



Laurea Triennale in informatica - Università di Salerno
Corso di *Ingegneria del software*- Prof.ssa F. Ferrucci



Object Design Document

YourBook - YB

Riferimento	
Versione	0.00.007
Data	13/02/2021
Destinatario	Prof.ssa F.Ferrucci
Presentato da	D'Urso Serena (0512105709) Petta Paolo (0512105013), Amaro Riccardo (0512105841)
Approvato da	



Revision History

Data	Versione	Descrizione	Autori
04/01/2021	0.00.001	Creazione documento	S.D'Urso
05/01/2021	0.00.002	Aggiunta dal punto 2.1.3.3	S.D'urso
05/01/2021	0.00.003	Aggiunta dal punto 1	R.Amaro
05/01/2021	0.00.004	Aggiunta punto 2 fino a 2.3.3.2	P. Petta
06/01/2021	0.00.005	Concluso le descrizioni dei punti dal 2 fino al 2.3.3.2	P. Petta
19/01/2021	0.00.006	Modifica dei diagrammi presenti	S. D'Urso
13/02/2021	0.00.007	Modifica template	S.D'Urso



1. Introduzione	4
1.1 Object Design Trade-offs	4
1.2 Linee Guida per la Documentazione delle Interfacce	4
1.3 Definizioni, acronimi e abbreviazioni	6
1.4 Riferimenti	6
2. Packages	6
2.1 Packages core	7
2.1.1 Packages model	7
2.1.2 Packages Manager	8
2.1.3 Packages Control	9
2.1.3.0 Servlet	9
2.1.3.1 Servlet Admin	10
2.1.3.2 Servlet Utente	10
2.1.4 Package Exception	11
2.1.5 Package View	12
2.1.5.1 Errors	14
2.1.5.2 Functions	15
3. Design Patterns	16



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 semplificano 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.

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 `updateNomeVariabile()`.

Esempio: `getBook()`, `updateBook()`

- 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

- Libro: Object-Oriented Software Engineering (Using UML, Patterns, and Java) Third Edition;
- Autori: Bernd Bruegge & Allen H. Dutoit;
- Documenti SDD e RAD del progetto YourBook;

2. Packages

La gestione del nostro sistema è suddivisa in tre livelli:

- Interface layer
- Application Logic layer
- Storage layer

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.
------------------------	---

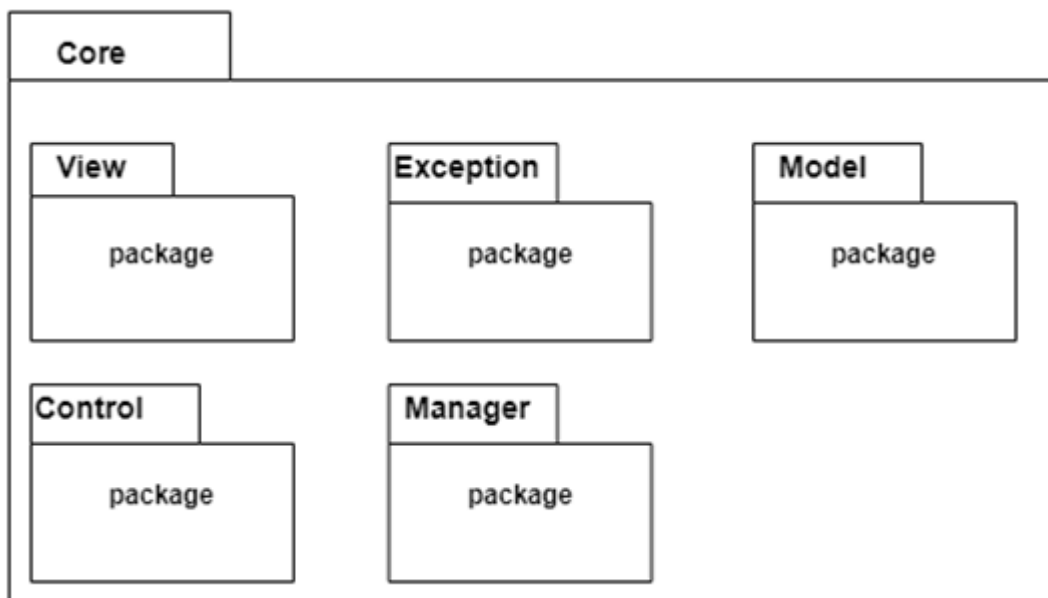
Application Logic Layer	Ha il compito di elaborare i dati da inviare al client, e tramite lo Storage Layer, accede ai dati persistenti (database). Si occupa di varie gestioni del sistema: <ol style="list-style-type: none">1. Gestione Utente2. Gestione Libri3. Gestione Wishlist
--------------------------------	---



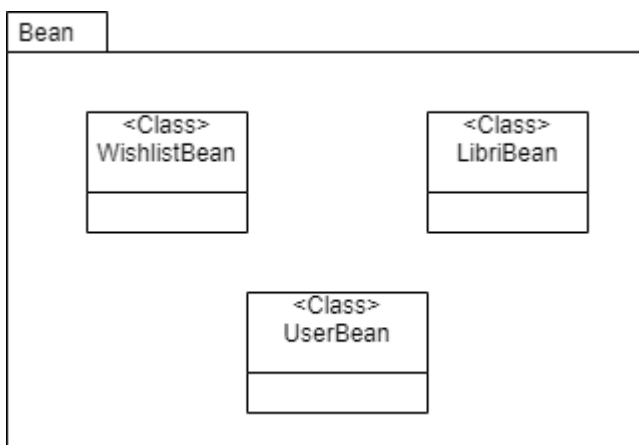
Storage Layer

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

2.1 Packages core



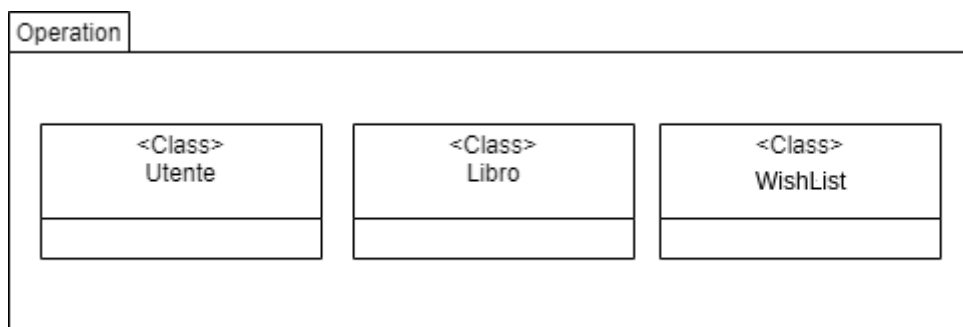
2.1.1 Packages model





Classe	Descrizione
WishlistBean	Descrive la wishlist e gli oggetti contenuti
LibriBean	Descrive gli oggetti contenuti nel DB
UserBean	Descrive gli utenti registrati al sistema

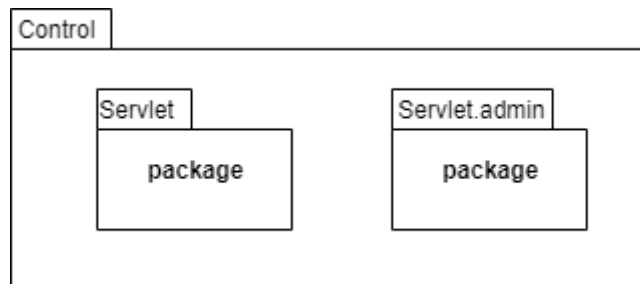
2.1.2 Packages Manager



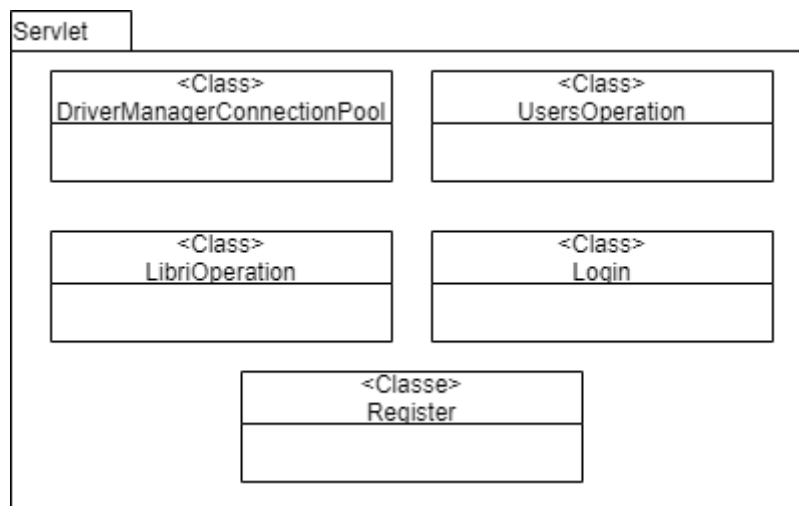
Classe	Descrizione
Utente	Permette ad un utente di inserire, modificare, cancellare e visualizzare il proprio profilo
Libro	Permette la gestione dei libri come l'inserimento, cancellazione e ricerca
WishList	Permette la gestione delle WishList come l'inserimento e cancellazione di libri in esse



2.1.3 Packages Control



2.1.3.0 Servlet

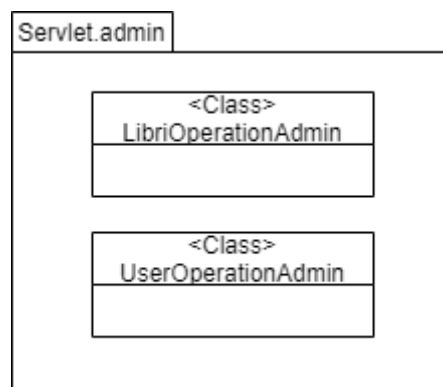


Classe	Descrizione
DriverManagerConnectionPool	Gestisce la connessione con il DB
LibriOperation	Gestisce le operazioni con i libri, lato cliente, tra cui la visualizzazione della sezione shop, l'inserimento di un prodotto nel carrello, la ricerca dei prodotti
UsersOperation	Gestisce le operazioni riguardanti gli utenti nel sistema, lato cliente, tra cui la modifica e la visualizzazione dei propri dati
WishListOperation	Gestisce le operazioni riguardanti le WishList nel sistema, lato cliente, tra cui la modifica e la creazione



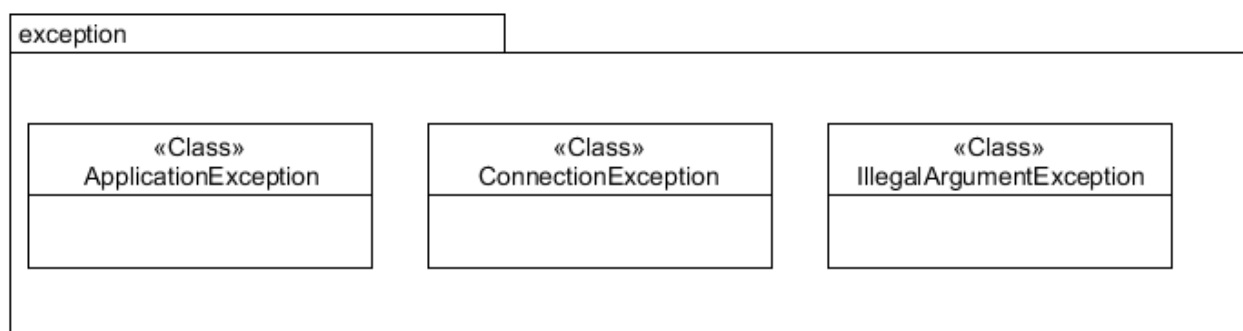
Login	Gestisce il login degli utenti e degli admin
Register	Gestisce la registrazione dei nuovi utenti

2.1.3.1 Servlet Admin



Classe	Descrizione
LibriOperationsAdmin	Gestisce le operazioni relative ai titoli, lato admin
UserOperationsAdmin	Gestisce le operazioni riguardanti gli utenti, lato admin

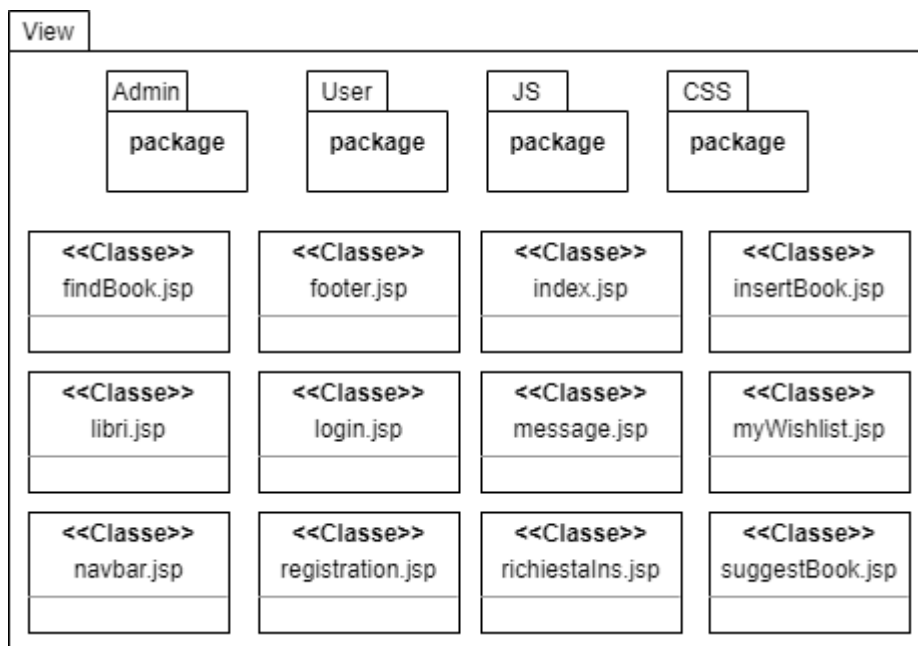
2.1.4 Package Exception





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 Package View



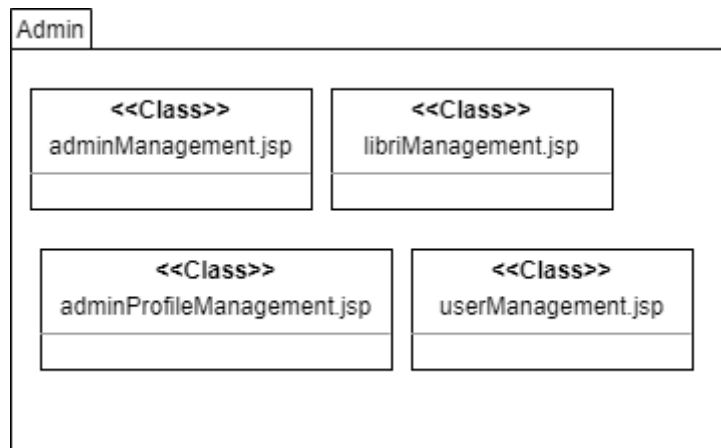
Classe	Descrizione
index.jsp	Visualizza la pagina principale del sito
myaccount.jsp	Visualizza la pagina che permette di modificare e visualizzare i propri dati personali.



myWishList.jsp	Visualizza la pagina che permette di visualizzare i propri titoli piaciuti
registration.jsp	Visualizza la pagina che permette di registrarsi al sistema
findBook.jsp	Visualizza la pagina per trovare il libro adatto
insertBook.jsp	Visualizza la pagina per inserire un titolo già letto
footer.jsp	Footer da includere nelle altre pagine
libri.jsp	Mostra tutti i libri
login.jsp	Permette di loggarsi
navbar.jsp	Navbar da includere nelle altre pagine
richiestaIns.jsp	Permette di effettuare una richiesta inserimento libri all'admin
suggestBook.jsp	Mostra i libri suggeriti

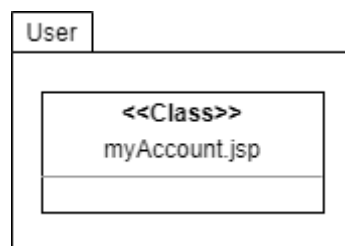


2.1.5.1 Admin



Classe	Descrizione
AdminManagement.jsp	Visualizza la pagina dashboard dell'admin con i dati statistici
libriManagement.jsp	Visualizza la pagina di gestione dei libri
adminProfileManagement.jsp	Visualizza la pagina di gestione profili admin
userManagement.jsp	Visualizza la pagina di gestione degli utenti

2.1.5.3 User



Classe	Descrizione
myAccount.jsp	Visualizza le info dell'utente



3. Design Patterns

YourBook 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.