**Università degli Studi di Salerno  
Corso di Ingegneria del Software**

****

**urNotes  
ODD**

**Object Design Document**

****

Data: 18/12/2018

**Coordinatore del progetto:**

|  |  |
| --- | --- |
| Nome | Matricola |
| Francesco Parisi | 0512103582 |

**Partecipanti:**

|  |  |
| --- | --- |
| Nome | Matricola |
| Marco La Cortiglia | 0512103630 |
| Matteo Maiorano | 0512104578 |

**Revision History**

|  |  |  |  |
| --- | --- | --- | --- |
| Data | Versione | Descrizione | Autore |
| 18/12/2018 | 0.1 | Aggiunta Packages,Sottosistemi e Revisione Organizzazione Package | Francesco Parisi |
| 29/12/2018 | 0.2 | Completamento e revisione documento | Tutti |
|  |  |  |  |

Indice

1. Introduzione ……………....................................................................................................................................................................4

1.1 Object design trade-offs……………........................................................................................................................................4

1.2 Linee guida per la documentazione dell'Interfaccia…………….....................................................................................5

1.3 Definizioni, Acronimi e Abbreviazioni……………................................................................................................................8

1.4 Riferimenti……………..................................................................................................................................................................8

2. Packages…………….............................................................................................................................................................................9

2.1 Package Bean……………...................................................................................................................................................... 10

2.2 Package Connection...................................................................................................................................................... 11

**2.3** Package Model..................................................................................................................................................................12

**2.4** Package View.....................................................................................................................................................................12

**2.4.1 Package Pubblico** ....................................................................................................................................................13

**2.4.2 Package Gestore**.......................................................................................................................................................14

**2.4.3 Package Studente**.....................................................................................................................................................15

**2.5** Package Test.......................................................................................................................................................................16

**2.5.1** Package Account......................................................................................................................................................16

**2.5.2** Package Documenti................................................................................................................................................17

**2.5.3** Package Ordini...........................................................................................................................................................20

**2.5.4** Package Bean.............................................................................................................................................................22

**2.5.5 P**ackage Model..........................................................................................................................................................23

**2.6** Package Account..............................................................................................................................................................24

**2.7** Package Documenti........................................................................................................................................................25

**2.8** Package Ordini................................................................................................................................................................. 28

3. Interfacce delle classi………….…………….…………………………………………………………………………………….…...………………………….…….30

Glossario………….…………….……………………………………………………….……………………………………………………..……………….……………….… 45

1. ***Introduzione***
   1. **Object design trade-offs**

**Comprensibilità vs costi**

Si preferisce avere un aumento dei costi per la documentazione, al fine di rendere il codice quanto più comprensibile possibile per i vari utenti: vale a dire per chi non è coinvolto nel progetto, per chi non ha lavorato in una parte in particolare, ma anche per chi lo ha fatto, in modo da agevolarne il mantenimento o un eventuale modifica e in generale per averne una comprensione migliore.

**Interfaccia vs Usabilità**

L’interfaccia grafica fa utilizzo di form e bottoni accompagnati da etichette con indicazioni brevi e concise, in modo da agevolare l’utilizzo del sistema all’utente finale, rendendo il tutto semplice, chiaro e intuitivo.

**Memoria vs efficienza**

La priorità viene data all’efficienza, poiché il sistema ha lo scopo di fornire all’utente finale una visione chiara e corretta della fase iniziale della richiesta di Tirocinio, quindi si deve avere la certezza che le operazioni effettuate, come la ricerca di offerte formative in base a determinati parametri, siano corrette. La memorizzazione dei dati ovviamente non viene a mancare, in quanto bisogna mantenere la tracciabilità di tutte le richieste ed è indispensabile il salvataggio dei profili utenti.

**Sicurezza vs efficienza**

La sicurezza è un aspetto molto importante del nostro sistema. L’autenticazione di un utente avviene, ovviamente dopo opportuna registrazione, attraverso username e password. Ogni utente può accedere solo a specifiche funzionalità del sistema, a seconda del tipo di account a cui fa capo.

* 1. **Linee guida per la documentazione dell'Interfaccia**

Ogni metodo ed ogni file devono essere preceduti da un commento, ovvero da una documentazione che riporti l’obiettivo che si vuole e deve raggiungere. Vi saranno inoltre, commenti e giustificazioni in merito a particolari decisioni o calcoli.

La prima lettera può essere sia maiuscola, come il nome delle classi in Java, che minuscola, come le proprietà di un oggetto in Java.

**Organizzazione dei file**

Ogni file deve:

o Essere sviluppato e correlato ad un’unica funzionalità che persegue. Ogni pagina deve essere implementata in file separato (Accedi, SalvaUtente, etc). All’interno dello stesso file potranno, tuttavia, essere inserite classi private associate alla classe pubblica.

o Essere diviso in più file se raggiunge una lunghezza tale da divenire difficile da leggere e comprendere.

Per quanto riguarda i nomi dei file, operazioni e variabili, si è scelto di utilizzare nomi evocativi, in lingua italiana o inglese, ma che in ogni caso risultano di facile comprensione e di immediata intuizione. I termini GET e SET, ad esempio, non sono stati tradotti in quanto la traduzione in italiano non regge il confronto. Di conseguenza viene inserito il rispettivo significato nella terminologia.

I file delle librerie usate e altre risorse utilizzate per lo sviluppo del processo verranno organizzati nelle varie cartelle.

In generale, i file hanno la seguente struttura:

**Commento iniziale**: un commento che specifichi il nome della classe.

**Package e Import**: le prime linee di codice specificano in quale package il file è contenuto e le righe successive possono contenere direttive di import per l’utilizzo di classi non contenute dal package corrente.

**Descrizione della classe**: la dichiarazione della classe è preceduta da un commento più o meno breve che descrive approfonditamente le funzionalità delle classi, le metodologie utilizzate, l’autore della classe e la versione. Il commento fa utilizzo delle annotazioni javadoc.

**Dichiarazione della classe (o dell’interfaccia**): gli elementi della classe devono essere dichiarati nel modo seguente:

1. La prima istruzione deve essere la dichiarazione della classe (o interfaccia) pubblica.

2. Le variabili sono dichiarate all’inizio della classe, seguendo il seguente ordine: prima le costanti (final, static), poi le variabili di classe(static), in seguito le variabili di istanza (public, protected, private).

3. I costruttori vengono dichiarati prima dei metodi.

4. Infine, vengono dichiarati i metodi, raggruppati in base alle funzionalità e non in base alla visibilità.

**Istruzioni**

Nel caso di istruzioni semplici, ogni linea deve contenere al massimo una sola istruzione. Mentre nel caso di istruzioni composte vanno rispettate le seguenti regole:

* Le istruzioni racchiuse all’interno di un blocco (esempio: for), devono essere indentate di un’unita all’interno dell’istruzione composta.
* La parentesi di apertura del blocco deve trovarsi alla fine della riga dell’istruzione composta.
* La parentesi di chiusura del blocco deve trovarsi allo stesso livello di indentazione dell’istruzione composta
* Le istruzioni composte formate da un’unica istruzione devono essere racchiuse da parentesi.

**Spazi bianchi**

Uno spazio bianco dovrebbe essere usato nelle seguenti condizioni:

* Dopo le virgole nelle liste di argomenti.
* Tutti gli operatori binari eccetto “. “(operatore punto) dovrebbero essere separati dai loro operandi tramite spazi. Gli spazi bianchi non dovrebbero mai separare gli operatori unari come l’operatore meno, l’incremento e il decremento.

**Convenzioni di nomi**

**Classi**

I nomi di classe devono essere sostantivi, con lettere minuscole e, sia la prima lettera del nome della classe sia la prima lettera di ogni parola interna, deve essere maiuscola. I nomi delle classi dovrebbero essere semplici, descrittivi e che rispettino il dominio applicativo. Usare parole intere evitando acronimi e abbreviazioni. Non dovrebbero essere usati underscore per legare nomi.

**Metodi**

I nomi dei metodi iniziano con una lettera minuscola (non sono consentiti caratteri speciali) e seguono la notazione a cammello. Dovranno essere semplici, descrittivi e che rispettino il dominio applicativo

**Variabili**

La scelta di un nome deve essere mnemonica e di facile intuizione, rispettando il dominio applicativo. Dovrebbero essere evitati nomi di variabili con un solo carattere.

**Pagine Html**

Le pagine HTML, statiche e dinamiche, devono essere aderenti allo standard HTML5.

o Nella prima riga, va dichiarato il tipo del documento

o Per i nomi degli elementi, HTML5 consente l’uso del minuscolo e del maiuscolo. Non è buona norma mischiare il minuscolo e il maiuscolo. Gli sviluppatori sono soliti usare il minuscolo, che risulta più “pulito”, è più semplice da scrivere.

o In HTML5 non bisogna chiudere obbligatoriamente tutti gli elementi. Per consentire una lettura più chiara e scorrevole, si preferisce chiudere tutti gli elementi.

o HTML5 consente l’uso del minuscolo e del maiuscolo per gli attributi, è consigliato utilizzare il minuscolo.

o In HTML5 non sono necessari i tag principali: <html>, <body>, <head>. Tuttavia si preferisce utilizzarli, per evitare problemi di validazione del documento.

o Usare una sintassi semplice per includere fogli di stile esterni.

**Fogli di stile CSS**

Per una più chiara lettura, si consiglia di scrivere le regole CSS su più righe, In generale:

o Aprire la parentesi graffa sulla stessa riga del selettore.

o Usare un solo spazio prima della parentesi graffa.

o Usare due punti più uno spazio tra una proprietà e il suo valore.

o Usare uno spazio dopo una virgola o dopo un punto e virgola (regola che vale in qualsiasi caso).

o Chiudere le parentesi graffe su una nuova riga, senza spazi.

* 1. **Definizioni, acronimi e abbreviazioni**

|  |  |
| --- | --- |
| ***urNotes*** | ***Nome del progetto*** |
| ***RAD*** | *Requirement Analysis Document* |
| ***SDD*** | System Design Document. |
| ***ODD*** | Object Design Document |
| **DB** | *Database* |
| **MySql** | Database Open Source basato sul linguaggio SQL, composto da un client a riga di comando e un server. |
| **DBMS** | Database Management System |
| ***SQL*** | Linguaggio standardizzato per database basati sul modello relazionale |
| ***CSS*** | Linguaggio usato per definire la formattazione di pagine web. |
| ***HTML5*** | Linguaggio di markup per la strutturazione delle pagine web. |

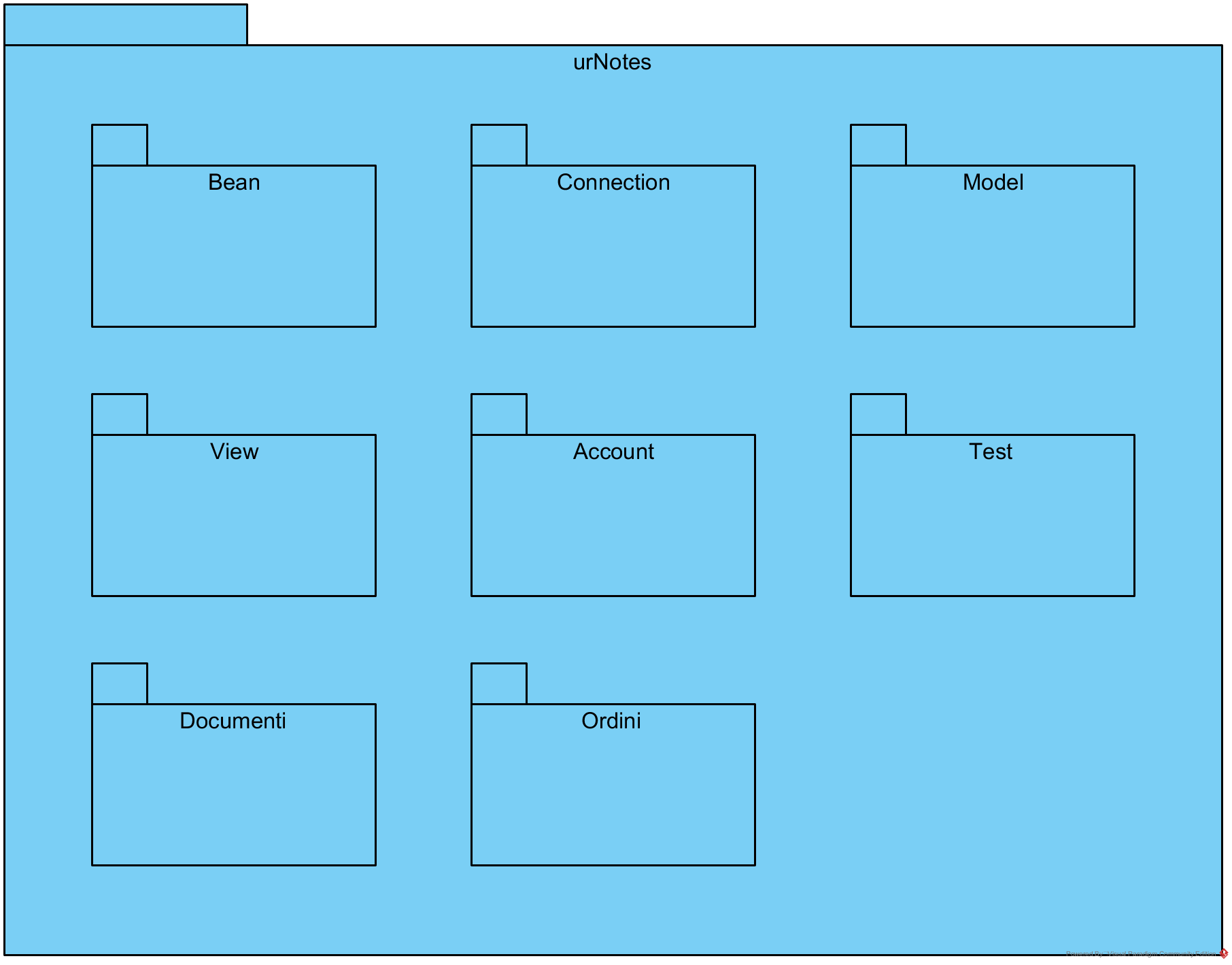
* 1. **Riferimenti**
* Slide del corso, presenti sulla piattaforma e-learning,
* B. Bruegge, A.H. Dutoit, Object Oriented Software Engineering – Using UML, Patterns and Java, Prentice Hall
* [*https://www.studocu.com/it*](https://www.studocu.com/it) *Piattaforma dedicata alla condivisione di vari documenti (Appunti, Libri, Riassunti ecc.)*
* Documenti RAD\_urNotes
* Documenti SDD\_urNotes

1. ***Packages***

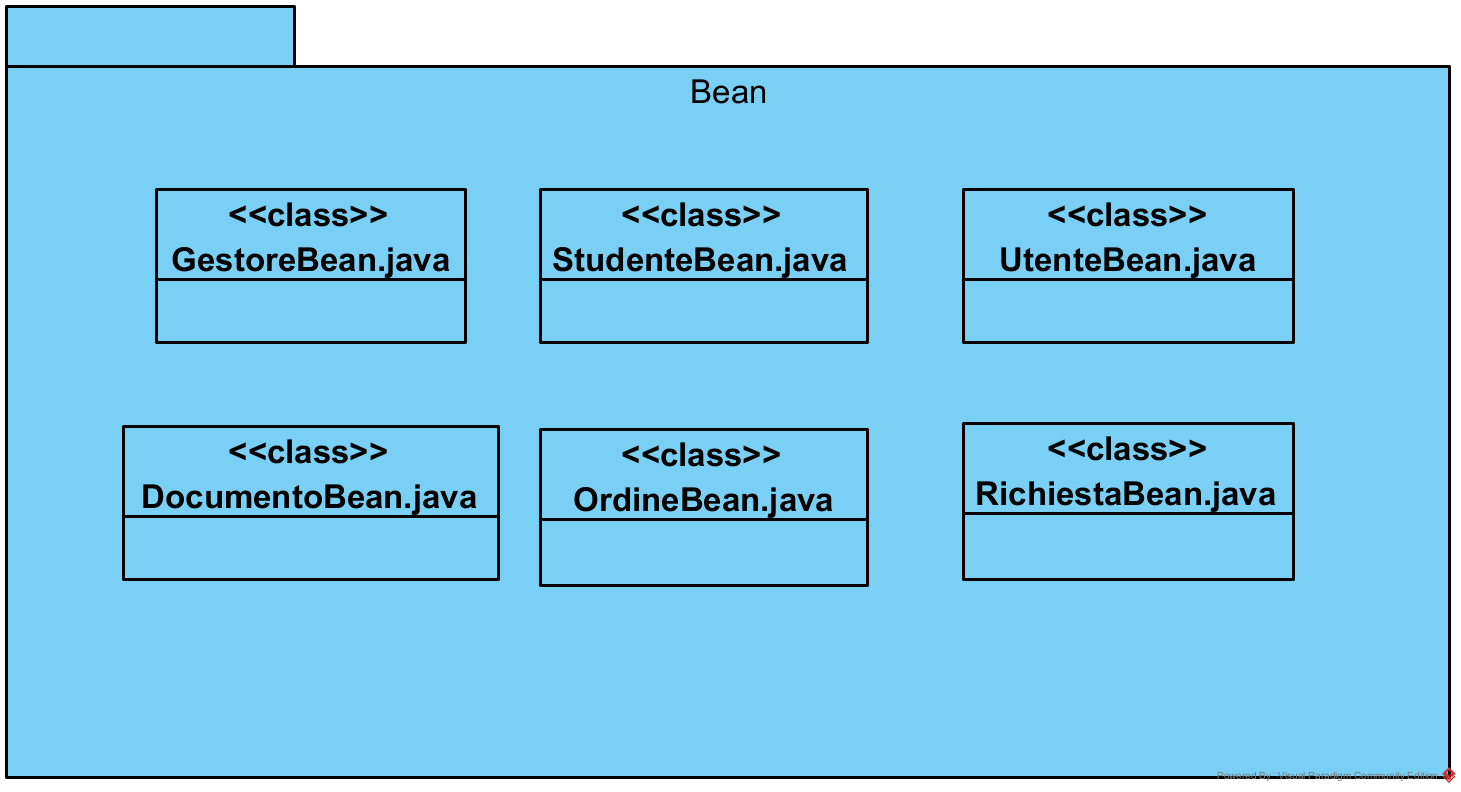
La gestione del nostro sistema è suddivisa in tre livelli (three-tier):

* Presentation Layer
* Application Logic Layer
* Storage Layer

Il package urNotes contiene sottopackage che a loro volta inglobano classi il cui scopo è quello di gestire le richieste dell’utente.

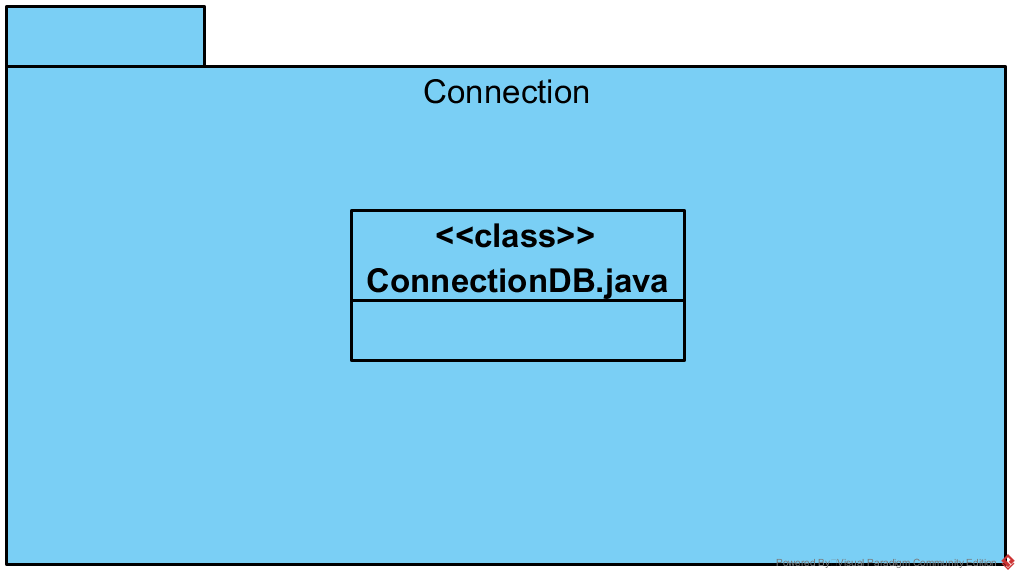


* 1. **Package Bean**

****

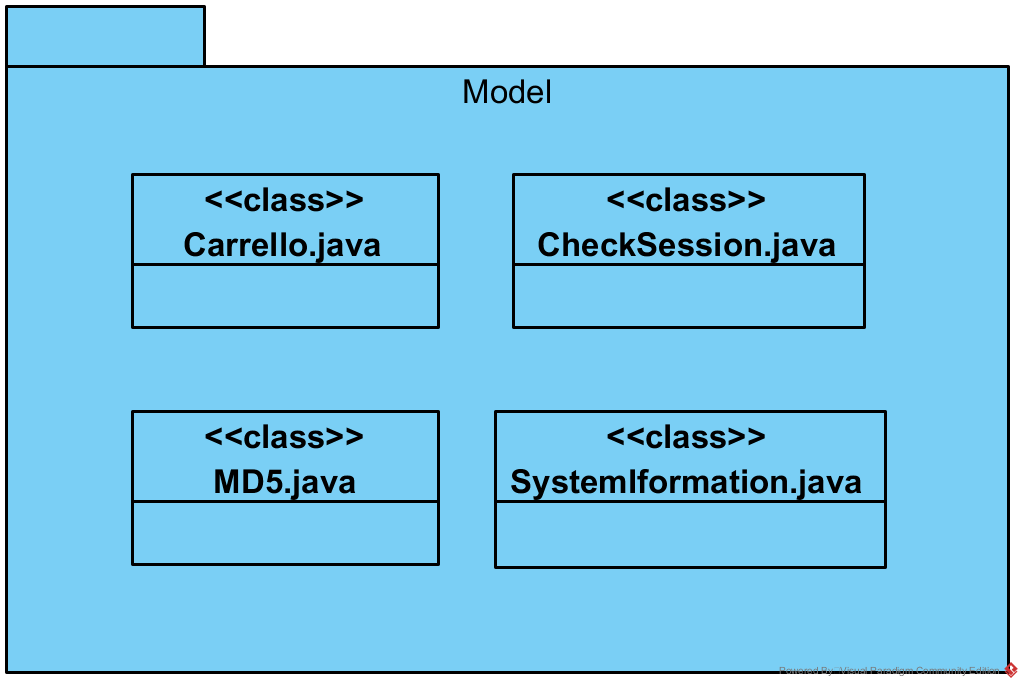
|  |  |
| --- | --- |
| Classe | Descrizione |
| GestoreBean.java | Descrive il gestore del sistema. |
| StudenteBean.java | Descrive uno studente dell’Università. |
| UtenteBean.java | Descrive un utente registrato al sito. |
| DocumentoBean.java | Descrive un appunto/dispensa. |
| OrdineBean.java | Descrive un acquisto. |
| RichiestaBean.java | Descrive una richiesta di pubblicazione di un appunto. |

* 1. **Package Connection**



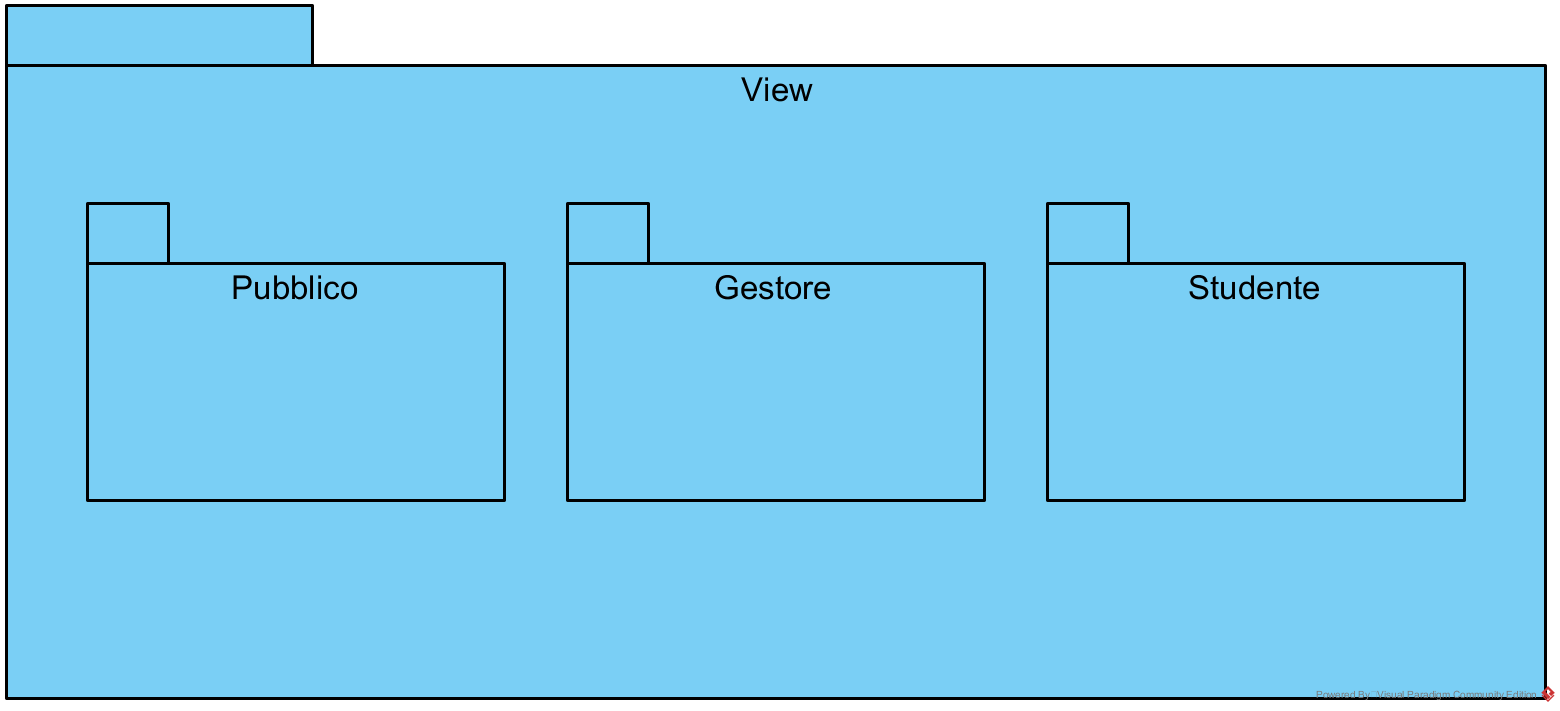
|  |  |
| --- | --- |
| Classe | Descrizione |
| ConnectionDB.java | La classe che si occupa di gestire la connesione con il DataBase. |

* 1. **Package Model**

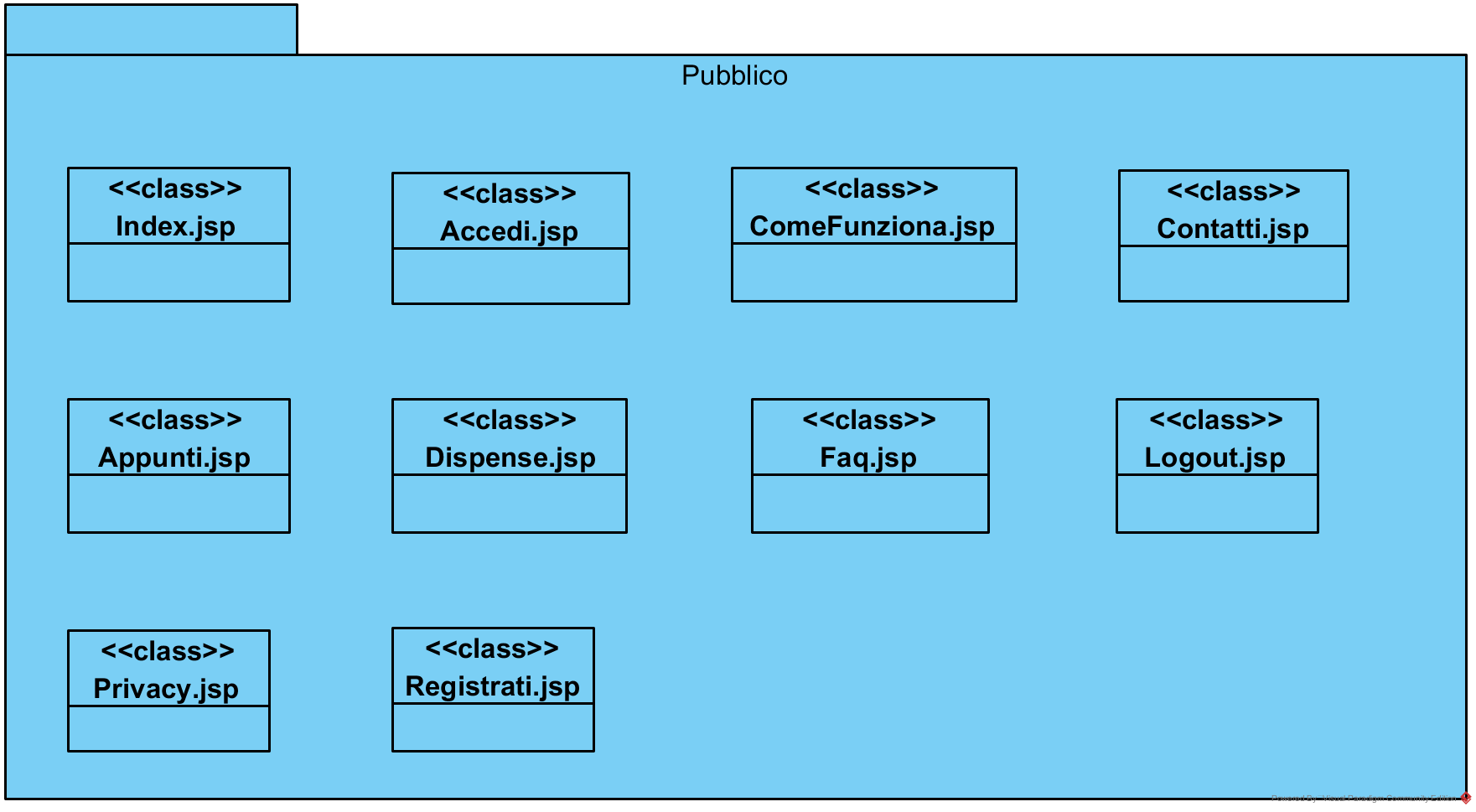


|  |  |
| --- | --- |
| Classe | Descrizione |
| Carrello.java | Descrive il carrello dello studente. |
| CheckSession.java | Descrive la sessione di un utente . |
| MD5.java | Descrive la crittografia scelta per proteggere la password di un utente . |
| SystemInformation.java | Descrive le informazioni di sistema. |

* 1. **Package View**

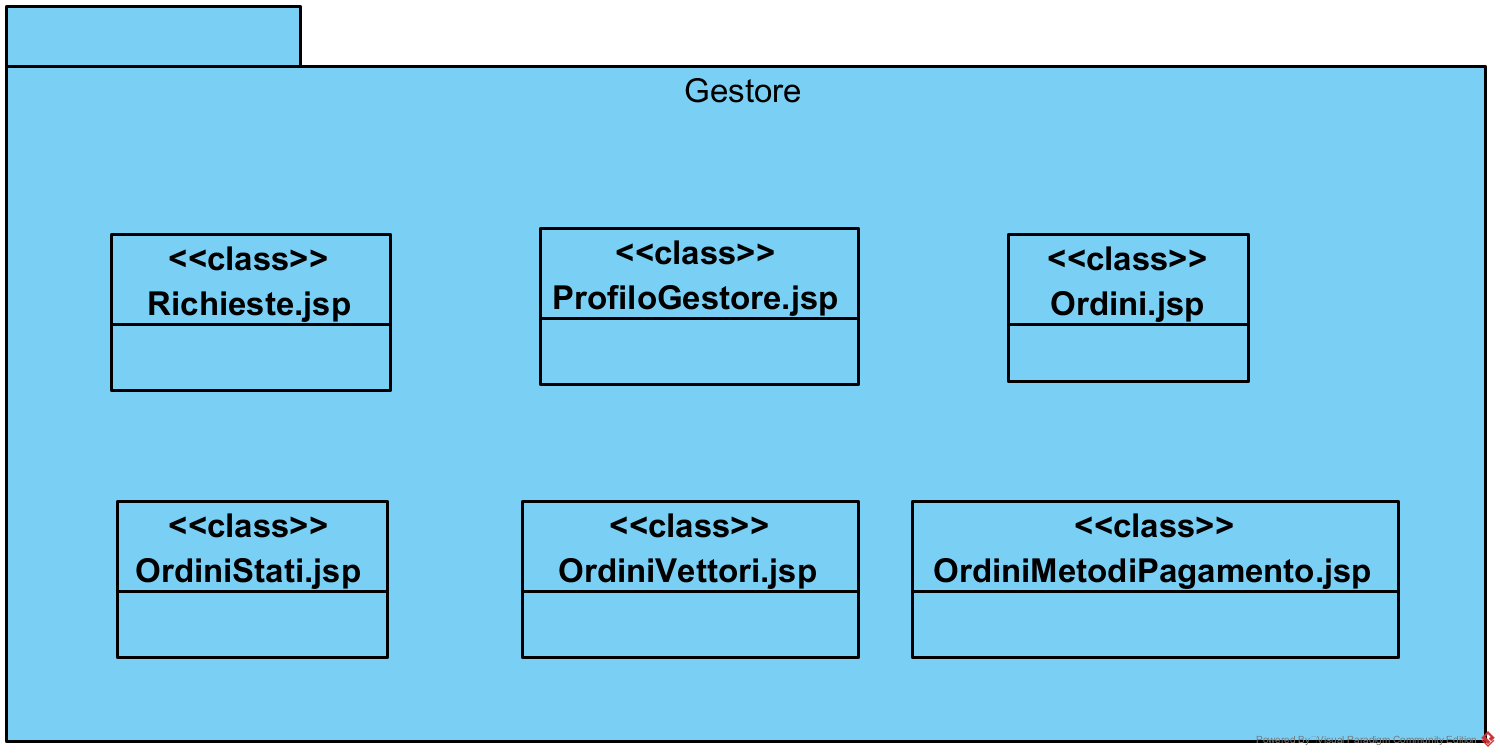


* + 1. **Package Pubblico**



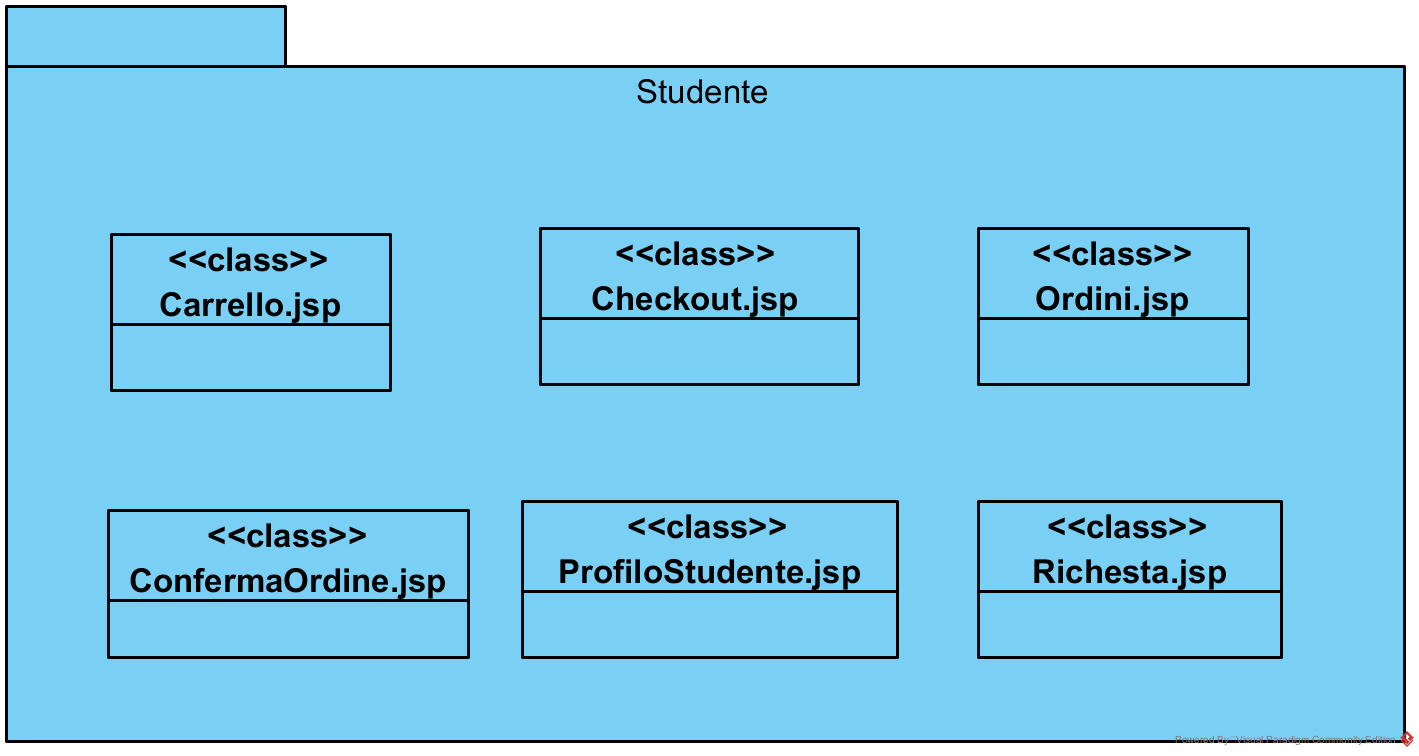
|  |  |
| --- | --- |
| Classe | Descrizione |
| Index.jsp | La view che mostra l’HomePage agli utenti non loggati al sistema. |
| Accedi.jsp | La view che mostra la form per accedere al sistema. |
| ComeFunziona.jsp | La view che spiega l’utilizzo del sistema. |
| Contatti.jsp | La view che mostra i contatti principali del sistema. |
| Appunti.jsp | La view che mostra il catalogo di appunti presenti nel sistema. |
| Dispense.jsp | La view che mostra il catalogo di dispense presenti nel sistema. |
| Faq.jsp | La view che mostra Domande&Risposte relative all’utilizzo del sistema. |
| Logout.jsp | La view che mostra l’uscita dell’utente dal sistema. |
| Privacy.jsp | La view che mostra le informazioni su come vengono gestiti i dati personali |
| Registrati.jsp | La view che mostra la form per la registrazione dello studente. |

* + 1. **Package Gestore**



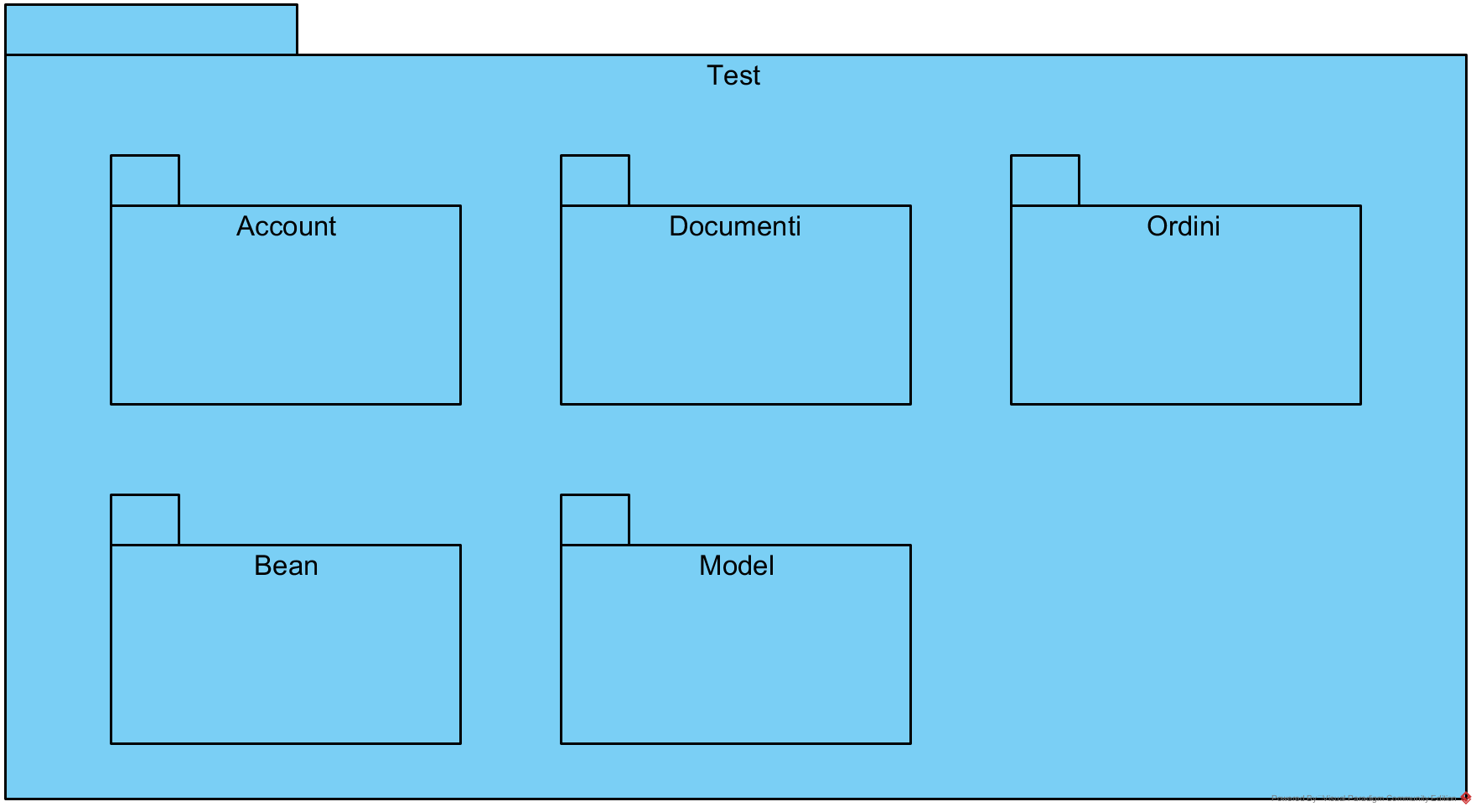
|  |  |
| --- | --- |
| Classe | Descrizione |
| Richieste.jsp | La view che mostra l’elenco di richieste inviate dagli studenti. |
| ProfiloGestore.jsp | La view che mostra il profilo personale del gestore del sistema. |
| Ordini.jsp | La view che mostra gli ordini effettuati dagli studenti. |
| OrdiniStati.jsp | La view che mostra gli lo stato degli ordini dagli studenti. |
| OrdiniVettori.jsp | La view che mostra i vettori utilizzati per la consegna dei documenti. |
| OrdiniMetodiPagamento.jsp | La view che mostra i metodi di pagamento presenti nel sistema. |

* + 1. **Package Studente**

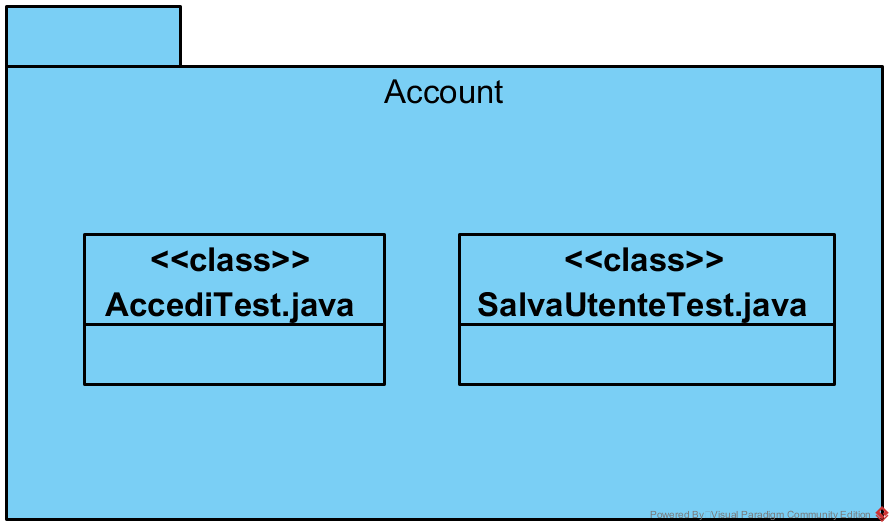


|  |  |
| --- | --- |
| Classe | Descrizione |
| Carrello.jsp | La view che mostra il carrello dello studente. |
| Checkout.jsp | La view che mostra la pagina relativa al finalizzazione dell’ordine. |
| ConfermaOrdine.jsp | La view che mostra i documenti da acquistare. |
| Ordini.jsp | La view che mostra gli ordini effettuati dallo studente. |
| ProfiloStudente.jsp | La view che mostra il profilo personale dello studente. |
| InvioRichiesta.jsp | La view che mostra la pagina relativa all’invio di una richiesta di pubblicazione di un appunto. |

* 1. **Package Test**

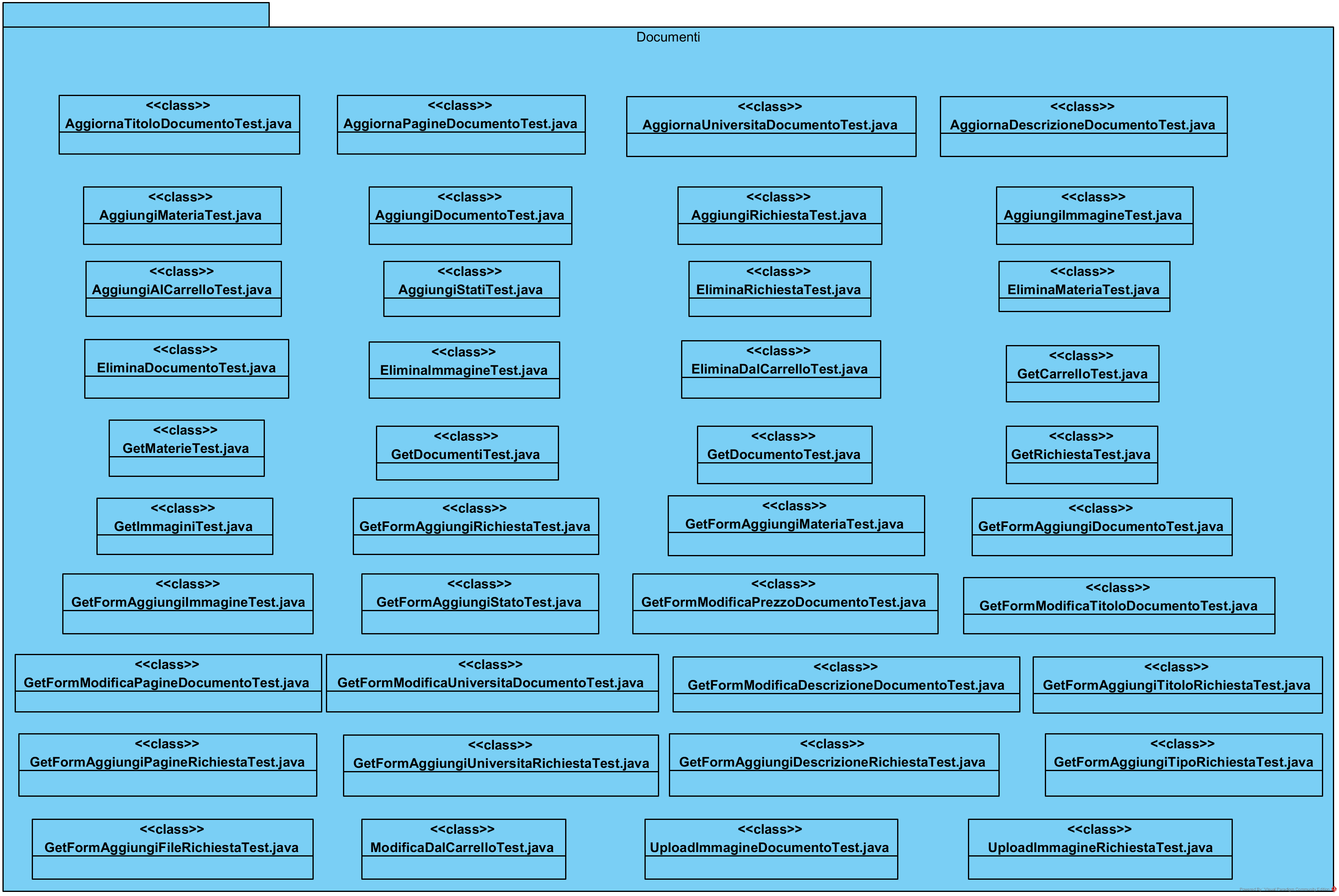
****

* + 1. **Package Account**



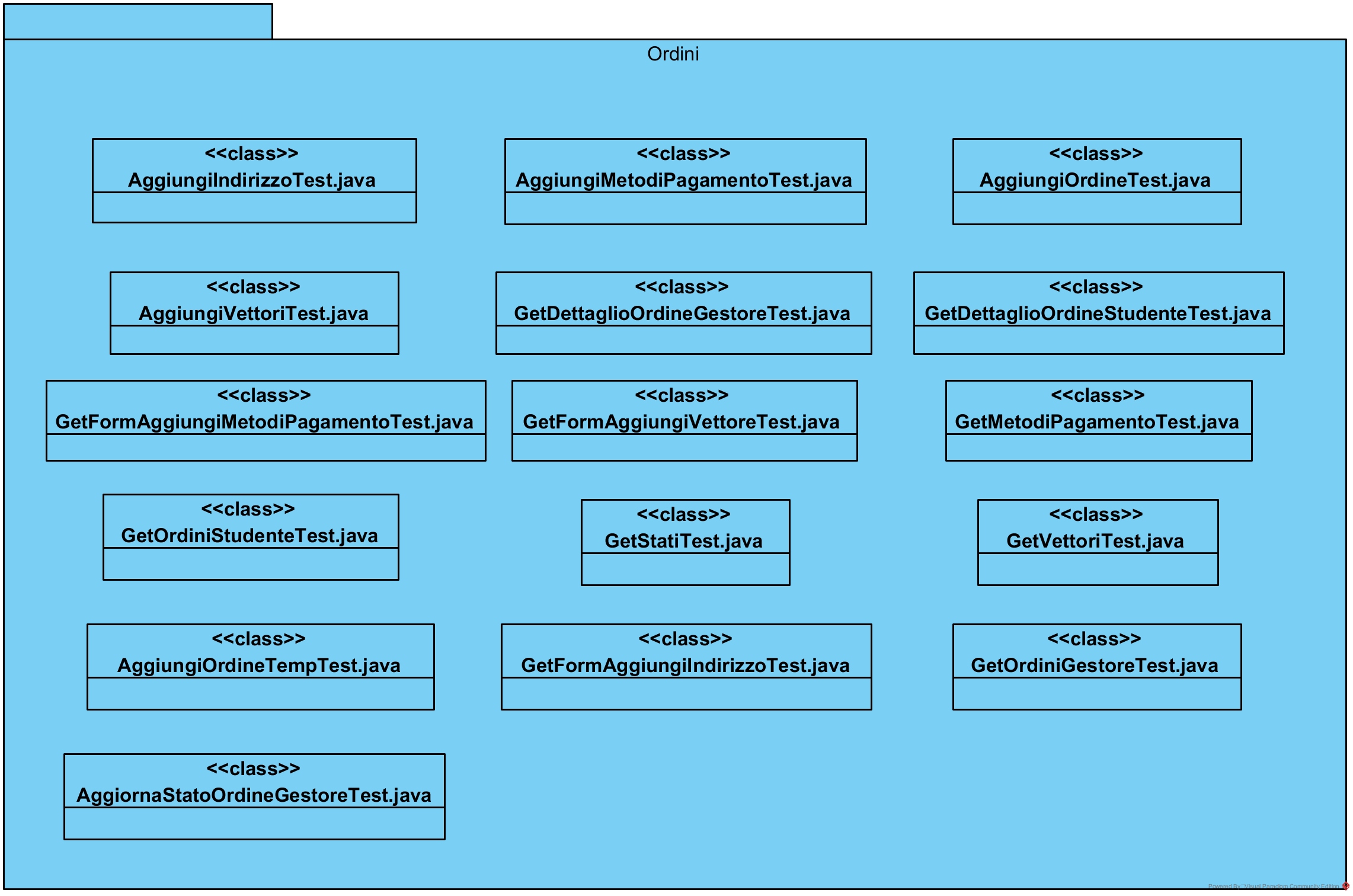
|  |  |
| --- | --- |
| Classe | Descrizione |
| AccediTest.java | La classe che consente di testare i metodi della servlet Accedi. |
| SalvaUtenteTest.java | La classe che consente di testare i metodi della servlet SalvaUtente. |

* + 1. **Package Documenti**



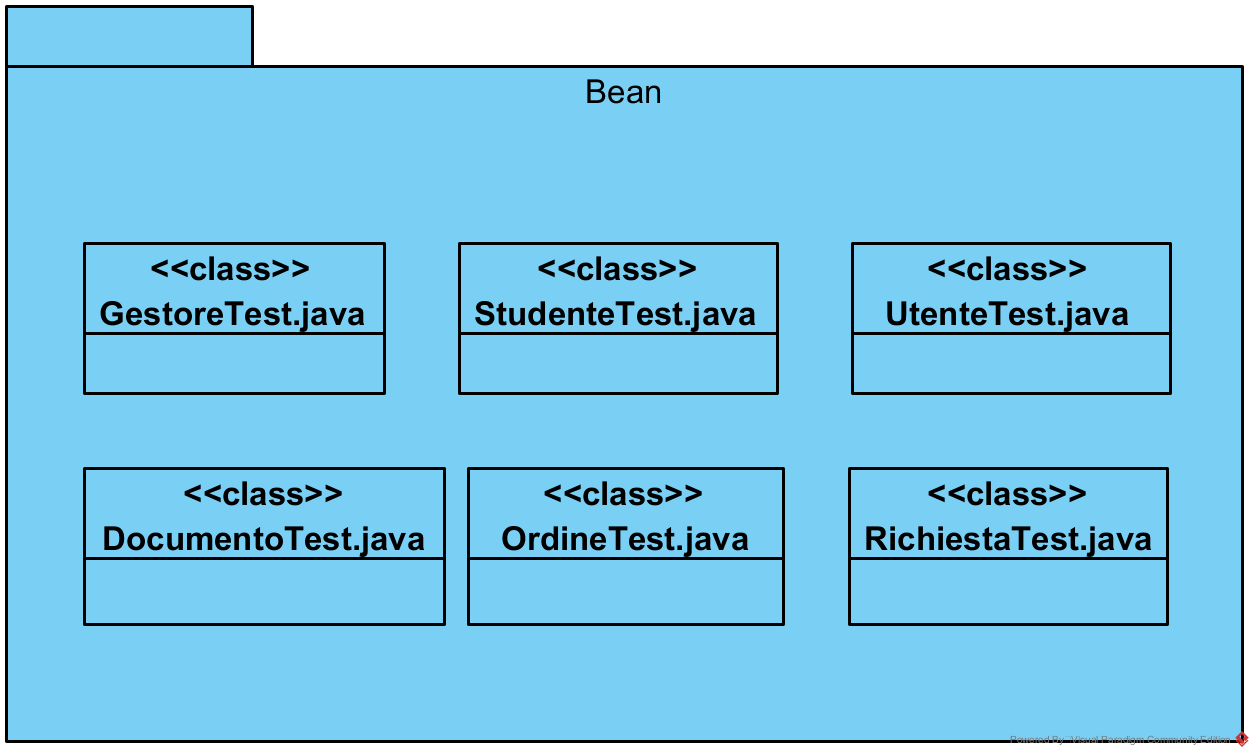
|  |  |
| --- | --- |
| Classe | Descrizione |
| AggiornaPrezzoDocumentoGestoreTest.java | La classe che consente di testare i metodi della servlet AggiornaPrezzoDocumentoGestore. |
| AggiornaTitoloDocumentoTest.java | La classe che consente di testare i metodi della servlet AggiornaTitoloDocumento. |
| AggiornaPagineDocumento.java | La classe che consente di testare i metodi della servlet AggiornaPagineDocumento. |
| AggiornaUniversitaDocumentoTest.java | La classe che consente di testare i metodi della servlet AggiornaUniversitaDocumento. |
| AggiornaDescrizioneDocumentoTest.java | La classe che consente di testare i metodi della servlet AggiornaDescrizioneDocumento. |
| AggiungiRichiestaTest.java | La classe che consente di testare i metodi della servlet AggiungiRichiesta.java. |
| AggiungiMateriaTest.java | La classe che consente di testare i metodi della servlet AggiungiMateria. |
| AggiungiDocumentoTest.java | La classe che consente di testare i metodi della servlet AggiungiDocumento. |
| AggiungiImmagineTest.java | La classe che consente di testare i metodi della servlet AggiungiImmagine. |
| AggiungiAlCarrelloTest.java | La classe che consente di testare i metodi della servlet AggiungiAlCarrello. |
| AggiungiStatiTest.java | La classe che consente di testare i metodi della servlet AggiungiStati. |
| EliminaRichiestaTest.java | La classe che consente di testare i metodi della servlet EliminaRichiesta. |
| EliminaMateriaTest.java | La classe che consente di testare i metodi della servlet EliminaMateria |
| EliminaDocumentoTest.java | La classe che consente di testare i metodi della servlet EliminaDocumento. |
| EliminaImmagineTest.java | La classe che consente di testare i metodi della servlet EliminaImmagine. |
| EliminaDalCarrelloTest.java | La classe che consente di testare i metodi della servlet EliminaDalCarrello. |
| GetCarrelloTest.java | La classe che consente di testare i metodi della servlet GetCarrello. |
| GetMaterieTest.java | La classe che consente di testare i metodi della servlet GetMaterie. |
| GetDocumentiTest.java | La classe che consente di testare i metodi della servlet GetDocumenti. |
| GetDocumentoTest.java | La classe che consente di testare i metodi della servlet GetDocumento. |
| GetRichiestaTest.java | La classe che consente di testare i metodi della servlet GetRichiesta. |
| GetImmaginiTest.java | La classe che consente di testare i metodi della servlet GetImmagini. |
| GetFormAggiungiRichiestaTest.java | La classe che consente di testare i metodi della servlet GetFormAggiungiRichiesta. |
| GetFormAggiungiMateriaTest.java | La classe che consente di testare i metodi della servlet GetFormAggiungiMateria. |
| GetFormAggiungiDocumentoTest.java | La classe che consente di testare i metodi della servlet GetFormAggiungiDocumento. |
| GetFormAggiungiImmagineTest.java | La classe che consente di testare i metodi della servlet GetFormAggiungiImmagine. |
| GetFormAggiungiStatoTest.java | La classe che consente di testare i metodi della servlet GetFormAggiungiStato. |
| GetFormModificaPrezzoDocumentoTest.java | La classe che consente di testare i metodi della servlet GetFormModificaPrezzoDocumento. |
| GetFormModificaTitoloDocumentoTest.java | La classe che consente di testare i metodi della servlet GetFormModificaTitoloDocumento. |
| GetFormModificaPagineDocumentoTest.java | La classe che consente di testare i metodi della servlet GetFormModificaPagineDocumento. |
| GetFormModificaUniversitaDocumentoTest.java | La classe che consente di testare i metodi della servlet GetFormModificaUniversitaDocumento. |
| GetFormModificaDescrizioneDocumentoTest.java | La classe che consente di testare i metodi della servlet GetFormModificaDescrizioneDocumento. |
| GetFormAggiungiTitoloRichiestaTest.java | La classe che consente di testare i metodi della servlet GetFormAggiungiTitoloRichiesta. |
| GetFormAggiungiPagineRichiestaTest.java | La classe che consente di testare i metodi della servlet GetFormAggiungiPagineRichiesta. |
| GetFormAggiungiUniversitaRichiestaTest.java | La classe che consente di testare i metodi della servlet GetFormAggiungiUniversitaRichiesta. |
| GetFormAggiungiDescrizioneRichiestaTest.java | La classe che consente di testare i metodi della servlet GetFormAggiungiDescrizioneRichiesta. |
| GetFormAggiungiTipoRichiestaTest.java | La classe che consente di testare i metodi della servlet GetFormAggiungiTipoRichiesta. |
| GetFormAggiungiFileRichiestaTest.java | La classe che consente di testare i metodi della servlet GetFormAggiungiFileRichiesta. |
| ModificaDalCarrelloTest.java | La classe che consente di testare i metodi della servlet ModificaDalCarrello. |
| UploadImmagineDocumentoTest.java | La classe che consente di testare i metodi della servlet UploadImmagineDocumento. |
| UploadImmagineRichiestaTest.java | La classe che consente di testare i metodi della servlet UploadImmagineRichiesta. |

* + 1. **Package Ordini**



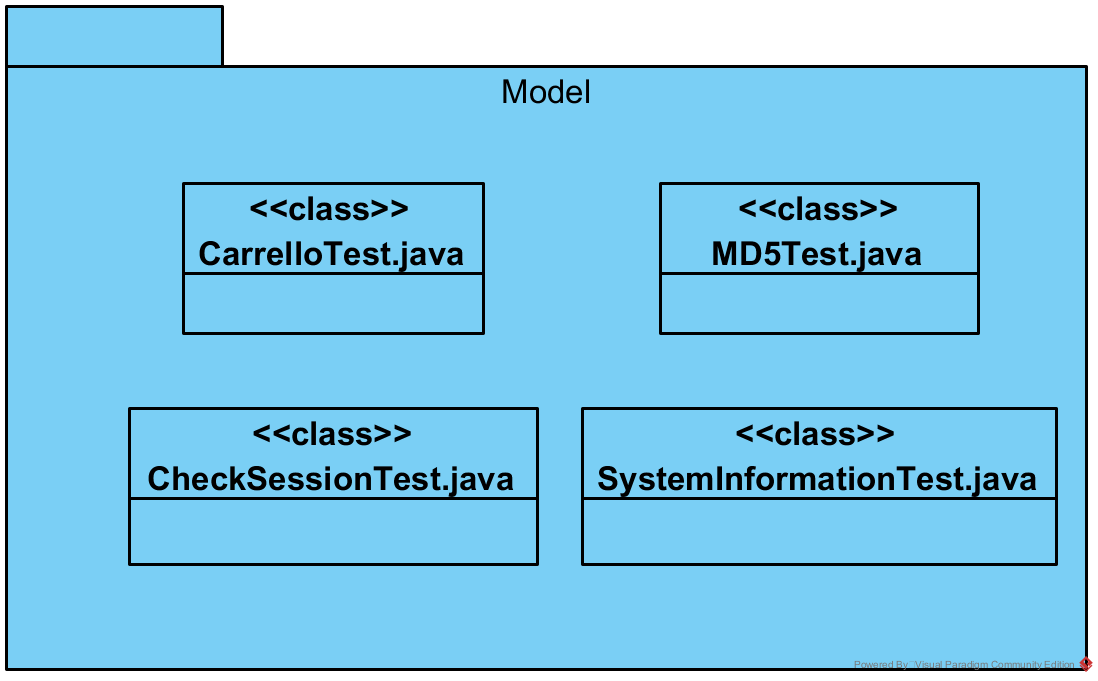
|  |  |
| --- | --- |
| Classe | Descrizione |
| AggiungiIndirizzoTest.java | La classe che consente di testare i metodi della servlet AggiungiIndirizzoTest |
| AggiungiMetodiPagamentoTest.java | La classe che consente di testare i metodi della servlet AggiungiMetodiPagamentoTest |
| AggiungiOrdineTest.java | La classe che consente di testare i metodi della servlet AggiungiOrdineTest |
| AggiungiOrdineTempTest.java | La classe che consente di testare i metodi della servlet AggiungiOrdineTempTest |
| AggiungiVettoriTest.java | La classe che consente di testare i metodi della servlet AggiungiVettoriTest |
| GetDettaglioOrdineGestoreTest.java | La classe che consente di testare i metodi della servlet GetDettaglioOrdineGestoreTest |
| GetDettaglioOrdineStudenteTest.java | La classe che consente di testare i metodi della servlet GetDettaglioOrdineStudenteTest |
| GetFormAggiungiIndirizzoTest.java | La classe che consente di testare i metodi della servlet GetFormAggiungiIndirizzoTest |
| GetFormAggiungiMetodiPagamentoTest.java | La classe che consente di testare i metodi della servlet GetFormAggiungiMetodiPagamentoTest |
| GetFormAggiungiVettoreTest.java | La classe che consente di testare i metodi della servlet GetFormAggiungiVettoreTest |
| GetFormModificaStatoOrdineGestoreTest.java | La classe che consente di testare i metodi della servlet GetFormModificaStatoOrdineGestoreTest |
| GetMetodiPagamentoTest.java | La classe che consente di testare i metodi della servlet GetMetodiPagamentoTest |
| GetOrdiniGestoreTest.java | La classe che consente di testare i metodi della servlet GetOrdiniGestoreTest |
| GetOrdiniStudenteTest.java | La classe che consente di testare i metodi della servlet GetOrdiniStudenteTest |
| GetStatiTest.java | La classe che consente di testare i metodi della servlet GetStatiTest |
| GetVettoriTest.java | La classe che consente di testare i metodi della servlet GetVettoriTest |

* + 1. **Package Bean**

******

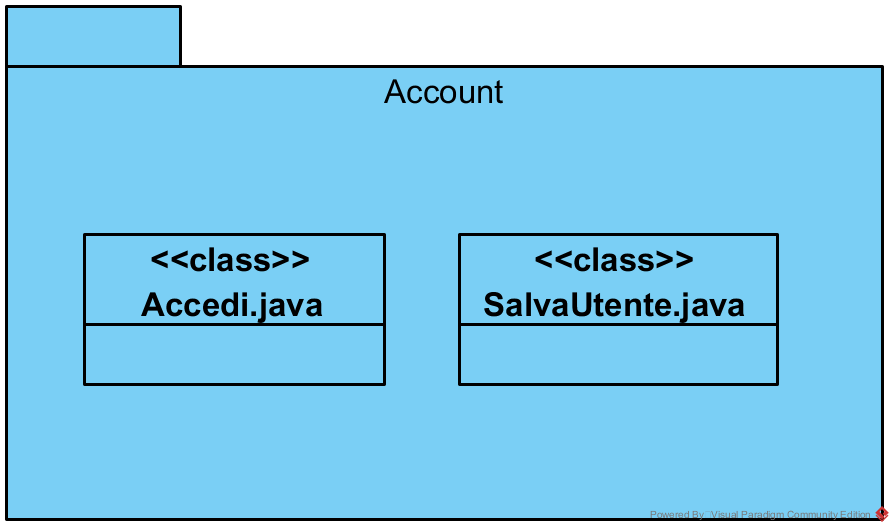
|  |  |
| --- | --- |
| Classe | Descrizione |
| GestoreBeanTest.java | La classe che consente di testare i metodi del bean Gestore. |
| StudenteBeanTest.java | La classe che consente di testare i metodi del bean Studente. |
| UtenteBeanTest.java | La classe che consente di testare i metodi del bean Utente. |
| DocumentoBeanTest.java | La classe che consente di testare i metodi del bean Documento. |
| OrdineBeanTest.java | La classe che consente di testare i metodi del bean Ordine. |
| RichiestaBeanTest.java | La classe che consente di testare i metodi del bean Richiesta. |

* + 1. **Package Model**



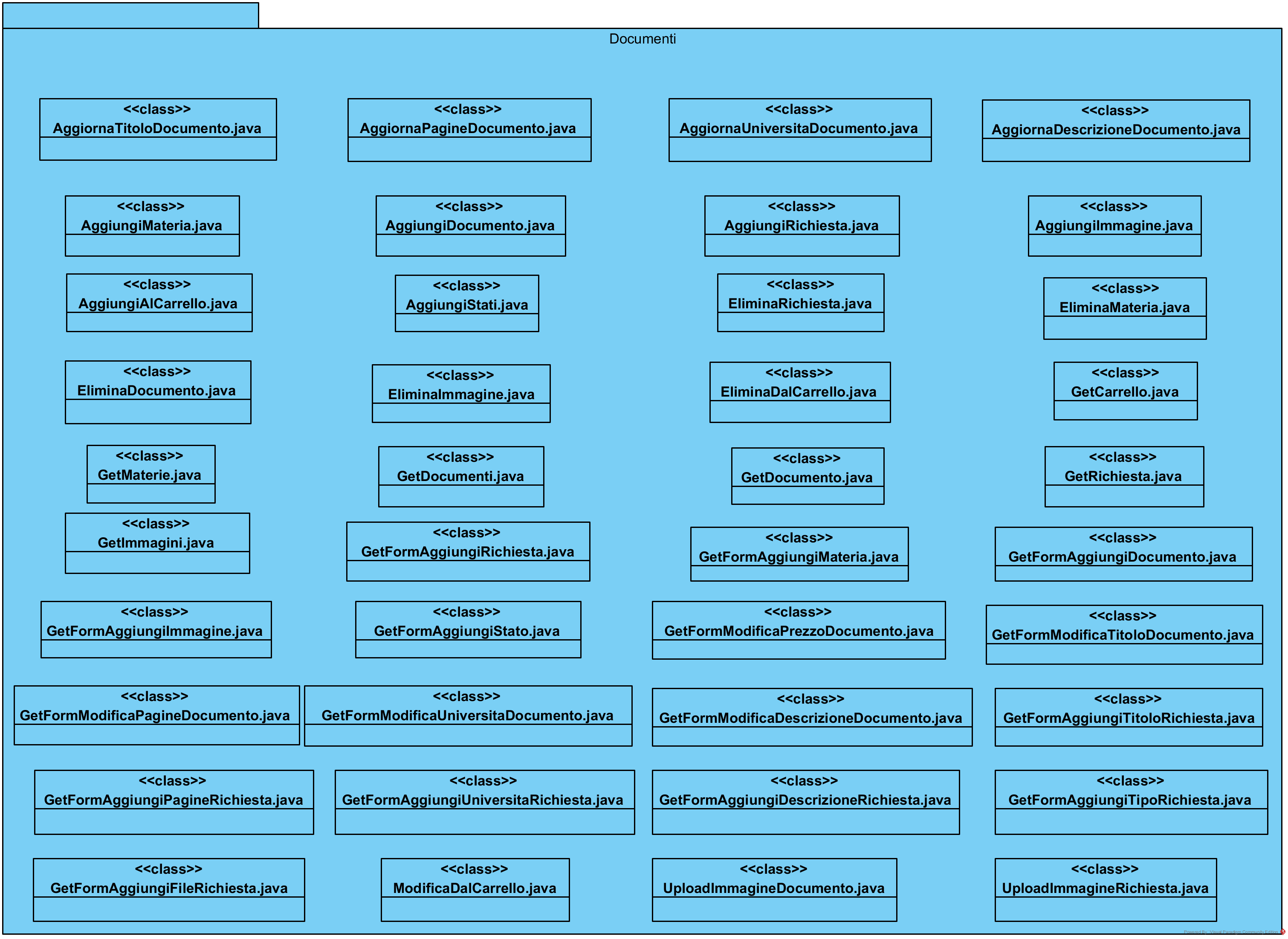
|  |  |
| --- | --- |
| Classe | Descrizione |
| CarrelloTest.java | La classe che consente di testare i metodi del model Carrello. |
| CheckSessionTest.java | La classe che consente di testare i metodi del model CheckSession |
| MD5Test.java | La classe che consente di testare i metodi del model MD5. |
| SystemInformationTest.java | La classe che consente di testare i metodi del model SystemInformation. |

* 1. **Package Account**

****

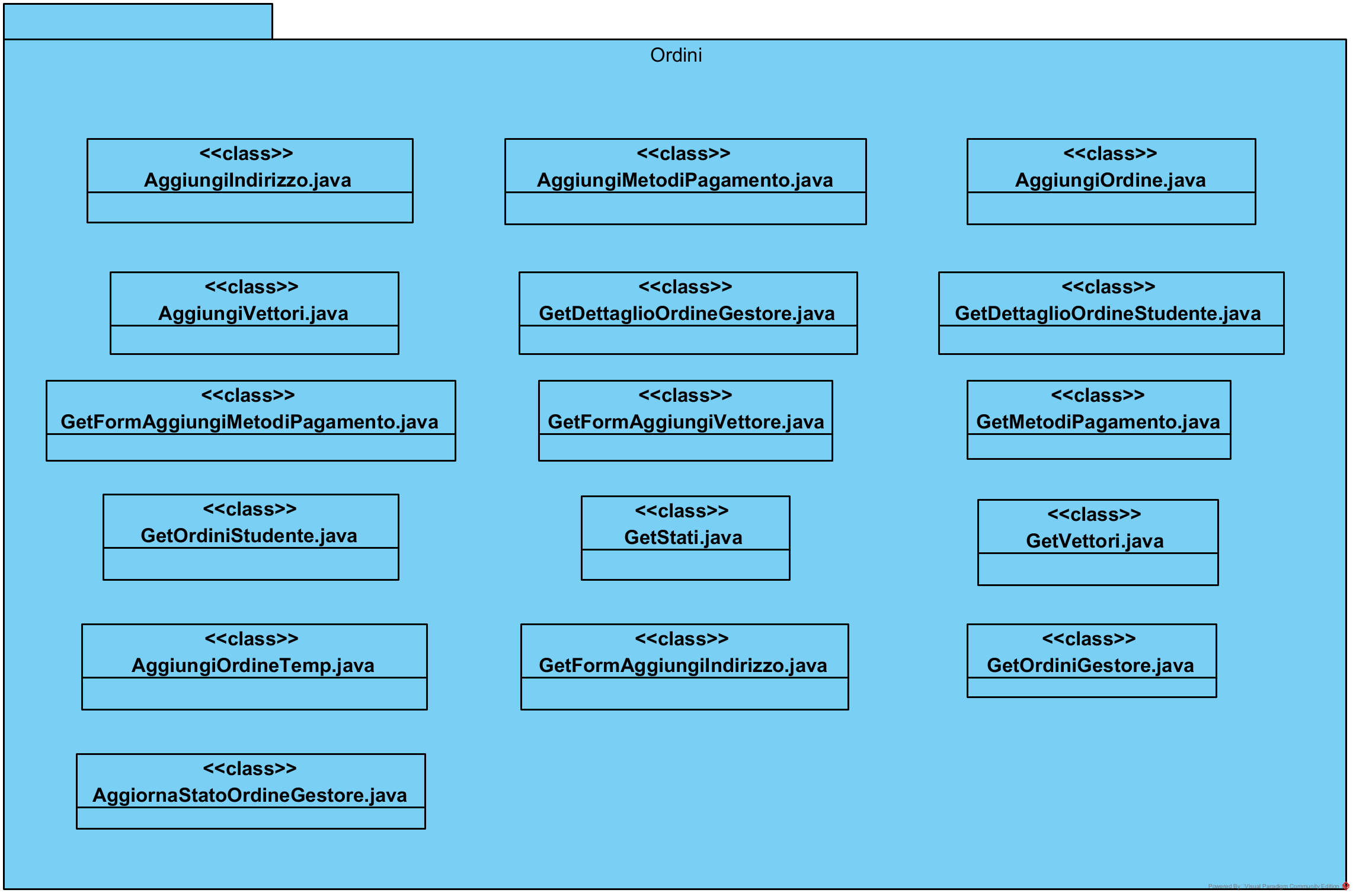
|  |  |
| --- | --- |
| Classe | Descrizione |
| Accedi.java | Servlet che permette la gestione del login. |
| SalvaUtente.java | Servlet che permette la gestione della registrazione dello studente. |

* 1. **Package Documenti**



|  |  |
| --- | --- |
| Classe | Descrizione |
| AggiornaPrezzoDocumentoGestore.java | Servlet che permette la gestione dell’aggiornamento del prezzo del documento. |
| AggiornaTitoloDocumento.java | Servlet che permette la gestione dell’aggiornamento del titolo del documento. |
| AggiornaPagineDocumento.java | Servlet che permette la gestione dell’aggiornamento delle pagine del documento. |
| AggiornaUniversitaDocumento.java | Servlet che permette la gestione dell’aggiornamento dell’’università del documento. |
| AggiornaDescrizioneDocumento.java | Servlet che permette la gestione dell’aggiornamento della descrizione del documento. |
| AggiungiRichiesta.java | Servlet che permette la gestione di aggiunta di una richiesta. |
| AggiungiMateria.java | Servlet che permette la gestione di aggiunta di una materia. |
| AggiungiDocumento.java | Servlet che permette la gestione di aggiunta di un documento. |
| AggiungiImmagine.java | Servlet che permette la gestione di aggiunta di un’ immagine. |
| AggiungiAlCarrello.java | Servlet che permette la gestione di aggiunta di un documento al carrello. |
| AggiungiStati.java | Servlet che permette la gestione di aggiunta dello stato di un ordine. |
| EliminaRichiesta.java | Servlet che permette la gestione di rimozione di una richiesta. |
| EliminaMateria.java | Servlet che permette la gestione di rimozione di una materia. |
| EliminaDocumento.java | Servlet che permette la gestione di rimozione di un documento. |
| EliminaImmagine.java | Servlet che permette la gestione di rimozione di un’immagine. |
| EliminaDalCarrello.java | Servlet che permette la gestione di rimozione di un appunto dal carrello. |
| GetCarrello.java | Servlet che permette la gestione di visualizzazione del carrello. |
| GetMaterie.java | Servlet che permette la gestione di visualizzazione delle materie. |
| GetDocumenti.java | Servlet che permette la gestione di visualizzazione dei documenti. |
| GetDocumento.java | Servlet che permette la gestione di visualizzazione di un documento specifico. |
| GetRichiesta.java | Servlet che permette la gestione di visualizzazione delle richieste. |
| GetImmagini.java | Servlet che permette la gestione di visualizzazione di un’immagine. |
| GetFormAggiungiRichiesta.java | Servlet che permette la gestione di visualizzazione del form per inviare una richiesta. |
| GetFormAggiungiMateria.java | Servlet che permette la gestione di visualizzazione del form per aggiungere una materia. |
| GetFormAggiungiDocumento.java | Servlet che permette la gestione di visualizzazione del form per aggiungere un documento. |
| GetFormAggiungiImmagine.java | Servlet che permette la gestione di visualizzazione del form per aggiungere un’immagine. |
| GetFormAggiungiStato.java | Servlet che permette la gestione di visualizzazione del form per aggiungere uno stato. |
| GetFormModificaPrezzoDocumento.java | Servlet che permette la gestione di visualizzazione del form per modificare un documento. |
| GetFormModificaTitoloDocumento.java | Servlet che permette la gestione di visualizzazione del form per modificare il titolo di un documento. |
| GetFormModificaPagineDocumento.java | Servlet che permette la gestione di visualizzazione del form per modificare le pagine di un documento. |
| GetFormModificaUniversitaDocumento.java | Servlet che permette la gestione di visualizzazione del form per modificare l’università di un documento. |
| GetFormModificaDescrizioneDocumento.java | Servlet che permette la gestione di visualizzazione del form per modificare la descrizione di un documento. |
| GetFormAggiungiTitoloRichiesta.java | Servlet che permette la gestione di visualizzazione del form per aggiungere il titolo di una richiesta. |
| GetFormAggiungiPagineRichiesta.java | Servlet che permette la gestione di visualizzazione del form per aggiungere il numero di pagine di una richiesta. |
| GetFormAggiungiUniversitaRichiesta.java | Servlet che permette la gestione di visualizzazione del form per aggiungere l’università di una richiesta |
| GetFormAggiungiDescrizioneRichiesta.java | Servlet che permette la gestione di visualizzazione del form per aggiungere la descrizione di una richiesta. |
| GetFormAggiungiTipoRichiesta.java | Servlet che permette la gestione di visualizzazione del form per aggiungere il tipo di una richiesta. |
| GetFormAggiungiFileRichiesta.java | Servlet che permette la gestione di visualizzazione del form per aggiungere il file di una richiesta |
| ModificaDalCarrello.java | Servlet che permette la gestione di modifica dei documenti dal carrello. |
| UploadImmagineDocumento.java | Servlet che permette la gestione dell’upload dell’immagine del documento. |
| UploadImmagineRichiesta.java | Servlet che permette la gestione dell’upload dell’immagine della richiesta. |

* 1. **Package Ordini**



|  |  |
| --- | --- |
| Classe | Descrizione |
| AggiungiIndirizzo.java | Servlet che permette la gestione di aggiunta dell’indirizzo. |
| AggiungiMetodiPagamento.java | Servlet che permette la gestione di aggiunta del metodo di pagamento. |
| AggiungiOrdine.java | Servlet che permette la gestione di aggiunta dell’ordine. |
| AggiungiOrdineTemp.java | Servlet che permette la gestione di aggiunta dell’ordine temporaneo. |
| AggiungiVettori.java | Servlet che permette la gestione di aggiunta del vettore. |
| GetDettaglioOrdineGestore.java | Servlet che permette la gestione di visualizzazione degli ordini da parte del gestore. |
| GetDettaglioOrdineStudente.java | Servlet che permette la gestione di visualizzazione degli ordini da parte dello studente. |
| GetFormAggiungiIndirizzo.java | Servlet che permette la gestione di visualizzazione del form per aggiungere un indirizzo. |
| GetFormAggiungiMetodiPagamento.java | Servlet che permette la gestione di visualizzazione del form per aggiungere un metodo di pagamento. |
| GetFormAggiungiVettore.java | Servlet che permette la gestione di visualizzazione del form per aggiungere un vettore. |
| GetFormModificaStatoOrdineGestore.java | Servlet che permette la gestione di visualizzazione del form per modificare lo stato di un ordine. |
| GetMetodiPagamento.java | Servlet che permette la gestione di visualizzazione dei metodi di pagament disponibili. |
| GetOrdiniGestore.java | Servlet che permette la gestione di visualizzazione degli ordini da parte del gestore. |
| GetOrdiniStudente.java | Servlet che permette la gestione di visualizzazione degli ordini da parte dello studente. |
| GetStati.java | Servlet che permette la gestione di visualizzazione degli stati. |
| GetVettori.java | Servlet che permette la gestione di visualizzazione dei vettori disponibili. |

1. **Interfacce delle classi**

***Sottosistema Account***

|  |  |  |  |
| --- | --- | --- | --- |
| |  |  | | --- | --- | | Nome Classe |  | | Accedi |
| *Descrizione* | Questa classe rappresenta la gestione della funzionalità relativa al processo di login. |
| Pre-condizione |  |
| Post-condizione |  |
| Invarianti |  |

|  |  |  |  |
| --- | --- | --- | --- |
| |  |  | | --- | --- | | Nome Classe |  | | SalvaUtente |
| *Descrizione* | Questa classe gestisce la logica applicativa inerente alla funzionalità necessaria alla registrazione dello studente. |
| Pre-condizione |  |
| Post-condizione |  |
| Invarianti |  |

***Sottosistema Documenti***

|  |  |  |  |
| --- | --- | --- | --- |
| |  |  | | --- | --- | | Nome Classe |  | | AggiornaPrezzoDocumentoGestore |
| *Descrizione* | Questa classe rappresenta la gestione della funzionalità relativa all’aggiornamento del prezzo di un documento. |
| Pre-condizione |  |
| Post-condizione |  |
| Invarianti |  |

|  |  |  |  |
| --- | --- | --- | --- |
| |  |  | | --- | --- | | Nome Classe |  | | AggiornaTitoloDocumento |
| *Descrizione* | Questa classe rappresenta la gestione della funzionalità relativa all’aggiornamento del titolo di un documento. |
| Pre-condizione |  |
| Post-condizione |  |
| Invarianti |  |

|  |  |  |  |
| --- | --- | --- | --- |
| |  |  | | --- | --- | | Nome Classe |  | | AggiornaPagineDocumento |
| *Descrizione* | Questa classe rappresenta la gestione della funzionalità relativa all’aggiornamento del numero di pagine di un documento. |
| Pre-condizione |  |
| Post-condizione |  |
| Invarianti |  |

|  |  |  |  |
| --- | --- | --- | --- |
| |  |  | | --- | --- | | Nome Classe |  | | AggiornaUniversitaDocumento |
| *Descrizione* | Questa classe rappresenta la gestione della funzionalità relativa all’aggiornamento dell’università di un documento. |
| Pre-condizione |  |
| Post-condizione |  |
| Invarianti |  |

|  |  |  |  |
| --- | --- | --- | --- |
| |  |  | | --- | --- | | Nome Classe |  | | AggiornaDescrizioneDocumento |
| *Descrizione* | Questa classe rappresenta la gestione della funzionalità relativa all’aggiornamento della descrizione di un documento. |
| Pre-condizione |  |
| Post-condizione |  |
| Invarianti |  |

|  |  |  |  |
| --- | --- | --- | --- |
| |  |  | | --- | --- | | Nome Classe |  | | AggiungiAlCarrello |
| *Descrizione* | Questa classe gestisce la logica applicativa inerente alla funzionalità necessaria all’aggiunta di un documento al carrello. |
| Pre-condizione |  |
| Post-condizione |  |
| Invarianti |  |

|  |  |  |  |
| --- | --- | --- | --- |
| |  |  | | --- | --- | | Nome Classe |  | | AggiungiMateria |
| *Descrizione* | Questa classe gestisce la logica applicativa inerente alla funzionalità necessaria all’aggiunta della materia. |
| Pre-condizione |  |
| Post-condizione |  |
| Invarianti |  |

|  |  |  |  |
| --- | --- | --- | --- |
| |  |  | | --- | --- | | Nome Classe |  | | AggiungiDocumento |
| *Descrizione* | Questa classe gestisce la logica applicativa inerente alla funzionalità necessaria all’aggiunta di un nuovo documento. |
| Pre-condizione |  |
| Post-condizione |  |
| Invarianti |  |

|  |  |  |  |
| --- | --- | --- | --- |
| |  |  | | --- | --- | | Nome Classe |  | | AggiungiImmagine |
| *Descrizione* | Questa classe gestisce la logica applicativa inerente alla funzionalità necessaria all’aggiunta di un’immagine. |
| Pre-condizione |  |
| Post-condizione |  |
| Invarianti |  |

|  |  |  |  |
| --- | --- | --- | --- |
| |  |  | | --- | --- | | Nome Classe |  | | AggiungiAlCarrello |
| *Descrizione* | Questa classe gestisce la logica applicativa inerente alla funzionalità necessaria all’aggiunta di un documento al carrello. |
| Pre-condizione |  |
| Post-condizione |  |
| Invarianti |  |

|  |  |  |  |
| --- | --- | --- | --- |
| |  |  | | --- | --- | | Nome Classe |  | | AggiungiStati |
| *Descrizione* | Questa classe gestisce la logica applicativa inerente alla funzionalità necessaria all’aggiunta di uno stato ad un ordine. |
| Pre-condizione |  |
| Post-condizione |  |
| Invarianti |  |

|  |  |  |  |
| --- | --- | --- | --- |
| |  |  | | --- | --- | | Nome Classe |  | | EliminaRichiesta |
| *Descrizione* | Questa classe gestisce la logica applicativa inerente alla funzionalità necessaria alla rimozione della richiesta inviata dallo studente. |
| Pre-condizione |  |
| Post-condizione |  |
| Invarianti |  |

|  |  |  |  |
| --- | --- | --- | --- |
| |  |  | | --- | --- | | Nome Classe |  | | EliminaMateria |
| *Descrizione* | Questa classe gestisce la logica applicativa inerente alla funzionalità necessaria alla rimozione di una materia inserita dal Gestore. |
| Pre-condizione |  |
| Post-condizione |  |
| Invarianti |  |

|  |  |  |  |
| --- | --- | --- | --- |
| |  |  | | --- | --- | | Nome Classe |  | | EliminaDocumento |
| *Descrizione* | Questa classe gestisce la logica applicativa inerente alla funzionalità necessaria alla rimozione di una documento inserito dal Gestore. |
| Pre-condizione |  |
| Post-condizione |  |
| Invarianti |  |

|  |  |  |  |
| --- | --- | --- | --- |
| |  |  | | --- | --- | | Nome Classe |  | | EliminaImmagine |
| *Descrizione* | Questa classe gestisce la logica applicativa inerente alla funzionalità necessaria alla rimozione di un’immagine. |
| Pre-condizione |  |
| Post-condizione |  |
| Invarianti |  |

|  |  |  |  |
| --- | --- | --- | --- |
| |  |  | | --- | --- | | Nome Classe |  | | EliminaDalCarrello |
| *Descrizione* | Questa classe gestisce la logica applicativa inerente alla funzionalità necessaria alla rimozione di un documento dal carrello. |
| Pre-condizione |  |
| Post-condizione |  |
| Invarianti |  |

|  |  |  |  |
| --- | --- | --- | --- |
| |  |  | | --- | --- | | Nome Classe |  | | GetCarrello |
| *Descrizione* | Questa classe gestisce la logica applicativa inerente alla funzionalità necessaria per visualizzare il carrello. |
| Pre-condizione |  |
| Post-condizione |  |
| Invarianti |  |

|  |  |  |  |
| --- | --- | --- | --- |
| |  |  | | --- | --- | | Nome Classe |  | | GetMaterie |
| *Descrizione* | Questa classe gestisce la logica applicativa inerente alla funzionalità necessaria per visualizzare le materie. |
| Pre-condizione |  |
| Post-condizione |  |
| Invarianti |  |

|  |  |  |  |
| --- | --- | --- | --- |
| |  |  | | --- | --- | | Nome Classe |  | | GetDocumenti |
| *Descrizione* | Questa classe gestisce la logica applicativa inerente alla funzionalità necessaria per visualizzare i documenti. |
| Pre-condizione |  |
| Post-condizione |  |
| Invarianti |  |

|  |  |  |  |
| --- | --- | --- | --- |
| |  |  | | --- | --- | | Nome Classe |  | | GetDocumento |
| *Descrizione* | Questa classe gestisce la logica applicativa inerente alla funzionalità necessaria per visualizzare il documento scelto. |
| Pre-condizione |  |
| Post-condizione |  |
| Invarianti |  |

|  |  |  |  |
| --- | --- | --- | --- |
| |  |  | | --- | --- | | Nome Classe |  | | GetRichiesta |
| *Descrizione* | Questa classe gestisce la logica applicativa inerente alla funzionalità necessaria per visualizzare le richieste. |
| Pre-condizione |  |
| Post-condizione |  |
| Invarianti |  |

|  |  |  |  |
| --- | --- | --- | --- |
| |  |  | | --- | --- | | Nome Classe |  | | GetImmagini |
| *Descrizione* | Questa classe gestisce la logica applicativa inerente alla funzionalità necessaria per visualizzare un’immagine. |
| Pre-condizione |  |
| Post-condizione |  |
| Invarianti |  |

|  |  |  |  |
| --- | --- | --- | --- |
| |  |  | | --- | --- | | Nome Classe |  | | GetFormAggiungiRichiesta |
| *Descrizione* | Questa classe gestisce la logica applicativa inerente alla funzionalità necessaria alla visualizzazione del form per aggiungere una richiesta. |
| Pre-condizione |  |
| Post-condizione |  |
| Invarianti |  |

|  |  |  |  |
| --- | --- | --- | --- |
| |  |  | | --- | --- | | Nome Classe |  | | GetFormAggiungiMateria |
| *Descrizione* | Questa classe gestisce la logica applicativa inerente alla funzionalità necessaria alla visualizzazione del form per aggiungere una materia. |
| Pre-condizione |  |
| Post-condizione |  |
| Invarianti |  |

|  |  |  |  |
| --- | --- | --- | --- |
| |  |  | | --- | --- | | Nome Classe |  | | GetFormAggiungiDocumento |
| *Descrizione* | Questa classe gestisce la logica applicativa inerente alla funzionalità necessaria alla visualizzazione del form per aggiungere un documento. |
| Pre-condizione |  |
| Post-condizione |  |
| Invarianti |  |

|  |  |  |  |
| --- | --- | --- | --- |
| |  |  | | --- | --- | | Nome Classe |  | | GetFormAggiungiStato |
| *Descrizione* | Questa classe gestisce la logica applicativa inerente alla funzionalità necessaria alla visualizzazione del form per aggiungere lo stato di un ordine. |
| Pre-condizione |  |
| Post-condizione |  |
| Invarianti |  |

|  |  |  |  |
| --- | --- | --- | --- |
| |  |  | | --- | --- | | Nome Classe |  | | GetFormModificaPrezzoDocumento |
| *Descrizione* | Questa classe gestisce la logica applicativa inerente alla funzionalità necessaria alla visualizzazione del form per modificare il prezzo di un documento. |
| Pre-condizione |  |
| Post-condizione |  |
| Invarianti |  |

|  |  |  |  |
| --- | --- | --- | --- |
| |  |  | | --- | --- | | Nome Classe |  | | GetFormModificaTitoloDocumento |
| *Descrizione* | Questa classe gestisce la logica applicativa inerente alla funzionalità necessaria alla visualizzazione del form per modificare il titolo di un documento. |
| Pre-condizione |  |
| Post-condizione |  |
| Invarianti |  |

|  |  |  |  |
| --- | --- | --- | --- |
| |  |  | | --- | --- | | Nome Classe |  | | GetFormModificaPagineDocumento |
| *Descrizione* | Questa classe gestisce la logica applicativa inerente alla funzionalità necessaria alla visualizzazione del form per modificare il numero di pagine di un documento. |
| Pre-condizione |  |
| Post-condizione |  |
| Invarianti |  |

|  |  |  |  |
| --- | --- | --- | --- |
| |  |  | | --- | --- | | Nome Classe |  | | GetFormModificaUniversitaDocumento |
| *Descrizione* | Questa classe gestisce la logica applicativa inerente alla funzionalità necessaria alla visualizzazione del form per modificare l’università di un documento. |
| Pre-condizione |  |
| Post-condizione |  |
| Invarianti |  |

|  |  |  |  |
| --- | --- | --- | --- |
| |  |  | | --- | --- | | Nome Classe |  | | GetFormModificaDescrizioneDocumento |
| *Descrizione* | Questa classe gestisce la logica applicativa inerente alla funzionalità necessaria alla visualizzazione del form per modificare la descrizione di un documento. |
| Pre-condizione |  |
| Post-condizione |  |
| Invarianti |  |

|  |  |  |  |
| --- | --- | --- | --- |
| |  |  | | --- | --- | | Nome Classe |  | | GetFormAggiungiTitoloRichiesta |
| *Descrizione* | Questa classe gestisce la logica applicativa inerente alla funzionalità necessaria alla visualizzazione del form per aggiungere il titolo del documento da inviare. |
| Pre-condizione |  |
| Post-condizione |  |
| Invarianti |  |

|  |  |  |  |
| --- | --- | --- | --- |
| |  |  | | --- | --- | | Nome Classe |  | | GetFormAggiungiPagineRichiesta |
| *Descrizione* | Questa classe gestisce la logica applicativa inerente alla funzionalità necessaria alla visualizzazione del form per aggiungere il numero di pagine del documento da inviare. |
| Pre-condizione |  |
| Post-condizione |  |
| Invarianti |  |

|  |  |  |  |
| --- | --- | --- | --- |
| |  |  | | --- | --- | | Nome Classe |  | | GetFormAggiungiUniversitaRichiesta |
| *Descrizione* | Questa classe gestisce la logica applicativa inerente alla funzionalità necessaria alla visualizzazione del form per aggiungere l’università del documento da inviare. |
| Pre-condizione |  |
| Post-condizione |  |
| Invarianti |  |

|  |  |  |  |
| --- | --- | --- | --- |
| |  |  | | --- | --- | | Nome Classe |  | | GetFormAggiungiDescrizioneRichiesta |
| *Descrizione* | Questa classe gestisce la logica applicativa inerente alla funzionalità necessaria alla visualizzazione del form per aggiungere la descrizione del documento da inviare. |
| Pre-condizione |  |
| Post-condizione |  |
| Invarianti |  |

|  |  |  |  |
| --- | --- | --- | --- |
| |  |  | | --- | --- | | Nome Classe |  | | GetFormAggiungiTipoRichiesta |
| *Descrizione* | Questa classe gestisce la logica applicativa inerente alla funzionalità necessaria alla visualizzazione del form per aggiungere il tipo di documento da inviare. |
| Pre-condizione |  |
| Post-condizione |  |
| Invarianti |  |

|  |  |  |  |
| --- | --- | --- | --- |
| |  |  | | --- | --- | | Nome Classe |  | | GetFormAggiungiFileRichiesta |
| *Descrizione* | Questa classe gestisce la logica applicativa inerente alla funzionalità necessaria alla visualizzazione del form per aggiungere il file del documento da inviare. |
| Pre-condizione |  |
| Post-condizione |  |
| Invarianti |  |

|  |  |  |  |
| --- | --- | --- | --- |
| |  |  | | --- | --- | | Nome Classe |  | | UploadImmagineDocumento |
| *Descrizione* | Questa classe gestisce la logica applicativa inerente alla funzionalità necessaria all’upload dell’immagire relativa al documento. |
| Pre-condizione |  |
| Post-condizione |  |
| Invarianti |  |

|  |  |  |  |
| --- | --- | --- | --- |
| |  |  | | --- | --- | | Nome Classe |  | | UploadImmagineRichiesta |
| *Descrizione* | Questa classe gestisce la logica applicativa inerente alla funzionalità necessaria all’upload dell’immagire relativa alla richiesta. |
| Pre-condizione |  |
| Post-condizione |  |
| Invarianti |  |

***Sottosistema Account***

|  |  |  |  |
| --- | --- | --- | --- |
| |  |  | | --- | --- | | Nome Classe |  | | AggiungiIndirizzo |
| *Descrizione* | Questa classe gestisce la logica applicativa inerente alla funzionalità necessaria all’aggiunta di un indirizzo. |
| Pre-condizione |  |
| Post-condizione |  |
| Invarianti |  |

|  |  |  |  |
| --- | --- | --- | --- |
| |  |  | | --- | --- | | Nome Classe |  | | AggiungiMetodiPagamento |
| *Descrizione* | Questa classe gestisce la logica applicativa inerente alla funzionalità necessaria all’aggiunta di un metodo di pagamento. |
| Pre-condizione |  |
| Post-condizione |  |
| Invarianti |  |

|  |  |  |  |
| --- | --- | --- | --- |
| |  |  | | --- | --- | | Nome Classe |  | | AggiungiOrdine |
| *Descrizione* | Questa classe gestisce la logica applicativa inerente alla funzionalità necessaria all’aggiunta di un ordine. |
| Pre-condizione |  |
| Post-condizione |  |
| Invarianti |  |

|  |  |  |  |
| --- | --- | --- | --- |
| |  |  | | --- | --- | | Nome Classe |  | | AggiungiOrdineTemp |
| *Descrizione* | Questa classe gestisce la logica applicativa inerente alla funzionalità necessaria all’aggiunta di un ordine temporaneo. |
| Pre-condizione |  |
| Post-condizione |  |
| Invarianti |  |

|  |  |  |  |
| --- | --- | --- | --- |
| |  |  | | --- | --- | | Nome Classe |  | | AggiungiVettori |
| *Descrizione* | Questa classe gestisce la logica applicativa inerente alla funzionalità necessaria all’aggiunta di un vettore per la spedizione. |
| Pre-condizione |  |
| Post-condizione |  |
| Invarianti |  |

|  |  |  |  |
| --- | --- | --- | --- |
| |  |  | | --- | --- | | Nome Classe |  | | GetDettaglioOrdineGestore |
| *Descrizione* | Questa classe gestisce la logica applicativa inerente alla funzionalità necessaria per visualizzare l’ordine effettuato da uno studente. |
| Pre-condizione |  |
| Post-condizione |  |
| Invarianti |  |

|  |  |  |  |
| --- | --- | --- | --- |
| |  |  | | --- | --- | | Nome Classe |  | | GetFormAggiungiIndirizzo |
| *Descrizione* | Questa classe gestisce la logica applicativa inerente alla funzionalità necessaria alla visualizzazione del form per aggiungere l’indirizzo di spedizione. |
| Pre-condizione |  |
| Post-condizione |  |
| Invarianti |  |

|  |  |  |  |
| --- | --- | --- | --- |
| |  |  | | --- | --- | | Nome Classe |  | | GetFormAggiungiMetodiPagamento |
| *Descrizione* | Questa classe gestisce la logica applicativa inerente alla funzionalità necessaria alla visualizzazione del form per aggiungere il metodo di pagamento scelto. |
| Pre-condizione |  |
| Post-condizione |  |
| Invarianti |  |

|  |  |  |  |
| --- | --- | --- | --- |
| |  |  | | --- | --- | | Nome Classe |  | | GetFormAggiungiVettore |
| *Descrizione* | Questa classe gestisce la logica applicativa inerente alla funzionalità necessaria alla visualizzazione del form per aggiungere il vettore scelto. |
| Pre-condizione |  |
| Post-condizione |  |
| Invarianti |  |

|  |  |  |  |
| --- | --- | --- | --- |
| |  |  | | --- | --- | | Nome Classe |  | | GetFormModificaStatoOrdineGestore |
| *Descrizione* | Questa classe gestisce la logica applicativa inerente alla funzionalità necessaria alla visualizzazione del form per modificare lo stato dell’ordine da parte del gestore. |
| Pre-condizione |  |
| Post-condizione |  |
| Invarianti |  |

|  |  |  |  |
| --- | --- | --- | --- |
| |  |  | | --- | --- | | Nome Classe |  | | GetMetodiPagamento |
| *Descrizione* | Questa classe gestisce la logica applicativa inerente alla funzionalità necessaria per visualizzare i metodi di pagamento. |
| Pre-condizione |  |
| Post-condizione |  |
| Invarianti |  |

|  |  |  |  |
| --- | --- | --- | --- |
| |  |  | | --- | --- | | Nome Classe |  | | GetOrdiniGestore |
| *Descrizione* | Questa classe gestisce la logica applicativa inerente alla funzionalità necessaria per visualizzare gli ordini effettuati dagli studenti. |
| Pre-condizione |  |
| Post-condizione |  |
| Invarianti |  |

|  |  |  |  |
| --- | --- | --- | --- |
| |  |  | | --- | --- | | Nome Classe |  | | GetOrdiniStudente |
| *Descrizione* | Questa classe gestisce la logica applicativa inerente alla funzionalità necessaria per visualizzare gli ordini effettuati. |
| Pre-condizione |  |
| Post-condizione |  |
| Invarianti |  |

|  |  |  |  |
| --- | --- | --- | --- |
| |  |  | | --- | --- | | Nome Classe |  | | GetStati |
| *Descrizione* | Questa classe gestisce la logica applicativa inerente alla funzionalità necessaria per visualizzare lo stato di un ordine. |
| Pre-condizione |  |
| Post-condizione |  |
| Invarianti |  |

|  |  |  |  |
| --- | --- | --- | --- |
| |  |  | | --- | --- | | Nome Classe |  | | GetVettori |
| *Descrizione* | Questa classe gestisce la logica applicativa inerente alla funzionalità necessaria per visualizzare i vettori per la spedizione. |
| Pre-condizione |  |
| Post-condizione |  |
| Invarianti |  |

|  |  |  |  |
| --- | --- | --- | --- |
| |  |  | | --- | --- | | Nome Classe |  | | AggiornaStatoOrdineGestore |
| *Descrizione* | Questa classe gestisce la logica applicativa inerente alla funzionalità necessaria per modificare lo stato di un ordine. |
| Pre-condizione |  |
| Post-condizione |  |
| Invarianti |  |

***Glossario***

|  |  |
| --- | --- |
| **Sigla/Termine** | **Definizione** |
| urNotes | Nome del sistema che verrà sviluppato. |
| Servlet | Le Servlet sono oggetti scritti in linguaggio Java che operano all’interno di un server web (es. Tomcat) permettendo la creazione di applicazioni web. |
| Query | Interrogazione di un database per estrarre o aggiornare i dati che soddisfano un certo criterio di ricerca. |
| Sito web | Un insieme di pagine web correlate. |
| Layer | Utilizzato per indicare la suddivisione nell’architettura three-tier (tre strati). |
| DBMS | Sistema di gestione di basi di dati, cioè un sistema software progettato per consentire la creazione, la manipolazione e l’interrogazione efficiente di un database. |
| MySQL | È un Relational database management system (RDBMS) composto da un client a riga di comando e un server. |
| Apache Tomcat | È un application server nella forma di contenitore servlet open source. |