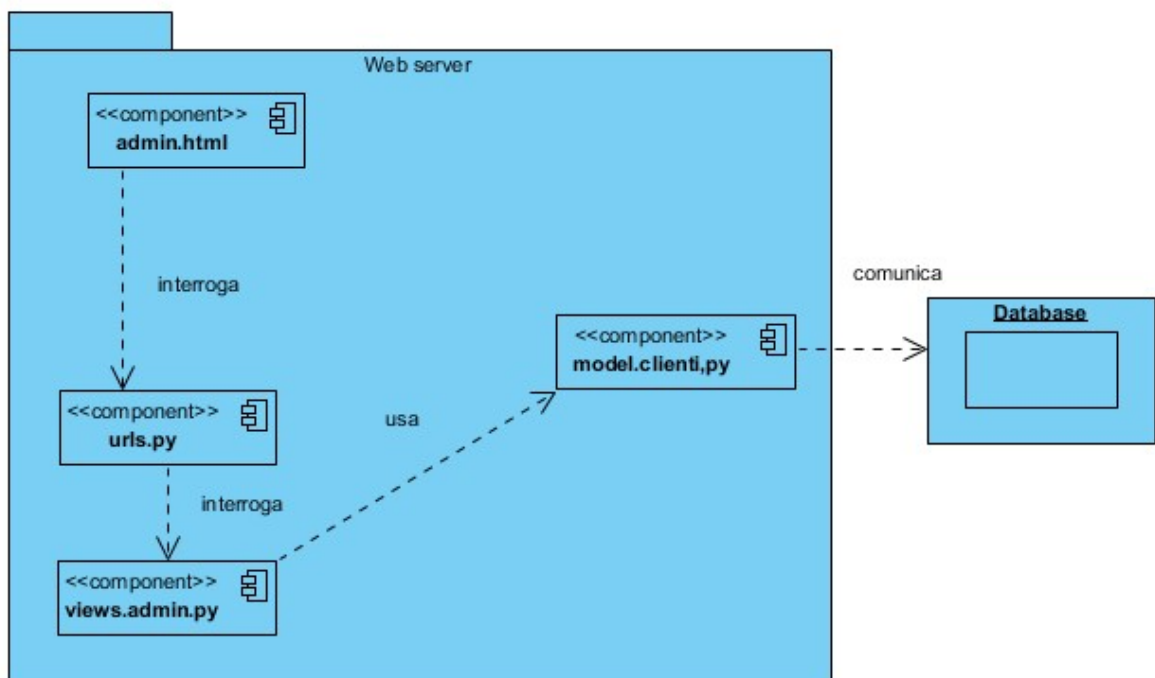
Gestioni profilo



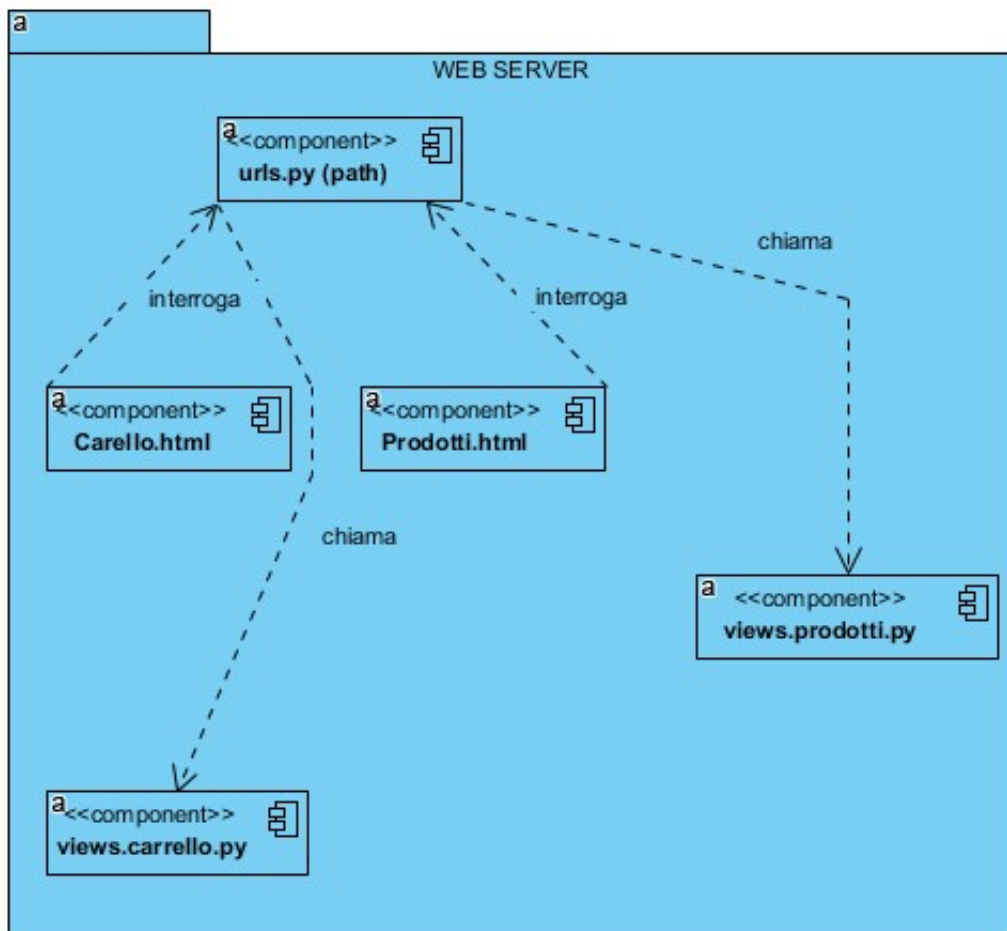
Gestione prodotti



Gestione profili



Gestione carrello



3.4 Gestione dei dati persistenti

I dati persistenti vengono salvati all'interno di un database, le entità di questo database sono: utente, prodotto, indirizzo, gestore.

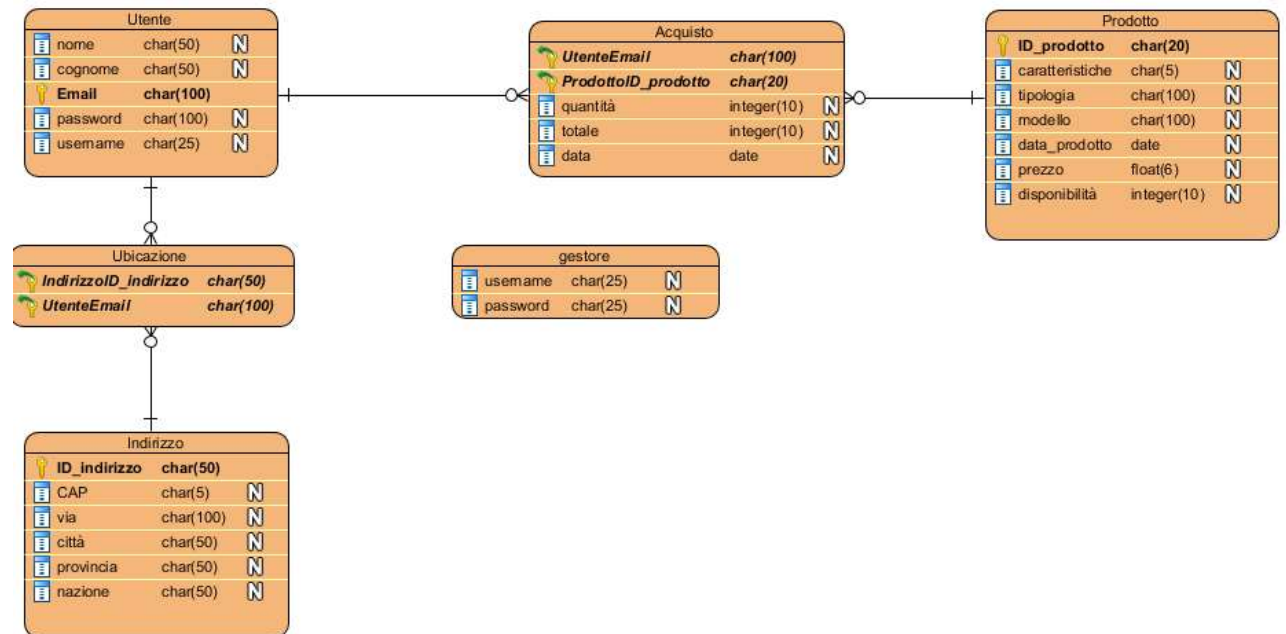
L'entità gestore verrà gestita come un'entità separata, che però sarà presente nel database fisicamente.

Dal class diagram del RAD riusciamo facilmente a ricavare le entità sopraelencate.

La lista degli utenti è memorizzata in una tabella utente, mentre il prodotto nella tabella prodotti, queste due tabelle sono legate da una relazione molti a molti che quindi verrà mappata come tabella acquista che consente di tenere traccia di quale utente ha acquistato un determinato prodotto e acquisirà le chiavi primarie di utente(e mail) e prodotto (ID). Lo stesso tipo di meccanismo viene adottato anche per le tabelle utente e indirizzo che tiene traccia degli indirizzi a cui dovranno essere recapitati gli acquisti, dato che ad un utente possono corrispondere molti indirizzi e molti utenti possono corrispondere ad un indirizzo (caso condominio) allora la relazione sarà di tipo molti a molti, di

conseguenza sarà possibile creare una tabella ubicazione che contiene la chiave primaria di indirizzo e quella di utente.

Schema ER



In tale schema sono riportati tutti i meccanismi di chiavi necessari per il database e il suo corretto funzionamento.

3.5 controllo degli accessi e sicurezza

Per la piattaforma “El Tamborito” possiamo identificare 3 tipi diversi di utenti: utente registrato, utente non registrato e gestore (anche se non presente nella tabella utente). L’utente non registrato può effettuare ricerche, visualizzare il carrello e inserire prodotti in esso in modo che possa salvare le sue scelte, un utente registrato in aggiunta alle funzionalità precedentemente descritte può anche acquistare i prodotti all’interno del carrello e visualizzare il suo profilo per poi modificare informazioni al suo interno.

Il gestore della piattaforma viene trattato come entità separata alla quale è permesso di applicare modifiche ai prezzi, aggiungere e rimuovere prodotti, rimuovere utenti e visualizzare i dettagli che riguardano questi.

Attore	Utente non registrato	Utente registrato	Gestore
Oggetto			
Gestione profilo	Registrazione	Login Modifica dati Log out	Log in Log out
Gestione profili			Visualizza profilo utente Visualizza lista utenti Rimuovi utente
Gestione acquisti		Acquista prodotto	
Gestione prodotti			Inserisci prodotto Rimuovi prodotto Sconta prodotto Visualizza prodotto
Gestione Carrello	Aggiungi prodotto Rimuovi prodotto	Aggiungi prodotto Rimuovi prodotto	

3.6 Controllo software

Il flusso di controllo nella piattaforma “el tamborito” è di tipo esplicito, centralizzato e event driven, ovvero guidato dagli eventi: non esiste una sequenza di operazioni definite a priori, è l’utente a scegliere che operazione seguire di volta in volta. L’architettura è composta da un web server e da un web client, di conseguenza parliamo di un’architettura di tipo client server. Il server riceve le richieste dei client, elabora la richiesta e invia l’output al client provvedendo a modificare opportunamente le interfacce grafiche.

3.7 Condizione 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.

3.7.1 Avvio del sistema

Per lo start-up del sistema è necessario avviare il web server. In seguito, tramite l’interfaccia di Login, sarà possibile autenticarsi tramite opportune credenziali (username e password) come Amministratore, Operatore o Cliente.

3.7.2 Terminazione del sistema

Al momento della corretta chiusura dell’applicazione, si ha la terminazione del sistema con un regolare Logout. Per terminare il server è necessario l’intervento dell’amministratore. Questo dovrà effettuare la procedura di terminazione dopo la quale non sarà possibile per nessun client connettersi al sistema.

3.7.3 Fallimento del sistema

Nel caso si verifichi un errore dovuto all’hardware o al software si cercherà di ripristinare una configurazione del sistema precedente allo stato d’errore. Visto che i dati sono gestiti dal DBMS non c’è rischio di perderli. Tuttavia, se si

verifica un guasto al supporto di memorizzazione si può avere una perdita di dati in caso di assenza di ridondanza dei dati.

Scenari

Nome scenario	Start-up Sistema
Istanze di attori partecipanti	Antonio: Amministratore
Flusso di Eventi	<ol style="list-style-type: none"> 1. Antonio decide di voler avviare il sistema e effettua la procedura di avvio; 2. Il sistema, con le opportune procedure di avvio, attiva i server e i relativi servizi in remoto rendendosi disponibile ad eventuali richieste; 3. Il sistema notifica il successo della procedura.

Nome scenario	Shutdown Sistema
Istanze di attori partecipanti	Antonio: Amministratore
Flusso di Eventi	<ol style="list-style-type: none"> 1. Antonio decide di voler arrestare il sistema e effettua la procedura di arresto; 2. Il sistema effettua una scansione per verificare se ci sono ancora richieste in sospeso; 3. Il sistema porta a termine le eventuali richieste in sospeso; 4. Tramite le opportune procedure di arresto il sistema disattiva i servizi in remoto e il server; 5. Il sistema notifica il successo della procedura.

Casi d'uso

ID	UC_Startup
Nome caso d'uso	Startup Server
Istanze di Attori partecipanti	Amministratore
Condizione di entrata	L'amministratore visualizza la console per effettuare l'avvio del sistema.
Flusso di Eventi	<div> <div>Utente</div> <div>Sistema</div> </div> <p>L'amministratore effettua la procedura d'avvio</p> <p>Il sistema attiva i propri servizi rendendosi disponibile in tutte le sue funzionalità.</p>
Condizione di uscita	Il server è attivo e i relativi servizi sono disponibili.
Eccezioni	Errore Startup.

ID	UC_Shutdown
Nome caso d'uso	Shutdown Server
Istanze di Attori partecipanti	Amministratore
Condizione di entrata	L'amministratore visualizza la console per effettuare la terminazione del sistema.
Flusso di Eventi	<div> <div> <p>Utente</p> <p>L'amministratore effettua la procedura d'arresto</p> </div> <div> <p>Sistema</p> <p>Il sistema effettua la procedura per la terminazione dei servizi e del server.</p> </div> </div>
Condizione di uscita	Il server si è spento correttamente.
Eccezioni	Errore Shutdown.

4. Servizi dei sottosistemi

4.1 Gestione Profilo

Sottosistema	Gestione profilo
Descrizione	Sottosistema che gestisce e raccoglie tutte le funzionalità di cui possono usufruire gli utenti del sistema, ovvero: registrazione di un profilo, login, modifica del profilo creato.
Servizi Offerti	
Login	Questa funzionalità permette di effettuare l'accesso come utente registrato.
Logout	Questa funzionalità permette di effettuare la disconnessione come utente registrato.
Modifica profilo	Questa funzionalità permette di modificare le proprie credenziali d'accesso.

4.2 Gestione Profili

Sottosistema	Gestione profili
Descrizione	Sottosistema che gestisce e raccoglie tutte le funzionalità relative alla gestione dei profili di cui può usufruire l'amministratore del sistema, ovvero: visualizzare la lista dei profili o le informazioni di un solo profilo.
Servizi Offerti	
Visualizza profili	Questa funzionalità permette di visualizzare la lista dei clienti registrati al sistema.
Visualizza profilo	Questa funzionalità permette di visualizzare le informazioni relative a uno dei profili registrati al sistema.

4.3 Gestione prodotti

Sottosistema	Gestione prodotti
Descrizione	Sottosistema che gestisce e raccoglie tutte le funzionalità relative alla gestione dei prodotti di cui può usufruire l'amministratore del sistema, ovvero aggiunta o eliminazione di un prodotto dal magazzino.
Servizi Offerti	
Aggiungi prodotto	Questa funzionalità permette di aggiungere un prodotto al magazzino.
Rimozione prodotto	Questa funzionalità permette di rimuovere un prodotto al magazzino.
Modifica prodotto	Questa funzionalità permette di modificare il prezzo del prodotto

4.4 Gestione Acquisti

Sottosistema	Gestione acquisti
Descrizione	Sottosistema che gestisce gli acquisti, permette agli utenti di effettuare acquisti e di rimuovere le quantità desiderate dal magazzino
Servizi Offerti	
Acquista prodotto	Questa funzionalità permette di aggiungere un prodotto al magazzino.

4.5 Gestione Carrello

Sottosistema	Gestione carrello
Descrizione	Sottosistema che gestisce e raccoglie le funzionalità del carrello.
Servizi Offerti	
Aggiungi prodotto al carrello	Questa funzionalità permette di aggiungere uno o più prodotti al carrello.
Rimuovi prodotto dal carrello	Questa funzionalità permette di rimuovere uno o più prodotti presenti nel carrello.

5. Glossario

Client: componente che accede ai servizi o alle risorse di un'altra componente, detta server.

Deployment Diagram: Schema che descrive la struttura dinamica del sistema

DBMS: programma informatico (o, più frequentemente, un insieme di programmi) progettato per gestire un database, ovvero un insieme di numerosi dati strutturati. Le operazioni, normalmente, sono richieste da un gran numero di utenti.

Form: finestra di dialogo incorporata in una pagina Web che consente all'utente di inserire informazioni destinate ad un server. Generalmente richiede un programma sul server che si occupi di esaminare le informazioni inviate. E' composto da spazi (campi) predefiniti, ad esempio menù a tendina, elenchi puntati o caselle di testo libero.

JDBC: API per il linguaggio di programmazione Java che serve ai client per connettersi a un database. Fornisce metodi per interrogare e modificare i dati. È orientata ai database relazionali.

Login: Procedura attraverso la quale ci si collega con un qualsiasi servizio in linea. All'utente viene assegnato un nome di login ed una password che vengono richiesti dal sistema ogni volta che ci si collega.

Layer: E' un insieme di classi con funzionalità simile (tipicamente raggruppati in un unico package).

Logout: Operazione attraverso la quale si termina un collegamento con un sistema al quale si ha accesso attraverso un nome utente e una password (vedi login).

MySQL: Database management system relazionale, composto da un client con interfaccia a caratteri e un server, disponibile su molte piattaforme.

Password: È un metodo di sicurezza che, mediante una stringa di caratteri, permette di identificare un utente specifico. Generalmente le password sono formate da una sequenza di lettere e numeri; digitando correttamente questi caratteri, si può avere accesso al computer o alla rete.

Package: Un Package rappresenta una collezione di classi ed interfacce che possono essere raggruppate in base alla funzione comune da esse svolta

Server: Programma di gestione di un servizio che invia informazioni in un particolare formato ricevuto e interpretato da un programma Client dal lato ricevente.

Shutdown: fase di terminazione del sistema.

Start-up: processo di accensione e di avvio di un computer, di un dispositivo o di un sistema.

Django: framework web usato per l'implementazione.