



UNIVERSITÀ DEGLI STUDI DI SALERNO

*Corso di Laurea in Informatica, prof. A. De Lucia,
a.a 2021/2022
Progetto di Ingegneria del Software*



System Design Document

Partecipanti	Matricola
Marta Coiro	0512108154
Katia Buonocore	0512106528
Rita Cuccaro	0512109495

Revision History

Data	Versione	Descrizione	Autore
16/11/2021	1.0	Prima stesura del documento (Problem <u>Statement</u>)	Membri del team
30/11/2021	1.0	Requirement Analysis Document	Membri del team
06/12/2021	1.0	System Design Document	Membri del team
20/12/2021	1.0	Gestione Dati Persistenti_MusicConsole	Membri del team
27/12/2021	1.0	Object Design Document	Membri del team
15/01/2022	1.0	Test Plan	Membri del team
28/01/2022	1.0	Test Execution Report	Membri del team
18/02/2022	1.0	Test Summary Report	Membri del team

Indice

1. INTRODUZIONE

- 1.1 Scopo del sistema.
- 1.2 Obiettivi di design.
 - 1.2.1 Criteri di Performance.
 - 1.2.2 Criteri di Affidabilità.
 - 1.2.3 Criteri di Manutenzione.
 - 1.2.4 Criteri per l'Utente Finale.
- 1.3 Definizioni, acronimi e abbreviazioni.
- 1.4 Riferimenti.
- 1.5 Panoramica.

2. ARCHITETTURA DEL SISTEMA CORRENTE

3. ARCHITETTURA DEL SISTEMA PROPOSTO

- 3.1 Panoramica
- 3.2 Decomposizione del sistema.
 - 3.2.1 Diagramma Generale.
- 3.3 Mapping Hardware/Software.
- 3.4 Gestione dei dati persistenti.
- 3.5 Controllo degli accessi e sicurezza.
- 3.6. Controllo software globale.
- 3.7. Boundary Conditions.
 - 3.7.1 Scenario di avvio.
 - 3.7.2 Scenario di terminazione.
 - 3.7.3 Fallimenti.

4. SERVIZI DEI SOTTOSISTEMI.

- 4.1 Gestione Account.

4.2 Gestione Acquisto.

4.3 Gestione Carrello.

4.4 Gestione Prodotti.

5. GLOSSARIO.

1. INTRODUZIONE

1.1 Scopo del sistema

Lo sviluppo del web ha reso popolare la vendita di oggetti tramite negozi online, il cosiddetto e-commerce, che durante gli ultimi anni ha cambiato radicalmente il modo in cui le persone fanno shopping. Questo sviluppo ha coinvolto anche il mondo musicale e ha fatto sì che la richiesta d'acquisti di prodotti musicali online crescesse in modo esponenziale.

Per questo Music Console si propone di diventare una piattaforma musicale dedicata allo streaming musicale e all'acquisto di brani, album, podcast, cd e vinili. L'obiettivo è quello di soddisfare i vari gusti musicali, permettendo a ciascun utente di scegliere tra la musica degli anni passati e quella contemporanea, offrendo la possibilità di accedere ad un vasto database di canzoni e podcast da poter ascoltare online e anche offline, acquistandoli.

Nel dettaglio il nostro sito si pone i seguenti obiettivi:

- Il sistema dovrà essere usabile.
- Il sistema si dovrà adattare graficamente ad ogni dispositivo.
- Il sistema dovrà gestire picchi di carico.
- Il sistema dovrà essere capace di rispondere a qualsiasi tipo di input.
- Il sistema dovrà garantire la privacy degli utenti.
- Il sistema dovrà dare la possibilità di ampliarsi successivamente.

1.2 Obiettivi di design

Il sistema MusicConsole deve poter essere il più efficiente ed intuitivo possibile. Tale efficienza sarà costruita attraverso rapidi tempi di risposta ad ogni genere di input ma anche differenti politiche di tolleranza all'errore. In più si punterà ad avere una buona manutenibilità attraverso l'inserimento di nuove funzionalità. Sarà intuitivo attraverso agevoli interfacce. Tutto ciò ci porta a considerare quattro distinte classi: Performance, Affidabilità, Manutenzione ed Utente Finale, per le quali definiremo anche una priorità.

1.2.1 Criteri di Performance

Tempo di Riposta	MusicConsole deve essere reattivo per tutte le operazioni più immediate come l'aggiunta di un brano. Per operazioni come l'ascolto di pochi secondi di un brano, deve garantire dei tempi di risposta nell'ordine delle decine di secondi, ma, essendo un sistema web, molto dipenderà dalla qualità della connessione e dalla congestione della rete sul sistema online.
Throughput	I picchi di carico, fino a circa 100 utenti simultaneamente collegati, devono essere gestiti dal sistema senza rallentamenti, garantendo fluidità e una latenza molto bassa. Il sistema, deve garantire consistenza in tutte le operazioni che includono anche chiamate al database, specialmente nei momenti di maggiore carico.
Memoria	Il sistema utilizza un database relazionale per memorizzare tutti i dati. La mole dei dati non rappresenterà un problema di performance del sistema.

1.2.2 Criteri di Affidabilità

Robustezza	I componenti devono essere affidabili ed essere in grado di poter mantenere i propri dati anche in caso di guasti come i problemi elettrici, guasti hardware, attacchi informatici, problemi legati al browser.
Disponibilità	MusicConsole deve essere disponibile all'uso, 24 ore su 24, da parte degli utenti, grazie ad un server sempre attivo.
Tolleranza all'errore	Il sistema deve essere capace di operare durante condizione d'errore. Ciò sarà reso possibile tra tutte quelle sotto-parti del sistema che hanno un grado di accoppiamento basso, in modo tale che l'errore in un sottosistema non intacchi le funzionalità di un altro. Inoltre, deve essere in grado di pianificare dei backup periodici dell'intero sistema.
Sicurezza	Le tecniche utilizzate, per garantire la sicurezza, sono basate prevalentemente su una 'login', la quale permette il riconoscimento dell'utente. Tutti gli accessi al sistema avvengono tramite un'operazione di routing che controlla se l'utente che sta operando sul sistema ha i permessi adeguati per usufruire di determinate funzionalità. C'è, quindi, un controllo che verifica i permessi degli utenti che verranno adeguatamente distribuiti sulla base del grado di importanza e responsabilità degli Utenti. In particolare un amministratore avrà i massimi permessi poiché ha facoltà di utilizzare qualsiasi tipo di funzione disponibile, mentre un cliente avrà le piene facoltà di gestione del proprio account, quindi delle modifiche del profilo e degli acquisti. Inoltre le password sono criptate.

1.2.3 Criteri di Manutenzione

Estensibilità	La progettazione del sistema sarà condotta in modo da agevolare la facile introduzione di nuove funzionalità utilizzando il linguaggio di markup HTML5, i fogli di stile CSS, Bootstrap e JQuery.
Modificabilità	Deve essere possibile intervenire sul codice esistente per correggere eventuali bugs o implementare nuove funzionalità. Bisogna garantire che il codice sia leggibile per rendere agevole la modifica.
Leggibilità	Il codice sarà ben strutturato per semplificare eventuali interventi su di esso.
Tracciabilità dei requisiti	Grazie alla tracciabilità dei requisiti, sarà possibile effettuare le modifiche necessarie al corretto funzionamento del sistema, valutando correttamente i costi e i rischi che le modifiche porteranno.

1.2.4 Criteri per l'Utente Finale

Usabilità	MusicConsole rende ogni funzione di semplice uso garantendo un'ottima esperienza lavorativa all'utente grazie ad interfacce grafiche intuitive cioè user-friendly.
------------------	--

1.3 DEFINIZIONI, ACRONIMI E ABBREVIAZIONI

- **MusicConsole:** Nome del Sistema che verrà sviluppato.
- **Guest:** utente ospite, non registrato al sistema, che può visitare il sito con funzionalità ridotte (visualizzare, ricercare e leggere informazioni di un prodotto, ascolto di un breve estratto di un brano).
- **Utente Registrato:** utente registrato al sito, può finalizzare un acquisto, visualizzare il carrello, visualizzare lo storico degli ordini, ascoltare per intero un brano, visualizzare, creare e aggiungere brani alle playlist e accedere alla pagina “MyAccount” per visualizzare ed eventualmente modificare le informazioni personali. Queste funzionalità vanno aggiunte a quelle definite in precedenza per un utente non registrato.
- **DBMS:** Data Base Management System.

1.4 RIFERIMENTI

- Requisiti funzionali: Sezione 3.2 del RAD.
- Requisiti non funzionali : Sezione 3.3 del RAD.

1.5 PANORAMICA

Nel documento verranno affrontati l’analisi dell’architettura del sistema proposto, la scomposizione in sottosistemi con la definizione della strategia di deploy e le condizioni limite. Verranno quindi definiti i servizi esposti da ciascun sottosistema. L’ultima parte del documento è costituita dal glossario che si occupa di elencare una serie di termini e fornire la relativa spiegazione in maniera tale da dare supporto a coloro che leggono il documento.

2. ARCHITETTURA DEL SISTEMA CORRENTE

Il sistema proposto è implementato da zero. Esistono altri sistemi simili come ad esempio Spotify, iTunes e Amazon Music.

3. ARCHITETTURA DEL SISTEMA PROPOSTO

3.1 Panoramica

MusicConsole è un'applicazione distribuita che fornisce un'interfaccia web. Essa, inoltre, ricorre all'utilizzo di un database relazionale per il salvataggio dei dati persistenti.

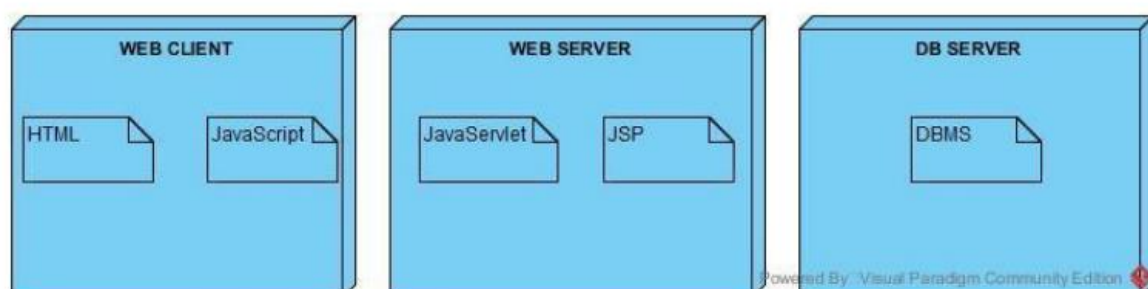
Nello sviluppo del nostro sistema utilizzeremo un'architettura di tipo MVC. IL pattern Model-View-Controller prevede che il software venga diviso in tre parti ognuna delle quali ha un compito diverso. In particolare:

Model si occupa di gestire i dati e quindi sarà responsabile dell'interazione con il database sottostante.

View si occupa di curare l'interazione con l'utente e quindi avrà il compito di gestire la formattazione dei dati che verranno visualizzati.

Controller dopo aver ricevuto i comandi forniti dall'utente si occuperà di elaborare i dati, passarli al model se necessario e inviare la risposta al view appropriato.

L'utilizzo del modello MVC comporta numerosi vantaggi tra i quali ricordiamo la possibilità di suddividere il lavoro più facilmente tra i vari componenti del team e la maggiore agilità negli interventi di manutenzione. Infatti, l'utilizzo di un modello e di regole standard facilita la comprensione anche a chi non ha inizialmente lavorato su quel sottosistema. All'interno del nostro sistema il **model** verrà realizzato utilizzando classi Java appropriate, la parte di **view** verrà implementata utilizzando pagine HTML e JavaScript e i **control** saranno realizzati tramite Servlet.



3.2 Decomposizione del sistema

Per realizzare il sistema è stata usata un'architettura three-tier. Questo è un caso particolare di un'architettura multi tier in cui la logica dell'applicazione viene suddivisa in tre layer:

- **Presentation layer:** composto dalle interfacce grafiche utili per l'interazione con l'utente.
- **Application Logic layer:** composto dagli oggetti che si occupano della gestione del controllo, dell'elaborazione dati e di notificare i cambiamenti al presentation layer. Questo strato interagisce con il database sottostante tramite lo storage layer.
- **Data Storage layer:** si occupa della memorizzazione dei dati persistenti e del loro recupero dal database ad esempio eseguendo delle query.

Il sottosistema Presentation Layer è costituito dal seguente sottosistema:

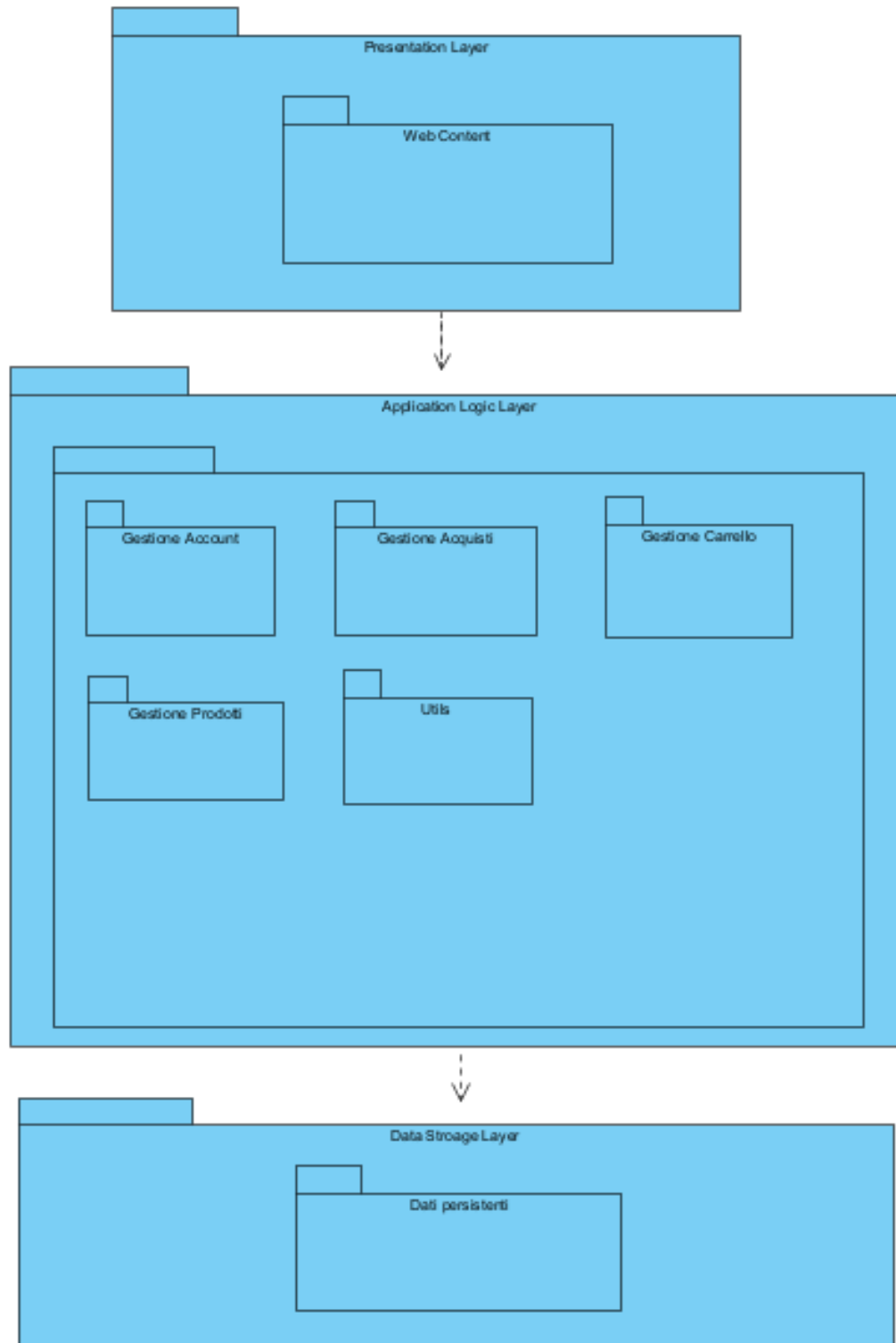
- **Web Content.** Include tutta la logica di presentazione.

Il sottosistema Application Layer è stato diviso in 5 sottosistemi:

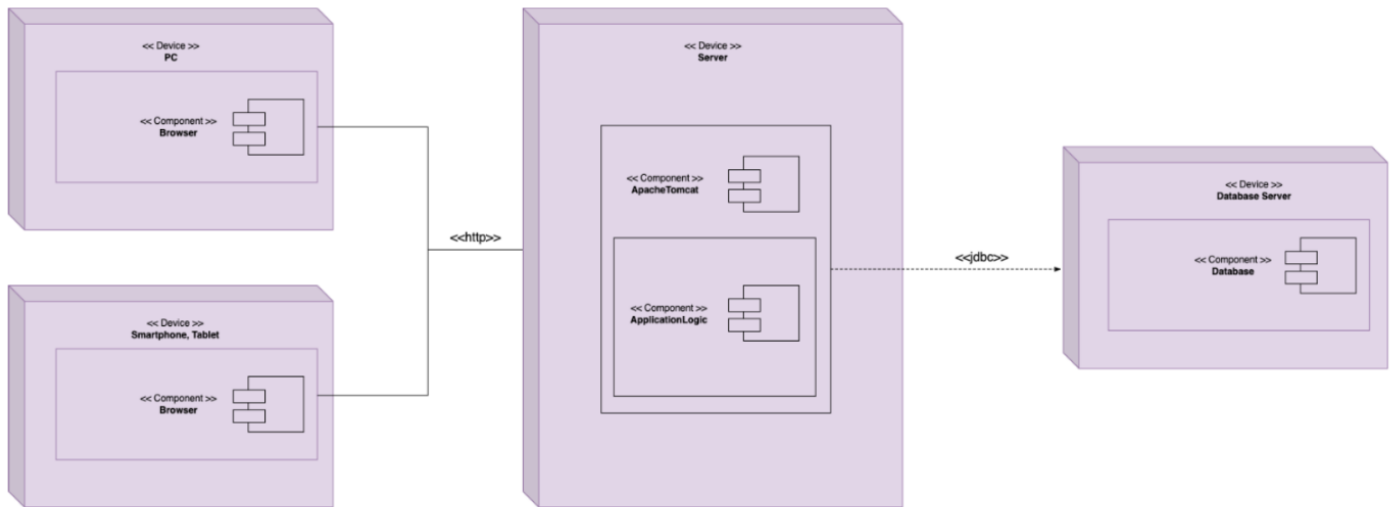
- **Gestione Account:** Include tutte le funzionalità legate agli account di utenti registrati/non registrati, amministratori e tutte le operazioni correlate ad essi.
- **Gestione Acquisti:** Include tutte le funzionalità riguardanti gli acquisti.
- **Gestione del Carrello:** Include tutte le funzionalità relative al carrello.
- **Gestione Prodotti:** Include tutte le funzionalità relative ai prodotti presenti sul nostro sito.
- **Gestione Utils:** Include le funzionalità di connessione al database, gestione password e utility del sistema.

3.2.1 Diagramma generale

Viene riportato di seguito il diagramma generale.



3.3. Mapping Hardware/Software



Il sistema utilizza un'architettura Client/Server. Il Web Server è rappresentato da Apache Tomcat 9 ed è situato su una singola macchina, la logica del sistema è costituita da Java Servlet, mentre l'interfaccia utente è realizzata utilizzando pagine JSP (Java Servlet Page). Il Client è rappresentato dal Web Browser utilizzato dall'utente. La comunicazione tra i nodi è rappresentata da richieste e risposte http tra client e server, e da query in JDBC tra server e database.

3.4 Gestione dei dati persistenti

Per gestire i dati persistenti è stato preferito l'utilizzo di un database relazionale, poiché è necessario un ampio spazio di memorizzazione. Inoltre, i database consentono l'accesso concorrente da parte di tutti. Per la gestione del database si sceglie un DBMS MySQL. Si rimanda al documento "Gestione Dati Persistenti_MusicConsole" allegato al presente.

3.5 Controllo degli accessi e sicurezza

Il controllo degli accessi è garantito tramite l'utilizzo di username e password per gli utenti del sistema che hanno possibilità di creare o modificare gli oggetti che modellano entità di dominio, così da prevenire accessi non autorizzati ad informazioni sensibili. Sottolineiamo che il sistema non fornirà un metodo di recupero della password, si ricorrerà quindi all'utilizzo della sessione del server per tenere traccia dell'utente loggato. Per questioni di efficienza, la sessione sarà attiva per soli 30 minuti dopo l'ultima interazione dell'utente col sistema. Il salvataggio delle password nel database sarà cifrato con [PASSWORD]. Le operazioni che gli utenti dell'applicazione web possono effettuare sugli oggetti sono riportate nella tabella che segue:

	Utenza	Libreria	Carrello
Utente	Registrazione	-Visualizzazione dei prodotti presenti nel catalogo. -Accesso alla pagina di info di un prodotto. -Ascolto di un brano. -Ricerca di un prodotto per categoria. -Visualizzazione delle playlist associate all'utente. -Visualizzazione	-Aggiunta prodotto al carrello. -Eliminazione prodotto dal carrello. -Ritornare allo shopping. -Procedere con l'ordine. -Visualizzare lo storico dei propri ordini.

		dei dati personali dell'utente.	
Gestore degli Ordini	Log-in.	-Visualizzazione storico ordini globale. -Assegnare l'ordine a uno specifico corriere. -Definire la data di consegna.	
Gestore del Magazzino	Log-in.	-Rifornire nuovi prodotti e prodotti già esistenti nel sito. -Aggiungere nuovi prodotti.	
Gestore dell'Inventario	Log-in.	-Modifica prodotto.	

3.6 Controllo software globale

Il sistema adotta un controllo del flusso globale di tipo thread-driven, questo perché il web container (Tomcat) permette l'interazione concorrente tra la WebApp e più client tramite l'intercettazione di eventi generati proprio da questi ultimi. Questa soluzione permette al sistema di poter rispondere a più utenti contemporaneamente ma richiede che gli accessi in scrittura ai dati persistenti avvengano sequenzialmente, gestendo opportunamente le sezioni critiche. In generale, ogni richiesta da parte di un utente verrà eseguita in un thread dedicato.

3.7 Boundary condition

Per consentire il corretto accesso al sistema, il cliente necessita di una connessione ad Internet, in modo che possa visualizzare il sito correttamente. Una volta entrati nel sito sono necessarie le credenziali per usufruire del sistema.

3.7.1 Scenario di avvio

Nome scenario	Startup del sistema
Attori partecipanti	Anna: Amministratore
Flusso degli eventi	<ol style="list-style-type: none">1. Anna accede al sistema operativo per avviare i servizi del web server e del database cliccando sul pulsante "Avvia".2. Il sistema con opportune procedure di avvio, attiva i server e i servizi in remoto rendendosi quindi disponibile ad eventuali richieste.3. Il sistema è disponibile da questo momento in poi.

3.7.2 Scenario di terminazione

Nome scenario	Shutdown del sistema
Attori partecipanti	Anna: Amministratore
Flusso degli eventi	<ol style="list-style-type: none">1. Anna decide di voler arrestare il sistema e quindi si reca nella pagina dedicata.2. Anna clicca sul pulsante "Arresta".3. Il sistema controlla se ci sono ancora richieste in sospeso.4. Il sistema porta a termine eventuali richieste in sospeso.5. Il sistema, quindi, disattiva i servizi in remoto e il server.6. Il sistema da questo momento in poi non è più disponibile.

3.7.3 Fallimenti

Possono verificarsi vari casi di fallimento del sistema:

- Errore hardware o software: si cerca di ripristinare una configurazione del sistema precedente all'errore. Non si perdono i dati visto che questi sono gestiti usando il DBMS.
- Interruzione della connettività con la rete: il servizio resterà interrotto fin quando non viene ripristinato il collegamento.

4. Servizi dei sottosistemi

4.1 Gestione Account

Sottosistema	Gestione Account
Descrizione	<p>Sottosistema che gestisce e possiede tutte le funzionalità di cui possono usufruire gli utenti ovvero:</p> <ul style="list-style-type: none">• Registrazione utenti.• Login utenti/amministratori.• Logout utenti/amministratori• Visualizzazione del proprio profilo.• Modifica profilo.• Eliminazione account.
Servizi offerti	
Servizio	Descrizione
Registrazione	Questa funzionalità permette al visitatore del sito di potersi registrare.
Login	Questa funzionalità permette all'utente/amministratore di effettuare l'accesso al sistema.
Logout	Questa funzionalità permette all'utente/amministratore di effettuare la disconnessione al sistema.
Modifica profilo	Questa funzionalità permette all'utente di poter modificare i suoi dati personali e poterli salvare.
Visualizza profilo	Questa funzionalità permette all'utente di poter visualizzare le sue informazioni personali relative al suo account.

Eliminazione Account

Questa funzionalità permette all'utente di poter eliminare il proprio account.

4.2 Gestione Acquisto

Sottosistema

Gestione Acquisto

Descrizione

Sottosistema che comprende:

- Tutte le operazioni di gestione ed emissione di un ordine da parte del Gestore degli Ordini.
- Visualizzazione dello storico degli ordini di un determinato utente.

Servizi offerti

Servizio

Descrizione

Gestione ordine

Questa funzionalità permette al Gestore degli Ordini di visualizzare la lista degli ordini da emettere.

Emissione ordine

Questa funzionalità permette al Gestore degli Ordini di emettere un ordine associato ad un determinato cliente.

Storico degli ordini

Questa funzionalità permette all'utente di visualizzare tutti gli ordini che ha effettuato.

4.3 Gestione Carrello

Sottosistema	Gestione Carrello
Descrizione	<p>Sottosistema che comprende:</p> <ul style="list-style-type: none">• Tutte le funzionalità legate all'aggiunta, eliminazione e visualizzazione dei prodotti presenti nel carrello.• Tutte le funzionalità legate alla gestione della Carta di Credito associata ad un determinato utente.• Tutte le funzionalità legate all'invio di un ordine.• La funzionalità che permette di svuotare il carrello.
Servizi offerti	
Servizi	Descrizione
Gestione Carrello	Questa funzionalità permette all'utente di aggiungere, eliminare e visualizzare i prodotti presenti nel suo carrello.
Gestione Carta di Credito	Questa funzionalità permette all'utente di inserire, durante la fase di checkout, tutte le informazioni relative alla propria carta di credito, ed eventualmente decidere di salvarle.
Gestione Checkout	Questa funzionalità permette all'utente di inserire, durante la fase di checkout, le informazioni relative all'indirizzo di fatturazione e spedizione e di concludere l'ordine.
Svuotamento Carrello	Questa funzionalità permette di svuot

Svuotamento carrello

Questa funzionalità permette di svuotare immediatamente il carrello.

4.4 Gestione Prodotti

Sottosistema	Gestione Prodotto
Descrizione	<p>Sottosistema che comprende:</p> <ul style="list-style-type: none">• Tutte le funzionalità legate alla gestione dei prodotti.• Tutte le funzionalità legate alla realizzazione delle Playlist da parte di un utente e aggiunta e rimozione dei brani dalla playlist.• Tutte le funzionalità legate alla ricerca dei prodotti nel sito da parte dell'utente.• Tutte le funzionalità permesse al Gestore dell'Inventario, legate alla modifica dei prodotti.• Tutte le funzionalità permesse al Magazziniere, legate all'aggiunta di nuovi prodotti e gestione della relativa quantità e la relativa eliminazione.

Servizi offerti

Servizi	Descrizione
Gestione Prodotto	Questa funzionalità permette al Gestore del Magazzino di poter aggiungere, eliminare un prodotto.
Gestione Playlist	Questa funzionalità permette ad un utente registrato di creare delle Playlist e aggiungere/rimuovere brani da esse.

Gestione Ricerca	Questa funzionalità permette ad un utente di ricercare dei prodotti all'interno del sistema.
Rifornimento Prodotti	Questa funzionalità permette al Gestore del Magazzino di rifornire nuovi prodotti e prodotti già esistenti all'interno del sito, tenendo conto della disponibilità del magazzino.
Modifica Prodotti	Questa funzionalità permette al Gestore dell'Inventario di poter modificare le informazioni dei prodotti.

5. GLOSSARIO

- **Client:** componente che accede ai servizi o alle risorse di un'altra componente, detta server.
- **Deployment Diagram:** Schema che descrive la struttura dinamica del sistema DBMS: programma informatico (o, più frequentemente, un insieme di programmi) progettato per gestire un database, ovvero un insieme di numerosi dati strutturati. Le operazioni, normalmente, sono richieste da un gran numero di utenti.
- **Form:** finestra di dialogo incorporata in una pagina Web che consente all'utente di inserire informazioni destinate ad un server. Generalmente richiede un programma sul server che si occupi di esaminare le informazioni inviate. E' composto da spazi (campi) predefiniti, ad esempio menù a tendina, elenchi puntati o caselle di testo libero.
- **JDBC:** API per il linguaggio di programmazione Java che serve ai client per connettersi a un database. Fornisce metodi per interrogare e modificare i dati. È orientata ai database relazionali.
- **Login:** Procedura attraverso la quale ci si collega con un qualsiasi servizio in linea. All'utente viene assegnato un nome di login ed una password che vengono richiesti dal sistema ogni volta che ci si collega.
- **Layer:** E' un insieme di classi con funzionalità simili (tipicamente raggruppati in un unico package).
- **Logout:** Operazione attraverso la quale si termina un collegamento con un sistema al quale si ha accesso attraverso un nome utente e una password (vedi login).
- **MySQL:** Database management system relazionale, composto da un client con interfaccia a caratteri e un server, disponibile su molte piattaforme.
- **Password:** È un metodo di sicurezza che, mediante una stringa di caratteri, permette di identificare un utente specifico. Generalmente le password sono formate da una sequenza di lettere e numeri; digitando correttamente questi caratteri, si può avere accesso al computer o alla rete.
- **Package:** Un Package rappresenta una collezione di classi ed interfacce che possono essere raggruppate in base alla funzione comune da esse svolta
- **Server:** Programma di gestione di un servizio che invia informazioni in un particolare formato ricevuto e interpretato da un programma Client dal lato ricevente.
- **Shutdown:** fase di terminazione del sistema.
- **Start-up:** processo di accensione e di avvio di un computer, di un dispositivo o di un sistema