



# **Università degli Studi di Salerno**

**Corso di Ingegneria del Software**

## **PlanetVideo System Design Document Versione 1.2**



Data: 09/03/2020

Progetto: PlanetVideo	Versione: 1.2
Documento: System Design Document	Data: 09/03/2020

### Coordinatore del progetto:

Nome	Matricola

### Partecipanti:

Nome	Matricola
Contente Antonio	0512103870
Mecca Vincenzo	0512103528

<b>Scritto da:</b>	Contente Antonio Mecca Vincenzo
--------------------	------------------------------------

## Revision History

Data	Versione	Descrizione	Autore
09/03/2020	1.0	Prima stesura	Contente Antonio Mecca Vincenzo
12/03/2020	1.1	Modifiche	Contente Antonio
16/03/2020	1.2	Modifiche e revisione	Mecca Vincenzo

# Indice

1.	INTRODUZIONE .....	4
1.1.	Scopo del sistema .....	4
1.2.	Obiettivi di design.....	4
1.3.	Definizioni, acronimi e abbreviazioni.....	5
1.4.	Riferimenti.....	5
2.	SISTEMA SOFTWARE ATTUALE.....	5
3.	SISTEMA SOFTWARE PROPOSTO .....	6
3.1.	Panoramica .....	6
3.1.1.	Vantaggi del modello MVC.....	6
3.1.2.	Svantaggi del modello MVC .....	6
3.2.	Decomposizione in sottosistemi .....	7
3.3.	Mapping hardware/software .....	10
3.4.	Gestione dei dati persistenti .....	10
3.5.	Controllo degli accessi e sicurezza .....	11
3.6.	Controllo globale del software .....	11
3.7.	Condizioni di boundary.....	11
4.	SERVIZI DEI SOTTOSISTEMI .....	13
4.1.	Gestione utente .....	13
4.2.	Gestione catalogo.....	13
4.3.	Gestione Content creator.....	13

# 1. INTRODUZIONE

## 1.1. Scopo del sistema

L'obiettivo principale di questo progetto è la realizzazione di un sito di video streaming on demand. Il servizio sarà accessibile esclusivamente tramite registrazione al sito e sottoscrizione da parte dell'utente di un abbonamento. Il costo dell'abbonamento è di 11,99 euro al mese.

Solo per i nuovi utenti, è possibile usufruire di un periodo di prova gratuito, della durata di un mese.

Al giorno d'oggi vi è un'ampissima scelta per l'utente nel campo della cinematografia e in quello delle serie tv. Questo ha portato alla nascita di tantissimi servizi atti allo scopo di fornire intrattenimento. Tra questi non possiamo non citare Netflix, un servizio di streaming che consente agli abbonati di guardare una varietà di serie tv, film, documentari, ecc. su diverse tipologie di dispositivi (pc, smart tv, tablet e smartphone). Un altro servizio parimenti significativo ci viene fornito da Infinity, il primo servizio di streaming on demand in Italia che mette a disposizione un ricchissimo catalogo di film, cartoni, Serie TV, programmi e fiction, da guardare su qualsiasi dispositivo abilitato che abbia una connessione internet. Come questi se ne potrebbero citare molti altri.

Lo scopo dunque è quello di inserirsi in questo fiorente panorama con un occhio di riguardo per l'usabilità. Il nostro target vuole essere quanto più ampio possibile e per ottenere ciò, la facilità di utilizzo e di fruizione dei vari materiali reperibili sul nostro sito deve essere ed è una priorità assoluta.

Una peculiarità del nostro sistema è quello di mettere a disposizione un account amministratore per le case cinematografiche o, in generale, per i creatori di contenuti multimediali, con lo scopo di poter caricare sul nostro sito i propri contenuti originali e aumentare in questo modo la scelta dell'utente finale, oltre a permettere ai creatori di avere un nuovo canale di diffusione.

## 1.2. Obiettivi di design

<b>Tempi di risposta</b>	Il sistema deve garantire tempi di risposta brevi per ogni funzionalità. Essendo interamente web based, i tempi di risposta dipendono, però, anche dalla qualità della connessione a disposizione dell'utente.
<b>Memoria</b>	Il sistema avrà a disposizione un database relazionale (MySQL) per la memorizzazione dei dati.
<b>Robustezza</b>	Il sistema deve reagire in modo dinamico agli input errati dell'utente. Alert e messaggi di errore verranno visualizzati per avvertire l'utente di eventuali dati inseriti non validi o mancanti.
<b>Sicurezza</b>	Ogni utente del sistema è provvisto di username e password, scelti durante la fase di registrazione. Tramite le due credenziali, è possibile accedere al sistema e alle sole funzionalità associate alla propria "categoria" di utente.
<b>Portabilità</b>	Il sistema è accessibile da qualsiasi dispositivo dotato di connessione a Internet e di un browser web. Il sito è responsive, per cui visibile in modo corretto su qualsiasi browser e dispositivo.
<b>Usabilità</b>	Il sistema sarà facilmente utilizzabile, grazie a un'interfaccia grafica intuitiva e user-friendly.

<b>Affidabilità</b>	Il sistema garantirà l'affidabilità, intesa come corretta gestione delle funzionalità. Le componenti del sistema devono essere completamente funzionanti e almeno l'80% di esse dovranno essere testate.
<b>Disponibilità</b>	Il sistema deve essere disponibile 24 ore su 24, implementando un server sempre attivo.
<b>Modificabilità</b>	Il sistema deve garantire una facile modificabilità, grazie a un'attenta e chiara stesura del codice.

### ***1.3. Definizioni, acronimi e abbreviazioni***

**JDBC:** Java DataBase Connectivity

**RAD:** Requirements Analysis Document

**SDD:** System Design Document

**MVC:** Model-View-Controller

**MySQL:** database Open Source basato sul linguaggio SQL

**DBMS:** Database Management System, Sistema di gestione del database

### ***1.4. Riferimenti***

Il materiale di riferimento utilizzato per la stesura di questo stesso documento comprende:

- Libro di testo: B.Bruegge, A.H. Dutoit, Object Oriented Software Engineering –Using UML, Patterns and Java, Prentice Hall, 3rd edition, 2009
- Slide del docente
- RAD\_PlanetVideo

## **2. SISTEMA SOFTWARE ATTUALE**

Al giorno d'oggi vi è un'ampissima scelta per l'utente nel campo della cinematografia e in quello delle serie tv. Questo ha portato alla nascita di tantissimi servizi atti allo scopo di fornire intrattenimento.

Tra questi non possiamo non citare Netflix, un servizio di streaming che consente agli abbonati di guardare una varietà di serie tv, film, documentari, ecc. su diverse tipologie di dispositivi (pc, smart tv, tablet e smartphone).

Un altro servizio parimenti significativo ci viene fornito da Infinity, il primo servizio di streaming on demand in Italia che mette a disposizione un ricchissimo catalogo di film, cartoni, Serie TV, programmi e fiction, da guardare su qualsiasi dispositivo abilitato che abbia una connessione internet.

Come questi se ne potrebbero citare molti altri.

### **3. SISTEMA SOFTWARE PROPOSTO**

#### **3.1. Panoramica**

L'architettura scelta per questo sistema è il Model-View-Controller (MVC). Essa si struttura in tre componenti, ognuno con un compito diverso all'interno del sistema:

- il model fornisce i metodi per accedere ai dati utili all'applicazione;
- la view visualizza i dati contenuti nel model e si occupa dell'interazione con l'utente;
- il controller riceve i comandi dell'utente (in genere attraverso la view) e li attua modificando lo stato degli altri due componenti.

##### **3.1.1. Vantaggi del modello MVC**

- MVC supporta lo sviluppo rapido e parallelo. Se un modello MVC viene utilizzato per sviluppare una particolare applicazione Web, è possibile che un programmatore possa lavorare su una view mentre un altro su un controller per creare la logica di business dell'applicazione web. Quindi, in questo modo, l'applicazione sviluppata utilizzando il modello MVC può essere completata tre volte più velocemente rispetto alle applicazioni sviluppate utilizzando altri modelli di sviluppo;
- Nel modello MVC, è possibile creare più view per un model. Inoltre, grazie a questo modello, la duplicazione del codice è molto limitata perché vengono separati i dati e la logica di business dalle interfacce grafiche;
- Per qualsiasi applicazione Web, l'interfaccia utente tende a cambiare molto velocemente. Vengono apportati di continuo modifiche alla propria applicazione Web, come cambiamenti nei colori, caratteri, layout dello schermo e aggiunta di nuovo supporto per dispositivi mobili (cellulari o tablet). L'aggiunta di un nuovo tipo di view è molto semplice con il modello MVC poiché la parte Model non dipende dalla parte View. Pertanto, eventuali modifiche nel Model non influenzeranno l'intera architettura;
- Il pattern MVC restituisce i dati senza applicare alcuna formattazione. Quindi, gli stessi componenti possono essere utilizzati e richiamati per l'uso con qualsiasi interfaccia.

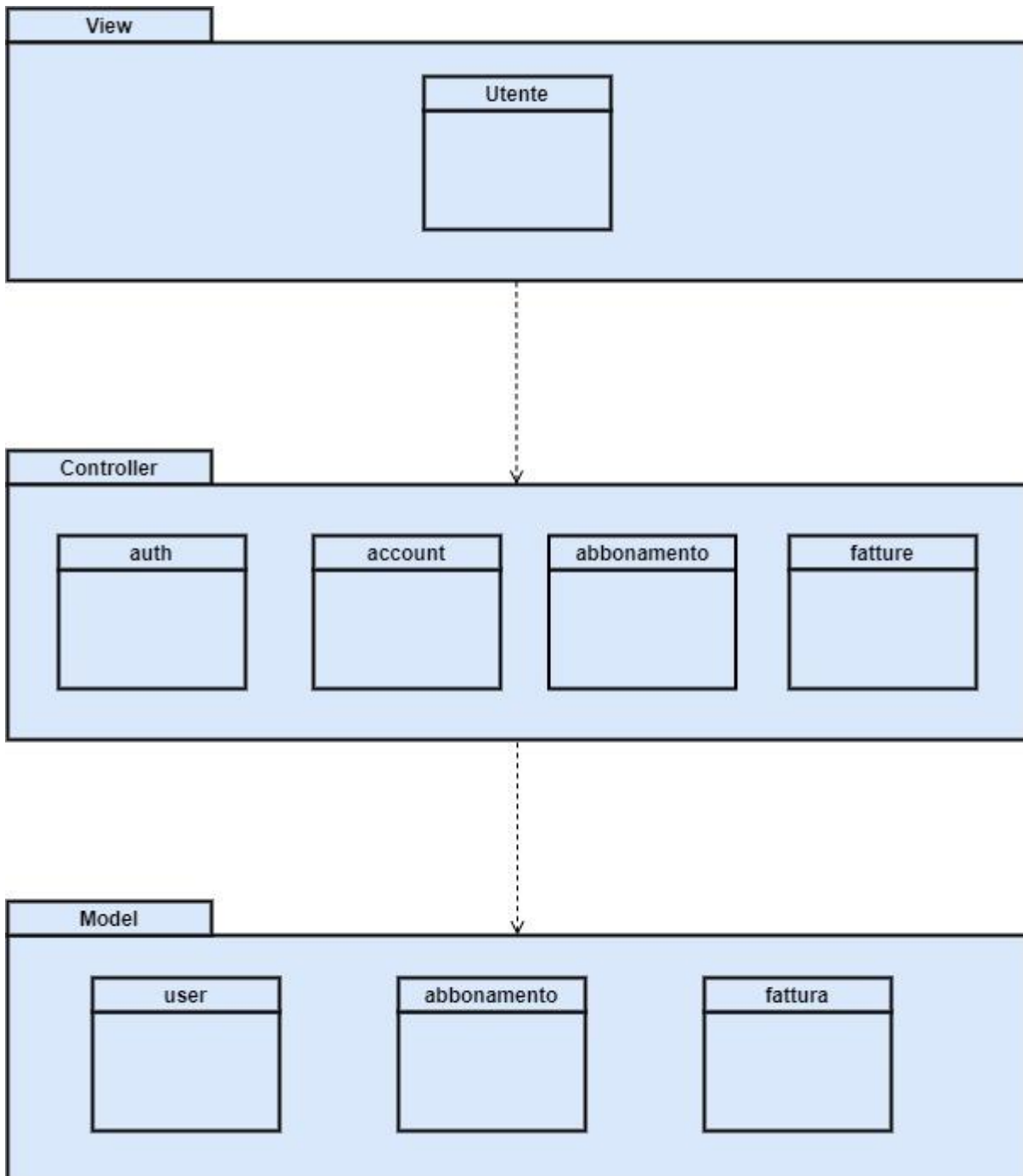
##### **3.1.2. Svantaggi del modello MVC**

- Modello di design complesso. Ha bisogno di una buona conoscenza del flusso di controllo tra view, logica di business e controller, altrimenti sarà difficile eseguire il debug;
- Se diversi team lavorano su classi UI e Model, allora devono avere lo stesso ritmo di sviluppo;
- Il processo di sviluppo isolato per la UI, la logica di business e i controller può portare a ritardi nello sviluppo dei rispettivi moduli.

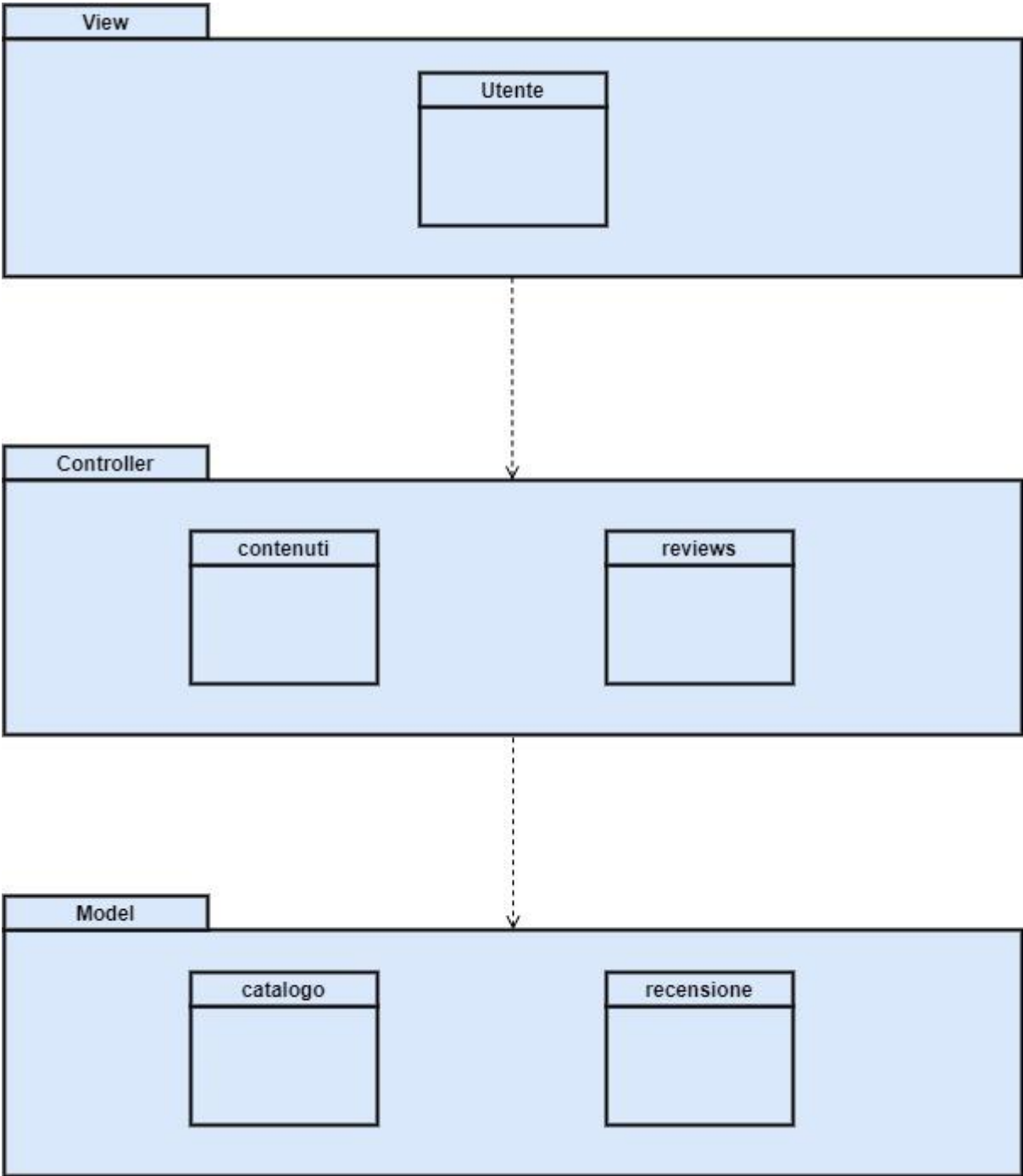
### 3.2. Decomposizione in sottosistemi

Il sistema è stato diviso in tre sottosistemi, identificati in tre gestioni: **Gestione utente**, **Gestione catalogo**, **Gestione Content creator**.

La Gestione utente si occupa dell'autenticazione dell'utente, della visualizzazione delle sue informazioni riguardanti account, abbonamento e fatture. Inoltre si occupa del rinnovo dell'abbonamento, se esso è scaduto.

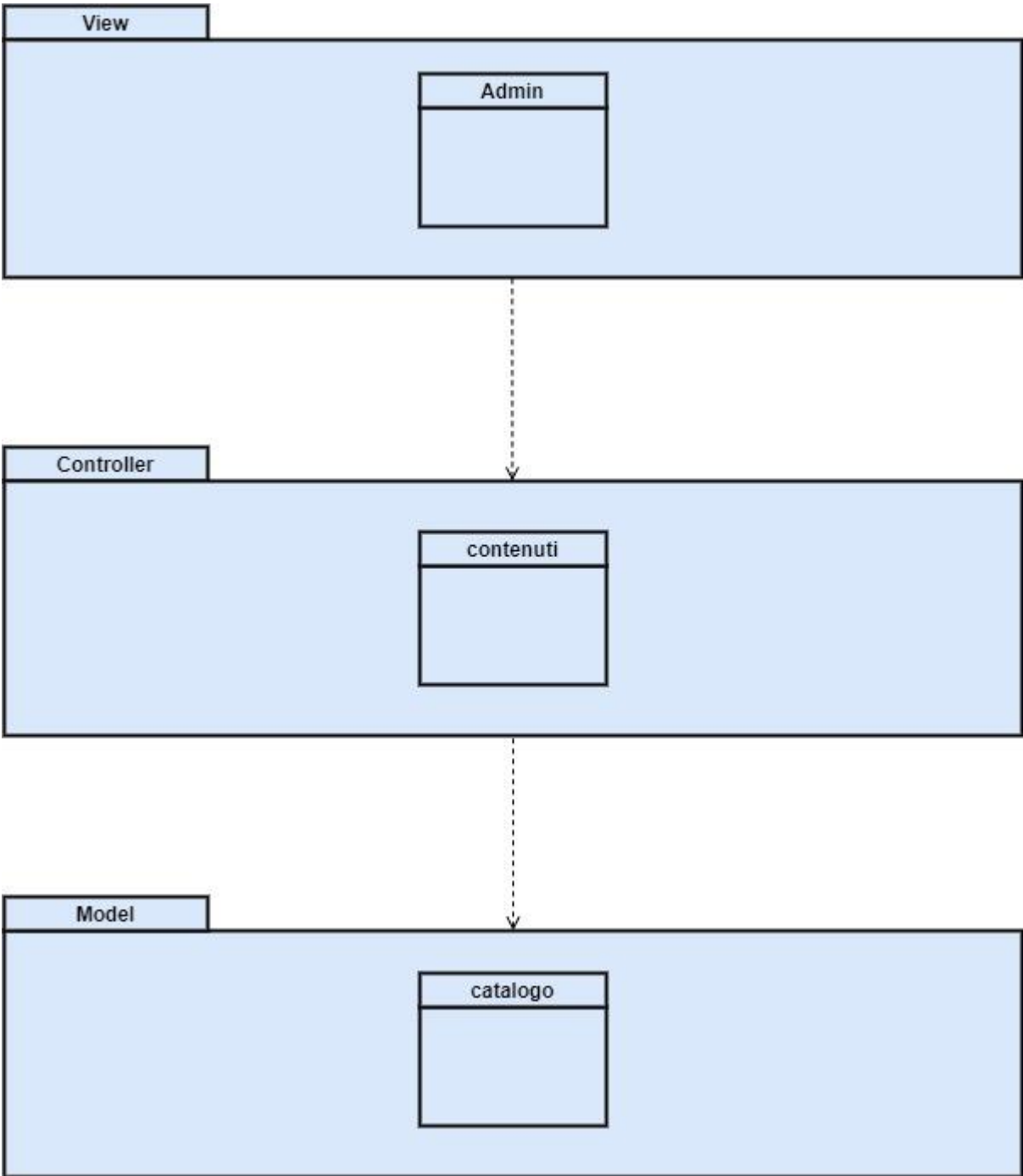


La Gestione catalogo si occupa della visualizzazione dei contenuti (film e serie tv), della visualizzazione dei dettagli di un contenuto, dell'aggiunta/rimozione di un contenuto del catalogo dalla lista preferiti. Inoltre, si occupa anche della visualizzazione delle recensioni e dell'aggiunta di una recensione.

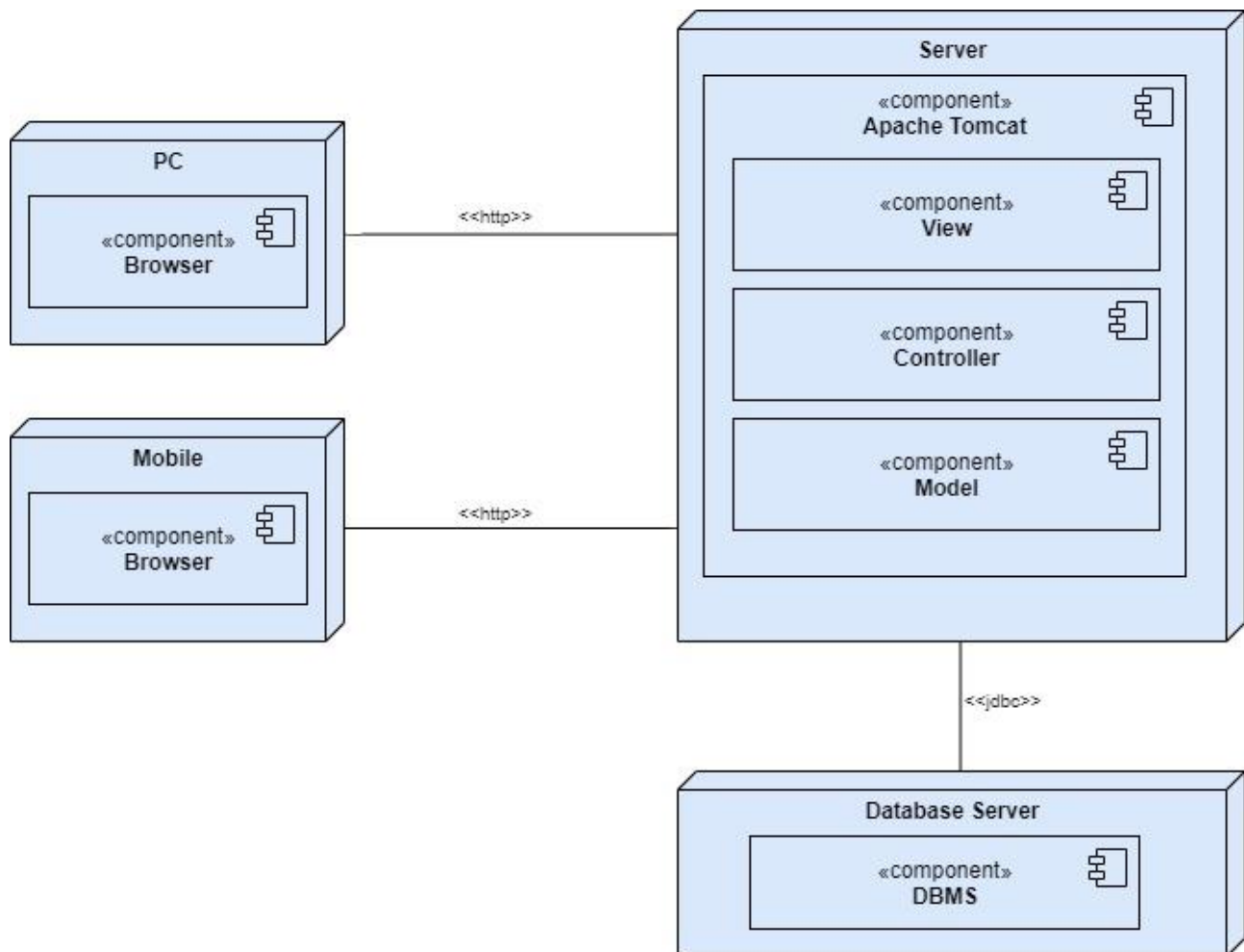




Infine, la Gestione Content creator si occupa dell'aggiunta/rimozione di un contenuto del catalogo.



### 3.3. Mapping hardware/software



Il sistema si presenta con un'architettura Client/Server. Il WebServer è rappresentato da Apache Tomcat 9 ed è situato su una singola macchina, la logica del sistema è costituita da Java Servlet mentre l'interfaccia utente è realizzata utilizzando pagine JSP (Java Servlet Page). Il Client è rappresentato dal Web Browser utilizzato dall'utente. La comunicazione tra i nodi è rappresentata da richieste e risposte http tra client e server e da query SQL con JDBC tra server e database.

### 3.4. Gestione dei dati persistenti

Per la memorizzazione delle informazioni che necessitano di essere rese persistenti, la soluzione ideale è stata l'utilizzo di un database relazionale, in modo da avere un'agevole gestione dell'accesso concorrente ai dati. Inoltre, questo approccio garantisce la consistenza, la privacy e l'integrità dei dati stessi.

La gestione dei dati persistenti è affidata al Database relazionale MySQL.

Per informazioni più dettagliate, si rimanda al documento GestioneDatiPersistenti\_PlanetVideo.

### 3.5. Controllo degli accessi e sicurezza

All'interno del sistema, gli attori hanno il permesso di eseguire diverse operazioni sui vari sottosistemi. Per schematizzare il controllo, si è deciso di usare la matrice degli accessi riportata di seguito:

<b>Attore</b>	<b>Utente registrato</b>	<b>Content creator</b>
<b>Sottosistema</b>		
<b>Gestione utente</b>	<ul style="list-style-type: none"><li>- Login</li><li>- Logout</li><li>- Visualizza info utente</li><li>- Visualizza info abbonamento</li><li>- Rinnova l'abbonamento, se scaduto</li><li>- Visualizza le fatture</li></ul>	<ul style="list-style-type: none"><li>- Login</li><li>- Logout</li><li>- Visualizza info utente</li><li>- Visualizza info abbonamento</li><li>- Rinnova l'abbonamento, se scaduto</li><li>- Visualizza le fatture</li></ul>
<b>Gestione catalogo</b>	<ul style="list-style-type: none"><li>- Visualizza i contenuti</li><li>- Visualizza dettagli contenuti</li><li>- Aggiunge contenuti alla Lista Preferiti</li><li>- Rimuove contenuti dalla Lista Preferiti</li><li>- Visualizza le recensioni</li><li>- Aggiunge nuove recensioni</li></ul>	<ul style="list-style-type: none"><li>- Visualizza i contenuti</li><li>- Visualizza dettagli contenuti</li><li>- Aggiunge contenuti alla Lista Preferiti</li><li>- Rimuove contenuti dalla Lista Preferiti</li><li>- Visualizza le recensioni</li><li>- Aggiunge nuove recensioni</li></ul>
<b>Gestione Content creator</b>		<ul style="list-style-type: none"><li>- Aggiunge nuovi contenuti</li><li>- Elimina contenuti</li></ul>

### 3.6. Controllo globale del software

L'applicazione web non ha una sequenza di operazioni prestabilita, ma è l'utente, per mezzo dell'interfaccia grafica, a scegliere le operazioni che desidera effettuare.

L'architettura è composta da un web server che rimane in ascolto, in attesa di una richiesta da parte di un web browser. Appena riceve una richiesta, questa viene smistata alla Java Servlet opportuna che si occuperà di gestirla ed elaborarla, invocando i metodi necessari sui sottosistemi coinvolti nell'operazione. Infine, l'output viene inviato al client.

### 3.7. Condizioni di boundary

Le condizioni di boundary riguardano l'accensione e lo spegnimento del sistema per quanto riguarda il lato Server.

Dal lato Client, si riferiscono agli errori di connessione al server.

Per lo startup del sistema è necessario avviare il web server, per consentire agli utenti di accedere al sistema tramite web browser.

Per terminare il sistema è necessario effettuare la procedura di terminazione del web server, dopo la quale non sarà più possibile per gli utenti accedere al sistema.

<b>Nome scenario</b>	<b>Startup del sistema</b>
<b>Attori partecipanti</b>	Federico: Amministratore
<b>Flusso di eventi</b>	<ol style="list-style-type: none"> <li>1. Federico decide di avviare il sistema e quindi clicca sul pulsante “Avvia”.</li> <li>2. Il sistema, con le opportune procedure di avvio, attiva il server e si mette in ascolto di eventuali richieste.</li> <li>3. Il sistema notifica il successo della procedura.</li> </ol>

<b>Nome scenario</b>	<b>Shutdown del sistema</b>
<b>Attori partecipanti</b>	Federico: Amministratore
<b>Flusso di eventi</b>	<ol style="list-style-type: none"> <li>1. Federico decide di terminare il sistema e quindi clicca sul pulsante “Stop”.</li> <li>2. Il sistema effettua le procedure necessarie per la terminazione del server. Da questo momento nessun utente sarà in grado di connettersi al server</li> <li>3. Il sistema notifica il successo della procedura.</li> </ol>

<b>Nome scenario</b>	<b>Startup del web server</b>
<b>Attori partecipanti</b>	Federico: Amministratore
<b>Condizione d’ingresso</b>	Federico visualizza la console per effettuare l’avvio del server
<b>Flusso di eventi</b>	<ol style="list-style-type: none"> <li>1. Federico clicca sul pulsante “Avvia”</li> <li>2. Il server attiva i propri servizi rendendosi disponibile in tutte le sue funzionalità</li> </ol>
<b>Condizione d’uscita</b>	Il server è attivo e disponibile a ricevere connessioni

<b>Nome scenario</b>	<b>Shutdown del web server</b>
<b>Attori partecipanti</b>	Federico: Amministratore
<b>Condizione d’ingresso</b>	Federico visualizza la console per effettuare l’arresto del server
<b>Flusso di eventi</b>	<ol style="list-style-type: none"> <li>1. Federico clicca sul pulsante “Stop”</li> <li>2. Il sistema effettua le procedure per l’arresto del server</li> </ol>
<b>Condizione d’uscita</b>	Il server si è arrestato correttamente

Nel caso si verifichi un errore dovuto all’hardware o al software si cercherà di ripristinare una configurazione del sistema precedente allo stato d’errore. Visto che i dati sono gestiti dal DBMS non c’è rischio di perderli. Tuttavia, se si verifica un guasto al supporto di memorizzazione si può avere una perdita di dati.

## 4. SERVIZI DEI SOTTOSISTEMI

### 4.1. Gestione utente

Servizio	Descrizione
<b>Login</b>	Consente di effettuare l'accesso al sistema inserendo le proprie credenziali (username e password).
<b>Logout</b>	Consente di effettuare l'uscita dal sistema.
<b>Registrazione</b>	Permette ad un utente di registrarsi come per iniziare ad usufruire dei servizi offerti dal sistema.
<b>Visualizza informazioni account</b>	Permette ad un utente di visualizzare i propri dati personali, inseriti durante la fase di registrazione.
<b>Rinnovo abbonamento</b>	Consente all'utente di estendere di un mese la data di scadenza del proprio abbonamento, se scaduto.
<b>Visualizza info abbonamento</b>	Permette ad un utente di visualizzare le informazioni del proprio abbonamento, quali costo, data di inizio e data di scadenza.
<b>Visualizza info fatture</b>	Permette ad un utente di visualizzare le informazioni relative alle fatture dei propri pagamenti per la sottoscrizione al servizio.

### 4.2. Gestione catalogo

Servizio	Descrizione
<b>Visualizza contenuti</b>	Permette ad un utente di visualizzare i contenuti offerti dal catalogo, quali film e serie tv.
<b>Visualizza dettagli contenuto</b>	Permette ad un utente di visualizzare i dettagli di un contenuto del catalogo, quali regista, anno di produzione, genere, ecc.
<b>Aggiungi alla Lista Preferiti</b>	Consente all'utente di aggiungere un contenuto alla Lista dei Preferiti.
<b>Rimuovi dalla Lista Preferiti</b>	Consente all'utente di rimuovere un contenuto dalla Lista dei Preferiti.
<b>Visualizza recensioni</b>	Consente ad un utente di visualizzare le recensioni per un determinato contenuto del catalogo.
<b>Inserisci recensione</b>	Permette all'utente di inserire una nuova recensione per un determinato contenuto del catalogo, inserendo voto e testo della recensione.

### 4.3. Gestione Content creator

Servizio	Descrizione
<b>Aggiungi contenuto</b>	Consente di aggiungere nuovi contenuti al catalogo offerto.
<b>Elimina contenuto</b>	Consente di rimuovere contenuti dal catalogo offerto.