

Object Design document

1. Introduzione

Object Design Trade-offs

Ci apprestiamo a definire gli aspetti implementativi del nostro sistema, aspetti che nei documenti precedenti (e.g. "RAD", "SSD" etc.) sono stati tralasciati per definire in modo chiaro gli obiettivi del sistema. Con il seguente documento puntiamo a creare un modello capace di descrivere in modo preciso tutte le funzionalità individuate nei modelli precedenti.

Nel particolare, definiremo le interfacce delle classi, le operazioni, i tipi e gli argomenti e le "signature" dei sottosistemi definiti nel System Design.

Sono necessari però definire una serie di trade-off e di linee guida:

Comprensibilità vs Tempo:

Un codice comprensibile e commentato sarà sempre preferito rispetto ad un codice scritto rapidamente; questo perché è importante che il codice rimanga di facile fruizione per tutti i collaboratori del progetto, anche se potrà rallentarci più avanti nello sviluppo.

Prestazioni vs Costi:

Il progetto è portato avanti senza alcuna sovvenzione economica, di conseguenza tutti i componenti che formeranno il sistema sono scritti da 0, si farà il possibile per evitare di intaccare il meno possibile le prestazioni del sistema per abbassare il più possibile i costi dell'intero sistema.

Interfaccia vs Usabilità:

L'interfaccia fa dell'usabilità il suo punto di forza. Fin dal primo momento la comodità di utilizzo è stata messa in primo piano durante il design, scelta che si rispecchia fin dai primi mock-up.

Sicurezza vs Efficienza:

Il poco tempo a disposizione non ci permette di rendere il sistema resistente a possibili attacchi esterni. Metteremo comunque il minimo di sicurezza possibile nel progetto, assicurandoci che il login/registrazione e tutte le altre interazioni con il database siano sicure e quanto più efficienti possibili.

1.2 Linea guida per la documentazione delle interfacce

Per creare un progetto chiaro, pulito e univocamente compreso, gli sviluppatori dovranno sottostare ad alcune linee guida per la stesura del codice:

Code style:

È buona norma rispettare gli standard di Java circa la formattazione del codice, nello specifico:

- Tutto il codice sorgente deve essere indentato con tabs
- Il codice deve essere formattato secondo lo stile vigente dell'IDE utilizzato (si consiglia caldamente Eclipse)

Esempio:

```
for(Object x: listaElementi){  
    System.out.println(x.getNome());  
}
```

Naming Convention:

È buona norma utilizzare nomi che siano:

- Descrittivi
- Pronunciabili
- Di uso comune
- Non abbreviati (se non per variabili momentanee)
- Utilizzando solo i caratteri consentiti (A-Z, a-z, 0-9)

Variabili:

È buona norma rispettare gli standard di Java circa la nomenclatura delle variabili, nello specifico:

- Devono iniziare per lettera minuscola e ogni parola seguente deve iniziare con una maiuscola.
Esempio: nomeDellaVariabile

- Nel caso di variabili costanti, il nome deve essere interamente in maiuscola ed eventuali spazi devono essere sostituiti da underscore ('_'). Esempio: ORE_IN_UN_GIORNO

Metodi:

È buona norma rispettare gli standard di Java circa la nomenclatura dei metodi, nello specifico:

- Devono iniziare per lettera minuscola e ogni parola seguente deve iniziare con una maiuscola.
Esempio: nomeDelMetodo()

- Eventuali metodi per l'accesso e la modifica delle variabili d'istanza di una classe devono essere del tipo `getNomeVariabile()` e `setNomeVariabile()`

- I commenti devono essere raggruppati in base alla loro funzionalità, devono essere situati prima della dichiarazione del metodo. È obbligatorio usare il sistema "JavaDocs" per poter avere una documentazione completa ed uniforme.

Classi:

È buona norma rispettare gli standard di Java circa la nomenclatura delle classi, nello specifico:

- Devono iniziare per lettera maiuscola e ogni parola seguente deve iniziare con una maiuscola.

Esempio: NomeDellaClasse

- Devono avere un nome autodescrittivo.

1.3 Definizioni, Acronimi, Abbreviazioni e Riferimenti

Acronimi:

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

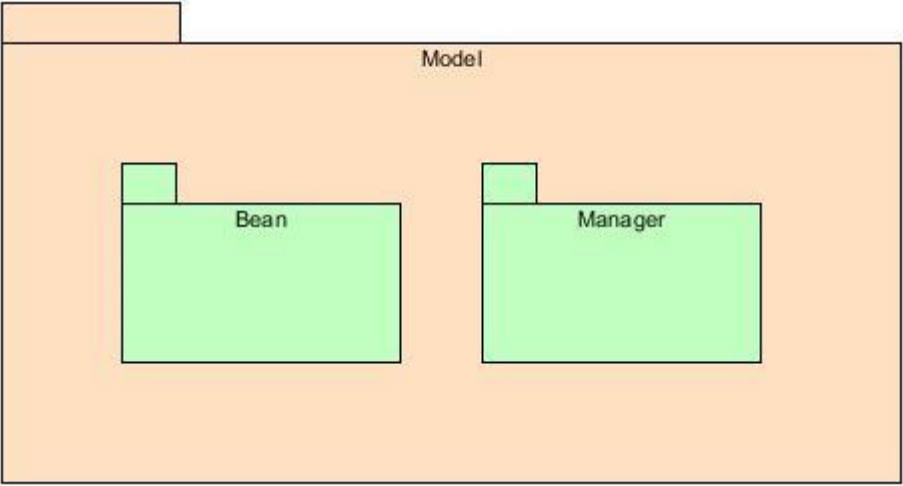
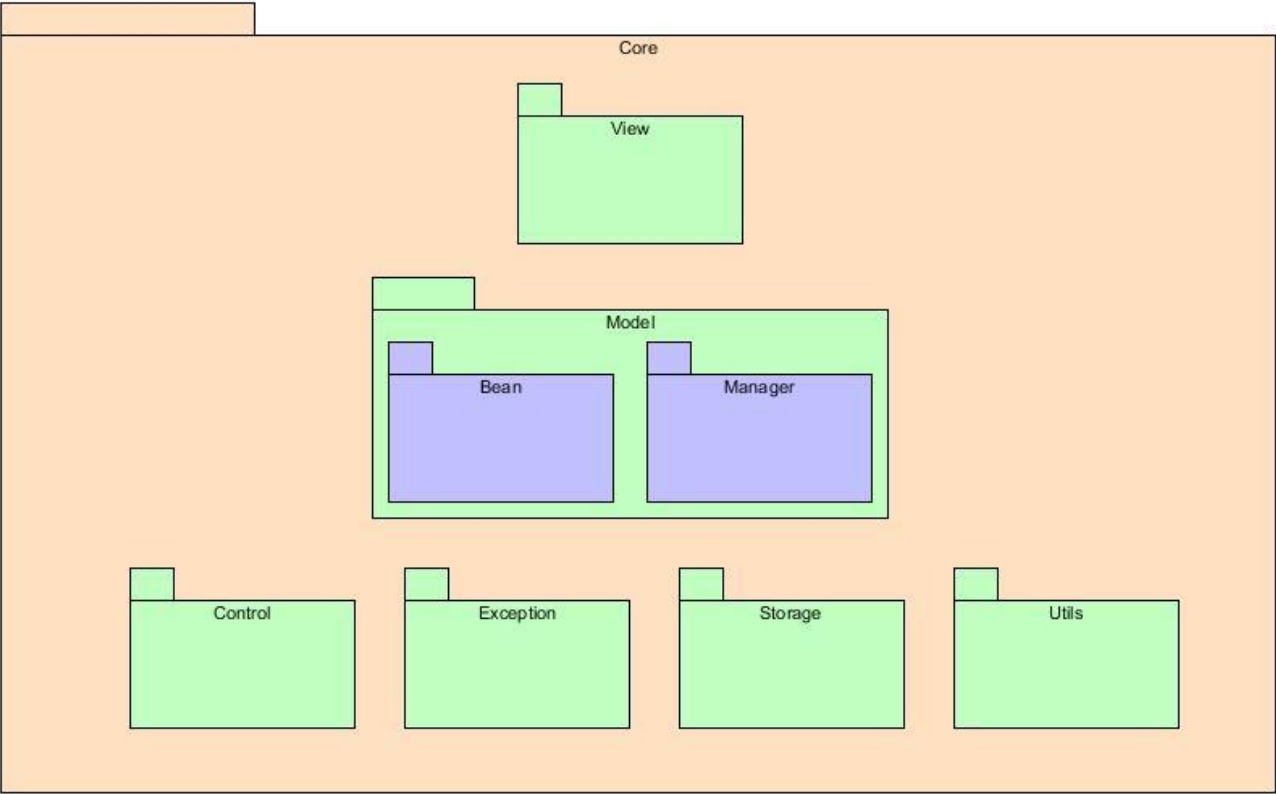
Abbreviazioni:

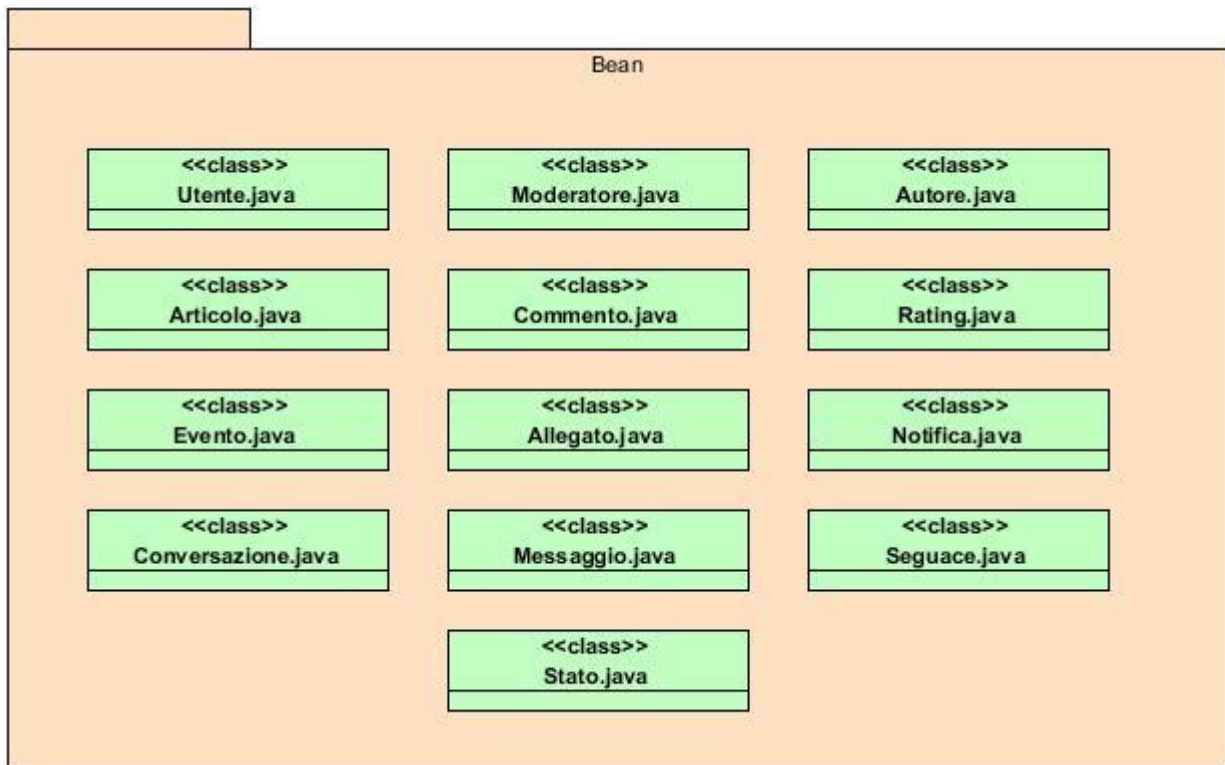
- DB: Database

Riferimenti:

- Libro di testo: Bruegge, A.H. Dutoit, Object Oriented Software Engineering.
- Slide fornite dal professore, reperibili su e-learning.

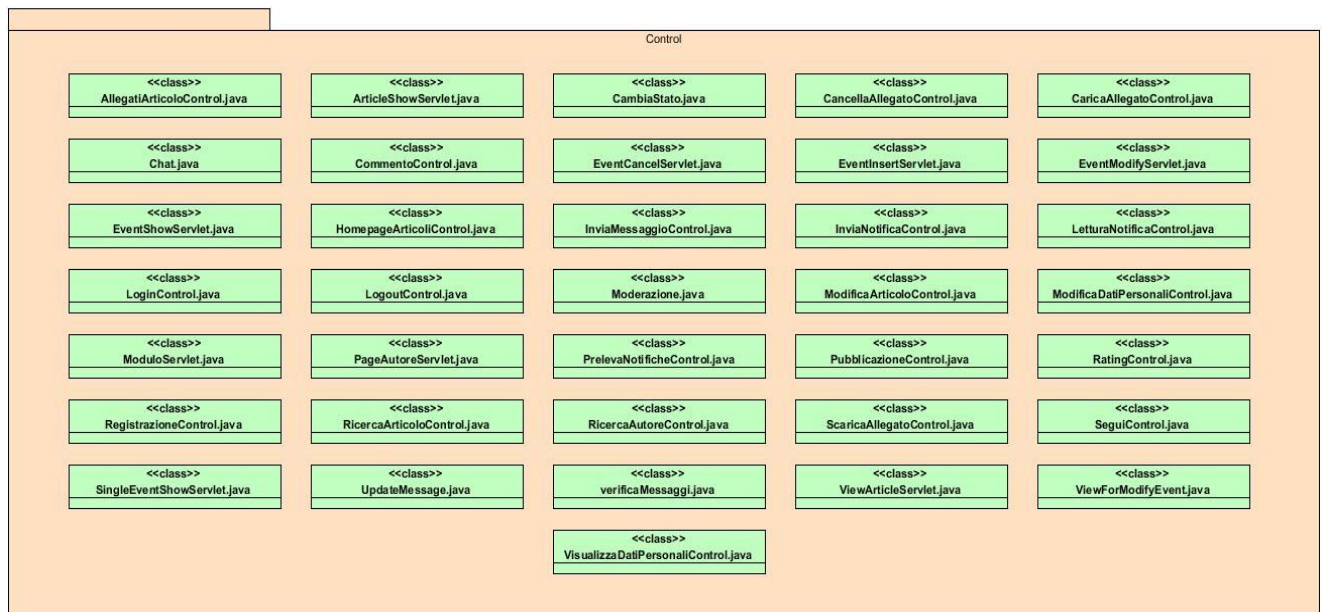
2. Packages





Classe:	Descrizione:
Utente.java	Classe che definisce un utente.
Moderatore.java	Classe che definisce un moderatore.
Autore.java	Classe che definisce un autore.
Articolo.java	Classe che definisce un articolo.
Commento.java	Classe che definisce un commento di un articolo nel sistema.
Rate.java	Classe che definisce il rating di un articolo nel sistema.
Evento.java	Classe che definisce un evento organizzato da un autore.
Allegato.java	Classe che definisce un allegato caricato da un autore.
Notifica.java	Classe che definisce una notifica.
Messaggio.java	Classe che definisce il messaggio inviato da un utente e ricevuto da un autore.
Conversazione.java	Classe che definisce una conversazione tra autore e utente.

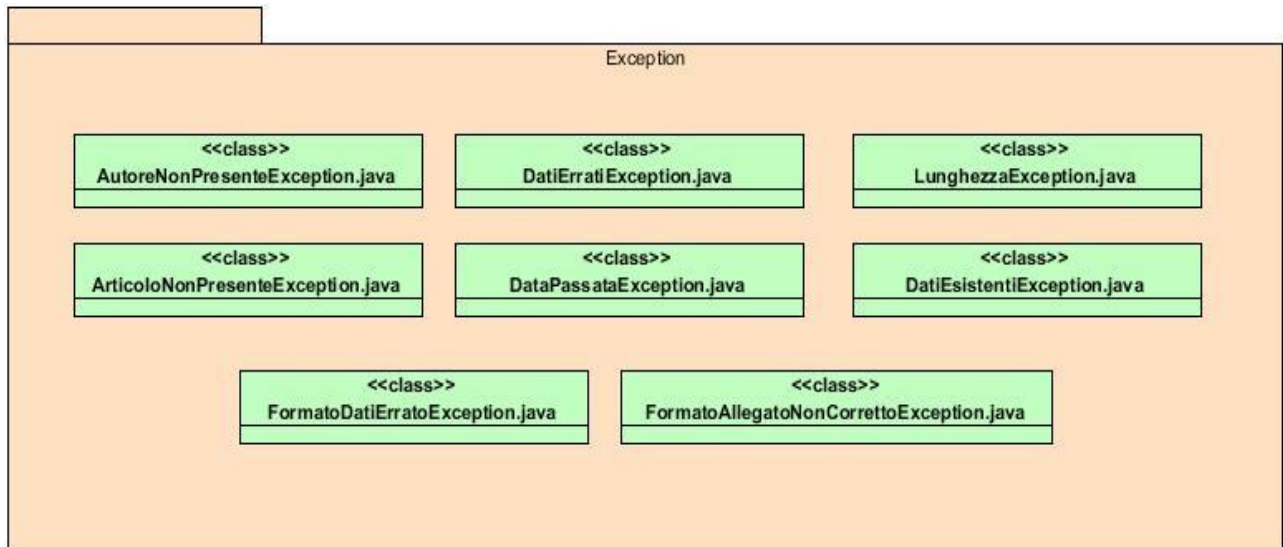
Seguace.java	Classe che definisce un seguace di un autore.
Stato.java	Classe che definisce lo stato di una notifica.



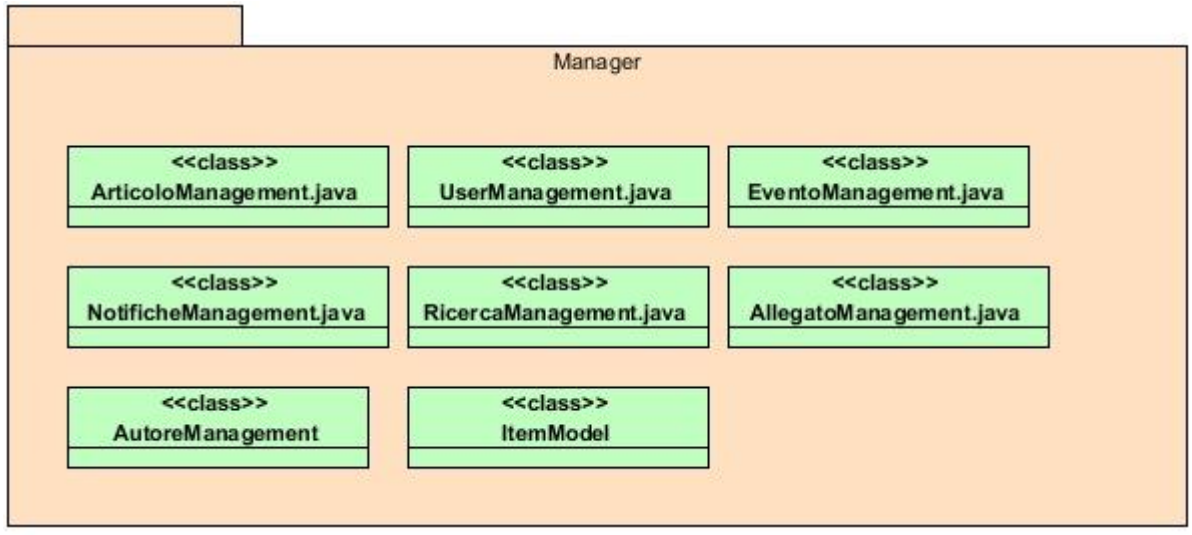
Classe:	Descrizione:
LoginControl.java	Si occupa della gestione delle informazioni relative alla fase di autenticazione
RegistrazioneControl.java	Si occupa della gestione delle informazioni per poter effettuare la registrazione
ModerazioneControl.java	Si occupa di gestire le funzionalità relative alla moderazione dell'articolo.
PubblicazioneControl.java	Si occupa di gestire le informazioni e le funzionalità relativa alla pubblicazione di articoli
VisualizzaArticoloControl.java	Si occupa di gestire le informazioni da visualizzare di un articolo
CommentoControl.java	Si occupa di gestire le informazioni riguardanti i commenti lasciati dagli utenti
ContattaControl.java	Si occupa di gestire le informazioni relative al contatto di un autore
ModificaArticoloControl.java	Si occupa di gestire le informazioni per poter effettuare le modifiche di un articolo
SeguiAutoreControl.java	Si occupa di gestire le informazioni relative a quali autori vengono seguiti dagli utenti

RicercaControl.java	Si occupa di gestire le informazioni relative alla ricerca di autori e articoli da parte di utenti
OrganizzaEventoControl.java	Si occupa di gestire le informazioni sugli eventi dagli autori in maniera da renderli visibili sulla piattaforma
CaricaAllegatoControl.java	Si occupa di gestire le informazioni riguardo il caricamento di allegati all'interno di un articolo o di un evento.
ScaricaAllegatiControl.java	Si occupa di rendere disponibili le informazioni relative agli allegati all'utente
CancellaControl.java	Si occupa di effettuare la rimozione delle informazioni sugli allegati di un articolo
ModificaDatiPersonaliControl.java	Si occupa di gestire le informazioni relative alla modifica dei dati personali degli utenti della piattaforma
RatingControl.java	Si occupa di gestire le informazioni relative alla valutazione dell'utente in merito ad un articolo
LogoutControl.java	Si occupa di gestire le informazioni dell'utente durante la fase di terminazione delle sue attività
CancellaEventoControl.java	Si occupa di gestire le informazioni sull'evento che deve essere cancellato.
VisualizzaEventoControl.java	Si occupa di gestire le informazioni sugli eventi che voglio essere visualizzati dall'utente
VisualizzaDatiPersonaliControl.java	Si occupa di gestire le informazioni relative ai dati personali che verranno visualizzate all'utente
VisualizzaPaginaAutoreControl.java	Si occupa di gestire quelle che sono le informazioni relative all'autore e ai suoi articoli durante la fase di visualizzazione della pagina dell'autore stesso.
ModificaEventoControl.java	Si occupa di gestire le informazioni relative agli eventi per poterne effettuare la modifica
VisualizzaCommentiControl.java	Si occupa di gestire le informazioni relative ai commenti di un articolo durante la fase di visualizzazione
NotificaControl.java	Si occupa di visualizzare le informazioni relative alle notifiche ricevute da un autore
VisualizzaRatingControl.java	Si occupa di gestire le informazioni relative alle valutazioni degli utenti durante la visualizzazione di esse

VisualizzaAllegatiControl.java	Si occupa reindirizzare alla pagina per visualizzare le informazioni relative all'allegato dell'articolo
---------------------------------------	--

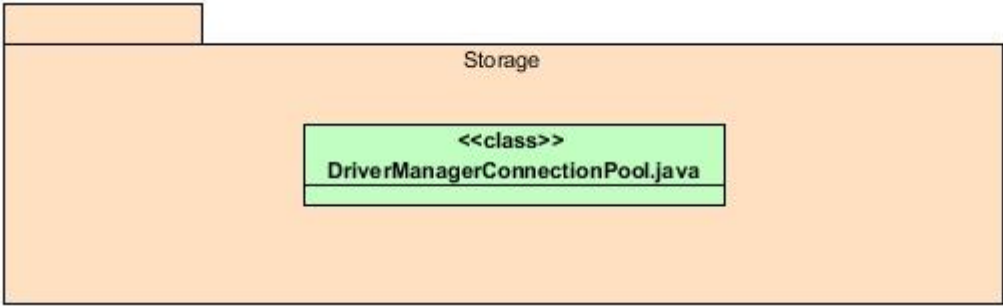


Classe:	Descrizione:
AutoreNonPresenteException.java	Eccezione che viene lanciata dopo aver eseguito una ricerca, se l'autore non è presente.
DatiErratiException.java	Eccezione che viene lanciata quando i dati di accesso sono errati.
LunghezzaException.java	Eccezione che viene lanciata quando la lunghezza del testo è troppo lunga.
ArticoloNonPresenteException.java	Eccezione che viene lanciata dopo aver eseguito una ricerca, se l'articolo non è presente.
DataPassataException.java	Eccezione che viene lanciata quando si inserisce una data precedente al giorno corrente.
DatiEsistentiException.java	Eccezione che viene lanciata quando i dati sono già presenti.
FormatoDatiErratoException.java	Eccezione che viene lanciata quando il formato dei dati è errato.
FormatoAllegatoNonCorrettoException.java	Eccezione che viene lanciata quando il formato dell'allegato non è corretto.



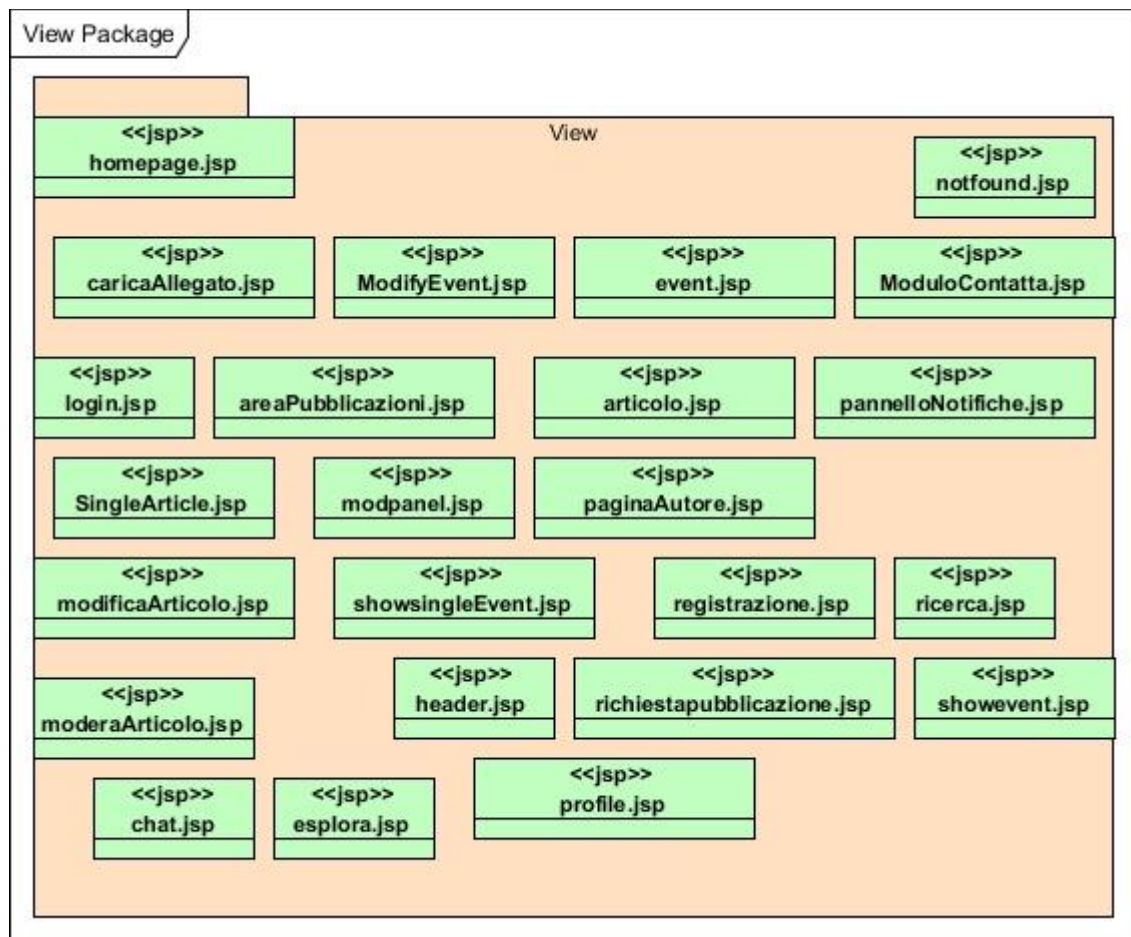
Classe:	Descrizione:
ArticoloManagement.java	Gestisce tutte le funzionalità per la pubblicazione/moderazione/modifica dell’articolo.
UserManagement.java	Gestisce tutte le funzionalità per la gestione delle informazioni utente.
EventoManagement.java	Gestisce tutte le funzionalità per la gestione degli eventi.
NotificheManagment.java	Gestisce tutte le funzionalità per gestire le notifiche.
RicercaManagement.java	Gestisce tutte le funzionalità per la ricerca.
AllegatoManagement.java	Gestisce tutte le funzionalità per per la gestione degli allegati.

Storage package



Classe:	Descrizione:
DriverManagerConnectionPool.java	Si occupa di gestire il pool delle connessioni al database.

View package



Classe:	Descrizione:
homepage.jsp	Gestisce la visualizzazione della Homepage
profile.jsp	Gestisce la visualizzazione delle informazioni del profilo utente
modPanel.jsp	Gestisce la visualizzazione del pannello di controllo del moderatore
caricaAllegato.jsp	Gestisce la visualizzazione della pagina per caricare gli allegati
areaPubblicazioni.jsp	Gestisce la visualizzazione degli articoli pubblicati da parte di un autore
organizzaEvento.jsp	Gestisce la visualizzazione della pagina per creare un evento

rifiutoArticolo.jsp	Gestisce la visualizzazione della pagina per rifiutare un articolo
profiloAutore.jsp	Gestisce la visualizzazione della pagina autore
articolo.jsp	Gestisce la visualizzazione della pagina articolo
modificaArticolo.jsp	Gestisce la visualizzazione della pagina per la modifica di un articolo
richiestaPubblicazione.jsp	Gestisce la visualizzazione della pagina per inviare la pubblicazione di un articolo.
areaModeraPubblicazione.jsp	Gestisce la visualizzazione della pagina per iniziare la moderazione di un articolo presenti nella lista degli articoli in attesa di moderazione
Ricerca.jsp	Gestisce la visualizzazione della pagina per mostrare gli articoli o autori trovati
Registrazione.jsp	Gestisce la visualizzazione della pagina per registrare un'utente nel sistema
login.jsp	Gestisce la visualizzazione della pagina per permettere l'accesso ad un utente
moderaArticolo.jsp	Gestisce la visualizzazione della pagina per permettere la moderazione di un articolo
showSingleEvent.jsp	Visualizza l'evento
modPanel.jsp	Visualizza il pannello di controllo del moderatore
NotFound.jsp	Pagina per gestire gli errori
Modifiyevent.jsp	Permettere di modificare un evento
Event.jsp	Visualizza il form per l'inserimento
ModuloContatta.jsp	Visualizza il form per contattare l'autore
pannelloNotifiche.jsp	Visualizza le notifiche
paginaAutore.jsp	Visualizza la pagina dell'autore

modificaArticolo.jsp	Form per modificare l'articolo
richiestaPubblicazione.jsp	Form che permette di creare un articolo e farne richiesta di pubblicazione
Header.jsp	Parte superiore dell'homepage
Chat.jsp	Visualizza il form per inviare e ricevere messaggi
Esplora.jsp	Visualizza tutti gli articoli resi pubblici nel sistema
Profile.jsp	Permette la modifica del profilo di utenti, autori e mod
SingleArticle.jsp	Permette la visualizzazione dell'articolo

3. Interfaccia delle Classi

Ogni metodo e ogni classe sarà opportunamente descritta nella documentazione "javadoc" allegata.

4. Design Patterns

Model-view-controller MVC, è un pattern architetturale molto diffuso nello sviluppo di sistemi software, in particolare nell'ambito della programmazione orientata agli oggetti, in grado di separare la logica di presentazione dei dati dalla logica di business. Il componente centrale del MVC, il modello, cattura il comportamento dell'applicazione in termini di dominio del problema, indipendentemente dall'interfaccia utente. Il modello gestisce direttamente i dati, la logica e le regole dell'applicazione. Una vista può essere una qualsiasi rappresentazione in output di informazioni, come un grafico o un diagramma. Sono possibili viste multiple delle stesse informazioni, come ad esempio un grafico a barre per la gestione e la vista tabellare per l'amministrazione. La terza parte, il controller, accetta l'input e lo converte in comandi per il modello e/o la vista.