

**EvoCinema**

***“*** *Sistema per la gestione di un cinema “*

Versione 1.0

**System Design Document**

**Coordinatori del progetto**

|  |
| --- |
| Prof. Andrea De Lucia - Top Manager |
| Francesco Vicidomini - Project Manager |
| Ferdinando D’Avino - Project Manager |

**Partecipanti**

|  |  |  |
| --- | --- | --- |
| Luca Strefezza | 0512102474 | strluca94@gmail.com |
| Angelo Stefano D’Auria | 0512102630 | angelodauria91@gmail.com |
| Gianluca Villani | 0512102990 | lucassalerno1995@gmail.com |
| Giuseppe D’Ambrosio | 0512103472 | giuseppe.dambrosio14@gmail.com |
| Giuseppe Apuzzo | 0512103920 | g.apuzzo94@gmail.com |
| Sara De Filippo | 0512103430 | s.defilippo93@gmail.com |
| Antonio Giulio | 0512103098 | antonio.giulio96@gmail.com |
| Michele Delli Paoli | 0512103820 | mikeledellipaoli@gmail.com |
| Giuseppe Del Gaudio | 0512103690 | ciaogiuseppe96@gmail.com |
| Pietro Dell’Isola | 0512103866 | dellisola.pietro@gmail.com |
| Emanuele Buono | 0512102370 | squareman93@gmail.com |
| Francesco De Feo | 0512103274 | francescodefeo94@gmail.com |

**Revision history**

|  |  |  |  |
| --- | --- | --- | --- |
| ***Data*** | ***Versione*** | ***Descrizione*** | ***Autori*** |
| 19/12/2017 | 1.0 | Stesura SDD | Team Members |

**INDICE**

[**Introduzione**](#_bh8htuiwzrse) **5**

[1.1 Descrizione del problema](#_5oidn3q9taqk) 5

[1.2 Design Goals](#_jby4jlao3cxf) 5

[1.2.1 DG\_0 Criteri di Performance](#_v3yzme8jaybc) 6

[1.2.2 DG\_1 Criteri di Affidabilità](#_4ytw2nhiz257) 6

[1.2.3 DG\_2 Criteri di Costo](#_4sqqgnwzxi9u) 6

[1.2.4 DG\_3 Criteri di Mantenimento](#_v1v7y5l9nido) 7

[1.2.5 DG\_4 Criteri di Utente](#_sifh0absh62l) 7

[1.3. Definizioni, Acronimi e abbreviazioni](#_w5m42v4a1wpd) 7

[1.4. Riferimenti](#_rwh7jreeyrh6) 8

[RAD [Ufficiale]EvoCinema v1.0](#_idx6nplo1wbt) 8

[1.5. Panoramica](#_j62656izu7oy) 8

[**Architettura del sistema**](#_v2s44q1fn9hg) **8**

[2.1 Panoramica](#_sazy83bgbo7w) 8

[2.2 Decomposizione in sottosistemi](#_hrl9oyw130nl) 9

[2.2.1 Macrodecomposizione in sistemi](#_dhog6xvrmol3) 9

[2.2.2 Microdecomposizione in sottosistemi](#_il8htg5j8nl4) 10

[2.2.3 Gestione Account](#_hu0hdqm85o4f) 11

[GestioneAccountPresentationLayer](#_qvnvdgfe2ct3) 12

[GestioneAccountApplicationLayer](#_mhr1ecqo5f3d) 13

[GestioneAccountDataLayer](#_qyra3hmmk7f) 13

[2.2.4 Gestione Prenotazioni](#_jmwtv0o51331) 13

[GestionePrenotazioniPresentationLayer](#_n2xeb65yayq1) 14

[GestionePrenotazioniApplicationLayer](#_6rz38t62ephg) 14

[GestionePrenotazioniDataLayer](#_ub9u49lomhzw) 14

[2.2.5 Gestione Acquisti](#_20w0lz454b7y) 15

[GestioneAcquistiPresentationLayer](#_a9vtld47kmsi) 15

[GestioneAcquistiApplicationLayer](#_hbzmca2de2el) 15

[GestioneAcquistiDataLayer](#_88e181rcu6pa) 16

[2.2.6 Gestione Programmazione](#_ue48518qics7) 16

[GestioneProgrammazionePresentationLayer](#_c859za1j148p) 17

[GestioneProgrammazioneApplicationLayer](#_1i9ctoo2xaha) 17

[GestioneProgrammazioneDataLayer](#_v7k8z3op01n2) 17

[2.2.7 Gestione Libreria](#_2gaw2lwlrug9) 18

[GestioneLibreriaPresentationLayer](#_njg0qdkbhmbi) 19

[GestioneLibreriaApplicationLayer](#_ulq8y97wc6td) 20

[GestioneLibreriaDataLayer](#_mpl9w31afxx5) 20

[2.2.8 Gestione Sconti](#_6ty6ktnx4rkp) 21

[GestioneScontiPresentationLayer](#_z7qlhsi7o3ww) 22

[GestioneScontiApplicationLayer](#_d5f0cva0vvay) 23

[2.2.9 Gestione Sala](#_75a9zw6chp7) 23

[GestioneSalaPresentationLayer](#_5veiogoiod3l) 24

[GestioneSalaApplicationLayer](#_r7sj8a5trs2n) 25

[GestioneSalaDataLayer](#_z4qsc85enhrn) 25

[2.2.10 Gestione Analytics](#_pc0tu9wd336k) 25

[GestioneAnalyticsPresentationLayer](#_q6focomzoogy) 26

[GestioneAnalyticsApplicationLayer](#_bdvlnzpe83p1) 26

[GestioneAnalyticsDataLayer](#_9ofwi2d9o90j) 27

[2.3 Mappatura Hardware/Software](#_eb3b0ms54ym9) 27

[2.4 Gestione dei dati persistenti](#_i8h90fnw4hcq) 28

[2.4.1 Modello E-R](#_7uiqhsgu5r96) 28

[2.4.2 Modello E-R Ristrutturato](#_cvmmnbjiw0ff) 29

[2.4.3 Modello logico](#_gy1lob3rvcei) 30

[2.4.4 Struttura delle tabelle](#_v9h3hlpmp412) 30

[2.5 Gestione degli accessi](#_6ppfw5n2attl) 34

[**3. Servizi dei sottosistemi**](#_vhqs64yv9gv7) **35**

[3.1. SS\_ACC - Gestione account](#_ued7dioqp4s2) 35

[3.2. SS\_SAL - Gestione Sala](#_2jt28dah0e0m) 36

[3.2 SS\_LIB - Gestione Libreria](#_wkhgwtiqmfxz) 36

[3.3 SS\_PRG - Gestione programmazione](#_vezkill3toef) 37

[3.4. SS\_SCN - Gestione sconti](#_4pyrdhhwxueh) 37

[3.5. SS\_PRN - Gestione prenotazioni](#_g1pfvocxl90c) 38

[3.6. SS\_ACQ - Gestione acquisti](#_u40bznsydlfk) 38

[3.7. SS\_ANL - Gestione analytics](#_rhzsdb89g6q1) 39

[**4. Glossario**](#_48tdoj8n4xug) **39**

# **Introduzione**

## **1.1 Descrizione del problema**

Grazie al progetto che stiamo sviluppando, gestire un cinema non sarà più un operazione noiosa e faticosa.

Gli obiettivi principali del sistema sono quelli di ottimizzare la gestione per il personale e semplificare le funzionalità del cliente.

Per rendere questo possibile verrà creata una web application con un’interfaccia chiara ed intuitiva.

Le funzionalità che il sistema offre al Gestore sono:

* Gestire i film in programmazione
* Gestire le politiche di sconto
* Visualizzare gli incassi
* Strumenti per conoscere i film più apprezzati.

Le funzionalità che il sistema offre all’operatore sono:

* Gestire vendita e prenotazione biglietto
* Applicare sconti

Le funzionalità che il sistema offre all’utente sono:

* Visualizzare i film in programmazione o in libreria
* Prenotare o acquistare biglietti

## **1.2 Design Goals**

Il sistema Evo Cinema punterà ad essere il più lineare ed intuitivo possibile.

Per far ciò si avvarrà di una struttura grafica chiara e completa, con bottoni, finestre di dialogo, icone e menù a tendina. Cercherà inoltre di isolare le informazioni necessarie per indirizzare più facilmente l’utente verso la funzionalità da lui richiesta.

Valore aggiunto all'applicazione sarà la sua semplicità, che permetterà anche agli utenti con scarsa conoscenza del sistema di portare a termine le loro operazione, evitando di inserire dati sbagliati o commettere errori durante l’utilizzo del programma.

Il sistema proposto rispetterà i seguenti criteri di design:

### **1.2.1 DG\_0 Criteri di Performance**

* DG\_0.1 Tempi di risposta.Il sistema deve garantire tempi di risposta brevi per ogni funzionalità. Mediamente una richiesta dovrà essere soddisfatta in un tempo non superiore ai 5 secondi. Ovviamente quest’ultimo può oscillare in base alla velocità di connessione; tanto maggiore sarà la velocità quanto minore sarà il tempo di risposta.
* DG\_0.2 Troughput:Il sistema sarà capace di gestire una media di 200 utenti, consentendo loro di effettuare tutte le operazioni senza subire alcun rallentamento. La piattaforma sarà anche in grado di gestire un maggiore afflusso di utenti qualora sia necessario.
* DG\_0.3 Dati persistenti: Il sistema, per memorizzare tutti i dati relativi a spettacoli, cinema, sala e posti, utilizzerà un database relazionale. La scelta di quest’ultimo farà in modo che le informazioni saranno disponibili il più velocemente possibile.

### **1.2.2 DG\_1 Criteri di Affidabilità**

* DG\_1.1 Robustezza. Il sistema mostrerà un messaggio che avvertirà l’utente nel caso in cui i dati inseriti siano mancanti o errati. Questo consentirà di non immettere dati sbagliati all’interno del database.
* DG\_1.2 Affidabilità. Il sistema garantirà il corretto svolgimento di tutte le funzionalità e cercherà di produrre l’output atteso evitando errori indesiderati.
* DG\_1.3 Disponibilità. Il sistema dovrà essere sempre fruibile agli utenti. Non sono previsti periodi di chiusura del sistema se non se non quelli per motivi di manutenzione.
* DG\_1.4 Sicurezza. Il sistema, tramite username e password, riuscirà ad individuare il tipo di utente e gli permetterà di effettuare solo le operazioni appartenenti alla sua categoria.

### **1.2.3 DG\_2 Criteri di Costo**

* DG\_2.1 Costo. Prima dell’inizio dello sviluppo della piattaforma verranno valutati i costi di sviluppo del sistema, la sua installazione e il training degli utenti, eventuali costi per convertire i dati del sistema precedente, costi di manutenzione e costi di amministrazione.

### **1.2.4 DG\_3 Criteri di Mantenimento**

* DG\_3.1 Estendibilità. Il sistema verrà costruito in modo da poter aggiungere nuove funzionalità senza la necessità di modificare quelle esistenti o di modificare la struttura del progetto.
* DG\_3.2 Modificabilità. Il sistema permetterà di apportare modifiche alle funzionalità già implementate senza la necessità di modificare i sotto-sistemi esistenti.
* DG\_3.3 Tracciabilità dei requisiti. Grazie all’utilizzo di una matrice di tracciabilità si potrà verificare quali requisiti funzionali, use case e scenari sono stati realizzati nel progetto e quali ancora devono essere implementati.

### **1.2.5 DG\_4 Criteri di Utente**

* DG\_4.1 Utilità. Il sistema punterà a soddisfare tutte le esigenze degli utenti sviluppando tutti i requisiti funzionali raccolti in fase di analisi.
* DG\_4.2 Usabilità. Il sistema dovrà essere chiaro ed intuitivo e anche l’utente con meno esperienza dovrà essere in grado di effettuare le operazioni desiderate. La piattaforma consentirà “l’autoapprendimento” cioè l’utente effettuerà le operazioni in maniera sempre più rapida con il maggiore utilizzo del sistema.

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

|  |  |
| --- | --- |
| Acronimo | Descrizione |
| RAD | RequirementAnalysisDocument |
| JDBC | Java DataBase Connectivity |
| DBMS | Database Management System |
| HTTP | HyperText Transfer Protocol |
| GUI | Graphic User Interface |
| DB | Database |

## 

## **1.4. Riferimenti**

### RAD [Ufficiale]EvoCinema v1.0

* Object-Oriented Software Engineering Using UML, Patterns, and Java™ Third Edition
* https://it.wikipedia.org/

## **1.5. Panoramica**

Il documento si compone di 4 parti fondamentali:

1. Introduzione: In questa sezione verranno illustrati gli obiettivi del sistema proposto e saranno descritte, in modo non approfondito, le varie funzionalità messe a disposizione per gli attori. Verranno poi mostrati gli obiettivi del design, in particolar modo i criteri che il sistema dovrà rispettare.
2. Architettura del sistema corrente: In questa sezione verrà spiegata l’architettura del sistema corrente, ossia l’implementazione del progetto, gli attori del sistema e le varie funzionalità messe a disposizione.
3. Architettura del sistema proposto: In questa sezione verrà elaborata la descrizione generale del sistema proposto, la microcomposizione e la macrocomposizione in sottosistemi, la gestione dei dati persistenti, il mapping hardware e software del sistema, il controllo degli accessi sicurezza e le condizioni limite.
4. Servizi dei sottosistemi e Glossario: In questa sezione troveremo la descrizione per ogni servizio fornito dal sottosistema e un glossario contenente una raccolta dei termini contenuti nel sistema proposto.

# **Architettura del sistema**

## **2.1 Panoramica**

Attualmente la gestione del cinema avviene tramite un applicativo scritto in Java e non prevede un funzionamento online, fondamentale per un progetto di questo tipo.

Il sistema attuale non prevede l’uso dell’applicativo in modalità Operatore ma presenta una modalità per il Gestore ed una per l’Utente.

In particolare il Gestore può: inserire, modificare ed eliminare spettacoli, gestire politiche di sconto per i biglietti, gestire i posti all’interno delle sale e visualizzare i film in programmazione per numero di posti o per incasso.

L’Utente dopo aver effettuato la registrazione ed il login può solo visualizzare la programmazione odierna e settimanale, prenotare un posto per uno spettacolo, disdire la prenotazione ed acquistare un biglietto. Manca, nel sistema attuale, la gestione del credito per l’acquisto dei biglietti, non essendo presente la versione online.

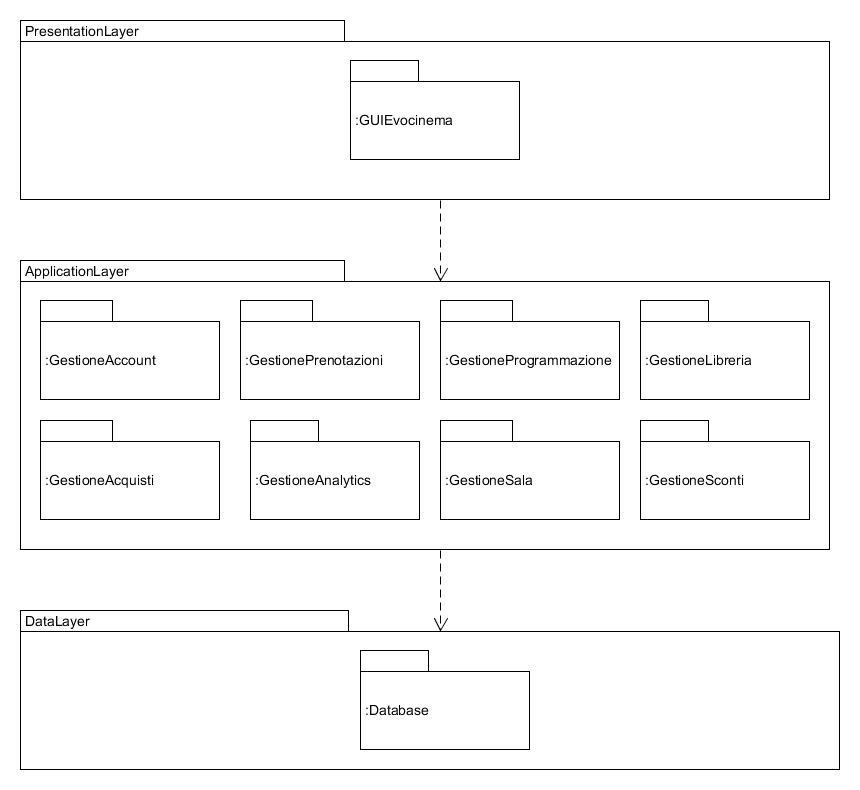
## **2.2 Decomposizione in sottosistemi**

### **2.2.1 Macrodecomposizione in sistemi**

La macro decomposizione divide il sistema nei seguenti sottosistemi:

1. Gestione Account.
2. Gestione Prenotazione.
3. Gestione Acquisti.
4. Gestione Programmazione.
5. Gestione Libreria.
6. Gestione Sconti.
7. Gestione Sala.
8. Gestione Analytics.

Di seguito è mostrato il diagramma in cui vengono evidenziate le relazioni di dipendenza tra i package di ogni sottosistema e tra i diversi sottosistemi.

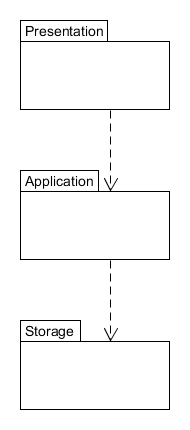


Gli utenti che useranno il sistema lo faranno dal proprio computer comunicando gli input all’interfaccia del Server Web i quali verranno gestiti dal Database in cui sono contenute tutte le informazioni dell’intero sistema. Il Database sarà gestito da un DBMS che si occupa di inserire, cercare e aggiornare i dati presenti al suo interno, elaborando la richiesta degli utenti da parte del Server. Il DBMS si occuperà anche di gestire gli accessi concorrenti al Database.

### **2.2.2 Microdecomposizione in sottosistemi**

Per semplificare la progettazione e lo sviluppo dell’applicazione, i sottosistemi saranno decomposti secondo lo schema previsto dall’architettura software Three-Layer, ovvero:

1. Presentation: si occupa di gestire l’interfaccia grafica del sottosistema e contiene tutti gli elementi che interagiscono con l’utente.
2. Application: si occupa, dati gli input dagli utenti, di elaborarli e fornire dei risultati che verranno poi mostrati graficamente.
3. Storage: è la parte del sottosistema che interagisce col Database.

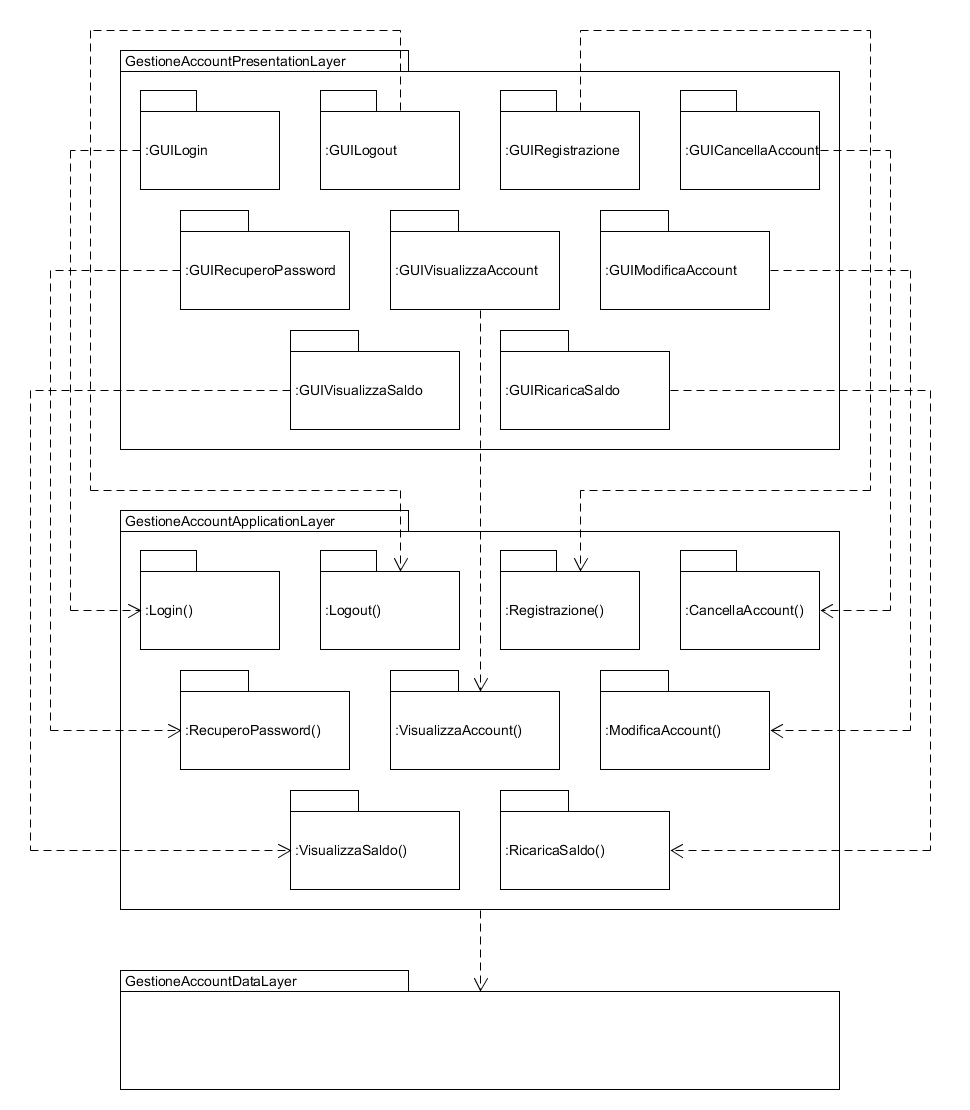


Ecco riportato uno schema generale:

### **2.2.3 Gestione Account**

Questo sottosistema si occupa della gestione dell’account, in particolare fornisce le funzioni di:

* Login
* Logout
* Registrazione
* Cancellazione account
* Recupero password
* Visualizza account
* Modifica account
* Ricarica saldo
* Visualizza saldo



#### **GestioneAccountPresentationLayer**

Include tutti gli elementi dell’interfaccia grafica che offrono funzionalità riguardanti la gestione dell’account. Comprende:

* GUILogin: comprende le interfacce che consentono all’Utente Registrato di effettuare l’accesso al sistema.
* GUILogout: comprende le interfacce che consentono all’Utente Registrato di scollegarsi dal sistema.
* GUIRegistrazione: comprende le interfacce che consentono ad un utente non registrato di registrarsi all’interno del sistema.
* GUICancellaAccount: comprende le interfacce che consentono all’Utente Registrato di cancellarsi dal sistema.
* GUIRecuperoPassword: comprende le interfacce che consentono all’Utente Registrato di recuperare la propria password di accesso.
* GUIVisualizzaAccount: comprende le interfacce che consentono all’Utente Registrato di visualizzare le informazioni relative al proprio account.
* GUIModificaAccount: comprende le interfacce che consentono all’Utente Registrato di modificare le informazioni relative al proprio account.
* GUIVisualizzaSaldo: comprende le interfacce che consentono all’Utente Base di visualizzare le informazioni relative al saldo del proprio account.
* GUIRicaricaSaldo: comprende le interfacce che consentono all’Utente Base e all’Operatore di effettuare una ricarica al saldo del proprio account o di quello di un Utente che lo richiede.

#### **GestioneAccountApplicationLayer**

* Login(): operazioni per l’accesso al sistema.
* Logout(): operazioni per uscire dal sistema.
* Registrazione(): operazioni per registrarsi al sistema.
* CancellaAccount(): operazioni per cancellare il proprio account dal sistema.
* RecuperoPassword(): operazioni per il recupero della password.
* VisualizzaAccount(): operazioni per la visualizzazione delle informazioni relative al proprio account.
* ModificaAccount(): operazioni per la modifica delle informazioni relative al proprio account.
* VisualizzaSaldo(): operazioni per la visualizzazione delle informazioni relative al saldo del proprio account.
* RicaricaSaldo(): operazioni per la ricarica del saldo di un account.

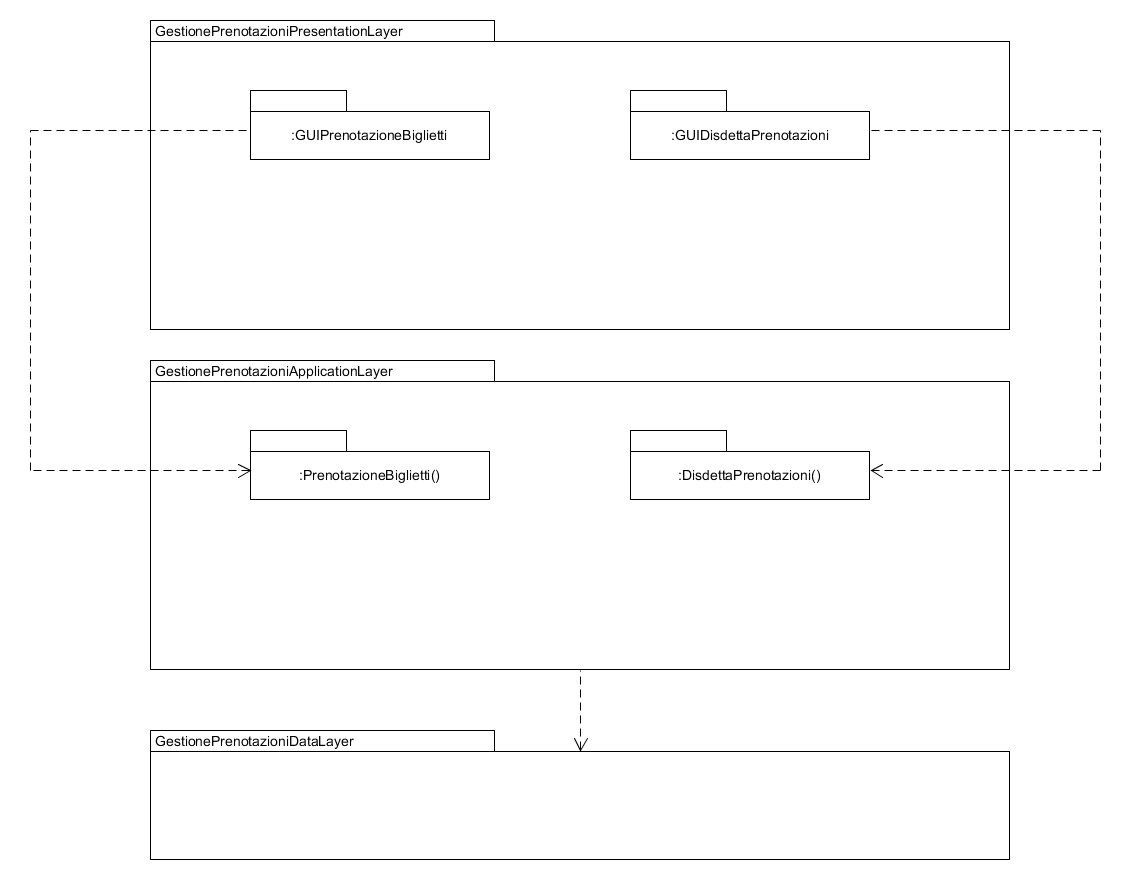
#### **GestioneAccountDataLayer**

Si occupa di rendere reperibili i dati, presenti all’interno del database, relativi agli account.

## **2.2.4 Gestione Prenotazioni**

Questo sottosistema si occupa della gestione delle prenotazioni, in particolare fornisce le funzioni di:

* Prenotazione Biglietti
* Disdetta Prenotazioni



#### **GestionePrenotazioniPresentationLayer**

Include tutti gli elementi dell’interfaccia grafica che offrono funzionalità riguardanti la gestione delle prenotazioni. Comprende:

* GUIPrenotazioneBiglietti: comprende le interfacce che consentono all’Utente Base e all’Operatore di effettuare prenotazioni di biglietti.
* GUIDisdettaPrenotazioni: comprende le interfacce che consentono all’Utente Base e all’Operatore di disdire prenotazioni di biglietti.

#### **GestionePrenotazioniApplicationLayer**

* PrenotazioneBiglietti(): operazioni per la prenotazione di biglietti.
* DisdettaBiglietti(): operazioni per la disdetta di prenotazioni

#### **GestionePrenotazioniDataLayer**

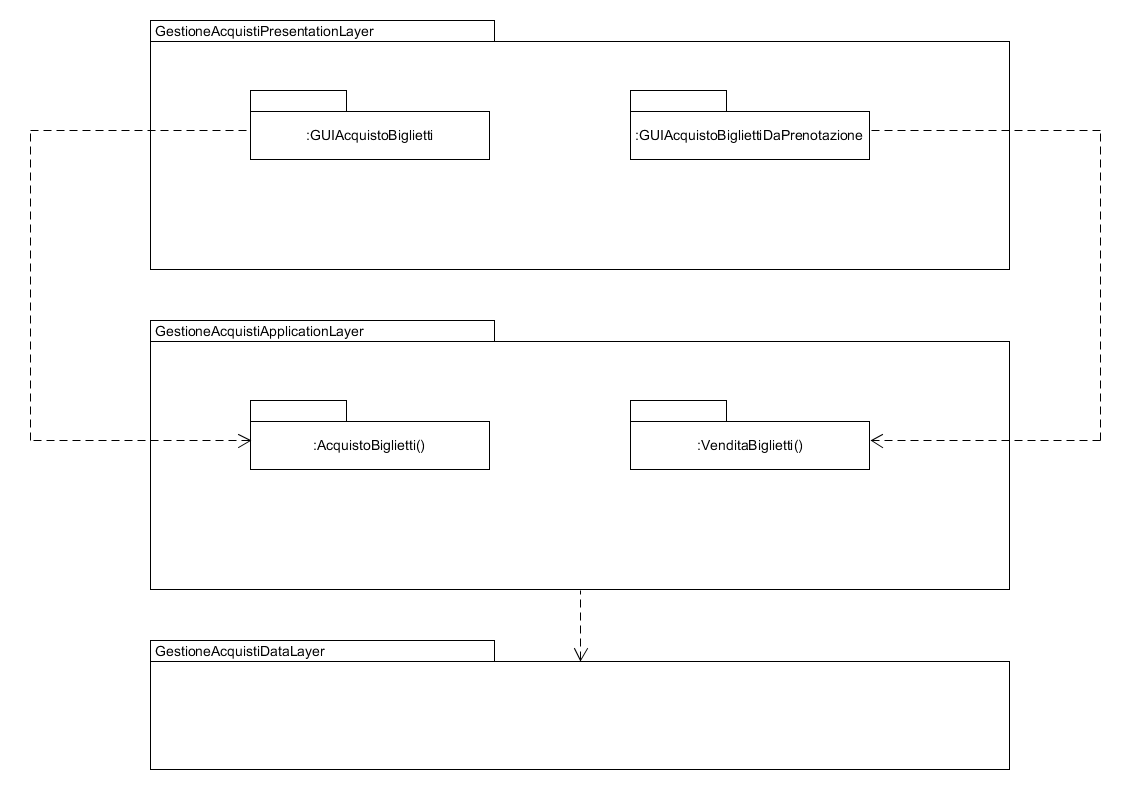
Si occupa di rendere reperibili i dati, presenti all’interno del database, relativi alle prenotazioni.

## 

## **2.2.5 Gestione Acquisti**

Questo sottosistema si occupa della gestione degli acquisti, in particolare fornisce le funzioni di:

* Acquisto biglietti
* Acquisto biglietti da prenotazione



#### **GestioneAcquistiPresentationLayer**

Include tutti gli elementi dell’interfaccia grafica che offrono funzionalità riguardanti la gestione degli acquisti. Comprende:

* GUIAcquistoBiglietti: comprende le interfacce che consentono ad un Utente Base di acquistare biglietti e ad un Operatore di vendere biglietti.
* GUIAcquistoBigliettiDaPrenotazione: comprende le interfacce che consentono ad un Utente Base di acquistare biglietti a partire da una prenotazione e ad un Operatore di vendere biglietti a partire da una prenotazione.

#### **GestioneAcquistiApplicationLayer**

Include tutte le componenti che si occupano del corretto funzionamento del sistema nella gestione degli acquisti. Comprende:

* AcquistoBiglietti(): operazioni per l’acquisto e la vendita dei biglietti.
* AcquistoBigliettiDaPrenotazione(): operazioni per l’acquisto e la vendita di biglietti prenotati.

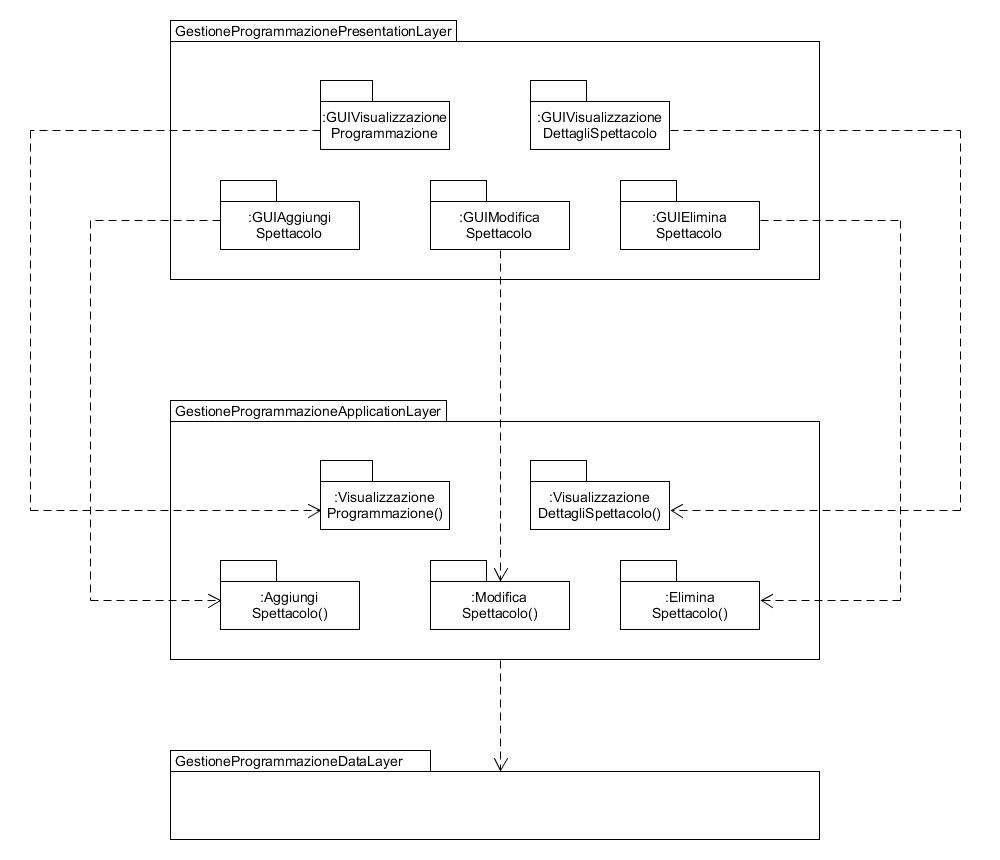
#### **GestioneAcquistiDataLayer**

Si occupa di rendere reperibili i dati, presenti all’interno del database, relativi agli acquisti.

## **2.2.6 Gestione Programmazione**

Questo sottosistema si occupa della gestione della programmazione, in particolare fornisce le funzioni di:

* Visualizzazione della programmazione
* Visualizzazione dei dettagli di uno spettacolo
* Aggiunta di uno spettacolo
* Modifica di uno spettacolo
* Eliminazione di uno spettacolo



#### **GestioneProgrammazionePresentationLayer**

Include tutti gli elementi dell’interfaccia grafica che offrono funzionalità riguardanti la gestione della programmazione degli spettacoli.

L’interfaccia sarà accessibile dall’ Ospite, ma solo il Gestore potrà usufruire delle funzioni di Aggiunta, Modifica ed Eliminazione di uno spettacolo.

Comprende:

* GUIVisualizzazioneProgrammazione: tramite questa interfaccia l’Ospite può visualizzare gli spettacoli in programmazione.
* GUIVisualizzazioneDettagliSpettacolo: tramite questa interfaccia l’Ospite può visualizzare i dettagli di uno spettacolo selezionato, e il Gestore potrà inoltre accedere alle funzionalità di Modifica o Eliminazione dello spettacolo.
* GUIAggiungiSpettacolo: tramite questa interfaccia il Gestore può aggiungere uno spettacolo alla programmazione.
* GUIModificaSpettacolo: tramite questa interfaccia il Gestore può modificare lo spettacolo selezionato.
* GUIEliminaSpettacolo: tramite questa interfaccia il Gestore può eliminare dalla programmazione lo spettacolo selezionato.

#### **GestioneProgrammazioneApplicationLayer**

Include tutte le componenti che si occupano del corretto funzionamento del sistema, in particolare della gestione della programmazione degli spettacoli. Agisce con il Database per recuperare i dati degli spettacoli e per aggiornarli in caso di aggiunta, modifica o eliminazione.

Comprende:

* VisualizzazioneProgrammazione(): operazioni per visualizzare gli spettacoli in programmazione.
* VisualizzazioneDettagliSpettacolo(): operazioni per visualizzare i dettagli di uno spettacolo selezionato e per accedere alle funzionalità di Modifica o Eliminazione dello spettacolo (queste ultime esclusive per il Gestore).
* AggiungiSpettacolo(): esegue l’operazione di aggiunta di uno spettacolo alla programmazione.
* ModificaSpettacolo(): esegue l’operazione di modifica dello spettacolo selezionato.
* EliminaSpettacolo(): esegue l’operazione di eliminazione dalla programmazione dello spettacolo selezionato.

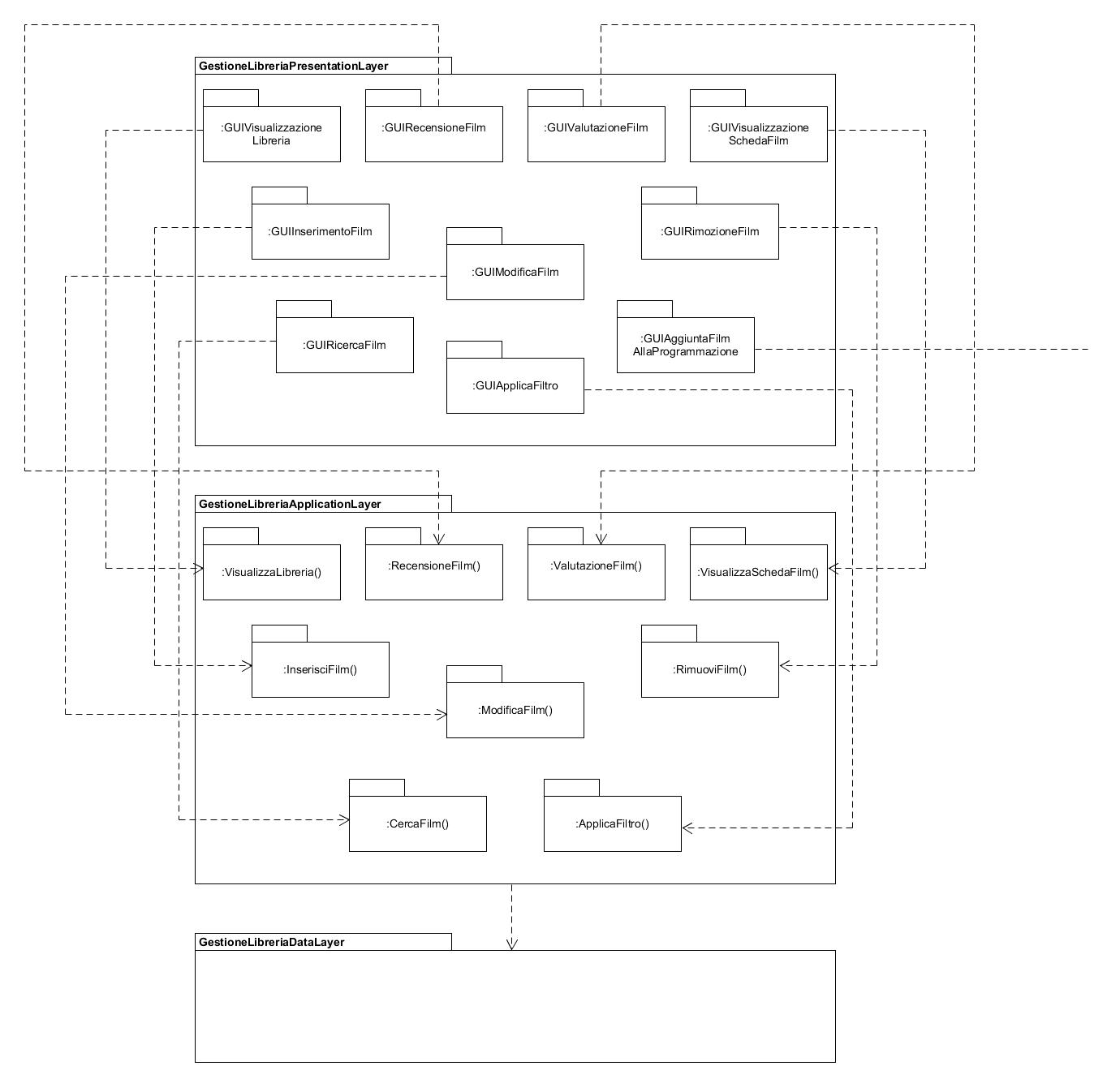
#### **GestioneProgrammazioneDataLayer**

Si occupa di rendere reperibili i dati, presenti all’interno del database, relativi agli Spettacoli in programmazione.

## **2.2.7 Gestione Libreria**

Questo sottosistema si occupa della gestione della libreria, in particolare fornisce le funzioni di:

* Visualizzazione della libreria
* Visualizzazione delle recensione del film
* Visualizzazione del singolo film
* Visualizzazione della scheda del film
* Rimozione di un film
* Inserimento di un film
* Ricerca di un film
* Modifica di un film
* Aggiunta di un film in libreria
* Applicazione di filtri alla visualizzazione



#### **GestioneLibreriaPresentationLayer**

Include tutti gli elementi dell’interfaccia grafica che offrono funzionalità riguardanti la gestione della libreria.

L’interfaccia sarà accessibile dall’ Ospite, solo il Gestore potrà usufruire delle funzioni di Aggiunta, AggiuntaAllaProgrammazione, Modifica e Rimozione di un film, solo l’UtenteBase potrà usufruire delle funzioni di Valutazione e Recensione.

Comprende:

* GUIVisualizzazioneLibreria: tramite questa interfaccia l’Ospite può visualizzare la lista di tutti i film presenti in libreria e la lista dei film risultanti da una ricerca o dall’applicazione di un filtro.
* GUIRicercaFilm: tramite questa interfaccia l’Ospite può effettuare una ricerca inserendo delle keywords nell’apposito campo.
* GUIApplicaFiltro: tramite questa interfaccia l’Ospite può selezionare uno dei filtri di ricerca per modificare la lista di film nella GUIVisualizzazioneLibreria.
* GUIVisualizzazioneSchedaFilm: tramite questa interfaccia l’Ospite visualizza le informazioni relative al film precedentemente selezionato
* GUIValutazioneFilm: tramite questa interfaccia l’UtenteBase può inserire una valutazione inerente a un film indicando il valore scelto sulle stelline che da grigie diventeranno gialle.
* GUIRecensioneFilm: tramite questa interfaccia l’UtenteBase, dopo aver inserito la valutazione di un film, può inserire una recensione digitando nell’apposito campo.
* GUIInserimentoFilm: questa interfaccia, accessibile solamente dal Gestore, visualizza il form per l’inserimento di un film in libreria.
* GUIModificaFilm: questa interfaccia, accessibile solamente dal Gestore, visualizza il form per la modifica delle informazioni inerenti a un film.
* GUIRimozioneFilm: tramite questa interfaccia il Gestore può rimuovere un film dalla libreria.
* GUIAggiuntaFilmAllaProgrammazione: tramite questa interfaccia il Gestore può aggiungere un film alla programmazione.

#### **GestioneLibreriaApplicationLayer**

Include tutte le componenti che si occupano del corretto funzionamento del sistema, in particolare della gestione della Libreria. Agisce con il Database per recuperare i dati dei film e per aggiornarli in caso di aggiunta, modifica o eliminazione.

Comprende:

* VisualizzaLibreria(): esegue l’operazione di visualizzare le liste dei film.
* CercaFilm(): esegue le operazioni di ricerca dei film che corrispondono alle keyword inserite dall’Ospite.
* ApplicaFiltro(): esegue le operazioni relative all’applicazione dei filtri sulle liste dei film.
* VisualizzaSchedaFilm(): esegue le operazioni relative alla visualizzazione delle informazioni di un film e per accedere alle funzioni di: Valutazione e Recensione da parte dell’UtenteBase; Modifica Rimozione e AggiuntaAllaProgrammazione da parte del Gestore.
* ValutazioneFilm(): esegue le operazioni relative all’inserimento di una valutazione da parte di un UtenteBase.
* RecensioneFilm(): esegue le operazioni relative all’inserimento di una recensione da parte di un UtenteBase.
* InserisciFilm(): esegue l’operazione di inserimento di un film alla libreria.
* ModificaFilm(): esegue l’operazione di modifica di un film della libreria.
* RimuoviFilm():esegue l’operazione di rimozione di un film dalla libreria.

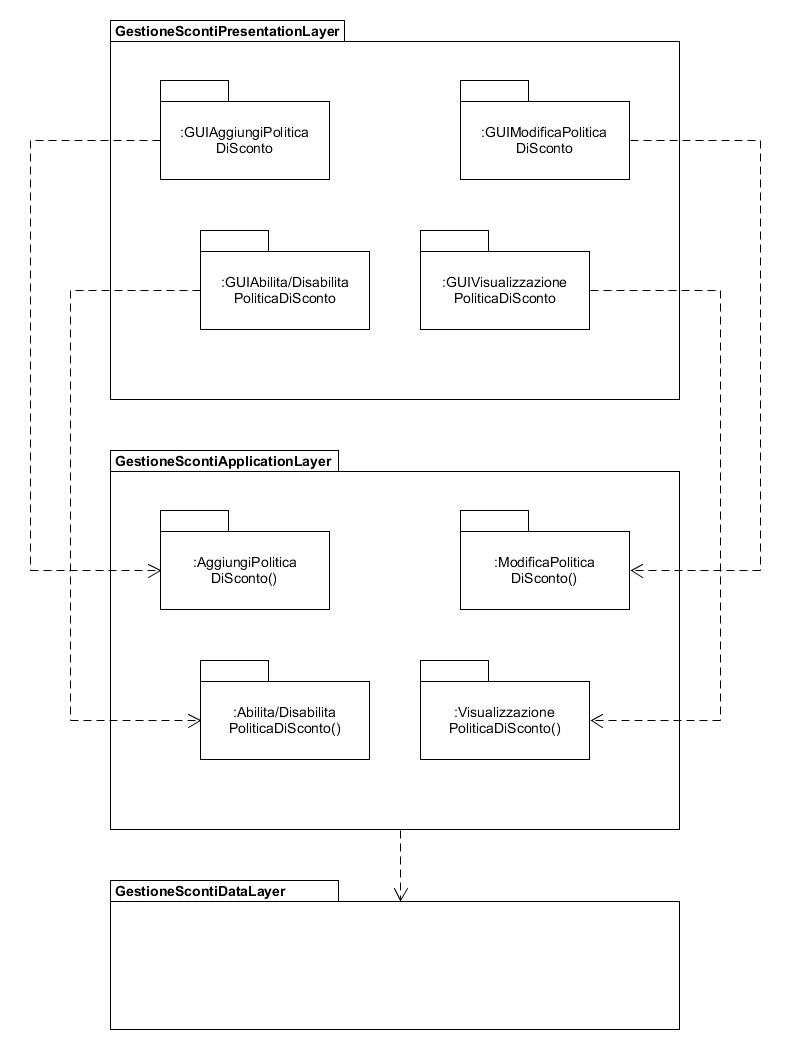
#### **GestioneLibreriaDataLayer**

Si occupa di rendere reperibili i dati, presenti all’interno del database, relativi ai film.

## **2.2.8 Gestione Sconti**

Questo sottosistema si occupa della gestione degli sconti, in particolare fornisce le funzioni di:

* Aggiunta di una politica di sconto
* Modifica di una politica di sconto
* Abilitazione o disabilitazione di una politica di sconto
* Visualizzazione di una politica di sconto



#### **GestioneScontiPresentationLayer**

Include tutti gli elementi dell’interfaccia grafica che offrono funzionalità riguardanti la gestione della libreria.

L’interfaccia sarà accessibile solamente dal Gestore.

Comprende:

* GUIAggiungiPoliticaDiSconto:
* GUIModificaPoliticaDiSconto:
* GUIAbilita/DisabilitaPoliticaDiSconto:
* GUIVisualizzazionePoliticaDiSconto:

#### **GestioneScontiApplicationLayer**

Include tutte le componenti che si occupano del corretto funzionamento del sistema, in particolare della gestione degli Sconti. Agisce con il Database per recuperare i dati degli sconti e per aggiornarli in caso di aggiunta, modifica, abilitazione o disabilitazione.

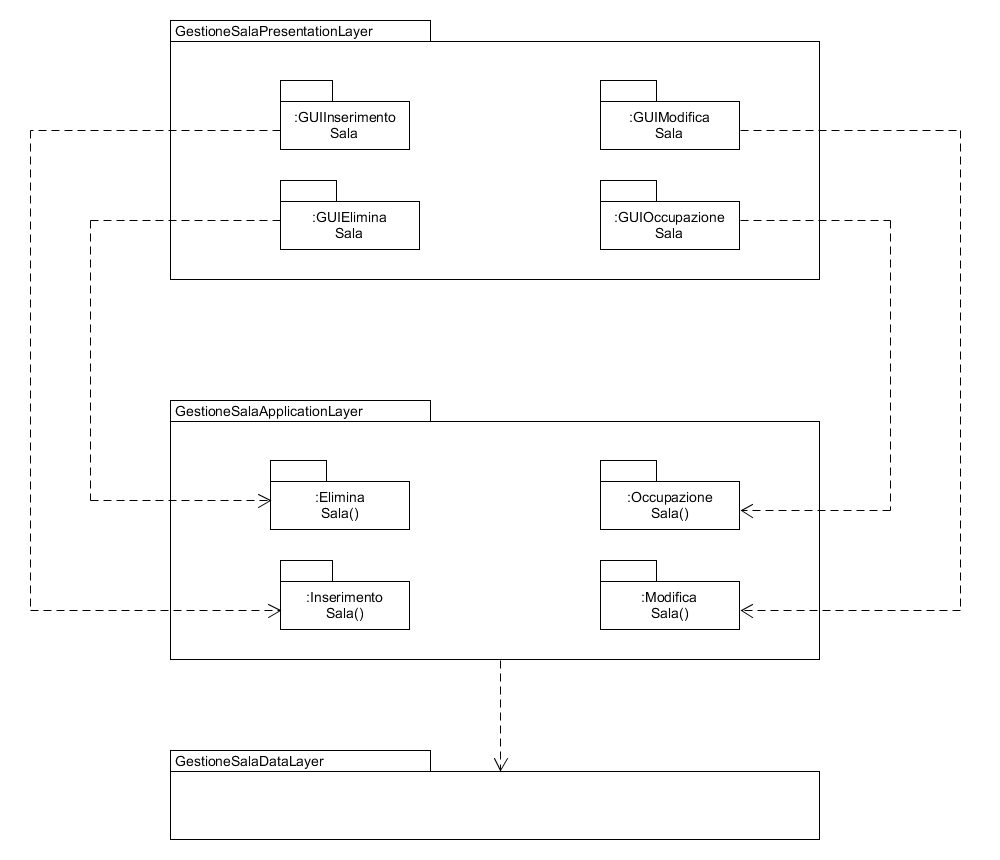
Comprende:

* AggiungiPoliticaDiSconto():
* ModificaPoliticaDiSconto():
* Abilita/DisabilitaPoliticaDiSconto():
* VisualizzazionePoliticaDiSconto():

## **2.2.9 Gestione Sala**

Questo sottosistema si occupa della gestione delle sale, in particolare fornisce le funzioni di:

* Inserimento sala
* Modifica sala
* Elimina sala
* Occupazione sala



#### **GestioneSalaPresentationLayer**

Include tutti gli elementi dell’interfaccia grafica che offrono funzionalità riguardanti la gestione delle sale del cinema.

L’interfaccia sarà accessibile esclusivamente dal Gestore che potrà usufruire delle funzioni di Aggiunta, Modifica, Eliminazione ed Occupazione di un’ intera sala.

Comprende:

* GUIInserimentoSala: tramite questa interfaccia il Gestore può aggiungere una nuova sala nel sistema.
* GUIModificaSala: tramite questa interfaccia il Gestore può modificare la struttura della sala, tra cui la disposizione dei posti.
* GUIEliminaSala: tramite questa interfaccia il Gestore può eliminare una sala dal sistema.
* GUIOccupazioneSala: tramite questa interfaccia il Gestore può rendere indisponibile un’intera sala in una determinata data.

#### **GestioneSalaApplicationLayer**

Include tutte le componenti che si occupano del corretto funzionamento del sistema, in particolare della gestione delle sale del cinema. Agisce con il Database per recuperare i dati delle sale e per aggiornarli in caso di aggiunta, modifica, eliminazione o occupazione

Comprende:

* InserimentoSala(): operazione per aggiungere una nuova sala nel sistema.
* ModificaSala(): operazione per modificare la struttura della sala.
* EliminaSala(): operazione per eliminare una sala dal sistema.
* OccupazioneSala(): operazione per rendere indisponibile un’intera sala in una determinata data.

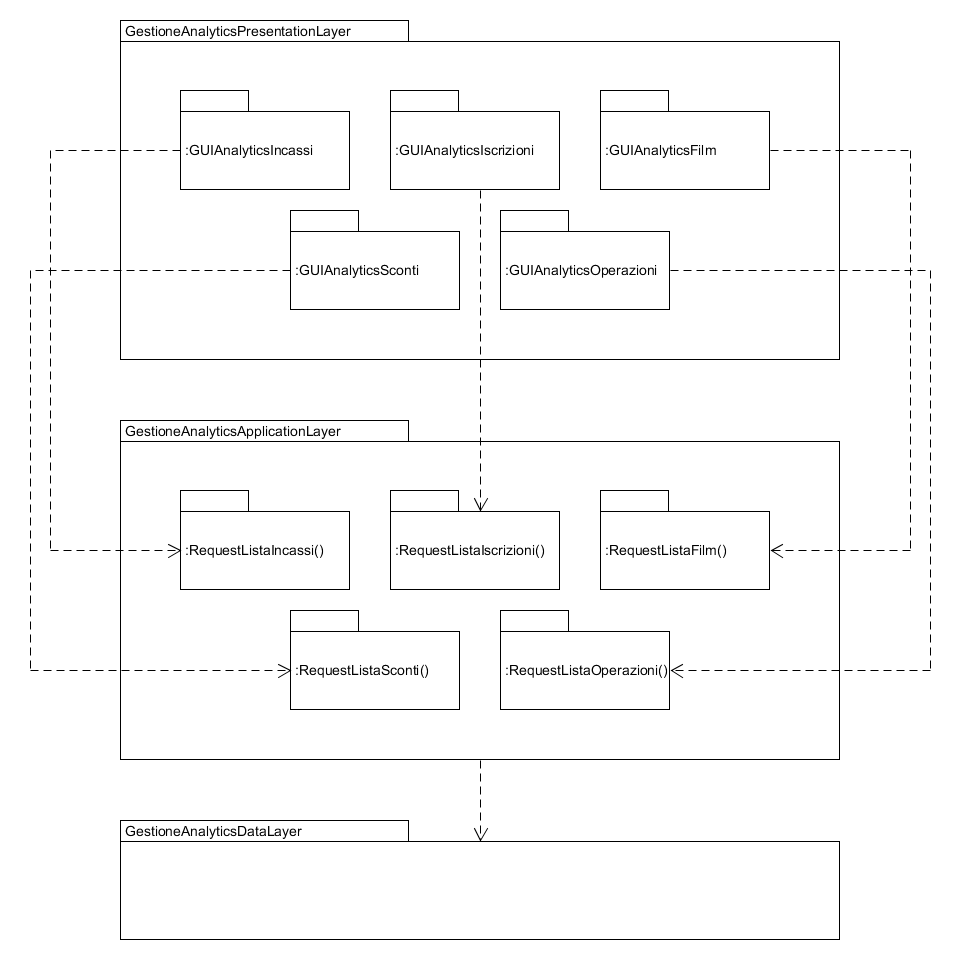
#### **GestioneSalaDataLayer**

Si occupa di rendere reperibili i dati, presenti all’interno del database, relativi alle Sale del cinema.

## **2.2.10 Gestione Analytics**

Questo sottosistema si occupa della gestione delle analytics, in particolare fornisce le funzioni di visualizzazione dei grafici di:

* Incassi
* Iscrizioni
* Film
* Sconti
* Operazioni



#### **GestioneAnalyticsPresentationLayer**

Include tutti gli elementi dell’interfaccia grafica che offrono funzionalità riguardanti la gestione delle analytics. Comprende:

* GUIAnalyticsIncassi: comprende le interfacce che consentono al Gestore di visualizzare le analytics relative agli incassi.
* GUIAnalyticsIscrizioni: comprende le interfacce che consentono al Gestore di visualizzare le analytics relative alle iscrizioni.
* GUIAnalyticsFilm: comprende le interfacce che consentono al Gestore di visualizzare le analytics relative ai film.
* GUIAnalyticsSconti: comprende le interfacce che consentono al Gestore di visualizzare le analytics relative agli sconti.
* GUIAnalyticsOperazioni: comprende le interfacce che consentono al Gestore di visualizzare le analytics relative alle operazioni.

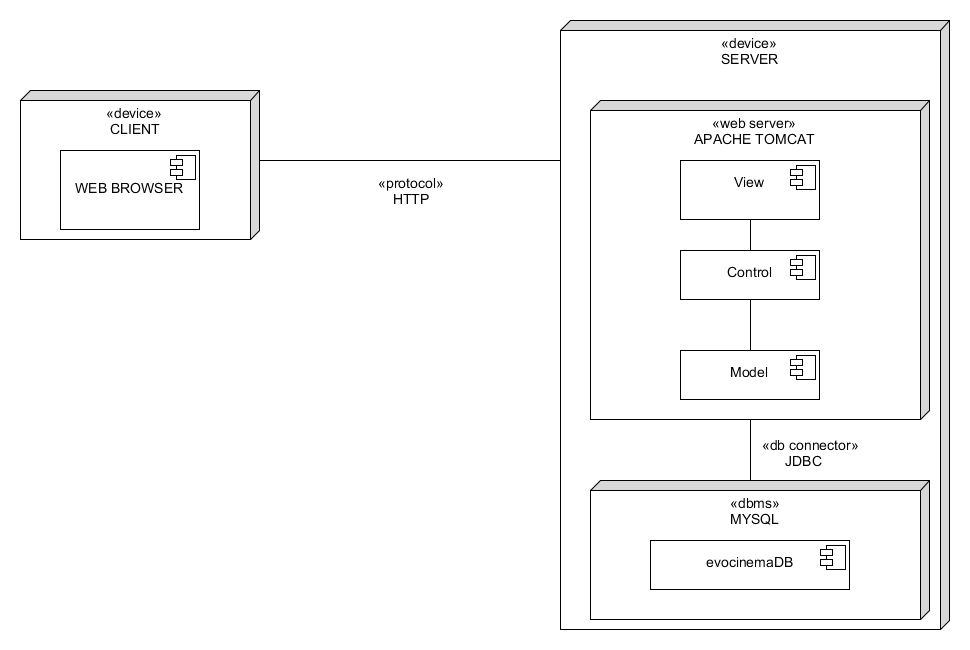
#### **GestioneAnalyticsApplicationLayer**

* RequestListaIncassi(): operazioni per l’accesso al sistema.
* RequestListaSconti(): operazioni per uscire dal sistema.
* RequestListaIscrizioni(): operazioni per registrarsi al sistema.
* RequestListaOperazioni(): operazioni per cancellare il proprio account dal sistema.
* RequestListaFilm(): operazioni per il recupero della password.

#### **GestioneAnalyticsDataLayer**

Si occupa di rendere reperibili i dati, presenti all’interno del database, relativi a tutti gli elementi analizzati dal sistema di Analytics.

## **2.3 Mappatura Hardware/Software**

****

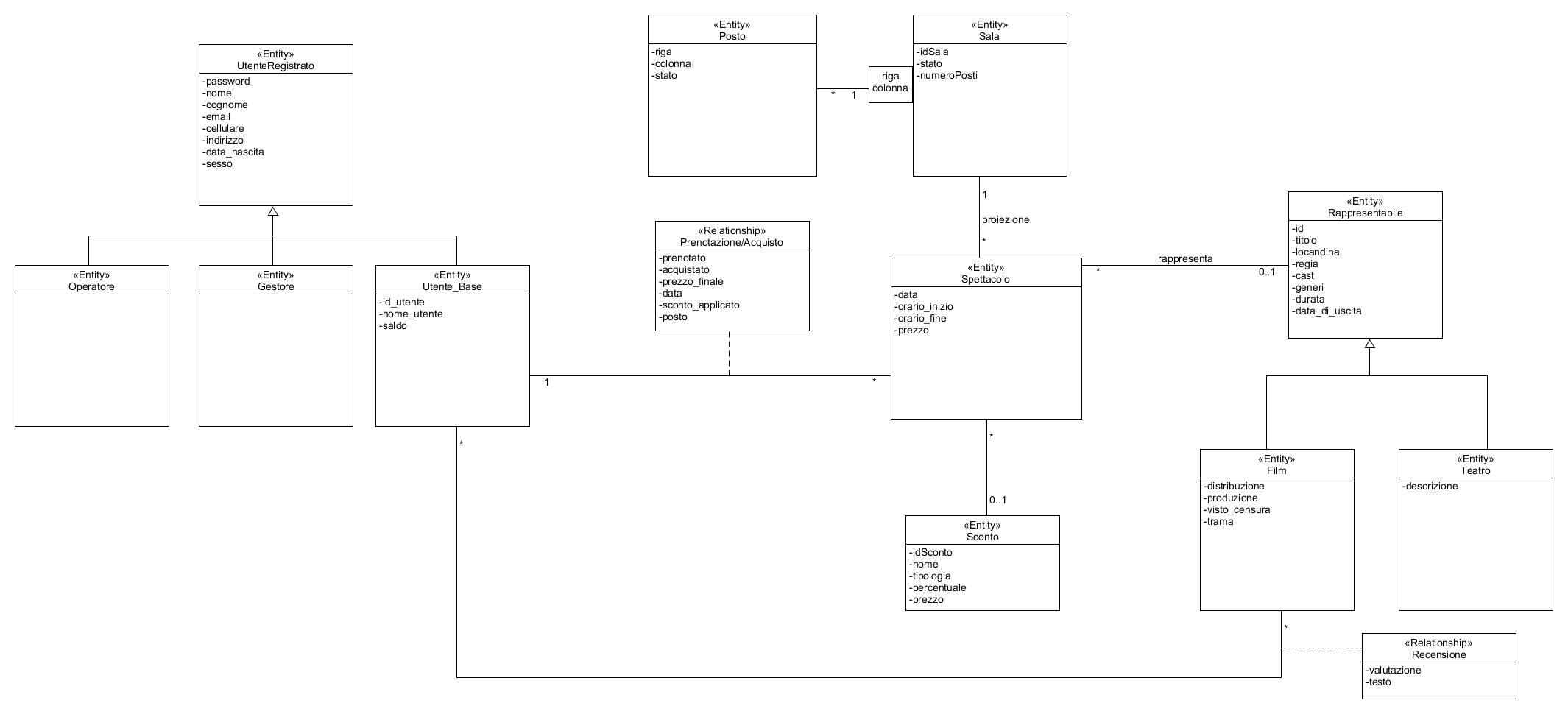
Il sistema utilizza un’architettura client/server, in cui un server fornisce servizi a più client. Su una macchina client è eseguito un browser web che consente all’utente di interagire a livello View (lato server) per inoltrare richieste e visualizzare le risposte ricevute. La macchina server gestisce la logica applicativa e i dati persistenti. La comunicazione tra client e server avviene tramite protocollo HTTP. Questo protocollo permette di trasferire ipertesti, consentendo a due macchine, client e server, di interagire attraverso un meccanismo di richiesta e risposta. Il client inoltra una richiesta al server che verrà soddisfatta con la risposta di quest’ultimo.

Per il client, le specifiche hardware sono una qualsiasi macchina dotata di connessione a internet, mentre per quel che riguarda il software, un sistema operativo con un web browser installato.

Per il server, invece, le specifiche hardware consistono di una macchina connessa a internet capace di immagazzinare dati a sufficienza. Le specifiche software necessarie comprendono un Database Management System (MySQL) per la gestione dei dati persistenti, un Web Server (Apache Tomcat) per la gestione della logica applicativa e della comunicazione con più client.

## **2.4 Gestione dei dati persistenti**

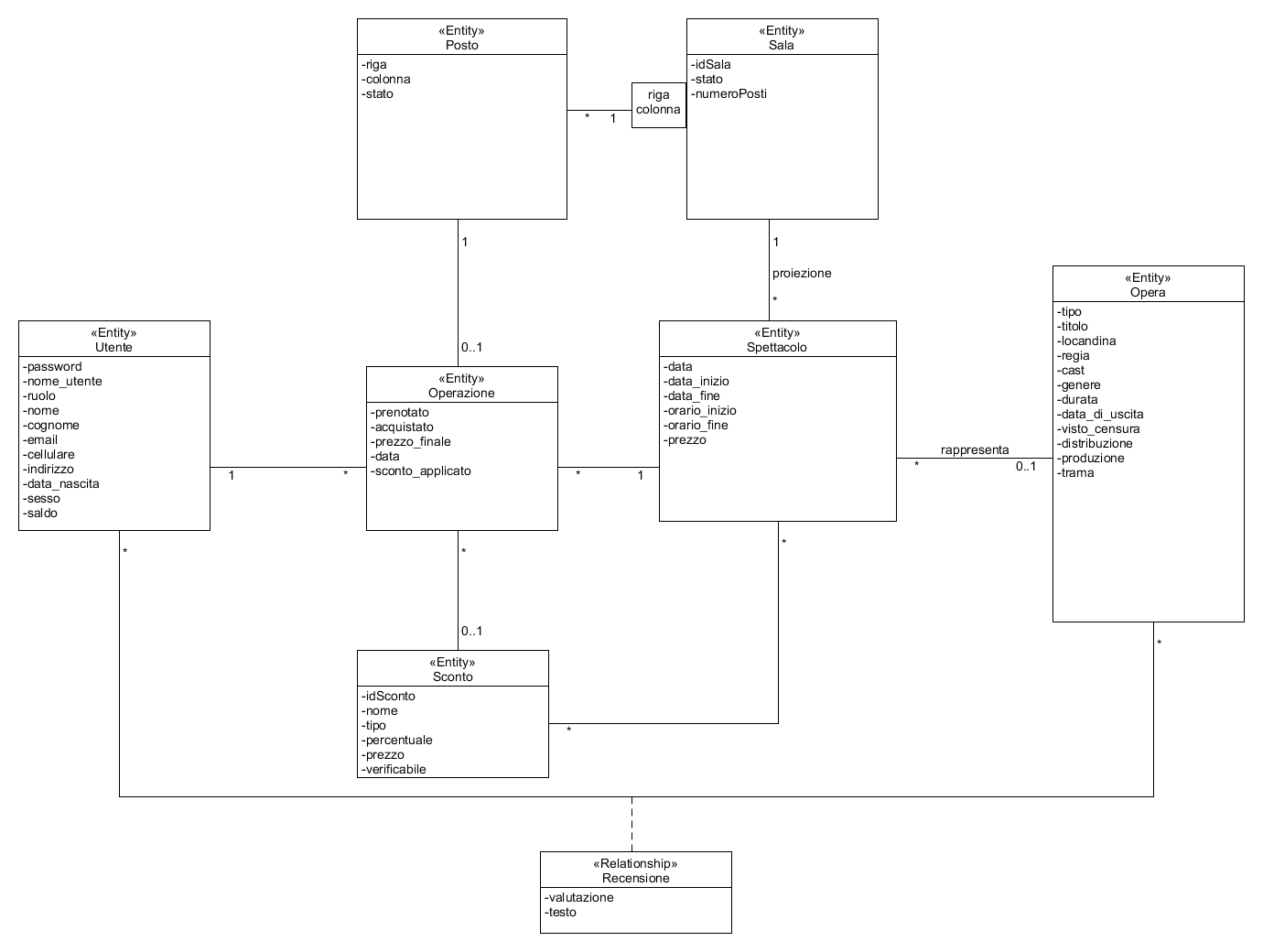
### **2.4.1 Modello E-R**

****

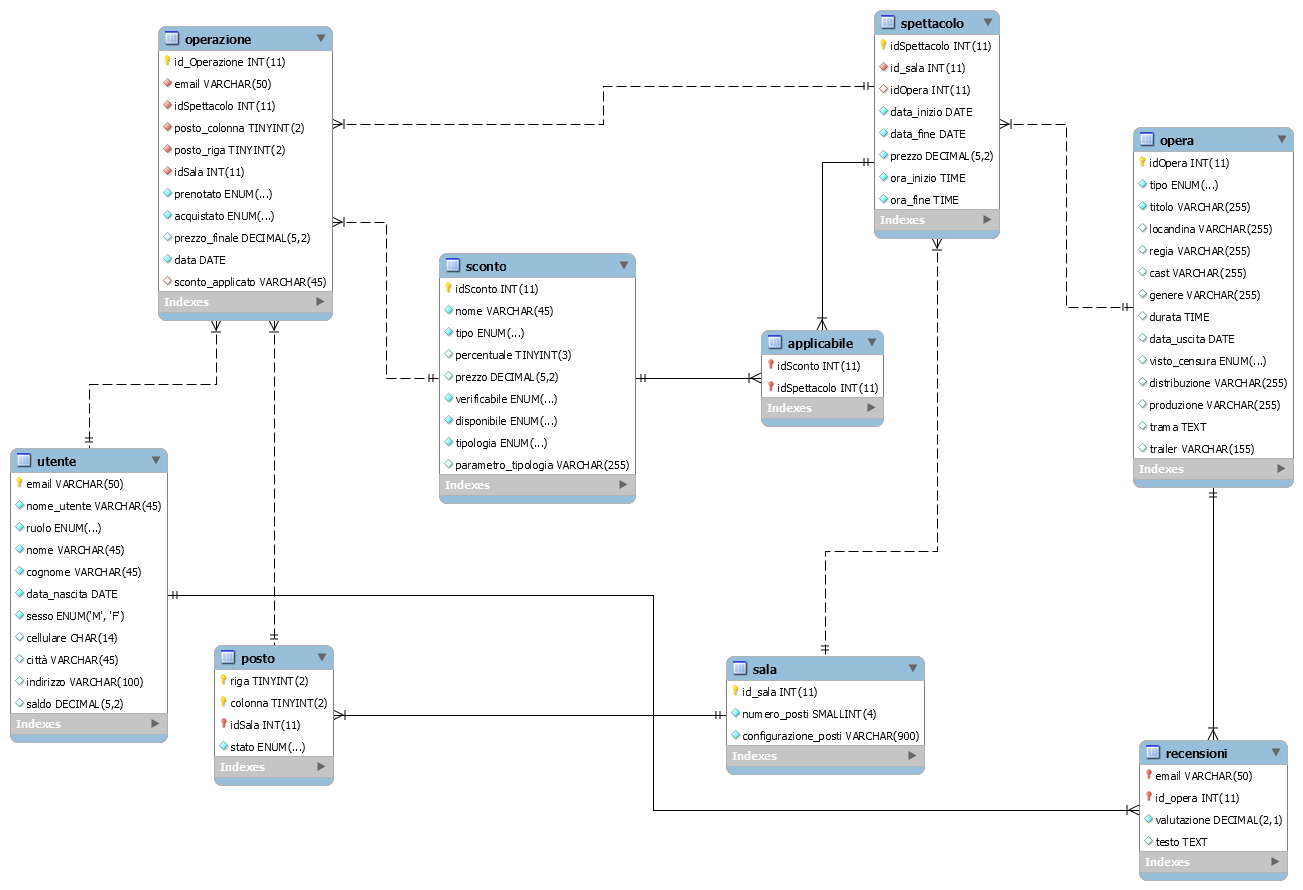
### 

### 

### **2.4.2 Modello E-R Ristrutturato**



### **2.4.3 Modello logico**



### **2.4.4 Struttura delle tabelle**

**Tabella: Utente**

|  |  |  |
| --- | --- | --- |
| **Attributo** | **Tipo** | **Chiave** |
| email | VARCHAR(50) | Primaria |
| nome\_utente | VARCHAR(45) |  |
| ruolo | ENUM(...) |  |
| nome | VARCHAR(45) |  |
| cognome | VARCHAR(45) |  |
| data\_nascita | DATE |  |
| sesso | ENUM(‘M’,’F’) |  |
| cellulare | CHAR(14) |  |
| città | VARCHAR(45) |  |
| indirizzo | VARCHAR(100) |  |
| saldo | DECIMAL(5,2) |  |

**Tabella: Operazione**

|  |  |  |
| --- | --- | --- |
| **Attributo** | **Tipo** | **Chiave** |
| id\_Operazione | INT(11) | Primaria |
| email | VARCHAR(50) | Esterna |
| idSpettacolo | INT(11) | Esterna |
| posto\_colonna | TINYINT(2) | Esterna |
| posto\_riga | TINYINT(2) | Esterna |
| idSala | INT(11) | Esterna |
| prenotato | ENUM(...) |  |
| acquistato | ENUM(...) |  |
| prezzo\_finale | DECIMAL(5,2) |  |
| data | DATE |  |
| sconto\_applicato | VARCHAR(45) |  |

**Tabella: Sconto**

|  |  |  |
| --- | --- | --- |
| **Attributo** | **Tipo** | **Chiave** |
| idSconto | INT(11) | Primaria |
| nome | VARCHAR(45) |  |
| tipo | ENUM(...) |  |
| percentuale | TINYINT(3) |  |
| prezzo | DECIMAL(5,2) |  |
| verificabile | ENUM(...) |  |
| disponibile | ENUM(...) |  |
| tipologia | ENUM(...) |  |
| parametro\_tipologia | VARCHAR(255) |  |

**Tabella: Posto**

|  |  |  |
| --- | --- | --- |
| **Attributo** | **Tipo** | **Chiave** |
| riga | TINYINT(2) | Primaria |
| colonna | TINYINT(2) | Primaria |
| idSala | INT(11) | Esterna |
| stato | ENUM(...) |  |

**Tabella: Spettacolo**

|  |  |  |
| --- | --- | --- |
| **Attributo** | **Tipo** | **Chiave** |
| idSpettacolo | INT(11) | Primaria |
| id\_Sala | INT(11) | Esterna |
| idOpera | INT(11) | Esterna |
| data\_inizio | DATE |  |
| data\_fine | DATE |  |
| prezzo | DECIMAL(5,2) |  |
| ora\_inizio | TIME |  |
| ora\_fine | TIME |  |

**Tabella: Applicabile**

|  |  |  |
| --- | --- | --- |
| **Attributo** | **Tipo** | **Chiave** |
| idSconto | INT(11) | Esterna |
| idSpettacolo | INT(11) | Esterna |

**Tabella: Sala**

|  |  |  |
| --- | --- | --- |
| **Attributo** | **Tipo** | **Chiave** |
| idSala | INT(11) | Primaria |
| numero\_posti | SMALLINT(4) |  |
| configurazione\_posti | VARCHAR(900) |  |

**Tabella: Opera**

|  |  |  |
| --- | --- | --- |
| **Attributo** | **Tipo** | **Chiave** |
| idOpera | ENUM(...) | Primaria |
| tipo | VARCHAR(255) |  |
| titolo | VARCHAR(255) |  |
| locandina | VARCHAR(255) |  |
| regia | VARCHAR(255) |  |
| cast | VARCHAR(255) |  |
| genere | VARCHAR(255) |  |
| durata | TIME |  |
| data\_uscita | DATE |  |
| visto\_censura | ENUM(...) |  |
| distribuzione | VARCHAR(255) |  |
| produzione | VARCHAR(255) |  |
| trama | TEXT |  |
| trailer | VARCHAR(155) |  |

**Tabella: Recensioni**

|  |  |  |
| --- | --- | --- |
| **Attributo** | **Tipo** | **Chiave** |
| email | VARCHAR(50) | Esterna |
| id\_opera | INT(11) | Esterna |
| valutazione | DECIMAL(2,1) |  |
| testo | TEXT |  |

## 

## **2.5 Gestione degli accessi**

Essendo EvoCinema un sistema multiutente, diversi tipi di utenti possono visualizzare differenti insiemi di oggetti ed invocare differenti operazioni su di essi.

Per documentare in modo succinto i diritti di accesso, sfruttiamo una matrice rappresentata dalla seguente tabella, contenente i diversi attori del sistema sulle colonne, i diversi oggetti sulle righe, e le operazioni consentite ad ognuno degli attori su ognuno degli oggetti nella cella corrispondente.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Attori**  **Oggetti** | **Ospite** | **Gestore** | **Operatore** | **Utente Base** |
| -Utente | -Registrazione()  -Login() | -RequestLista  Iscrizioni()  -Logout()  -CancellaAccount()  -RecuperoPassword()  -VisualizzaAccount()  -ModificaAccount() | -RicaricaSaldo()  -Logout()  -Recupero  Password()  -Visualizza  Account()  -ModificaAccount() | -Logout()  -CancellaAccount()  -RecuperoPassword()  -VisualizzaAccount()  -ModificaAccount()  -VisaulizzaSaldo() |
| -Sala |  | -InserimentoSala() -ModificaSala() -OccupazioneSala() -EliminaSala() |  |  |
| -Opera | -VisualizzaLibreria()  -VisualizzaSchedaFilm()  -CercaFilm()  -ApplicaFiltro() | -InserisciFilm()  -ModificaFilm()  -RimuoviFilm()  -RequestListaFilm() |  |  |
| -Recensione |  |  |  | -ValutazioneFilm()  -RecensioneFilm() |
| -Spettacolo | -VisualizzazioneProgrammazione()  -VisualizzazioneDettagli  Spettacolo() | -AggiungiSpettacolo()  -ModificaSpettacolo()  -EliminaSpettacolo() |  |  |
| -Sconto |  | -AggiuntaPoliticaDi  Sconto()  -ModificaPoliticaDi  Sconto()  -AbilitaPoliticaDi  Sconto()  -DisabilitaPoliticaDi  Sconto()  -VisualizzazionePoliticaDiSconto()  -RequestListaSconti() |  |  |
| -Operazione |  | -RequestListaIncassi()  -RequestLista  Operazioni() | -Prenotazione  Biglietti()  -DisdettaBiglietti()  -VenditaBiglietti  -VenditaBigliettiDa  Prenotazione() | -PrenotazioneBiglietti()  -DisdettaBiglietti()  -AcquistoBiglietti()  -AcquistoBigliettiDa  Prenotazione() |

# 

# **3. Servizi dei sottosistemi**

## **3.1. SS\_ACC - Gestione account**

|  |  |
| --- | --- |
| **Sottosistema** | **Descrizione** |
| Gestione account | Gestisce le operazioni riguardanti la gestione dell’account di un utente del sistema |
| **Servizio** | **Descrizione** |
| Login() | Permette ad un utente registrato di effettuare l’accesso al sistema. |
| Logout() | Permette ad un ad un utente registrato di effettuare il logout dal sistema. |
| Registrazione() | Permette ad un utente non registrato di effettuare la registrazione al sistema. |
| CancellaAccount() | Permette ad un utente base o un gestore di cancellare l’account. |
| RecuperoPassword() | Permette ad un utente registrato di richiedere una nuova password. |
| VisualizzaAccount() | Permette ad un utente registrato di visualizzare le informazioni relative al proprio account. |
| ModificaAccount() | Permette ad un utente registrato di modificare le informazioni relative al proprio account. |
| VisualizzaSaldo() | Permette ad un utente base di visualizzare le informazioni relative al proprio saldo. |
| RicaricaSaldo() | Permette ad un operatore di ricaricare il saldo di un utente. |

## 

## **3.2. SS\_SAL - Gestione Sala**

|  |  |
| --- | --- |
| **Sottosistema** | **Descrizione** |
| Gestione Sala | Sottosistema che gestisce le operazioni riguardanti la gestione di una sala del cinema. |
| **Servizio** | **Descrizione** |
| EliminaSala() | Permette ad un gestore di eliminare una sala. |
| InserimentoSala() | Permette ad un gestore di inserire una nuova sala. |
| ModificaSala() | Permette ad un gestore di modificare una sala. |
| OccupazioneSala() | Permette ad un gestore di modificare lo stato di una sala. |

## **3.2 SS\_LIB - Gestione Libreria**

|  |  |
| --- | --- |
| **Sottosistema** | **Descrizione** |
| Gestione Libreria | Gestisce le operazioni riguardanti la gestione della libreria dei film. |
| **Servizio** | **Descrizione** |
| VisualizzaLibreria() | Permette ad un utente registrato o ad un ospite di visualizzare la libreria contenente i film. |
| RecensioneFilm() | Permette ad un utente base di aggiungere una recensione ad un film in libreria. |
| VisualizzaSchedaFilm() | Permette ad un utente registrato o ad un ospite di visualizzare la scheda di un film. |
| ValutazioneFilm() | Permette ad un utente base di valutare un film in libreria. |
| InserisciFilm() | Permette ad un gestore di inserire un film in libreria. |
| ModificaFilm() | Permette ad un gestore di modificare un film in libreria. |
| RimuoviFilm() | Permette ad un gestore di eliminare un film dalla libreria. |
| CercaFilm() | Permette ad un ospite di effettuare la ricerca di un film attraverso parole chiave. |
| ApplicaFiltro() | Permette ad un ospite di ottenere una lista di film attraverso l’applicazione di filtri. |

## **3.3 SS\_PRG - Gestione programmazione**

|  |  |
| --- | --- |
| **Sottosistema** | **Descrizione** |
| Gestione programmazione | Gestisce le operazioni riguardanti la gestione della programmazione del cinema. |
| **Servizio** | **Descrizione** |
| VisualizzazioneProgrammazione | Permette ad un ospite di visualizzare tutti gli spettacoli in programmazione con le relative informazioni. |
| VisualizzazioneDettagliSpettacolo | Permette ad un ospite di visualizzare i dettagli di uno spettacolo in programmazione. |
| AggiungiSpettacolo | Permette ad un gestore di aggiungere un nuovo spettacolo alla programmazione. |
| ModificaSpettacolo | Permette ad un gestore di modificare uno spettacolo in programmazione. |
| EliminaSpettacolo | Permette ad un gestore di eliminare uno spettacolo dalla programmazione. |

## **3.4. SS\_SCN - Gestione sconti**

|  |  |
| --- | --- |
| **Sottosistema** | **Descrizione** |
| Gestione sconti | Gestisce le operazioni riguardanti la gestione della libreria dei film. |
| **Servizio** | **Descrizione** |
| AggiuntaPoliticaDiSconto() | Permette ad un gestore di aggiungere una nuova politica di sconto. |
| ModificaPoliticaDiSconto() | Permette ad un gestore modificare una politica di sconto. |
| Abilita/DisabilitaPoliticaDiSconto() | Permette a un Gestore di abilitare o disabilitare una politica di sconto. |
| VisualizzazionePoliticaDiSConto() | Permette a un Gestore di visualizzare le politiche di sconto. |

## 

## **3.5. SS\_PRN - Gestione prenotazioni**

|  |  |
| --- | --- |
| **Sottosistema** | **Descrizione** |
| Gestione prenotazioni | Sottosistema che gestisce le operazioni riguardanti la gestione delle prenotazioni |
| **Servizio** | **Descrizione** |
| PrenotazioneBiglietti() | Permette a un utente base o ad un operatore la prenotazione di un biglietto |
| DisdettaBiglietti() | Permette a un utente base o ad un operatore la disdetta di un biglietto prenotato |

# 

## **3.6. SS\_ACQ - Gestione acquisti**

|  |  |
| --- | --- |
| **Sottosistema** | **Descrizione** |
| Gestione acquisti | Gestisce le operazioni riguardanti la gestione degli acquisti |
| **Servizio** | **Descrizione** |
| AcquistoBiglietti() | Permette ad un Utente base di acquistare biglietti per uno spettacolo |
| VenditaBiglietti() | Permette ad un Operatore di vendere biglietti per uno spettacolo |
| AcquistoBigliettiDaPrenotazione() | Permette ad un Utente base di acquistare biglietti prenotato per uno spettacolo |
| VenditaBigliettiDaPrenotazione() | Permette ad un Operatore di vendere biglietti prenotati per uno spettacolo |

## 

## **3.7. SS\_ANL - Gestione analytics**

|  |  |
| --- | --- |
| **Sottosistema** | **Descrizione** |
| Gestione analytics | Permette ad un gestore di accedere alla sezione analytics. |
| **Servizio** | **Descrizione** |
| RequestListaIncassi() | Permette ad un gestore di visualizzare un grafico relativo agli incassi. |
| RequestListaIscrizioni() | Permette ad un gestore di visualizzare un grafico relativo a tutte le informazioni degli utenti. |
| RequestListaFilm() | Permette ad un gestore di visualizzare un grafico relativo a tutte le informazioni riguardanti i film. |
| RequestListaSconti() | Permette ad un gestore di visualizzare un grafico relativo a tutte le informazioni sugli sconti. |
| RequestListaOperazioni() | Permette ad un gestore di visualizzare un grafico relativo a tutte le operazioni effettuate dagli operatori. |

# **4. Glossario**

|  |  |
| --- | --- |
| **Termine** | **Descrizione** |
| EvoCinema | Nome del sistema che si sta sviluppando. |
| Utente registrato | Il termine identifica una persona correttamente registrata al sistema che ha effettuato l’accesso, comprende gestore, operatore e utente base. |
| Utente non registrato | Il termine identifica una persona che non ha effettuato l’accesso al sistema. |
| Utente base | Il termine identifica un utente registrato che ha la possibilità di effettuare acquisti, prenotazioni e valutazioni. |
| Operatore | Il termine identifica un utente registrato che presta servizio all’interno del cinema e opera direttamente alla cassa. |
| Gestore | Il termina identifica un utente registrato che ha la possibilità di amministrare il sistema. Può essere identificato come il proprietario del cinema o un suo incaricato. |
| Ospite | Il termina identifica una persona generica che interagisce con il sistema, essa può essere un utente base, un gestore, un operatore o un utente non registrato. |
| Deployment Diagram | Diagramma UML che descrive come avviene l’interazione tra la web application e l’hardware. |
| Mapping | Mappatura del sistema. |
| Database | Contenitore dei dati. |
| Schema ER | Schema usato per il modello relazionale del sistema. |
| Modello logico | Schema usato per il modello logico del sistema. |
| Two-Tiers | Architettura scelta per la suddivisione in sottosistemi. |
| Layer | Livello del sottosistema. |
| PresentationLayer | Livello di un sottosistema che contiene le interfacce grafiche agli utenti. |
| ApplicationLayer | Livello di un sottosistema che contiene le procedure e i controlli che vengono  realizzati ad ogni sottomissione di una operazione. |
| DataLayer | Livello che si occupa della gestione dei dati persistenti. |