

Progetto: 7Movies	Versione: 1.1
Documento: Object Design Document	Data: 18/01/2020

Università degli Studi di Salerno

Corso di Ingegneria del Software

7Movies

Object Design Document
Versione 1.1



Data: 18/01/2020

Progetto: 7Movies	Versione: 1.1
Documento: Object Design Document	Data: 18/01/2020

Coordinatore del progetto:

Nome	Matricola

Partecipanti:

Nome	Matricola
Mario Castigliano	0512104718
Luca Siviero	0512104862

Scritto da:	Mario Castigliano, Luca Siviero
--------------------	---------------------------------

Revision History

Data	Versione	Descrizione	Autore
16/01/20	1.0	Prima stesura	Mario Castigliano
18/01/20	1.1	Revisione e finalizzazione	L.Siviero, M. Castigliano

Progetto: 7Movies	Versione: 1.1
Documento: Object Design Document	Data: 18/01/2020

1. INTRODUZIONE

1.1. Object Design Trade-off

Nella fase di Object Design sorgono diversi compromessi di progettazione ed è necessario stabilire quali punti rispettare e quali rendere opzionali. Per quanto riguarda la realizzazione del sistema 7Movies sono stati individuati i seguenti trade-off:

Efficienza vs Memoria: considerando il tipo di informazioni che la piattaforma deve manipolare si è scelto di preferire una migliore e più sicura gestione della memoria a discapito di un contenuto (e quindi misurato) calo di efficienza. Alcune risorse non sono infatti caricate localmente, bensì importate da fonti terze. Questo ovviamente rallenterà il caricamento di tali risorse (quali sono le locandine dei film ed alcuni frame sample), ma consentirà agli admin di manipolare più facilmente il catalogo (dovendo interagire solo con link e non con file locali).

La scelta di utilizzare un database è scaturita dai diversi vantaggi che ne derivano:

- Gestione consistente dei dati,
- Tempo di risposta basso rispetto all'utilizzo di un file system,
- Accesso veloce e concorrente ai dati;

Interfaccia vs Easy-use: L'interfaccia si presenta semplice e intuitiva grazie ad uno stile minimalista che punta all'essenziale e ai form che guidano gli utenti nell'eseguire alcune operazioni previste dal sistema (come registrazione, login, lasciare feedback etc).

Sicurezza vs Prestazioni: Nel nostro sistema ogni richiesta del client viene validata attraverso l'uso di sessioni, poiché è necessario effettuare tali controlli per rispettare i requisiti iniziali.

1.2. Linee guida per la documentazione

Nell'implementazione del sistema, i programmatori dovranno attenersi alle linee guida di seguito definite.

1.2.1. Classi e interfacce Java

Nella scrittura di codice per le classi Java ci si atterrà allo standard [CCJPL] nella sua interezza, con particolare attenzione a:

1. Convenzione sui nomi;
2. Struttura dei file;
3. Accesso alle variabili;
4. Commenti speciali
5. Utilizzo degli spazi bianchi;

Progetto: 7Movies	Versione: 1.1
Documento: Object Design Document	Data: 18/01/2020

Si praticano, inoltre, le seguenti restrizioni e variazioni:

1. All'inizio di ogni file devono essere presenti alcune informazioni strutturate come segue:

```
/*
NomeClasse
Breve descrizione della classe
*/
```

2. Tutte le variabili dovrebbero essere private. (private)
3. Tutte le variabili che non vengono mai modificate dovrebbero essere dichiarate come costanti (final).
4. Si devono usare le tabulazioni per l'indentazione.

1.2.2. Base di dati

Le tabelle della base di dati dovrebbero rispettare la terza forma normale di Codd (3NF). Ove ciò non si verifichi, tale fatto deve essere riportato e motivato nella documentazione della base di dati.

Le scelte di gestione nel trattamento dell'integrità referenziale devono essere riportate e motivate nella documentazione della base di dati.

1.2.3. Pagine lato server (JSP)

Le pagine JSP devono, quando eseguite, produrre, in ogni circostanza, un documento conforme allo standard HTML versione 5.

Le parti Java delle pagine devono aderire alle convenzioni per la codifica in Java, con le seguenti puntualizzazioni:

1. Il tag di apertura (<%) è seguito immediatamente dalla fine della riga;
2. Il tag di chiusura (%>) si trova all'inizio di una riga;
3. È possibile emendare alle due regole precedenti, se il corpo del codice Java consiste in una singola istruzione:

```
<! Accettabile >
<% for (String par : paragraphs) {%>
<p class='item'> <% out.print(par); %> </p>
<% } %>

<! Non accettabile >
<p class='item'> <% List<String> paragraphs = getParagraphs();
out.print(paragraphs.get(i++));%></p>
```

Progetto: 7Movies	Versione: 1.1
Documento: Object Design Document	Data: 18/01/2020

1.2.4. Pagine HTML

Le tabelle della base di dati dovrebbero rispettare la terza forma normale di Codd (3NF). Ove ciò non si verifichi, tale fatto deve essere riportato e motivato nella documentazione della base di dati.

Le scelte di gestione nel trattamento dell'integrità referenziale devono essere riportate e motivate nella documentazione della base di dati. Le pagine HTML, statiche e dinamiche, devono essere totalmente aderenti allo standard HTML versione 5.

Inoltre, il codice HTML statico deve utilizzare l'indentazione, per facilitare la lettura, secondo le seguenti regole:

1. Un'indentazione consiste in una tabulazione;
2. Ogni tag deve avere un'indentazione maggiore del tag che lo contiene
3. Ogni tag di chiusura deve avere lo stesso livello di indentazione del corrispondente tag di apertura;

```

<!-- Accettabile -->
<div>
  <span>
    <ol>
      <li>Uno</li>
      <li>Due</li>
    </ol>
  </span>
</div>
<!-- Non accettabile -->
<div>
  <span>
    <ol>
      <li>Uno</li>
      <li>
        Due</li>
      </li>
    </ol>
  </span>
</div>

```

4. I tag di commento devono seguire le stesse regole che si applicano ai tag normali;

Progetto: 7Movies	Versione: 1.1
Documento: Object Design Document	Data: 18/01/2020

Script JavaScript

Gli script che svolgono funzioni distinte dal rendering della pagina dovrebbero essere collocati in file dedicati.

Il codice Javascript deve seguire le stesse convenzioni per il layout e i nomi del codice Java. I documenti Javascript devono essere iniziati da un commento analogo a quello presente nei file Java.

Le funzioni Javascript devono essere documentate in modo analogo ai metodi Java.

Gli oggetti Javascript devono essere preceduti da un commento in stile Javadoc, che segue il seguente formato:

```
/**
 * Descrizione breve
 * Eventuale ulteriore descrizione
 *
 * Metodo nomeMetodo1
 * Descrizione breve
 * Eventuale ulteriore descrizione
 *
 * Metodo nomeMetodo2
 * Descrizione breve
 * Eventuale ulteriore descrizione
 *
 * ...
 */
function ClasseX(a, b, c) {
```

1.2.5. Database SQL

Le tabelle della base di dati dovrebbero rispettare la terza forma normale di Codd (3NF). Ove ciò non si verifichi, tale fatto deve essere riportato e motivato nella documentazione della base di dati.

Le scelte di gestione nel trattamento dell'integrità referenziale devono essere riportate e motivate nella documentazione della base di dati.

I nomi delle tabelle devono seguire le seguenti regole:

- devono essere costituiti da sole lettere minuscole;
- il nome deve essere un sostantivo singolare tratto dal dominio del problema ed esplicativo del contenuto.

I nomi dei campi devono seguire le seguenti regole:

- devono essere costituiti da sole lettere minuscole;

Progetto: 7Movies	Versione: 1.1
Documento: Object Design Document	Data: 18/01/2020

- il nome deve essere un sostantivo singolare tratto dal dominio del problema ed esplicativo del contenuto.

1.3. Definizioni, acronimi e abbreviazioni

CCJPL: Code Conventions For the Java Programming.

RAD: Requirements Analysis Document.

SDD: System Design Document.

ODD: Object Design Document.

SQL: Structured Query Language.

JSP: Java Server Pages.

HTML: HyperText Markup Language.

CSS: Cascading Style Sheets.

3NF: Terza forma normale.

COTS: Component Off-The-Shelf.

API: Application Programming Interface.

1.4. Riferimenti

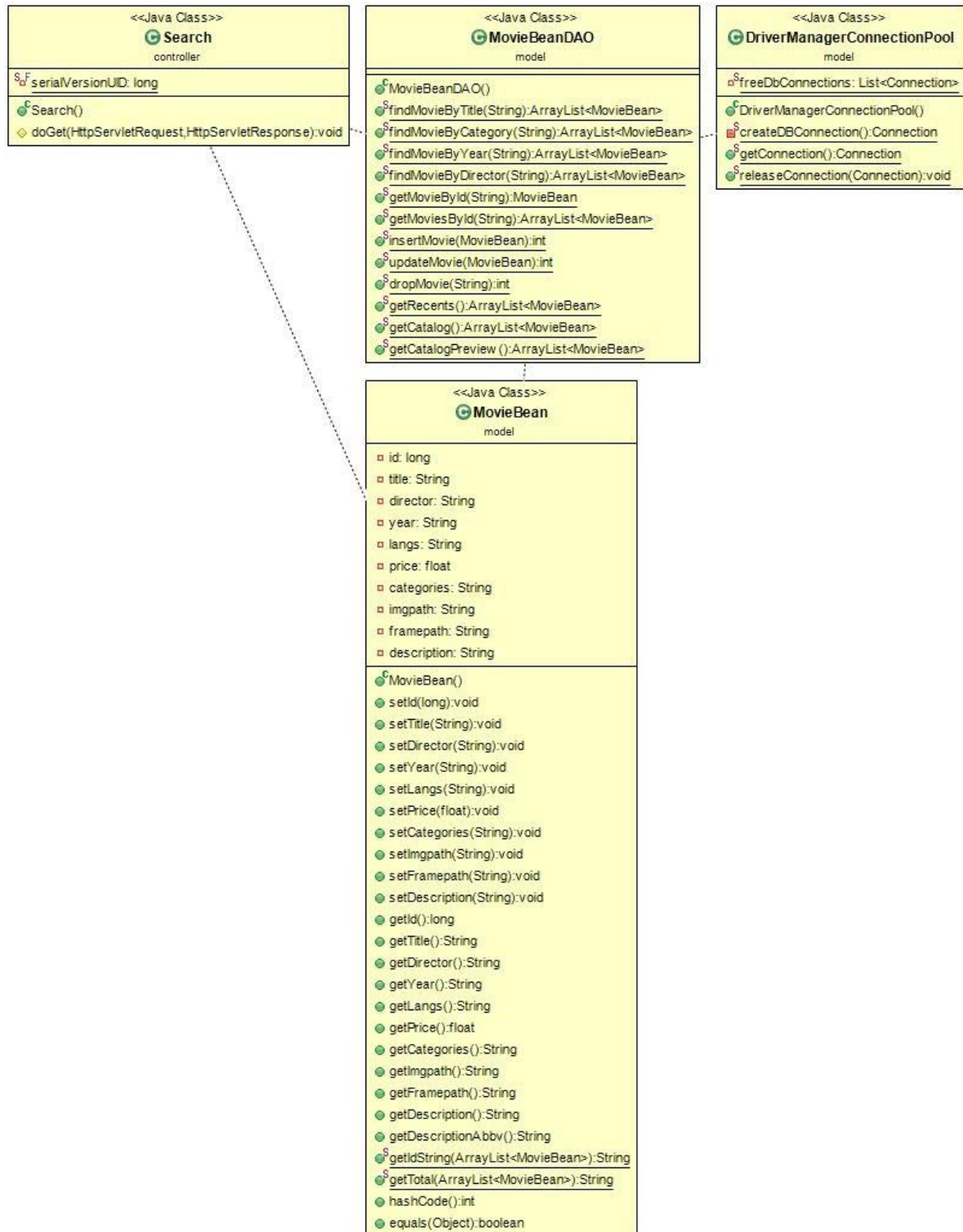
Il presente ODD riferisce alle ultime versioni dei precedenti documenti rilasciati su Hardship, in particolare:

7Movies_RAD;
7Movies_SDD;

Progetto: 7Movies	Versione: 1.1
Documento: Object Design Document	Data: 18/01/2020

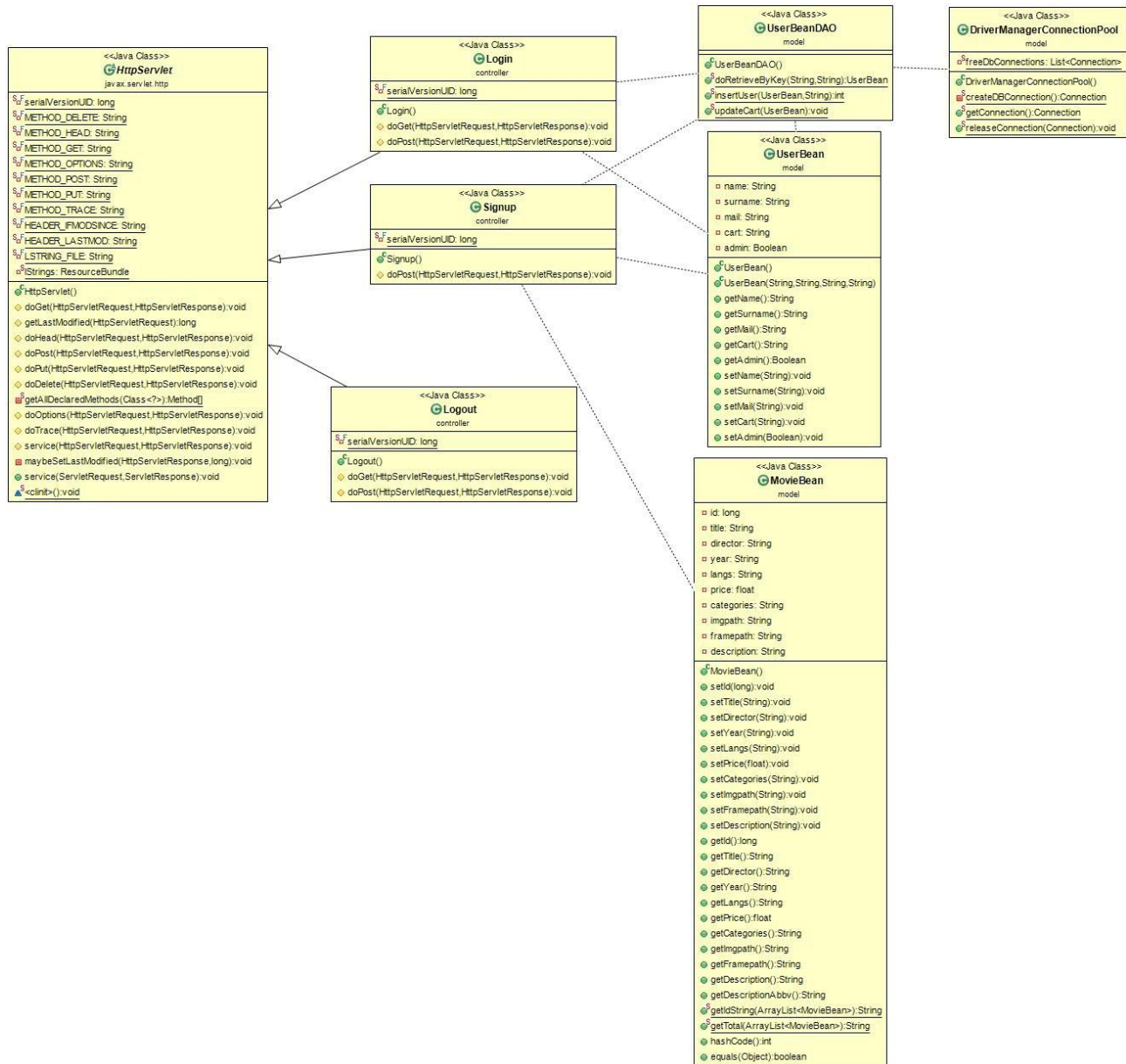
2. PACKAGING E CLASS INTERFACES

2.1 Gestione sistema



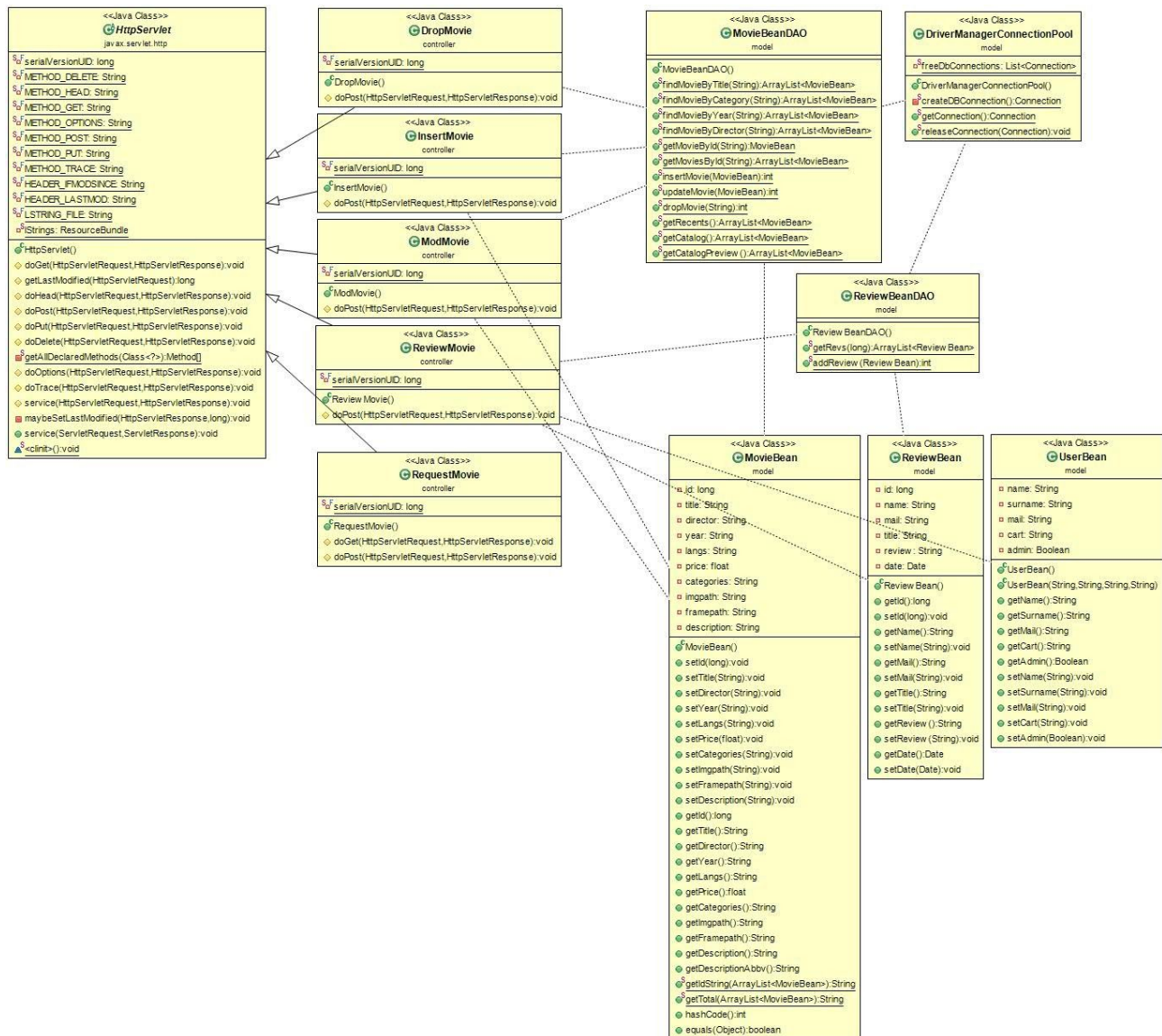
Progetto: 7Movies	Versione: 1.1
Documento: Object Design Document	Data: 18/01/2020

2.2 Gestione Utente



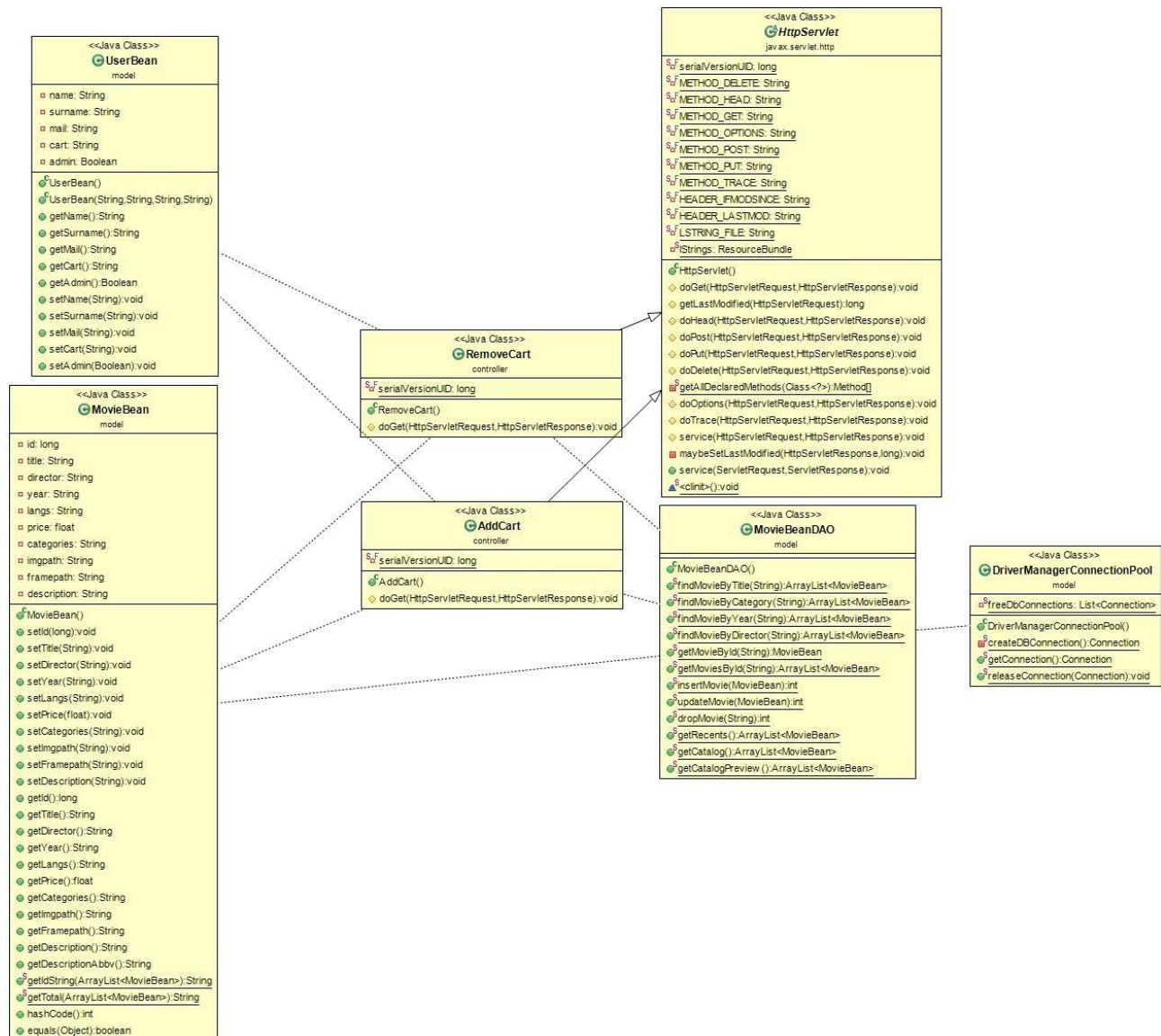
Progetto: 7Movies	Versione: 1.1
Documento: Object Design Document	Data: 18/01/2020

2.3 Gestione Film



Progetto: 7Movies	Versione: 1.1
Documento: Object Design Document	Data: 18/01/2020

2.4 Gestione Carrello



Progetto: 7Movies	Versione: 1.1
Documento: Object Design Document	Data: 18/01/2020

3. DOCUMENTAZIONE DEL RIUSO

3.1. Component Off-The-Shelf (COTS)

La piattaforma 7Movies farà uso di componenti off-the-shelf, componenti software disponibili per facilitare la creazione del progetto, allo scopo di contenere i costi di sviluppo.

Il sistema 7Movies si servirà di un DBMS per gestire i dati persistenti; in particolare sarà utilizzato PhpMyAdmin.

Andremo ad utilizzare, inoltre, Bootstrap, un framework open source che contiene una raccolta di strumenti per la creazione di siti e applicazioni per il Web. Esso offre modelli di progettazione basati su HTML, CSS e JavaScript. La scelta di usufruire di tale framework deriva prevalentemente dalla necessità di garantire l'uso della piattaforma su qualsiasi device grazie all'utilizzo di responsive design.

4. GLOSSARIO

3NF: Una base dati è in 3NF (terza forma normale) se è in 2NF e tutti gli attributi non-chiave dipendono solo dalla chiave, ossia non esistono attributi che dipendono da altri attributi non-chiave. Tale normalizzazione elimina la dipendenza transitiva degli attributi dalla chiave.

Bootstrap: Bootstrap è una raccolta di strumenti liberi per la creazione di siti e applicazioni per il Web. Essa contiene modelli di progettazione basati su HTML e CSS, sia per la tipografia, che per le varie componenti dell'interfaccia, come moduli, pulsanti e navigazione, così come alcune estensioni opzionali di JavaScript.

COTS: L'espressione componente COTS o componente OTS, in inglese (Commercial) Off-the-Shelf component, si riferisce a componenti hardware e software disponibili sul mercato per l'acquisto da parte di aziende di sviluppo interessate a utilizzarli nei loro progetti.

DBMS: Database Management System, software progettato per la creazione e la manipolazione di database.

Design Pattern: Un design pattern (traducibile in lingua italiana come schema progettuale, schema di progettazione, schema architettuale), è un concetto che può essere definito "una soluzione progettuale generale ad un problema ricorrente". Si tratta di una descrizione o modello logico da applicare per la risoluzione di un problema che può presentarsi in diverse situazioni durante le fasi di progettazione e sviluppo del software.

HTML: L'HyperText Markup Language (HTML; traduzione letterale: linguaggio a marcatori per ipertesti), in informatica è il linguaggio di markup solitamente usato per la formattazione e impaginazione di documenti ipertestuali disponibili nel World Wide Web sotto forma di pagine web.

Javadoc: Javadoc è un applicativo incluso nel Java Development Kit della Sun Microsystems, utilizzato per la generazione automatica della documentazione del codice sorgente scritto in

Progetto: 7Movies	Versione: 1.1
Documento: Object Design Document	Data: 18/01/2020

linguaggio Java.

ODD: Object Design Document.

RAD: Requirements Analysis Document, documento che tratta nel dettaglio l'analisi dei requisiti.

SDD: System Design Document, documento che tratta nel dettaglio della progettazione del sistema, e dei suoi obiettivi.

Tag: Il tag è una parola chiave o un termine associato a un'informazione (un'immagine, una mappa geografica, un post, un video clip ...), che descrive l'oggetto rendendo possibile la classificazione e la ricerca di informazioni basata su parole chiave.

Trade-off: In economia un trade-off (o trade off) è una situazione che implica una scelta tra due o più possibilità, in cui la perdita di valore di una costituisce un aumento di valore in un'altra.