



UNIVERSITÀ DEGLI STUDI DI SALERNO

**Corso di Ingegneria del Software
System Design Document
(SDD)**

Progetto: GetThatSound.it

Coordinatori del progetto:

Nome
De Lucia Andrea

Partecipanti:

Nome	Matricola
Molli Adriano	0512104348
Epifania Luigi	0512104714

Sommario

1. INTRODUZIONE.....	4
1.1. OBIETTIVI DEL SISTEMA.....	4
2. DESIGN GOALS.....	5
2.1. CRITERI DI PERFORMANCE.....	5
2.2. CRITERI DI AFFIDABILITÀ.....	5
2.3. MANUTENZIONE.....	5
2.4. CRITERI DI USABILITÀ.....	5
3. DEFINIZIONI, ACRONIMI E ABBREVIAZIONI.....	6
4. RIFERIMENTI.....	6
5. ARCHITETTURA DEL SISTEMA E DEI SOTTOSISTEMI.....	7
5.1. ARCHITETTURA DEL SISTEMA PROPOSTO.....	7
5.1.1. <i>Panoramica</i>	7
5.1.2. <i>Decomposizione in Sottosistemi</i>	8
5.1.3. <i>Diagramma di Deployment</i>	9
5.1.4. <i>Mapping Hardware / Software</i>	9
6. GESTIONE DEI DATI PERSISTENTI.....	10
7. GESTIONE E CONTROLLO DELLA SICUREZZA.....	11
8. BOUNDARY CONDITION.....	12
9. SERVIZI DEI SOTTOSISTEMI.....	13
9.1. GESTIONE BRANI IN STALLO.....	13
9.2. GESTIONE BRANII.....	14
9.3. GESTIONE UTENTI.....	15
10. RILASCIO DEL SISTEMA.....	16

1. Introduzione

1.1. Obiettivi del sistema

Il sistema che si intende realizzare è un sito di streaming e sharing di file musicali: **GetThatSound.it**.

Un **brano** è caratterizzato dal numero, titolo, album, artista e una foto.

Un **cliente** può registrarsi al sito indicando nome, cognome, mail, username e password.

Un **cliente registrato** può riprodurre i brani, salvarli e caricarne di propri.

Sul lato amministrativo del sito sono identificati due tipi di ruoli: **amministratore** e **manager**.

Ognuno effettua l'accesso dalla pagina login del sito.

Il **manager** ha il compito di approvare o meno i brani caricati dagli utenti, in base alle politiche di rispetto del copyright e della trasmissione di contenuti adatti a tutti.

L'**amministratore** ha le facoltà di eliminare un utente, nominare i manager o declassarli ad utenti regolari.

Design Goals

1.2. Criteri di performance

Tempi di risposta.

- Per la riproduzione di un brano il tempo massimo è di 1 secondo.
- Per la visualizzazione del profilo utente è di 2 secondi.
- Per l'upload di un brano è di massimo 5 minuti.

1.3. Criteri di affidabilità

Affidabilità.

- I dati sensibili dell'utente non sono visibili se non dall'amministratore del sito.

Robustezza.

- Eventuali input non validi immessi dall'utente saranno opportunamente segnalati attraverso messaggi di errore.

1.4. Manutenzione

Supportabilità.

- Il sistema richiede capacità di adattamento nel caso in cui si decida di trasformarlo in un vero e proprio social musicale.

1.5. Criteri di usabilità

Leggibilità.

- Il sistema userà font sans-serif di grandezza non inferiore ai 15px.

Navigazione.

- Ogni sezione del sito è raggiungibile in non più di 4 click.

Look&Feel.

- L'interfaccia deve adattarsi a qualsiasi dispositivo che acceda al sito.

2. Definizioni, acronimi e abbreviazioni

RAD: Requirements Analysis Document.

SDD: System Design Document.

ODD: Object Design Document.

DB: Database.

MVC: Model Control View.

Manager: utente del sito che gestisce i brani in attesa di approvazione.

Amministratore: amministratore del sito. Gestisce tutte le login registrate.

GetThatSound.it: il sito che si intende sviluppare.

3. Riferimenti

4. Architettura del sistema e dei sottosistemi

Di seguito verrà presentata l'architettura del sistema proposto in cui gestiremo la decomposizione in sottosistemi, il mapping hardware/software, i dati persistenti, il controllo degli accessi e sicurezza. Verranno, inoltre, presentati in maggiore dettaglio i servizi dei sottosistemi.

4.1. Architettura del sistema proposto

4.1.1. Panoramica

Il sistema proposto è un'applicazione web, distribuita secondo il modello client server.

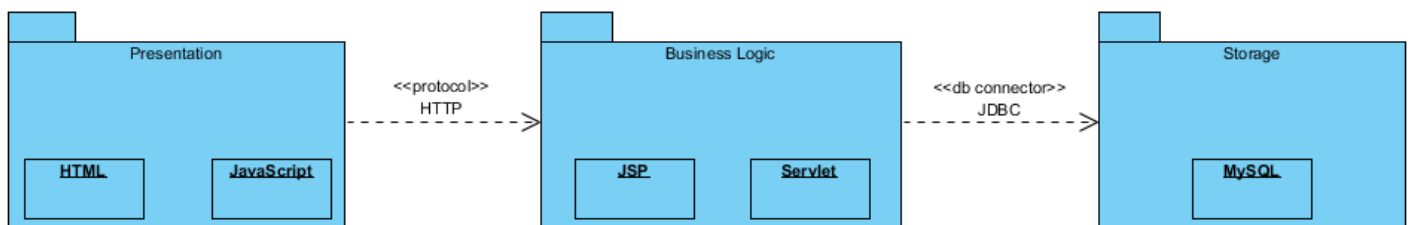
Gli obiettivi posti sono:

- Permettere a un potenziale cliente del sito di ascoltare e caricare brani musicali;
- Diminuire i tempi di upload e download.

Il client gestirà la parte di presentazione e la parte di logica direttamente connessa all'interfaccia grafica, il server invece gestirà la parte di logica relativa ai dati e i dati stessi che saranno salvati in un database. L'intenzione è quella di rendere il sistema manutenibile e di permettere facilmente l'aggiunta di nuove funzionalità qualora se ne presentasse la necessità.

I gestori saranno individuati in base alle funzionalità per poter rendere massima la coesione e minimo l'accoppiamento tra i sottosistemi in modo che i cambiamenti in un sottosistema non influiscano sugli altri.

Come decomposizione in sottosistemi utilizzeremo l'architettura three tier layer.

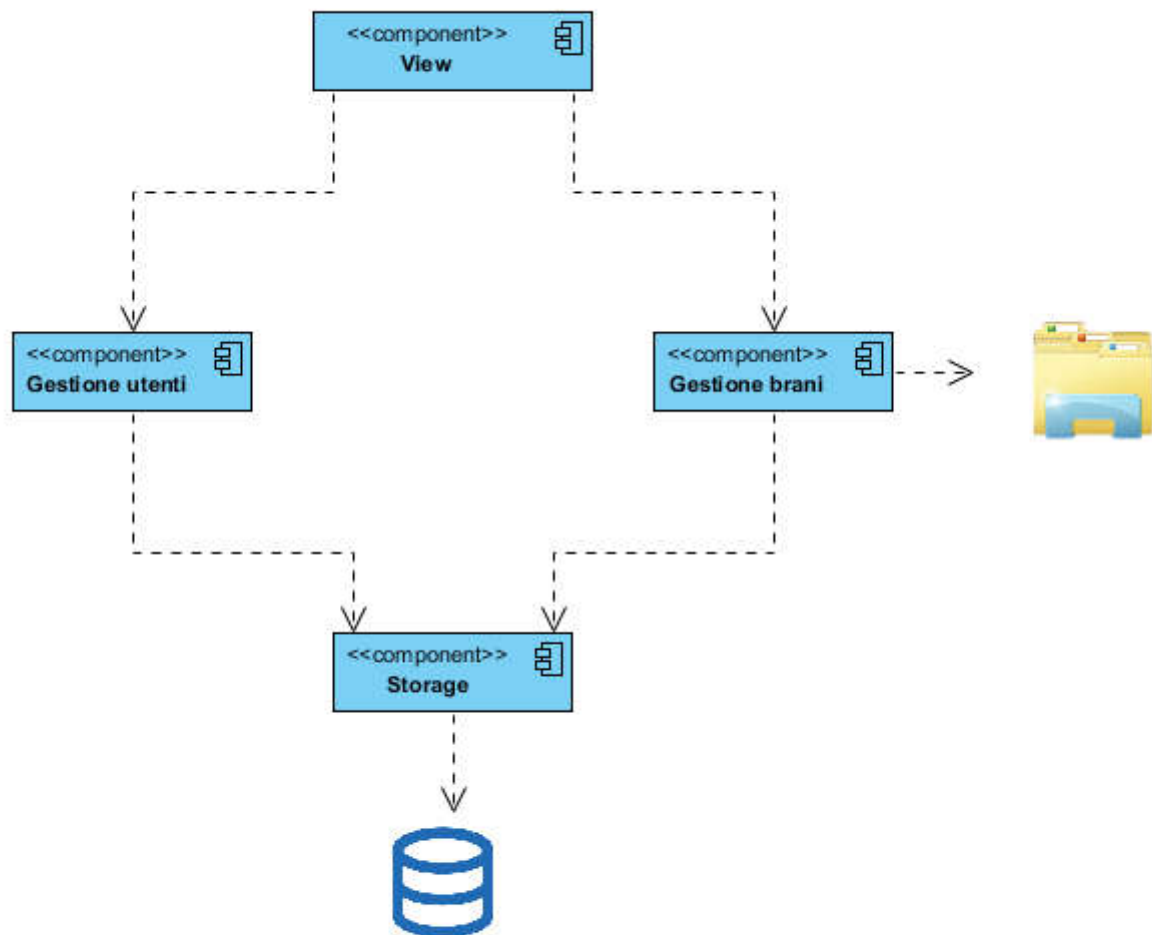


4.1.2. Decomposizione in Sottosistemi

Dopo un'attenta analisi abbiamo deciso di suddividere il sistema nel seguente modo, con l'obiettivo di minimizzare l'accoppiamento ed avere un'alta coesione.

Il sistema si compone nei seguenti 4 sottosistemi:

- View
- Gestione utenti
- Gestione brani
- Model

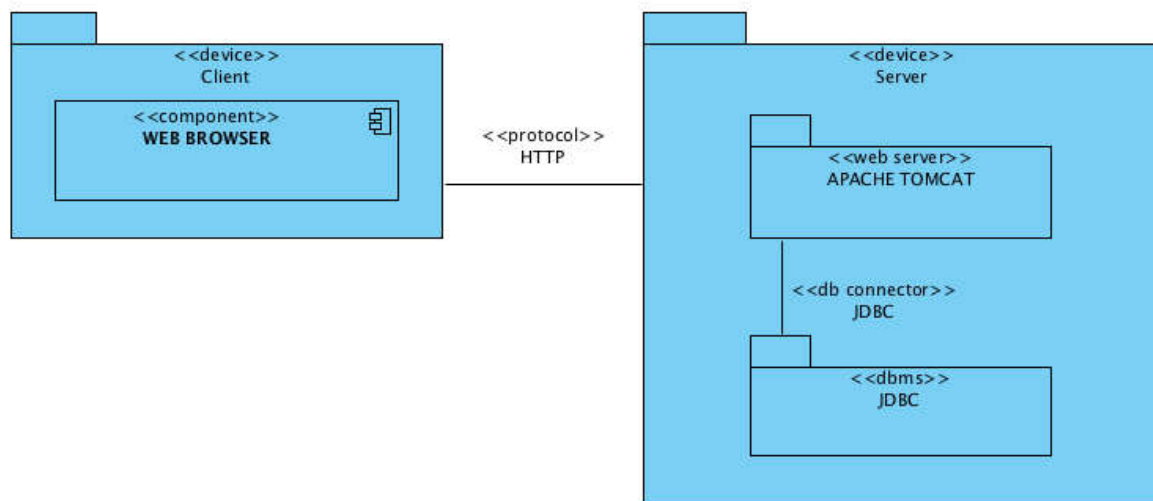


4.1.3. Diagramma di Deployment

L'utente richiede le funzionalità tramite un'interfaccia web. Questa costituirà, insieme ad un minimo di logica di business che consiste nel controllo della validità dei dati che l'utente può inserire, tutta la parte che verrà eseguita sul client. Per la parte server includerà la logica di business e la gestione della persistenza.

Sia il client che il server per il corretto funzionamento del sistema devono avere la possibilità di connettersi ad internet.

4.1.4. Mapping Hardware / Software



Il sistema che si vuole realizzare sarà installato su una macchina che fungerà da server. Questa dovrà includere un web server (con sufficiente spazio di memoria), in particolare Apache Tomcat ed un DBMS, useremo MySQL. Il client potrà essere un qualsiasi computer con un browser che supporta Javascript ed una connessione ad Internet.

5. Gestione dei dati persistenti

Nella gestione dei dati persistenti, si è convenuto salvare tutto all'interno di un database MySQL. Il vantaggio di usare un database è dovuto prima di tutto dal fatto che i dati sono logicamente organizzati, nonché dalla maggiore sicurezza garantita nel recupero di dati sensibili.

Il database sarà per la prima implementazione locale, ma se eventualmente si decidesse di renderlo remoto, ciò non costituirebbe un problema. Infatti, tramite l'uso del driver JDBC, il business layer scritto in Java sarà in grado di comunicare con il database, qualunque sia la sua locazione fisica.

La progettazione del database è stata effettuata in un documento esterno chiamato "SCHEMA E-R". Sostanzialmente si tratta della progettazione iniziale dello schema e-r, seguito da una ristrutturazione dove viene specificata solo quella parte di progetto che verrà effettivamente resa pubblica in un primo rilascio.

6. Gestione e controllo della sicurezza

Attori \ Sottosistemi	Gestione		
	Brani da approvare	Brani	Utenti
Manager	<ul style="list-style-type: none"> Visualizza brani da approvare Approva brani Elimina brani da approvare 	<ul style="list-style-type: none"> Ascolta brano 	<ul style="list-style-type: none"> Visualizza dati Visualizza brani
Amministratore		<ul style="list-style-type: none"> Ascolta brano 	<ul style="list-style-type: none"> Login Aggiungi manager Rimuovi manager Rimuovi cliente Visualizza dati Visualizza brani
Cliente	<ul style="list-style-type: none"> Aggiungi/Rimuovi brano da approvare Visualizza brani da approvare 	<ul style="list-style-type: none"> Carica brano Ascolta brano Modifica brano Visualizza brani Salva brano 	<ul style="list-style-type: none"> Visualizza dati Visualizza brani
Cliente non registrato		<ul style="list-style-type: none"> Visualizza brani Ascolta brani 	<ul style="list-style-type: none"> Effettua registrazione

7. Boundary condition

Use case name <u>Startup del sistema</u>					
Participating actors	Amministratore				
Entry conditions	<ul style="list-style-type: none">L'amministratore è connesso tramite una shell al server che ospita il web server				
Flow events	<table><tr><td>1. L'amministratore scrive in console <code>"/startup.sh"</code></td><td></td></tr><tr><td></td><td>0. Il sistema inizia l'esecuzione creando manager inizializzandoli.</td></tr></table>	1. L'amministratore scrive in console <code>"/startup.sh"</code>			0. Il sistema inizia l'esecuzione creando manager inizializzandoli.
1. L'amministratore scrive in console <code>"/startup.sh"</code>					
	0. Il sistema inizia l'esecuzione creando manager inizializzandoli.				
Exit conditions	<ul style="list-style-type: none">Il sito è raggiungibile da qualsiasi web browser connesso alla rete.				

Use case name <u>Shutdown del sistema</u>							
Participating actors	Amministratore						
Entry conditions	<ul style="list-style-type: none">L'amministratore è loggato in una qualsiasi pagina.L'amministratore ha cliccato sul pulsante di shutdown						
Flow events	<table><tr><td>1. L'amministratore clicca su ok per confermare l'azione</td><td></td></tr><tr><td></td><td>0. Il sistema notifica tutti gli utenti del sito che lo shutdown avverrà in 5 minuti.</td></tr><tr><td></td><td>0. Il sistema esegue lo script di shutdown di Tomcat.</td></tr></table>	1. L'amministratore clicca su ok per confermare l'azione			0. Il sistema notifica tutti gli utenti del sito che lo shutdown avverrà in 5 minuti.		0. Il sistema esegue lo script di shutdown di Tomcat.
1. L'amministratore clicca su ok per confermare l'azione							
	0. Il sistema notifica tutti gli utenti del sito che lo shutdown avverrà in 5 minuti.						
	0. Il sistema esegue lo script di shutdown di Tomcat.						
Exit conditions	<ul style="list-style-type: none">Il sito non è più raggiungibile.						

8. Servizi dei sottosistemi

8.1. Gestione brani in stallo

Sottosistema	Descrizione
Gestione Brani in stallo	Sottosistema che permette la gestione degli brani non ancora approvati

Servizio	Descrizione
ApprovaBrano	Servizio che permette al manager di approvare un brano.
UploadBrano	Servizio che permette al cliente di eseguire l'upload di un brano.
GetBraniStallo	Servizio che permette di visualizzare i brani da approva.

8.2. Gestione brani

Sottosistema	Descrizione
Gestione Brani.	Sottosistema che permette la modifica, l'upload e l'ascolto di brani.

Servizio	Descrizione
CercaBrano	Servizio che permette al cliente di ricercare brani nella libreria del sito, utilizzando parole chiave.
GetHomeBrani	Servizio che permette a tutti i clienti di visualizzare tutti i brani nella home.
GetAudio	Servizio che permette al cliente di riprodurre un brano.
ModificaBrano	Servizio che permette al gestore di modificare alcuni parametri di un determinato brano.
EliminaBrano	Servizio che permette al gestore di eliminare un proprio brano dalla libreria.
GetBraniSalvati	Servizio che permette ai clienti di visualizzare i propri brani salvati.
GetBraniUtente	Servizio che permette ai clienti di visualizzare i brani di un utente specifico.
SalvaBrano	Servizio che permette ai clienti di salvare un brano.
RimuoviBranoSalvato	Servizio che permette ai clienti di rimuovere un brano salvato.

8.3. Gestione utenti

Sottosistema	Descrizione
Gestione utenti.	Sottosistema che permette ad un utente di effettuare la login, ad un cliente non registrato di effettuare la registrazione, ad un cliente registrato di visualizzare il proprio profilo e gestire i dati personali, ad un amministratore di creare login gestori e rimuovere login.

Servizio	Descrizione
Login	Servizio che permette ad un cliente già registrato di effettuare l'accesso nel sistema.
Registrazione	Servizio che permette ad un cliente non registrato di creare una login per poter accedere nel sistema.
GetUtentiAdmin	Servizio che permette all'amministratore di visualizzare le informazioni inerenti ad un cliente.
EliminaUtente	Servizio che permette all'amministratore di rimuovere un utente.
Logout	Servizio che permette ad un cliente già registrato di effettuare l'uscita dal sistema.

9. Rilascio del sistema

L'approccio utilizzato nell'implementazione del software è di tipo incrementale. Ovvero saranno aggiunte funzionalità in ordine di priorità. L'ordine sarà il seguente:

1. Gestione brani, accesso al sito e gestione utenti.
2. Gestione brani salvati.
3. Pagina amministratore.