

2021/22

# SDD

System Design Document



System design document

GERARDO LEONE, ANTONIO SANTOSUOSSO, MARIO LEZZI



## Sommario

<b>Revision History</b> .....	3
<b>Team Members</b> .....	3
<b>1. Introduzione</b> .....	4
1.1 Scopo del sistema .....	4
1.2 Design Goals .....	4
1.2.1. Criteri di Performance .....	4
1.2.2. Criteri di Affidabilità .....	4
1.2.3. Criteri di Manutenzione.....	5
1.2.4. Criteri per l'Utente Finale .....	6
1.2.5. Trade-off .....	6
1.3. Definizioni, acronimi e abbreviazioni .....	6
1.4. Riferimenti .....	7
1.5. Panoramica .....	7
<b>2. Architettura del Software Corrente</b> .....	7
<b>3. Architettura del Software Proposto</b> .....	8
3.1 Panoramica sulla sezione .....	8
3.2 Decomposizione del Sistema .....	8
3.2.1 Sottosistema Signup .....	11
3.2.2 Sottosistema Authentication .....	12
3.2.3 Sottosistema Info Film .....	13
3.2.4 Sottosistema Manage Film .....	14
3.2.5 Sottosistema Manage Review .....	15
3.2.6 Sottosistema Manage Show .....	16
3.2.7 Sottosistema Manage Purchase .....	17
3.3. Mapping Hardware/Software.....	18
3.4 Gestione di dati persistenti .....	19
3.4.1 Mapping del database .....	20
3.4.2 Dettagli della struttura delle tabelle .....	21
3.5. Controllo degli Accessi e Sicurezza .....	24
3.6. Controllo del Software Globale .....	25
3.7 Condizioni Boundary.....	25
3.7.1 Inizializzazione .....	25
3.7.2 Terminazione .....	25
3.7.3 Fallimento del sistema.....	26



3.7.4 Eccezioni .....	26
4. Servizi dei sottosistemi .....	29
4.1. Signup .....	29
4.2 Authentication .....	29
4.3. Info Film .....	30
4.4. Manage Film .....	30
4.5. Manage Review .....	31
4.6. Manage Show .....	31
4.7. Manage Purchase .....	32
5. Glossario .....	33



## Revision History

Data	Versione	Descrizione	Autore
12/11/2021	0.1	Prima stesura del documento.	LG
18/11/2021	0.2	Aggiunta di Design Goal e Decomposizione del sistem.	Tutto il Team
19/11/2021	0.3	Aggiunta della Gestione dei dati persistenti.	LG
19/11/2021	0.4	Aggiunta del Controllo degli Accessi e Sicurezza.	LM, SA
23/11/2021	0.5	Aggiunta Condizioni Boundary.	LM, SA
24/11/2021	0.6	Modifica Gestione dei dati persistenti e Aggiunta dei Servizi dei sottosistemi.	Tutto il Team
9/12/2021	0.7	Revisione e correzione di eventuali errori.	Tutto il Team
11/12/2021	1.0	Correzione decomposizione Sistema e aggiunta Diagramma Architettuale	Tutto il Team
12/12/2021	1.1	Revisione	SA, LG
30/12/2021	1.2	Modifiche alla decomposizione del Sistema, servizi dei sottosistemi e al Controllo degli Accessi e Sicurezza	Tutto il Team
18/01/2022	1.3	Modifica della gestione dei dati persistenti	LG
11/02/2022	1.4	revisione	Tutto il team
14/02/2022	2.0	Revisione finale	LM, SA

## Team Members

Nome	Ruolo del progetto	Acronimo	Informazioni di contatto
Lezzi Mario	Team Member	LM	m.lezzi@studenti.unisa.it
Leone Gerardo	Team Member	LG	g.leone35@studenti.unisa.it
Santosuosso Antonio	Team Member	SA	a.santosuosso3@studenti.unisa.it

## 1. Introduzione

### 1.1 Scopo del sistema

Il sistema UniCinema è realizzato tramite interfacce semplici e intuitive, in modo da rendere le sue funzionalità accessibili e comprensibili a tutti. Per gli sviluppatori del sistema lo scopo è quello di realizzare un software che sia in grado di facilitare anche la gestione e la modifica, in modo da ottenere un sistema altamente mantenibile e affidabile.

### 1.2 Design Goals

L'obiettivo del sistema UniCinema è quello di essere il più efficiente ed intuitivo possibile attraverso l'utilizzo di agevoli interfacce. Vuole offrire un servizio che fornisce risultati in tempi brevissimi, che sia tollerabile agli errori e che sia di facile manutenzione, dal momento che la gestione di un cinema richiede di essere revisionato nella sua interezza e di inserire nuove funzionalità occasionalmente.

#### 1.2.1. Criteri di Performance

<b>Tempo di risposta</b>	UniCinema risponde alle richieste più semplici, quali inserimento, rimozione e aggiornamento di un film, nel giro di pochi millisecondi. Per quanto riguarda operazioni più pesanti, quali visualizzazione della programmazione, acquisto del biglietto e così via, il sistema cerca di rispondere alle richieste nel giro di pochi secondi (generalmente $< 8s$ ). Ovviamente molti di questi valori dipenderanno dalla connessione degli utenti e dal numero di utenti che interagiranno con il sistema.
<b>Throughput</b>	In media circa mille persone usufruiranno dei servizi del sistema, il quale riuscirà a compiere le normali operazioni all'incirca nei tempi prefissati precedentemente, dunque privo di rallentamenti e di alta latenza.
<b>Memoria</b>	Per la memorizzazione dei dati, il sistema usufruisce di un database relazionale, dunque la mole dei dati non inficerà le performance di sistema.

#### 1.2.2. Criteri di Affidabilità

<b>Robustezza</b>	Il sistema riconosce gli input non validi immessi dall'utente e permette, in maniera molto elastica e intuitiva, di correggere l'immissione.
<b>Affidabilità</b>	È in grado di mantenere i propri dati anche in caso di guasti e attacchi informatici.

<b>Disponibilità</b>	Il sistema è disponibile all'uso 24 ore su 24, 7 giorni su 7, da tutti gli utenti.
<b>Tolleranza all'errore</b>	Il sistema punta ad avere il minor numero di accoppiamento (coupling) tra sottosistemi, in modo da essere capace di operare anche in condizione di errore. Così facendo, anche se dovesse presentarsi un errore all'interno di una sotto-parte, non verrebbe intaccata la funzionalità di un'altra. Saranno effettuati periodicamente dei backup di sistema per salvare i dati anche in caso di perdita.
<b>Sicurezza</b>	<p>Il sistema utilizza diverse tecniche per resistere agli attacchi informatici:</p> <ol style="list-style-type: none"> <li>1. Crittografia della password: grazie alla crittografia della password, è possibile occultarla all'interno del database, in modo che anche se qualche malintenzionato dovesse riuscire a bucare il database, non sarebbe comunque in grado di decriptare le password.</li> <li>2. Gestione dei permessi: gli utenti possono svolgere determinate azioni solo dopo aver effettuato il login, motivo per cui vengono svolti continui controlli sugli utenti che interagiscono con i componenti del sistema. Gli amministratori avranno i permessi per svolgere qualsiasi tipo di azione all'interno del sistema, mentre il cliente registrato avrà i permessi per gestire solo il proprio account, acquistare biglietti, visualizzare gli acquisti effettuati ed effettuare le recensioni dei film. L'ospite avrà solo la possibilità di visualizzare il lato più "superficiale" del sistema, dunque la programmazione e la homepage, senza avere la possibilità di usufruire delle funzionalità principali.</li> </ol>

### 1.2.3. Criteri di Manutenzione

<b>Estensibilità</b>	Lo sviluppo del sistema è agevolato proprio dall'utilizzo di strumenti idonei all'utilizzo di librerie, estensioni e classi. Utilizzando il linguaggio di markup HTML5, CSS, Bootstrap, JQuery, Ajax, Javascript e Java, risulterà molto facile aggiungere classi al sistema.
<b>Modificabilità</b>	Utilizzando i linguaggi sopracitati e rendendo il codice di facile leggibilità, il sistema permetterà una facile risoluzione dei bugs e implementazione di nuove funzionalità.

<b>Adattabilità</b>	Proprio grazie alle funzionalità di Bootstrap e CSS, il sistema è fruibile su tutti i tipi di browser.
<b>Portabilità</b>	Le tecnologie utilizzate per lo sviluppo del sistema, consentono di essere trasportate su altri tipi di piattaforme senza troppi problemi.
<b>Leggibilità</b>	Per semplificare la manutenzione e la comprensibilità, il linguaggio sarà ben strutturato, seguendo schemi logici.
<b>Tracciabilità dei requisiti</b>	Dopo aver valutato costi e rischi che le modifiche potranno portare, sarà possibile effettuare i cambiamenti necessari grazie alla tracciabilità dei requisiti.

#### 1.2.4. Criteri per l'Utente Finale

<b>Utilità</b>	Il sistema UniCinema supporta molto bene il lavoro dell'utente, fornendo gli strumenti necessari per svolgere un lavoro di qualità.
<b>Usabilità</b>	UniCinema permette di utilizzare in modo semplice e intuitivo il sistema, garantendo un'esperienza lavorativa di qualità attraverso l'utilizzo di interfacce grafiche user-friendly.

#### 1.2.5. Trade-off

<b>Prestazioni vs costi</b>	Il team cercherà di ottenere le migliori prestazioni utilizzando il budget a disposizione.
<b>Tempi di risposta vs sicurezza</b>	Aumentare la sicurezza attraverso sistemi, comporta un aumento dei tempi di risposta delle operazioni.

### 1.3. Definizioni, acronimi e abbreviazioni

- **UniCinema:** nome del sistema.
- **Utente Registrato:** rappresenta due categorie di utenti:
- **Amministratore**
- **Cliente**
- **Ospite:** utente non registrato.
- **Sottosistema:** un sottoinsieme dei servizi del dominio applicativo, formato da servizi legati da una relazione funzionale.
- **Design Goal:** le qualità sulle quali il sistema deve essere focalizzato.
- **Dati Persistenti:** dati che sopravvivono all'esecuzione del programma che li ha creati e che dunque vengono salvati.

- **Mapping Hardware/Software:** studio della connessione tra parti fisiche e logiche di cui si compongono il sistema.
- **SDD:** System Design Document.
- **RAD:** Requirements Analysis Document.

## 1.4. Riferimenti

- Slides del corso presenti sulla piattaforma e-learning;
- Libro di testo "Object Oriented Software Engineering Using UML Patterns and Java Prentice Hall 2010 Bernd Bruegge Allen H. Dutoit".

## 1.5. Panoramica

Il documento è suddiviso in cinque parti. Nella fase introduttiva del documento vengono illustrati i design goals, ovvero i requisiti che il sistema intende soddisfare, vengono illustrate definizioni e riferimenti. Nella seconda parte del documento "Architettura Software Corrente" contiene la descrizione di un sistema software simile già esistente.

Nella terza parte del documento, ovvero "Architettura del Software Proposto" viene mostrato il system design del sistema UniCinema. In questa sezione viene decomposto il sistema, mappate le componenti del sistema hardware, vengono gestiti e mappati i dati persistenti, descritti i controlli degli accessi da parte degli utenti con una matrice degli accessi (evidenziando le possibili operazioni per ogni tipo di utente), illustrato il controllo globale del software e infine vengono descritte le condizioni limite (boundary) e la gestione dei failure. La quarta parte del documento, "Servizi dei sottosistemi", descrive quali operazioni offre ognuno dei sottosistemi. La quinta e ultima parte del documento comprende il glossario.

## 2. Architettura del Software Corrente

Esistono software simili al nostro, ad esempio The Space, CineTime o Movieplex, le cui architetture sono tutte di tipo client/server. Il nostro obiettivo è quello di ispirarci a questi software e migliorarli in termini di semplicità, interattività e renderli più user-friendly, in modo che l'utenza possa usufruirne nel modo più semplice e veloce possibile.



## 3. Architettura del Software Proposto

### 3.1 Panoramica sulla sezione

L'architettura del sistema UniCinema è di tipo client/server, sfruttando l'architettura three-tier, scelta perché si ha una separazione tra logica di business e logica di presentazione, che migliora diverse qualità, ovvero:

- Leggibilità
- Manutenzione
- Riutilizzo

Per la gestione del database utilizzeremo MySQL jdbc;

Per la generazione delle view e la parte front-end utilizzeremo HTML e CSS;

Il sistema è realizzato con un'architettura three-tier:

1. **Interface layer:** include tutti gli oggetti boundary con cui può interagire l'utente (ad esempio form, bottoni ecc..) e interfacce grafiche;
2. **Application logic layer:** formato da tutti gli oggetti control, che si occupano dunque dell'elaborazione dei dati e dell'interazione con Storage layer e notifica i cambiamenti all'Interface layer.
3. **Storage layer:** si occupa della memorizzazione dei dati e della loro manipolazione attraverso le query.

### 3.2 Decomposizione del Sistema

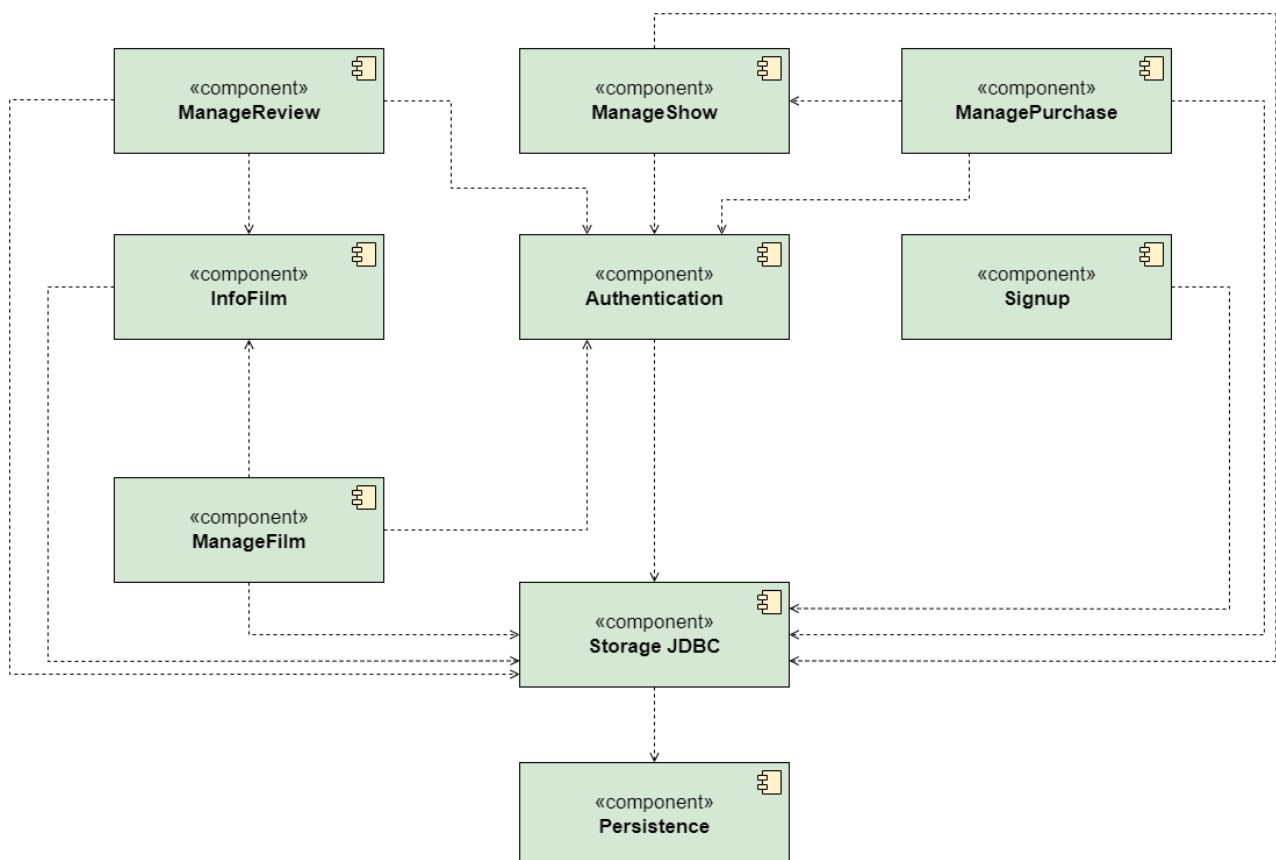
I sottosistemi individuati sono:

- **Signup:** include l'operazione di registrazione per l'Ospite.
- **Authentication:** include le operazioni di accesso per l'Utente Registrato, di Logout, di visualizzazione dati profilo e di modifica dati profilo per l'Utente Registrato;
- **Info Film:** include le operazioni che consentono la visualizzazione della programmazione, di ricercare un film all'interno di essa e visualizzare i dettagli di un dato film;
- **Manage Film:** include le operazioni che consentono all'Admin di aggiungere o rimuovere un film e di modificarne i dettagli;

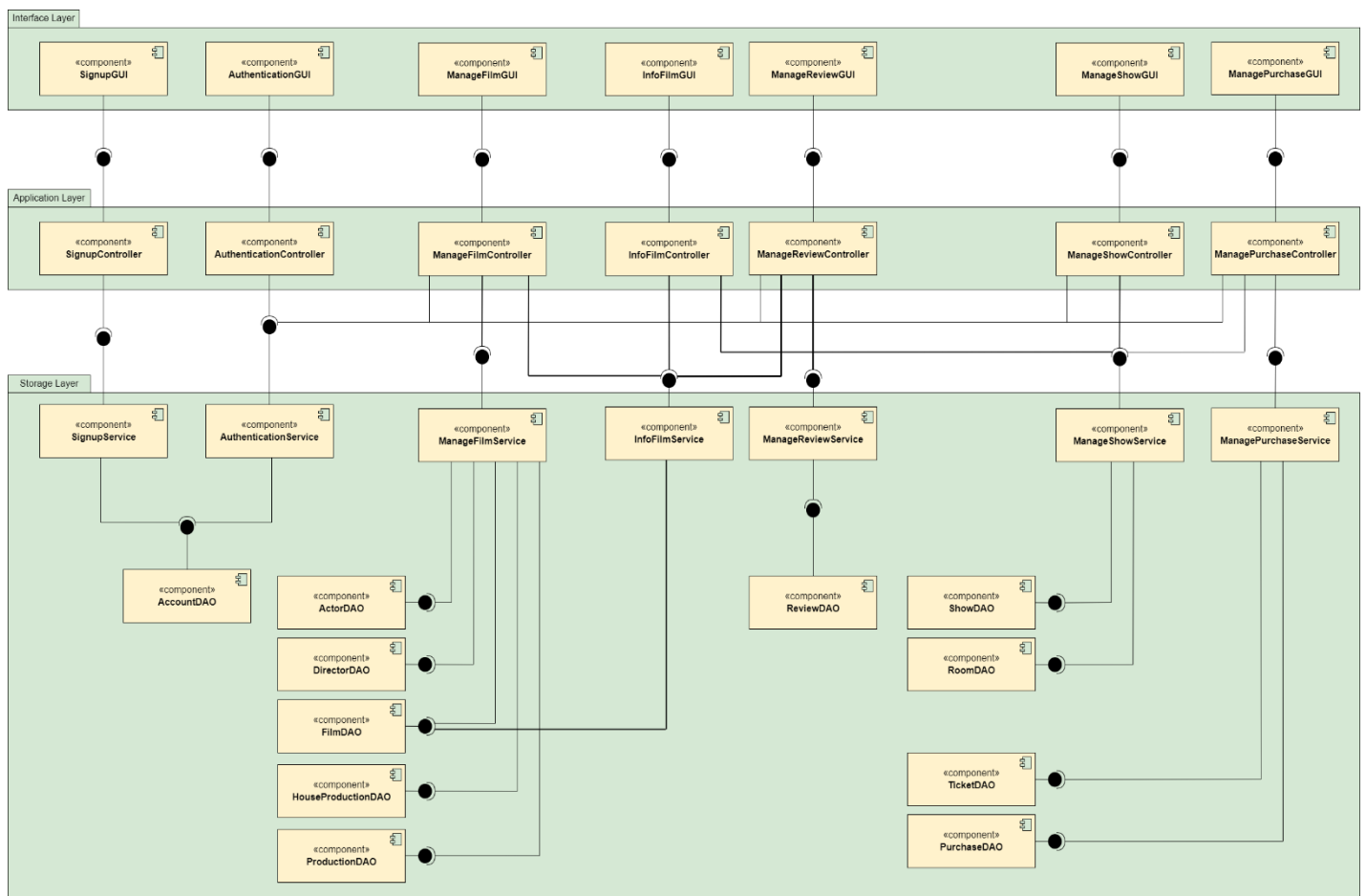
- **Manage Review:** include le operazioni che consentono all'Utente Registrato di aggiungere o rimuovere recensioni;
- **Manage Show:** operazioni che consentono all'Admin la gestione degli spettacoli, ovvero aggiunta, rimozione e modifica;
- **Manage Purchase:** operazioni riservate al cliente che consentono di acquistare un biglietto e visualizzare lo storico degli ordini.

Di seguito una vista dettagliata di ciascun sottosistema evidenziando le componenti principale:

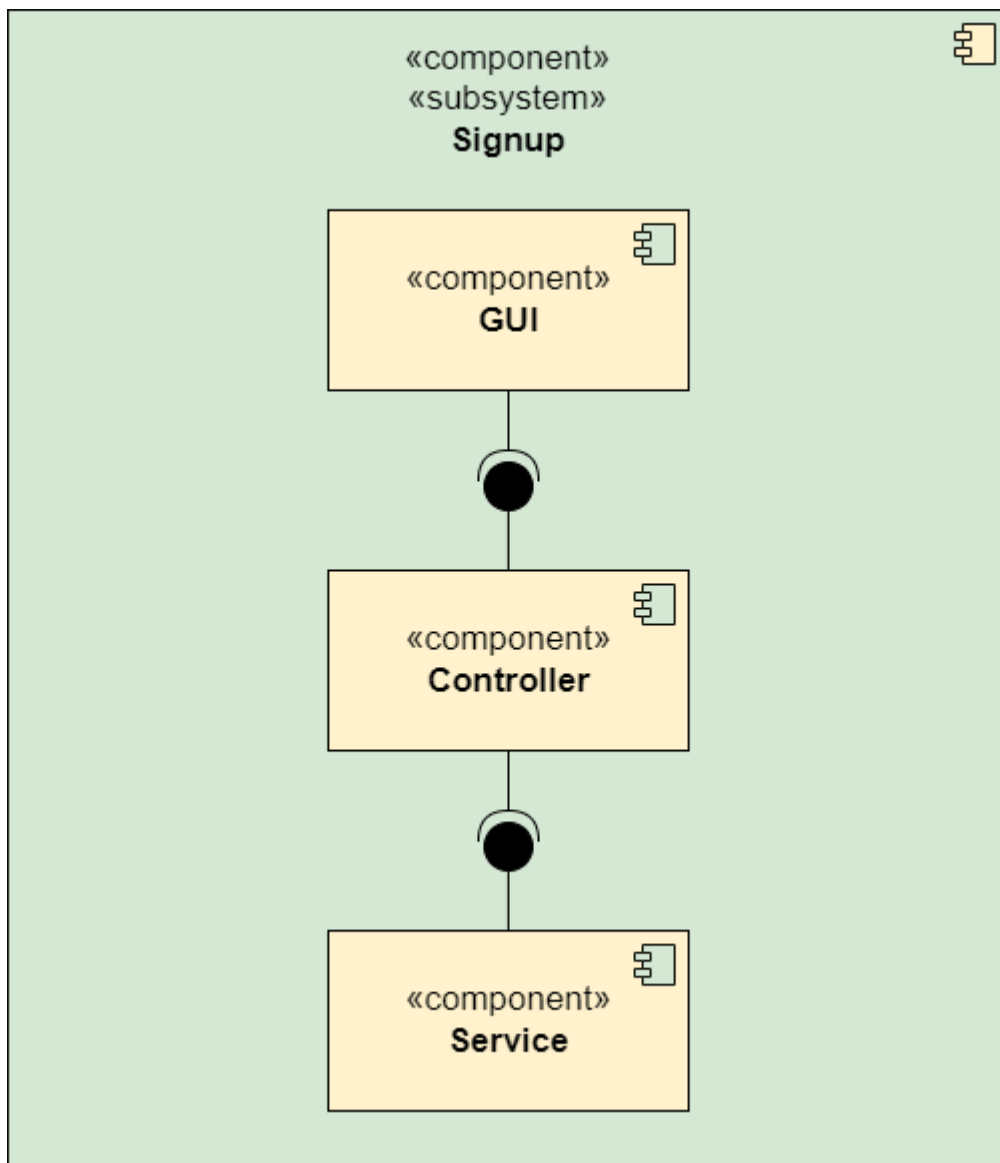
- **GUI:** Graphic User Interface, che contiene le varie view che saranno reindirizzate per creare le pagine web da mostrare al cliente.
- **Controller:** si occupa della logica per il controllo del sistema.
- **Service:** si occupa della logica di business.
- **DAO:** Data Access Object, che si occupa di fornire accesso ai dati persistenti.



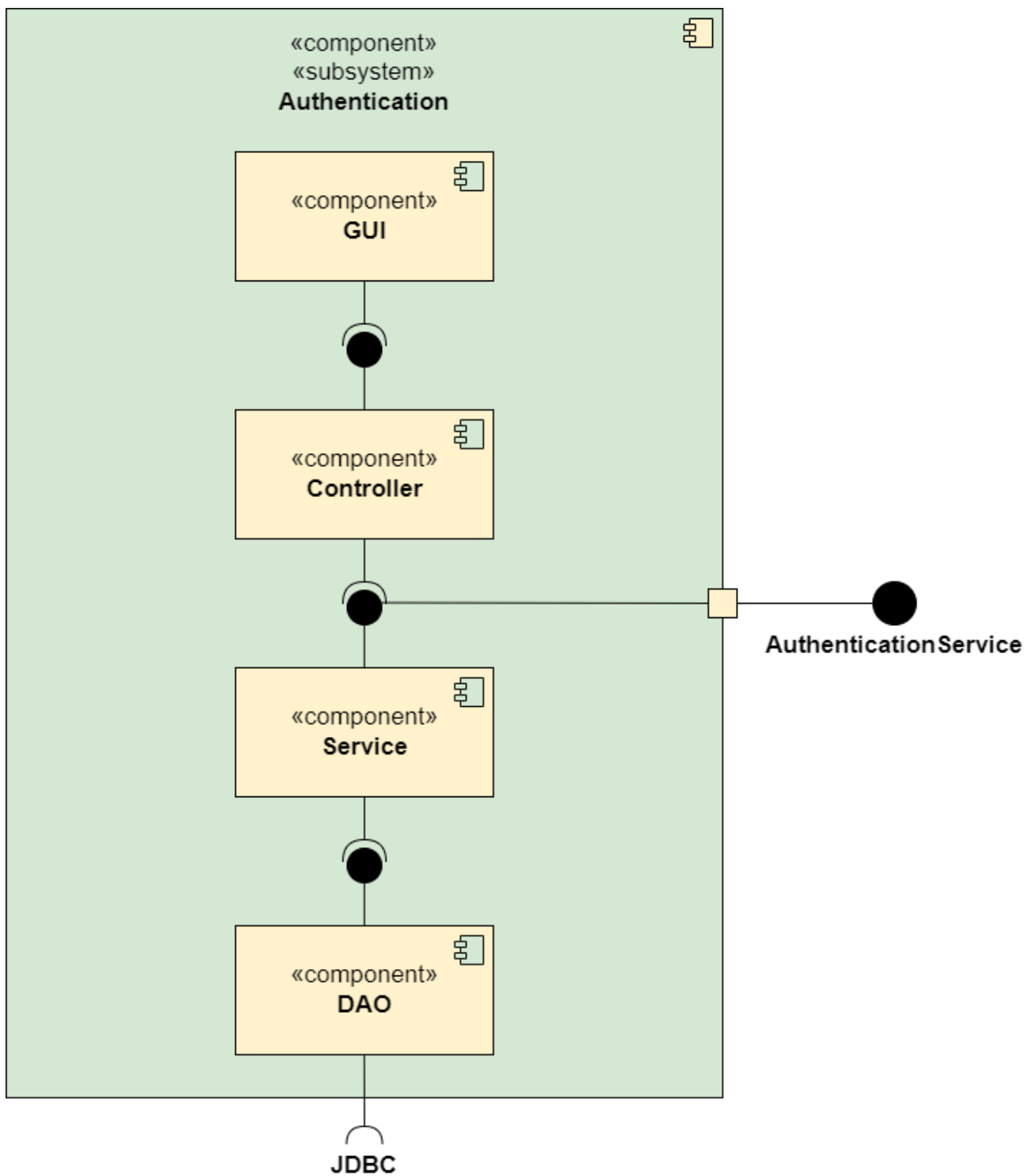
## Diagramma architetturale



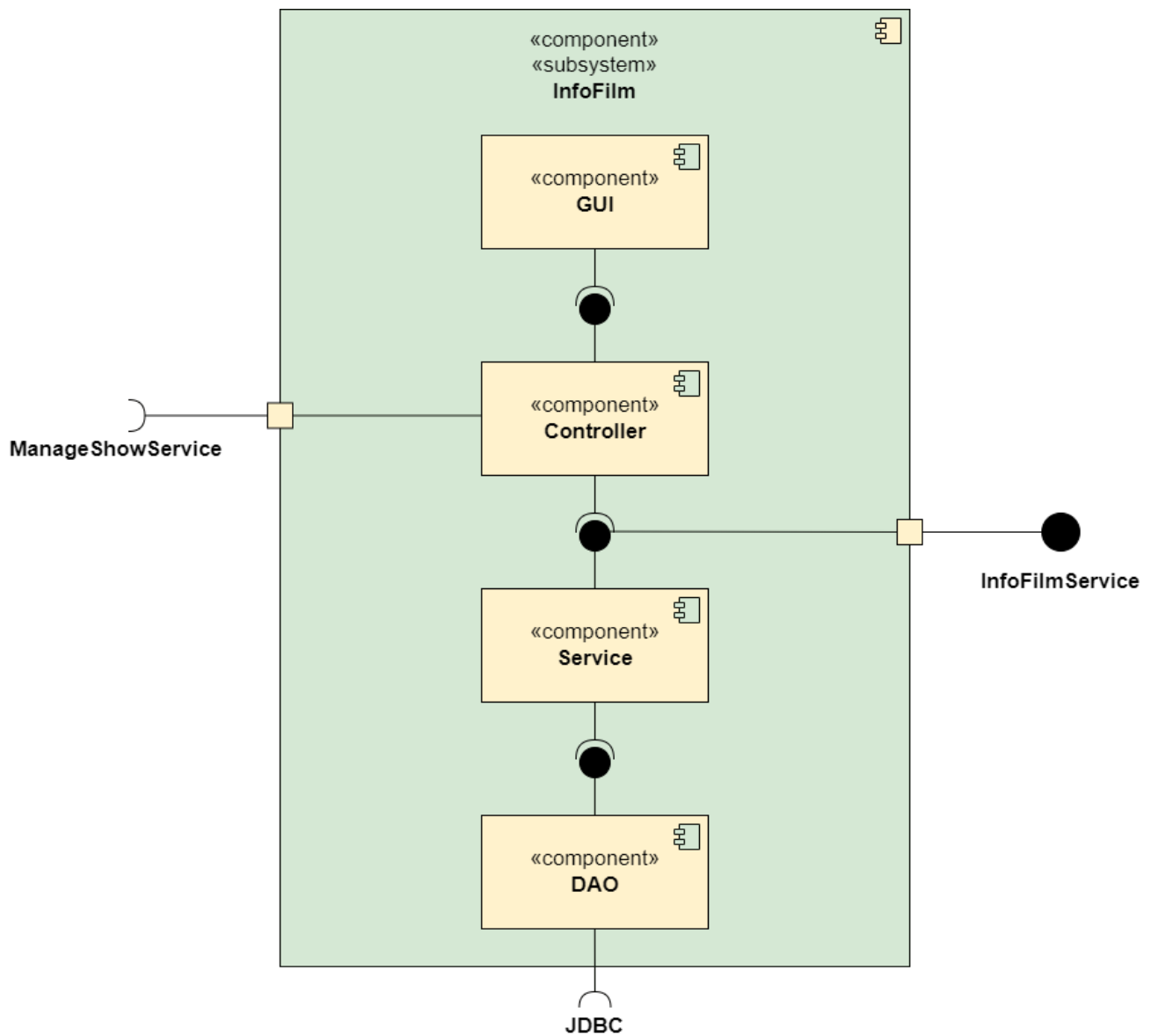
### 3.2.1 Sottosistema Signup



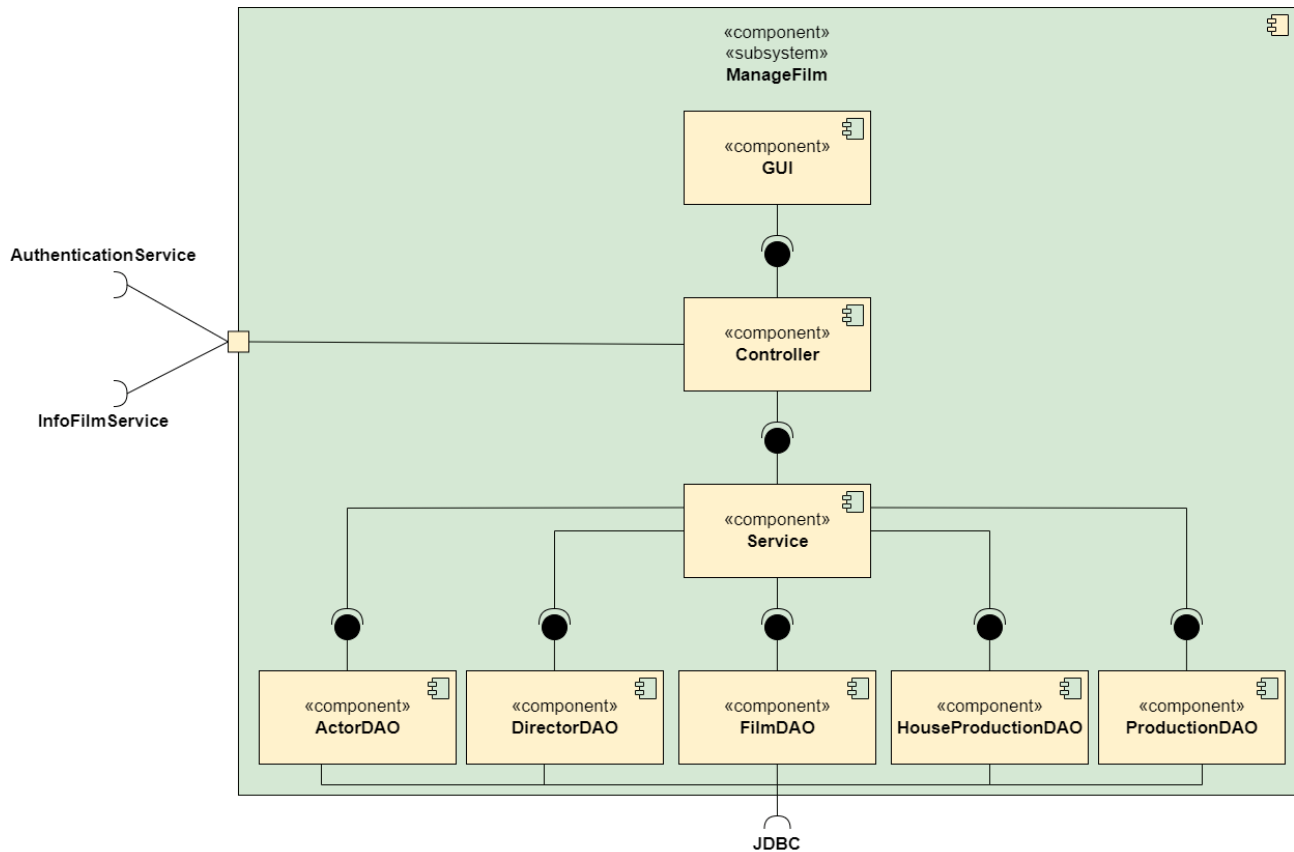
### 3.2.2 Sottosistema Authentication



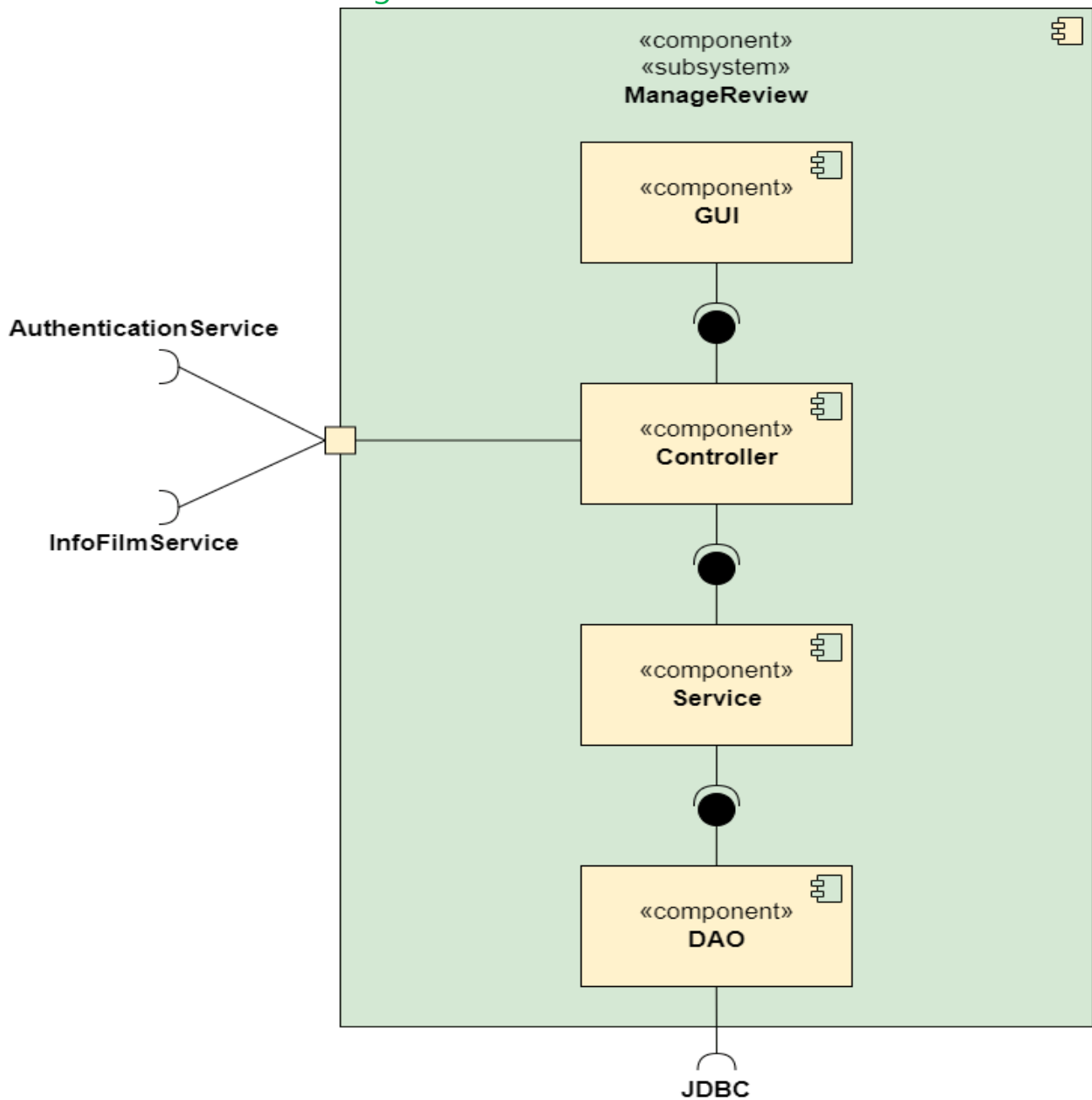
### 3.2.3 Sottosistema Info Film



## 3.2.4 Sottosistema Manage Film

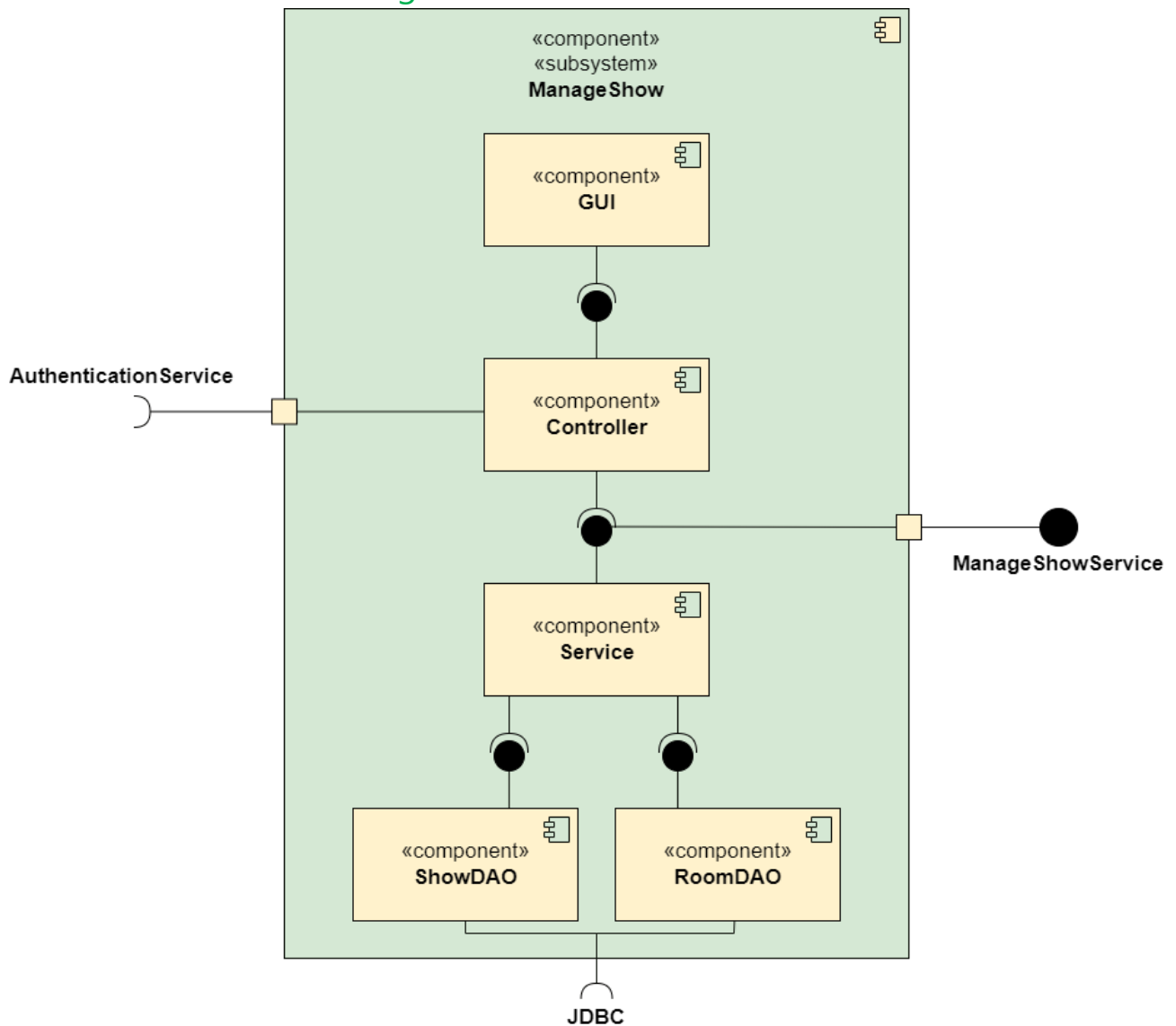


### 3.2.5 Sottosistema Manage Review

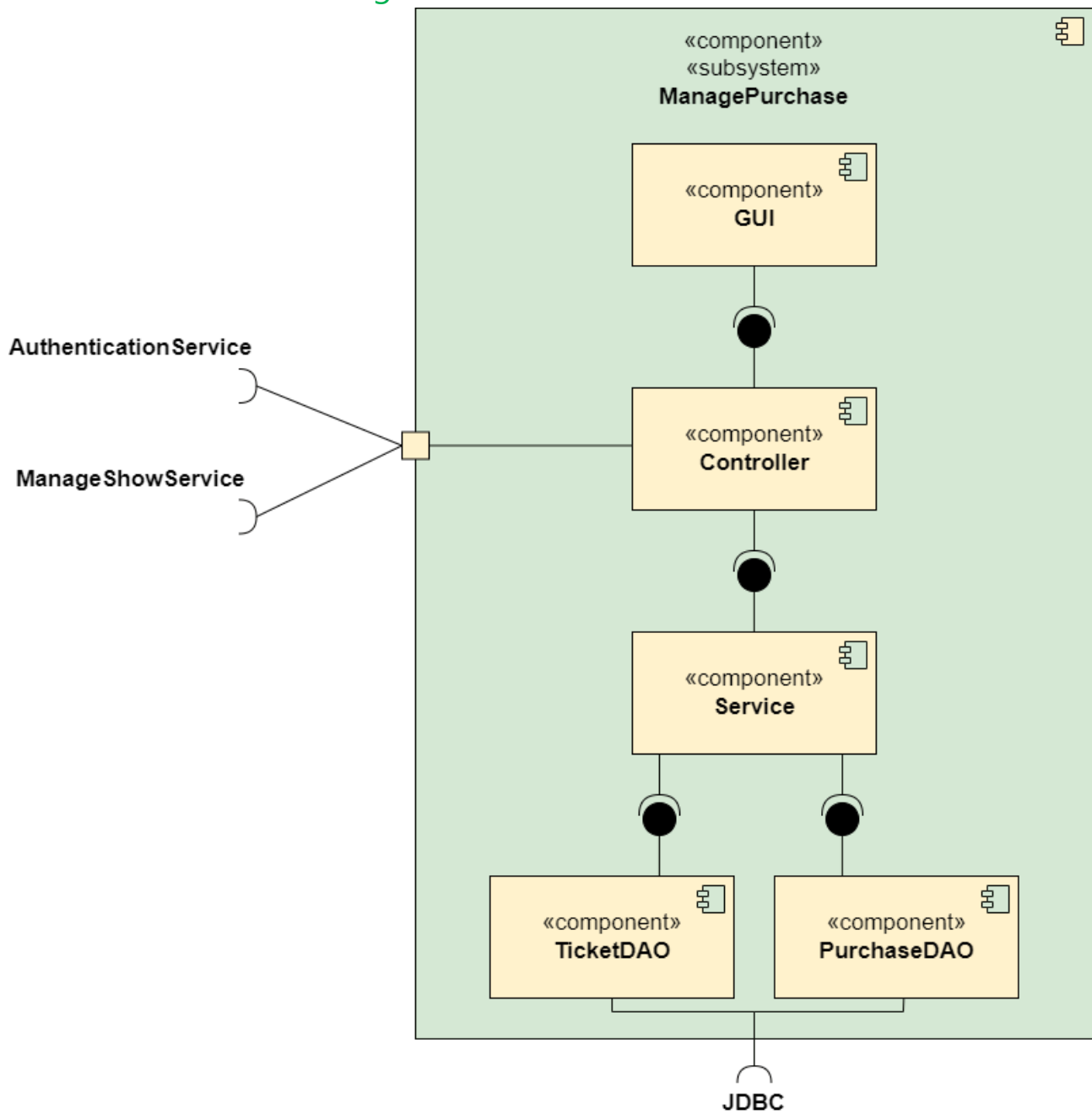




### 3.2.6 Sottosistema Manage Show

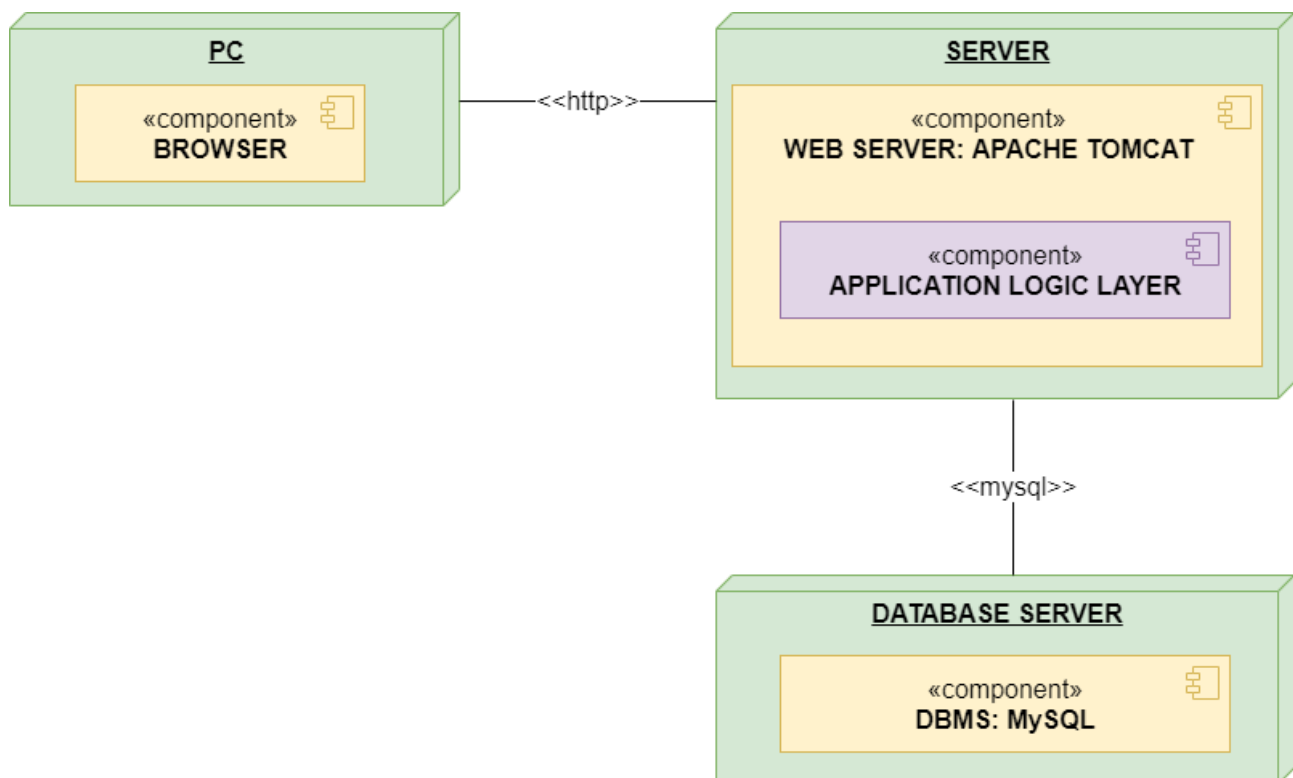


### 3.2.7 Sottosistema Manage Purchase



### 3.3. Mapping Hardware/Software

L' applicazione web che verrà sviluppata si basa su una piattaforma hardware costituita da un server che risponde alle richieste effettuate dai clienti da una qualsiasi macchina con un browser ed una connessione ad Internet. Essendo che il nostro sistema è una web application e risiede su un web server, e che si basa su un'architettura non distribuita, risiede su un solo nodo. Di seguito un UML deployment diagram che descrive il mapping hardware/software.

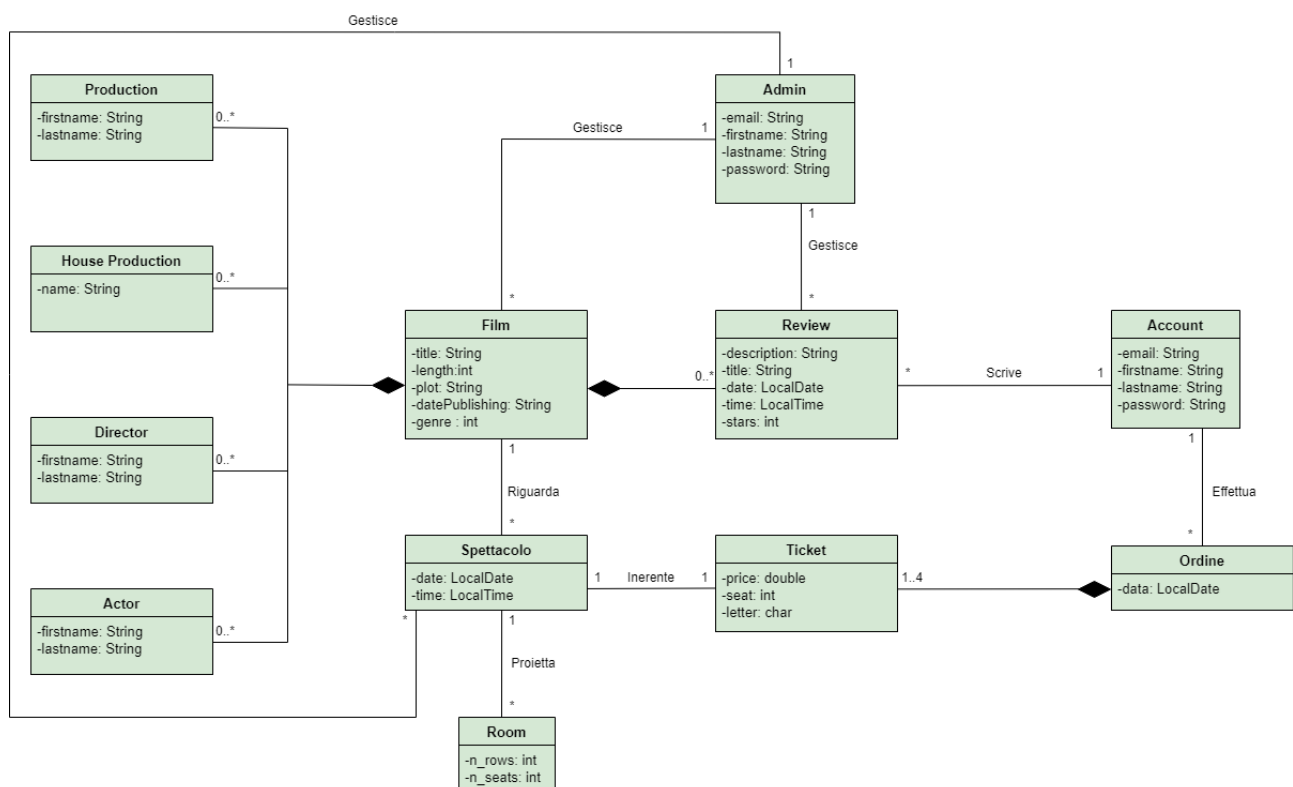


### 3.4 Gestione di dati persistenti

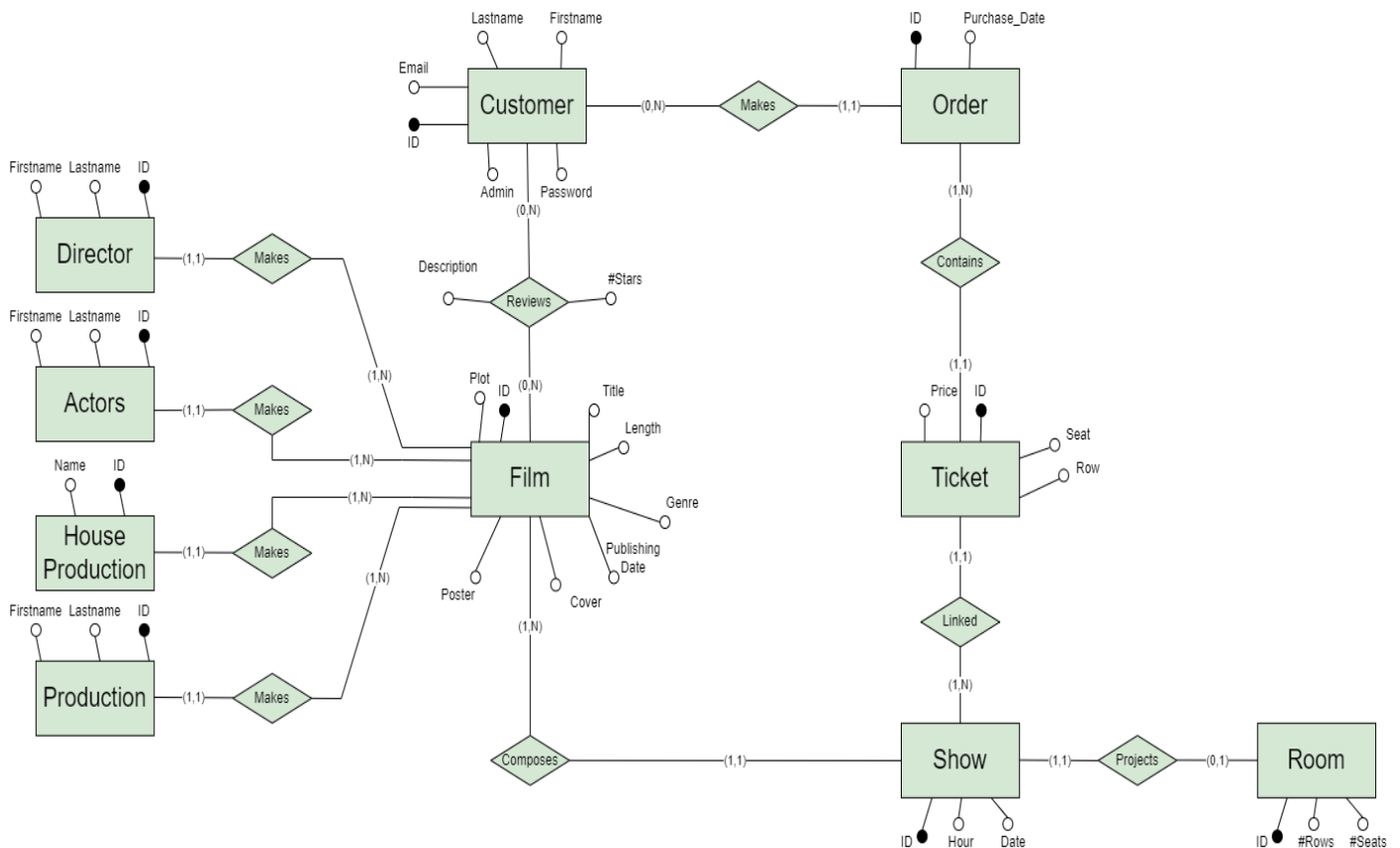
Abbiamo deciso di utilizzare un database relazionale per la gestione dei dati persistenti, in quanto permette di accedervi in modo facile e veloce. La gestione dei dati avviene tramite l'utilizzo di un DBMS, in quanto rispecchia i design goals nella loro interezza, dal momento che offre la possibilità di utilizzare vincoli di integrità sui dati, protezione dei dati da accessi multipli, gestione dei backup del database, che consentono di salvare delle copie periodicamente, in modo tale da non perdere dati oppure effettuare roll-back nel caso in cui si verificasse un'anomalia durante delle operazioni.

#### Entity Class Diagram ristrutturato

Durante l'analisi dei requisiti si era deciso di dividere in due entità l'utente registrato, precisamente in Admin e Cliente. Durante la ristrutturazione, è stato deciso di accorpare queste due entità in un'unica entità chiamata Utente Registrato e per stabilire se può effettuare o meno azioni amministrative, viene utilizzata una variabile flag "administrator". Laddove il ruolo di Utente Registrato non viene specificato, significa che sia Cliente che Admin possono effettuare quella determinata azione.



### 3.4.1 Mapping del database



Basandoci sul seguente schema ER, riportiamo il mapping del Database, in modo da offrire una visione completa della sua struttura.

**Customer** (id, email, psword, firstname, lastname, administrator)

**Order** (id, date\_purchase, id\_customer)

**Customer**(id\_customer) ha VIR con **Customer**(id)

**Review** (id\_customer, id\_film, description, #stars)

**Review**(id\_customer) ha VIR con **Customer**(id)

**Review**(id\_film) ha VIR con **Film**(id)

**Film** (id, title, length, date\_publishing, genre, plot)

**Director** (id, id\_film, firstname, lastname)

**Director**(id\_film) ha VIR con **Film**(id)

**Actors** (id, id\_film, firstname, lastname)

**Actors**(id\_film) ha VIR con **Film**(id)



**House\_Production**(id, id\_film, name)

**House\_Production**(id\_film) ha VIR con **Film**(id)

**Production** (id, id\_film, firstname, lastname)

**Production**(id\_film) ha VIR con **Film**(id)

**Show** (id, hour, date, id\_film, id\_room)

**Show**(id\_film) ha VIR con **Film**(id)

**Show**(id\_room) ha VIR con **Room**(id)

**Ticket** (id, price, seat, row, id\_show)

**Ticket**(id\_show) ha VIR con **Show**(id)

**Room** (id, #rows, #seats)

### 3.4.2 Dettagli della struttura delle tabelle

- **Customer**

Attributo	Tipo	Vincoli	Tipo di chiave
id	int	NOT NULL, AUTO_INCREMENT	PRIMARY KEY
e-mail	varchar (30)	NOT NULL	
pswr	varchar (200)	NOT NULL	
firstname	varchar(25)	NOT NULL	
lastname	varchar (25)	NOT NULL	
administrator	boolean	NOT NULL	

- **Order**

Attributo	Tipo	Vincoli	Tipo di chiave
id	int	NOT NULL AUTO_INCREMENT	PRIMARY KEY
date_purchase	date	NOT NULL	
id_customer	int	NOT NULL	FOREIGN KEY

- **Review**

Attributo	Tipo	Vincoli	Tipo di chiave
id_customer	int	NOT NULL	PRIMARY KEY, FOREIGN KEY
id_film	int	NOT NULL	FOREIGN KEY
description	varchar(500)	NOT NULL	



stars	int	NOT NULL	
-------	-----	----------	--

- **Film**

Attributo	Tipo	Vincoli	Tipo di chiave
id	int	NOT NULL AUTO_INCREMENT	PRIMARY KEY
title	Varchar (50)	NOT NULL	
length	int	NOT NULL	
date_publishing	date	NOT NULL	
genre	int	NOT NULL	
plot	Varchar (1000)	NOT NULL	
cover	Varchar (100)	NOT NULL	
poster	Varchar (100)	NOT NULL	

- **Director**

Attributo	Tipo	Vincoli	Tipo di chiave
id	int	NOT NULL AUTO_INCREMENT	PRIMARY KEY
id_film	int	NOT NULL	FOREIGN KEY
firstname	varchar(30)	NOT NULL	
lastname	varchar(30)	NOT NULL	

- **Actors**

Attributo	Tipo	Vincoli	Tipo di chiave
id	int	NOT NULL AUTO_INCREMENT	PRIMARY KEY
id_film	int	NOT NULL	FOREIGN KEY
firstname	varchar(30)	NOT NULL	
lastname	varchar(30)	NOT NULL	

- **House Production**

Attributo	Tipo	Vincoli	Tipo di chiave
id	int	NOT NULL, AUTO_INCREMENT	PRIMARY KEY
id_film	int	NOT NULL	FOREIGN KEY
name	varchar(30)	NOT NULL	



- **Production**

Attributo	Tipo	Vincoli	Tipo di chiave
id	int	NOT NULL AUTO_INCREMENT	PRIMARY KEY
id_film	int	NOT NULL	FOREIGN KEY
firstname	varchar(30)	NOT NULL	
lastname	varchar(30)	NOT NULL	

- **Show**

Attributo	Tipo	Vincoli	Tipo di chiave
id	int	NOT NULL AUTO_INCREMENT	PRIMARY KEY
show_hour	time	NOT NULL	
show_date	date	NOT NULL	
id_film	int	NOT NULL	FOREIGN KEY
id_room	int	NOT NULL	FOREIGN KEY

- **Ticket**

Attributo	Tipo	Vincoli	Tipo di chiave
id	int	NOT NULL AUTO_INCREMENT	PRIMARY KEY
price	double	NOT NULL	
rowletter	char	NOT NULL	
id_spectacle	int	NOT NULL	FOREIGN KEY
id_purchase	int	NOT NULL	FOREIGN KEY

- **Room**

Attributo	Tipo	Vincoli	Tipo di chiave
id	int	NOT NULL, AUTO_INCREMENT	PRIMARY KEY
n_rows	int	NOT NULL	
n_seats	int	NOT NULL	



### 3.5. Controllo degli Accessi e Sicurezza

Il sistema UniCinema supporta diversi tipi di utente che hanno permessi e accessi diversi tra loro. Per schematizzare il controllo degli accessi utilizziamo una matrice d'accesso per suddividere la tipologia di utente e le azioni consentite da ognuno di loro.

<b>Attori</b> <b>Oggetti</b>	<b>Admin</b>	<b>Ospite</b>	<b>Cliente</b>
<b>Registrazione</b>		<ul style="list-style-type: none"> <li>• Registrazione</li> </ul>	
<b>Autenticazione</b>	<ul style="list-style-type: none"> <li>• Login</li> <li>• Logout</li> <li>• Visualizza Profilo</li> <li>• Modifica Profilo</li> </ul>		<ul style="list-style-type: none"> <li>• Login</li> <li>• Logout</li> <li>• Visualizza Profilo</li> <li>• Modifica Profilo</li> </ul>
<b>Info Film</b>	<ul style="list-style-type: none"> <li>• Visualizza Film</li> <li>• Ricerca Film</li> <li>• Visualizza Dettagli Film</li> </ul>	<ul style="list-style-type: none"> <li>• Visualizza Film</li> <li>• Ricerca Film</li> <li>• Visualizza Dettagli Film</li> </ul>	<ul style="list-style-type: none"> <li>• Visualizza Film</li> <li>• Ricerca Film</li> <li>• Visualizza Dettagli Film</li> </ul>
<b>Gestione Film</b>	<ul style="list-style-type: none"> <li>• Modifica Dettagli Film</li> <li>• Aggiunta Film</li> <li>• Rimozione Film</li> </ul>		

<b>Gestione Recensione</b>	<ul style="list-style-type: none"> <li>• Aggiunta Recensione</li> <li>• Rimozione Recensione</li> </ul>		<ul style="list-style-type: none"> <li>• Aggiunta Recensione</li> <li>• Rimozione Recensione</li> </ul>
<b>Gestione Spettacolo</b>	<ul style="list-style-type: none"> <li>• Aggiunta Spettacolo</li> <li>• Rimozione Spettacolo</li> <li>• Modifica info Spettacolo</li> </ul>		
<b>Gestione Acquisto</b>	<ul style="list-style-type: none"> <li>• Acquista Biglietto</li> <li>• Visualizza Storico</li> </ul>		<ul style="list-style-type: none"> <li>• Acquista Biglietto</li> <li>• Visualizza Storico</li> </ul>

### 3.6. Controllo del Software Globale

UniCinema è un sistema interattivo, in quanto le funzionalità vengono avviate tramite un comando impartito dall'utente, attraverso l'utilizzo di un'interfaccia grafica. Utilizzato il comando, sarà generato un evento, gestito dalla Servlet, che si occuperà di indirizzare il flusso degli eventi al sottosistema che si occupa di svolgere la determinata azione impartita dall'utente.

### 3.7 Condizioni Boundary

#### 3.7.1 Inizializzazione

Il sistema lato server parte all'avvio del Web Server quindi il sistema è sempre attivo dato che deve essere utilizzabile 24h su 24 (ad eccezione delle ore di manutenzione). Il sistema lato client viene inizializzato nel momento in cui il cliente inserisce l'URL del sito nel browser, successivamente inserisce e-mail e password nel form dell'opportuna schermata per autenticarsi come Admin o Cliente.

#### 3.7.2 Terminazione

La terminazione del sistema lato client avviene nel momento in cui l'utente chiude l'applicazione (o il browser), come se fosse un normale log-out. Per la terminazione lato server l'admin stoppa il web server ed impedisce ad ogni

client di collegarsi al sistema (operazione che solitamente viene usata per la manutenzione del sito).

### 3.7.3 Fallimento del sistema

Il sistema lato server fallisce nel momento in cui si verificano condizioni eccezionali come la mancanza di elettricità o guasti all' hardware; in questi casi la perdita di dati avviene solo in caso di guasti al supporto di memorizzazione dato che il DBMS li gestisce.

Il sistema lato client fallisce dopo un guasto temporaneo che non influisce sul lato server.

Per recuperare da un failure lato server basta riavviare il Web Server.

### 3.7.4 Eccezioni

Quando gli utenti non riescono ad accedere al sistema, il browser del client comunica che la pagina non è disponibile con un messaggio "http 404 Not Found".

## Scenari

<b>Nome Scenario</b>	Startup Sistema
<b>Istanze di Attori Partecipanti</b>	Luca: Admin
<b>Flusso di Eventi</b>	<ol style="list-style-type: none"><li>1. Luca vuole avviare il sistema, quindi accende il Web Server.</li><li>2. Il sistema attiva i server e i servizi per poter rispondere alle richieste dei client.</li></ol>

<b>Nome Scenario</b>	Shutdown Sistema
<b>Istanze di Attori Partecipanti</b>	Luca: Admin
<b>Flusso di Eventi</b>	<ol style="list-style-type: none"> <li>1. Luca vuole arrestare il sistema, quindi spegne il Web Server.</li> <li>2. Il sistema disattiva i servizi in remoto e il server.</li> <li>3. Il sistema mostra il messaggio "404 – Not Found" ai client che desiderano connettersi.</li> </ol>

## Use cases

<b>Identificativo</b> <i>UC_Startup</i>	<i>Startup del Sistema</i>	<i>Data</i>	<i>10/12/2021</i>
		<i>Vers.</i>	<i>0.00.002</i>
		<i>Autore</i>	<i>Lezzi Mario</i>
<b>Descrizione</b>	<i>Lo UC fornisce la funzionalità di avviare il Sistema.</i>		
<b>Attore Principale</b>	<b>Admin</b> È interessato ad avviare il sistema.		
<b>Attori Secondari</b>	NA		
<b>Entry Condition</b>	L'admin visualizza il pulsante per avviare il Web Server.		
<b>Exit condition</b> On success.	Il server è attivo.		
<b>Exit condition</b>	Il server non viene avviato.		

On failure.		
<b>Rilevanza/User Priority</b>		Alta.
<b>Frequenza stimata</b>		1-2 usi/settimana
<b>Extension point</b>		NA
<b>Generalization of</b>		NA
<b>FLUSSO DI EVENTI PRINCIPALE/MAIN SCENARIO</b>		
1	Admin:	Preme sul pulsante di avvio del Web Servers.
2	Sistema:	Il sistema attiva i propri servizi rendendosi disponibile ad ogni richiesta da parte dei client.
<b>Eccezioni</b>		<b>Sistema</b> Errore Startup, il sistema non si attiva.

<b>Identificativo</b> <i>UC_Shutdown</i>	<i>Shutdown del Sistema.</i>	<i>Data</i>	<i>23/11/2021</i>
		<i>Vers.</i>	<i>0.00.001</i>
		<i>Autore</i>	<i>Lezzi Mario</i>
<b>Descrizione</b>	<i>Lo UC fornisce la funzionalità di fare lo spegnimento del sistema.</i>		
<b>Attore Principale</b>	<b>Admin</b> È interessato a effettuare lo shutdown del sistema.		
<b>Attori Secondari</b>	NA		
<b>Entry Condition</b>	L'admin visualizza il comando per stoppare il Web Server.		
<b>Exit condition</b> On success.	Il sistema termina il server.		
<b>Exit condition</b> On failure.	Il sistema è ancora attivo.		
<b>Rilevanza/User Priority</b>	Alta.		
<b>Frequenza stimata</b>	1-2 usi/settimana		
<b>Extension point</b>	NA		
<b>Generalization of</b>	NA		
<b>FLUSSO DI EVENTI PRINCIPALE/MAIN SCENARIO</b>			
1	Admin:	Preme sul pulsante per stoppare il Web Server.	
2	Sistema:	Effettua le procedure per la terminazione del server.	
<b>Eccezioni</b>	<b>Sistema</b>	Errore shutdown, il sistema è ancora attivo.	

## 4. Servizi dei sottosistemi

### 4.1. Signup

<b>Sottosistema</b>	Signup
<b>Descrizione</b>	Sottosistema che comprende la funzionalità di registrazione per gli ospiti.
<b>Servizi Offerti</b>	
<b>Servizio</b>	<b>Descrizione</b>
Registrazione	Funzione che permette agli utenti non registrati di registrarsi.

### 4.2 Authentication

<b>Sottosistema</b>	Authentication
<b>Descrizione</b>	Sottosistema che gestisce le funzionalità per l'autenticazione e la visualizzazione dei dati dell'account: login e logout per i clienti e admin, visualizzazione e modifica del profilo per gli utenti registrati.
<b>Servizi Offerti</b>	
<b>Servizio</b>	<b>Descrizione</b>
Log-in	Funzionalità che permette di accedere al sistema tramite le proprie credenziali per sfruttare tutte le funzionalità che offre.
Log-out	Funzionalità che disconnette l'utente dal sistema.
Visualizza Profilo	Funzionalità che permette agli utenti di visualizzare i dati relativi al proprio profilo.
Modifica Profilo	Funzionalità che permette di modificare le informazioni inerenti al profilo.

### 4.3. Info Film

<b>Sottosistema</b>	Info Film
<b>Descrizione</b>	Sottosistema che comprende funzionalità per l'ospite inerenti alla ricerca di un film, alla visualizzazione della programmazione e alla visualizzazione dei dettagli di un film.
<b>Servizi Offerti</b>	
<b>Servizio</b>	<b>Descrizione</b>
Ricerca Film	Questa funzionalità permette a tutti gli utenti di ricercare un film.
Visualizza Film	Questa funzionalità permette di visualizzare l'elenco dei film.
Visualizza dettagli Film	Questa funzionalità permette di visualizzare tutti i dettagli inerenti ai film.

### 4.4. Manage Film

<b>Sottosistema</b>	Manage Film
<b>Descrizione</b>	Sottosistema che comprende funzionalità per la gestione dei film da parte dell'admin.
<b>Servizi Offerti</b>	
<b>Servizio</b>	<b>Descrizione</b>
Aggiunta Film	Questa funzionalità permette all'admin di aggiungere un film.
Rimozione Film	Questa funzionalità permette all'admin di rimuovere un film.
Modifica Dettagli Film	Questa funzionalità permette all'admin di modificare i dettagli di un film.

## 4.5. Manage Review

<b>Sottosistema</b>	Manage Review
<b>Descrizione</b>	Sottosistema che comprende funzionalità per la gestione delle recensioni.
<b>Servizi Offerti</b>	
<b>Servizio</b>	<b>Descrizione</b>
Aggiunta recensione	Questa funzionalità permette di aggiungere una recensione ai film.
Rimozione recensione	Questa funzionalità permette di rimuovere una recensione ai film

## 4.6. Manage Show

<b>Sottosistema</b>	Manage Show
<b>Descrizione</b>	Sottosistema che gestisce le funzionalità per la gestione degli spettacoli.
<b>Servizi Offerti</b>	
<b>Servizio</b>	<b>Descrizione</b>
Aggiunta Spettacolo	Funzionalità che permette all'admin di aggiungere uno spettacolo.
Rimozione Spettacolo	Funzionalità che permette all'admin di rimuovere uno spettacolo.
Modifica info Spettacolo	Funzionalità che permette all'admin di modificare le informazioni relative allo spettacolo.



## 4.7. Manage Purchase

<b>Sottosistema</b>	Manage Purchase
<b>Descrizione</b>	Sottosistema che gestisce le funzionalità per la gestione degli acquisti.
<b>Servizi Offerti</b>	
<b>Servizio</b>	<b>Descrizione</b>
Acquisto Biglietto	Funzionalità che permette all'utente di acquistare da uno a quattro biglietti per spettacolo, scegliendo posti e sala che desidera.
Visualizza Storico	Funzionalità che permette all'utente di visualizzare tutti gli acquisti effettuati sul sistema.



## 5. Glossario

- **MySQL:** è un relational database management system composto da un client a riga di comando e un server.
- **VIR:** Vincolo di Integrità Referenziale
- **GUI:** Graphic User Interface
- **DAO:** Data Access Object