SDD

System Design Document

Fidelio

Riferimento	NC_16
Versione	2.0
Data	12/12/2025
Destinatario	Prof. Carmine Gravino
Presentato da	Nikolas Tullo, Luca Staiano, Guglielmo Forte
Approvato da	

Revision History

Data	Versione	Descrizione	Autori
10/11/2025	0.1	Prima stesura	NT, LS
10/11/2025	0.2	Inserimento dei Design Goals	NT, LS
11/11/2025	0.3	Scrittura del sistema attuale e inizio scrittura sistema proposto	Tutto il team
11/11/2025	0.4	Stesura Component Diagram e Diagramma Architetturale	NT, LS
17/11/2025	0.5	Stesura Decomposizione dei sottosistemi e Mapping Hardwarer/Software	LS, GF
17/11/2025	0.6	Inizio stesura Gestione dei Dati Persistenti	LS, NT
17/11/2025	0.7	Stesura Controllo degli accessi e della sicurezza	NT
19/11/2025	0.8	Stesura Condizioni limite	LS
22/11/2025	0.9	Stesura servizi dei sottosistemi e schema ER	LS
24/11/2025	1.0	Stesura Glossario	GF
25/11/2025	1.1	Stesura primo design pattern	LS
26/11/2025	1.2	Stesura secondo design pattern	NT
12/12/2025	2.0	Revisione	Tutto il team

Team members

Nome	Ruolo nel progetto	Acronimo	Informazioni di contatto
Luca Staiano	Project Manager	LS	**l.staiano5@studenti.unisa.it**
Guglielmo Forte	Project Manager	GF	**g.forte29@studenti.unisa.it**
Nikolas Tullo	Project Manager	NT	**n.tullo@studenti.unisa.it**

Sommario

1. Introduzione

1.1 Scopo del sistema

Fidelio si propone di creare una community vibrante per gli appassionati di cinema, al fine di valorizzare la cultura cinematografica italiana e internazionale creando uno strumento di condivisione e scoperta per persone interessate al mondo dei film.

Il sistema, gestito da uno o più Amministratori, permette l'iscrizione da parte di tre categorie di utenti: Cinefili (appassionati generici di cinema), Critici (esperti impiegati presso testate giornalistiche che lavorano o collaborano con riviste specializzate e si occupano di animare discussioni tematiche attraverso thread), e Fedeli (professionisti del settore cinematografico come attori, registi o membri della troupe che creano community dedicate per interagire direttamente con i fan).

Le due componenti principali del sistema sono un catalogo cinematografico completo – integrato con API esterne (TMDB/IMDb) che fornisce informazioni dettagliate su film, cast, trailer e sinossi – e un sistema di community e thread di discussione, spazi moderati da Critici e animati da Fedeli, dove gli utenti possono commentare, discutere recensioni e rimanere constantemente aggiornati sull'attuale panorama cinematografico.

Inoltre, al fine di personalizzare l'esperienza di scoperta cinematografica degli utenti, il sistema integra FRA (Fidelio's Recommendation Agent), un'intelligenza artificiale che analizza le recensioni scritte dall'utente, le sue valutazioni e i suoi comportamenti sulla piattaforma per consigliare film con percentuali di affinità personalizzate, generando liste curate dinamicamente nella sezione "Per Te" e ottimizzando continuamente i suggerimenti in base ai gusti individuali dell'utente.

1.2 Obiettivi di Design (Design Goals)

Nella presente sezione si andranno a presentare i Design Goals, le qualità sulle quali il sistema deve essere focalizzato, sono stati suddivisi nelle seguenti categorie:

- **Performance**: Includono i requisiti di spazio e veloctà
- **Dependability**: Determinano quanto sforzo deve essere speso per minimizzare i fallimenti del sistema (crash, falle di sicurezza, etc..) e le loro conseguenze.
- **Maintenance**: Determina quanto sforzo è necessario per modificare il sistema dopo il suo rilascio.
- **End User**: Includono qualità che sono desiderabili dal punto di vista dell'utente, ma che non sono state scoperte dai criteri di Performance e Dependability.

Ciascun Design Goal è descritto da:

- **Rank**, che ne specifica un valore di priorità compreso tra 1 e 12.
- **ID Design Goal**, un identificatore univoco e un nome esplicativo.
- **Descrizione**, una descrizione del Design Goal.
- **Categoria**, la categoria di appartenenza del Design Goal.
- **RNF di origine**, il requisito non funzionale che lo ha generato.

Design Goals

Rank	ID Design Goal	Descrizione	Categoria	RNF di origine
3	DG_1 Tempi di risposta della ricerca	Il sistema deve garantire un tempo di risposta non superiore a 3 secondi	Performance	RNF_P_1
5	DG_2 Gestione della concorrenza	Il sistema dovrà essere correttamente funzionante anche con un elevato numero (1000) di utenti connessi in contemporanea.	Performance	RNF_P_3
7	DG_3 Efficienza Raccomandazioni	Il sistema deve presentare suggerimenti personalizzati all'utente in un tempo massimo di 3 secondi	Performance	RNF_P_4
1	DG_4 Backup e Ripristino dati	Il sistema deve effettuare backup giornalieri di tutti i dati degli utenti. Deve permettere un ripristino completo in meno di 24 ore.	Dependability	RNF_A_2
6	DG_5 Atomicità	In caso di fallimento di un'operazione, il sistema deve inviare un messaggio di errore comprensibile.	Dependability	RNF_A_3
2	DG_6 Protezione dati personali	Il sistema deve proteggere tutti i dati personali degli utenti da accessi non autorizzati e gestiti in conformità con le normative sulla privacy.	Dependability	RNF_S_2

9	DG_7 Compatibilità cross-Browser	Il sistema deve essere pienamente compatibile con le ultime due versioni dei principali browser	Maintanence	RNF_C_2
8	DG_8 Estendibilità	Il sistema si presta facilmente all'aggiunte di nuove funzionalità	Maintanence	RNF_C_3
11	DG_9 Design responsive	L'interfaccia grafica del sistema deve adattarsi correttamente e garantire la piena funzionalità su schermi di diverse dimensioni.	End user	RNF_C_1
12	DG_10 Condivisione social	Il sistema deve permettere di poter condividere agli utenti le proprie recensioni o liste sui principali social.	End user	RNF_I_2
10	DG_11 Deployment Web	Il sistema deve supportare un processo di deployment automatizzato e affidabile che consente il rilascio di nuove versioni dell'applicazione web	Maintenance	RNF_C_4
4	DG_12 Prevenzione attacchi comuni	Il sistema deve garantire la protezione contro le vulnerabilità conformemente all'edizione più recente del progetto OWASP Top 10.	Dependability	RNF_S_3

Trade-off

Trade-off	Descrizione
Tempi di risposta vs sicurezza	Per garantire una sicurezza dell'applicazione si punta ad implementare sistemi che aumentino la stessa a discapito della velocità delle prestazioni, le quali potrebbero impiegare fino a 6 secondi.
Performance vs funzionalità	Per garantire feed e liste personalizzate occorre conteggiare il numero di like, commenti, interazioni di ogni utente, il che richiederebbe una maggiore spesa in termini di prestazioni data dall'elevato numero di query costose al/ai Database usato/i.

1.3 Definizioni, acronimi, e abbreviazioni

Vengono riportati di seguito alcune definizioni presenti nel documento corrente:

- **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 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

Di seguito una lista di riferimenti ed altri documenti utili durante la lettura:

- Statement of Work;
- Business Case;
- Requirements Analysis Document;
- System Design Document;
- Object Design Document;
- Test Plan;
- Matrice di tracciabilità;
- Manuale di installazione;
- Manuale utente;

1.5 Organizzazione del documento

Il presente documento di System Design è composto di quattro sezioni:

Introduzione: Viene descritto in generale lo scopo del sistema, gli obiettivi di design che il sistema propone di raggiungere.

Architettura software corrente: Viene descritto lo stato attuale dell'architettura del software già presente.

Architettura software proposta: Viene descritto come il sistema sarà definito e partizionato in sottosistemi, il loro mapping Hardware/Software, la gestione dei dati persistenti. Verranno poi presentate la struttura dei singoli sottosistemi e le boundary conditions riguardanti l'intero sistema.

Glossario: Contiene la lista dei termini utilizzati nel documento con annessa spiegazione.

2 Architettura del sistema corrente

Attualmente esistono due principali software che forniscono, seppur in maniere separate, i servizi che Fidelio offre.

Letterboxd è un social network co-fondato da Matthew Buchanan e Karl von Randow nel 2011. È stata lanciata come app social incentrata sulla condivisione di opinioni per i film.

Il sito è pensato per condividere i gusti dei suoi membri nei film. Gli utenti possono scrivere le loro opinioni sui film, tenere traccia dei film che hanno visto in passato o che sono intenzionati a vedere, fare elenchi di film e mostrare i loro preferiti, nonché incontrare e interagire con altri cinefili.

MovieLens invece è un sistema di raccomandazione basato sul web e una comunità virtuale che consiglia i film da guardare ai propri utenti, in base alle loro preferenze cinematografiche utilizzando il filtraggio collaborativo delle valutazioni dei film e delle recensioni dei film dei membri. Contiene circa 11 milioni di valutazioni per circa 8500 film. MovieLens è stato creato nel 1997 da GroupLens Research, un laboratorio di ricerca presso il Dipartimento di Informatica e Ingegneria dell'Università del Minnesota, al fine di raccogliere dati di ricerca su raccomandazioni personalizzate.

3 Architettura del sistema proposto

3.1 Panoramica della sezione

Il sistema proposto è basato sullo stile architetturale Three Tier, implementato utilizzando Spring MVC. Il motivo della presente scelta è che tale architettura è perfetta per lo sviluppo di web application come il nostro sistema, poiché la separazione della logica di presentazione da quella di elaborazione, migliora una serie di qualità, tra le quali:

- **Leggibilità**
- **Manutenzione**
- **Riuso**

Nello sviluppo del sistema verranno usati HTML5, CSS3 e JSP per la parte di front-end e la generazione della view.

Per la logica applicativa e quindi il back-end sarà utilizzato **Java Spring**.

Per la gestione del database saranno usati:

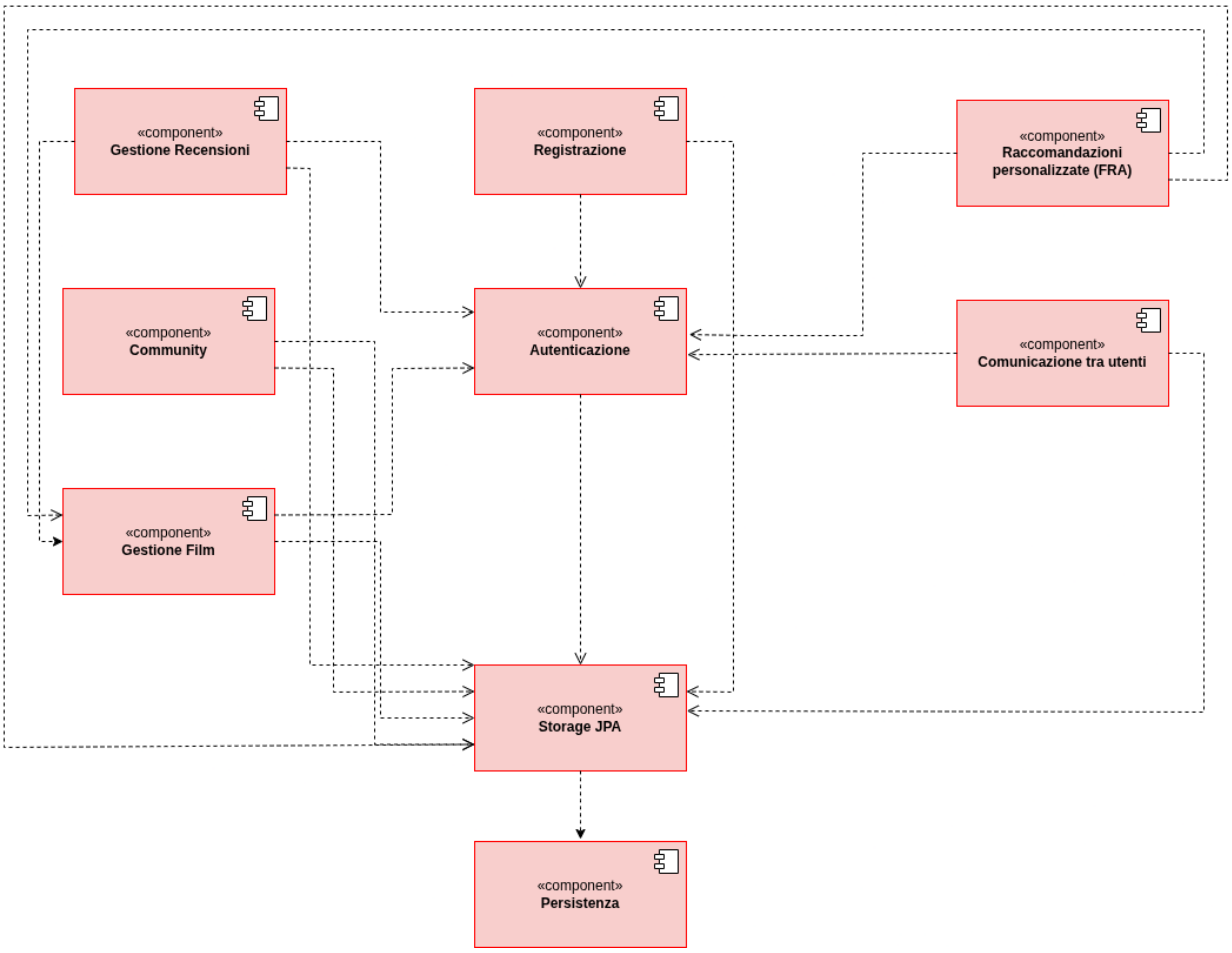
- **JPA** come standard per il mapping **ORM**
- **Docker e SQL** per il database.

3.2 Decomposizione in sottosistemi

I sottosistemi individuati sono:

- **Registrazione:** si occupa di gestire la registrazione dei vari tipi di utente: critico, fedele e cinefilo.
- **Autenticazione:** è responsabile delle funzionalità di Login, Logout, visualizzazione area utente e la modifica dati account.
- **Community:** responsabile della visualizzazione delle varie community a cui iscriversi, la partecipazione a queste community e la creazione da parte di un Fedele.
- **Gestione Recensioni:** si occupa dell'inserimento e della gestione delle recensioni di film, con i relativi servizi di segnalazione, avviso spoiler.
- **Gestione Film:** si occupa delle funzioni riguardanti la visualizzazione dei film e della possibilità di creare liste personalizzate.
- **Raccomandazioni personalizzate (FRA):** si occupa di generare liste raccomandate, utili per aiutare gli utenti nella scelta di un particolare film.
- **Comunicazione tra utenti**: si occupa della gestione di canali di comunicazione per mettere in contatto i vari utenti.
- **Persistenza**: si occupa di gestire la persistenza dei dati con un database.
- **Storage JPA**: si interpone tra i vari sottosistemi e il sottosistema di Persistenza.

Sono mostrate di seguito le dipendenze tra i sottosistemi attraverso un **component Diagram UML.**



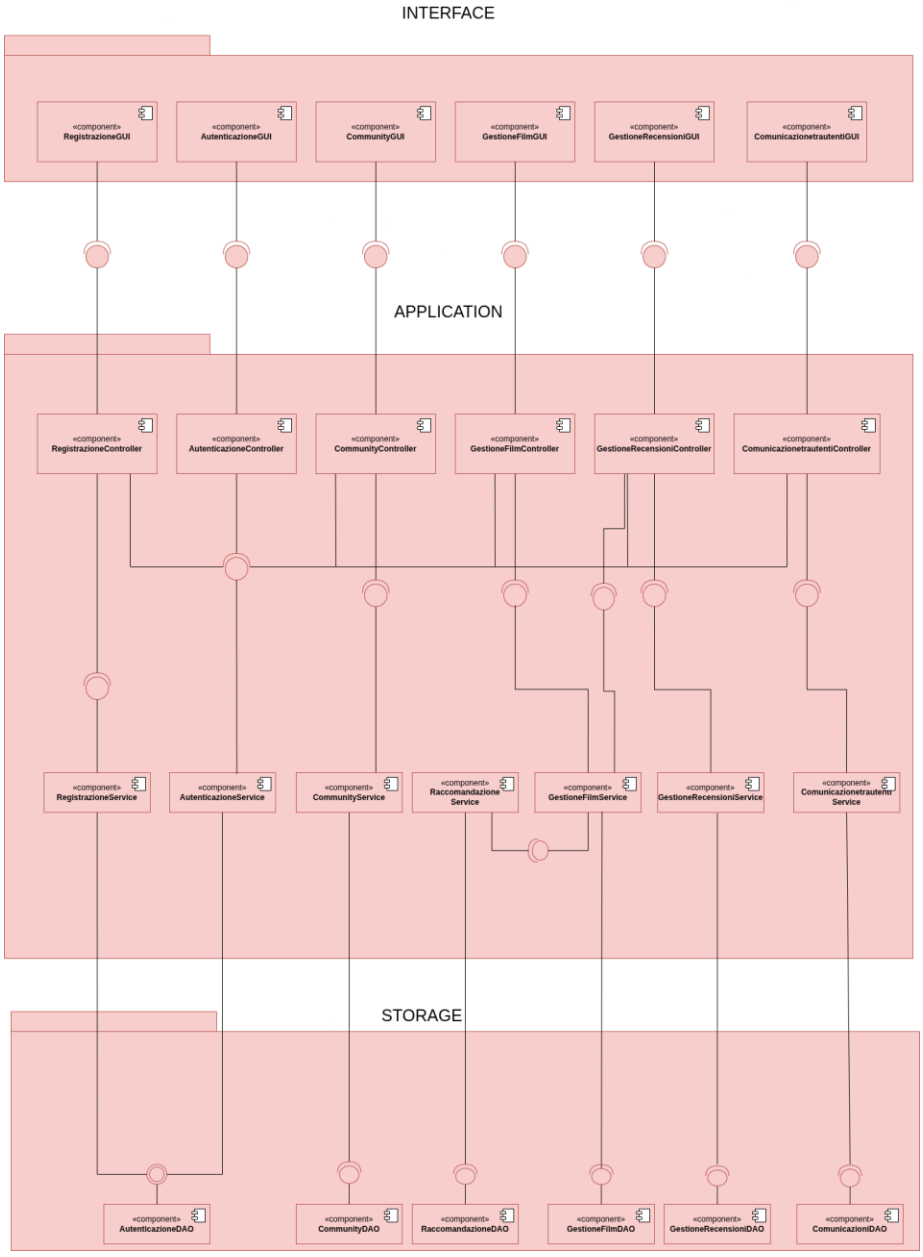
Alcuni sottosistemi saranno seguiti da componenti COTS (Commercial off the shelf), di seguito un elenco:

- Storage JPA verrà gestito tramite Spring Data JPA.
- Persistenza sarà gestita attraverso un DBMS relazionale su sistema cloud Docker.

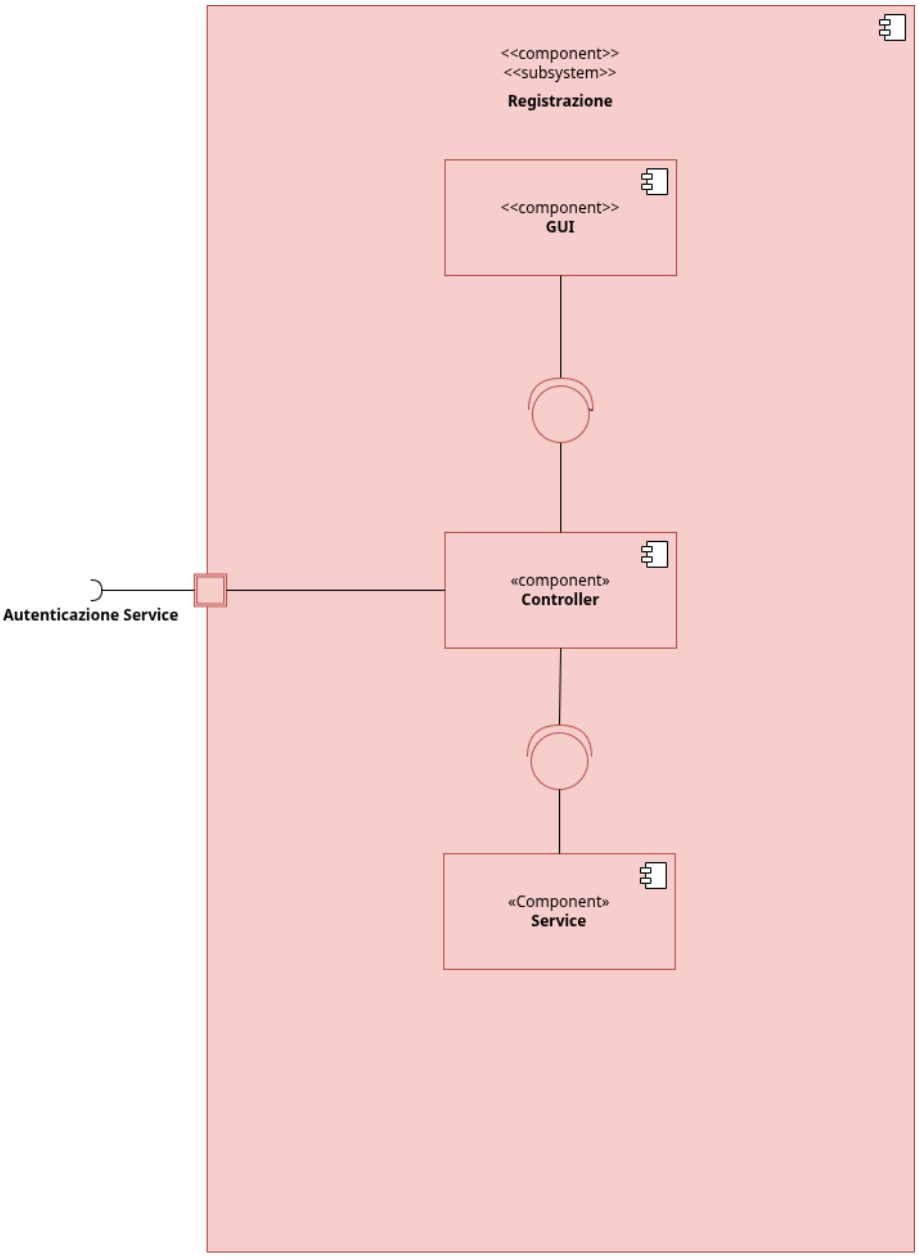
Di seguito una vista dettagliata di ciascun sottosistema evidenziando le componenti principale:

- **GUI:** Graphic User Interface, che contiene le varie view che saranno renderizzate 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.

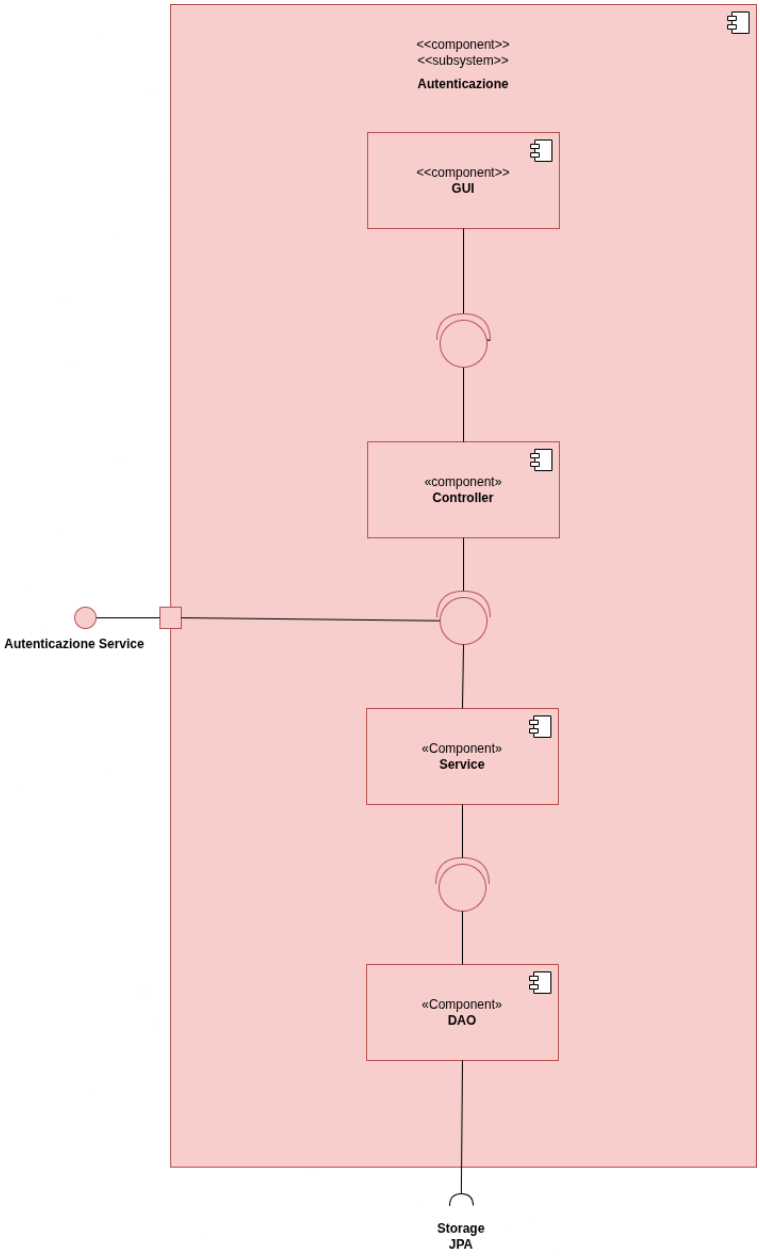
Diagramma Architetturale



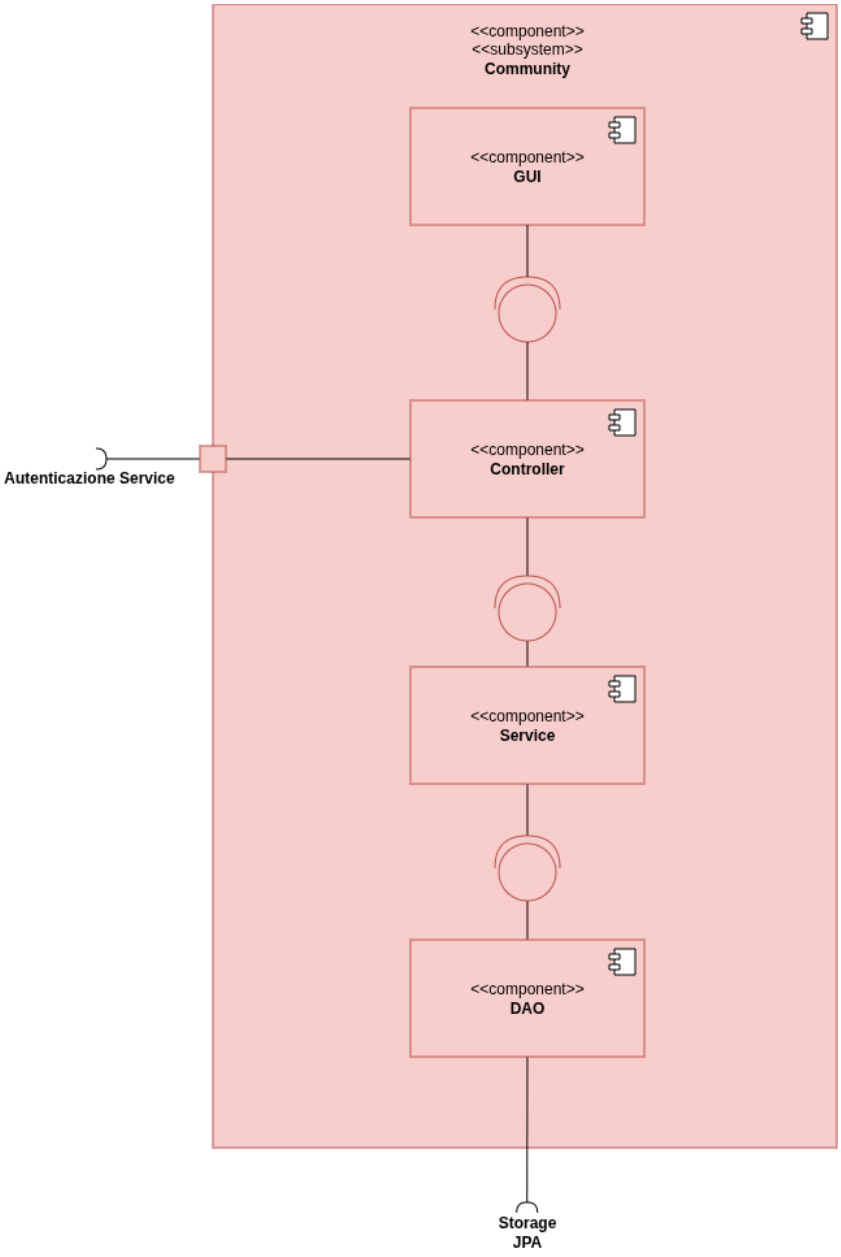
Sottosistema registrazione



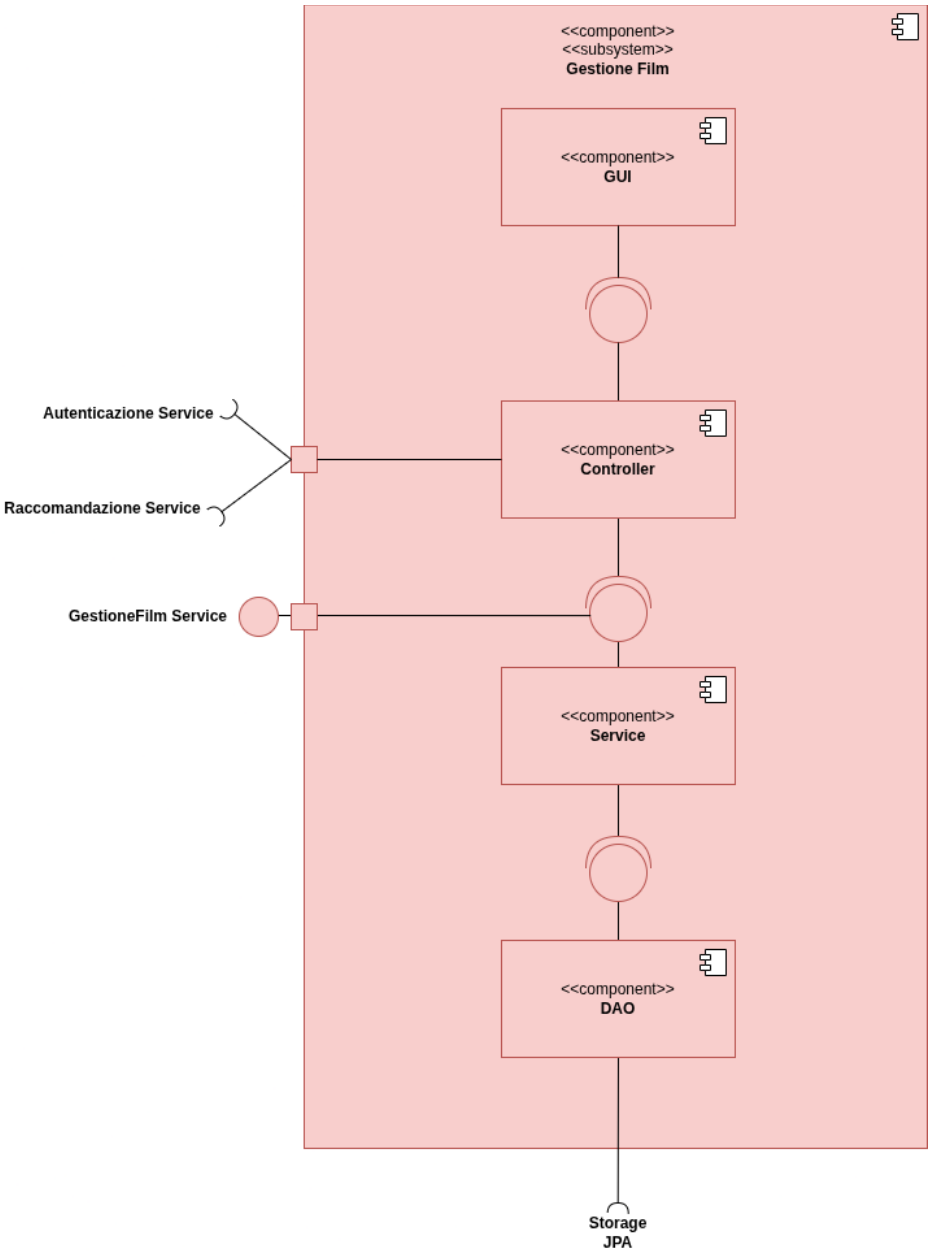
Sottosistema autenticazione



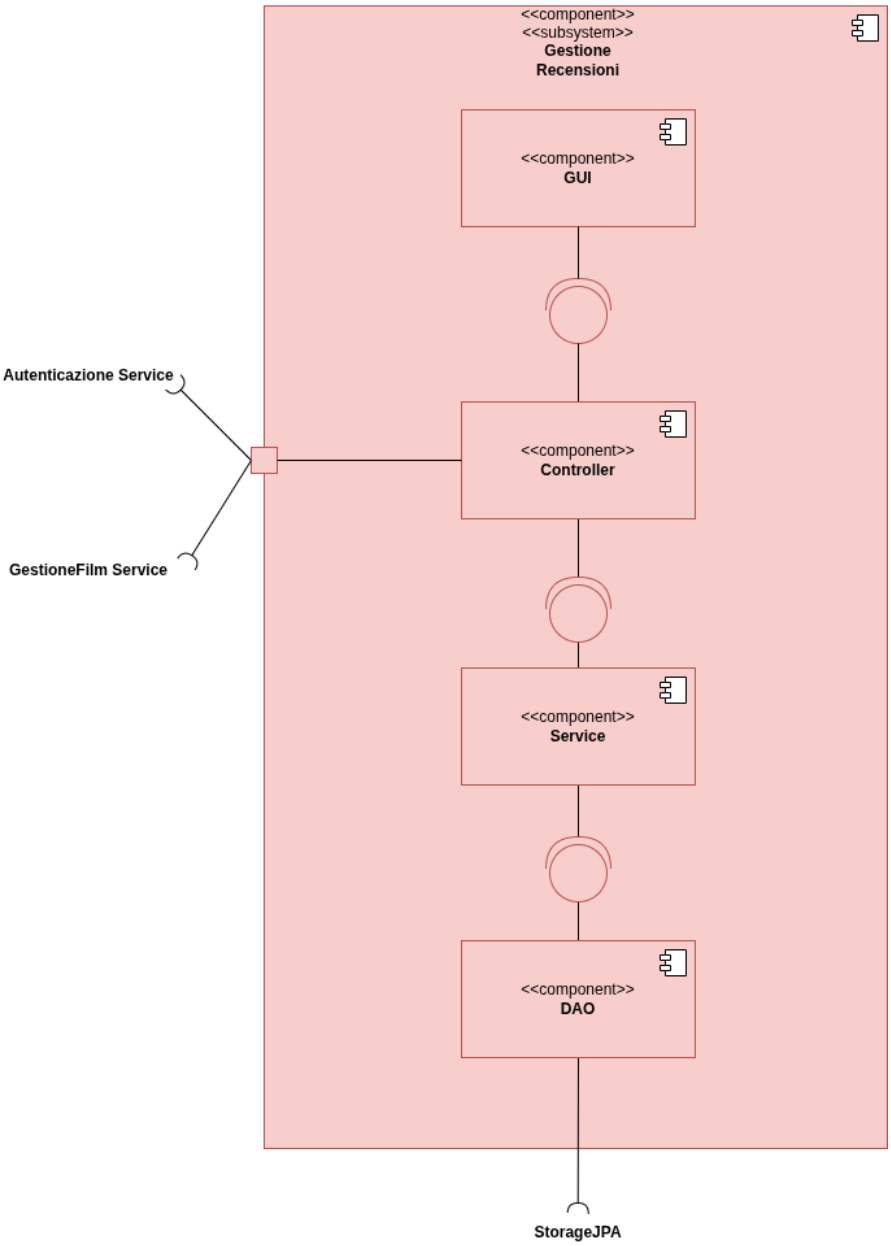
Sottosistema Community



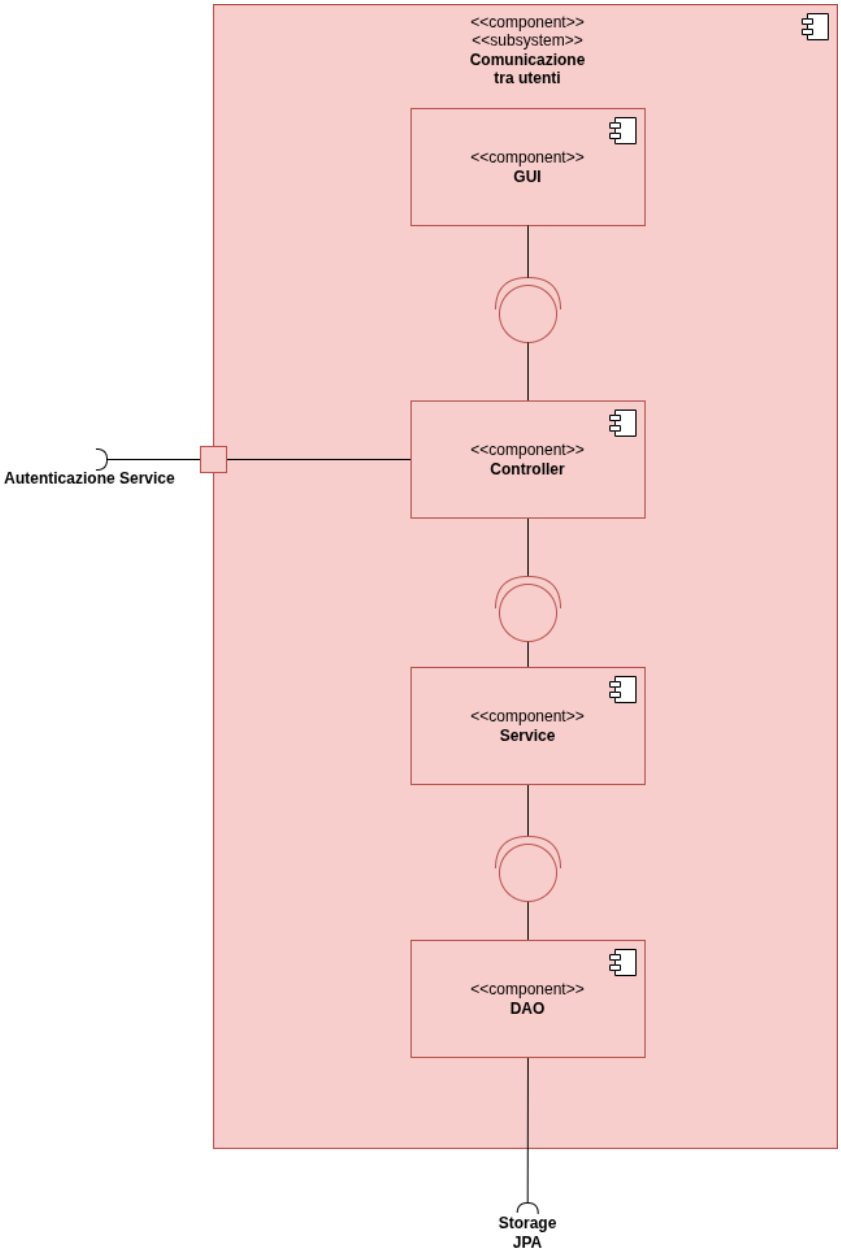
Sottosistema Gestione Film



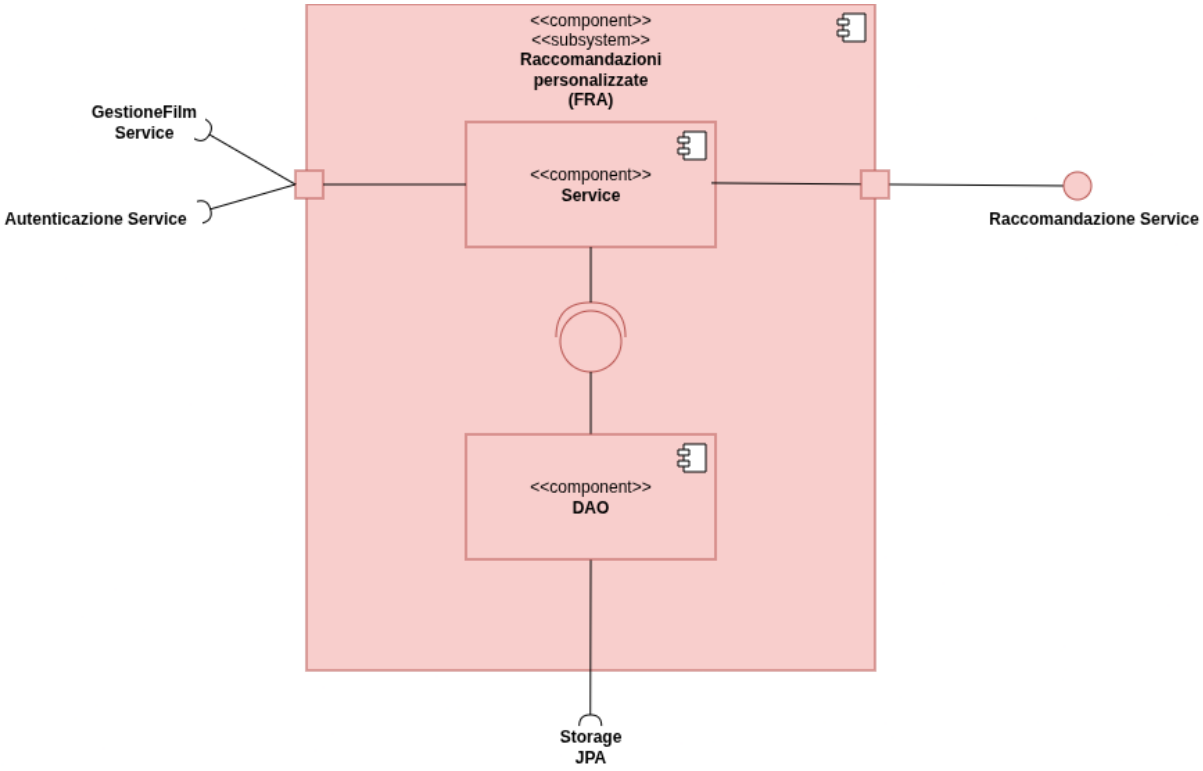
Sottosistema GestioneRecensioni



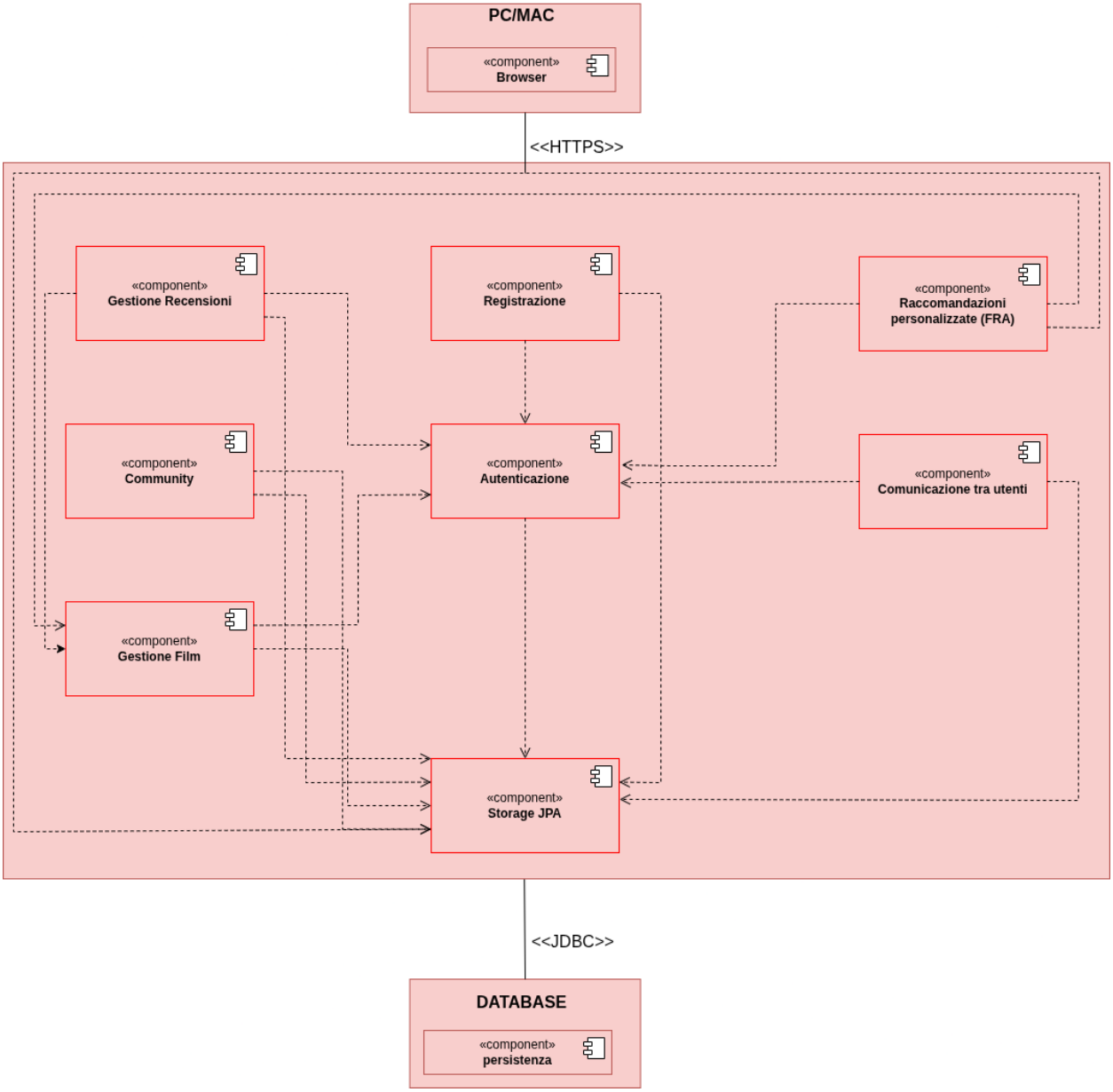
Sottosistema Comunicazione tra utenti



Sottosistema Raccomandazioni personalizzate (FRA)



3.3 Mapping hardware/software



3.4 Gestione dati persistenti

Per la gestione dei dati persistenti è stato scelto l'utilizzo di un database relazionale poiché in primis è la tecnologia con cui il team è più familiare, inoltre garantisce un maggiore coerenza con i design goal e offre diversi vantaggi significativi considerando la natura dei dati, tra cui:

1) **Organizzazione dei dati strutturata:** un RDBMS permette di organizzare dati in tabelle ben strutturate, con relazioni chiare tra entità.

2) **Facilità di Query:** Un RDBMS, grazie a SQL, consente di estrarre facilmente dati complessi attraverso query.

3) **Gestione delle transazioni complesse:** Gestione delle transazioni in modo affidabile, grazie alla proprietà ACID (Atomicità, Consistenza, Isolamento, Durabilità) dei database relazionali.

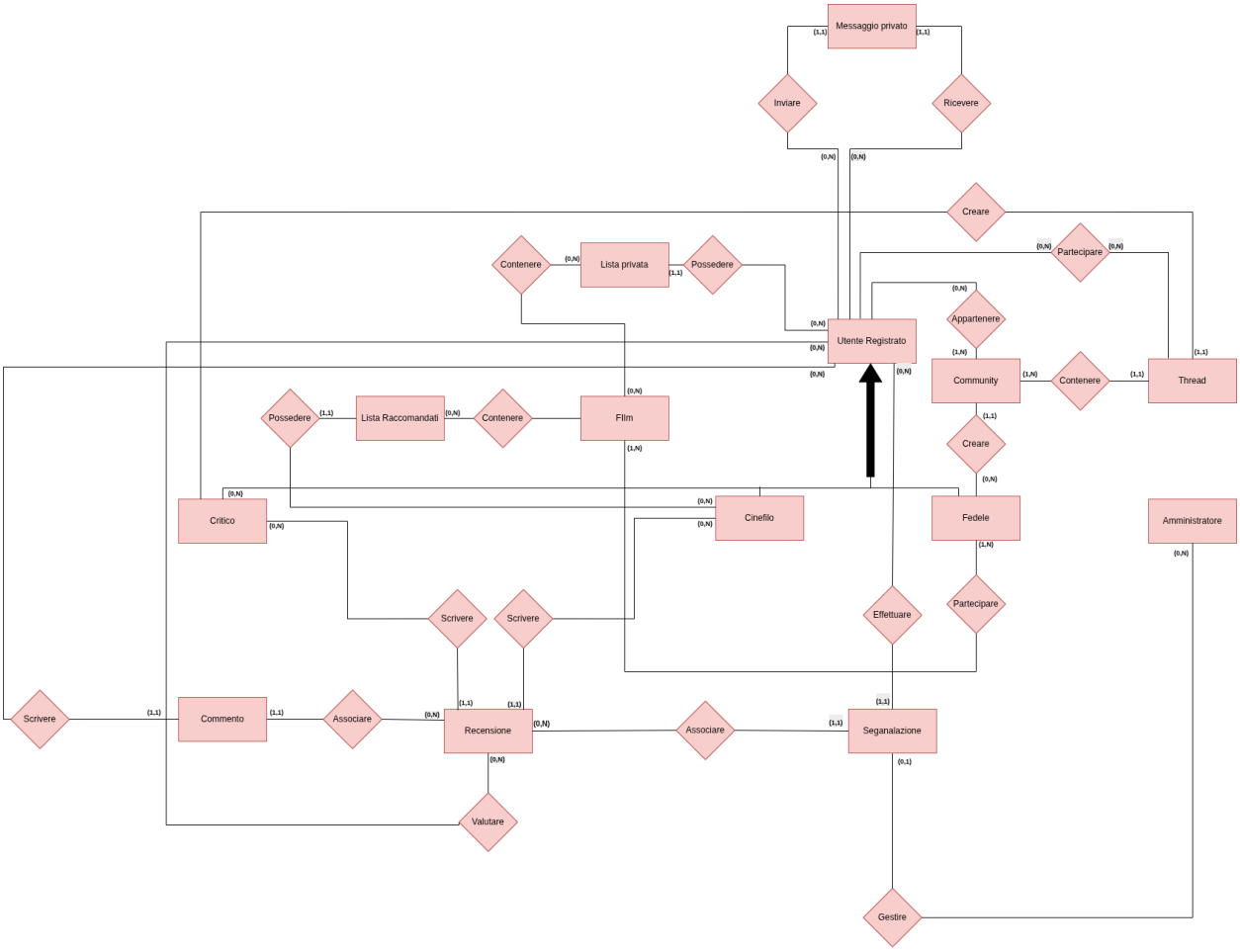
4) **Affidabilità dei dati:** L'organizzazione strutturata dei dati nei database relazionali assicura coerenza, integrità e accesso efficiente, riducendo errori e duplicazioni.

5) **Backup e ripristino efficaci:** Un RDBMS consente backup regolari e affidabili, necessari per evitare perdite di dati critici in caso di guasto o attacco Informatico.

Per garantire una collaborazione più efficiente nel trattare il database, i nostri dati persistenti sono gestiti attraverso un database in cloud Docker, avendo così a disposizione un'istanza di database comune a tutti i membri del team.

Inoltre, utilizzando una risorsa non presente localmente sui dispositivi, riduciamo i potenziali errori individuali derivati dall'hosting locale della base di dati, potenzialmente riducendo di conseguenza il tempo speso a risolvere tali errori.

ER: Schema ER del database



Dizionario delle Entità

Entità	Descrizione	Attributi	Identificatore
Utente Registrato	Entità padre che rappresenta un utente generico iscritto alla piattaforma. Generalizza Cinefilo, Critico, Fedele e Amministratore.	ID_Utente, Email, Password, Username, Data_Iscrizione, Avatar	ID_Utente
Cinefilo	Specializzazione di Utente Registrato. Rappresenta l'appassionato di cinema standard.	Livello_Attività, Generi_Preferiti	ID_Utente
Critico	Specializzazione di Utente Registrato. Rappresenta un utente professionale verificato o associato ad una rivista.	Rivista, ID_Tesserino, Biografia	ID_Utente
Fedele	Specializzazione di Utente Registrato. Rappresenta un utente verificato che lavora nel settore cinematografico.	Carriera, Credit_refernce, Casa_produzione	ID_Utente
Amministratore	Specializzazione di Utente Registrato. Utente con privilegi di	Livello_Permessi	ID_Utente

	gestione e moderazione.		
Messaggio privato	Messaggio scambiato direttamente tra due utenti.	ID_Messaggio, Testo, Data_Invio, Letto(bool)	ID_Messaggio
Lista privata	Elenco di film creato e gestito manualmente da un utente.	ID_Lista, Nome, Descrizione, Pubblica(bool)	ID_Lista
Lista Raccomandata	Elenco di film generato da FRA specificamente per un cinefilo.	ID_ListaRaccomandata, Data_Generazione	ID_ListaRaccomandata
Film	Opera cinematografica presente nel catalogo.	ID_Film, Titolo, Anno, Regista, Durata, Genere	ID_Film
Recensione	Opinione strutturata, composta da testo e voto, da un utente su un film.	ID_Recensione, Titolo, Testo, Voto, Spoiler (bool) Data_Pubblicazione	ID_Recensione
Commento	Breve risposta testuale associata ad una recensione.	ID_Commento, Testo, Data_Invio	ID_Commento
Segnalazione	Avviso inviato agli amministratori riguardo contenuti inappropriati.	ID_Segnalazione, Motivo, Stato_Gestione, Data	ID_Segnalazione
Community	Gruppo o spazio tematico a cui gli utenti possono iscriversi.	ID_Community, Nome, Descrizione, Regolamento	ID_Community
Thread	Discussione specifica aperta all'interno di una community.	ID_Thread, Titolo, Data_Apertura, Stato(Aperto/Chiuso)	ID_Thread

Dizionario delle Relazioni

Relazione	Descrizione	Entità Coinvolte	Attributi
Inviare	L'utente invia un messaggio privato.	**Utente Registrato** (0,N) **Messaggio privato** (1,1)	-
Ricevere	L'utente riceve un messaggio privato.	**Utente Registrato** (0,N) **Messaggio privato** (1,1)	-
Possedere (Lista P.)	L'utente possiede delle liste personali.	**Utente Registrato** (0,N) **Lista privata** (1,1)	-
Contenere (Lista P.)	Relazione molti-a-molti tra liste private e film.	**Lista privata** (0,N) **Film** (0,N)	Data_Aggiunta
Possedere (Lista R.)	Il Cinefilo possiede liste di raccomandazioni.	**Cinefilo** (0,N) **Lista Raccomandati** (1,1)	-
Contenere (Lista R.)	Relazione tra liste raccomandate e film.	**Lista Raccomandati** (0,N) **Film** (1,N)	Score_Match
Scrivere (Critico)	Il Critico scrive una recensione.	**Critico** (0,N) **Recensione** (1,1)	-
Scrivere (Cinefilo)	Il Cinefilo scrive una recensione.	**Cinefilo** (0,N) **Recensione** (1,1)	-
Valutare	Un Cinefilo valuta (es. like/dislike) una recensione altrui.	**Cinefilo** (0,N) **Recensione** (0,N)	Voto
Scrivere (Commento)	Un utente generico scrive un commento.	**Utente Registrato** (0,N) **Commento** (1,1)	-
Associare (Comm-Rec)	Un commento è associato ad una specifica recensione.	**Commento** (1,1) **Recensione** (0,N)	-
Associare (Rec-Segn)	Una segnalazione è associata ad una	**Recensione** (0,N) **Segnalazione** (1,1)	-

	recensione specifica.		
Effettuare	Un utente effettua una segnalazione.	**Utente Registrato** (0,N) **Segnalazione** (1,1)	-
Gestire	L'amministratore gestisce la segnalazione.	**Amministratore** (0,N) **Segnalazione** (0,1)	Esito_Gestione
Appartenere	Un utente generico è membro di una community.	**Utente Registrato** (0,N) **Community** (1,N)	Data_Iscrizione
Creare (Community)	Un utente "Fedele" crea una community.	**Fedele** (0,N) **Community** (1,1)	Data_Creazione
Partecipare (Community)	Un utente "Fedele" partecipa attivamente (o ha ruolo speciale) in una community.	**Fedele** (1,N **Community** (0,N)	Ruolo_Specifico
Contenere (Thread)	Una community contiene dei thread.	**Community** (1,N) **Thread** (1,1)	-
Creare (Thread)	Un utente crea un thread di discussione.	**Utente Registrato** (0,N) **Thread** (1,1)	-
Partecipare (Thread)	Un utente partecipa (risponde) ad un thread.	**Utente Registrato** (0,N) **Thread** (0,N)	Data_Ultimo_Accesso
Partecipare (Fedele – Film)	Un utente “Fedele” partecipa ad uno o più Film, cioè ha lavorato a quel film, (ad es. come regista, scenografo, sceneggiatore, attore, tecnico, ...)	**Fedele** (1,N) **Film** (1,N)	Impiego

3.5 Controllo degli accessi in sicurezza

Di seguito viene mostrata la matrice degli accessi per poter tenere traccia di quali attori possono accedere ai quali dei servizi offerti dal sistema:

Attori / Oggetti	Cinefilo	Critico
Registrazione		
Autenticazione	Login Logout VisualizzaAreaUtente ModificaDatiUtente CancellazioneAccount	Login Logout VisualizzaAreaUtente ModificaDatiUtente CancellazioneAccount
Gestione Film	AggiungiFilmaListaPrivata VisualizzaDettagliFilmModificaListaPrivata RicercaFilm VisualizzaListaRaccomandata RimuoviFilmListaRaccomandati	AggiungiFilmaListaPrivata VisualizzaDettagliFilmModificaListaPriv RicercaFilm
Community	VisualizzaCommunit IscrivitiCommunity DiscrivitiCommunity PubblicazioneThread	CreaThread CancellaThread VisualizzaThread VisualizzaCommunityVisualizzaDatiThr PubblicazioneThread
Gestione Recensioni	AggiungiRecensione ModificaRecensione EliminaRecensione VisualizzaRecensioneValutareRecensione SegnalareRecensioneAvvisoSpoiler	AggiungiRecensione ModificaRecensione EliminaRecensione VisualizzaRecensioneValutareRecensio SegnalareRecensioneAvvisoSpoiler

	CommentaRecensione	CommentaRecensione PubblicaCommentoTecnico
Raccomandazioni Personalizzate	VisualizzaLista AggiornaLista ValutazioneFilm	VisualizzaLista AggiornaLista ValutazioneFilm
Comunicazione fra utenti	IniziaConversazione ScriviMessaggio RiceviNotifica SegnalaChat	IniziaConversazione ScriviMessaggio RiceviNotifica SegnalaChat

Attori / Oggetti	Amministratore	Ospite
Registrazione		RegistrazioneCinefilo RegistrazioneCritico RegistrazioneFedele
Autenticazione	Login Logout VisualizzaAreaUtente ModificaDatiUtente CancellazioneAccount	
Gestione Film	AggiungiFilmaListaPrivata VisualizzaDettagliFilm ModificaListaPrivata RicercaFilm VisualizzaListaRaccomandata	VisualizzaDettagliFilm RicercaFilm
Community	VisualizzaCommunity IscrivitiCommunity DiscrivitiCommunity PubblicazioneThread	VisualizzaCommunity
Gestione Recensioni	AggiungiRecensione ModificaRecensione EliminaRecensione VisualizzaRecensione ValutareRecensione SegnalareRecensioneAvvisoSpoiler CommentaRecensione	VisualizzaRecensione
Raccomandazioni Personalizzate	VisualizzaLista AggiornaLista ValutazioneFilm	

Comunicazione fra utenti	IniziaConversazione ScriviMessaggio RiceviNotifica SegnalaChat	

3.6 Controllo globale del software

Il sistema Fidelio è un sistema interattivo in cui ogni funzionalità viene attivata attraverso comandi impartiti dall'utente mediante un'interfaccia grafica.

Quando un utente desidera accedere a una funzionalità del sistema, interagisce con la GUI che individua il controllo appropriato. Questa azione genera un evento che viene catturato e gestito dal relativo Handler.
L'Handler indirizza quindi il flusso di controllo al sottosistema dedicato alla logica di controllo, il quale si interfaccia con i servizi responsabili della logica applicativa.

Data la natura di web Application del sistema, il meccanismo di controllo del flusso adottato è di tipo Event-Driven, in cui ogni interazione dell'utente innesca una catena di eventi che attraversano i vari livelli architetturali del sistema.

3.7 Condizioni limite

Avvio del sistema

Identificativo	UCBC_1 – Avvio del Sistema	Data	19/11/2025
		Vers.	1.0
		Autore	LS
Descrizione	Lo UC permette l'avvio del sistema		
Attore principale	Amministratore		
Attori secondari	NA		
Entry Condition	L'Amministratore accede al Server		
Exit Condition On success	Il Sistema viene avviato correttamente		
Exit Condition On failure	Il Sistema non viene avviato		

FLUSSO DI EVENTI PRINCIPALE		
1	Amministratore	Esegue sulla macchina il comando che avvia il sistema.
2	Sistema	Verifica la sanità dei dati persistenti e, se sani, rende disponibili i suoi servizi e le sue funzionalità agli utenti
Scenario/Flusso di eventi alternativo: I Dati Persistenti sono danneggiati		
2.a1	Sistema	Notifica l'Amministratore di problemi ai dati persistenti e non effettua l'avvio
2.a2	Amministratore	Corregge i dati persistenti
2.a3	Sistema	Esegue il Passo 1

Spegnimento del sistema

Identificativo	UCBC_2 – Spegnimento del Sistema	Data	19/11/2025
		Vers.	1.0
		Autore	LS
Descrizione	Lo UC permette lo spegnimento del sistema		
Attore principale	Amministratore		
Attori secondari	NA		
Entry Condition	L'Amministratore accede al Server AND Il Sistema è stato precedentemente avviato AND Il Sistema è ancora acceso		
Exit Condition On success	Il Sistema viene spento correttamente		
Exit Condition On failure	Il Sistema non viene spento		

FLUSSO DI EVENTI PRINCIPALE		
1	Amministratore	Invia un segnale di spegnimento al Sistema.
2	Sistema	Verifica che non ci siano connessioni aperte da o verso l'esterno, in tal caso, termina l'esecuzione del sistema.
Scenario/Flusso di eventi alternativo: Ci sono connessioni ancora aperte.		
2.a1	Sistema	Notifica l'Amministratore la presenza di connessioni ancora aperte da o verso l'esterno.
2.a2	Amministratore	Attende una quantità di tempo per rispondere ad eventuali richieste dall'esterno non generando nuove connessioni se non per rispondere a richieste già in corso.
2.a3	Sistema	Verifica che non ci siano connessioni aperte da o verso l'esterno, in tal caso, termina l'esecuzione del sistema.

2.a4	Sistema	Notifica l'Amministratore dell'avvenuto spegnimento del sistema.
Scenario/Flusso di eventi alternativo: Ci sono connessioni ancora aperte.		
2.a3.a1	Sistema	Recide le connessioni verso l'esterno.
2.a3.a2	Sistema	Notifica l'Amministratore dell'avvenuto spegnimento del sistema e del numero di connessioni recise.

Fallimento del sistema

Identificativo	UCBC_3 – Fallimento del Sistema	Data	19/11/2025
		Vers.	1.0
		Autore	LS
Descrizione	Lo UC definisce il comportamento del Sistema in caso di fallimento.		
Attore principale	Amministratore		
Attori secondari	NA		
Entry Condition	Il Sistema viene terminato inaspettatamente		
Exit Condition On success	Il sistema viene riavviato correttamente		
Exit Condition On failure	Il sistema non viene riavviato correttamente		

FLUSSO DI EVENTI PRINCIPALE		
1	Amministratore	***Include UCBC_1***

Errore di Accesso ai Dati Persistenti

Identificativo	UCBC_4 – Errore di accesso ai Dati Persistenti	Data	19/11/2025
		Vers.	1.0
		Autore	LS
Descrizione	Lo UC descrive il comportamento del sistema nel caso in cui fosse impossibile accedere ai dati persistenti o questi risultassero danneggiati/corrotti.		
Attore principale	Amministratore		
Attori secondari	NA		
Entry Condition	Il sistema non può accedere ai dati persistenti OR I dati persistenti risultano corrotti		
Exit Condition On success	Il Sistema riprende il normale funzionamento		
Exit Condition On failure	Il Sistema riprende il normale funzionamento		

FLUSSO DI EVENTI PRINCIPALE		
1	Sistema	Notifica l'amministratore dell'impossibilità di accedere ai dati persistenti
2	Sistema	Cessa di processare eventuali richieste dall'esterno e risponde a tutte le richieste con un messaggio di errore
3	Amministratore	***Include UCBC_2***
4	Amministratore	Provvede a rendere accessibili i dati persistenti ristabilendone la sanità.
2.a3	Amministratore	***Include UCBC_1***

4 Servizi dei sottosistemi

In questa sezione vengono descritti i servizi di ogni sottosistema precedentemente elencati.

Sottosistema Registrazione

Servizio	Descrizione	Interfaccia
Registrazione Cinefilo	Questa funzionalità permette di registrarsi sulla piattaforma come cinefilo.	RegistrazioneService
Registrazione Critico	Questa funzionalità permette di registrarsi sulla piattaforma come critico.	RegistrazioneService
Registrazione Fedele	Questa funzionalità permette di registrarsi sulla piattaforma come fedele.	RegistrazioneService

Sottosistema Autenticazione

Servizio	Descrizione	Interfaccia
Login	Questa funzionalità permette di effettuare l'accesso al sistema tramite le proprie credenziali per sfruttare le funzionalità offerte.	AutenticazioneService
Logout	Questa funzionalità permette di disconnettersi dal sistema	AutenticazioneService
Visualizza area utente	Questa funzionalità permette di visualizzare i dati relativi alla propria area utente.	AutenticazioneService
Modifica dati utente	Questa funzionalità permette di modificare i dati relativi alla propria area utente.	AutenticazioneService
Cancellazione account	Questa funzionalità permette di cancellare il proprio account sulla piattaforma.	AutenticazioneService

Sottosistema Community

Servizio	Descrizione	Interfaccia
Visualizzare lista Community	Questa funzionalità permette di visualizzare la lista delle Community disponibili.	CommunityService
Creazione Community	Questa funzionalità permette ad un fedele creare una community.	CommunityService
Cancella Community	Questa funzionalità permette ad un fedele di cancellare la propria Community.	CommunityService
Modifica dati Community	Questa funzionalità permette di modificare i dati relativi alla propria Community.	CommunityService
Visualizza membri Community	Questa funzionalità permette di cancellare il proprio account sulla piattaforma.	CommunityService
Partecipa Community	Questa funzionalità permette ad un cinefilo o critico di partecipare ad una Community.	CommunityService
Abbandona Community	Questa funzionalità permette ad un membro di una Community di abbandonarla.	CommunityService
Creazione Thread	Questa funzionalità permette ad un critico di creare un Thread di discussione.	CommunityService
Cancellazione Thread	Questa funzionalità permette ad un critico di eliminare un proprio Thread di discussione.	CommunityService

Sottosistema Gestione Recensioni

Servizio	Descrizione	Interfaccia
Scrivi recensione	Questa funzionalità permette ad un critico o cinefilo di scrivere una recensione riguardo un film.	GestioneRecensioniService
Modifica recensione	Questa funzionalità permette ad un critico o cinefilo di modificare una recensione scritta.	GestioneRecensioniService
Rimuovi recensione	Questa funzionalità permette ad un cinefilo o critico di rimuovere una propria recensione.	GestioneRecensioniService
Commento recensione	Questa funzionalità permette ad un cinefilo o critico di commentare una recensione.	GestioneRecensioniService
Valutazione recensione	Questa funzionalità permette ad un cinefilo o critico di valutare una recensione.	GestioneRecensioniService

Sottosistema Gestione Film

Servizio	Descrizione	Interfaccia
Aggiungi Film a lista privata	Questa funzionalità permette di aggiungere un film alla propria lista privata.	GestioneFilmService
Consulta catalogo film	Questa funzionalità permette di consultare la lista di film disponibili.	GestioneFilmService
Ricerca film	Questa funzionalità permette di effettuare la ricerca di un film.	GestioneFilmService
Rimozione film lista privata	Questa funzionalità permette la rimozione di un film dalla propria lista provata.	GestioneFilmService
Rimozione film lista raccomandati	Questa funzionalità permette la rimozione di un film dalla propria lista di raccomandati.	GestioneFilmService
Visualizzazione dettagli film	Questa funzionalità permette la visualizzazione dei dettagli di un film.	GestioneFilmService

Sottosistema Comunicazione tra utenti

Servizio	Descrizione	Interfaccia
Avvia comunicazione	Questa funzionalità permette di iniziare una comunicazione con un utente.	ComunicazioneTraUtentiService
Elimina comunicazione	Questa funzionalità permette di eliminare ogni traccia della comunicazione con un utente.	ComunicazioneTraUtentiService

Sottosistema Raccomandazioni personalizzate (FRA)

Servizio	Descrizione	Interfaccia
Genera raccomandazioni	Questa funzionalità permette di generare una lista di film raccomandati per un utente.	RegistrazioneService
Aggiorna preferenze	Questa funzionalità permette di aggiornare la lista raccomandati in base alle preferenze dell'utente.	RegistrazioneService

5 Cenni ODD

5.1 Design Pattern

Nella presente sezione si andranno a descrivere e dettagliare i design patterns utilizzati nello sviluppo del sistema.

Facade Design Pattern

Nel contesto del nostro progetto di Ingegneria del Software. Basato sull'architettura Three-Tier, abbiamo identificato la necessità di adottare il design pattern Facade al fine di fornire un'interfaccia unificata e semplificata ai sottosistemi applicativi. Ogni volta che l'Interface Layer necessita di eseguire operazioni complesse, esso invoca un singolo metodo del Facade, il quale coordina internamente le chiamate ai servizi necessari.

Si rivela una soluzione vantaggiosa per diversi motivi:

- **Semplicità d'uso:** I Controller interagiscono con un'interfaccia semplificata, riducendo la complessità del codice di presentazione e facilitando l'integrazione delle funzionalità.
- **Riduzione dell'accoppiamento:** L'Interface Layer e i sottosistemi di Business restano separati; modifiche interne ai sottosistemi non impattano direttamente l'interfaccia utente.
- **Sicurezza e Controllo centralizzato:** Le validazioni di sicurezza e le logiche trasversali (logging) vengono gestite in un unico punto, garantendo che ogni operazione rispetti i requisiti di Dependability prima di accedere ai dati sensibili.

DAO - Data Access Object

Nel contesto del nostro progetto, utilizziamo il design pattern **DAO (Data Access Object)** per gestire in modo strutturato e centralizzato l'accesso e le operazioni sul database relative alle nostre classi, che rappresentano entità persistenti mappate sulle tabelle del database.

Le classi sono annotate con ***@Entity*** e ***@Table,*** consentendo a **JPA** (Java Persistence API) di mappare automaticamente i suoi attributi sui campi della tabella. Questo approccio garantisce una stretta corrispondenza tra il modello Java e il livello di persistenza.

Vantaggi dell'uso dei DAO:

- **Astrazione e incapsulamento**: Le operazioni di accesso ai dati, come lettura, scrittura, aggiornamento ed eliminazione, sono centralizzate in una classe DAO dedicata, separando la logica del database dal resto dell'applicazione.
- **Manutenibilità**: Le modifiche alla struttura del database possono essere facilmente integrate aggiornando solo la classe DAO senza impatti significativi sul resto del codice.
- **Riutilizzabilità**: I metodi DAO possono essere riutilizzati in più contesti dell'applicazione, riducendo la duplicazione del codice.
- **Compatibilità con JPA**: L'uso di JPA semplifica l'interazione con il database, fornendo metodi standardizzati per eseguire operazioni CRUD e query personalizzate.

Proxy Design Pattern

Nel contesto del nostro progetto di Ingegneria del Software per la piattaforma Fidelio, abbiamo identificato la necessità di adottare il **Proxy Design Pattern** per gestire l'accesso controllato a risorse esterne e operazioni critiche del sistema. Il pattern Proxy agisce come intermediario tra il client e l'oggetto reale, fornendo un livello di controllo aggiuntivo prima di delegare le richieste all'implementazione effettiva.

Nel sistema Fidelio, il Proxy viene applicato principalmente in questo contesto strategico:

Remote Proxy per l'integrazione con TMDB API

L'interazione con servizi esterni come The Movie Database (TMDB) avviene attraverso un Proxy che incapsula la complessità delle chiamate HTTP, la gestione della cache e il rate limiting. Il **TmdbService** agisce come Remote Proxy, nascondendo al resto dell'applicazione i dettagli implementativi dell'API esterna e fornendo metodi semplificati per la ricerca film, il recupero dei dettagli e l'ottenimento delle raccomandazioni. Questo approccio garantisce che la chiave API rimanga protetta nel backend e che le risposte vengano automaticamente salvate nella cache per ridurre il numero di chiamate verso il servizio esterno.

Vantaggi dell'adozione del Proxy Pattern:

- **Sicurezza e Controllo degli accessi**: Il Proxy verifica le credenziali e i permessi prima di consentire l'accesso alle risorse protette, impedendo accessi non autorizzati e garantendo la conformità con i requisiti di sicurezza (RNF_S).
- **Ottimizzazione delle prestazioni**: Attraverso meccanismi di caching integrati nel Proxy, le risposte delle API esterne vengono memorizzate temporaneamente, riducendo la latenza e migliorando i tempi di risposta del sistema (RNF_P).
- **Isolamento dalle dipendenze esterne**: Il Proxy nasconde la complessità dell'integrazione con servizi di terze parti, permettendo al sistema di adattarsi facilmente a cambiamenti nelle API esterne o alla sostituzione dei provider senza impattare il resto dell'architettura.
- **Logging e Monitoring centralizzato**: Tutte le richieste passano attraverso il Proxy, che può registrare metriche di utilizzo, gestire errori e implementare strategie di retry automatico in caso di fallimento temporaneo delle risorse esterne.

6. Glossario

Termine	Definizione
Docker	Piattaforma di containerizzazione che consente alle applicazioni, incluso il software di database, di essere confezionate ed eseguite in ambienti isolati chiamati container
JPA	Jakarta Persistence o Java Persistance API, framework per il linguaggio di programmazione Java che si occupa della gestione della persistenza dei dati di un DBMS relazionale nelle applicazioni che usano le piattaforme Java Platform, Standard Edition e Jakarta EE.
TMDB/IMDb	The Movie Database e Internet Movie Database, servizi che mettono a disposizione una collezione di tutti i film pubblicati, in costante aggiornamento, forniscono API per potervi interagire.
ORM	Tecnica di programmazione che favorisce l'integrazione di sistemi software aderenti al paradigma della programmazione orientata agli oggetti con sistemi RDBMS
Handler	Modulo software che gestisce e reindirizza eventi all’appropriato sottosistema, secondo le logiche e le indicizzazioni di cui ha a disposizione, sviluppate sulla base delle responsabilità.