

**SoundShelf**  
**Object Design Document**  
**Versione 2.0**



Data: 09/01/2025

Progetto: SoundShelf	Versione: 2.0
Documento: Object Design Document	Data: 09/01/2025

**Coordinatore del progetto:**

Nome	Matricola

**Partecipanti:**

Nome	Matricola
Marco Della Greca	0512116959
Luca Mastino	0512108160
Giuseppe Caiazzo	0512104992
Michele Quaglia	0512118399

Scritto da:	Intero Team
-------------	-------------

**Revision History**

Data	Versione	Descrizione	Autore
15/12/2024	1.0	Stesura delle interfacce delle classi	Intero Team
09/01/2025	2.0	Completamento delle parti mancanti dell'ODD e revisione delle interfacce delle classi	Caiazzo Giuseppe, Della Greca Marco e Quaglia Michele

Progetto: SoundShelf	Versione: 2.0
Documento: Object Design Document	Data: 09/01/2025

## Indice

1.	INTRODUZIONE .....	4
1.1.	Scelte di progettazione degli oggetti .....	4
1.1.1.	Complessità vs. Manutenibilità .....	4
1.1.2.	Riusabilità vs. Prestazioni .....	4
1.1.3.	Specializzazione vs. Generalizzazione .....	4
1.1.4.	Flessibilità vs. Sicurezza .....	4
1.1.5.	Incapsulamento vs. Accessibilità .....	5
1.1.6.	Modularità vs. Interdipendenza .....	5
1.2.	Linee guida per la documentazione delle interfacce .....	5
1.2.1.	Convenzioni di denominazione .....	5
1.2.2.	Trattamento dei casi limite e degli errori .....	5
1.2.3.	Struttura delle interfacce .....	6
1.2.4.	Meccanismi di gestione delle eccezioni .....	6
1.2.5.	Convenzioni per il codice .....	6
1.3.	Definizioni, acronimi e abbreviazioni .....	6
1.3.1.	Definizioni .....	6
1.3.2.	Acronimi .....	6
1.3.3.	Abbreviazioni .....	6
1.4.	Riferimenti .....	7
2.	PACCHETTI .....	7
2.1.	Descrizione dei pacchetti .....	7
2.2.	Diagramma delle dipendenze .....	8
3.	INTERFACCE .....	9
3.1.	Utente Registrato .....	9
3.2.	Cliente .....	10
3.3.	Rimborso .....	10
3.4.	Richiesta supporto .....	10
3.5.	Richiesta rimborso .....	11
3.6.	Catalogo Richiesta Rimborso .....	11
3.7.	Catalogo Richiesta Supporto .....	11
3.8.	Ordine .....	12
3.9.	Prodotto .....	13
3.10.	Carrello .....	13
3.11.	Catalogo Ordini .....	14
3.12.	Recensione .....	14
3.13.	Catalogo Recensioni .....	14
3.14.	Catalogo Prodotto .....	15
3.15.	Genere .....	15
3.16.	Artista .....	15
4.	GLOSSARIO .....	16

Progetto: SoundShelf	Versione: 2.0
Documento: Object Design Document	Data: 09/01/2025

## 1. INTRODUZIONE

### *1.1. Scelte di progettazione degli oggetti*

La progettazione orientata agli oggetti si basa su principi fondamentali come l'incapsulamento, l'ereditarietà e il polimorfismo. Tuttavia, l'applicazione di questi principi comporta una serie di compromessi che devono essere attentamente valutati durante il processo di sviluppo del software. Di seguito vengono analizzati alcuni dei principali trade-off della progettazione degli oggetti nel contesto del progetto SoundShelf.

#### **1.1.1. Complessità vs. Manutenibilità**

Un modello orientato agli oggetti complesso, con molte classi e relazioni, offre maggiore flessibilità e possibilità di riuso del codice. Tuttavia, può diventare difficile da comprendere e mantenere, specialmente in progetti con team di sviluppo numerosi.

Nel caso di SoundShelf:

- L'uso di classi separate per gestire utenti, prodotti, ordini e recensioni è essenziale per garantire la modularità.
- Il costo è un aumento della complessità iniziale, mitigato attraverso una documentazione dettagliata e diagrammi UML.

#### **1.1.2. Riusabilità vs. Prestazioni**

La progettazione di classi e metodi altamente riusabili è un obiettivo primario nell'ingegneria del software. Tuttavia, questo approccio potrebbe introdurre overhead computazionale e rallentare le prestazioni del sistema.

Esempio:

- La classe Prodotto include metodi generici per la gestione dei dettagli di tutti i tipi di dischi (vinile, CD, ecc.). Questo design aumenta la flessibilità, ma comporta un carico computazionale maggiore rispetto a metodi specifici per ciascun tipo di prodotto.

#### **1.1.3. Specializzazione vs. Generalizzazione**

Un design specializzato per una funzionalità specifica può essere più efficiente e facile da implementare. Al contrario, un design generalizzato può supportare futuri cambiamenti dei requisiti con minori modifiche al codice.

Nel contesto di SoundShelf:

- Il sottosistema Utente è stato generalizzato per includere classi come Guest, Cliente e GestoreSito, sfruttando l'ereditarietà per ridurre duplicazioni di codice.
- Ciò garantisce flessibilità, ma introduce una complessità maggiore nel garantire l'integrità dei dati.

#### **1.1.4. Flessibilità vs. Sicurezza**

L'uso di polimorfismo e dinamismo rende il sistema più flessibile e adattabile a nuove funzionalità. Tuttavia, ciò può esporre il sistema a vulnerabilità di sicurezza, come attacchi basati su input malevoli o accessi non autorizzati.

Nel progetto:

	Ingegneria del Software	Pagina 4 di 17
--	-------------------------	----------------

Progetto: SoundShelf	Versione: 2.0
Documento: Object Design Document	Data: 09/01/2025

- Sono stati implementati controlli rigorosi nei metodi di accesso alle classi sensibili (es. Ordine e Rimborso) per ridurre i rischi.
- Questo approccio bilancia flessibilità e sicurezza, con una penalizzazione minima in termini di prestazioni.

### 1.1.5. Incapsulamento vs. Accessibilità

L'incapsulamento protegge lo stato interno degli oggetti nascondendo i dettagli di implementazione. Tuttavia, può limitare l'accesso diretto a dati o metodi, rallentando lo sviluppo.

Scelta adottata:

- I dati sensibili, come le credenziali degli utenti, sono stati incapsulati nelle classi corrispondenti e accessibili solo tramite metodi specifici.
- Per contro, sono stati aggiunti metodi getter e setter, aumentando il numero di interfacce pubbliche, ma mantenendo la sicurezza.

### 1.1.6. Modularità vs. Interdipendenza

Una progettazione modulare consente di separare le responsabilità, ma potrebbe richiedere meccanismi complessi di interazione tra i moduli.

Nel sistema SoundShelf:

- Il sottosistema Recensioni comunica con Utente e Prodotto per garantire coerenza nei dati. Questo design richiede una gestione accurata delle dipendenze per evitare cicli e problemi di sincronizzazione.

## 1.2. Linee guida per la documentazione delle interfacce

Di seguito vengono illustrate le principali linee guida adottate.

### 1.2.1. Convenzioni di denominazione

Le convenzioni di denominazione sono state definite per rendere immediatamente comprensibile il ruolo di classi, metodi, attributi e parametri:

- **Classi:** Devono essere nominate utilizzando sostantivi singolari che descrivano chiaramente la loro funzione. Esempio: Product, Order, User.
- **Metodi:** I metodi devono avere nomi che iniziano con un verbo che descriva l'azione eseguita. Esempio: calculateTotal(), addProduct().
- **Attributi e parametri:** Devono essere denominati con sostantivi che riflettano chiaramente il dato rappresentato. Esempio: productName, orderDate, emailAddress.

### 1.2.2. Trattamento dei casi limite e degli errori

Una gestione coerente degli errori è fondamentale per garantire la stabilità e la prevedibilità del sistema:

- Gli errori devono essere comunicati tramite **eccezioni** e non tramite valori di ritorno, per evitare ambiguità nei dati restituiti.
- Le eccezioni devono essere ben documentate e specifiche. Esempio: InvalidProductException, OrderNotFoundException.
- I metodi devono prevedere la gestione dei casi limite. Esempio: un metodo che restituisce una lista di prodotti deve restituire una lista vuota, non null, se non ci sono prodotti disponibili.

Progetto: SoundShelf	Versione: 2.0
Documento: Object Design Document	Data: 09/01/2025

### 1.2.3. Struttura delle interfacce

Per garantire la coerenza e la manutenibilità del codice, le interfacce devono seguire una struttura uniforme:

- **Robustezza delle enumerazioni:** Le enumerazioni restituite devono gestire in modo sicuro la modifica degli elementi, come aggiunte o rimozioni, durante l'iterazione.

### 1.2.4. Meccanismi di gestione delle eccezioni

Per garantire la leggibilità e la prevedibilità del codice:

- Tutti i metodi pubblici devono documentare le eccezioni che possono essere generate.
- Devono essere adottati meccanismi centralizzati per il logging degli errori, con l'obiettivo di monitorare e diagnosticare facilmente eventuali problemi in fase di esecuzione.

### 1.2.5. Convenzioni per il codice

Oltre alla documentazione delle interfacce, sono stati definiti standard di codifica che promuovono leggibilità e manutenibilità:

- **Indentazione e spaziatura:** Seguono uno stile uniforme.
- **Commenti:** Sebbene non siano utilizzati direttamente per le specifiche tecniche (come richiesto dall'utente), la documentazione esterna deve essere completa e descrittiva.
- **Rispettare la Single Responsibility Principle:** Ogni classe e metodo deve avere una responsabilità chiara e definita.

## 1.3. Definizioni, acronimi e abbreviazioni

In questa sezione vengono fornite le definizioni, gli acronimi e le abbreviazioni utilizzate nel documento per garantire una comprensione uniforme da parte di tutti i lettori.

### 1.3.1. Definizioni

- **Oggetto:** Un'entità del software che combina dati e comportamenti. Esempio: un oggetto Product rappresenta un disco in vendita.
- **Interfaccia:** Una specifica che definisce i metodi pubblici di una classe o di un modulo, consentendo la comunicazione con altri componenti.
- **Eccezione:** Una condizione di errore o evento imprevisto che interrompe il flusso normale del programma.

### 1.3.2. Acronimi

- **ODD:** Object Design Document (Documento di Progettazione degli Oggetti).
- **API:** Application Programming Interface (Interfaccia di Programmazione delle Applicazioni).
- **DBMS:** Database Management System (Sistema di Gestione del Database).

### 1.3.3. Abbreviazioni

- **UI:** User Interface (Interfaccia Utente).
- **UX:** User Experience (Esperienza Utente).
- **ID:** Identifier (Identificatore).

Progetto: SoundShelf	Versione: 2.0
Documento: Object Design Document	Data: 09/01/2025

### ***1.4. Riferimenti***

Tutti i riferimenti riguardano i documenti di SDD e RAD.

## **2. PACCHETTI**

Ogni pacchetto rappresenta un'unità logica di funzionalità, progettata per garantire modularità, riusabilità e facilità di manutenzione. Sono elencati i dettagli di ciascun pacchetto, incluse le sue dipendenze, la sua struttura e il suo utilizzo previsto.

La struttura del progetto si basa sui seguenti pacchetti principali:

- **Control**
- **Ordini**
- **Prodotti**
- **Recensione**
- **Rimborsi**
- **Supporto**
- **Utente**
- **Util**

### ***2.1. Descrizione dei pacchetti***

#### **1. Control**

- **Scopo:** Contiene le servlet di controllo responsabili della gestione della homepage, incluse le funzionalità di caricamento dei prodotti da visualizzare.
- **Componenti principali:**
  - Servlet per la gestione della Home.
- **Uso previsto:** Fornire il punto di ingresso principale per l'applicazione, visualizzando i prodotti sulla homepage.

#### **2. Ordini**

- **Scopo:** Gestire la logica relativa agli ordini, compreso il carrello, la finalizzazione degli acquisti e la gestione dello stato degli ordini.
- **Componenti principali:**
  - Entity: Ordini, ElementoOrdine, Cart, CartItem.
  - DAO: Per la gestione dei dati degli ordini e degli elementi degli ordini.
  - Enumeration: StatoOrdine.
  - Servlet: Per la gestione del carrello e degli ordini.
- **Uso previsto:** Consentire agli utenti di effettuare ordini e monitorarne lo stato, e al gestore di mantenerne il controllo.

#### **3. Prodotti**

- **Scopo:** Fornire tutte le funzionalità per gestire i prodotti disponibili sul sito, inclusi artisti e generi.
- **Componenti principali:**
  - Entity: Prodotto, Artista, Genere.
  - DAO: per la gestione dei dati relativi ai prodotti.
- **Uso previsto:** Gestione dell'inventario dei prodotti e delle relative informazioni, con possibilità di aggiunta, rimozione, ricerca, e visualizzazione.

#### **4. Recensione**

	Ingegneria del Software	Pagina 7 di 17
--	-------------------------	----------------

Progetto: SoundShelf	Versione: 2.0
Documento: Object Design Document	Data: 09/01/2025

- **Scopo:** Gestire le recensioni dei prodotti da parte degli utenti.
- **Componenti principali:**
  - Entity e DAO per le recensioni.
- **Uso previsto:** Consentire agli utenti di lasciare recensioni sui prodotti acquistati, e ai gestori di rimuoverle se necessario.

#### 5. Rimborsi

- **Scopo:** Gestire le richieste di rimborso per i prodotti acquistati.
- **Componenti principali:**
  - Entity e DAO per le richieste di rimborso e i rimborsi stessi.
  - Enumeration: StatoRimborso.
- **Uso previsto:** Consentire agli utenti di richiedere rimborsi per i prodotti acquistati, ai gestori di controllarle e accettarle/rifiutarle.

#### 6. Supporto

- **Scopo:** Gestire le richieste di supporto da parte degli utenti.
- **Componenti principali:**
  - Entity e DAO per le richieste di supporto.
  - Enumeration: StatoSupporto.
- **Uso previsto:** Offrire un sistema di supporto agli utenti per risolvere problemi o ricevere assistenza, consentendo agli utenti di inviare richieste e ai gestori di controllarle, e una funzionalità di scambio informazioni tramite risposte e domande.

#### 7. Utente

- **Scopo:** Gestire tutte le funzionalità relative agli utenti, incluse registrazione, login e gestione dei profili.
- **Componenti principali:**
  - Entity: per rappresentare gli utenti.
  - DAO: per la gestione dei dati utente.
  - Enumeration: per i ruoli utente (ad esempio, Gestore Sito, Cliente).
- **Uso previsto:** Fornire la gestione degli utenti e autenticare gli accessi al sistema.

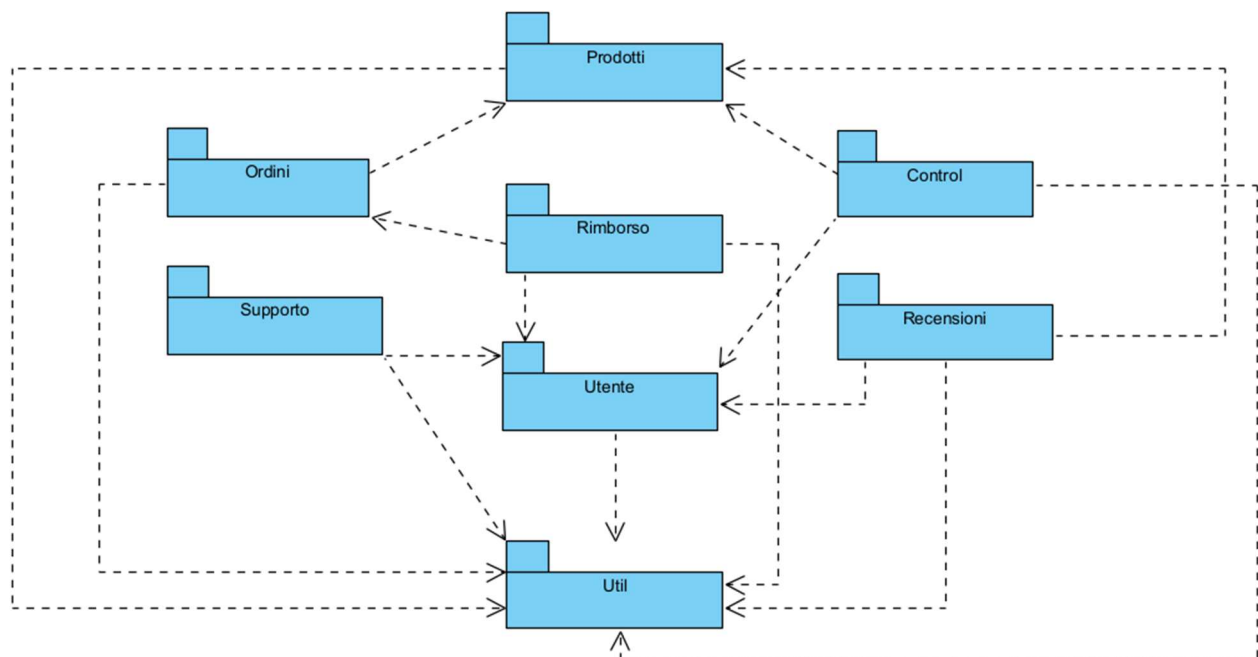
#### 8. Util

- **Scopo:** Fornire classi di utilità per funzionalità comuni.
- **Componenti principali:**
  - Gestione del DataSource per il database.
  - Sanitizzazione dell'input per garantire la sicurezza e la validità dei dati.
- **Uso previsto:** Fornire supporto generale e strumenti utili per altre componenti del sistema.

### 2.2. Diagramma delle dipendenze

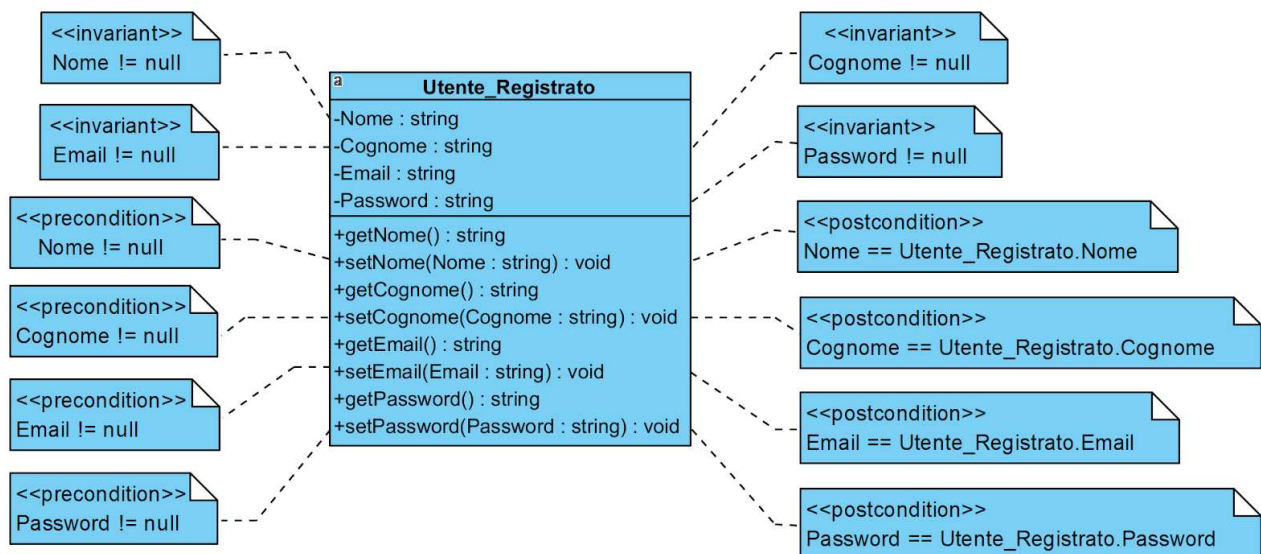


Progetto: SoundShelf	Versione: 2.0
Documento: Object Design Document	Data: 09/01/2025



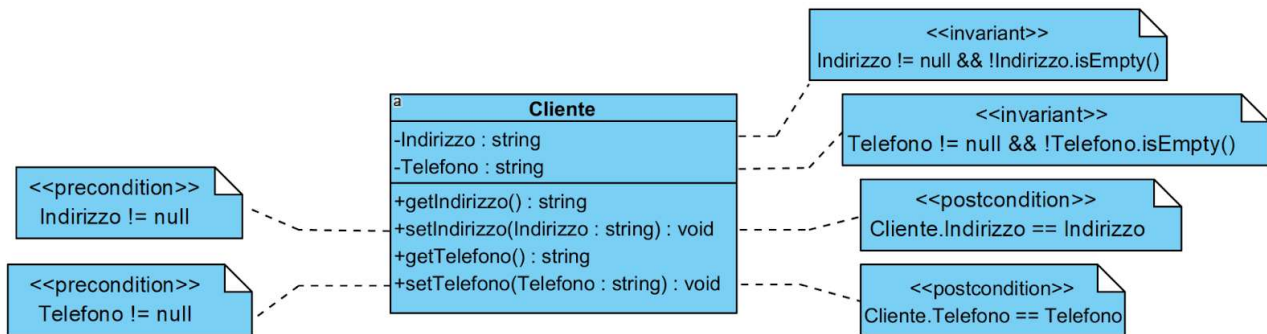
### 3. INTERFACCE

#### 3.1. Utente Registrato

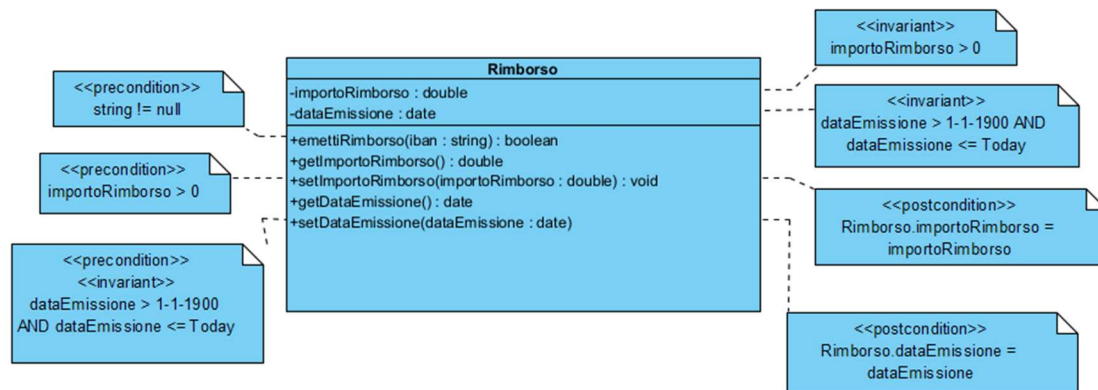


Progetto: SoundShelf	Versione: 2.0
Documento: Object Design Document	Data: 09/01/2025

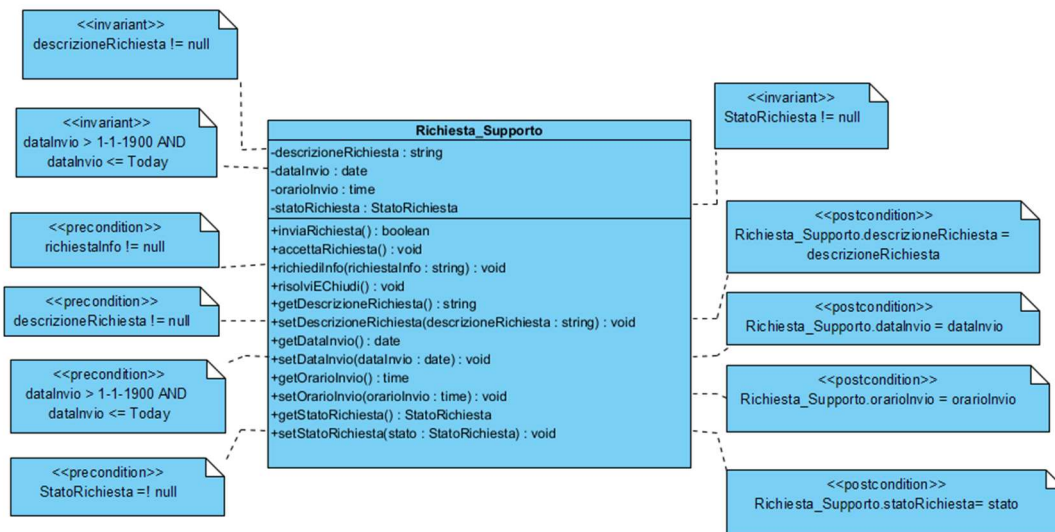
### 3.2. Cliente



### 3.3. Rimborso

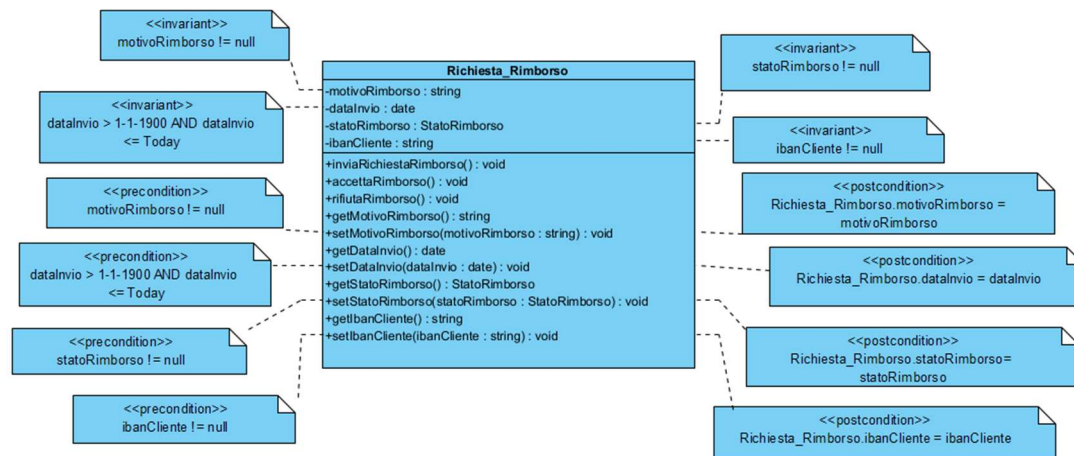


### 3.4. Richiesta supporto



Progetto: SoundShelf	Versione: 2.0
Documento: Object Design Document	Data: 09/01/2025

### 3.5. Richiesta rimborso



### 3.6. Catalogo Richiesta Rimborso

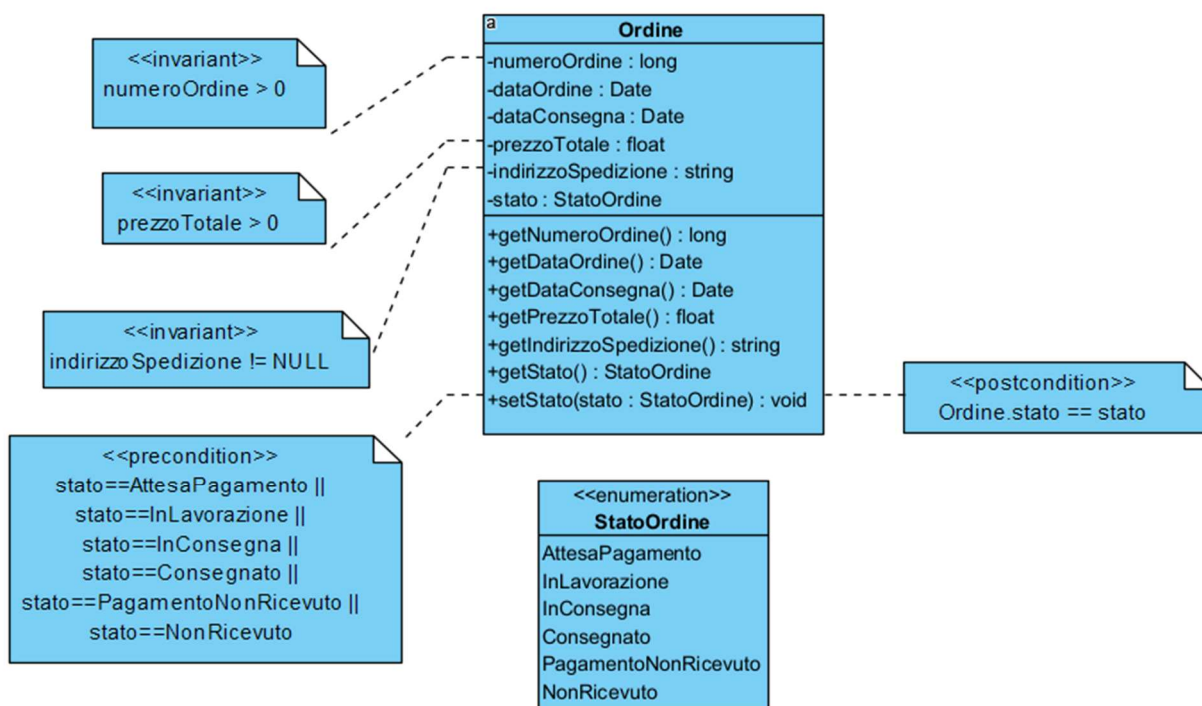


### 3.7. Catalogo Richiesta Supporto



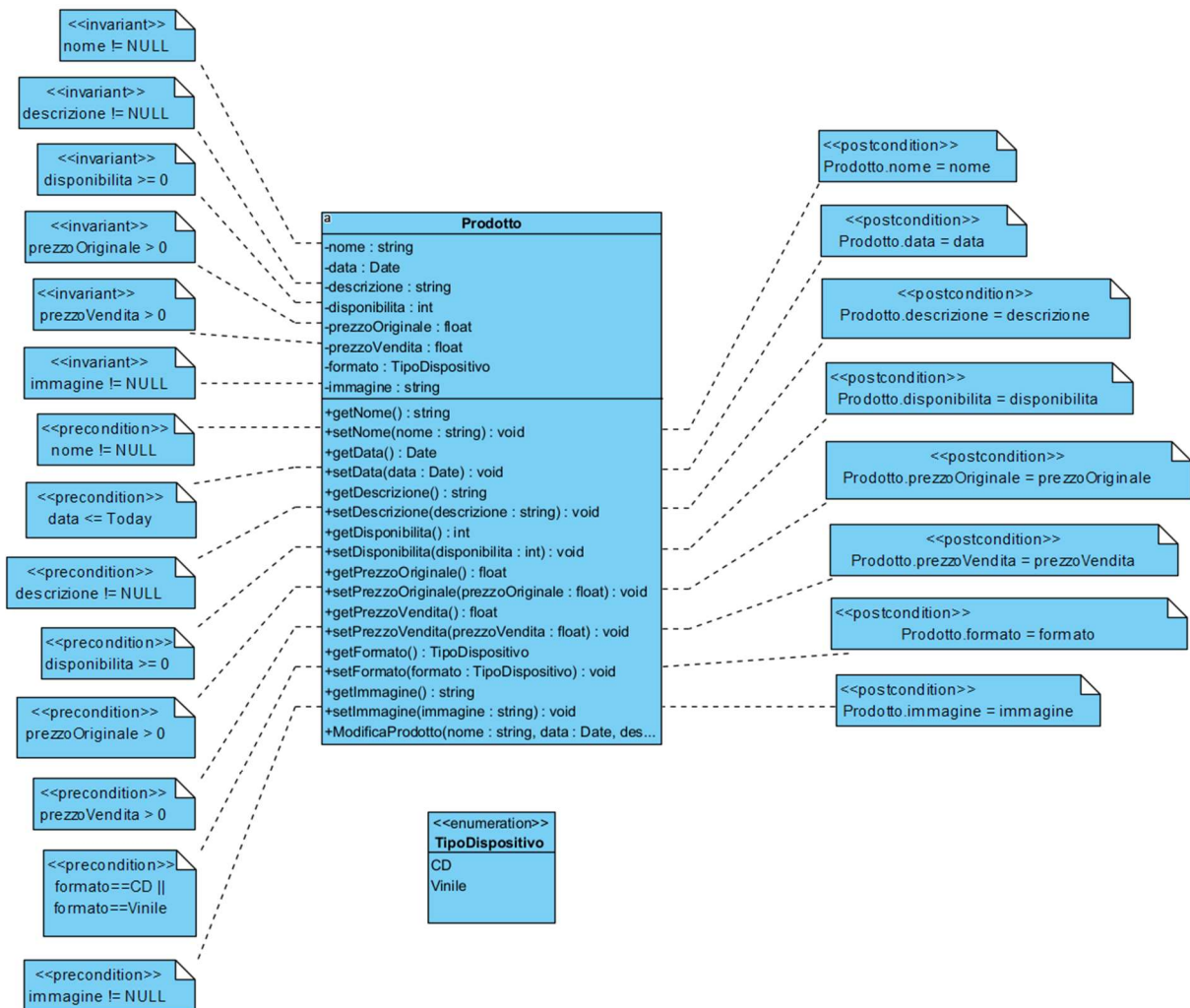
Progetto: SoundShelf	Versione: 2.0
Documento: Object Design Document	Data: 09/01/2025

### 3.8. Ordine



Progetto: SoundShelf	Versione: 2.0
Documento: Object Design Document	Data: 09/01/2025

### 3.9. Prodotto



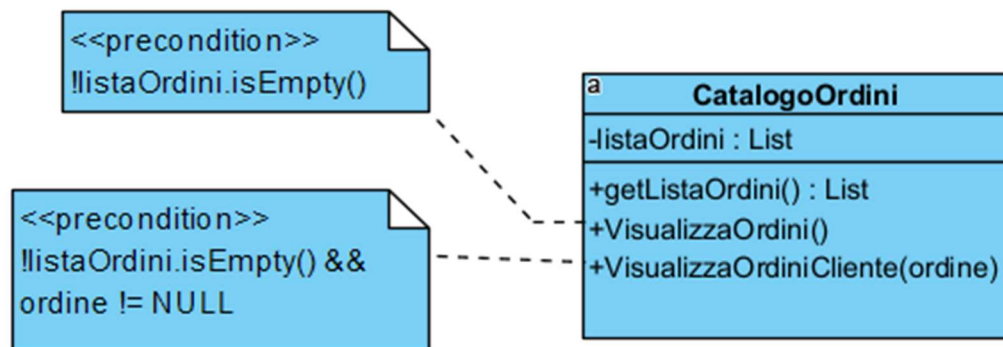
### 3.10. Carrello



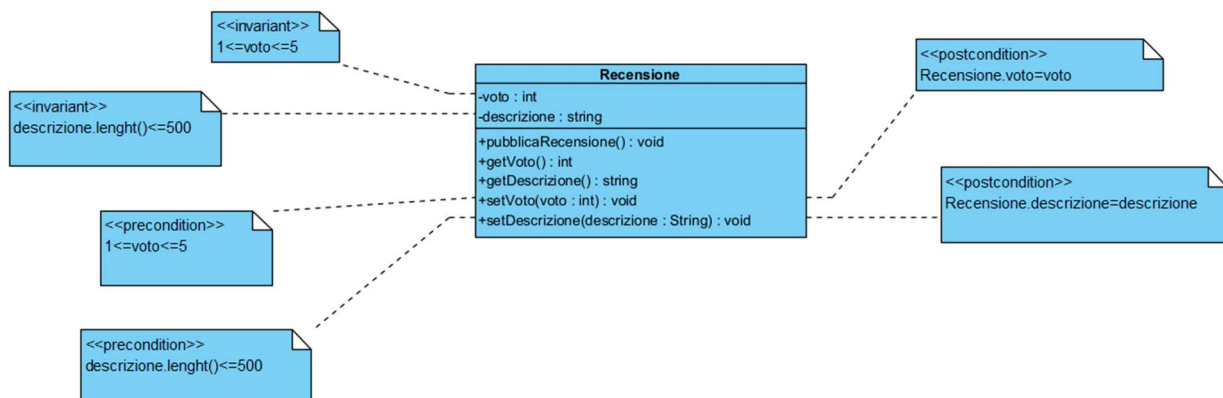


Progetto: SoundShelf	Versione: 2.0
Documento: Object Design Document	Data: 09/01/2025

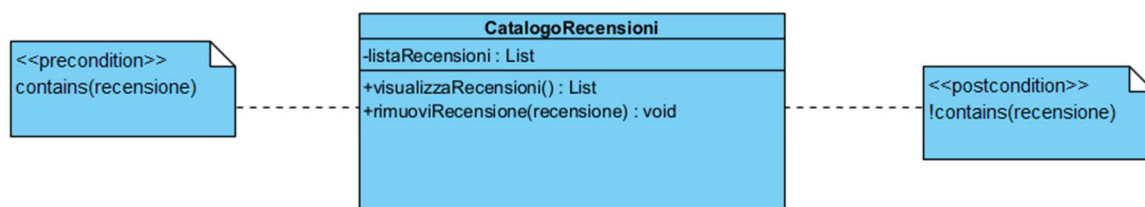
### 3.11. *Catalogo Ordini*



### 3.12. *Recensione*

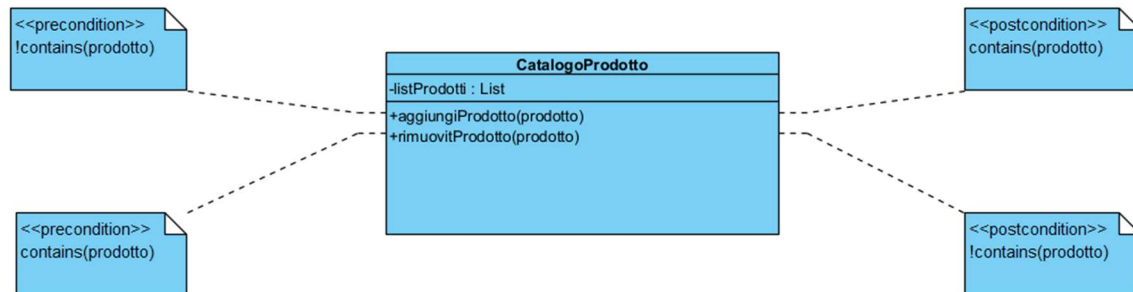


### 3.13. *Catalogo Recensioni*

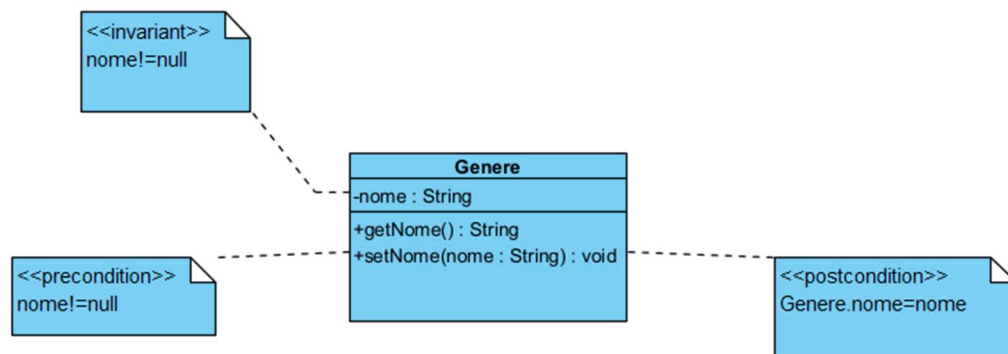


Progetto: SoundShelf	Versione: 2.0
Documento: Object Design Document	Data: 09/01/2025

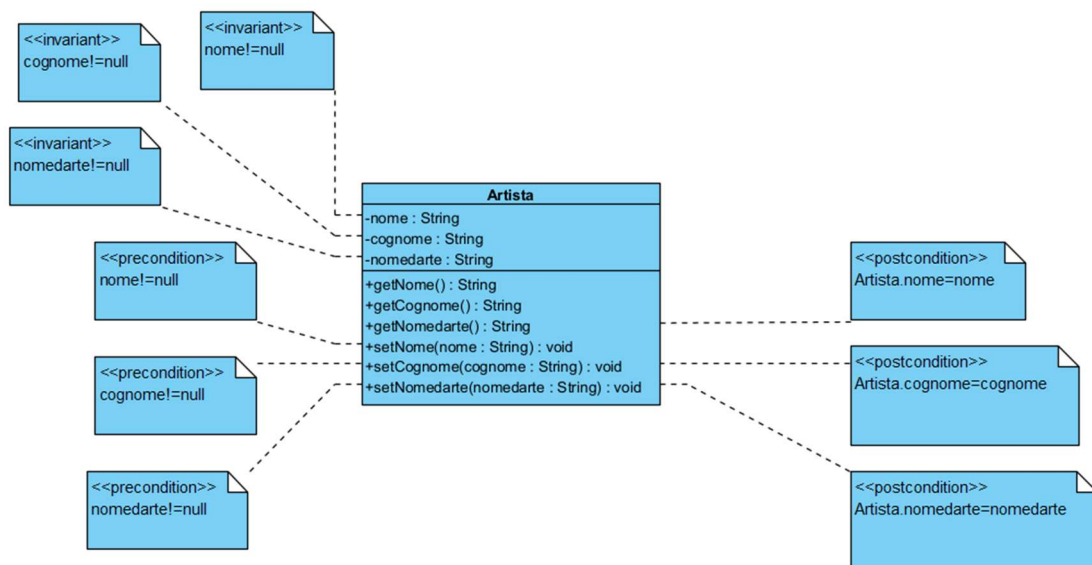
### 3.14. *Catalogo Prodotto*



### 3.15. *Genere*



### 3.16. *Artista*



Progetto: SoundShelf	Versione: 2.0
Documento: Object Design Document	Data: 09/01/2025

## 4. GLOSSARIO

Nome	Definizione
<b>Carrello</b>	Sezione del sito in cui gli utenti possono aggiungere i prodotti che intendono acquistare. Il carrello consente agli utenti di visualizzare e modificare gli articoli prima di procedere al checkout.
<b>Checkout</b>	Il processo finale in cui l'utente fornisce i dettagli di spedizione per completare l'acquisto di uno o più prodotti.
<b>Prodotto</b>	Un vinile, CD, o altro articolo musicale messo in vendita su <i>Soundshelf</i> .
<b>Ordine</b>	Un acquisto completato da un utente, contenente uno o più prodotti. Ogni ordine ha un identificativo unico e include dettagli come lo stato della spedizione.
<b>Recensione</b>	Un commento lasciato dagli utenti per esprimere un giudizio su un prodotto acquistato. Le recensioni includono una valutazione in stelle (da 1 a 5) e un testo con il feedback dell'utente.
<b>Rimborso</b>	Restituzione di denaro a un utente per un prodotto che è stato restituito per un ordine che è stato cancellato.
<b>Supporto</b>	Un servizio clienti fornito tramite il sito, che permette agli utenti di inviare richieste di assistenza riguardo ordini, pagamenti, o altri problemi.
<b>Inventario</b>	La gestione e il monitoraggio delle quantità di prodotti disponibili nel magazzino di <i>Soundshelf</i> .
<b>Vinile</b>	Un formato di disco in vinile, tipicamente usato per la riproduzione di musica in formato analogico. I vinili sono uno dei principali prodotti venduti su <i>Soundshelf</i> .
<b>Formato</b>	Il tipo di supporto fisico o digitale di un prodotto musicale, che può includere vinile, CD, o digitale. Ogni prodotto in catalogo ha un formato associato.
<b>Artista</b>	Il creatore musicale dietro a un prodotto (album, vinile, ecc.). L'artista può essere un singolo individuo o un gruppo musicale.
<b>Stato ordine</b>	Indica la fase in cui si trova un ordine.
<b>Tipo di utente</b>	La classificazione di un utente in base al suo livello di accesso e interazione con il sito. Gli utenti possono essere non registrati, registrati, o gestori sito.
<b>Utente</b>	Una persona che interagisce con il sito, che sia registrata o non registrata. Gli utenti possono navigare il catalogo, fare acquisti, lasciare recensioni e altro.
<b>Quantità disponibile</b>	La quantità di un prodotto che è attualmente in stock e pronta per essere venduta.
<b>Data di spedizione</b>	La data in cui un ordine viene effettivamente inviato dal magazzino per la consegna al cliente.



Progetto: SoundShelf	Versione: 2.0
Documento: Object Design Document	Data: 09/01/2025

<b>Data di consegna</b>	La data prevista per la consegna di un ordine, indicata dal metodo di spedizione scelto.
<b>Recensione Prodotto</b>	Un'analisi o una valutazione di un prodotto scritta da un utente che ha acquistato l'articolo. Le recensioni includono il punteggio e commenti.
<b>Data di richiesta rimborso</b>	La data in cui un utente invia una richiesta per il rimborso di un ordine o di un prodotto.
<b>Rimborso approvato</b>	Stato di una richiesta di rimborso che è stata esaminata e approvata dal sistema o dal team di supporto, indicando che l'utente riceverà il rimborso.
<b>Rimborso rifiutato</b>	Stato di una richiesta di rimborso che è stata esaminata e rifiutata, indicando che l'utente non riceverà il rimborso per l'ordine o il prodotto.
<b>Data di chiusura supporto</b>	La data in cui una richiesta di supporto viene risolta o chiusa dopo essere stata esaminata e trattata.
<b>Supporto risolto</b>	Stato di una richiesta di supporto che è stata risolta e per la quale non è più necessario alcun intervento.
<b>Supporto in lavorazione</b>	Stato di una richiesta di supporto che è attualmente in fase di esame o risoluzione da parte del team di assistenza clienti.