

De Martino Daniele
De Rosa Federico
Esposito Christian
Durante Francesco
Torre Orazio



HITCHPICKS

System Design Document



Hitchpicks: System Design Document

Sommario

Revision History.....	3
Team Members.....	4
1 Introduzione.....	5
1.1 Obiettivo del Sistema.....	5
1.2 Obiettivi di Design.....	5
1.2.1 Trade-offs.....	7
1.3 Acronimi e abbreviazioni.....	8
1.4 Riferimenti.....	8
2 Architettura software corrente.....	8
3 Architettura software proposta.....	9
3.1 Panoramica.....	9
3.2 Decomposizione in sottosistemi.....	10
3.3 Mapping hardware/software.....	13
3.4 Gestione dei dati persistenti.....	14
3.5 Controllo degli accessi e sicurezza.....	16
3.6 Controllo globale del software.....	17
3.7 Condizioni limite.....	19
4 Servizi dei sottosistemi.....	23
5 Cenni di object design.....	25
6 Glossario.....	28



Hitchpicks: System Design Document

Revision History

Data	Versione	Descrizione	Autori
12/11/2024	0.1	Prima stesura	Team
12/11/2024	0.2	Introduzione e design goals	Team
13/11/2024	0.3	Architettura corrente	DDM
15/11/2024	0.4	Panoramica archit. corrente e gestione dati persistente	DDM, OT
15/11/2024	0.5	Component diagram	CE, FD
15/11/2024	0.6	Panoramica decomposizione in sottosistemi	CE, FD
15/11/2024	0.7	Diagramma architetturale	CE, FD, DDM
15/11/2024	0.8	Diagramma classi ristr.	OT
15/11/2024	0.9	Diagramma E/R	FDR
16/11/2024	0.10	Mapping hardware/software e deployment diagram	DDM
18/11/2024	0.11	Condizioni limite	DDM, FDR, OT
18/11/2024	0.12	Controllo degli accessi	FDR
18/11/2024	0.13	Revisione design goals	DDM, OT
25/11/2024	0.14	Servizi dei sottosistemi	CE, FD
26/11/2024	0.15	Glossario	DDM
04/12/2024	0.16	Revisione class diagram ristrutturato e servizi dei sottosistemi dopo RAD v0.17	Team
21/12/2024	0.17	Cenni di object design (design patterns)	Team
06/01/2025	0.18	Revisione diagramma architetturale	DDM
07/01/2025	1.0	Revisione	Team



Hitchpicks: System Design Document

Team Members

Nome	Ruolo	Matricola	Acronimo
Daniele De Martino	Membro e coordinatore	0512116151	DDM
Federico De Rosa	Membro del team	0512117279	FDR
Christian Esposito	Membro del team	0512116472	CE
Francesco Durante	Membro del team	0512117846	FD
Orazio Torre	Membro del team	0512117426	OT



Hitchpicks: System Design Document

1 Introduzione

1.1 Obiettivo del Sistema

Hitchpicks è una **piattaforma web** progettata per assistere gli appassionati nel tenere traccia dei **film** e delle **serie TV** che hanno già visto o desiderano guardare in futuro. La piattaforma presenta un **catalogo** dei contenuti, e permette agli utenti di classificare i contenuti attraverso un sistema di **liste personalizzabili**, oltre che di **recensirli** pubblicamente.

1.2 Obiettivi di Design

I **design goals** sono indicatori della qualità del sistema e derivano direttamente dai requisiti non funzionali descritti nel RAD.

Sono suddivisi nelle categorie di **Performance**, **Dependability**, **Cost**, **Maintenance** ed **End user** seguendo quanto suggerito da *Object Oriented Software Engineering* by Bernd Brugge & Allen H. Dutoit Third Edition (2009).

ID	Nome	Descrizione	Categoria	RNF di origine	Rank
DG_01	Tempo di risposta	Il sistema deve garantire una risposta entro 2 secondi da una richiesta inviata dall'utente.	Performance	RNF8_P2	4
DG_02	Memoria	Il sistema deve essere in grado di gestire un carico di dati elevato, almeno pari a 10 GB di archiviazione.	Performance	RNF10_P4	7
DG_03	Throughput	Il sistema deve garantire un corretto funzionamento e prestazioni fino a 5000 utenti in contemporanea.	Performance	RNF9_P3	5
DG_04	Robustezza	Il sistema deve gestire almeno il 90% delle volte gli input invalidi degli utenti, comunicando l'errore. Il resto delle volte, il sistema non deve comunque andare in crash.	Dependability	RNF6_A4	3



Hitchpicks: System Design Document

Hitchpicks: System Design Document					
Requirement ID	Requirement Description	Acceptance Criteria	Priority	Dependencies	Impact Score
DG_05	Disponibilità	Il sistema deve garantire una disponibilità del 95% del tempo su base mensile. Inoltre, in caso di un riavvio imprevisto del sistema, il sistema deve essere in grado di recuperare lo stato corrente entro 5 minuti.	Dependability	RNF7_P1 RNF4_A2	6
DG_06	Sicurezza	Il sistema deve garantire la sicurezza dei dati conservati utilizzando l'algoritmo di hashing SHA-256 per la gestione dei dati sensibili (come le password) e rispettando le normative del Regolamento UE 2016/679 (GDPR).	Dependability	RNF3_A1 RNF18_L1	1
DG_07	Gestione permessi	Il sistema deve implementare un'appropriata separazione dei permessi e controllo degli accessi tramite ruoli, garantendo l'esclusività necessaria delle operazioni.	Dependability	RNF5_A3	2
DG_08	Estensibilità	Il sistema deve essere modulare abbastanza da poter essere facilmente esteso con nuove funzionalità, tra cui nuovi dati da associare ai contenuti.	Maintenance	RNF11_S1	9
DG_09	Leggibilità	Il codice sorgente deve essere leggibile e aderire almeno al 90% allo standard Checkstyle di Google, oltre che essere ben documentato tramite Javadocs.	Maintenance	RNF11_S1	11
DG_10	Distribuzione	Il sistema deve essere sviluppato e fruibile come piattaforma web-based, compatibile con i principali browser (Chrome, Firefox, Safari, Edge) e deve supportare sia desktop che dispositivi mobili.	Maintenance	RNF12_S2 RNF13_IM1	8

Hitchpicks: System Design Document

DG_11	Usabilità	Il sistema deve essere intuitivo e adatto anche a utenti meno esperti, senza utilizzare termini tecnici. Deve presentare un'interfaccia responsive e supportare schermi di qualsiasi tipo di dimensione, da una risoluzione minima di larghezza 320px e massima di larghezza 2560px.	End user	RNF1_U1 RNF2_U2	10
-------	-----------	--	----------	--------------------	----

1.2.1 Trade-offs

Alcuni trade-off considerati che potrebbero presentarsi sono:

- **Tempi di risposta vs. sicurezza:** l'utilizzo di algoritmi di hashing avanzati può impattare negativamente sulle prestazioni del sistema ed aumentare i tempi di risposta;
- **Tempi di consegna vs. funzionalità:** se lo sviluppo del sistema è in ritardo, può essere opportuno ignorare funzionalità di priorità minore (bassa, o anche media nei casi peggiori) per rilasciare il sistema quanto prima, e poi svilupparle in seguito;
- **Tempi di consegna vs. qualità:** se il testing del sistema è in ritardo, può essere opportuno ignorare bug noti di rilevanza minore per rilasciare il sistema quanto prima, e poi risolverli in seguito;
- **Tempi di consegna vs. usabilità (e sviluppo multi-browser):** se lo sviluppo del sistema è in ritardo, può essere opportuno ridurre i vincoli sull'usabilità, evitando magari di implementare la responsiveness e il supporto con tutti i browser, da rilasciare poi in secondo momento.

Hitchpicks: System Design Document

1.3 Acronimi e abbreviazioni

Gli acronimi e le abbreviazioni utilizzati nel documento sono i seguenti:

- **SOW**: Statement of Work;
- **RAD**: Requirement Analysis Document;
- **SDD**: System Design Document;
- **DG**: Design goal;
- **E/R**: Entity/Relationship (vedasi [glossario](#));
- **ID**: identificativo;
- **IMDB**: Internet Movie Database (piattaforma con dati di film e serie TV);
- **REST**: Representational State Transfer (stile architetturale per API HTTP-based);
- **GDPR**: General Data Protection Regulation (regolamento (UE) n. 2016/679).

1.4 Riferimenti

La stesura del documento ha comportato diversi riferimenti ad altri testi di cui si consiglia la lettura, tra i quali:

- [Hitchpicks: Statement of Work](#)
- [Hitchpicks: Requirement Analysis Document](#)
- *Object Oriented Software Engineering* by Bernd Brugge & Allen H. Dutoit
Third Edition (2009)
- [FURPS+](#)
- [GDPR](#)

2 Architettura software corrente

Il sistema correntemente attivo più simile ad Hitchpicks è **Letterboxd**. Purtroppo, non c'è mai stato un approfondimento pubblico o documentazione relativa l'architettura software della piattaforma, quindi non è possibile descriverla.

In ogni caso, avendo effettuato delle indagini a partire dagli API resi disponibili da Letterboxd, è possibile assumere che utilizzi un'**architettura distribuita** su più livelli e che utilizzi un servizio di **hosting in cloud**.

Inoltre, utilizza l'API di **TMDb** (servizio analogo a quello utilizzato da Hitchpicks, IMDb) per poter recuperare dati relativi ai film.

Hitchpicks: System Design Document

3 Architettura software proposta

3.1 Panoramica

Il sistema proposto utilizza il pattern architettonico **three-tier**, suddividendo quindi i sottosistemi in tre strati:

- **Interface layer**, che si occupa dell'interazione ed interfaccia con l'utente (contiene il cosiddetto **front-end**);
- **Application logic layer**, che si occupa del flusso e del controllo della logica di business applicativa;
- **Storage layer**, che si occupa della memorizzazione e recupero degli oggetti persistenti (assieme all'application logic layer, forma il **back-end**).

Utilizza, tra le varie tecnologie:

- **Java**, linguaggio OO per lo sviluppo del back-end;
- **Spring**, framework per lo sviluppo della web app, con l'uso di **Hibernate** e **Spring JPA** per accedere al DBMS;
- **PostgreSQL**, DBMS relazionale;
- **HTML, CSS e JavaScript**;
- **TailwindCSS** e **PralineUI**, librerie di styling e componenti d'interfaccia;
- **HeroIcons**, libreria di icone open-source;
- **Checkstyle**, per il linting in Java;
- **Javadoc**, per la documentazione in Java.

E ovviamente l'API esterno di **IMDb**, il quale necessita di un'API key (gratuita entro il limite di 1000 richieste giornaliere, di gran lunga non superate dall'uso regolare del sistema).

Una scelta necessaria è quella che riguarda la **gestione delle immagini**: poiché non è stato specificato alcun vincolo, si preferisce realizzarlo tramite un semplice valore testuale in cui è possibile inserire un URL ad un'immagine già caricata su qualche macchina, e non di permettere l'upload diretto di un file immagine. Questo è avvalorato anche dal fatto che IMDb fornisce propri URL relativi alle immagini delle locandine dei contenuti, considerato che nell'utilizzo pratico del sistema i contenuti saranno automaticamente compilati dall'integrazione IMDb.

Hitchpicks: System Design Document

3.2 Decomposizione in sottosistemi

Il sistema è suddiviso nei seguenti sottosistemi:

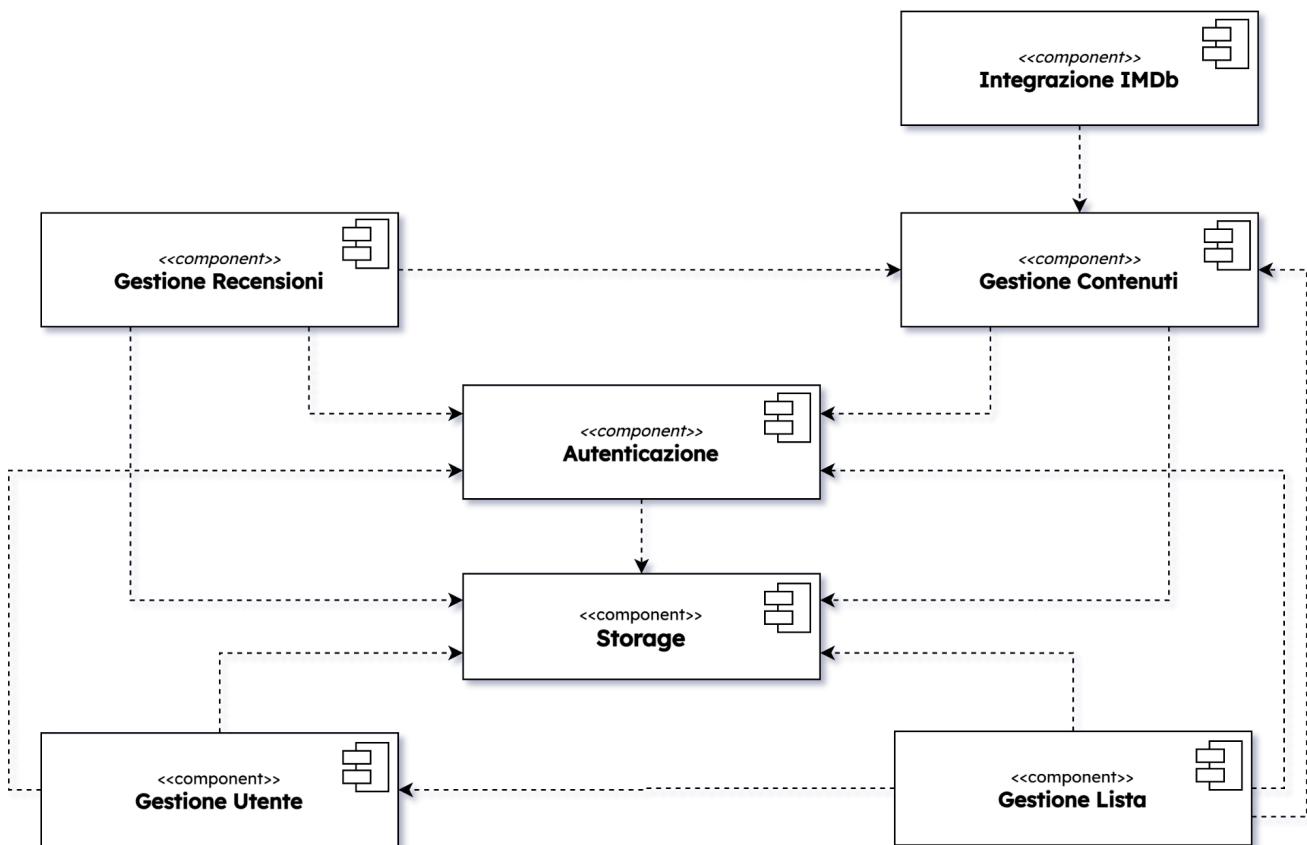
- **Storage:** opera da intermediario tra i vari sottosistemi e la base di dati, permettendo l'accesso e la manipolazione dei dati in un unico punto di accesso per tutti i componenti;
- **Autenticazione:** permette i servizi di login, logout, registrazione e verifica le credenziali dell'utente. È responsabile anche della gestione della sessione utente e tutti i componenti dipendono da esso poiché necessitano di discernere i permessi dell'utente per effettuare le operazioni negli altri sottosistemi;
- **Gestione Utente:** si occupa di tutte le operazioni legate agli utenti, tra cui la visualizzazione e gestione del profilo, la ricerca di altri utenti, e la rimozione di account da parte degli amministratori;
- **Gestione Lista:** permette agli utenti di creare, modificare e cancellare liste di contenuti personalizzate. Consente inoltre di visualizzare liste, applicando ricerca e filtri in esse e di visualizzare statistiche per ciascuna lista;
- **Gestione Contenuti:** si occupa di tutte le operazioni sui contenuti della piattaforma, inclusa la visualizzazione, aggiunta, modifica, rimozione e ricerca filtrata di contenuti;
- **Gestione Recensioni:** gestisce la visualizzazione, la scrittura e la rimozione delle recensioni degli utenti, oltre che l'approvazione o disapprovazione da parte dei gestori;
- **Integrazione IMDB:** è responsabile del collegamento con la piattaforma IMDb, permettendo di importare automaticamente informazioni dettagliate sui contenuti evitando l'inserimento manuale.

Hitchpicks: System Design Document

Diagramma dei componenti

Di seguito, è riportato il diagramma dei componenti rappresentante i sottosistemi, completo di dipendenze.

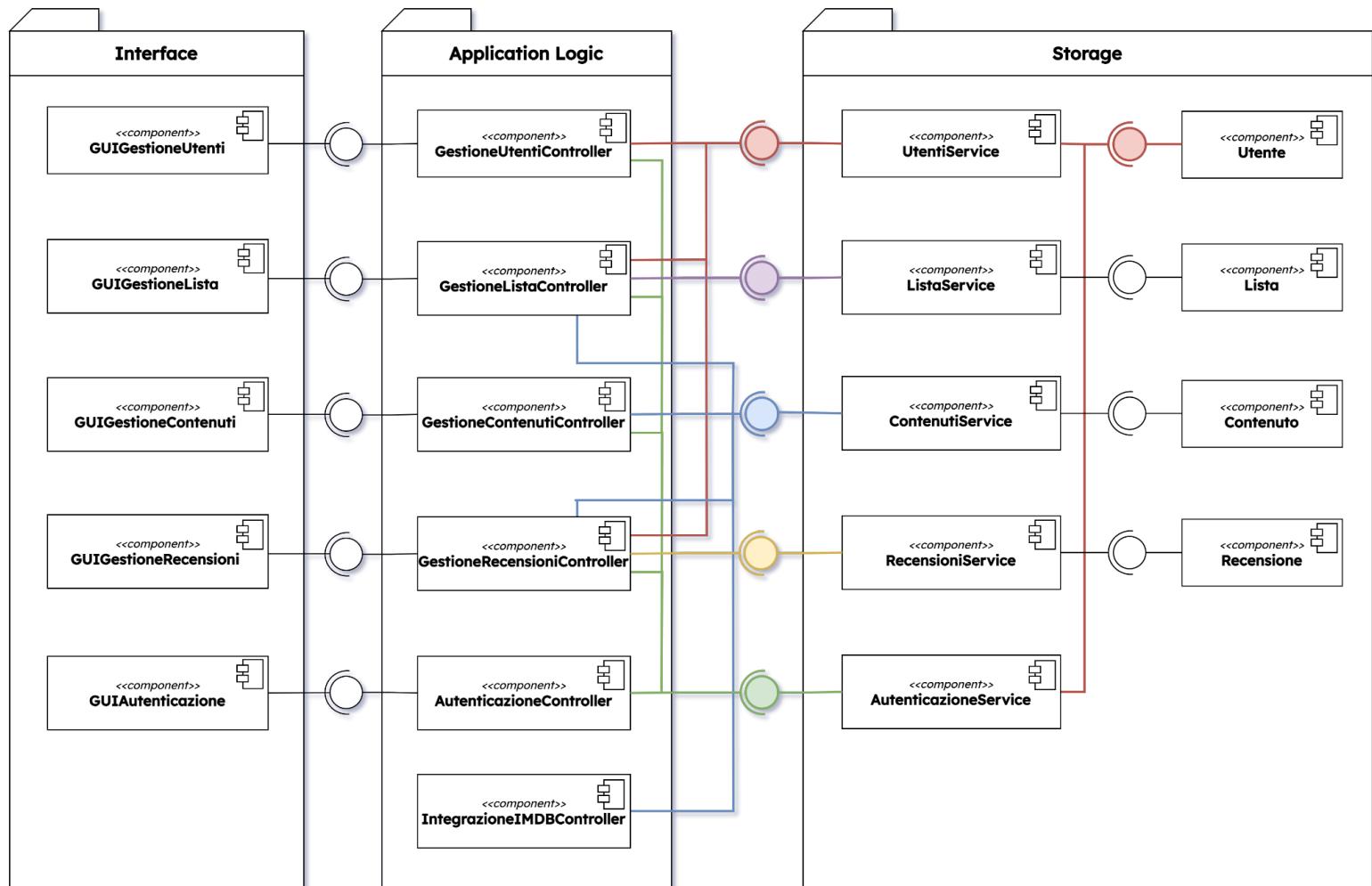
Autori: Christian Esposito
Francesco Durante



Inoltre, nella pagina seguente, è presente una più dettagliata rappresentazione mediante **diagramma architetturale** dei componenti.

Hitchpicks: System Design Document

Autori: Christian Esposito
Francesco Durante
Daniele De Martino



Hitchpicks: System Design Document

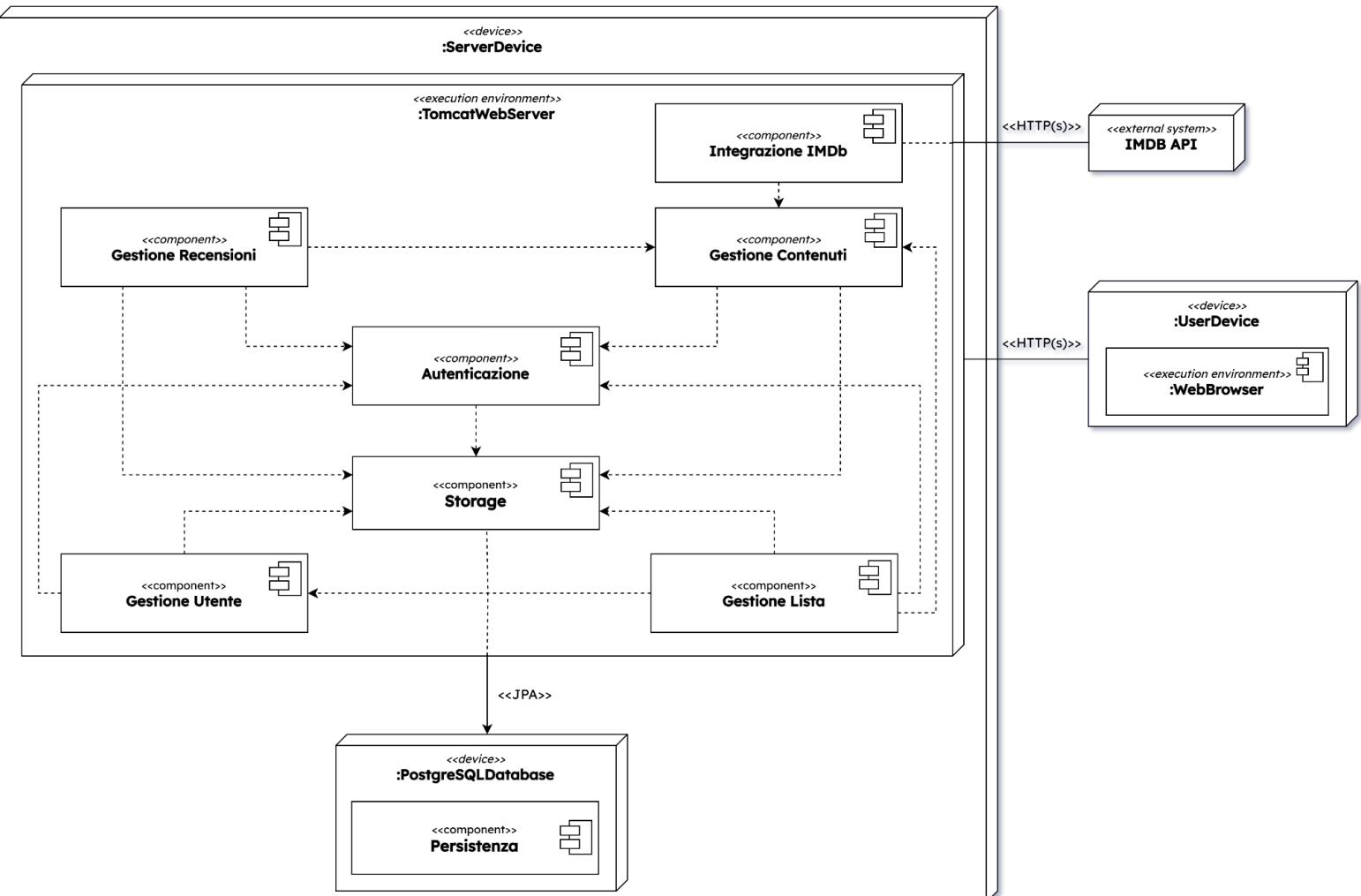
3.3 Mapping hardware/software

Il **mapping hardware/software** del sistema si suddivide nei dispositivi:

- **Server**, rappresentante la macchina (Unix o Windows che sia) che eseguirà il **Web Server Tomcat** e il **DBMS PostgreSQL**. Si noti che i due interagiscono mediante il sottosistema “**Storage**”, il quale fa uso della tecnologia **JPA** (ed internamente quindi **JDBC**) per accedere alla base di dati;
- **Client**, reso nel diagramma come “**UserDevice**”, rappresentante la macchina dell’utente finale, la quale eseguirà un qualunque **Web Browser** ed interagirà col server mediante HTTP(s);
- **Servizio esterno IMDB**, di natura fisica ignota, il quale comunicherà col solo sottosistema “**Integrazione IMDb**” tramite HTTP(s).

I sottosistemi “**Storage**” e “**Integrazione IMDb**” fungono da interfacce allo scopo di ridurre l’**accoppiamento**. Di seguito, il **deployment diagram** del sistema:

Autore: Daniele De Martino



Hitchpicks: System Design Document

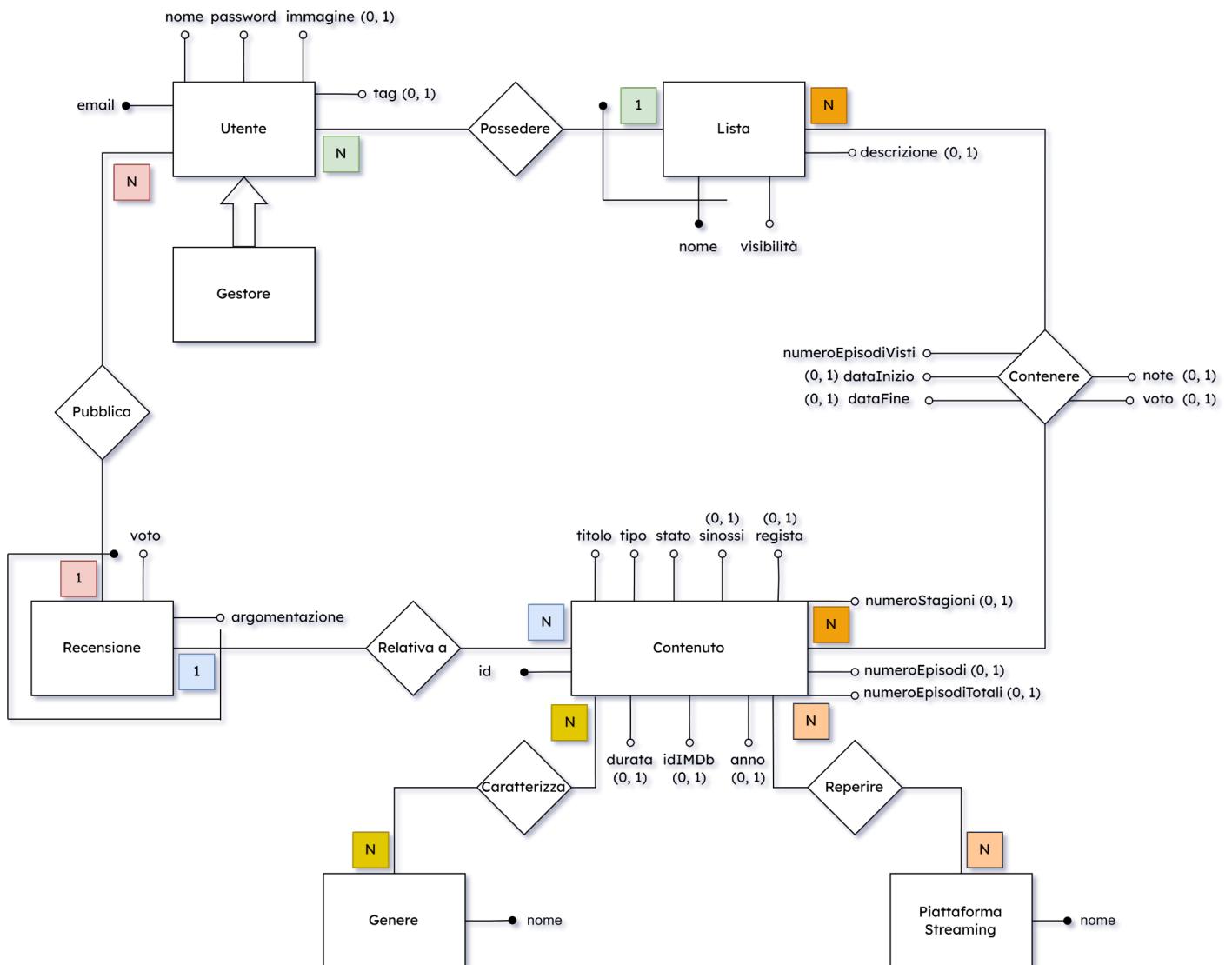
3.4 Gestione dei dati persistenti

Per la gestione dei **dati persistenti**, si è scelto l'utilizzo di un **DBMS**, in particolare il software open-source **PostgreSQL**, in quanto semplifica l'accesso **concorrente, consistente ed atomico** ai dati. La motivazione della scelta rispetto ad altri più noti DBMS sono le funzionalità aggiuntive di PostgreSQL - ad esempio il tipo di dato array - che sicuramente risparmiano complessità e tempo in fase di sviluppo.

E/R: Schema concettuale Entity-Relationship del database

Nonostante la visione concettuale del database già fornita dal diagramma delle classi (non ristrutturato), si è preferito rappresentarlo anche tramite **schema Entity/Relationship** per offrire una panoramica differente della base di dati.

Autore: Federico De Rosa



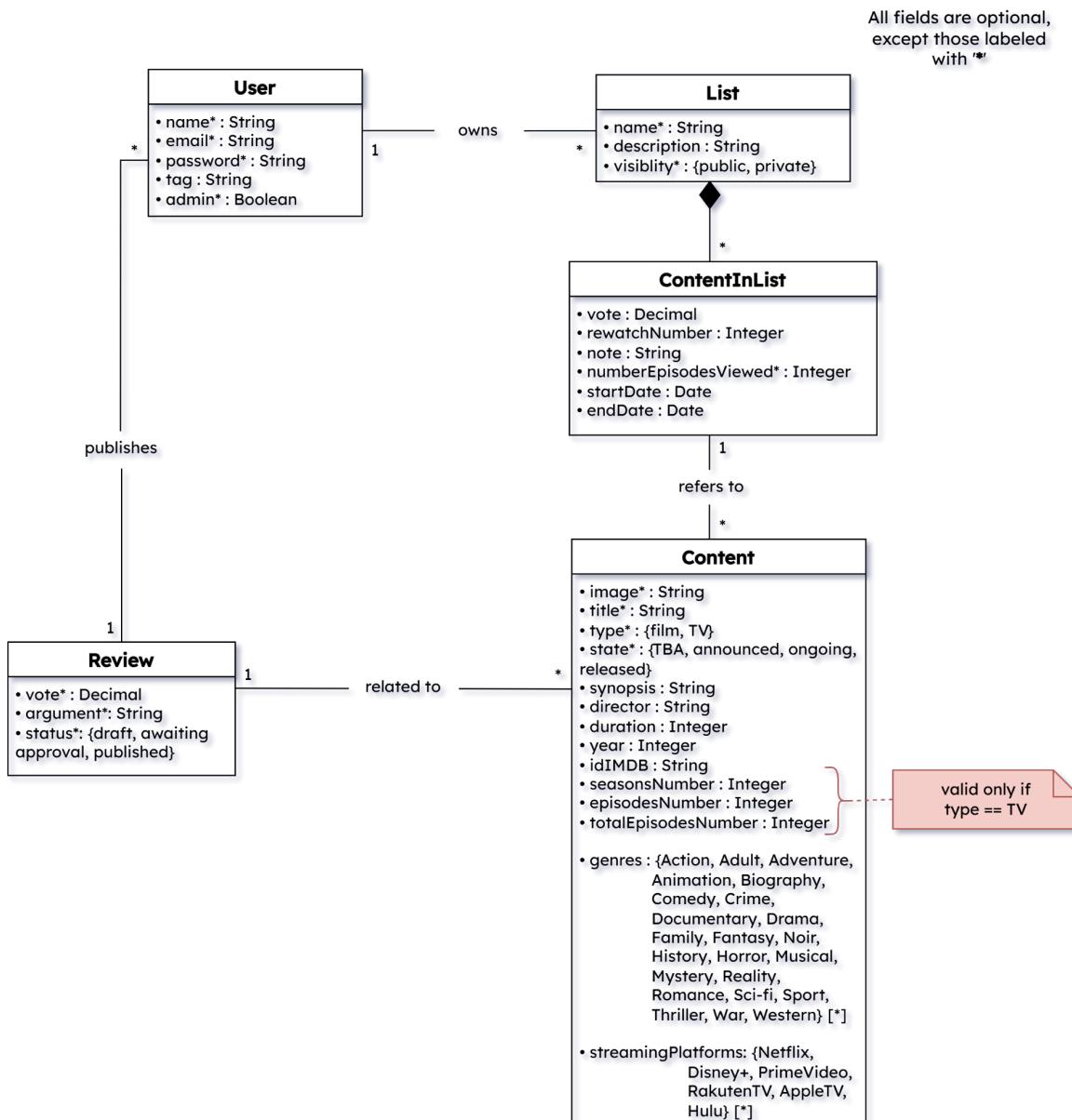
Hitchpicks: System Design Document

Diagramma delle classi ristrutturato

Si è ritenuto necessario **ristrutturare** il diagramma delle classi, sfruttando il potenziale della tecnologia DBMS scelta (PostgreSQL), per ridurre il gap tra il diagramma e l'effettivo schema fisico del database. Le modifiche effettuate sono:

- **rimozione** della classe **Gestore**: per identificare un gestore, è sufficiente l'attributo booleano “gestore” nella classe **Utente**;
- **rimozione** delle classi **Genere** e **PiattaformaStreaming**, sostituite da un array di enum **chiaramente specificati**;
- **traduzione** di tutti gli attributi e valori enum in lingua **inglese**, per avvicinarli agli effettivi nomi poi utilizzati nel sistema.

Autore: Orazio Torre



Hitchpicks: System Design Document

3.5 Controllo degli accessi e sicurezza

Il sistema, per quanto riguarda la sicurezza, utilizza un classico sistema di autenticazione tramite **indirizzo email** e **password**. Un nuovo utente della piattaforma può quindi effettuare la **registrazione**.

Un utilizzatore della piattaforma è identificato come ospite quando non è autenticato, diventando utente (o gestore, se lo è) qualora effettui l'accesso tramite **log-in**.

I gestori possono accedere al **pannello dei gestori**.

Un utente può divenire gestore solo se in primo luogo un gestore gli fornisce i permessi, tramite la “gestione utenti” nel pannello.

Le password, in quanto dati sensibili, vengono gestite e trattate secondo il **GDPR**, tramite l'utilizzo di **algoritmi di hashing** (nello specifico SHA-256, vedi [DG_06](#)) e tecnica di **salt**.

Di seguito, è riportata la **matrice degli accessi**, riportante quali attori possono accedere a quali funzionalità offerte dal sistema (derivanti dai servizi del sistema).

Se un attore prova ad accedere ad una funzionalità per la quale non è autorizzato, il sistema non deve necessariamente rispondere in modo intuitivo e chiaro (es. mostrare un messaggio d'errore chiaro), poiché il sistema è progettato per non permettere dal principio che tale accesso si verifichi. Questo quindi non va in conflitto col [DG_04](#).

Matrice degli accessi

Autore: Federico De Rosa

	Ospite	Utente	Gestore
Autenticazione	<ul style="list-style-type: none">• Log-in• Registrazione	<ul style="list-style-type: none">• Log-in• Log-out	<ul style="list-style-type: none">• Log-in• Log-out• VisualizzaPannelloGestore
Gestione	<ul style="list-style-type: none">• VisualizzaContenuto	<ul style="list-style-type: none">• VisualizzaContenuto	<ul style="list-style-type: none">• VisualizzaContenuto

Hitchpicks: System Design Document

Contenuti	<ul style="list-style-type: none"> • RicercaContenuti • FiltraContenuti 	<ul style="list-style-type: none"> • RicercaContenuti • FiltraContenuti • AggiungiContenuto Lista • ModificaContenuto Lista • RimuoviContenuto Lista 	<ul style="list-style-type: none"> • RicercaContenuti • FiltraContenuti • AggiungiContenuto Lista • ModificaContenuto Lista • RimuoviContenuto Lista • AggiungiContenuto Catalogo • ModificaContenuto Catalogo • RimuoviContenuto Catalogo
Gestione Recensioni	<ul style="list-style-type: none"> • VisualizzaRecensione 	<ul style="list-style-type: none"> • VisualizzaRecensione • CreaRecensione • EliminaRecensione Personale 	<ul style="list-style-type: none"> • VisualizzaRecensione • CreaRecensione • RimuoviRecensione
Gestione Utente	<ul style="list-style-type: none"> • RicercaUtente • VisualizzaUtente 	<ul style="list-style-type: none"> • RicercaUtente • VisualizzaUtente 	<ul style="list-style-type: none"> • RicercaUtente • VisualizzaUtente • RimuoviUtente • VisualizzaElenco Utenti
Gestione Lista	<ul style="list-style-type: none"> • VisualizzaLista • FiltraLista • VisualizzaStatistiche Lista 	<ul style="list-style-type: none"> • VisualizzaLista • FiltraLista • VisualizzaStatistiche Lista • CreaLista Personalizzata • ModificaLista Personalizzata • RimuoviLista Personalizzata 	<ul style="list-style-type: none"> • VisualizzaLista • FiltraLista • VisualizzaStatistiche Lista • CreaLista Personalizzata • ModificaLista Personalizzata • RimuoviLista Personalizzata
Integrazione IMDb			<ul style="list-style-type: none"> • Compilazione AutomaticaIMDb

3.6 Controllo globale del software

Come specificato nel [DG_09](#), il sistema interattivo è disponibile come piattaforma web. Quindi, è presente un'interfaccia grafica accessibile dal browser ed ogni



Hitchpicks: System Design Document

comando è manualmente impartito dall'utente, il quale, ricevuto dal sistema, esegue poi la funzionalità associata.

Il sistema è quindi **event-driven**, sempre in attesa di eventuali comandi da parte dell'utilizzatore, ai quali risponderà di conseguenza.

Non solo, però: il sistema è anche basato su **threads**, in quanto permette la parallelizzazione sia di multipli comandi impartiti dallo stesso utente, sia di molteplici utenti in contemporanea su dispositivi differenti.

La gestione della comunicazione degli eventi avviene tramite **HTTP** e, trattandosi di un protocollo stateless, a meno che l'utente non sia autenticato, non è necessario distinguere tra più richieste impartite dallo stesso utente.

La gestione dei threads, invece, è effettuata automaticamente dal framework applicativo **Spring**, il quale permette la creazione di un thread per ciascuna nuova richiesta HTTP.

Hitchpicks: System Design Document

3.7 Condizioni limite

Si riportano i casi d'uso relativi alle **condizioni limite** del sistema, ossia **avvio, spegnimento e fallimento dell'accesso ai dati persistenti**.

CUBC_01 - Avvio del sistema

Identificativo	CUBC_01	Autore	Daniele De Martino
Titolo	Avvio del sistema		
Descrizione	Lo UC permette l'avvio del sistema.		
Attore Principale	Gestore		
Entry Condition	Il gestore possiede l'accesso alla macchina su cui è ospitato il sistema AND Il gestore accede alla macchina (localmente o da remoto) AND Il gestore può effettuare comandi e gestire i file della macchina.		
Exit Condition ON SUCCESS	Il sistema si avvia.		
Exit Condition ON FAILURE	Il sistema non si avvia AND Il sistema mostra un messaggio di errore.		

FLUSSO DI EVENTI PRINCIPALE/MAIN SCENARIO

1	Gestore	Esegue il comando che avvia il sistema sulla macchina.
2	Sistema	Si accerta che la rete sia disponibile e che il DBMS è in esecuzione ed accessibile con le credenziali che possiede.
3	Sistema	Si avvia correttamente e rende disponibili i suoi servizi ai client.

I Scenario/Flusso di eventi di errore: la rete non è disponibile

2A.1	Sistema	Mostra un messaggio che indica che la rete non è disponibile.
2A.2	Sistema	Non si avvia.



Hitchpicks: System Design Document

II Scenario/Flusso di eventi di errore: il DBMS non è disponibile o accessibile

2B.1	Sistema	Mostra un messaggio che indica che il DBMS non è disponibile o le credenziali sono errate.
2B.2	Sistema	Non si avvia.



Hitchpicks: System Design Document

CUBC_02 - Spegnimento del sistema

Identificativo	CUBC_02	Autore	Orazio Torre
Titolo	Spegnimento del sistema		
Descrizione	Lo UC permette lo spegnimento del sistema.		
Attore Principale	Gestore		
Entry Condition	Il gestore possiede l'accesso alla macchina su cui è ospitato il sistema AND Il gestore accede alla macchina (localmente o da remoto) AND Il gestore può effettuare comandi e gestire i file della macchina AND Il sistema è in esecuzione.		
Exit Condition ON SUCCESS	Il sistema si spegne.		
Exit Condition ON FAILURE	N/A		

FLUSSO DI EVENTI PRINCIPALE/MAIN SCENARIO

1	Gestore	Esegue un comando che spegne il sistema sulla macchina, o arresta il processo (o i processi) in esecuzione.
2	Sistema	Il sistema si spegne, chiudendo immediatamente tutte le connessioni aperte. Eventuali transazioni in corso vengono automaticamente ripristinate allo stato precedente (rollback).

Hitchpicks: System Design Document

CUBC_03 - Fallimento dell'accesso ai dati persistenti

Identificativo	CUBC_03	Autore	Federico De Rosa
Titolo	Fallimento dell'accesso ai dati persistenti		
Descrizione	Lo UC <i>gestisce</i> la condizione limite nella quale il sistema, durante l'esecuzione, non può accedere ai dati persistenti.		
Attore Principale	Gestore		
Entry Condition	Il sistema è in esecuzione AND Il sistema prova ad accedere al DBMS e l'accesso non va a buon fine.		
Exit Condition ON SUCCESS	Il sistema riesce a riconnettersi ed accedere al DBMS.		
Exit Condition ON FAILURE	Il sistema non riesce a riconnettersi e ritenta finché non riesce AND Il sistema mostra un messaggio d'errore per ogni tentativo fallito.		

FLUSSO DI EVENTI PRINCIPALE/MAIN SCENARIO

1	Sistema	Il sistema, tentando l'accesso al DBMS per effettuare una query, si accorge che l'accesso ai dati persistenti è impossibile.
2	Sistema	Il sistema ritenta l'accesso al DBMS.
3	Sistema	Il sistema riesce a connettersi correttamente e torna alla normale operatività.

I Scenario/Flusso di eventi di errore: il tentativo non va a buon fine

2A.1	Sistema	Mostra un messaggio, visibile solo ai gestori con l'accesso alla macchina, che indica che è stato impossibile connettersi al DBMS (con la ragione).
2A.2	Sistema	Ritenta l'accesso al DBMS finché non riesce (torna al punto 2 dello use case).

Hitchpicks: System Design Document

4 Servizi dei sottosistemi

Autori: Christian Esposito
Francesco Durante

Sottosistema “Autenticazione”

Servizio	Descrizione	Interfaccia
Login	Il servizio permette di effettuare l'accesso al sistema tramite le proprie credenziali.	AutenticazioneService
Logout	Il servizio permette di disconnettersi dal sistema.	AutenticazioneService
Registrazione	Il servizio permette di registrare un nuovo utente al sistema.	AutenticazioneService

Sottosistema “Gestione Utente”

Servizio	Descrizione	Interfaccia
Visualizzazione complessiva degli utenti	Il servizio permette di visualizzare un elenco completo di tutti gli utenti registrati nella piattaforma.	UtentiService
Rimozione utente	Il servizio permette ad un gestore di poter rimuovere un utente dalla piattaforma.	UtentiService
Ricerca utente	Il servizio permette di poter ricercare un utente tramite chiave di ricerca.	UtentiService

Hitchpicks: System Design Document

Sottosistema “Gestione Lista”

Servizio	Descrizione	Interfaccia
Creazione lista	Il servizio permette ad un utente di creare una lista personalizzata personale.	ListaService
Eliminazione lista	Il servizio permette ad un utente di rimuovere una lista personalizzata personale.	ListaService
Modifica lista	Il servizio permette ad un utente di poter modificare una lista personalizzata.	ListaService
Visualizzazione lista	Il servizio permette di visualizzare una lista.	ListaService
Visualizzazione statistiche lista	Il servizio permette di visualizzare determinate statistiche per una lista.	ListaService

Sottosistema “Gestione Contenuti”

Servizio	Descrizione	Interfaccia
Aggiunta contenuti a lista	Il servizio permette ad un utente di inserire un contenuto all'interno di una lista, con relativi dati e note personali.	ContenutiService
Modifica contenuti in lista	Il servizio permette ad un utente di modificare un contenuto all'interno di una lista, con relativi dati e note personali.	ContenutiService
Rimozione contenuti da lista	Il servizio permette ad un utente di rimuovere un contenuto da una lista.	ContenutiService
Aggiunta contenuti	Il servizio permette di inserire un nuovo contenuto all'interno del catalogo.	ContenutiService
Modifica contenuti	Il servizio permette di modificare un contenuto all'interno della catalogo.	ContenutiService
Rimozione contenuti	Il servizio permette di rimuovere un contenuto all'interno del catalogo.	ContenutiService

Hitchpicks: System Design Document

Visualizzazione contenuto	Il servizio permette di visualizzare un contenuto presente nel catalogo.	ContenutiService
Ricerca contenuti	Il servizio permette di ricercare contenuti presenti nel catalogo tramite chiave di ricerca e filtri.	ContenutiService

Sottosistema “Gestione Recensioni”

Servizio	Descrizione	Interfaccia
Inserimento recensioni	Il servizio permette ad un utente di scrivere una recensione relativa a un contenuto.	RecensioniService
Approvazione recensioni	Il servizio permette ad un gestore di approvare una recensione.	RecensioniService
Visualizzazione recensione	Il servizio permette di visualizzare una recensione.	RecensioniService
Visualizzazione lista recensioni da approvare	Il servizio permette ad un gestore di visualizzare tutte le recensioni da approvare.	RecensioniService
Visualizzazione lista recensioni personali rimosse	Il servizio permette ad un utente di visualizzare tutte le recensioni personali rimosse o disapprovate.	RecensioniService
Rimozione recensioni pubbliche	Il servizio permette ad un utente di rimuovere le proprie recensioni pubbliche, o ad un gestore di rimuovere recensioni pubbliche di altri utenti.	RecensioniService

Hitchpicks: System Design Document

5 Cenni di object design

Di seguito saranno riportati i due **design pattern** identificati utilizzati nella fase di object design.

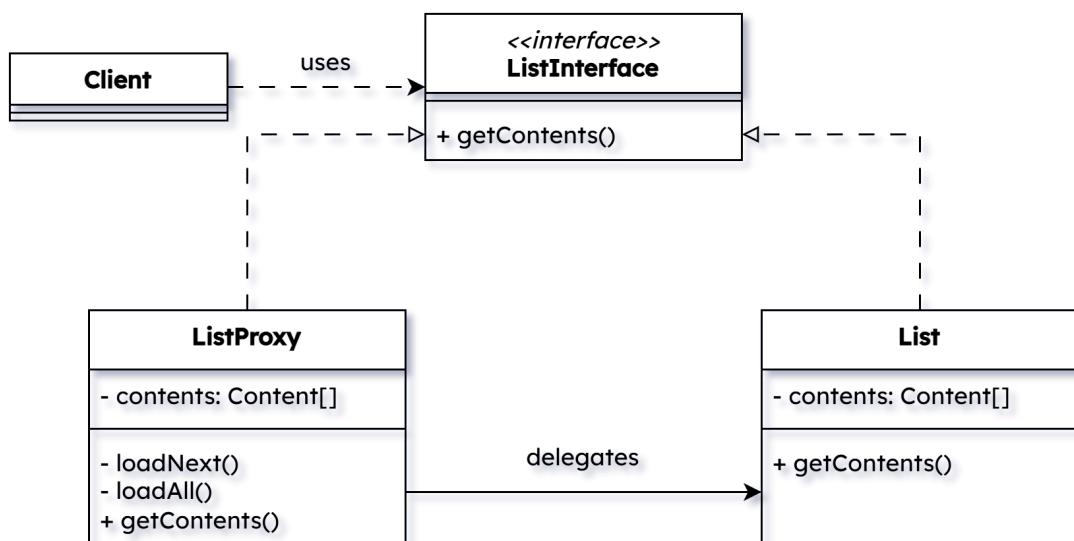
Proxy pattern

Il **proxy** è un **design pattern strutturale** definito dalla Gang of Four (GoF), che fornisce un sostituto o intermediario per controllare l'accesso a un oggetto, aggiungendo funzionalità come caching o logging senza modificare l'oggetto originale.

Nel caso di Hitchpicks, il pattern proxy è utilizzato per controllare l'accesso all'oggetto **Lista**, il quale è computazionalmente molto costoso, poiché l'accesso ad una lista con numerosi contenuti comporta una singola query molto lunga.

Per risolvere questo problema, e migliorare di conseguenza le prestazioni, si preferisce utilizzare l'oggetto **ListaProxy**, il quale carica solo un certo numero di contenuti per volta (n contenuti, dove n è un parametro dato all'oggetto), permettendo quindi di effettuare una vera e propria paginazione dei contenuti - inizialmente saranno caricati solo i primi n , e i successivi n solo quando necessari, tramite il metodo **loadNext()**.

In rari casi può essere strettamente necessario caricare comunque tutti i contenuti, e questo è permesso dall'oggetto **ListaProxy** tramite il metodo **loadAll()**.



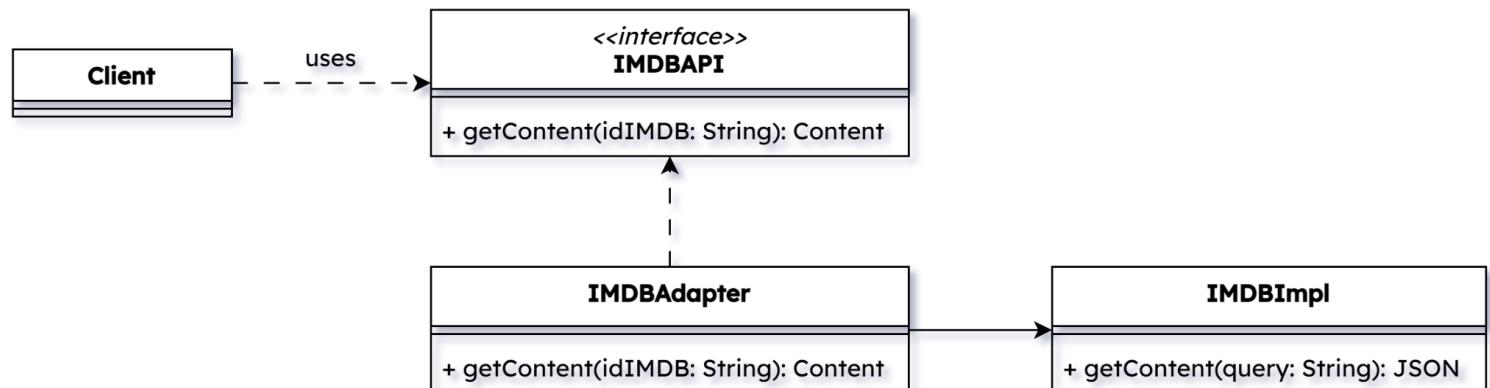
Hitchpicks: System Design Document

Adapter pattern

L'**adapter** è un **design pattern strutturale** definito dalla Gang of Four (GoF), che permette a due interfacce incompatibili di collaborare, convertendo l'interfaccia di una classe nell'altra richiesta dal client.

Nel caso di Hitchpicks, il pattern adapter è utilizzato per permettere l'accesso ai dati dell'API esterno di IMDB, il quale richiede un input in formato JSON (poiché utilizza GraphQL) e restituisce dati in formato JSON.

Le classi che utilizzano l'API, però, si ritroverebbero ogni volta a dover convertire i dati da JSON a classe Java, quindi questo è effettuato dall'Adapter, il quale converte i dati JSON nell'oggetto **Content** per facilitare l'accesso.



Hitchpicks: System Design Document

6 Glossario

Di seguito sono riportate le definizioni precise di alcuni termini utilizzati. I termini nel documento sono da interpretarsi sempre secondo la definizione assegnata.

Termine	Definizione
API	Application Programming Interface, insieme di definizioni e protocolli per la creazione e l'integrazione di applicazioni software.
Crash	Interruzione completa del funzionamento del sistema.
Tomcat	Implementazione open-source delle servlet Jakarta ed expression language, realizzata da Apache. Utilizzata di default da Spring.
JPA	Jakarta Persistence API, libreria utilizzata per la gestione dello schema e l'accesso ad un database, che internamente fa uso di JDBC API, che è un driver di accesso diretto ai database.
Off-the-shelf	Un software già pronto all'uso, fornito da terze parti, integrato nel sistema.
DBMS	DataBase Management System, un software off-the-shelf progettato per memorizzare e gestire dati persistenti.
Concorrente	Un DBMS è definito concorrente se può operare in parallelo, con più operazioni in esecuzione nello stesso istante.
Consistente	Un DBMS è definito consistente se i vincoli d'integrità sono soddisfatti prima e dopo un'operazione, cioè si trova in uno stato logicamente coerente.
Atomico	Un DBMS è definito atomico se mantiene la coerenza dei dati in caso di guasti, ripristinando tutte le modifiche parziali fatte precedentemente, per garantire la consistenza.
Query	Interrogazione ad un DBMS, per accedere a specifici attributi di determinati dati.
Entity/Relationship	Un tipo di schema concettuale per modellare oggetti di interesse in un dominio specifico, spesso utilizzato per modellare concettualmente basi di dati.
Hashing	Una funzione che permette di tradurre dati di arbitraria dimensione in un valore di fissa dimensione, generalmente in modo irreversibile.
Salt	Una sequenza casuale di bit utilizzata, insieme ad una password,

Hitchpicks: System Design Document

	come input di una funzione di hashing, per salvaguardare le password nel sistema.
Event-driven	È un approccio alla progettazione software in cui un componente viene eseguita solo in risposta alla ricezione di un evento.
HTTP	Hypertext Transfer Protocol, protocollo utilizzato da un client (browser, generalmente) per recuperare risorse da un server, tra cui file HTML.