Università degli Studi di Salerno
Corso di Ingegneria del Software

SoundShelf
System Design Document
Versione 1.0

Data: 24/11/2024

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
24/11/2024	1.0	Stesura System Design Document	Intero Team

Indice

1. INTRODUZIONE

1.1. Purpose of the system

Il sistema è progettato per offrire una piattaforma intuitiva e affidabile per la vendita di dischi musicali. Lo scopo è garantire agli utenti un'esperienza di acquisto fluida, fornendo informazioni dettagliate sui prodotti e strumenti per facilitare la scelta e l'acquisto.

1.2. Design goals

RANK	ID	DESCRIZIONE	CATEGORIA	RNF DI ORIGINE
5	DG1	il sistema deve garantire un tempo di risposta inferiore ai 3 secondi e gestire almeno 500 utenti	PERFORMANCE	NFR 6
1	DG2	Il sistema deve fornire un messaggio di errore in caso di input non corretto scritto da parte dell'utente.	DEPENDABILITY	NFR 5
2	DG3	il sistema deve avere un'interfaccia grafica che permetta all'utente di non avere difficoltà a comprenderne il funzionamento, inoltre deve consentire all'utente di capire quali sono i dati obbligatori da compilare in fase di registrazione	END USER	NFR1, NFR2
4	DG4	Il sistema deve garantire una buona manutenibilità, consentendo modifiche e aggiornamenti senza impatti significativi sul funzionamento esistente	MAINTENANCE	NFR7
3	DG5	L'obiettivo di progettazione legato ai costi mira a mantenere un equilibrio tra la qualità del sistema e la sua sostenibilità economica. Il design deve essere orientato a minimizzare i costi di sviluppo, implementazione e manutenzione senza compromettere le prestazioni	COST	-

1.3. Definitions, acronyms, and abbreviations

- **Dischi fisici**: CD, vinili e altri supporti materiali
- **Checkout**: Procedura per completare un acquisto online.
- **Backend**: Parte del sistema che gestisce i dati e le operazioni.
- **Frontend**: Interfaccia visibile agli utenti per la navigazione e gli acquisti

1.4. References

I riferimenti sono stati presi dal Requirements Analysis Document prodotto in precedenza.

1.5. Overview

Il documento fornisce una panoramica del progetto, descrivendo l'architettura del sito e i requisiti di progettazione. Successivamente, saranno presentati i dettagli relativi ai sottosistemi principali del sistema.

2. CURRENT SOFTWARE ARCHITECTURE

Le piattaforme di e-commerce moderne tipicamente seguono un'architettura **client-server**

2.1. Caratteristiche tipiche e limitazioni

- **Caratteristiche**:
  - **Gestione dell'inventario**: Monitoraggio automatico delle scorte e aggiornamento in tempo reale.
  - **Elaborazione dei pagamenti**: Integrazione con gateway di pagamento sicuri
- **Limitazioni**:
  - **Scalabilità limitata**: Alcuni sistemi incontrano difficoltà a gestire un traffico elevato senza un'architettura a sottosistemi
  - **Esperienza utente discontinua**: Interfacce meno responsive o non ottimizzate per dispositivi mobili.
  - **Sicurezza**: Protezione insufficiente contro attacchi come **SQL Injection**.

2.2. Ipotesi e problemi comuni

- **Ipotesi**: La maggior parte degli utenti utilizza dispositivi mobili per navigare e acquistare; quindi, è fondamentale garantire un design responsivo.
- **Problemi comuni**:
  - **Tempi di risposta**: Ritardi durante l'accesso a grandi database o la gestione di un numero elevato di richieste.
  - **Manutenzione e aggiornamenti**: Architetture monolitiche tendono a richiedere aggiornamenti complessi.
  - **Sicurezza**: La protezione dei dati sensibili degli utenti è spesso inadeguata in sistemi progettati senza considerare gli standard di sicurezza moderni.

2.3. Considerazioni per il nuovo sistema

Il nuovo sistema affronterà queste sfide implementando un'architettura modulare e scalabile basata su tecnologie moderne. La sicurezza sarà una priorità con l'uso di crittografia e protocolli di autenticazione robusti. Inoltre, l'interfaccia utente sarà progettata per essere completamente responsiva, migliorando l'esperienza utente su dispositivi mobili e desktop.

3. PROPOSED SOFTWARE ARCHITECTURE

La struttura del sistema è organizzata in tre livelli principali:

- **Presentation Layer**:
  Questo livello si occupa dell'interfaccia utente e gestisce l'interazione tra l'utente e il sistema.
- **Application Layer**:
  Qui si trova la logica applicativa principale, che coordina l'elaborazione dei dati tra il livello Presentation e il Data Layer.
- **Data Layer**:
  Questo livello gestisce e fornisce accesso ai dati persistenti, con ogni sottosistema associato a un database specifico.

Questa struttura garantisce un'elevata manutenibilità, una facile scalabilità e una chiara separazione delle responsabilità tra i componenti

3.1. Overview

Presentation Layer

- **ProdottiInterface**: Gestisce l'interazione relativa ai prodotti.
- **UtenteInterface**: Responsabile della gestione degli utenti.
- **OrdiniInterface**: Si occupa dell'interazione con gli ordini.
- **RimborsiInterface**: Consente agli utenti di richiedere rimborsi.
- **RecensioniInterface**: Gestisce le recensioni e il feedback degli utenti.
- **SupportoInterface**: Fornisce accesso ai servizi di supporto.

Application Layer

- **ProdottiControl**: gestisce i dati relativi ai prodotti
- **UtenteControl**: gestisce gli utenti e le loro informazioni
- **OrdiniControl**: gestisce gli ordini
- **RimborsiControl**: gestisce le richieste di rimborso
- **RecensioniControl**: gestisce le recensioni degli utenti
- **SupportoControl**: coordina i servizi di supporto

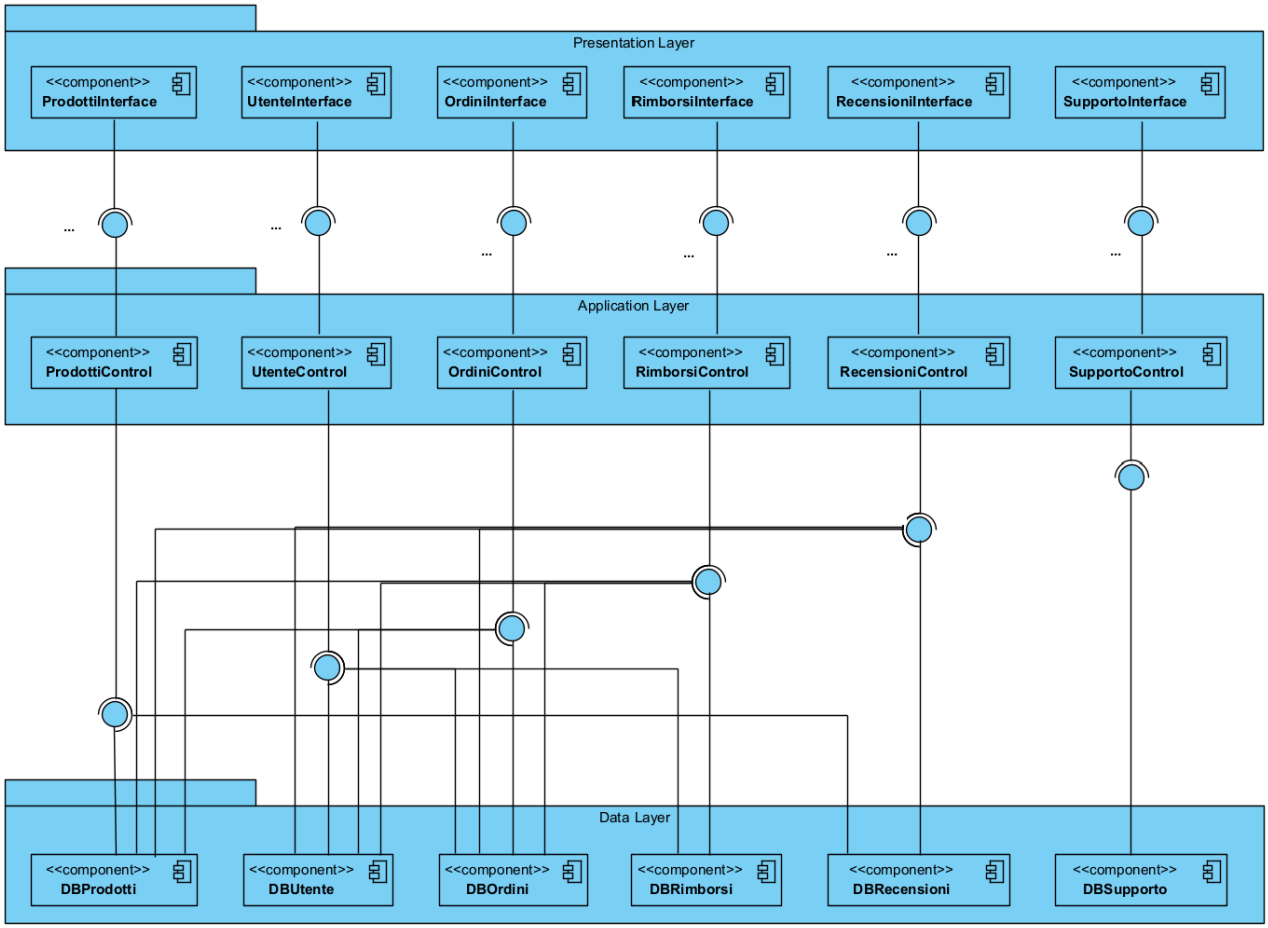
Data Layer

- **DBProdotti**: Gestisce i dati relativi ai prodotti.
- **DBUtente**: Conserva le informazioni sugli utenti.
- **DBOrdini**: Gestisce i dati relativi agli ordini.
- **DBRimborsi**: Contiene le informazioni sulle richieste di rimborso.
- **DBRecensioni**: Conserva le recensioni degli utenti.
- **DBSupporto**: Archivia i dati relativi al supporto clienti.

3.2.Subsystem decomposition

3.2.1. Descrizione dei sottosistemi e responsabilità

La decomposizione è rappresentata attraverso il seguente diagramma UML.



Di seguito vi è una spiegazione più dettagliata di ogni sottosistema, con relative interazioni con gli altri sottosistemi.

Presentation Layer

I componenti di questo livello sono interfacce dedicate a diverse funzionalità dell'applicazione.

- **ProdottiInterface**: Gestisce l'interazione relativa ai prodotti.
- **UtenteInterface**: Responsabile della gestione degli utenti.
- **OrdiniInterface**: Si occupa dell'interazione con gli ordini.
- **RimborsiInterface**: Consente agli utenti di richiedere rimborsi.
- **RecensioniInterface**: Gestisce le recensioni e il feedback degli utenti.
- **SupportoInterface**: Fornisce accesso ai servizi di supporto.

Application Layer

I componenti principali di questo livello elaborano la logica del sistema, interagendo sia con il Presentation Layer che con il Data Layer.

- **ProdottiControl**
  - *Responsabilità*: gestione dei dati relativi ai prodotti.
  - Interazione con: DBProdotti, DBRecensioni.

- **UtenteControl**
  - Responsabilità: gestione degli utenti e delle loro informazioni.
  - Interazione con: DBUtente, DBOrdini, DBRimborsi.

- **OrdiniControl**
  - Responsabilità: gestione degli ordini e dei dati correlati.
  - Interazione con: DBOrdini, DBProdotti, DBUtente.

- **RimborsiControl**
  - Responsabilità: gestione delle richieste di rimborso.
  - Interazione con: DBRimborsi, DBOrdini, DBUtente, DBProdotti.

- **RecensioniControl**
  - Responsabilità: gestione delle recensioni degli utenti.
  - Interazione con: DBRecensioni, DBProdotti, DBUtente, DBOrdini.

- **SupportoControl**
  - *Responsabilità: coordinamento dei servizi di supporto.*
  - *Interazione* con: DBSupporto.

Data Layer

I componenti del livello Data Layer forniscono accesso ai dati e garantiscono la persistenza delle informazioni.

- **DBProdotti**: Gestisce i dati relativi ai prodotti.
- **DBUtente**: Conserva le informazioni sugli utenti.
- **DBOrdini**: Gestisce i dati relativi agli ordini.
- **DBRimborsi**: Contiene le informazioni sulle richieste di rimborso.
- **DBRecensioni**: Conserva le recensioni degli utenti.
- **DBSupporto**: Archivia i dati relativi al supporto clienti.

3.3. Hardware/Software mapping

La progettazione del sistema di e-commerce *Soundshelf* si basa su un'infrastruttura scalabile e distribuita che garantisce affidabilità, prestazioni elevate e sicurezza. Il sistema è organizzato in tre livelli principali: **front-end**, **back-end** e **database**. Questi livelli interagiscono tra loro attraverso protocolli standard come HTTPS, supportando le funzionalità richieste dai sottosistemi. I componenti si suddividono in:

1. **Front-end (Interfaccia Utente)**
   Accessibile tramite dispositivi client (desktop, tablet, smartphone).
   Utilizzo di HTML, CSS e JavaScript, per un'interfaccia utente interattiva e responsiva.
   Gestisce l'interazione con gli utenti, fornendo un'esperienza fluida per l'acquisto, la gestione degli ordini e la visualizzazione del catalogo.

2. **Back-end (Logica Applicativa)**
   Eseguito su un server applicativo basato su Linux con risorse scalabili nel cloud.
   Supporta le funzionalità di business logic e comunica con il database e servizi esterni.
   Implementa i sottosistemi principali come la gestione degli ordini, del catalogo e delle richieste di supporto.

3. **Database**
   Server dedicato per la gestione dei dati, ospitato su un'infrastruttura cloud.
   MySQL come sistema di gestione del database relazionale.
   Memorizza in modo sicuro i dati persistenti, come utenti, ordini, prodotti e recensioni.

Ogni componente software viene distribuito su specifici elementi hardware per ottimizzare le prestazioni del sistema:

Sottosistema	**Hardware**	**Software (Off-the-Shelf)**	**Descrizione**
Interfaccia Utente	Dispositivi client	Browser (Chrome, Firefox, Safari)	Implementato con HTML5, CSS3 e JavaScript per l'interazione utente.
Gestione Ordini	Server applicativo	Tomcat	Implementa la logica di gestione degli ordini, come la creazione e il tracking.
Gestione Utenti	Server applicativo	Tomcat	Fornisce autenticazione, registrazione e gestione profili.
Gestione Catalogo	Server applicativo	Tomcat	Permette al gestore di aggiornare i prodotti disponibili.

Gestione Supporto	Server applicativo	Tomcat	Gestisce le richieste di assistenza degli utenti tramite form o API.
Gestione Recensioni	Server applicativo	Tomcat	Consente agli utenti di scrivere, modificare e moderare recensioni sui prodotti.
Gestione Rimborso	Server applicativo	Tomcat	Gestisce le richieste di rimborso, inclusi gli stati approvato/rifiutato.
Database relazionale	Server database	MySQL	Memorizza dati strutturati: utenti, ordini, prodotti, recensioni e rimborsi.
Servizio E-mail	Server SMTP (Cloud-based)	Gmail API	Invia notifiche di ordini, rimborsi e aggiornamenti tramite e-mail.

3.3.1. Mapping degli oggetti

Oggetto	Hardware	Software	Descrizione
Utente	Server Applicativo	Database (MySQL)	Memorizzato nel database con dati personali e credenziali criptate.
Ordine	Server applicativo	Database (MySQL)	Gestito con relazioni a Utente e Prodotto.
Prodotto	Server applicativo	Database (MySQL)	Memorizzato con dettagli e disponibilità nel catalogo.
Carrello	Server applicativo	Sessione	Dati temporanei, cancellati al termine della sessione.
RichiestaSupporto	Server applicativo	Database (MySQL)	Associato a Utente, include dettagli e stato della richiesta.
Recensione	Server applicativo	Database (MySQL)	Moderata e memorizzata con riferimento a Utente e Prodotto.
Rimborso	Server applicativo	Database (MySQL)	Stato gestito tramite workflow definiti.
RichiestaRimborso	Server Applicativo	Database (MySQL)	Oggetto creato dall'utente, collegato a Ordine, per gestire il processo di rimborso.
GestoreSito	Server Applicativo	Pannello Admin (Web-based)	Permette al gestore di controllare e gestire tutto ciò che riguarda il sito.

L'architettura del sistema Soundshelf è stata pensata per essere distribuita su più nodi, così da garantire scalabilità ed elevata disponibilità. Tuttavia, questa scelta comporta numerose difficoltà, dividibili in due macroaree.

3.3.2. Problemi legati ai nodi multipli

- **Latenza della rete e sincronizzazione**
  Quando il sistema è distribuito su più nodi, i dati devono essere sincronizzati tra di loro attraverso la rete. Questo processo può causare ritardi che si traducono in latenze, influenzando negativamente le prestazioni, soprattutto durante operazioni in tempo reale come il tracciamento degli ordini o l'aggiornamento della disponibilità dei prodotti. Inoltre, è fondamentale mantenere i dati coerenti su tutti i nodi, cosa che diventa più difficile nei momenti di traffico elevato.

- **Bilanciamento del carico**
  Per garantire prestazioni ottimali è necessario distribuire in modo equo il traffico tra i nodi. Senza un bilanciamento efficace, infatti, alcuni nodi potrebbero essere sovraccaricati e altri rimanere inutilizzati.
  Questo potrebbe causare rallentamenti o, nel peggiore dei casi, il blocco di alcune funzionalità del sistema. È necessario quindi adottare politiche ottimali per il bilanciamento e la scalabilità del sistema.

- **Gestione dei guasti**
  In un sistema distribuito, se un nodo smette di funzionare, gli altri dovranno essere in grado di gestire il carico aggiuntivo senza interruzioni. Per farlo, è necessario implementare soluzioni di backup e replica dei dati, i quali inevitabilmente aumenteranno la complessità e richiederanno risorse aggiuntive.

- **Manutenzione e aggiornamenti**
  Aggiornare o modificare un sistema distribuito è più complicato rispetto a farlo per un'architettura centralizzata. Ogni aggiornamento deve essere espanso a tutti i nodi, evitando problemi di coerenza tra essi. Questo meccanismo richiede una pianificazione e una gestione ottimale del codice.

- **Sicurezza dei dati**
  In un sistema distribuito la comunicazione tra i nodi avviene tramite la rete. Ciò va ad aumentare il rischio di accessi non autorizzati.
  Sono quindi necessarie politiche di sicurezza per garantire che i dati trasmessi vengano protetti attraverso crittografia o altri sistemi di sicurezza.

3.3.3. Problemi legati al riutilizzo del software

- **Compatibilità delle versioni**
  Il riutilizzo di librerie o moduli software già esistenti può introdurre problemi di compatibilità. Ad esempio, alcune parti del sistema potrebbero dipendere da versioni diverse dello stesso componente, causando conflitti tra le parti.

- **Dipendenze tra componenti**

Riutilizzare software riduce i tempi di sviluppo, ma può creare dipendenze che rendono il sistema più difficile da gestire. Se un componente venisse aggiornato, altre parti del sistema potrebbero smettere di funzionare correttamente, richiedendo modifiche e test approfonditi.

- **Prestazioni**
  L'integrazione di librerie o moduli riutilizzati potrebbe portare a un sovraccarico di codice, includendo funzionalità che non sono realmente necessarie. Questo potrebbe rallentare il sistema, occupare più memoria e peggiorare le prestazioni complessive.

- **Testing e qualità**
  Anche se i componenti riutilizzati sono stati testati in modo indipendente, è necessario verificarne il funzionamento all'interno del sistema. Eventuali problemi possono richiedere modifiche ai componenti stessi, aumentando i tempi di sviluppo.

- **Licenze**
  L'uso di software di terze parti richiede attenzione alle licenze. Alcuni componenti open-source, ad esempio, potrebbero avere restrizioni sul loro utilizzo, che devono essere rispettate per evitare problemi legali.

3.4. Persistent data managment

3.4.1. Identificazione oggetti persistenti

In SoundShelf abbiamo la necessità di salvare diversi dati che vanno mantenuti anche durante eventuali system failure o system restart. È fondamentale mantenere la persistenza dei dati relativi a *Prodotti* e *Utenti*, nonché tutti gli *Ordini* di ogni singolo Utente, le sue *Recensioni*, le sue *RichiesteSupporto* e le sue *RichiesteRimborso* con relativi *Rimborsi*. Questi oggetti sono ben definiti e di centrale importanza per il sistema; quindi, si pensa non subiranno molti cambiamenti durante la vita del software.

3.4.2. Strategia di archiviazione dati

Visti i requisiti non funzionali di SoundShelf (in particolare **NFR 6**) vi è la necessità di un sistema di archiviazione che permetta azioni rapide e quasi immediate per l'utente. Inoltre, vista la naturale predisposizione dei dati trattati da SoundShelf per un'archiviazione sotto forma di tabelle e la problematica dell'accesso concorrente ai dati, viene scelta un'implementazione tramite DBMS relazionale.

3.5.Access control and security

Il sito SoundShelf implementa un modello di controllo degli accessi basato su una **matrice di accesso** per definire le autorizzazioni relative ai diversi ruoli utente. Di seguito è presentata la matrice in questione dove sulle righe sono riportati gli oggetti individuati in fase di analisi e sulle colonne gli attori che possono eseguire operazioni su tali oggetti.

Actor Object	Guest	Cliente	Gestore Sito
Autenticazione	registrazione	login logout	login logout
CatalogoProdotto	visualizzaProdotti	visualizzaProdotti	aggiungiProdotto rimuoviProdotto
Prodotto	aggiungiProdottoCarrello	aggiungiProdottoCarrello	modificaProdotto
Carrello	eliminaProdotto	rimuoviProdotto completaAcquisto	
ElementoCarrello	modificaQuantità	modificaQuantità	
Ordine		ordineRicevuto	ordineSpedito pagamentoNonRicevuto
CatalogoOrdini		visualizzaOrdiniCliente	visualizzaOrdini
Recensione			rimuoviRecensione
CatalogoRecensioni	visualizzaRecensioni	pubblicaRecensione visualizzaRecensioni	visualizzaRecensioni
RichiestaSupporto	richiediSupporto	richiediSupporto	presaInCarico richiestaInformazioni risolvi
CatalogoRichiesta Supporto			visualizzaRichiesteSupporto
RichiestaRimborso		richiediRimborso	accettaRichiestaRimborso rifiutaRichiestaRimborso
CatalogoRichiesta Rimborso		visualizzaRimborsoOrdine	visualizzaRichiesteRimborso
Rimborso			emettiRimborso

Descrizione del Modello

- **Guest**: Ha accesso limitato alla sola visualizzazione dei prodotti, può effettuare una richiesta di supporto; può, inoltre, aggiungere prodotti al carrello senza poter effettuare altre operazioni di modifica o gestione.
- **Cliente**: Può interagire con un sottoinsieme più ampio degli oggetti, come effettuare ordini, recensire prodotti e richiedere rimborsi.
- **Gestore del sito**: Ha pieno controllo su oggetti critici del sistema come catalogo prodotti, ordini e gestione di recensioni e rimborsi.

3.6. Global software control

Per garantire un sistema robusto e che rispetti la concorrenza dei file, si adottano le seguenti strategie di accesso concorrente ai dati condivisi:

- *Viene usato il DBMS per gestire gli accessi concorrenti al database*, il quale, utilizzando un modello ACID, garantirà la coerenza nel database.
- *I prodotti aggiunti al carrello non generano variazione di disponibilità* e quindi non comportano un accesso e modifica del database. La variazione viene registrata in quest'ultimo, solo a conclusione della fase di acquisto, evitando così problemi di concorrenza generati dalla variazione continua della quantità disponibile del prodotto.

3.7. Boundary conditions

- **Avvio del sistema**
  Durante l'avvio, il sistema carica progressivamente tutti i moduli necessari per il funzionamento, come la gestione degli utenti, il catalogo prodotti e il database. Prima di dichiararsi pienamente operativo esegue una serie di controlli interni per verificare che tutti i componenti funzionino correttamente. Se viene rilevato un problema, il sistema annulla il proprio avvio, notificando i gestori del fallimento di avvio.

- **Spegnimento del Sistema**
  Durante lo spegnimento, il sistema chiude in modo sicuro tutte le connessioni rimaste attive, in modo da salvaguardare i dati in corso e registrare lo stato finale. Questo garantisce che nessuna informazione venga persa e che, al successivo avvio, il sistema possa riprendere il proprio lavoro senza anomalie.

- **Errori e situazioni eccezionali**
  In caso di errori il sistema è progettato per ridurre al minimo l'impatto sugli utenti. Ad esempio, se un modulo non risponde, il sistema proverà a riavviarlo automaticamente.
  Se il problema dovesse persistere, verrebbe inviata una notifica ai gestori del sito, e agli utenti verrebbe mostrato un messaggio richiedente attesa per la sistemazione del danno.
  Inoltre, eventuali errori critici sono registrati in un log accessibile agli amministratori per l'analisi e la risoluzione.

4. SUBSYSTEM SERVICES

GLOSSARY

1. **Carrello**
   - **Definizione**: 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.
2. **Checkout**
   - **Definizione**: Il processo finale in cui l'utente fornisce i dettagli di spedizione per completare l'acquisto di uno o più prodotti.
3. **Prodotto**
   - **Definizione**: Un vinile, CD, o altro articolo musicale messo in vendita su *Soundshelf*.
4. **Ordine**
   - **Definizione**: 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.
5. **Recensione**
   - **Definizione**: 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.
6. **Rimborso**
   - **Definizione**: Restituzione di denaro a un utente per un prodotto che è stato restituito per un ordine che è stato cancellato.
7. **Supporto**
   - **Definizione**: Un servizio clienti fornito tramite il sito, che permette agli utenti di inviare richieste di assistenza riguardo ordini, pagamenti, o altri problemi.
8. **Inventario**
   - **Definizione**: La gestione e il monitoraggio delle quantità di prodotti disponibili nel magazzino di *Soundshelf*.
9. **Vinile**
   - **Definizione**: 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 *Soundshelf*.
10. **Formato**
    - **Definizione**: 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.
11. **Artista**
    - **Definizione**: Il creatore musicale dietro a un prodotto (album, vinile, ecc.). L'artista può essere un singolo individuo o un gruppo musicale.
12. **Stato ordine**
    - **Definizione**: Indica la fase in cui si trova un ordine.
13. **Tipo di utente**
    - **Definizione**: 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.
14. **Utente**
    - **Definizione**: 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.
15. **Quantità disponibile**
    - **Definizione**: La quantità di un prodotto che è attualmente in stock e pronta per essere

venduta.

16. **Data di spedizione**
    - **Definizione**: La data in cui un ordine viene effettivamente inviato dal magazzino per la consegna al cliente.
17. **Data di consegna**
    - **Definizione**: La data prevista per la consegna di un ordine, indicata dal metodo di spedizione scelto.
18. **Recensione Prodotto**
    - **Definizione**: Un'analisi o una valutazione di un prodotto scritta da un utente che ha acquistato l'articolo. Le recensioni includono il punteggio e commenti.
19. **Data di richiesta rimborso**
    - **Definizione**: La data in cui un utente invia una richiesta per il rimborso di un ordine o di un prodotto.
20. **Rimborso approvato**
    - **Definizione**: 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.
21. **Rimborso rifiutato**
    - **Definizione**: 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.
22. **Data di chiusura supporto**
    - **Definizione**: La data in cui una richiesta di supporto viene risolta o chiusa dopo essere stata esaminata e trattata.
23. **Supporto risolto**
    - **Definizione**: Stato di una richiesta di supporto che è stata risolta e per la quale non è più necessario alcun intervento.
24. **Supporto in lavorazione**
    - **Definizione**: Stato di una richiesta di supporto che è attualmente in fase di esame o risoluzione da parte del team di assistenza clienti.