

O.D.D.

OBJECT DESIGN DOCUMENT

Ingegneria del Software



Anno Accademico 2017/2018

GRUPPO:

Gaeta Francesco	05121 03566
Manzo Alessandro	05121 03628
Moliterni Marco	05121 03906
Valente Carmine	05121 03614

Sommario:

1. Panoramica	3
2. Linee guida per i criteri del sistema	3
2.1. Comprensibilità vs Tempo:	3
2.2. Interfaccia vs Usabilità:	3
2.3. Sicurezza vs Efficienza:	3
2.4. Prestazioni vs Costi	3
3. Linee guida per la documentazione delle interfacce	4
4. Definizione, acronimi e abbreviazioni	5
5. Pacchetti	6
5.1. Pacchetti beans	7
5.2. Pacchetti controller	8
5.2.1. Pacchetto registrazione	8
5.2.2. Pacchetto autenticazione	9
5.2.3. Pacchetto utente	9
5.2.4. Pacchetto visualizzazione	10
5.2.5. Pacchetto prodotti	11
5.2.6. Pacchetto preferiti	11
5.3. Pacchetti tests	12
5.4. Pacchetti connessione	13
5.5. Pacchetti classiDao	14
5.6. Pacchetti view	15
6. Interfaccia delle classi	16
6.1. Amministratore	16
6.2. Lezioni	16
6.3. Appunti	17
6.4. Libri	17
6.5. Utente	18

1. Panoramica

Dopo la realizzazione dei documenti RAD e SDD abbiamo descritto in linea di massima quello che sarà il nostro sistema e quindi i nostri obiettivi, tralasciando gli aspetti implementativi. 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 le signature dei sottosistemi definiti nel System Design. Inoltre, sono specificati i trade-off e le linee guida.

2. Linee guida per i criteri del sistema

2.1. 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.

2.2. 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 a quanti più utenti possibili.

2.3. Sicurezza vs Efficienza:

La sicurezza, come descritto nei requisiti non funzionali del RAD, 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.

Response Time vs Hardware:

Il sistema garantisce una certa reattività alle richieste, e quindi essere in grado di poter comunque offrire una contemporaneità di servizi agli utenti. Ovviamente questa caratteristica

sarà limitata dall'hardware del sistema.

2.4. Prestazioni vs Costi

Il sistema prevede l'utilizzo di template open source esterni per mantenere prestazioni elevate, essendo il progetto sprovvisto di budget.

3. Linee guida per la documentazione delle interfacce

Gli sviluppatori dovranno seguire le seguenti convenzioni per la scrittura del codice:

Convenzioni di denominazione

È buona norma utilizzare nomi:

- Descrittivi
- Pronunciabili
- Di uso comune
- Lunghezza medio-corta
- Non abbreviati
- Evitando la notazione ungara
- Utilizzando solo caratteri consentiti (a-z,A-Z,0-9)

Variabili e costanti

I nomi delle variabili devono cominciare con una lettera minuscola (es: prodotto). Se il nome della variabile è costituito da più parole, solo l'iniziale delle altre parole sarà maiuscola (es: codiceFiscale), inoltre è possibile abbreviare il nome della variabile solo se non peggiora la leggibilità e comprensibilità (es: numArticoli invece di numeroArticoli).

Le costanti dovranno essere scritte interamente in maiuscolo e se il nome costituito da più parole vengono separate con l'underscore (es: PI_GRECO).

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()`.

I commenti dei metodi devono essere raggruppati in base alla loro funzionalità, 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 e pagine

I nomi delle classi e delle pagine devono iniziare con una lettera maiuscola, le parole contenute al suo interno devono cominciare con lettera maiuscola.

Il nome deve fornire informazioni utili relative al loro scopo.

Ogni file sorgente contiene una singola classe e deve essere strutturato in un determinato modo:

- L'istruzione `package` che permette di inserire la classe in un determinato package.
- L'istruzione `import` che importa le librerie necessarie alla classe.

La dichiarazione di una classe deve essere caratterizzata da:

- Dichiarazione della classe pubblica
- Costruttore
- Metodi
- Dichiarazione di costanti e variabili, di classe e di istanza.

Esempio

```

public class UtenteBeans {

    public UtenteBeans() {

    }

    public UtenteBeans(String psw, int matricola, String nome, String cognome, String
facolta, String universita, String email, String citta) {
        this.login = false;
        this.errorLogin = false;
        this.errorCart = false;
        this.psw = psw;
        this.matricola = matricola;
        this.nome = nome;
        this.cognome = cognome;
        this.facolta = facolta;
        this.universita = universita;
        this.email = email;
        this.citta = citta;
        appunti = new ArrayList<AppuntiBeans>();
        libri = new ArrayList<LibroBeans>();
        lezioni = new ArrayList<LezioneBeans>();
    }

    public String getNome() {
        return nome;
    }

    public void setNome(String nome) {
        this.nome = nome;
    }
}

```

4. Definizione, acronimi e abbreviazioni

Gli sviluppatori dovranno seguire le seguenti convenzioni per gli acronimi e per le abbreviazioni:

Acronimi

- RAD: Requirements Analysis Document
- SDD: System Design Document
- ODD: Object Design Document
- DBMS: Database Management System
- API: Application Programming Interface
- JDBC: Java Database Connectivity

Abbreviazioni

- DB: DataBase

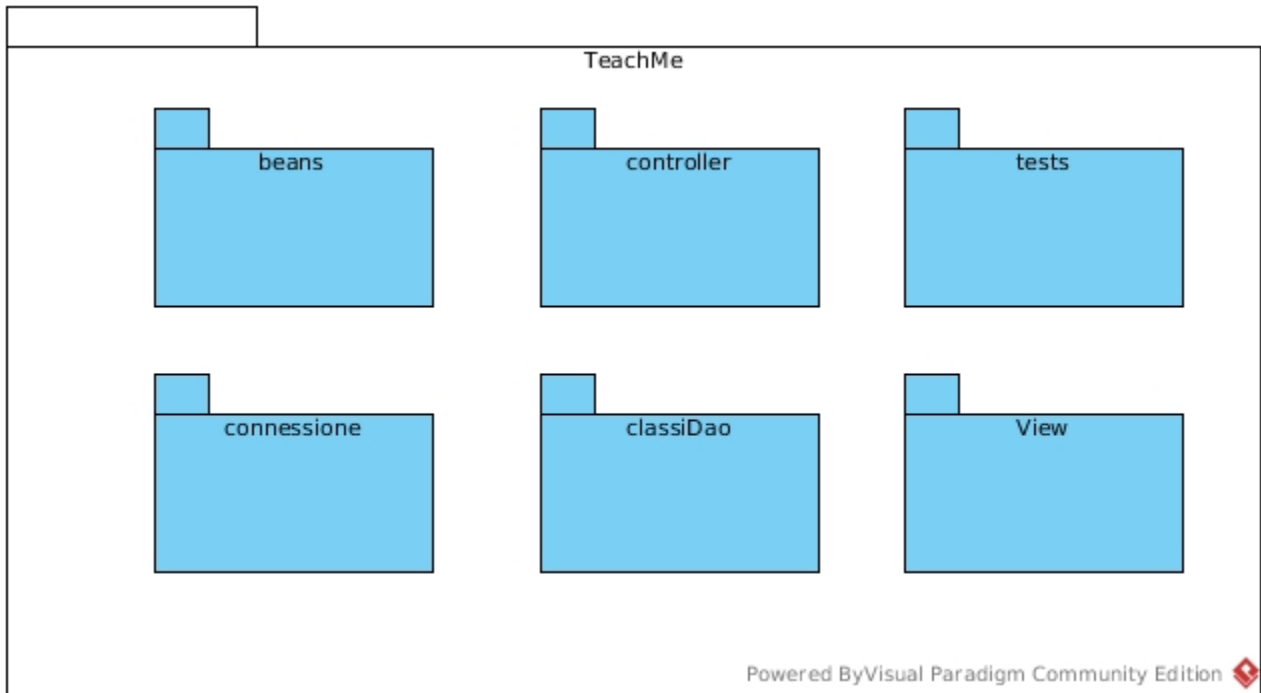
5. Pacchetti

Il sistema sarà diviso in tre livelli:

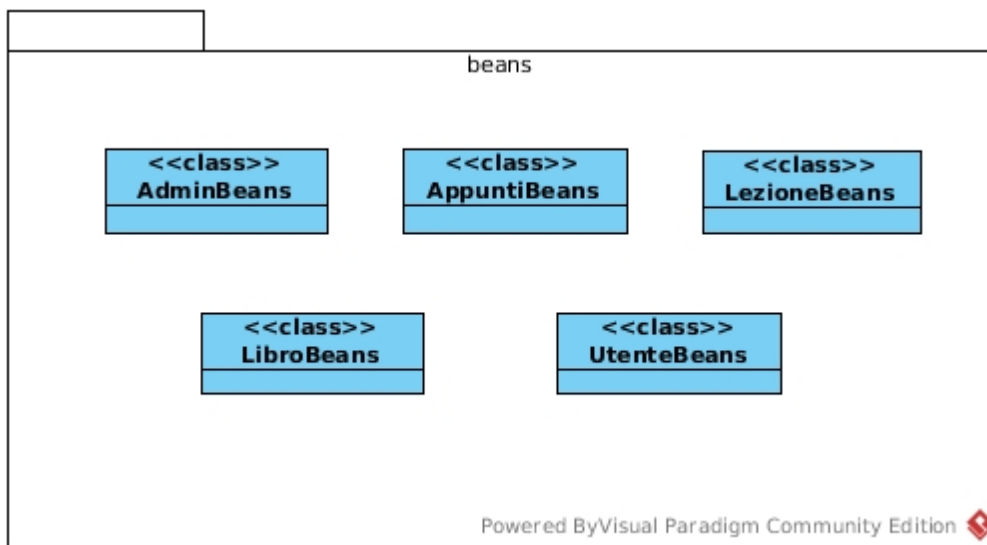
- Interface Layer
- Application Logic Layer
- Storage Layer

I sottopackage del sistema contengono classi atte allo svolgimento delle richieste pervenute dall'utente, sia per gestirne la logica sia per offrire risposte sotto forma di interfaccia grafica.

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	<p>Si occupa dell'elaborazione dei dati da inviare al client.</p> <p>Si occupa di varie gestioni quali:</p> <ol style="list-style-type: none"> 1. Gestione Prodotti 2. Gestione Preferiti 3. Gestione Autenticazione 4. Gestione Utente 5. Gestione Registrazione 6. Gestione Visualizzazione
Storage Layer	Ha il compito di memorizzare i dati sensibili del sistema, utilizzando un DBMS che gestisce i dati. Inoltre, lo Storage Layer riceve le varie richieste dall' Application Logic Layer inoltrandole al DBMS e restituendo i dati richiesti. La comunicazione verso il DBMS è realizzata tramite le API JDBC.

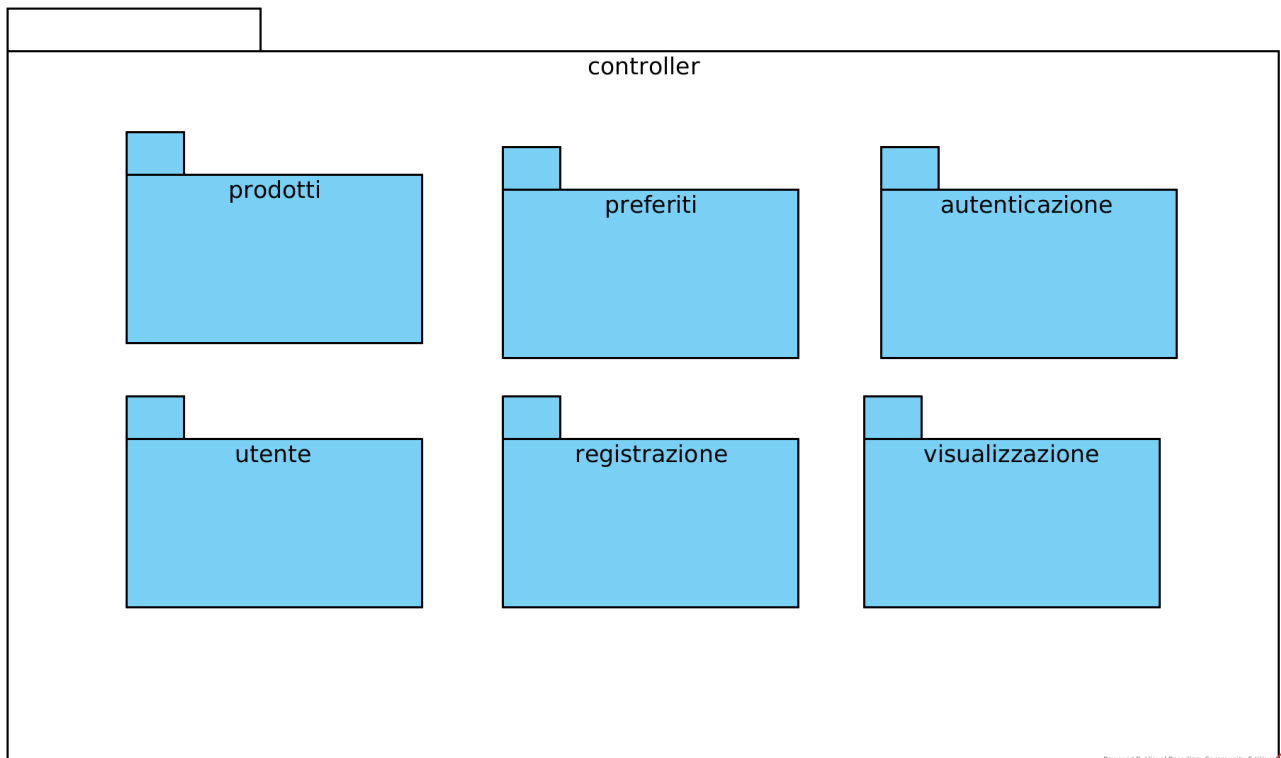


5.1. Pacchetti beans



Class	Descrizione
AdminBeans.java	Descrive l'amministratore del sistema
AppuntiBeans.java	Descrive gli appunti presenti nel sistema
LezioneBeans.java	Descrive le lezioni presenti nel sistema
LibroBeans.java	Descrive i libri presenti nel sistema
UtenteBeans.java	Descrive l'utente

5.2. Pacchetti controller

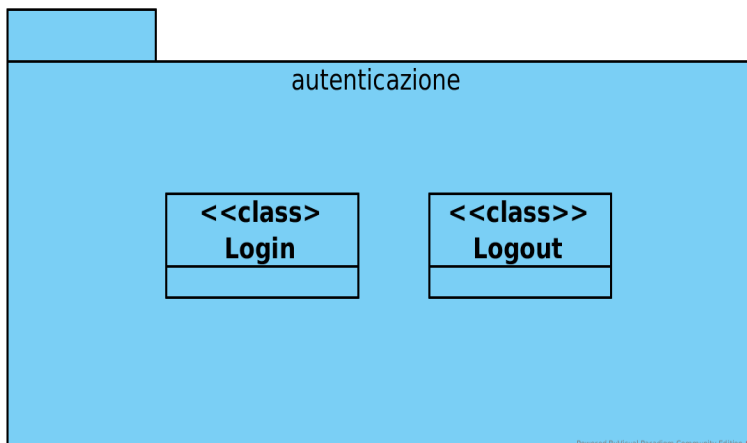


5.2.1. Pacchetto registrazione



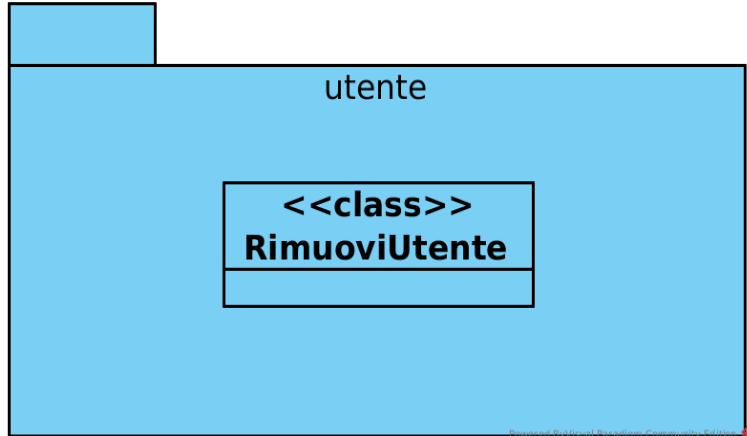
Classe	Descrizione
Registrazione.java	Servlet che gestisce la registrazione

5.2.2. Pacchetto autenticazione



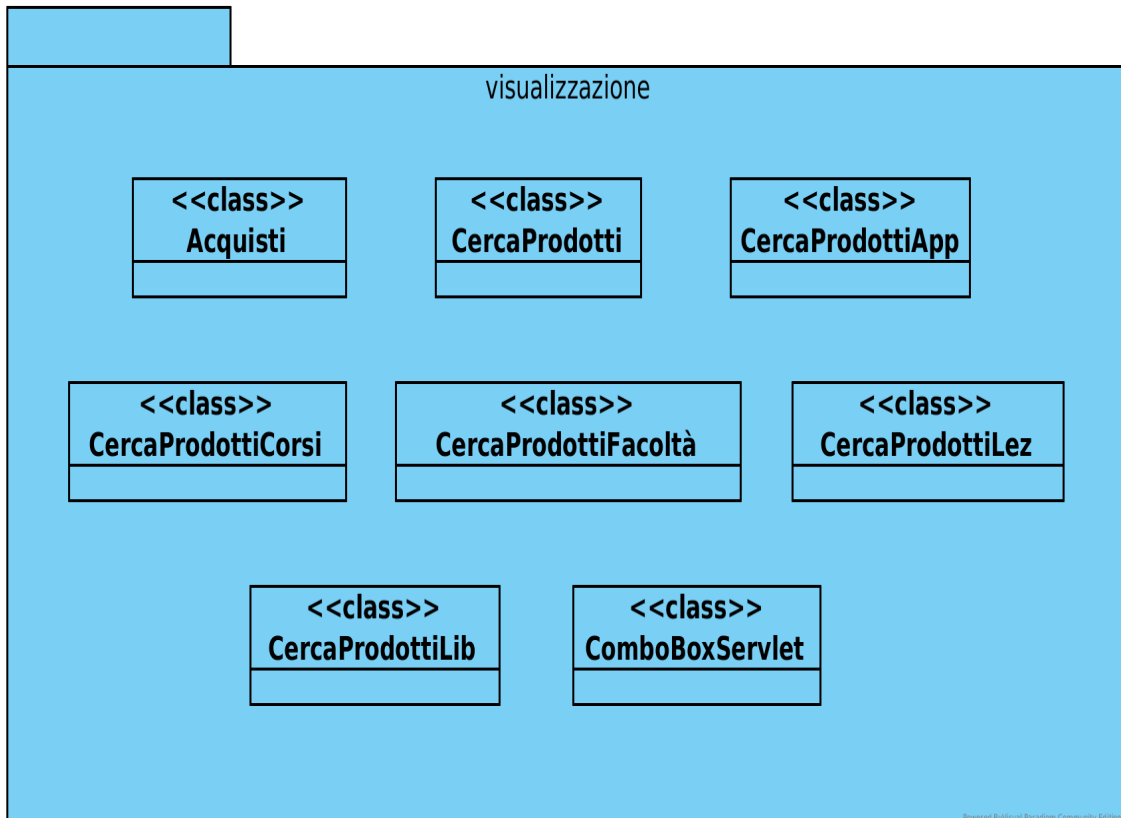
Classi	Descrizione
Login.java	Servlet che gestisce l'autenticazione
Logout.java	Servlet che gestisce l'uscita

5.2.3. Pacchetto utente



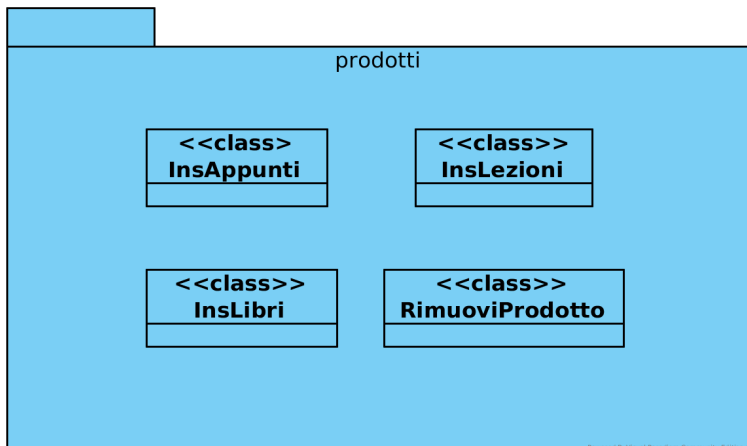
Classi	Descrizione
RimuoviUtente.java	Servlet che gestisce l'operazione dell'admin

5.2.4. Pacchetto visualizzazione



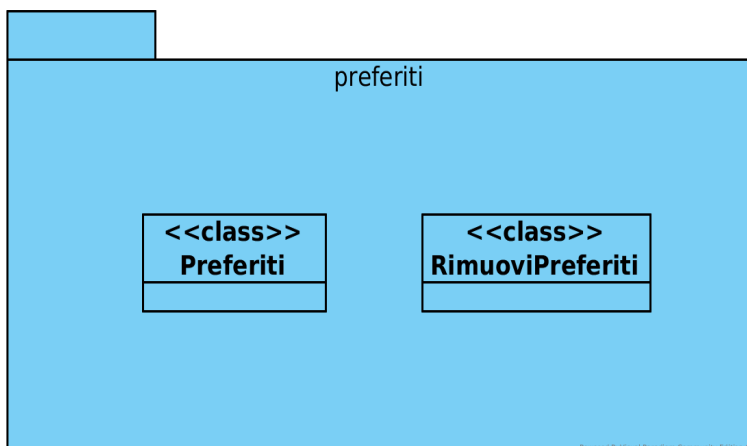
Classe	Descrizione
Acquisti.java	Servlet per a gestione degli acquisti
CercaProdotti.java	Servlet per la gestione della ricerca dei prodotti
CercaProdottiApp.java	Servlet per la gestione della ricerca degli appunti
CercaProdottiCorsi.java	Servlet per la gestione della ricerca per corsi
CercaProdottiFacoltà.java	Servlet per la gestione della ricerca per facoltà
CercaProdottiLez.java	Servlet per la gestione della ricerca delle lezioni
CercaProdottiLib.java	Servlet per le gestione della ricerca dei libri
ComboBoxServlet	Servlet per la gestione della facilitata della ricerca

5.2.5. Pacchetto prodotti



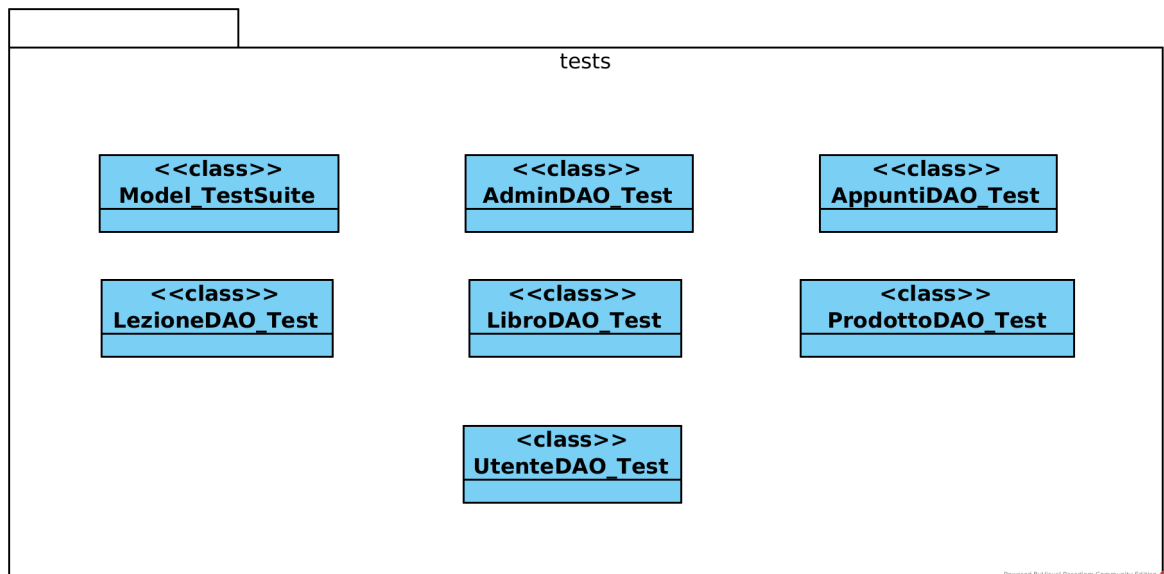
Classi	Descrizione
InsAppunti.java	Servlet che gestisce l'inserimento di appunti
InsLezioni.java	Servlet che gestisce l'inserimento di lezioni
InsLibri.java	Servlet che gestisce l'inserimento di libri
RimuoviProdotto.java	Servlet che gestisce la rimozione dei prodotti

5.2.6. Pacchetto preferiti



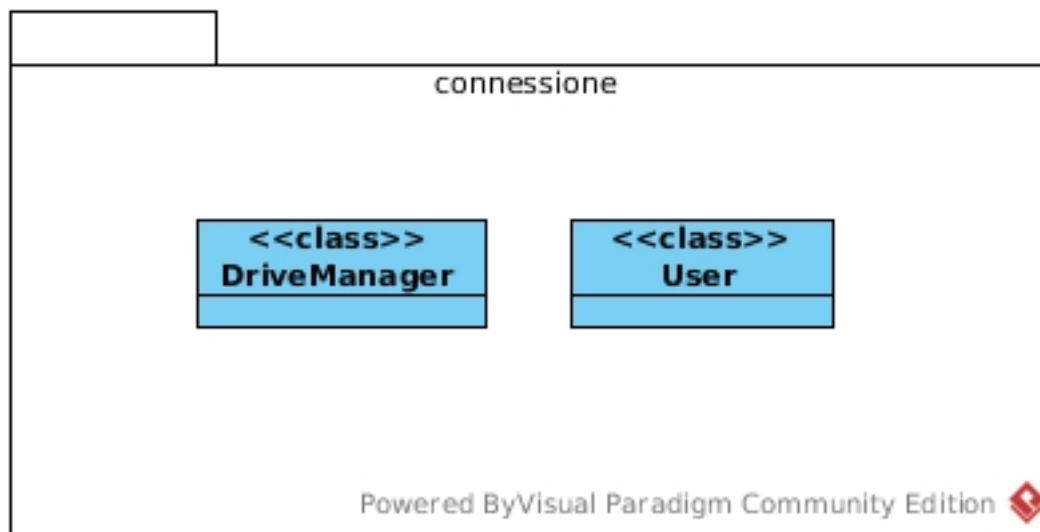
Classi	Descrizione
Preferiti.java	Servlet che gestisce i prodotti aggiunti ai preferiti
RimuoviPreferiti.java	Servlet che gestisce la rimozione dai preferiti

5.3. Pacchetti tests



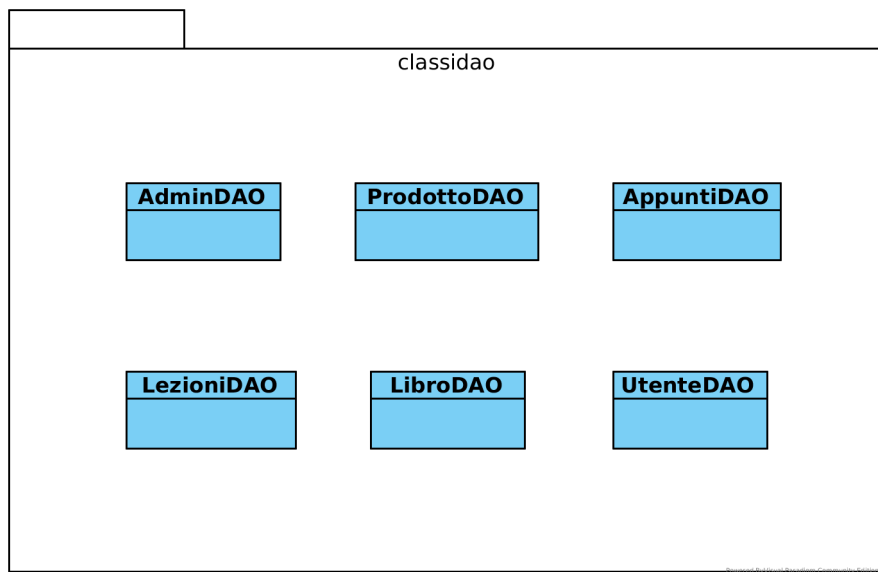
Classe	Descrizione
Model_TestSuite.java	Test Suite che include tutte le classi di test per il model
AdminDAO_test.java	Classe di test per AmministratoreDAO
AppuntiDAO_test.java	Classe di test per AppuntiDAO
LezioneDAO_test.java	Classe di test per LezioneDAO
LibroDAO_test.java	Classe di test per LibroDAO
ProdottoDAO_test.java	Classe di test per ProdottoDAO
UtenteDAO_test.java	Classe di test per UtenteDAO

5.4. Pacchetti connessione



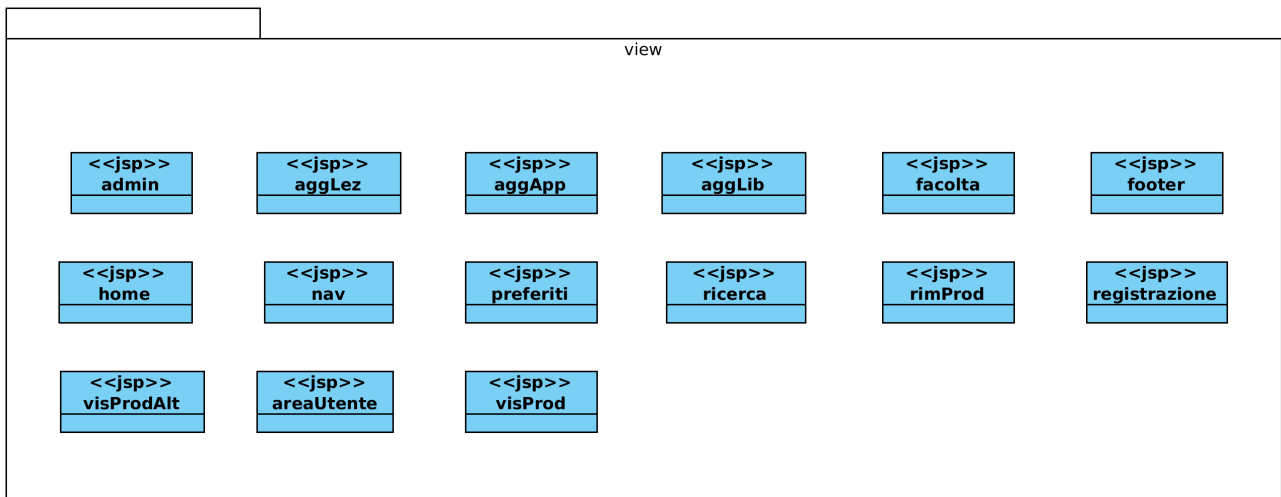
Classi	Descrizione
DriveManager.java	Serve per connettere la piattaforma al database
User.java	superflua

5.5. Pacchetti classiDao



Classe	Descrizione
AdminDAO.java	Descrive l'accesso al database relativo all'oggetto admin
ProdottoDAO.java	Descrive l'accesso al database relativo ai prodotti
AppuntiDAO.java	Descrive l'accesso al database relativo all'oggetto appunti
LezioneDAO.java	Descrive l'accesso al database relativo all'oggetto lezione
LibroDAO.java	Descrive l'accesso al database relativo all'oggetto libro
UtenteDAO.java	Descrive l'accesso al database relativo all'oggetto utente

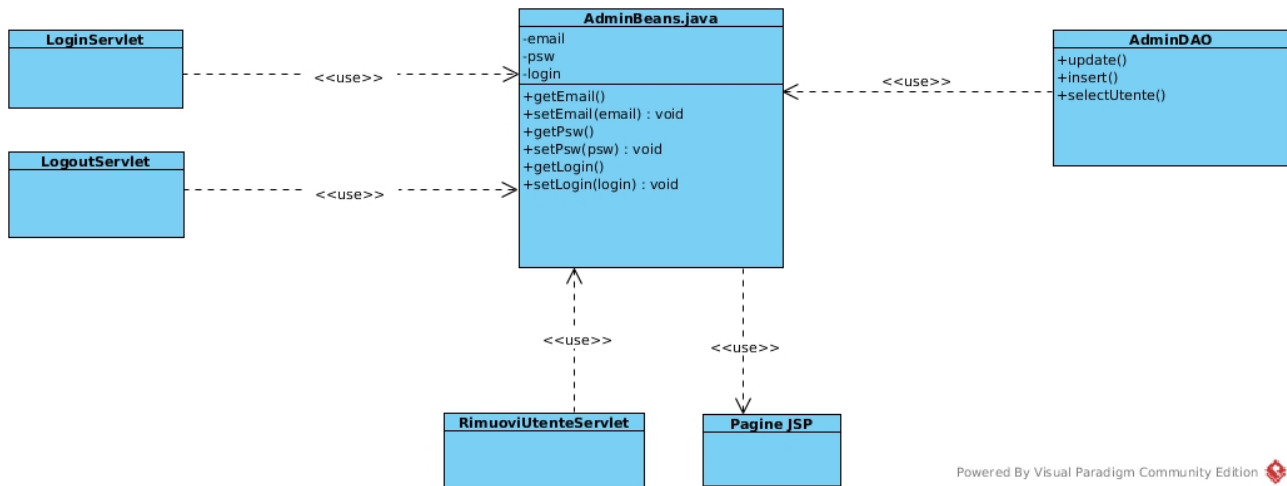
5.6. Pacchetti view



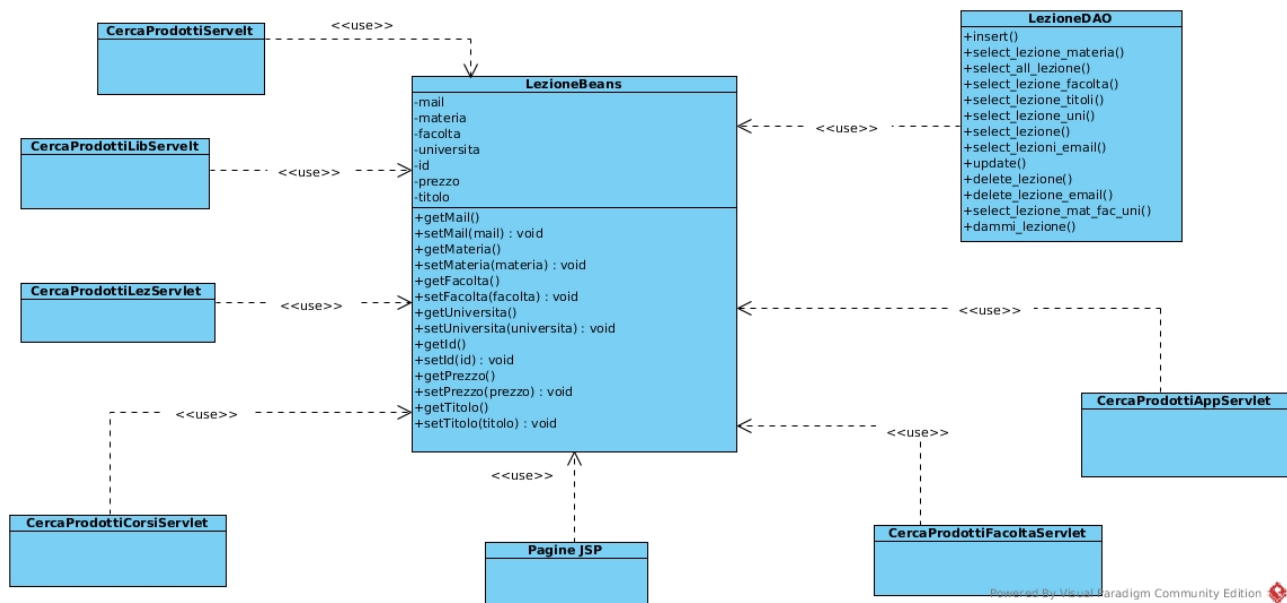
Classi	Descrizione
Admin.jsp	Pagina che permette all'admin la rimozione degli utenti
AggLez.jsp	Pagina che permette agli utenti di aggiungere le lezioni
AggApp.jsp	Pagina che permette agli utenti di aggiungere gli appunti
AggLib.jsp	Pagina che permette agli utenti di aggiungere i libri
Facolta.jsp	Pagina per la visualizzazione per facoltà dei prodotti
Footer.jsp	Permette di visualizzare il footer in tutte le pagine
Home.jsp	Pagina per la visualizzazione della homepage
Nav.jsp	Pagina che permette di visualizzare la nav
Preferiti.jsp	Pagina per la visualizzazione dei preferiti
Ricerca.jsp	Pagina che permette la ricerca dei prodotti
RimProd.jsp	Pagina per la rimozione dei prodotti
Registrazione.jsp	Pagina per la registrazione utente
VisProdAlt.jsp	Pagina per la visualizzazione dei prodotti
AreaUtente.jsp	Pagina che permette la visualizzazione dei dati utente
VisProd.jsp	Pagina per la visualizzazione dei prodotti

6. Interfaccia delle classi

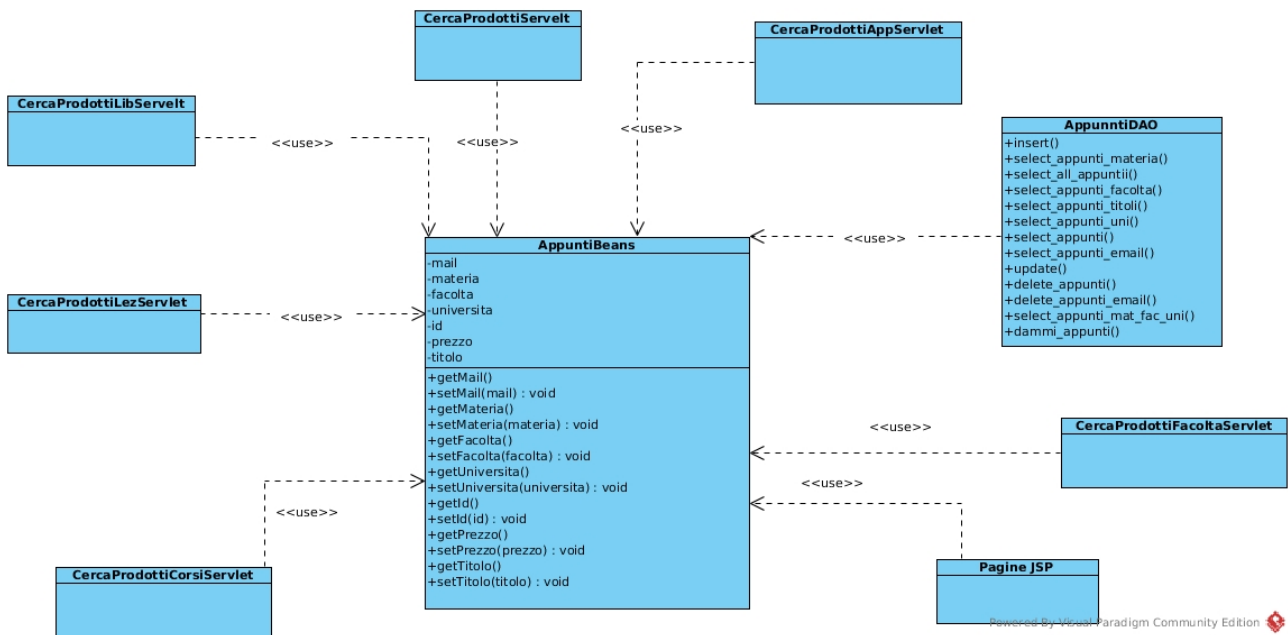
6.1. Amministratore



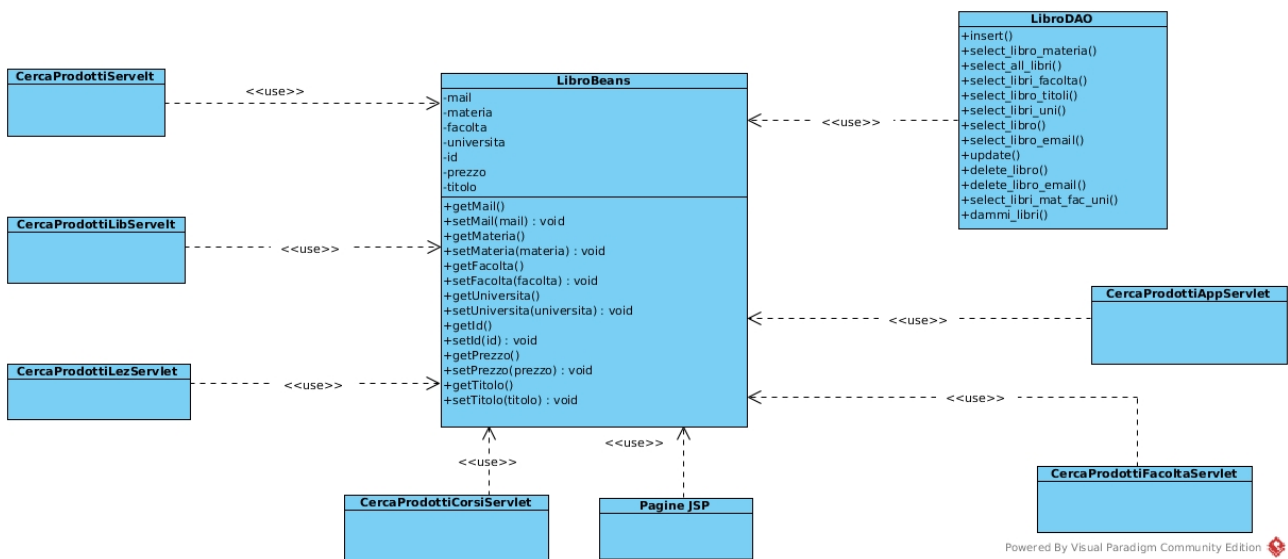
6.2. Lezioni



6.3. Appunti



6.4. Libri



6.5. Utente

