UNIVERSITÀ DEGLI STUDI DI SALERNO

Warehouse project

Object Design Document

Top Manager:

NOME
Prof. De Lucia Andrea

Partecipanti:

NOME	MATRICOLA
Michele Gargiulo	0512105530

Indice

1. Introduzione

1.1 Object Design Trade-offs

Il seguente documento ha lo scopo di produrre un modello capace di integrare in modo coerente e preciso tutte le funzionalità individuate nelle fasi precedenti. In particolare, definisce le interfacce delle classi, le operazioni, i tipi, gli argomenti e il signature dei sottosistemi definiti nel System Design.
Inoltre, sono specificati i trade-off e le linee guida.

Comprensibilità vs Tempo:

Il codice deve essere quanto più comprensibile possibile per facilitare la fase di testing ed eventuali future modifiche. Il codice sarà quindi accompagnato da commenti che ne semplifichino la comprensione. Ovviamente questa caratteristica aggiungerà un incremento di tempo allo sviluppo del nostro progetto.

Prestazioni vs Costi:
Essendo il progetto sprovvisto di budget, al fine di mantenere prestazioni elevate, per alcune funzionalità verranno utilizzati dei template open source Bootstrap.

Interfaccia vs Usabilità:
L'interfaccia grafica è stata realizzata in modo da essere molto semplice, chiara e concisa, fa uso di form e pulsanti disposti in maniera da rendere semplice l'utilizzo del sistema da parte dell'utente finale.

Sicurezza vs Efficienza:
La sicurezza rappresenta uno degli aspetti importanti del sistema. Tuttavia, dati i tempi di sviluppo molto limitati, ci limiteremo ad implementare sistemi di sicurezza basati su username e password degli utenti.

1.2 Linee Guida per la Documentazione delle Interfacce

Gli sviluppatori dovranno seguire alcune linee guida per la scrittura del codice:

Naming Convention

- È buona norma utilizzare nomi:

    1. Descrittivi
    2. Di uso comune
    3. Lunghezza medio-corta
    4. Utilizzando solo caratteri consentiti (a-z, A-Z, 0-9)
    5. È possibile utilizzare il carattere underscore "_", nel caso di variabili costanti o proprietà statiche

Variabili:

- I nomi delle variabili devono cominciare con una lettera minuscola, e le parole seguenti con la lettera maiuscola. Devono essere tutte allineate per facilitare la leggibilità.

Esempio: numeroOrdine
Esempio: COSTO_SPEDIZIONE
Metodi:

- I nomi dei metodi devono cominciare con una lettera minuscola, e le parole seguenti con la lettera maiuscola. Il nome del metodo tipicamente consiste di un verbo che identifica una azione, seguito dal nome di un oggetto. I nomi dei metodi per l'accesso e la modifica delle variabili dovranno essere del tipo getNomeVariabile() e setNomeVariabile().

Esempio: getOrders(), setOrder()

- La descrizione dei metodi deve apparire prima di ogni dichiarazione di metodo, e deve descriverne lo scopo. Deve includere anche informazioni sugli argomenti, sul valore di ritorno, e se applicabile, sulle eccezioni.

Classi:

- I nomi delle classi devono cominciare con una lettera maiuscola, e anche le parole seguenti all'interno del nome devono cominciare con una lettera maiuscola. I nomi di quest'ultime devono fornire informazioni sul loro scopo.

Esempio: AdminOperations.java

Pagine:

L'indentazione deve essere effettuata con un TAB e qualunque sia il linguaggio usato per la produzione di codice, ogni istruzione deve essere opportunamente indentata.

Es.

```
<html>
<head>
</head>
<body>
</body>
</html>
```

Deve essere sostituita da:

```
<html>
    <head>
    </head>
    <body>
    </body>
</html>
```

1.3 Definizioni, acronimi e abbreviazioni

- RAD: Requirements Analysis Document
- SDD: System Design Document
- ODD: Object Design Document
- DB: Database

1.4 Riferimenti

- B. Bruegge, A. H. Dutoit, Object Oriented Software Engineering - Using UML, Pattern and Java, Prentice Hall, 3rd edition, 2009
- Documenti SDD e RAD del progetto Warehouse

2. Packages

La gestione del nostro sistema è suddivisa in tre livelli:

- Interface layer
- Application Logic layer
- Storage layer

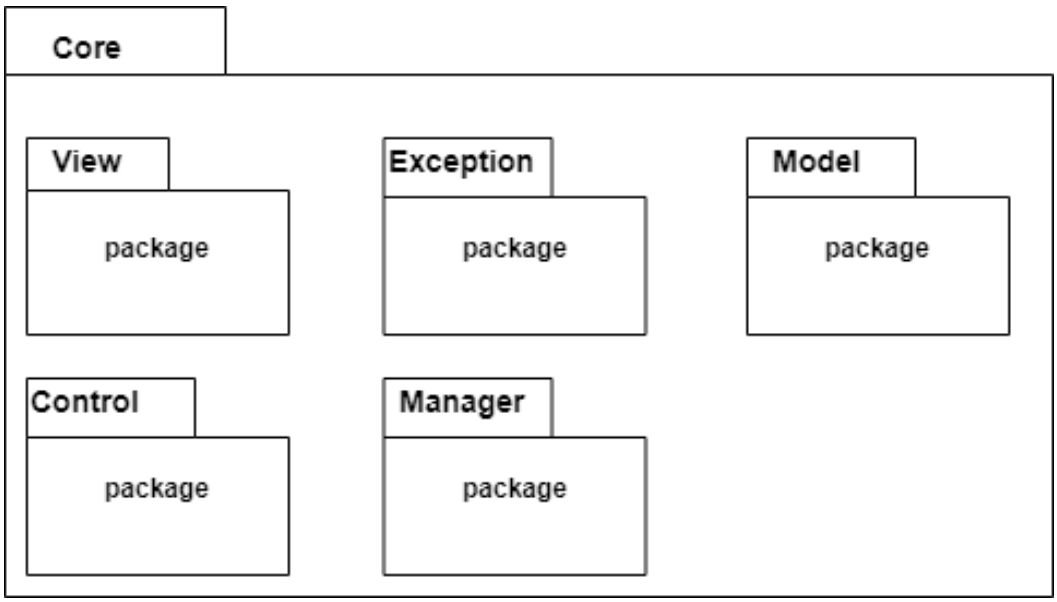
Il package Warehouse contiene sottopackage che a loro volta inglobano classi atte alla gestione delle richieste utente. Le classi contenute nel package svolgono il ruolo di gestore logico del sistema.

Interface layer	Rappresenta l'interfaccia del sistema, ed offre la possibilità all'utente di interagire con quest'ultimo, offrendo sia la possibilità di inviare, in input, che di visualizzare, in output, dati.

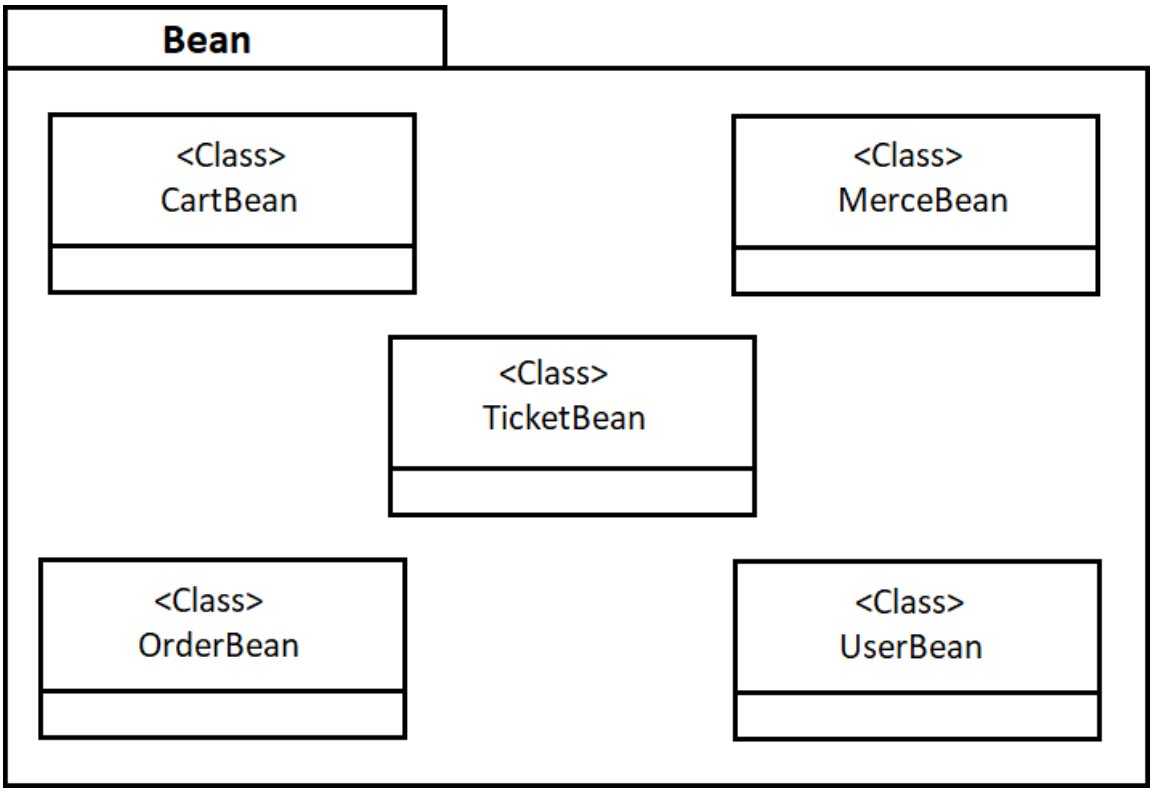
Application Logic layer	Ha il compito di elaborare i dati da inviare al client, e tramite lo Storage Layer, accede ai dati persistenti (database). Si occupa di varie gestioni del sistema: 1. **Gestione Utente** 2. **Gestione Shop** 3. **Gestione Acquisti** 4. **Gestione Ordini** 5. **Gestione Ticket**

Storage layer	Ha il compito di memorizzare i dati sensibili del sistema, utilizzando un DBMS, inoltre riceve le varie richieste dall' Application Logic layer inoltrandole al DBMS e restituendo i dati richiesti.

2.1 Packages core

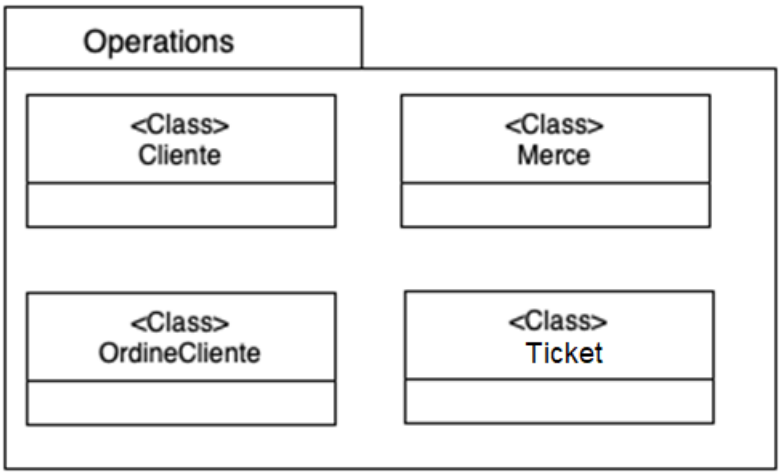


2.1.1 Packages model



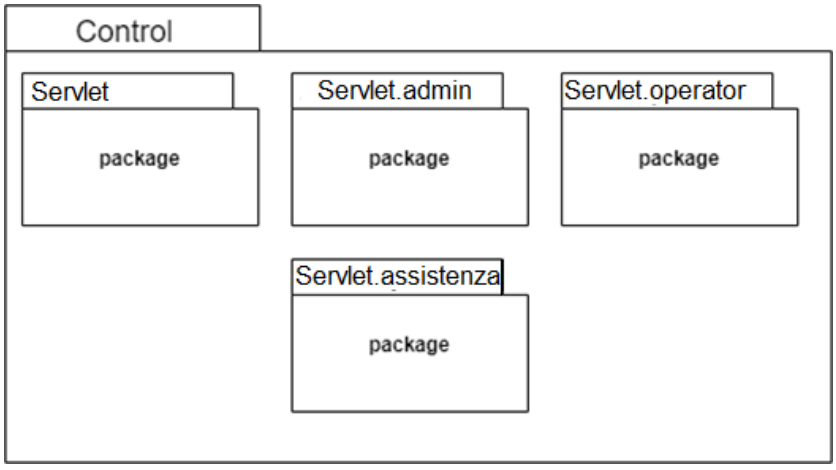
Classe:	**Descrizione:**
CartBean	Descrive il carello e gli oggetti contenuti in esso
MerceBean	Descrive gli oggetti contenuti nel database
OrderBean	Descrive l'ordine di un determinato utente
UserBean	Descrive gli utenti registrati al sistema
TicketBean	Descrive i ticket presenti nel sistema

2.1.2 Packages Manager

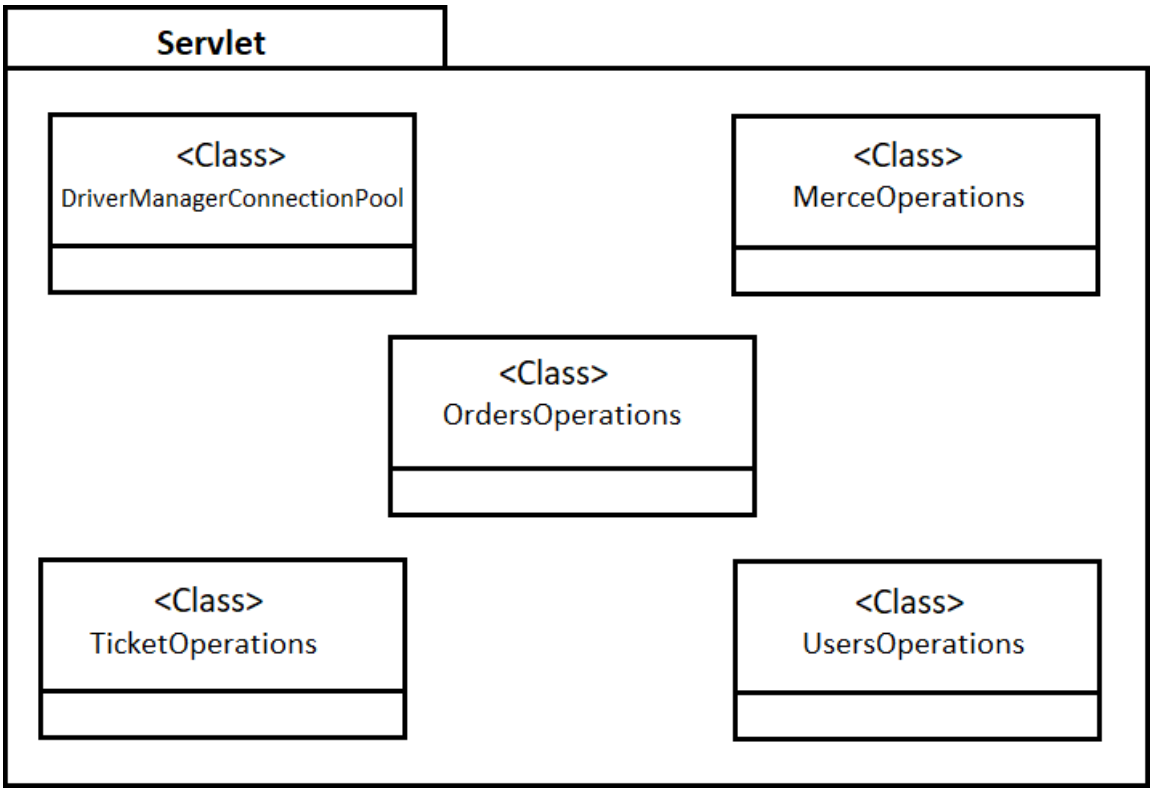


Classe:	**Descrizione:**
Cliente	Permette ad un utente di inserire, modificare, cancellare e visualizzare il proprio profilo.
Merce	Permette la gestione della merce (inserire, modificare, cancellare, ricercare: marca modello e categoria).
OrdineCliente	Permette la gestione degli ordini (inserire, cancellare, ricercare: numero ordine e codice fiscale).
Ticket	Permette la gestione dei ticket (inserire nuovi ticket, modificare ticket esistenti, ricerca dei ticket nel sistema)

2.1.3 Packages Control

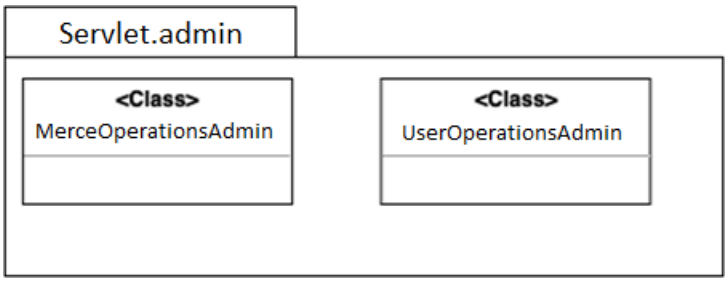


2.1.3.0 Servlet



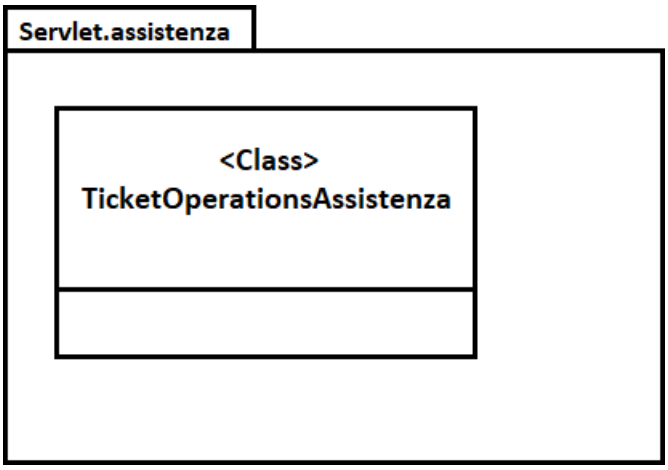
Classe:	**Descrizione:**
DriverManagerConnectionPool	Gestisce la connessione con il database
MerceOperations	Gestisce le operazioni con la merce, lato cliente, tra cui la visualizzazione della sezione shop, l'inserimento di un prodotto nel carrello, la ricerca dei prodotti all'interno della sezione shop.
OrdersOperations	Gestisce le operazioni riguardanti gli ordini, lato cliente, tra cui l'inserimento di un nuovo ordine, l'annullamento di un ordine, la ricerca di un ordine.
TicketOperations	Gestisce le operazioni riguardanti i ticket, lato cliente, tra cui l'inserimento di un nuovo ticket, la lista dei ticket aperti e chiusi
UsersOperations	Gestisce le operazioni riguardanti gli utenti nel sistema, lato cliente, tra cui la modifica e la visualizzazione dei propri dati

2.1.3.1 servlet.admin



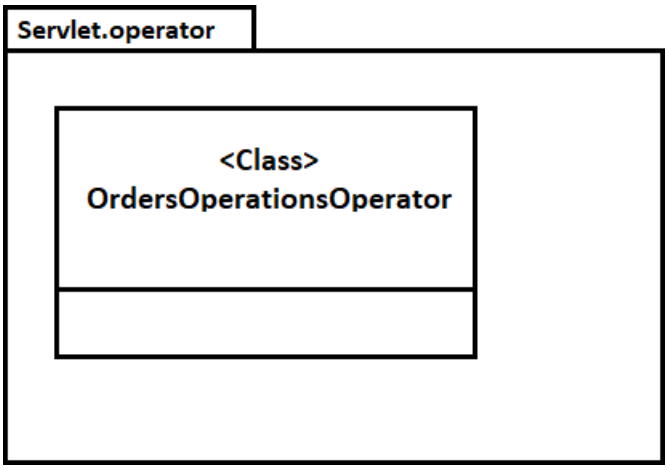
Classe:	Descrizione:
MerceOperationsAdmin	Gestisce le operazioni riguardante la merce, lato admin.
UserOperationsAdmin	Gestisce le operazioni riguardanti gli utenti, lato admin.

2.1.3.2 servlet.assistenza



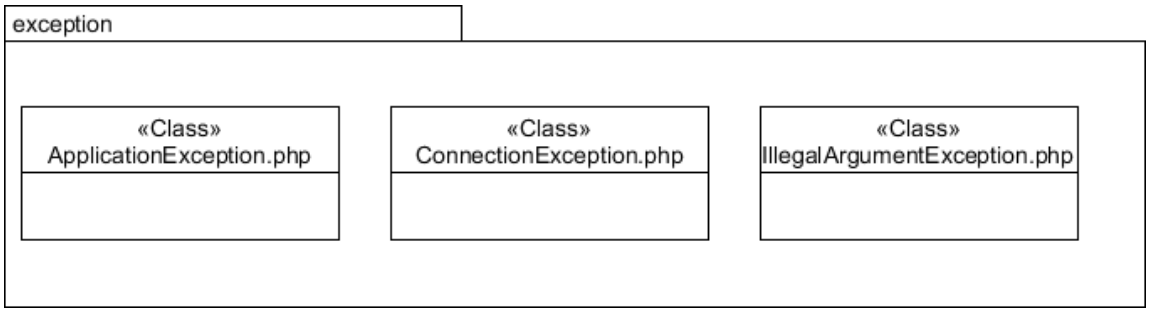
Classe:	Descrizione:
TicketOperationsAssistenza	Gestisce le operazioni riguardanti i ticket, lato assistenza.

2.1.3.3 servlet.operator



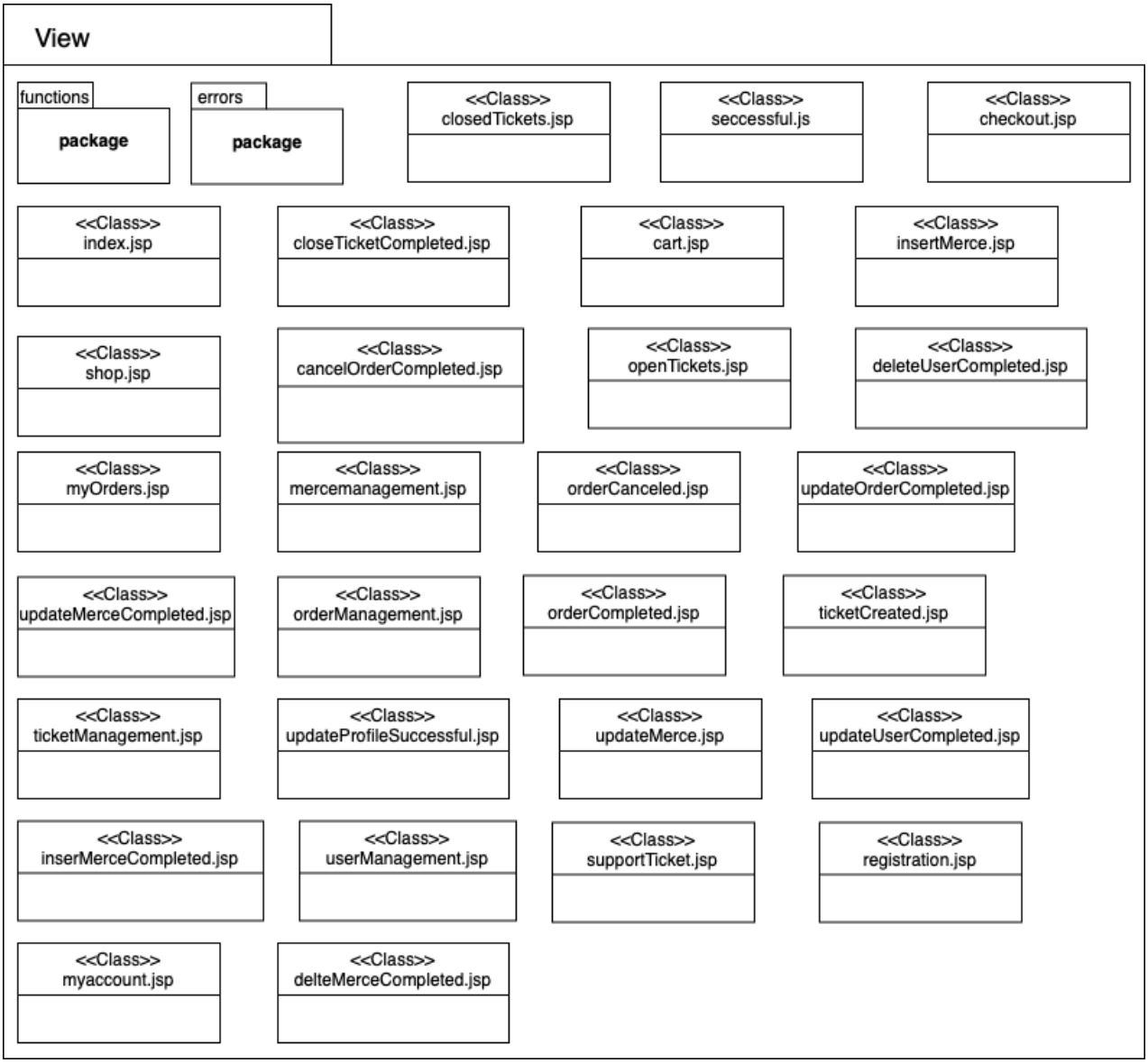
Classe:	Descrizione:
OrdersOperationsOperator	Gestisce le operazioni riguardante gli ordini, lato operatore.

2.1.4 Packages Exception



Classe:	Descrizione:
ApplicationException	L'eccezione che viene lanciata quando non viene eseguita una query.
ConnectionException	L'eccezione che viene lanciata quando non c'è connessione al database.
IllegalArgumentException	L'eccezione che viene lanciata quando i parametri passati ad un metodo non sono corretti.

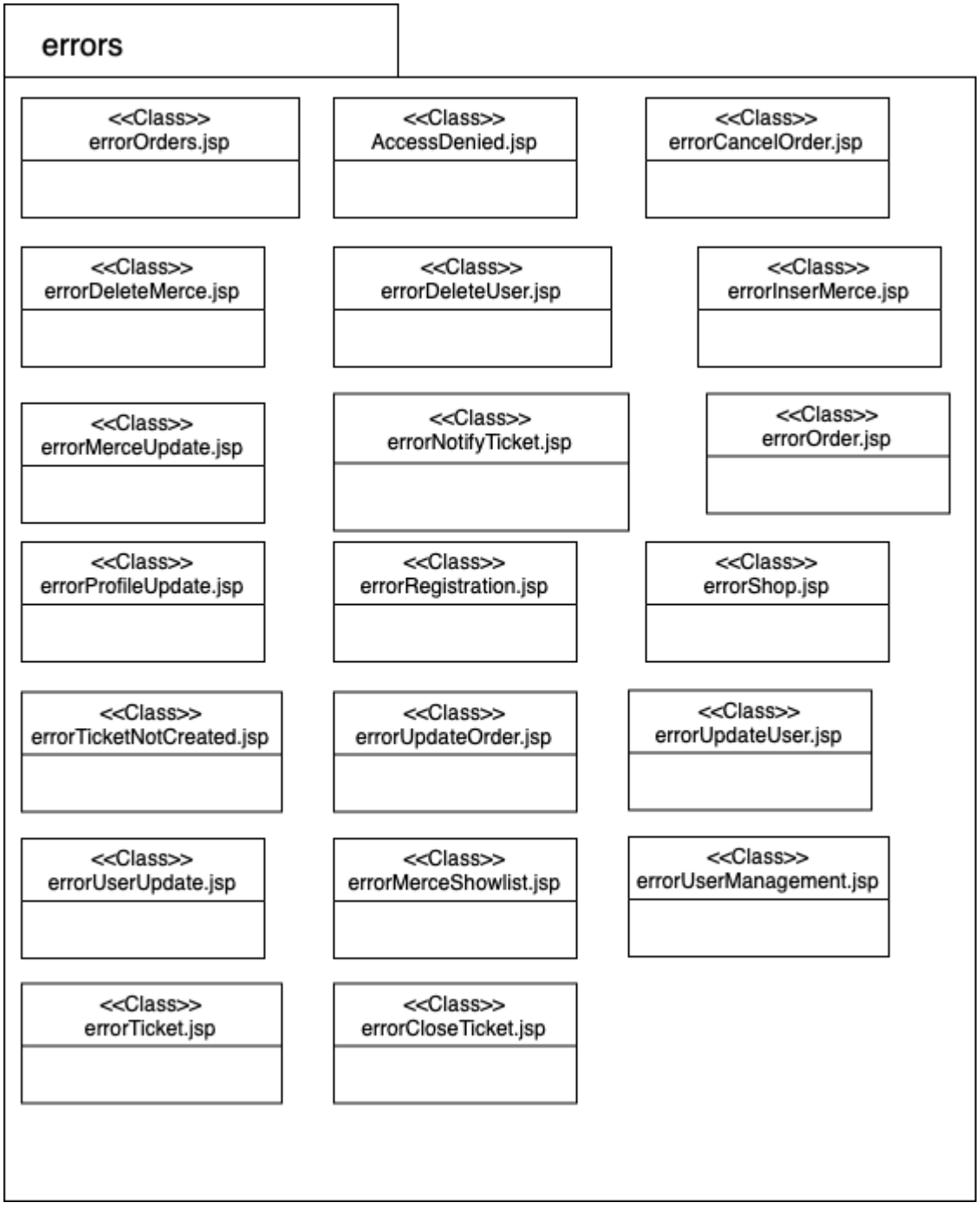
2.1.5 Packages View



Classe:	**Descrizione:**
index.jsp	Visualizza la pagina principale del sito
cancelOrderCompleted.jsp	Visualizza la pagina di conferma in seguito all'annullamento di un ordine, lato operatore.
cart.jsp	Visualizza la pagina del carrello
checkout.jsp	Visualizza la pagina contenente la form di acquisto dei prodotti.
closedTickets.jsp	Visualizza la pagina contenente la lista dei ticket chiusi.
closeTicketCompleted.jsp	Visualizza la pagina di conferma in seguito ad una chiusura del ticket.
deleteMerceCompleted.jsp	Visualizza la pagina di conferma in seguito alla cancellazione di un prodotto dal sistema

deleteUserCompleted.jsp	Visualizza la pagina di conferma in seguito alla cancellazione di un utente dal sistema
insertMerce.jsp	Visualizza la pagina che permette di inserire un prodotto nel sistema
insertMerceCompleted.jsp	Visualizza la pagina di conferma in seguito all'inserimento di un prodotto nel sistema
mercemanagement.jsp	Visualizza la pagina che permette di gestire la merce del sistema, lato admin.
myaccount.jsp	Visualizza la pagina che permette di modificare e visualizzare i propri dati personali.
myOrders.jsp	Visualizza la pagina che permette di visualizzare i propri ordini inseriti
openTickets.jsp	Visualizza la pagina contenente la lista dei ticket aperti
orderCanceled.jsp	Visualizza la pagina di conferma in seguito all'annullamento di un ordine, lato cliente.
orderCompleted.jsp	Visualizza la pagina di conferma in seguito all'inserimento di un nuovo ordine, lato cliente.
orderManagement.jsp	Visualizza la pagina che permette di gestire gli ordini nel sistema
registration.jsp	Visualizza la pagina che permette di registrarsi al sistema
shop.jsp	Visualizza la pagina principale dedicata allo shopping online.
successful.jsp	Visualizza la pagina di conferma in seguito ad una registrazione
supportTicket.jsp	Visualizza la pagina che permette di aprire un nuovo ticket e visualizzare quelli aperti o chiusi.
ticketCreated.jsp	Visualizza la pagina di conferma in seguito all'apertura di un nuovo ticket.
ticketManagement.jsp	Visualizza la pagina che permette di gestire i ticket nel sistema.
updateMerce.jsp	Visualizza la pagina che permette di modificare la merce presente nel sistema
updateMerceCompelted.jsp	Visualizza la pagina di conferma in seguito alla modifica di un prodotto del sistema.
updateOrderCompleted.jsp	Visualizza la pagina di conferma in seguito alla modifica di un ordine del sistema.
updateProfileSuccessful.jsp	Visualizza la pagina di conferma in seguito alla modifica del profilo personale
updateUserCompleted.jsp	Visualizza la pagina di conferma in seguito alla modifica di un utente del sistema.
userManagement.jsp	Visualizza la pagina che permette di gestire gli utenti

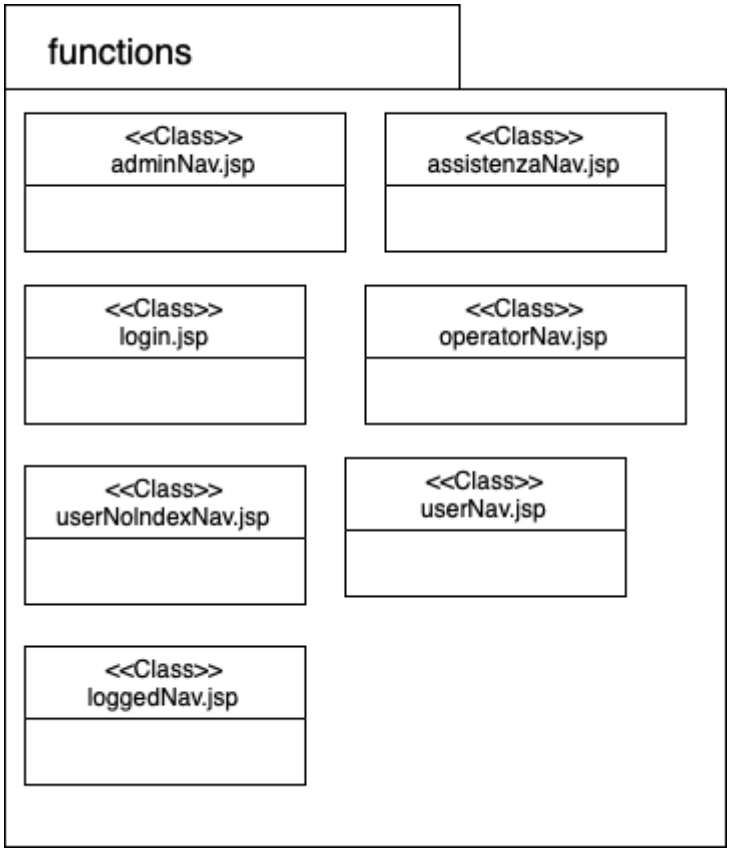
2.1.5.1 errors



accessDenied.jsp	Visualizza la pagina di errore per un accesso non autorizzato
errorCancelOrder.jsp	Visualizza la pagina di errore per un errore riscontrato durante l'annullamento di un ordine
errorCloseTicket.jsp	Visualizza la pagina di errore per un errore riscontrato durante la chiusura di un ticket.

errorDeleteMerce.jsp	Visualizza la pagina di errore per un errore riscontrato durante l'eliminazione di un prodotto.
errorDeleteUser.jsp	Visualizza la pagina di errore per un errore riscontrato durante l'eliminazione di un utente dal sistema
errorInsertMerce.jsp	Visualizza la pagina di errore per un errore riscontrato durante l'inserimento di un nuovo prodotto nel sistema
errorMerceShowlist.jsp	Visualizza la pagina di errore per un errore riscontrato durante la visualizzazione dei prodotti all'interno del sistema.
errorMerceUpdate.jsp	Visualizza la pagina di errore per un errore riscontrato durante la modifica di un prodotto.
errorNotifyTicket.jsp	Visualizza la pagina di errore per un errore riscontrato durante la notifica del cambiamento di stato di un ticket
errorOrder.jsp	Visualizza la pagina di errore per un errore generico riguardante gli ordini.
errorProfileUpdate.jsp	Visualizza la pagina di errore per un errore riscontrato durante la modifica del proprio profilo
errorRegistration.jsp	Visualizza la pagina di errore per un errore riscontrato durante la registrazione.
errorShop.jsp	Visualizza la pagina di errore per un errore generico della sezione shop.
errorTicket.jsp	Visualizza la pagina di errore per un errore generico riguardante i ticket.
errorTicketNotCreated.jsp	Visualizza la pagina di errore per un errore riscontrato durante la creazione di un nuovo ticket.
errorUpdateOrder.jsp	Visualizza la pagina di errore per un errore riscontrato durante la modifica di un ordine
errorUpdateUser.jsp	Visualizza la pagina di errore per un errore generico della modifica degli utenti
errorUserManagement.jsp	Visualizza la pagina di errore per un errore riscontrato nella sezione di gestione degli utenti
errorUserUpdate.jsp	Visualizza la pagina di errore per un errore riscontrato durante la modifica di un utente.

2.1.5 functions



adminNav.jsp	Visualizza la barra di navigazione dell'amministratore
assistenzaNav.jsp	Visualizza la barra di navigazione dell'assistenza
loggedNav.jsp	Visualizza la barra di navigazione del cliente
login.jsp	Visualizza la pagina di login
operatorNav.jsp	Visualizza la barra di navigazione dell'operatore
userNav.jsp	Visualizza la barra di navigazione di un utente frontend
userNoIndexNav.jsp	Visualizza la barra di navigazione di un utente frontend, nella sezione shop

3. Design Patterns

Warehouse fa uso del Bridge Pattern, perché abbiamo bisogno di utilizzare un'unica interfaccia per diversi accessi allo storage: offrendo un'unica interfaccia si garantisce che l'eventuale cambio di implementazione del database usato comporta la modifica solo a una componente e non a svariate componenti del sistema.