



SDD System Design Document

iLike

Riferimento	
Versione	4.4
Data	17/01/2023
Destinatario	Prof. Carmine Gravino e tutor IS a.a. 2022/23
Presentato da	Costante Luigina, Giorgione Francesco, Lo Conte Simona, Napolillo Marta
Approvato da	



Revision History

Data	Versione	Descrizione	Autori
26/11/2022	0.1	Aggiunta definizioni, riferimenti e organizzazione del documento. Aggiunta DG Usabilità.	Simona Lo Conte
26/11/2022	0.2	Aggiunta <i>Sistema Corrente</i> e DG Interfacce e Affidabilità	Marta Napolillo
26/11/2022	0.3	Aggiunta DG Implementazione e Supportabilità	Luigina Costante
27/11/2022	0.4	Aggiunta <i>Introduzione</i> e DG Performance	Francesco Giorgione
29/11/2022	0.5	Modifica <i>design goal</i> e aggiunta <i>trade-off</i>	Tutto il team
29/11/2022	0.6	Modifica DG_9 e DG_10	Francesco Giorgione
30/11/2022	0.7	Aggiunta descrizione Obiettivi di design e Aggiunta Component Diagram della divisione in sottosistemi.	Tutto il team
01/12/2022	0.8	Modifica Component Diagram e aggiunta descrizione	Francesco Giorgione
02/12/2022	1.0	Aggiunta matrice degli accessi e UCBC_1	Francesco Giorgione
02/12/2022	1.1	Aggiunta UCBC_2 e servizi dei sottosistemi Gestione account e Gestione Segnalazioni	Luigina Costante
02/12/2022	1.2	Aggiunta UCBC_3 e servizi dei sottosistemi Gestione Recensioni e Gestione Contenuti	Simona Lo Conte
04/12/2022	1.3	Aggiunta UCBC_4 e servizi dei sottosistemi Gestione Profilo e Gestione Liste	Marta Napolillo
05/12/2022	1.4	Modifiche varie. Aggiunta Class Diagram e aggiunta Diagramma Architettuale.	Tutto il team
05/12/2022	1.5	Aggiunta tabelle dizionario dei dati relative alle entità: contenuto,	Luigina Costante



Laurea Triennale in informatica - Università di Salerno
Corso di *Ingegneria del Software* – Prof. C. Gravino

		film, serie Tv, libro e album musicale	
05/12/2022	1.6	Aggiunta tabelle dizionario dei dati relative a Recensione, Segnalazione e Interazione	Simona Lo Conte
06/12/2022	1.7	Modifica dell'architettura del sistema corrente. Aggiunta dei punti 3.1 e 3.6.	Marta Napolillo
06/12/2022	1.8	Modifica del dizionario dei dati	Francesco Giorgione
07/12/2022	2.0	Aggiunta sezione 3.3 e modifiche varie	Francesco Giorgione
10/12/2022	2.1	Modifiche varie	Marta Napolillo
12/12/2022	3.0	Revisione	Tutto il team
14/12/2022	3.1	Aggiunta dei link alla repository di GitHub e al Drive condiviso	Simona Lo Conte
29/12/2022	3.2	Modifiche Dizionario dei dati	Simona Lo Conte
30/12/2022	3.3	Revisione	Francesco Giorgione
31/12/2022	3.4	Modifiche varie	Marta Napolillo
02/01/2023	3.5	Revisione trade-off e modifica class diagram (in seguito a feedback consegna intermedia)	Tutto il team
03/01/2023	3.6	Revisione dizionario dei dati	Francesco Giorgione
03/01/2023	3.7	Revisione dizionario dei dati	Marta Napolillo
04/01/2023	3.8	Modifica mapping hw-sw e class diagram ristrutturato	Francesco Giorgione
05/01/2023	3.9	Modifica trade-off	Francesco Giorgione
05/10/2023	3.10	Modifica dizionario dei dati	Francesco Giorgione
06/01/2023	3.11	Modifica class diagram	Francesco Giorgione
07/01/2023	3.12	Modifica class diagram	Marta Napolillo
07/01/2023	3.13	Modifica dizionario dei dati	Francesco Giorgione



07/01/2023	3.14	Modifica Class Diagram	Marta Napolillo
09//01/2023	3.15	Modifica sottosistema <i>Gestione contenuti</i>	Francesco Giorgione
10/01/2023	3.16	Modifica sottosistema <i>Gestione contenuti e design goal.</i>	Francesco Giorgione
10/01/2023	3.17	Modifica DG_10, DG_11	Francesco Giorgione
11/01/2023	4.0	Modifica suddivisione in sottosistemi e relativi diagrammi UML	Francesco Giorgione
14/01/2023	4.1	Modifica 3.1	Francesco Giorgione
15/01/2023	4.2	Modifica Class Diagram	Marta Napolillo
16/01/2023	4.3	Modifiche varie	Marta Napolillo
17/01/2023	4.4	Revisione	Francesco Giorgione

Sommario

Revision History	2
1 Introduzione	6
1.1 Scopo del Sistema	6
1.2 Obiettivi di Design	6
Design goals	6
Trade-off	8
1.3 Definizioni, acronimi, e abbreviazioni	9
1.4 Riferimenti	9
1.5 Organizzazione del documento	9
2 Architettura del sistema corrente	10
3 Architettura del sistema proposto	10



3.1	Panoramica sulla sezione	10
3.2	Decomposizione in sottosistemi	11
	Diagramma architetturale	13
3.3	Mapping hardware/software	14
3.4	Gestione dei dati persistenti	16
	CD_SDD: Entity Class Diagram ristrutturato	16
	Dizionario dei dati	18
3.5	Controllo degli accessi e sicurezza	25
3.6	Controllo globale del software	26
3.7	Condizione limite	27
4	Servizi dei sottosistemi	30
	Sottosistema Gestione account	30
	Sottosistema Gestione segnalazioni	30
	Sottosistema Gestione recensioni	31
	Sottosistema Gestione contenuti	31
	Sottosistema Gestione liste	31
5	Glossario	32



1 Introduzione

1.1 Scopo del Sistema

iLike è un'applicazione Android che consente ai suoi utenti di recensire libri, album musicali, film e serie TV. Considerata la costante crescita del numero di fruitori di tali contenuti, la recensione rappresenta una fondamentale forma di interazione tra le persone. A tale scopo, *iLike* si propone come luogo di incontro di persone che condividono gli stessi interessi. Inoltre, garantisce un'interconnessione – seppure indiretta – tra il pubblico e i content creator: questi ultimi potranno tener conto dei feedback degli utenti in vista di eventuali produzioni future. Ciò che differenzia il nostro sistema dai principali competitor è la presenza di contenuti di diverso tipo, per l'appunto libri, album musicali, film e serie TV. Attualmente, infatti, non esiste una piattaforma unica che consenta di recensire una così ampia gamma di contenuti.

1.2 Obiettivi di Design

In questa sezione si descrivono i Design Goals, ossia le caratteristiche di qualità dovrebbero essere ottimizzate nel sistema, fornendone una priorità, una descrizione e una categoria di appartenenza. Inoltre, vengono definiti i trade-off tra design goals del sistema che sono impari tra loro.

Design goals

Rank	ID Design Goal	Descrizione	Categoria	RNF di origine
11	DG_1 Interfaccia utente	Il sistema deve offrire un'interfaccia utente "facile da ricordare", in modo che l'utente possa interagirvi in modo efficace anche dopo un lungo periodo di inutilizzo.	End User	RNF_U_2



10	DG_2 Comprensibilità	Il sistema deve risultare comprensibile agli utenti anche senza far uso di documentazioni specifiche.	End User	RNF_U_3
7	DG_3 Gestione dei fallimenti	Il sistema deve saper reagire alle situazioni di fallimento inviando messaggi all'utente per notificare l'errore, permettendogli di completare l'operazione.	Dependability	RNF_A_1
1	DG_4 Affidabilità delle operazioni	Il sistema deve garantire che le operazioni avvengano con successo, o che in caso di errori l'utente ne sia notificato.	Dependability	RNF_A_1
2	DG_5 Gestione permessi	Il sistema deve gestire una classificazione degli utenti, al fine di limitare le funzionalità accessibili ad ognuno e mantenere l'integrità del sistema.	Dependability	RNF_A_2
3	DG_6 Sicurezza database	Il sistema dovrà garantire la sicurezza delle password degli utenti memorizzate nel DB utilizzando l'algoritmo di crittografia SHA-512.	Dependability	RNF_IN_1
4	DG_7 Manutenibilità	Il sistema dovrà essere stabile, cioè capace di evitare effetti indesiderati a seguito di modifiche, correzioni e adattamenti al software.	Maintenance	RNF_S_1
9	DG_8 Connessione ad Internet	Tutti gli smartphone connessi ad Internet dovranno poter utilizzare il sistema.	Dependability	RNF_S_2
6	DG_9 Tempo di risposta *	Il sistema dovrà garantire un tempo di risposta massimo di 10 secondi anche in situazioni di errore.	Performance	RNF_P_3



8	DG_10 Memoria	Il sistema dovrà avere a disposizione uno spazio di memoria sul dispositivo di almeno 150 MB.	Performance	RNF_P_4
5	DG_11 Utilizzo concorrente	Il sistema potrà essere usato contemporaneamente da un numero di utenti, fino ad un massimo di 500, funzionando in maniera corretta.	Performance	RNF_P_2

* Ai fini del raggiungimento del DG *Tempo di risposta*, si è provveduto a modificare il requisito funzionale *Ricerca contenuti* (RF_GC_3), prevedendo che la stringa di ricerca dovesse avere una lunghezza minima di 4 caratteri. Durante il testing, infatti, si è rilevato che una stringa di ricerca di dimensione inferiore non avrebbe consentito di rispettare il tempo di risposta indicato.

Trade-off

Trade-off	Descrizione
Memoria vs tempo di risposta	Al fine di garantire un tempo di risposta ottimale, si è disposti ad aumentare la memoria occupata dal sistema, ad esempio aggiungendo associazioni dirette tra entità per velocizzare l'accesso alle informazioni.
Tempo di risposta vs connessione ad Internet	Il sistema dovrà in ogni caso rispettare il massimo tempo di risposta indicato. In caso di connessione assente/scadente, verrà visualizzato un messaggio di errore entro i tempi stabiliti.
Effort vs interfaccia utente	Al fine di rispettare i vincoli di tempo e risorse stabiliti, si è disposti a realizzare un'interfaccia utente estremamente basilare. In particolare, considerato il budget limitato, la prima versione dell'applicazione funzionerà soltanto in <i>portrait</i> (orientamento verticale). Ci riserviamo di aggiungere il supporto al <i>landscape</i> (orientamento orizzontale) in successive versioni dell'applicazione.
Tempo di risposta vs manutenibilità	Al fine di garantire un'alta manutenibilità del software, si è disposti ad accettare un aumento dei tempi di risposta.



1.3 Definizioni, acronimi, e abbreviazioni

- **CD_SDD:** class diagram ristrutturato
- **COTS:** Commercial Off-The-Shelf
- **Dati Persistenti:** dati che devono essere salvati
- **DG:** Design goal, cioè caratteristiche che descrivono la qualità del sistema
- **Mapping Hardware/Software:** studio della connessione tra parti fisiche e logiche del sistema
- **RAD:** Requirements Analysis Document
- **SDD:** System Design Document
- **Sottosistema:** insieme di servizi del dominio applicativo
- **UCBC:** Use case boundary condition

1.4 Riferimenti

- Bern Bruegge, Allen H. Dutoit, Object-Oriented Software Engineering – Using UML, Patterns and Java, 3rd edition
- Materiale fornito dal docente e dai tutor sulla piattaforma e-learning
- [Repository GitHub](#)
- [Drive condiviso](#)

1.5 Organizzazione del documento

Il presente documento è strutturato nel seguente modo:

1. **Introduzione:** la seguente sezione contiene lo scopo del sistema, gli obiettivi di design oltre che una panoramica sulle definizioni, acronimi e abbreviazioni presenti nel documento.
2. **Architettura sistema corrente:** Fornisce una descrizione dell'architettura del software già esistente.
3. **Architettura sistema proposto:** descrive il sistema che si intende realizzare, definendone la decomposizione in sottosistemi, il mapping Hardware/Software e la gestione dei dati persistenti. Inoltre, viene gestito il controllo degli accessi e della sicurezza tramite la matrice degli accessi, insieme ad un controllo globale del sistema. Infine, si definiscono le condizioni limite del sistema.
4. **Glossario:** Descrive i termini tecnici presenti nel SDD.



2 Architettura del sistema corrente

Attualmente esistono piattaforme simili che permettono la gestione di recensioni per una specifica categoria tra film, serie TV, libri e album musicali. Tra queste un esempio è l'applicazione *TV Time*; oltre alle piattaforme che ne consentono la visualizzazione e l'acquisto. Tale applicazione è basata su tre livelli: *Storage*, che accede a *TheTVDB* per archiviare informazioni riguardanti film e serie TV, *Application* e *Presentation*. Il sistema corrente, dunque, è basato sullo stile architetturale *Three-Tier*.

3 Architettura del sistema proposto

3.1 Panoramica sulla sezione

Al fine di rispettare il trade-off *tempo di risposta vs manutenibilità*, abbiamo optato per un'architettura chiusa in quanto ciò favorisce la manutenibilità dell'applicazione a svantaggio del tempo di risposta. Il sistema proposto, dunque, è basato sullo stile architetturale *Three-Tier*, che organizza i sottosistemi in tre layer:

- livello di *Storage*, che si occupa della gestione dei dati persistenti;
- livello *Application*, che si occupa della logica di applicazione, in cui vengono eseguite le funzionalità del sistema;
- livello *Presentation*, che si occupa dell'interazione con l'utente.

Ciò consente di aumentare, inoltre, la leggibilità del nostro sistema. Pertanto, tale scelta è particolarmente consigliata per lo sviluppo delle applicazioni mobile.

Si assicura la compatibilità dell'applicazione sviluppata con le API Android 29+, ossia a partire da Android 10.



3.2 Decomposizione in sottosistemi

Sono stati individuati i cinque sottosistemi seguenti.

- **Gestione account:** è responsabile delle funzionalità di login, logout, registrazione dell'iscritto e visualizzazione del profilo personale dell'iscritto.
- **Gestione contenuti:** è responsabile delle funzionalità di visualizzazione dei contenuti nell'home page dell'app, visualizzazione dettagliata di un singolo contenuto e ricerca di un contenuto.
- **Gestione liste:** è responsabile delle funzionalità di creazione di una lista, aggiunta contenuti a una lista esistente e visualizzazione dei contenuti di una lista personale.
- **Gestione recensioni:** è responsabile delle funzionalità di pubblicazione di una recensione, visualizzazione di recensioni e segnalazione di una recensione.
- **Gestione segnalazioni:** è responsabile delle funzionalità – riservate al gestore - di visualizzazione delle segnalazioni, rifiuto di una segnalazione e cancellazione di una recensione.

Si ritiene che tale suddivisione sia in grado di massimizzare la coesione, in quanto tutti gli oggetti nello stesso sottosistema sono funzionalmente correlati. Inoltre, come evidenziato dall'UML component diagram seguente, l'accoppiamento tra sottosistemi non è eccessivo. Si noti che, al fine di ridurre le dipendenze, l'accesso alla componente *Persistence* è realizzato tramite un unico sottosistema *Storage*.

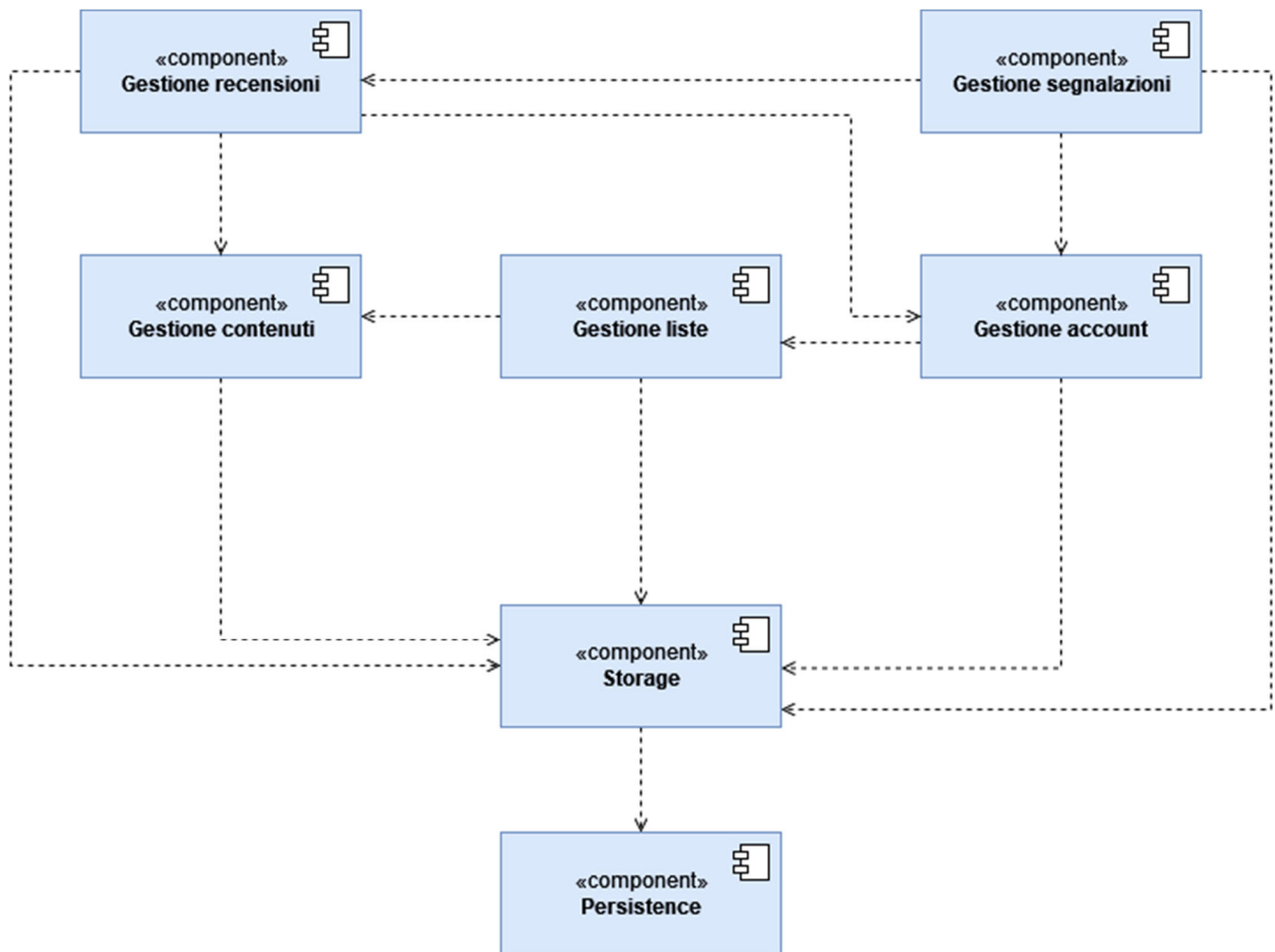
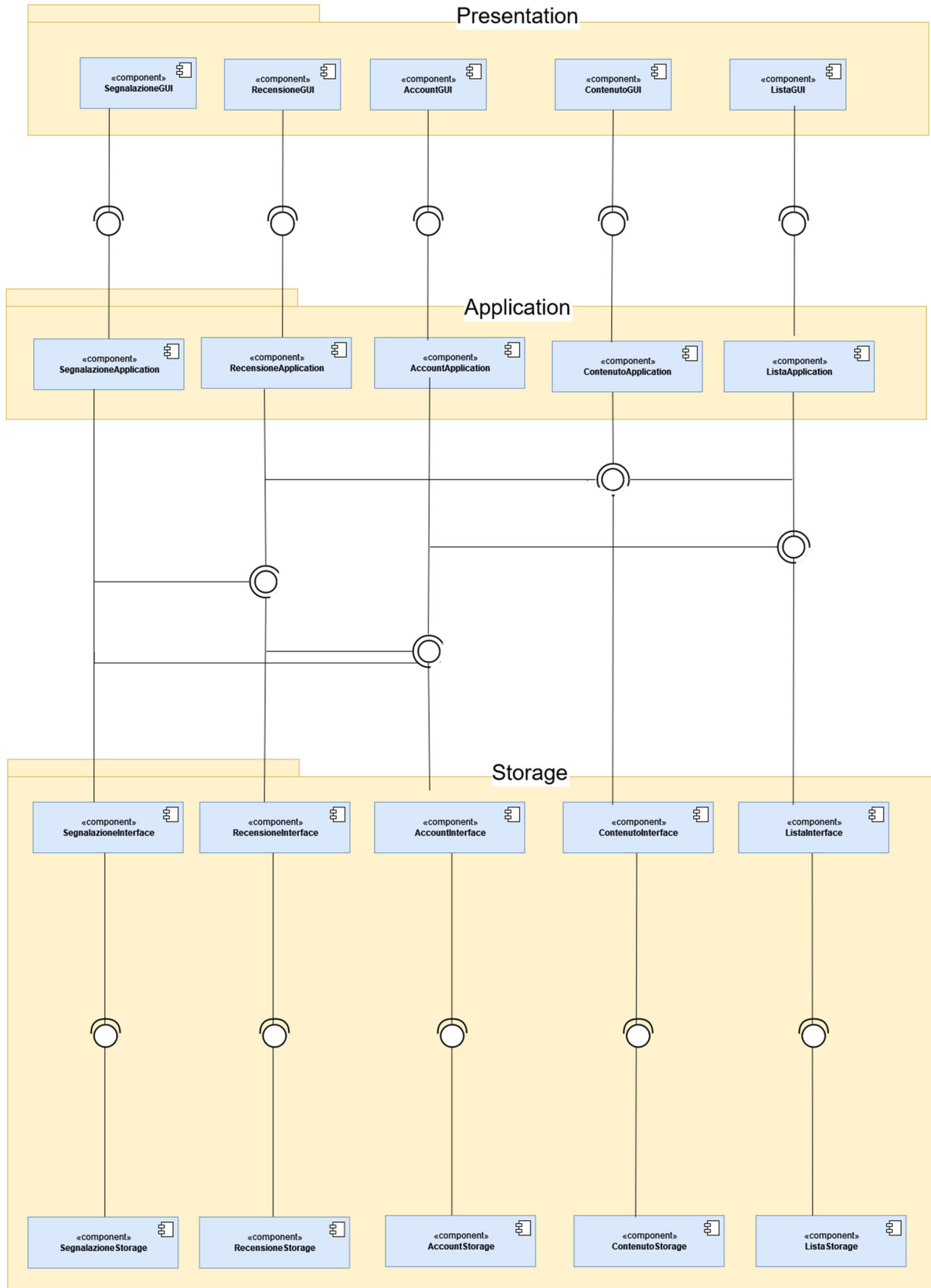


Diagramma architetturale





3.3 Mapping hardware/software

L'applicazione Android che verrà sviluppata si basa su una piattaforma hardware distribuita composta dagli elementi seguenti (si noti che le componenti COTS sono evidenziate in grassetto).

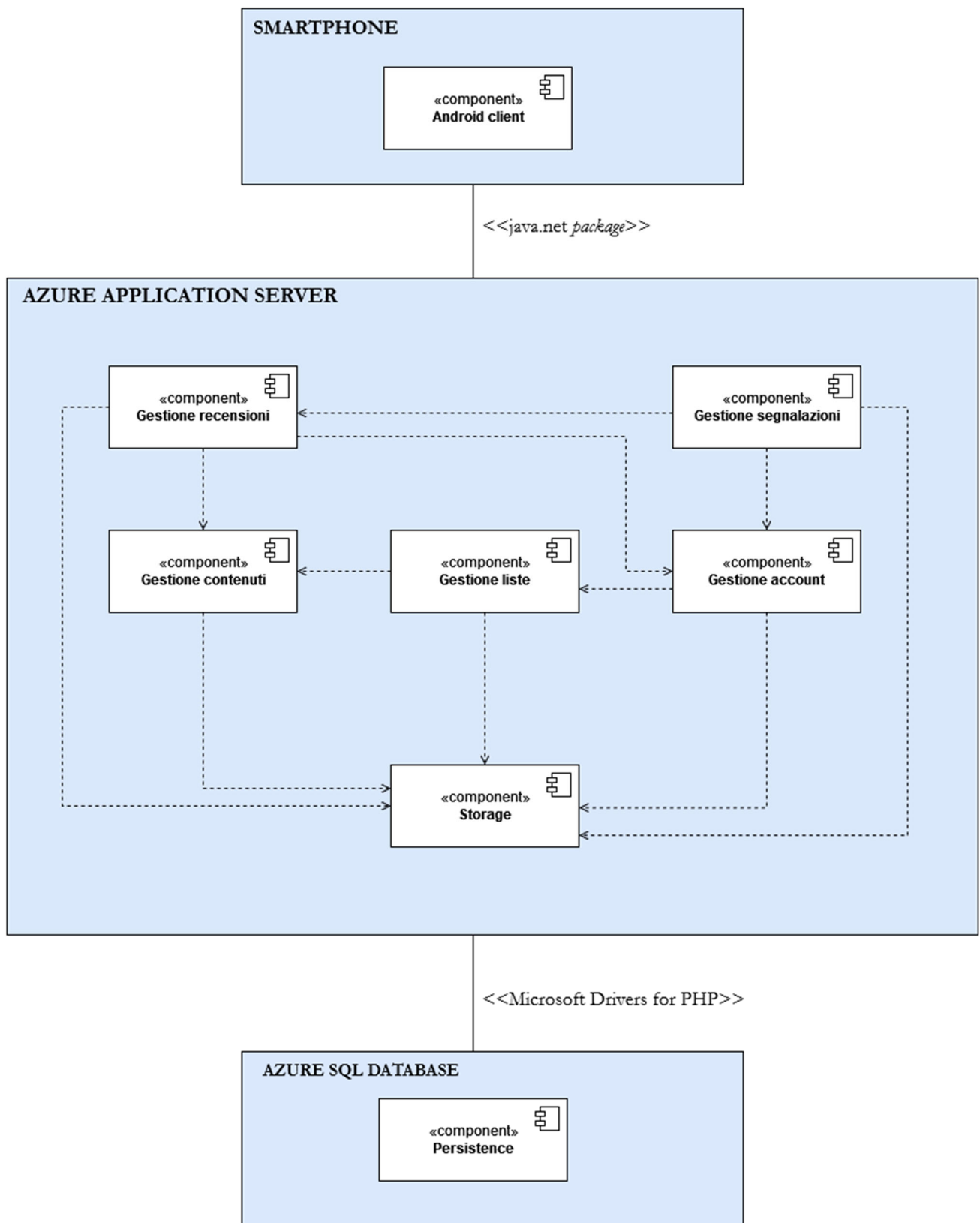
- Un **database SQL Azure** che non si interfaccia direttamente con l'applicazione.
- Un **application server Azure** che costituisce l'interfaccia tra l'applicazione e il database (maggiori dettagli di seguito).

Considerate la difficoltà e la sconvenienza – in termini di prestazioni - di collegare direttamente un'applicazione Android al database, abbiamo ritenuto opportuno caricare su un nostro application server Azure gli script PHP seguenti:

- uno script *select*, che esegue il fetching di dati dal database;
- uno script *update*, che esegue operazioni di aggiornamento (insert, update, delete) nel database.

Tramite richieste **HTTP**, tali script sono invocati dalla classe *it.unisa.iLike.QueryManager*, appositamente implementata per garantire un'interfaccia efficiente con la base di dati.

L'UML deployment diagram seguente fornisce una visione complessiva delle decisioni di mapping assunte.





3.4 Gestione dei dati persistenti

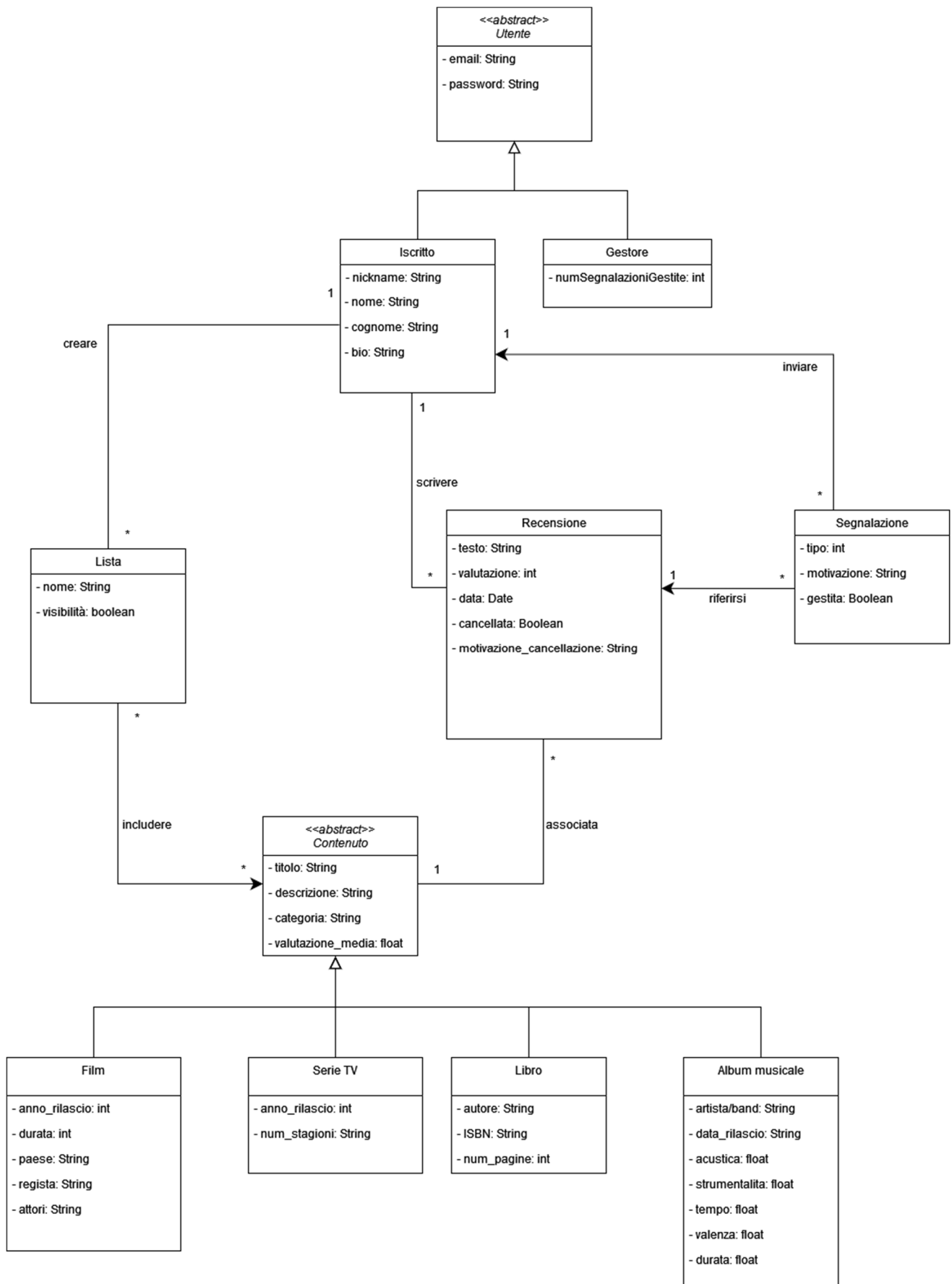
Per la gestione dei dati persistenti, si è ritenuto opportuno utilizzare una base di dati, in modo da usufruire di una serie di servizi forniti dal DBMS e ritenuti essenziali per il nostro sistema. A titolo puramente esemplificativo, si pensi alle operazioni transazionali, l'imposizione di vincoli di integrità sui dati e l'affidabilità.

In particolare, si è stabilito di utilizzare un database relazionale SQL fornito da Azure in cloud. Tale scelta ha come obiettivo prevalente quello di assicurare la massima consistenza delle informazioni ed evitare fenomeni di disallineamento tra i diversi membri del team.

CD_SDD: Entity Class Diagram ristrutturato

Di seguito è riportato l'Entity Class Diagram ristrutturato, che modifica la direzionalità di alcune delle associazioni tra le varie entità.

Gli attributi delle entity Film, Serie TV, Libro, Album musicale sono stati ricavati dai dataset Kaggle (RNF_IN_2).





Dizionario dei dati

Di seguito sono elencati gli attributi di ogni entità individuata.

Nome entità	Contenuto		
Descrizione:	contiene i dati relativi ad un contenuto		
Nome campo	Tipo	Vincolo di chiave	Altri vincoli
id	int	PRIMARY KEY	NOT NULL AUTO_INCREMENT
titolo	varchar(MAX)		NOT NULL
descrizione	varchar(MAX)		NOT NULL
categoria	varchar(MAX)		NOT NULL
tipo*	varchar(MAX)		NOT NULL
valutazione_media**	float		NOT NULL DEFAULT 0

* L'attributo *tipo* individua il tipo di contenuto. I suoi possibili valori sono *film*, *serie_tv*, *libro*, *album*. Si tratta di un attributo che, per ogni istanza *x* di *Contenuto*, può essere calcolato verificando in quale, tra le tabelle figlie *Film*, *SerieTV*, *Libro*, *AlbumMusicale* esiste un'istanza legata alla chiave primaria di *x* da un vincolo di chiave esterna. Tuttavia, considerato che si intende ridurre per quanto possibile il tempo di risposta del sistema (vedi trade-off *Memoria vs Tempo di risposta*), si è stabilito di mantenere tale attributo in *Contenuto* (solo nel DB, non nei Java Bean).

** L'attributo calcolabile *valutazione_media* individua la valutazione media del contenuto. Per le stesse ragioni di prima (soddisfacimento del trade-off *Memoria vs Tempo di risposta*), si è stabilito di mantenerlo.

Nome entità	Film		
Descrizione:	contiene i dati relativi ad un film		
Nome campo	Tipo	Vincolo di chiave	Altri vincoli
id	int	PRIMARY KEY FOREIGN KEY (Contenuto)	NOT NULL



anno_rilascio	int		
durata	int		NOT NULL
paese	varchar(MAX)		
registra	varchar(MAX)		NOT NULL
Attori	varchar(MAX)		NOT NULL

Nome entità:		Serie TV	
Descrizione:	contiene i dati relativi ad una serie TV		
Nome campo	Tipo	Vincolo di chiave	Altri vincoli
id	int	PRIMARY KEY FOREIGN KEY (Contenuto)	NOT NULL
anno_rilascio	int		
num_stagioni	varchar(12)		NOT NULL

Nome entità:	Libro		
Descrizione:	contiene i dati relativi ad un Libro		
Nome campo	Tipo	Vincolo di chiave	Altri vincoli
id	int	PRIMARY KEY FOREIGN KEY (Contenuto)	NOT NULL
autore	varchar(MAX)		NOT NULL
isbn	varchar(13)		
num_pagine	int		NOT NULL



Nome entità:	Album musicale		
Descrizione:	contiene i dati relativi ad un album musicale		
Nome campo	Tipo	Vincolo di chiave	Altri vincoli
id	int	PRIMARY KEY FOREIGN KEY (Contenuto)	NOT NULL
artista	varchar(MAX)		NOT NULL
data_rilascio	varchar(20)		
acustica	float		
strumentalita	float		
tempo	float		
valenza	float		
durata	float		

Nome entità:		Recensione	
Descrizione:	contiene i dati relativi ad una recensione		
Nome campo	Tipo	Vincolo di chiave	Altri vincoli
id	int	PRIMARY KEY	AUTO_INCREMENT NOT NULL
testo	varchar(1000)		NOT NULL



valutazione	int		CHECK(1<=valutazione<=5) NOT NULL
data	date		NOT NULL
cancellata	boolean		DEFAULT false NOT NULL
motivazione_ cancellazione	varchar(300)		CHECK ((NOT NULL AND cancellata = true) OR (IS NULL AND cancellata = false))
email_iscritto	varchar(100)	FOREIGN KEY (Iscritto)	NOT NULL
id_contenuto	int	FOREIGN KEY (Contenuto)	NOT NULL



Nome entità:		Segnalazione	
Descrizione:	contiene i dati relativi ad una segnalazione		
Nome campo	Tipo	Vincolo di chiave	Altri vincoli
id	int	PRIMARY KEY	AUTO_INCREMENT NOT NULL
tipo	int		NOT NULL Valori ammessi: <ul style="list-style-type: none">0: altre segnalazioni1: spoiler alert
motivazione	varchar(500)		NOT NULL
gestita	boolean		NOT NULL DEFAULT false
email_iscritto	varchar(100)	FOREIGN KEY (Iscritto)	NOT NULL
id_recensione	int	FOREIGN KEY (Recensione)	NOT NULL

Nome entità:		Iscritto	
Descrizione:	contiene i dati relativi ad un iscritto		
Nome campo	Tipo	Vincolo di chiave	Altri vincoli
email	varchar(100)	PRIMARY KEY	NOT NULL CHECK(REGEXP '^[A-z0-9\.\+_-]+@[A-z0-9\.\+_-]+\.[A-z]{2,6}\$')
password	varchar(MAX)		NOT NULL CHECK(REGEXP '(?=[A-Za-z])(?=.*[A-Za-z])(?=.*[0-9])(?=.*[!@#\$%^&*~` }{},;:~\.\+_-])\$')



nickname	varchar(30)		UNIQUE NOT NULL
nome	varchar(50)		NOT NULL
cognome	varchar(50)		NOT NULL
bio	varchar(200)		

Nome entità:		Gestore	
Descrizione:	contiene i dati relativi ad un gestore		
Nome campo	Tipo	Vincolo di chiave	Altri vincoli
email	varchar(100)	PRIMARY KEY	NOT NULL CHECK(REGEXP '^[A-z0-9\.\+_-]+@[A-z0-9\.\+_-]+\.[A-z]{2,6}\$')
password	varchar(MAX)		NOT NULL CHECK(REGEXP '^(?=.*[A-Za-z])(?=.*\d)[A-Za-z\d]{8,}\$')
num_segnalazioni_gestite	int		NOT NULL DEFAULT 0

Nome entità:		Lista	
Descrizione:	contiene i dati relativi ad una lista		
Nome campo	Tipo	Vincolo di chiave	Altri vincoli



nome	varchar(50)	PRIMARY KEY	NOT NULL
email_iscritto	varchar(100)	PRIMARY KEY FOREIGN KEY (Iscritto)	NOT NULL
visibilita	boolean		NOT NULL Valori ammessi: <ul style="list-style-type: none"> • 0: privata • 1: pubblica

Nome entità: Inclusioni			
Descrizione:	contiene i dati relativi a quali contenuti appartengono alle liste degli iscritti.		
Nome campo	Tipo	Vincolo di chiave	Altri vincoli
nome_lista	varchar(50)	PRIMARY KEY FOREIGN KEY (Lista)	NOT NULL
email_iscritto	varchar(100)	PRIMARY KEY FOREIGN KEY (Lista)	NOT NULL
id_contenuto	int	PRIMARY KEY FOREIGN KEY (Contenuto)	NOT NULL

Nota: nel campo *Altri vincoli* del dizionario delle entità *Iscritto* e *Gestore* sono riportate le seguenti espressioni regolari:

- *email*: `'^[A-z0-9\._+]+@[A-z0-9\._+]\.[A-z]{2,6}$'` per verificare che il formato dell'indirizzo di posta elettronica sia corretto;
- *password*: `'^(?=.*[A-Za-z])(?=.*\d)[A-Za-z\d]{8,}$'` per verificare che la password sia formata da almeno 8 caratteri, di cui almeno una lettera e un numero.



La validità delle suddette espressioni regolari non sarà verificata dal DBMS, ma soltanto dall'applicazione Android. Sarà quest'ultima, dunque, a dover garantire l'integrità dei dati che si vogliono salvare nel database.

Inoltre, la versione del server SQL utilizzata non supporta il tipo di dato *boolean*. Pertanto

- i campi booleani presenti nel dizionario dei dati saranno dichiarati come campi di tipo *bit*, con valore = 0 per FALSE e valore = 1 per TRUE;

Infine, si precisa che al fine di rispettare le convenzioni riguardanti le basi di dati, i nomi delle tabelle del DB saranno al plurale, sebbene nel dizionario dei dati siano riportati i nomi al singolare.

3.5 Controllo degli accessi e sicurezza

La seguente matrice degli accessi descrive i diritti di accesso di ciascun attore ai vari sottosistemi individuati.

Attore			
Sottosistema	Iscritto	Gestore	Utente
Gestione account	<ul style="list-style-type: none">• Login• Logout	<ul style="list-style-type: none">• Login• Logout	<ul style="list-style-type: none">• Registrazione iscritto
Gestione profili	<ul style="list-style-type: none">• Visualizzazione profilo personale		
Gestione contenuti	<ul style="list-style-type: none">• Visualizzazione home page• Visualizzazione dettagliata contenuti	<ul style="list-style-type: none">• Visualizzazione home page• Visualizzazione dettagliata contenuti• Ricerca contenuti	<ul style="list-style-type: none">• Visualizzazione home page• Visualizzazione dettagliata contenuti

	<ul style="list-style-type: none"> • Ricerca contenuti 		<ul style="list-style-type: none"> • Ricerca contenuti
Gestione liste	<ul style="list-style-type: none"> • Creazione liste • Aggiunta contenuti liste • Visualizzazione contenuti liste personali 		
Gestione recensioni	<ul style="list-style-type: none"> • Pubblicazione recensioni • Aggiunta segnalazioni alle recensioni • Visualizzazione recensioni 	<ul style="list-style-type: none"> • Visualizzazione recensioni 	<ul style="list-style-type: none"> • Visualizzazione recensioni
Gestione segnalazioni		<ul style="list-style-type: none"> • Visualizzazione segnalazioni • Cancellazione segnalazione • Rifiuto segnalazione 	

3.6 Controllo globale del software

L'applicazione iLike è un sistema interattivo, nel quale ogni funzionalità descritta può essere avviata tramite un'azione generata dall'utente tramite l'interfaccia grafica. Tale azione genera un evento che sarà gestito dal suo handler, il quale indirizzerà l'esecuzione al sottosistema che si occupa della logica di controllo e al gestore corrispondente. Questi ultimi, infine, si rivolgeranno ai servizi per la logica applicativa. Per questo motivo il sistema descritto utilizza un meccanismo di controllo del flusso di tipo *event-driven*.



3.7 Condizione limite

Di seguito sono descritte – tramite use case – le boundary condition relative all'avvio, allo spegnimento e al fallimento del sistema e ad errori di accesso ai dati persistenti.

Identificativo <i>UCBC_1</i>	Avvio del sistema	<i>Data</i>	02/11/2022
		<i>Vers.</i>	0.1
		<i>Autore</i>	<i>Francesco Giorgione</i>
Descrizione	Lo UC descrive l'avvio del sistema.		
Attore Principale	Amministratore		
Attori secondari	NA		
Entry Condition	L'amministratore vuole eseguire l'avvio del sistema. AND L'amministratore esegue il login sul server.		
Exit condition On success	Il sistema viene avviato correttamente.		
Exit condition On failure	Il sistema non viene avviato.		
FLUSSO DI EVENTI PRINCIPALE/MAIN SCENARIO			
1	Amministratore	Esegue il comando di avvio del sistema.	
2	Sistema:	Verifica che l'ultimo shutdown del sistema sia avvenuto senza errori.	
3	Sistema:	Esegue l'avvio e rende disponibili i suoi servizi agli utenti.	
I Scenario/Flusso di eventi Alternativo: l'ultimo shutdown del sistema è avvenuto in modo errato.			
3.a1	Sistema:	Comunica all'amministratore che, a causa di uno shutdown errato, i dati persistenti potrebbero essere non consistenti.	
3.a2	Amministratore:	Esegue un controllo di consistenza sul database e risolve eventuali inconsistenze.	
3.a3	Amministratore:	Esegue il passaggio 1.	



Identificativo <i>UCBC_2</i>	Crash del sistema	<i>Data</i>	02/11/2022
		<i>Vers.</i>	0.1
		<i>Autore</i>	<i>Luigina Costante</i>
Descrizione	Lo UC descrive l'avvio del sistema in caso di fallimento.		
Attore Principale	Amministratore		
Attori secondari	NA		
Entry Condition	Il sistema termina erroneamente.		
Exit condition On success	Il sistema viene riavviato correttamente.		
Exit condition On failure	Il sistema non viene riavviato.		
FLUSSO DI EVENTI PRINCIPALE/MAIN SCENARIO			
1	Amministratore	Include il comando UCBC_1.	

Identificativo <i>UCBC_3</i>	Arresto del sistema	Data	02/12/2022
		Vers.	0.1
		Autore	<i>Simona Lo Conte</i>
Descrizione	Lo UC descrive l'arresto del sistema.		
Attore Principale	Amministratore		
Attori secondari	NA		
Entry Condition	Il sistema è in esecuzione AND L'amministratore vuole effettuare l'arresto del sistema		
Exit condition On success	Il sistema viene spento correttamente		
Exit condition On failure	Il sistema non viene spento		



FLUSSO DI EVENTI PRINCIPALE/MAIN SCENARIO		
1	Amministratore:	Esegue il comando di arresto del sistema.
2	Sistema:	Controlla che non ci siano connessioni ancora aperte.
3	Sistema:	Termina l'esecuzione del sistema.
I Scenario/Flusso di eventi Alternativo: ci sono ancora connessioni aperte nel sistema		
3.a1	Sistema:	Notifica l'amministratore comunicandogli che ci sono ancora connessioni aperte.
3.a2	Sistema:	Dopo un'attesa, ricontrolla se ci sono ancora connessioni attive, e in caso contrario, prosegue con l'arresto del sistema.
3.a3	Sistema:	Comunica all'amministratore che l'arresto del sistema è avvenuto correttamente.

Identificativo <i>UCBC_4</i>	Accesso ai dati persistenti	<i>Data</i>	04/12/2022
		<i>Vers.</i>	0.1
		<i>Autore</i>	<i>Marta Napolillo</i>
Descrizione	Lo UC descrive il comportamento del sistema in caso di fallimento dovuto all'accesso ai dati persistenti.		
Attore Principale	Amministratore		
Attori secondari	NA		
Entry Condition	Il sistema è in esecuzione AND L'amministratore vuole effettuare l'accesso ai dati persistenti		
Exit condition On success	Il sistema accede correttamente ai dati persistenti		
Exit condition On failure	Il sistema non può accedere ai dati persistenti		
FLUSSO DI EVENTI PRINCIPALE/MAIN SCENARIO			
1	Amministratore:	Esegue il comando per accedere ai dati persistenti.	
2	Sistema:	Tenta di accedere ai dati persistenti richiesti.	
3	Sistema:	Notifica l'amministratore dell'impossibilità di accedere ai dati persistenti.	
4	Amministratore:	<i>Include UCBC_3.</i>	
5	Amministratore:	Ripristina l'accesso ai dati persistenti.	
6	Amministratore:	<i>Include UCBC_1.</i>	



4 Servizi dei sottosistemi

In questa sezione vengono descritti i servizi di ciascuno dei sottosistemi precedentemente elencati.

Sottosistema Gestione account

Servizio	Descrizione	Interfaccia
Login	Questo servizio consente ad un utente di autenticarsi all'interno del sistema tramite le proprie credenziali di accesso.	AccountInterface
Logout	Questo servizio consente ad un iscritto o ad un gestore di disconnettersi dal sistema.	AccountInterface
Registrazione iscritto	Questo servizio consente ad un utente di registrarsi al sistema in modo da poter usufruire di tutte le funzionalità offerte agli iscritti di iLike.	AccountInterface
Visualizzazione profilo personale	Questo servizio consente la visualizzazione del profilo dell'iscritto loggato.	AccountInterface

Sottosistema Gestione segnalazioni

Servizio	Descrizione	Interfaccia
Visualizzazione segnalazioni	Questo servizio consente al gestore di visualizzare l'elenco di segnalazioni presenti nel sistema.	SegnalazioneInterface
Rifiuto segnalazione	Questo servizio consente al gestore di rifiutare una segnalazione inviata da un iscritto in merito ad una recensione presente nel sistema.	SegnalazioneInterface
Cancellazione recensione	Questo servizio dovrà consentire al gestore di cancellare una recensione in seguito alla ricezione di una o più segnalazioni.	SegnalazioneInterface



Sottosistema Gestione recensioni

Servizio	Descrizione	Interfaccia
Pubblicazione recensioni	Questo servizio permette all'iscritto di poter aggiungere una recensione ad un contenuto.	RecensioneInterface
Visualizzazione recensioni	Questo servizio permette agli utenti di visualizzare l'elenco di recensioni relative ad un contenuto.	RecensioneInterface
Aggiunta segnalazioni alle recensioni	Questo servizio permette di inviare una segnalazione relativa ad una recensione.	RecensioneInterface

Sottosistema Gestione contenuti

Servizio	Descrizione	Interfaccia
Visualizzazione homepage	Questo servizio consente la visualizzazione non dettagliata di tre contenuti scelti casualmente all'interno del catalogo.	ContenutoInterface
Visualizzazione dettagliata contenuti	Questo servizio permette di visualizzare il contenuto insieme a tutti i suoi dettagli.	ContenutoInterface
Ricerca contenuti	Questo servizio permette la ricerca, da parte di un utente, di uno specifico contenuto sulla base del titolo.	ContenutoInterface

Sottosistema Gestione liste

Servizio	Descrizione	Interfaccia
Creazione liste	Questo servizio consente all'iscritto la creazione di una nuova lista.	ListaInterface



Aggiunta contenuti liste	Questo servizio permette di aggiungere un contenuto alla lista selezionata.	ListalInterface
Visualizzazione contenuti liste personali	Questo servizio permette la visualizzazione dei contenuti presenti nella lista dell'iscritto.	ListalInterface

5 Glossario

Sigla/Termine	Definizione
Categoria	Differenziazione di contenuti dello stesso tipo (es. horror e commedie per i film, jazz e pop per gli album musicali, ecc.).
Contenuto	Elemento appartenente all'insieme di film, serie TV, libri e album musicali.
Credenziali	Per il gestore sono l'indirizzo email e la password. Per l'iscritto sono l'indirizzo email (o in alternativa il nickname) e la password.
Star rating	Input type che consente di esprimere una valutazione indicata da stelle (un maggior numero di stelle indica una maggiore qualità).
Tipo	Film, serie TV, libro oppure album musicale.
Valutazione contenuto	Numero di stelle assegnate dal recensore al contenuto recensito.