



Multimodal MP3 Music Player

for Android

Marco Crescentini
Francesco Maggi
Domenico Toscani

Indice

Introduzione	5
Obiettivi	5
Piano del progetto	6
Glossario	7
1.1 Introduzione	7
1.2 Glossario	8
Specifiche Dei Requisiti	9
2.1 Introduzione	9
2.2 Elenco dei requisiti	10
2.2.1 Requisiti Funzionali	10
2.3 Specifiche dei requisiti	11
Modello Dei Casi D'uso	16
3.1 Introduzione	16
3.2 Diagramma dei casi d'uso	17
3.3 Descrizione dei casi d'uso	18
Architettura Di Sistema	24
4.1 Introduzione	24
4.2 Deployment diagram	25
4.3 Descrizione del diagramma	26
4.3.1 Descrizione componenti hardware	26
4.3.2 Descrizione componenti software	26
Implementazione Architettura Di Sistema	28
5.1 Introduzione	28
5.2 Diagrammi delle classi	29
5.3 Modello dei package	29
5.4 Modello delle classi	30
5.4.1 Package activity	30

5.4.2 Package gesture	31
5.4.3 Package speech	31
5.5.1 SEQ_01	33
5.5.2 SEQ_02	34
5.5.3 SEQ_03	35
Progettazione, Implementazione E Valutazione Interfaccia	36
6.1 Introduzione	36
6.2 Realizzazione interfaccia sistema	37
6.2.1 Progettazione interfaccia astratta	39
6.2.2 Progettazione interfaccia concreta	42
6.3 Implementazione GUI	47
6.4 Valutazione GUI e euristiche di Nielsen	55
Interazione Multimodale	57
7.1 Introduzione	57
7.2 Modalità di interazione	58
7.2.1 Interazione vocale	58
7.2.2 Interazione gestuale	58
7.3 Tecnologie utilizzate	59
7.3.1 Scelta del sistema operativo	59
7.3.2 Implementazione del riconoscimento vocale	59
7.3.3 Implementazione del riconoscimento dei gesti	60
Conclusione E Sviluppi Futuri	61
8.1 Risultati ottenuti	62
8.2 Sviluppi futuri	63
Bibliografia	64

INTRODUZIONE

Da molti anni nel mondo dell'interazione uomo macchina si sta facendo sempre più attenzione a nuove modalità di comunicazione. Gli scopi di queste ricerche sono molteplici: ci si pone l'obiettivo di rendere la comunicazione tra uomo e macchina sempre più semplice, intuitiva e naturale ma anche di infrangere quei limiti che le normali modalità di interazione hanno e che possono rappresentare ostacoli insormontabili per persone portatrici di handicap. Due tra le modalità su cui si sta effettuando moltissima ricerca sono la "speech interaction" e la "gesture interaction". La speech interaction consiste nell'interagire con la macchina tramite comandi vocali. Può essere considerata una delle tipologie di interazioni più naturali e intuitive. Tale interazione si basa sulla "speech recognition", ossia il processo che consiste nel convertire flussi vocali in una serie di parole che verranno interpretate da un opportuno algoritmo e associate a determinati comandi. I principali problemi di tale modalità consistono nell'interpretare correttamente parole che potrebbero essere pronunciate in modo leggermente diverso o interpretarle in presenza di rumore di disturbo. La gesture interaction, invece, consiste nell'interagire con la macchina tramite comandi gestuali. Anche tale modalità è una tra le più naturali: basti pensare che nella vita quotidiana utilizziamo continuamente i gesti, ad esempio quando parliamo, con lo scopo di rafforzare il nostro discorso. Anche questa modalità richiede un processo di interpretazione del gesto; tramite dispositivi ottici il gesto viene catturato, riconosciuto e associato ad un preciso comando. I problemi che si possono riscontrare sono dovuti soprattutto a condizioni di illuminazione non ottimali, come ambienti troppo illuminati o bui, oppure a occlusioni parziali delle mani che potrebbero impedire alla macchina di recepire il gesto nella sua interezza.

Obiettivi

La nostra idea per questo progetto è stata quella di cercare di inserire in un'applicazione per dispositivi mobili le due modalità di interazione sopracitate. Tali modalità unite a quella touch, offrono all'utente un'ampia libertà nell'utilizzo del dispositivo e rendono l'applicazione utilizzabile in contesti in cui non è possibile interagire con il touch screen del dispositivo. La scelta su quale applicazione sviluppare è ricaduta su un lettore mp3. Di seguito elenchiamo gli obiettivi del nostro lavoro:

- Creare un app che permetta di riprodurre e gestire l'esecuzione di brani musicali
- Interagire con il lettore musicale tramite touch
- Interagire con il lettore musicale tramite comandi gestuali
- Interagire con il lettore musicale tramite comandi vocali



Piano del progetto

Nel capitolo 1, viene presentato il glossario in cui sono descritti alcuni termini usati durante la fase di progettazione del sistema e, se disponibili, forniti dei sinonimi.

Nel capitolo 2, vengono definiti i requisiti funzionali e non funzionali alla base del lavoro.

Nel capitolo 3, vengono tradotti i requisiti funzionali in casi d'uso, che forniranno una descrizione dettagliata dei servizi e delle funzionalità offerte dal programma.

Nel capitolo 4, viene presentata l'architettura del sistema, descrivendo le componenti di cui è composta.

Nel capitolo 5, viene mostrato come l'architettura del sistema è stata implementata, tramite un diagramma dei package e la descrizione delle classi che essi contengono. Vengono inoltre analizzati esempi tipici di interazione attraverso diagrammi di sequenza.

Nel capitolo 6, viene mostrato come è stata realizzata l'interfaccia del sistema a partire dalla sua progettazione astratta, passando poi per la progettazione concreta e finendo a mostrare gli screenshot della GUI. Infine vengono descritti esempi di uso del sistema tramite screenshot della GUI e viene fornita una loro valutazione tramite euristiche.

Nel capitolo 7, vengono spiegate dettagliatamente le modalità per l'interazione fra l'utente e l'applicazione, parlando della loro utilità e dei problemi riscontrati, oltre che alle soluzioni tecnologiche adottate per la loro implementazione e gestione.

Nel capitolo 8, vengono analizzati i risultati ottenuti alla fine del lavoro e individuati possibili sviluppi futuri.

DOCUMENTO 1

GLOSSARIO

1.1 Introduzione

In questo documento vengono spiegati e, dove possibile, indicati i sinonimi di alcuni termini utilizzati nella trattazione progettuale. Se nel resto della trattazione si avesse difficoltà a capire il significato di alcune parole, si invita alla consultazione di questo documento in modo da evitare problemi di ambiguità o di cattiva comprensione.



1.2 Glossario

TERMINI	DESCRIZIONE	SINONIMO
<i>Barra di riproduzione</i>	Elemento dell’interfaccia grafica, presente nella schermata di riproduzione, che mostra in ogni momento l’attuale punto di riproduzione del brano in esecuzione.	
<i>Interfaccia touch</i>	Elemento di un dispositivo attraverso il quale l’utente può interagire con l’interfaccia grafica dello stesso mediante le dita o altri particolari oggetti.	Touchscreen
<i>Modalità gesture</i>	Quando tale modalità è attiva, l’utente può effettuare alcune operazioni sull’applicazione eseguendo gesti della mano di fronte alla fotocamera del dispositivo.	Modalità riconoscimento gesti
<i>Modalità voice</i>	Quando tale modalità è attiva, l’utente può eseguire alcune operazioni sull’applicazione pronunciando comandi vocali vicino al microfono del dispositivo.	Modalità riconoscimento vocale
<i>Playlist</i>	Lista con i nomi di tutti i brani disponibili per la riproduzione.	
<i>Repeat</i>	Quando tale modalità è attiva, al termine della riproduzione di un brano viene eseguito nuovamente lo stesso fino a quando l’utente non disattiva questa opzione o cambia brano.	
<i>Schermata di riproduzione</i>	Schermata principale dell’applicazione, visualizzata dall’utente quando questa viene lanciata.	
<i>Schermata playlist</i>	Schermata dell’applicazione contenente la playlist; vi si accede dalla schermata di riproduzione.	
<i>Shuffle</i>	Quando tale modalità è attiva, il brano precedente o successivo a quello in riproduzione viene scelto in modo casuale dalla playlist.	Modalità di riproduzione casuale

DOCUMENTO 2

SPECIFICHE DEI REQUISITI

2.1 Introduzione

In questo documento si esplicitano in modo formale e dettagliato i requisiti. Questi vengono identificati univocamente dal codice:

REQ_<tipo>_<numero>

dove con tipo si indica a suddivisione dei requisiti in:

- **Requisiti Funzionali:** indicano le funzionalità del programma e si dividono in requisiti funzionali utente (**FU**) e requisiti funzionali di sistema (**FS**)
- **Requisiti Non Funzionali:** Indicati con il tipo **NF**, rappresentano vincoli sui servizi offerti.



2.2 Elenco dei requisiti

Segue un elenco di tutti i requisiti che vengono descritti in dettaglio nella sezione 2.3. La lista presenta il codice identificativo del requisito (vedere sezione precedente).

2.2.1 Requisiti Funzionali

IDENTIFICATIVO	NOME
REQ_FU_01	Riproduzione brano
REQ_FU_02	Visualizzazione playlist
REQ_FU_03	Selezione brano
REQ_FU_04	Brano precedente/successivo
REQ_FU_05	Gestione progresso
REQ_FU_06	Gestione Volume
REQ_FU_07	Riproduzione casuale
REQ_FU_08	Ripetizione brano
REQ_FU_09	Attivazione modalità voce
REQ_FU_10	Attivazione modalità gesture
REQ_FS_01	Multimodalità
REQ_FS_02	Informazioni brano
REQ_FS_03	Informazioni volume
REQ_FS_04	Informazioni esecuzione
REQ_FS_05	Informazioni modalità di interazione
REQ_FS_06	Orientamento dispositivo

2.2.2 Requisiti Non Funzionali

IDENTIFICATIVO	NOME
REQ_NF_01	Portabilità
REQ_NF_02	Usabilità
REQ_NF_03	Efficienza
REQ_NF_04	Evolvibilità
REQ_NF_05	Standard



2.3 Specifiche dei requisiti

I requisiti vengono classificati in base al livello di priorità nel seguente modo:

- **Obbligatorio:** Il sistema richiede assolutamente questo vincolo per poter essere utilizzato appropriatamente.
- **Raccomandato:** Il sistema non ha strettamente bisogno di questo requisito ma può utilizzarlo per ottenere importanti benefici.

IDENTIFICATIVO	REQ_FU_01
NOME	Riproduzione brano
TIPO	Requisito funzionale utente
PRIORITÀ	Obbligatorio
DESCRIZIONE	L'utente deve essere in grado di avviare e mettere in pausa la riproduzione di un determinato brano

IDENTIFICATIVO	REQ_FU_02
NOME	Visualizzazione playlist
TIPO	Requisito funzionale utente
PRIORITÀ	Obbligatorio
DESCRIZIONE	L'utente deve poter visualizzare l'elenco dei brani riproducibili dall'applicazione

IDENTIFICATIVO	REQ_FU_03
NOME	Selezione brano
TIPO	Requisito funzionale utente
PRIORITÀ	Obbligatorio
DESCRIZIONE	L'utente deve essere in grado di riprodurre i brani selezionandoli dalla playlist

2. SPECIFICHE DEI REQUISITI



IDENTIFICATIVO	REQ_FU_04
NOME	Brano precedente/successivo
TIPO	Requisito funzionale utente
PRIORITÀ	Obbligatorio
DESCRIZIONE	L'utente deve poter avviare direttamente il brano successivo o precedente a quello in riproduzione

IDENTIFICATIVO	REQ_FU_05
NOME	Gestione progresso
TIPO	Requisito funzionale utente
PRIORITÀ	Obbligatorio
DESCRIZIONE	L'utente deve potersi muovere liberamente all'interno di un brano in modo da modificarne il punto di riproduzione

IDENTIFICATIVO	REQ_FU_06
NOME	Gestione volume
TIPO	Requisito funzionale utente
PRIORITÀ	Obbligatorio
DESCRIZIONE	L'utente deve poter modificare a proprio piacimento il livello del volume seguendo una scala specifica

IDENTIFICATIVO	REQ_FU_07
NOME	Riproduzione casuale
TIPO	Requisito funzionale utente
PRIORITÀ	Raccomandato
DESCRIZIONE	L'utente deve poter riprodurre casualmente un qualsiasi brano della sua playlist

IDENTIFICATIVO	REQ_FU_08
NOME	Ripetizione brano
TIPO	Requisito funzionale utente
PRIORITÀ	Raccomandato
DESCRIZIONE	L'utente deve poter ripetere automaticamente il brano in esecuzione una volta terminato

2. SPECIFICHE DEI REQUISITI



IDENTIFICATIVO	REQ_FU_09
NOME	Attivazione modalità voce
TIPO	Requisito funzionale utente
PRIORITÀ	Obbligatorio
DESCRIZIONE	L'utente deve essere in grado di attivare e disattivare la modalità per il riconoscimento vocale

IDENTIFICATIVO	REQ_FU_10
NOME	Attivazione modalità gesture
TIPO	Requisito funzionale utente
PRIORITÀ	Obbligatorio
DESCRIZIONE	L'utente deve essere in grado di attivare e disattivare la modalità per il riconoscimento dei gesti

IDENTIFICATIVO	REQ_FS_01
NOME	Multimodalità
TIPO	Requisito funzionale di sistema
PRIORITÀ	Obbligatorio
DESCRIZIONE	Il sistema deve fornire modalità di interazione basate su touch, voce e hand gesture

IDENTIFICATIVO	REQ_FS_02
NOME	Informazioni brano
TIPO	Requisito funzionale di sistema
PRIORITÀ	Obbligatorio
DESCRIZIONE	Il sistema deve mostrare l'artista e il titolo del brano in esecuzione

IDENTIFICATIVO	REQ_FS_03
NOME	Informazioni volume
TIPO	Requisito funzionale di sistema
PRIORITÀ	Obbligatorio
DESCRIZIONE	Il sistema deve poter mostrare l'attuale livello del volume

2. SPECIFICHE DEI REQUISITI



IDENTIFICATIVO	REQ_FS_04
NOME	Informazioni esecuzione
TIPO	Requisito funzionale di sistema
PRIORITÀ	Obbligatorio
DESCRIZIONE	Il sistema deve informare l'utente sullo stato dell'esecuzione del brano, includendo il punto attuale della riproduzione

IDENTIFICATIVO	REQ_FS_05
NOME	Informazioni modalità di interazione
TIPO	Requisito funzionale di sistema
PRIORITÀ	Obbligatorio
DESCRIZIONE	Il sistema deve mostrare quali modalità di interazione sono attive

IDENTIFICATIVO	REQ_FS_06
NOME	Orientamento dispositivo
TIPO	Requisito funzionale di sistema
PRIORITÀ	Raccomandato
DESCRIZIONE	Il sistema deve riconoscere l'orientamento del dispositivo in modo da gestire le diverse funzionalità

IDENTIFICATIVO	REQ_NF_01
NOME	Portabilità
TIPO	Requisito non funzionale
PRIORITÀ	Obbligatorio
DESCRIZIONE	L'applicazione deve funzionare su dispositivi con sistema operativo Android 4.1 o superiore

IDENTIFICATIVO	REQ_NF_03
NOME	Efficienza
TIPO	Requisito non funzionale
PRIORITÀ	Raccomandato
DESCRIZIONE	Il numero di comandi di interazione deve essere minimo

2. SPECIFICHE DEI REQUISITI



IDENTIFICATIVO	REQ_NF_04
NOME	Evolvibilità
TIPO	Requisito non funzionale
PRIORITÀ	Raccomandato
DESCRIZIONE	L'applicazione deve essere composta da un'architettura modulare

IDENTIFICATIVO	REQ_NF_05
NOME	Standard
TIPO	Requisito non funzionale
PRIORITÀ	Obbligatorio
DESCRIZIONE	L'applicazione deve essere in grado di riprodurre file in formato .mp3

DOCUMENTO 3

MODELLO DEI CASI D'USO

3.1 Introduzione

L'obiettivo principale di questo documento è presentare, a partire dai requisiti funzionali e attraverso specifiche testuali e diagrammi UML, i diversi casi d'uso e come gli utenti interagiscono con il sistema.

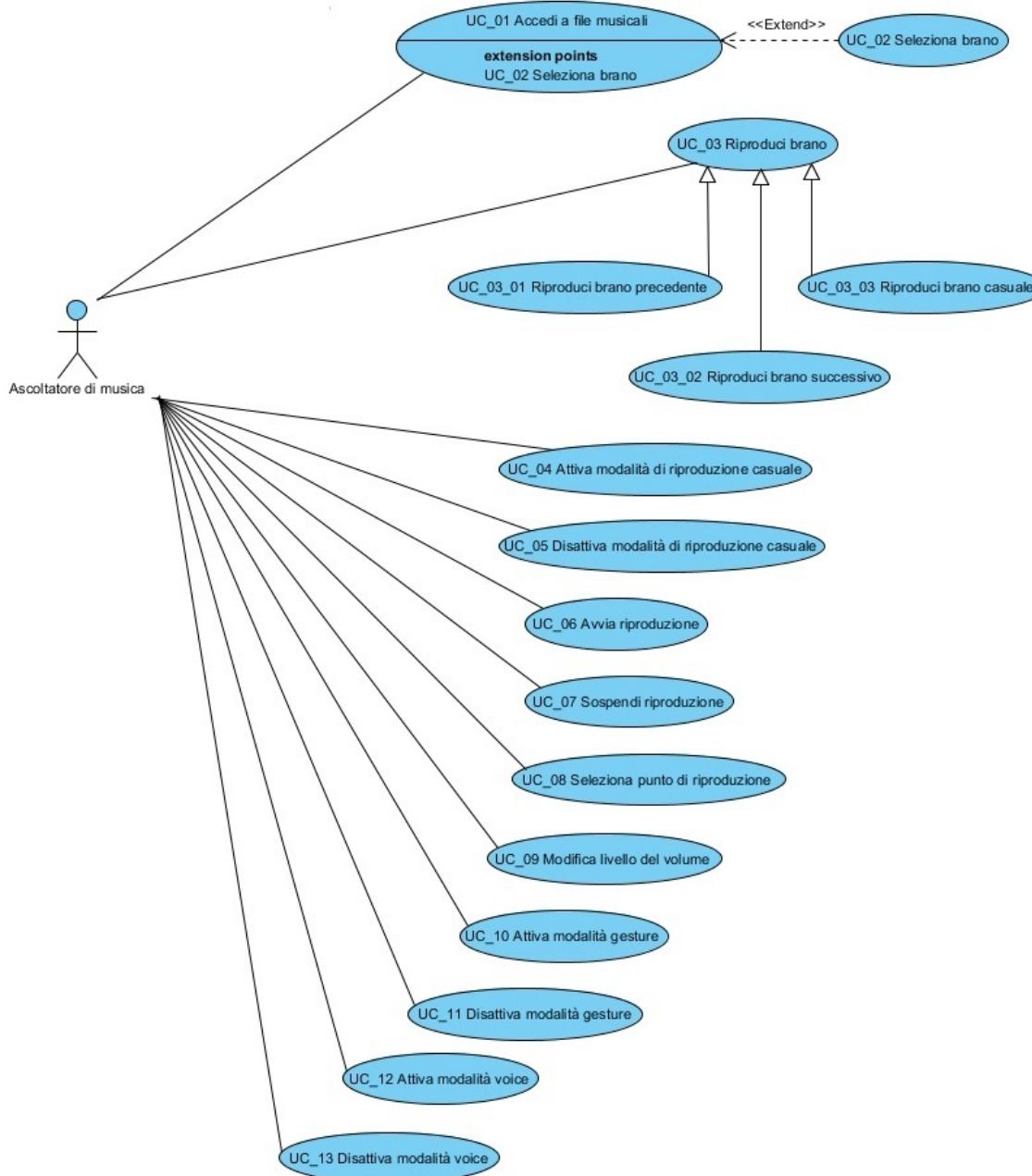
I casi d'uso sono identificati dal codice:

UC_<numero caso d'uso>_[numero sotto caso d'uso]



dove il numero che definisce il sotto caso d'uso è opzionale.

3.2 Diagramma dei casi d'uso





3.3 Descrizione dei casi d'uso

IDENTIFICATIVO	UC_01	
NOME	Accedi a file musicali	
OBIETTIVO	l'utente vuole accedere ai file musicali	
PRECONDIZIONI	<ul style="list-style-type: none"> l'applicazione è stata avviata 	
FLUSSO PRINCIPALE	UTENTE I. accede alla funzionalità "mostra playlist"	SISTEMA II. mostra la lista dei brani ordinati alfabeticamente III. extends "Seleziona brano" (UC_02)
POSTCONDIZIONI	<ul style="list-style-type: none"> il sistema mostra la lista dei brani presenti nel dispositivo 	

IDENTIFICATIVO	UC_02	
NOME	Seleziona brano	
OBIETTIVO	l'utente vuole riprodurre un brano presente nella playlist	
PRECONDIZIONE	<ul style="list-style-type: none"> il sistema sta mostrando la lista dei brani presenti nel dispositivo la lista dei brani non è vuota 	
FLUSSO PRINCIPALE	UTENTE I. seleziona un brano dalla playlist	SISTEMA II. riproduce il brano selezionato
POSTCONDIZIONI	<ul style="list-style-type: none"> il brano selezionato è in riproduzione la playlist viene chiusa e il sistema mostra la schermata di riproduzione 	

IDENTIFICATIVO	UC_03	
NOME	Riproduci brano	
OBIETTIVO	l'utente vuole riprodurre un nuovo brano	
DESCRIZIONE	<ul style="list-style-type: none"> l'utente vuole riprodurre una nuova canzone. Il caso d'uso si specializza in "Riproduci precedente" (UC_03_01), "Riproduci successivo" (UC_03_02), "Riproduci casuale" (UC_03_03). 	

3. MODELLO DEI CASI D'USO



IDENTIFICATIVO	UC_03_01					
NOME	Riproduci precedente					
OBIETTIVO	l'utente vuole riprodurre il brano precedente a quello in esecuzione nella playlist					
PRECONDIZIONE	<ul style="list-style-type: none"> il sistema sta mostrando la schermata di riproduzione la modalità di riproduzione casuale non è attiva 					
FLUSSO PRINCIPALE	<table border="1"> <thead> <tr> <th>UTENTE</th> <th>SISTEMA</th> </tr> </thead> <tbody> <tr> <td>I. avvia la funzionalità “riproduzione del brano precedente”</td> <td>II. riproduce il brano precedente a quello selezionato</td> </tr> </tbody> </table>	UTENTE	SISTEMA	I. avvia la funzionalità “riproduzione del brano precedente”	II. riproduce il brano precedente a quello selezionato	
UTENTE	SISTEMA					
I. avvia la funzionalità “riproduzione del brano precedente”	II. riproduce il brano precedente a quello selezionato					
POSTCONDIZIONI	• il brano precedente a quello selezionato è in riproduzione					

IDENTIFICATIVO	UC_03_02					
NOME	Riproduci successivo					
OBIETTIVO	l'utente vuole riprodurre il brano successivo a quello in esecuzione nella playlist					
PRECONDIZIONE	<ul style="list-style-type: none"> il sistema sta mostrando la schermata di riproduzione la modalità di riproduzione casuale non è attiva 					
FLUSSO PRINCIPALE	<table border="1"> <thead> <tr> <th>UTENTE</th> <th>SISTEMA</th> </tr> </thead> <tbody> <tr> <td>I. avvia la funzionalità “riproduzione del brano successivo”</td> <td>II. riproduce il brano successivo a quello selezionato</td> </tr> </tbody> </table>	UTENTE	SISTEMA	I. avvia la funzionalità “riproduzione del brano successivo”	II. riproduce il brano successivo a quello selezionato	
UTENTE	SISTEMA					
I. avvia la funzionalità “riproduzione del brano successivo”	II. riproduce il brano successivo a quello selezionato					
POSTCONDIZIONI	• il brano successivo è in riproduzione					

IDENTIFICATIVO	UC_03_03					
NOME	Riproduci brano casuale					
OBIETTIVO	l'utente vuole riprodurre un brano casuale					
PRECONDIZIONE	<ul style="list-style-type: none"> il sistema sta mostrando la schermata di riproduzione la modalità di riproduzione casuale è attiva 					
FLUSSO PRINCIPALE	<table border="1"> <thead> <tr> <th>UTENTE</th> <th>SISTEMA</th> </tr> </thead> <tbody> <tr> <td>I. avvia la funzionalità “riproduzione casuale di un brano”</td> <td>II. riproduce un brano casualmente</td> </tr> </tbody> </table>	UTENTE	SISTEMA	I. avvia la funzionalità “riproduzione casuale di un brano”	II. riproduce un brano casualmente	
UTENTE	SISTEMA					
I. avvia la funzionalità “riproduzione casuale di un brano”	II. riproduce un brano casualmente					
POSTCONDIZIONI	• il brano viene riprodotto					

3. MODELLO DEI CASI D'USO



IDENTIFICATIVO	UC_04	
NOME	Attiva modalità di riproduzione casuale	
OBIETTIVO	l'utente vuole attivare la modalità di riproduzione casuale	
PRECONDIZIONE	<ul style="list-style-type: none"> il sistema sta mostrando la schermata di riproduzione la modalità di riproduzione casuale non è attiva 	
FLUSSO PRINCIPALE	UTENTE I. richiede l'attivazione della modalità "riproduzione casuale"	SISTEMA II. attiva la modalità "riproduzione casuale"
POSTCONDIZIONI	<ul style="list-style-type: none"> la modalità di riproduzione casuale è attiva 	

IDENTIFICATIVO	UC_05	
NOME	Disattiva modalità di riproduzione casuale	
OBIETTIVO	l'utente vuole disattivare la modalità di riproduzione casuale	
PRECONDIZIONE	<ul style="list-style-type: none"> il sistema sta mostrando la schermata di riproduzione la modalità di riproduzione casuale è attiva 	
FLUSSO PRINCIPALE	UTENTE I. richiede la disattivazione della modalità "riproduzione casuale"	SISTEMA II. disattiva la modalità "riproduzione casuale"
POSTCONDIZIONI	<ul style="list-style-type: none"> la modalità di riproduzione casuale è disattiva 	

IDENTIFICATIVO	UC_06	
NOME	Avvia riproduzione	
OBIETTIVO	l'utente vuole avviare la riproduzione del brano corrente	
PRECONDIZIONE	<ul style="list-style-type: none"> il sistema sta mostrando la schermata di riproduzione Il brano corrente è in pausa 	
FLUSSO PRINCIPALE	UTENTE I. richiede la riproduzione del brano corrente	SISTEMA II. riproduce il brano corrente
POSTCONDIZIONI	<ul style="list-style-type: none"> il brano corrente è riprodotto 	

3. MODELLO DEI CASI D'USO



IDENTIFICATIVO	UC_07	
NOME	Sospendi riproduzione	
OBIETTIVO	l'utente vuole sospendere la riproduzione del brano corrente	
PRECONDIZIONE	<ul style="list-style-type: none"> il sistema sta mostrando la schermata di riproduzione Il brano corrente è in riproduzione 	
FLUSSO PRINCIPALE	UTENTE I. richiede la sospensione del brano corrente	SISTEMA II. sospende il brano corrente
POSTCONDIZIONI	<ul style="list-style-type: none"> il brano corrente è messo in pausa 	

IDENTIFICATIVO	UC_08	
NOME	Seleziona punto di riproduzione	
OBIETTIVO	l'utente vuole selezionare un nuovo punto di riproduzione del brano corrente	
PRECONDIZIONE	<ul style="list-style-type: none"> il sistema sta mostrando la schermata di riproduzione 	
FLUSSO PRINCIPALE	UTENTE I. seleziona un nuovo punto di riproduzione del brano corrente	SISTEMA II. sposta la riproduzione del brano al punto selezionato
POSTCONDIZIONI	<ul style="list-style-type: none"> la riproduzione del brano è stata spostata al punto selezionato 	

IDENTIFICATIVO	UC_09	
NOME	Modifica livello del volume	
OBIETTIVO	l'utente vuole modificare il livello del volume	
PRECONDIZIONE	<ul style="list-style-type: none"> l'applicazione è stata avviata 	
FLUSSO PRINCIPALE	UTENTE I. richiede la modifica del livello del volume	SISTEMA II. aggiorna il livello del volume a quello richiesto
POSTCONDIZIONI	<ul style="list-style-type: none"> il livello del volume è stato modificato 	

3. MODELLO DEI CASI D'USO



IDENTIFICATIVO	UC_10	
NOME	Attiva modalità gesture	
OBIETTIVO	l'utente vuole attivare la modalità di riconoscimento dei gesti	
PRECONDIZIONE	<ul style="list-style-type: none"> il sistema sta mostrando la schermata di riproduzione la modalità gesture non è attiva 	
FLUSSO PRINCIPALE	UTENTE I. richiede l'attivazione della modalità "riconoscimento dei gesti"	SISTEMA II. attiva la modalità "riconoscimento dei gesti"
POSTCONDIZIONI	<ul style="list-style-type: none"> la modalità di riconoscimento dei gesti è attiva 	

IDENTIFICATIVO	UC_11	
NOME	Disattiva modalità gesture	
OBIETTIVO	l'utente vuole disattivare la modalità di riconoscimento dei gesti	
PRECONDIZIONE	<ul style="list-style-type: none"> il sistema sta mostrando la schermata di riproduzione la modalità gestire è attiva 	
FLUSSO PRINCIPALE	UTENTE I. richiede la disattivazione della modalità "riconoscimento dei gesti"	SISTEMA II. disattiva la modalità "riconoscimento dei gesti"
POSTCONDIZIONI	<ul style="list-style-type: none"> la modalità di riconoscimento dei gesti non è attiva 	

IDENTIFICATIVO	UC_12	
NOME	Attiva modalità voice	
OBIETTIVO	l'utente vuole attivare la modalità di riconoscimento dei comandi vocali	
PRECONDIZIONE	<ul style="list-style-type: none"> il sistema sta mostrando la schermata di riproduzione la modalità voce non è attiva 	
FLUSSO PRINCIPALE	UTENTE I. richiede l'attivazione della modalità "riconoscimento dei comandi vocali"	SISTEMA II. attiva la modalità "riconoscimento dei comandi vocali"
POSTCONDIZIONI	<ul style="list-style-type: none"> la modalità di riconoscimento dei comandi vocali è attiva 	

3. MODELLO DEI CASI D'USO



IDENTIFICATIVO	UC_13	
NOME	Disattiva modalità voice	
OBIETTIVO	l'utente vuole disattivare la modalità di riconoscimento dei comandi vocali	
PRECONDIZIONE	<ul style="list-style-type: none">il sistema sta mostrando la schermata di riproduzionela modalità voce è attiva	
FLUSSO PRINCIPALE	UTENTE	SISTEMA
	I. richiede la disattivazione della modalità “riconoscimento dei comandi vocali”	II. disattiva la modalità “riconoscimento dei comandi vocali”
POSTCONDIZIONI	<ul style="list-style-type: none">la modalità di riconoscimento dei comandi vocali non è attiva	

DOCUMENTO 4

ARCHITETTURA DI SISTEMA

4.1 Introduzione

In questo capitolo viene descritto come l'architettura del sistema è stata implementata tramite un deployment diagram. Di seguito viene mostrato il diagramma e fornita una breve descrizione delle componenti che lo costituiscono.



4.2 Deployment diagram

Il deployment diagram è un diagramma previsto dal linguaggio UML per descrivere il sistema in termini di risorse hardware e di relazioni fra di esse [1]. Nel nostro caso tale diagramma è stato unito al component diagram per mostrare come le componenti software siano distribuite rispetto alle risorse fisiche, e come esse comunichino tra loro e con l'hardware a disposizione. Nel diagramma successivo le risorse fisiche sono rappresentate da parallelepipedi di colore giallo, mentre le risorse software con rettangoli di colore azzurro. Le componenti software comunicano tra loro in termini di interfacce "provided" fornite, e "required" richieste [2]. Le interfacce fornite rappresentano le funzioni o i servizi offerti dalla componente e sono rappresentate con un cerchio, mentre quelle richieste definiscono le funzioni, o i servizi presenti in altre componenti, di cui la componente necessita per compiere il proprio lavoro, e sono rappresentate da un semicerchio aperto. Le componenti software possono comunicare con componenti hardware tramite differenti relazioni. Nel diagramma è stata usata la relazione "use" semplicemente per indicare che una componente utilizza l'hardware.

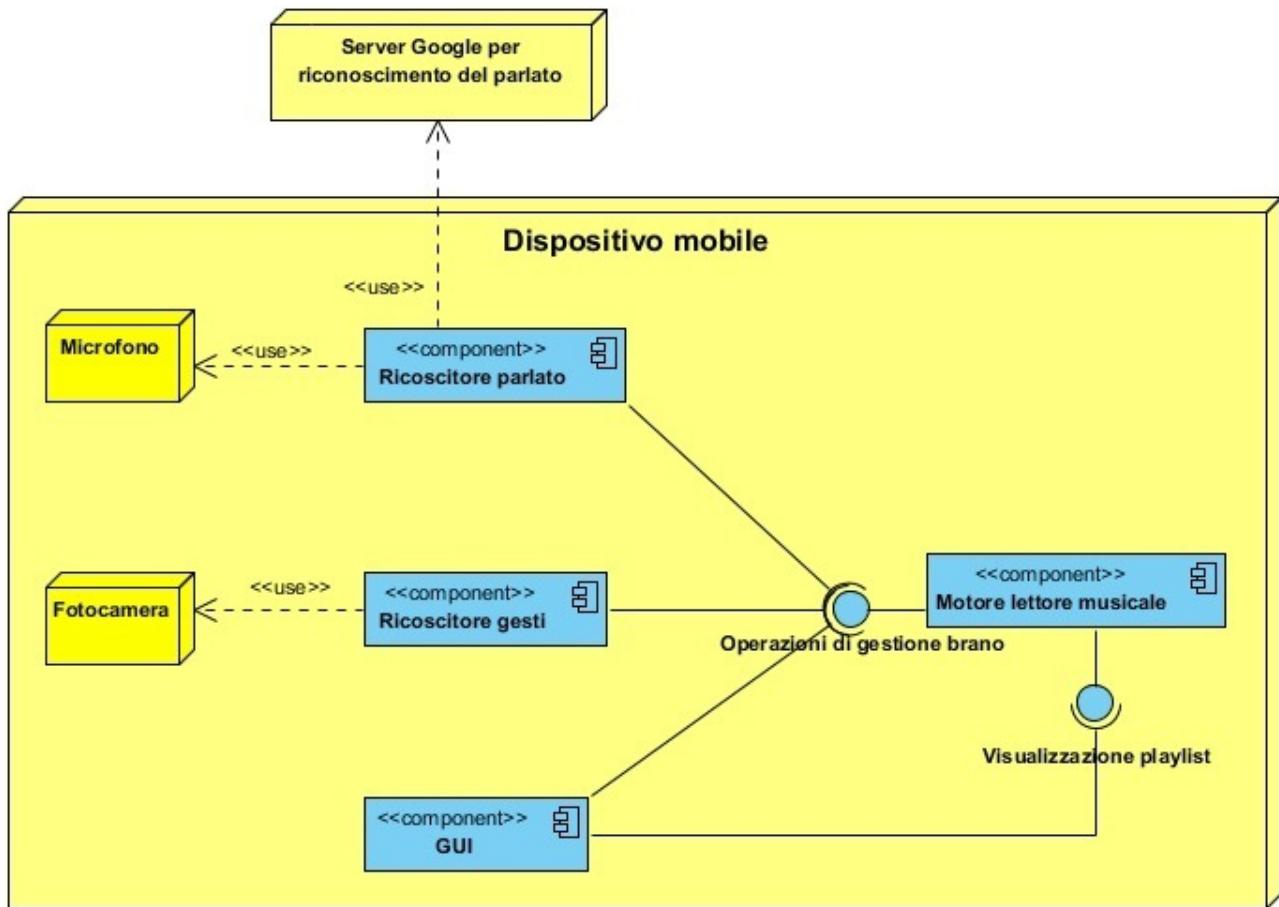


Figura 4.1: Deployment diagram



4.3 Descrizione del diagramma

In questo paragrafo viene data una descrizione dettagliata del deployment diagramma

4.3.1 Descrizione componenti hardware

Di seguito viene fornita una descrizione delle risorse hardware.

- **Dispositivo mobile:** È il dispositivo sul quale è installata l'app. Le componenti software all'interno della risorsa costituiscono le componenti dell'applicazione. Del dispositivo, in particolare, utilizzeremo le componenti hardware "Fotocamera" e "Microfono" che verranno descritte in dettaglio di seguito.
- **Fotocamera:** È la fotocamera frontale del dispositivo; è utilizzata dall'app, in particolare dalla componente "Riconoscitore gesti", per captare eventuali comandi gestuali da parte dell'utente.
- **Microfono:** È il microfono del dispositivo; è utilizzata dall'app, in particolare dalla componente "Riconoscitore parlato", per captare eventuali comandi vocali da parte dell'utente.
- **Server Google per riconoscimento del parlato:** È il server di Google che riceve in input parole (da usare un altro termine magari token vocale?) e restituisce un insieme di stringhe che rappresentano i possibili match. È utilizzato dalla componente "Riconoscitore parlato".

4.3.2 Descrizione componenti software

- **Motore lettore musicale:** Rappresenta il cuore dell'applicazione. Tale componente contiene tutte le funzioni che permettono di eseguire operazioni sui brani, ad esempio esegui brano, esegui brano successivo, metti in pausa, riprendi esecuzione ecc., e contiene anche le funzioni che permettono di accedere alla playlist. Fornisce due interfacce "Operazioni di gestione brano" e "Visualizzazione playlist", che offrono l'accesso alle funzioni sopracitate.
- **GUI:** Rappresenta l'interfaccia dell'applicazione. Tale componente funge da tramite tra l'utente e le varie componenti del sistema, infatti ogni volta che l'utente interagisce con l'interfaccia touch, ad esempio toccando un pulsante, la GUI comunica con una componente chiedendogli di eseguire una particolare funzione. Di seguito vengono elencate le componenti con cui la GUI comunica e il motivo di tale comunicazione:
 - **Motore lettore musicale:** Quando l'utente, tramite l'interfaccia touch, accede a funzionalità riguardanti la gestione dei brani, la GUI comunicherà con tale componente tramite le interfacce richieste "Operazioni di gestione brano" e "Visualizzazione playlist", e accederà a funzioni che si occuperanno di eseguire tali operazioni.



- **Riconoscitore parlato:** Rappresenta la componente che si occupa di riconoscere e elaborare eventuali comandi vocali impartiti dall'utente. Di seguito vengono elencate le componenti (software/hardware) con cui il riconoscitore parlato comunica e il motivo di tale comunicazione:
 - **Microfono:** Il riconoscitore utilizza il microfono del dispositivo per captare la voce dell'utente.
 - **Server Google per riconoscimento del parlato:** Una volta captata la voce dell'utente, la componente invia una richiesta al server Google che si occupa di trasformare ciò che è stato recepito in un insieme di possibili match di parole che le vengono restituiti.
 - **Motore lettore musicale:** Il riconoscitore vocale elabora l'insieme di match ricevuto, cercando una parola chiave corrispondente ad un comando. Una volta che il comando è stato individuato accede, tramite l'interfaccia richiesta "Operazioni di gestione brano", alle funzioni offerte dal motore lettore musicale per eseguire l'operazione associata al comando. (Per una spiegazione dettagliata della modalità di interazione vocale si rimanda al capitolo 7)
- **Riconoscitore gesti:** Rappresenta la componente che si occupa di riconoscere e elaborare eventuali comandi gestuali impartiti dall'utente. Di seguito vengono elencate le componenti (software/hardware) con cui il riconoscitore gesti comunica e il motivo di tale comunicazione:
 - **Fotocamera:** Il riconoscitore utilizza la fotocamera frontale del dispositivo per captare i movimenti dell'utente e riconoscere eventuali comandi.
 - **Motore lettore musicale:** Una volta che il comando è stato riconosciuto, il riconoscitore accede, tramite l'interfaccia richiesta "Operazioni di gestione brano", alle funzioni offerte dal motore lettore musicale per eseguire l'operazione associata al comando. (Per una spiegazione dettagliata della modalità di interazione vocale si rimanda al capitolo 7)

DOCUMENTO 5

IMPLEMENTAZIONE ARCHITETTURA DI SISTEMA

5.1 Introduzione

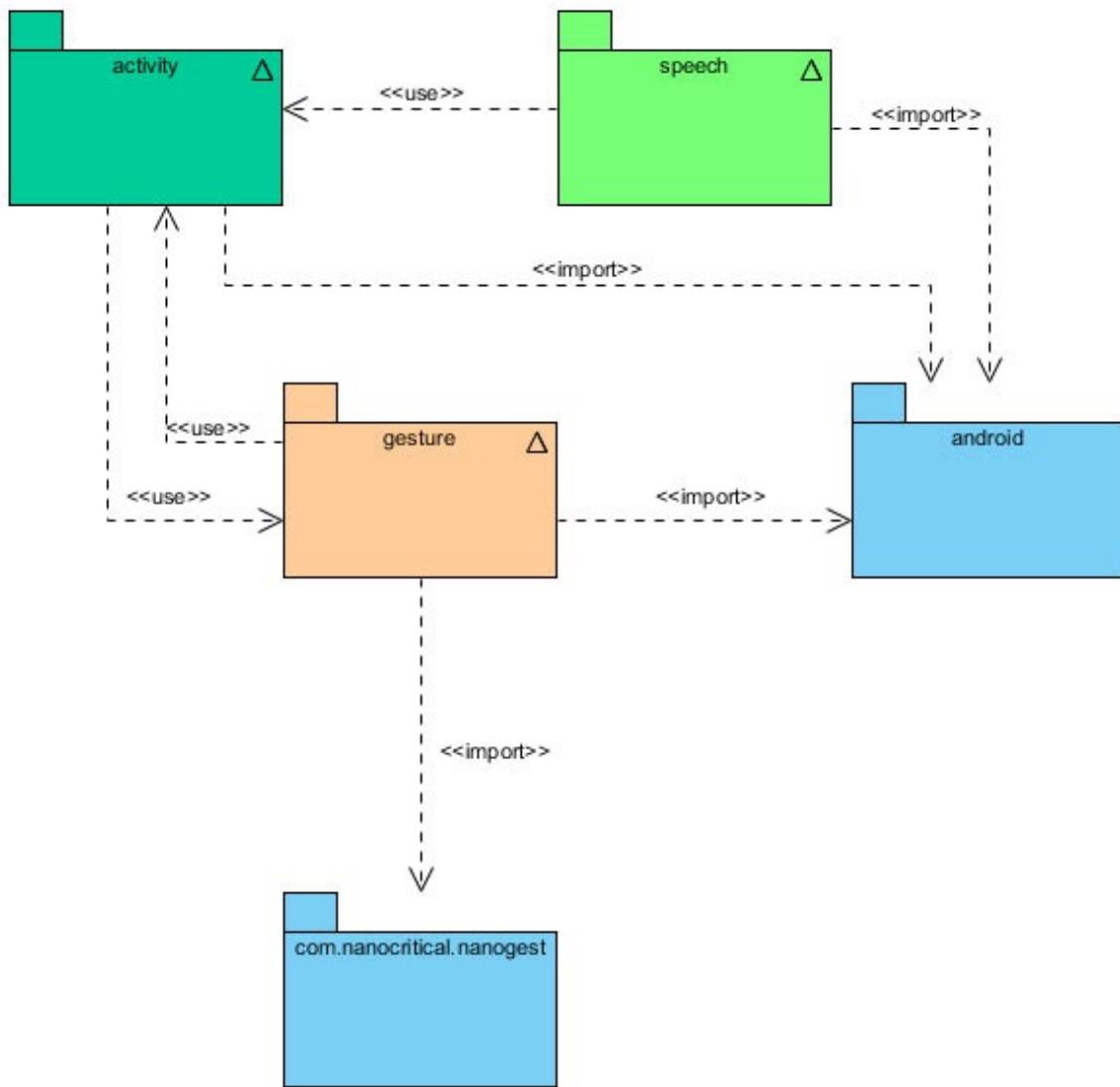
L'obiettivo principale di questo documento è dare una descrizione dettagliata dell'attività di design svolta durante la realizzazione del progetto. In questo documento si proporranno soluzioni e approcci concreti, trattando l'implementazione tecnica dell'applicazione. In particolare, si renderà definitiva l'architettura del sistema attraverso i diagrammi delle classi e i diagrammi di sequenza



5.2 Diagrammi delle classi

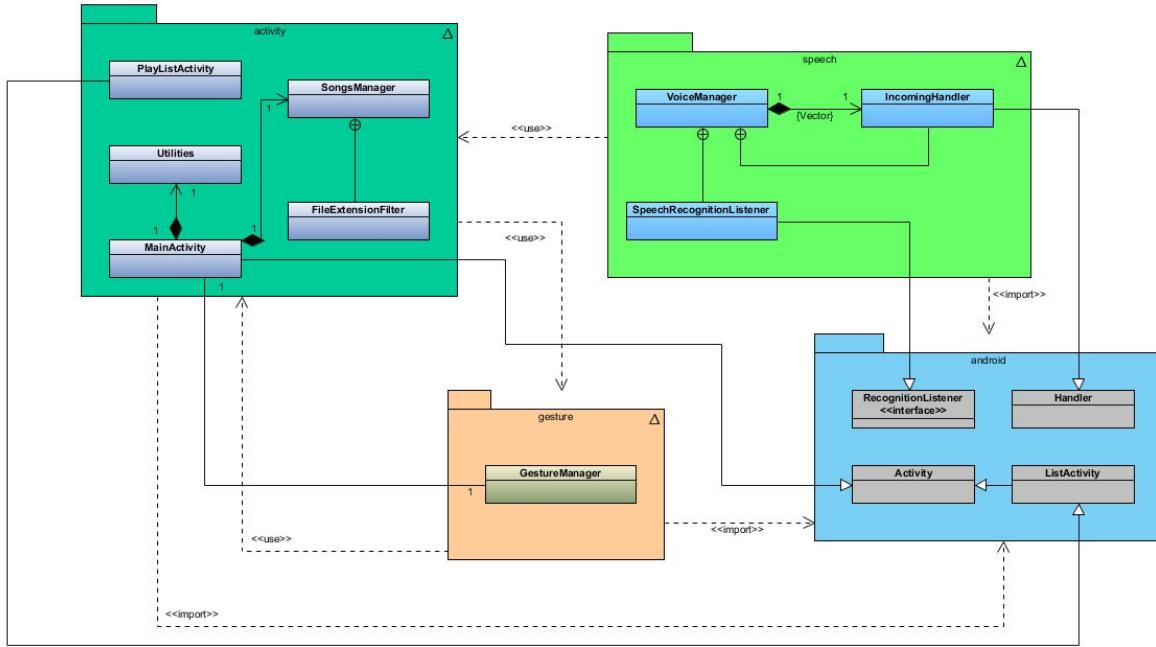
In questa sezione vengono mostrati i package e le classi utilizzati in questo progetto. Per la descrizione delle classi viene utilizzata una scrittura semplice e ad alto livello.

5.3 Modello dei package

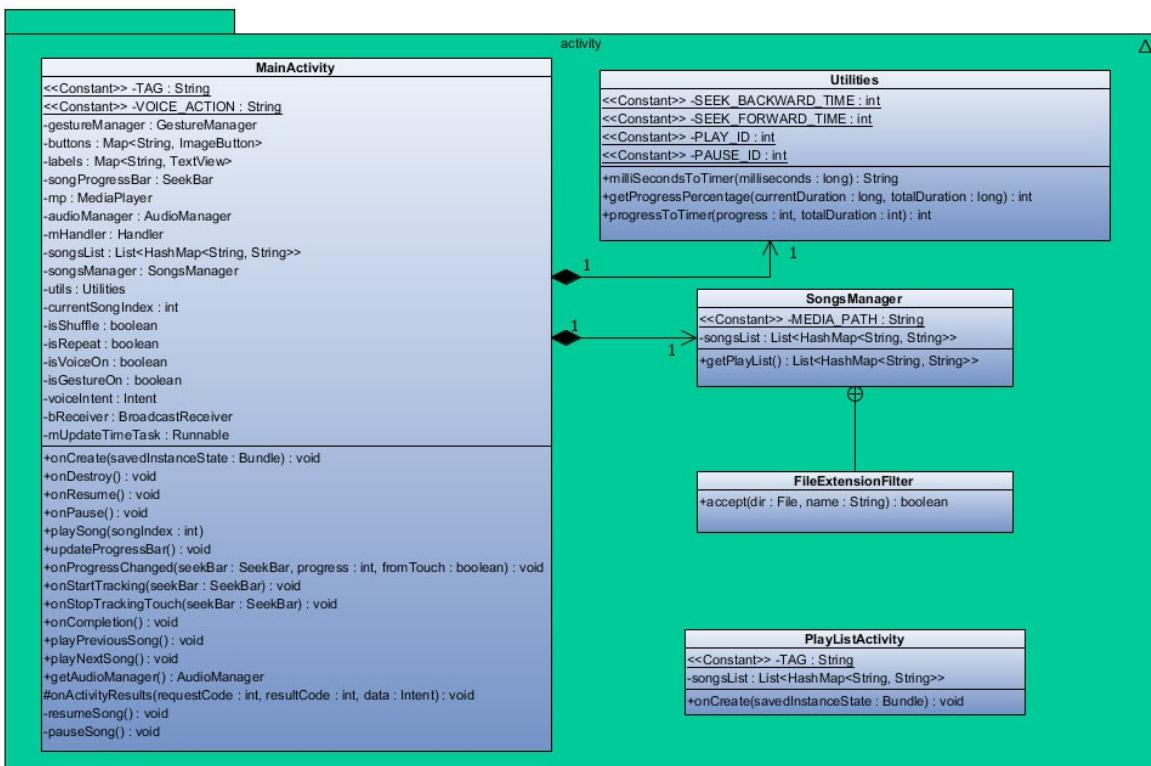




5.4 Modello delle classi

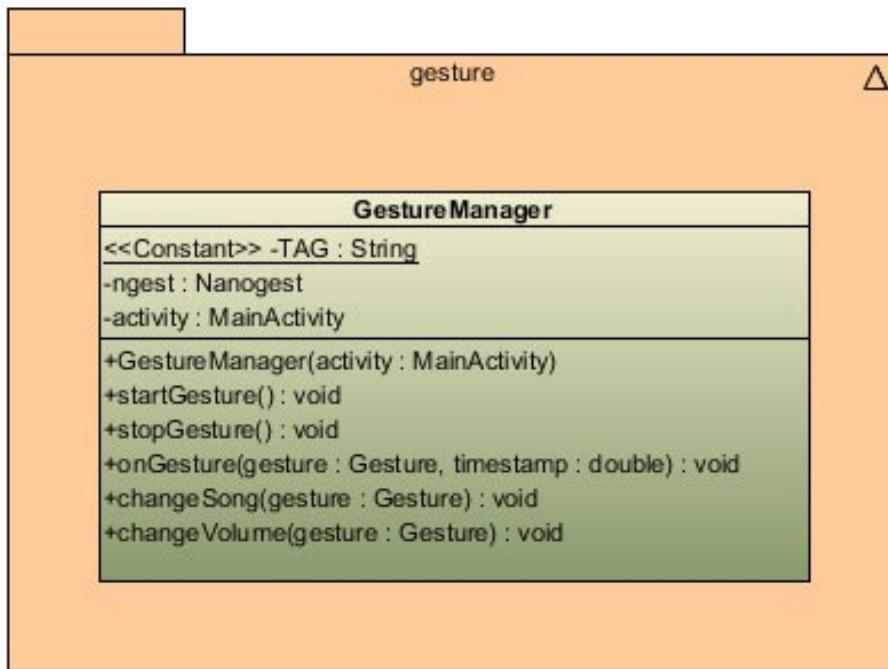


5.4.1 Package activity

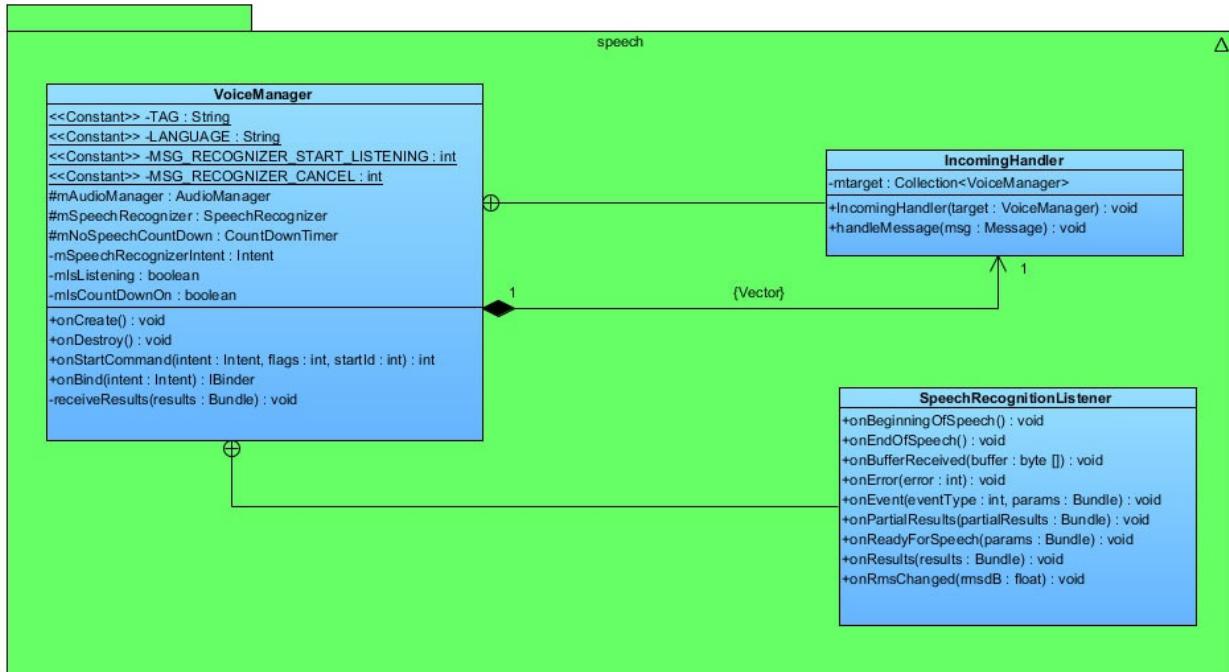




5.4.2 Package gesture



5.4.3 Package speech





5.5 Diagrammi di sequenza

Prendendo in esame alcuni dei casi d'uso determinanti per l'architettura del sistema, lo scopo di questa sezione è di mostrare la realizzazione dei casi d'uso nelle diverse modalità di interazione considerate. L'utilizzo di multiobject serve alla rappresentazione di generiche collezioni di oggetti.

I diagrammi di sequenza sono identificati da un codice del tipo:

SEQ_<numero diagramma di sequenza>

Le interazioni tra l'utente e il sistema verranno descritte mediante stereotipi così definiti:

<<enunciazione>>

Le enunciazioni sono delle parole pronunciate dall'utente che vengono riconosciute dal sistema e utilizzate per le esecuzioni delle principali operazioni.

<<gesto>>

Attraverso i gesti un utente può eseguire le azioni che vengono catturate dalla fotocamera e inviate all'applicazione.

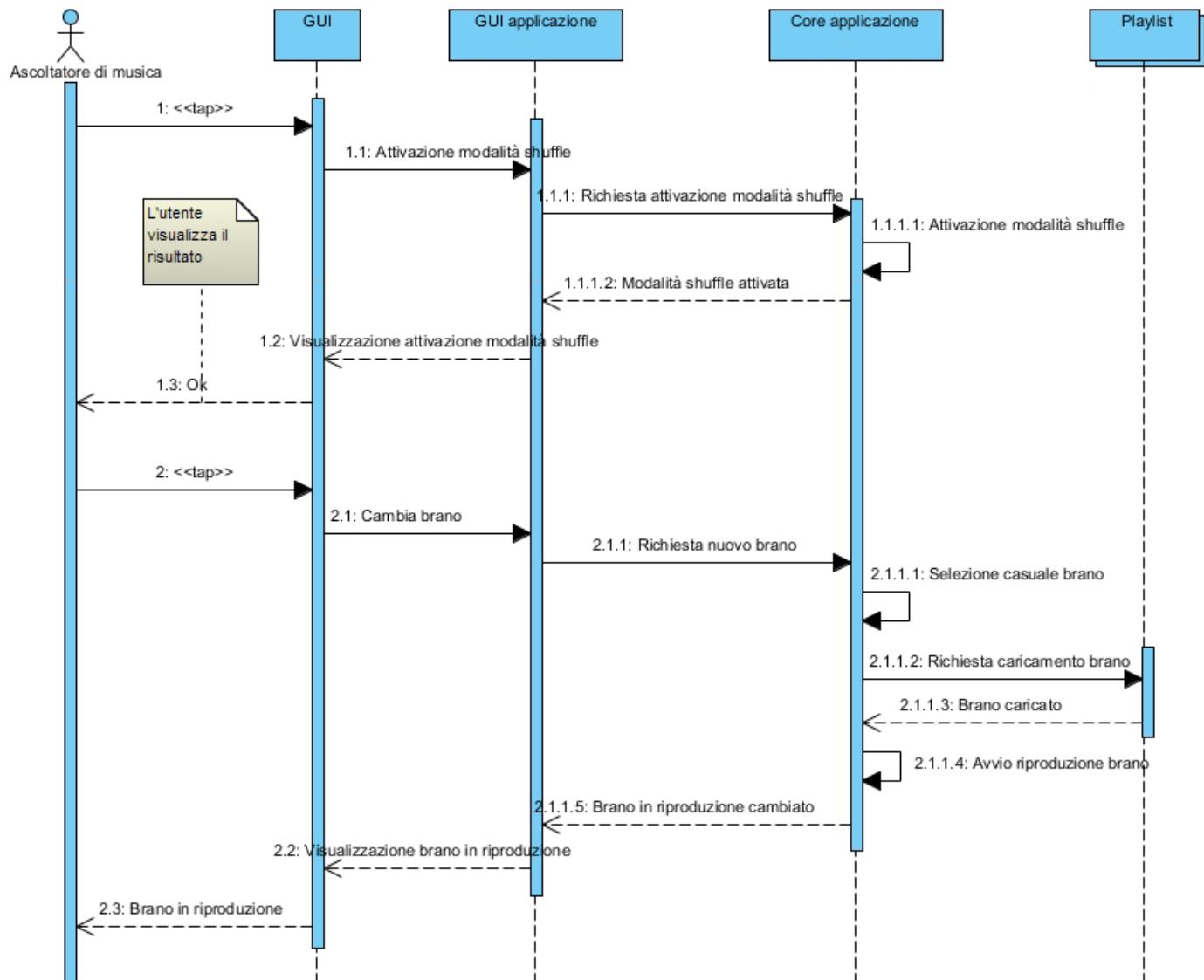
<<tap>>

Con tap si indica una singola pressione del touchscreen da parte dell'utente



5.5.1 SEQ_01

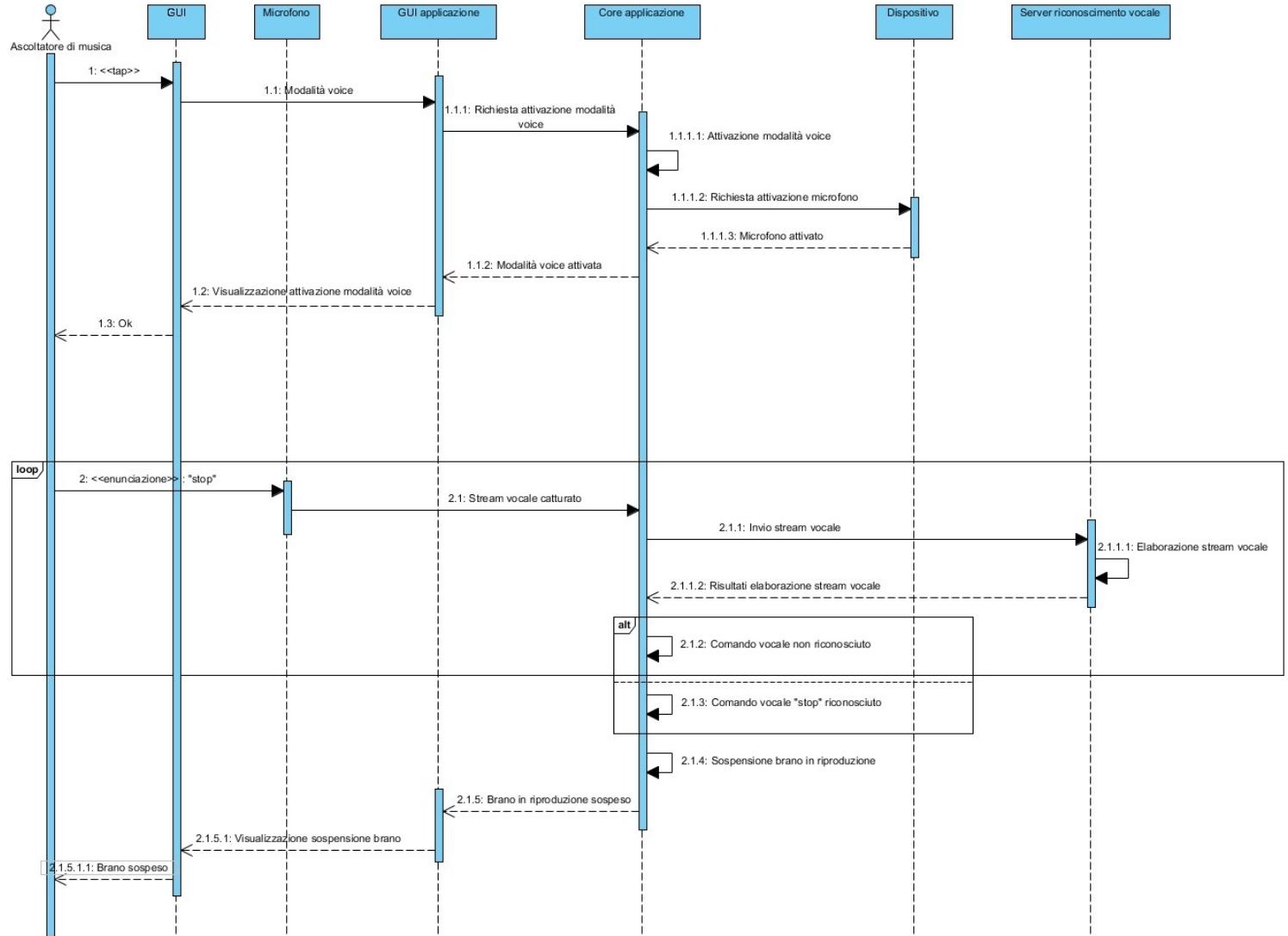
Il diagramma riproduce le azioni di un utente che attiva la modalità shuffle e riproduce un brano casuale





5.5.2 SEQ_02

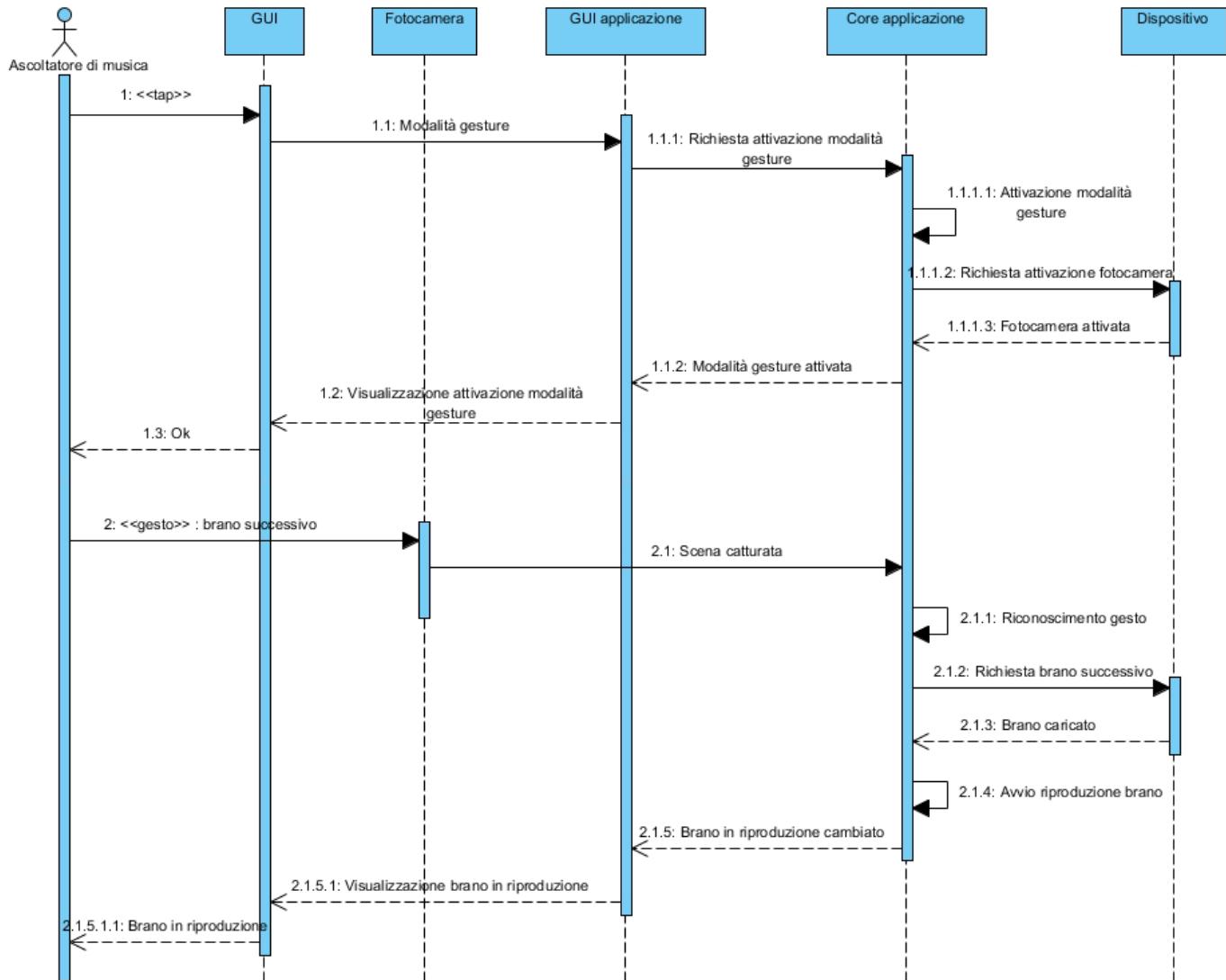
Il diagramma mostra l'attivazione della modalità vocale e la sospensione di un brano in riproduzione.





5.5.3 SEQ_03

Viene qui descritta l'attivazione della modalità di riconoscimento dei gesti e la selezione del brano successivo.



DOCUMENTO 6

PROGETTAZIONE, IMPLEMENTAZIONE E VALUTAZIONE INTERFACCIA

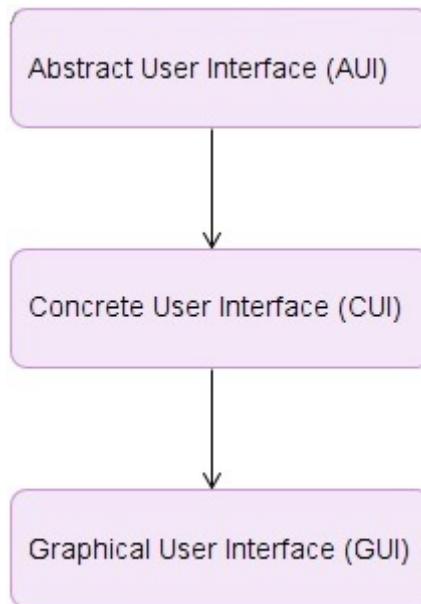
6.1 Introduzione

In questo capitolo viene mostrato come è stata realizzata l'interfaccia del sistema, a partire dalla sua progettazione astratta, in cui sono stati utilizzati contenitori e componenti astratti, passando poi per la progettazione concreta, in cui tali contenitori e componenti si concretizzano in oggetti come layout, textview, button, ecc, fino ad arrivare a mostrare alcune funzionalità tramite screenshot della GUI. Tali descrizioni non devono essere viste come una spiegazione dettagliata di come utilizzare il sistema ma, come un esempio di uso del sistema stesso. La valutazione di questi esempi, tramite le euristiche di Nielsen, ci dirà se l'interfaccia è stata ben strutturata o no.



6.2 Realizzazione interfaccia sistema

Per la realizzazione dell'interfaccia è stato utilizzato il modello UsiXML[3,4,5]. Tale modello è scomponibile in una serie di modelli tra cui lo auiModel (Abstract User Interface Model) e il cuiModel (Concrete User Interface Model) che sono stati utilizzati rispettivamente nella fase di progettazione astratta e concreta. Nel seguente diagramma, in cui i blocchi rappresentano le fasi di lavoro e le frecce il flusso di lavoro, viene riassunto in che modo si è arrivati creazione dell'interfaccia del sistema.



Come mostrato dal diagramma, quindi, nella prima fase si esegue una prima descrizione dell'interfaccia utilizzando il modello AUI (Abstract User Interface). Tale modello permette di descrivere un'interfaccia in termini di aio (Abstract Interaction Object) e di auiRelationship (Abstract User Interface Relationship). Gli aio sono gli elementi che popolano il diagramma e sono delle astrazioni di oggetti come layout, menu bar, button, textview, edittext, etc., tali oggetti si dividono in abstractContainer e in abstractIndividualComponent. Gli abstractContainer vengono definiti come degli spazi di interazione che permettono di raggruppare attività logicamente correlate facenti parte della stessa funzionalità; tali "spazi", che saranno in seguito realizzati concretamente come layout, menu bar, etc., possono contenere uno o più abstractContainer o uno o più abstractIndividualComponent. Gli abstractIndividualComponent rappresentano, invece, i singoli elementi che popolano l'interfaccia e possono diventare button, textview, etc.. Le auiRelationship, invece, descrivono le relazioni che vi sono tra le varie componenti astratte che popolano l'interfaccia e ne esistono di vario genere. Un'interfaccia descritta con tale modello si dice quindi astratta, in quanto non viene ancora specificato né il tipo effettivo di componenti utilizzati per la realizzazione degli aio, né la loro disposizione all'interno dell'interfaccia.

Nella seconda fase si è passati alla traduzione dell'interfaccia astratta in concreta tramite il modello CUI (Concrete User Interface). Tale modello consente la concretizzazione dell'interfaccia astratta traducendo gli aio in cio (Concrete Interaction Object) e le auiRelationship in cuiRelationship (Concrete User Interface Relationship). I cio sono gli



elementi che compongono l'interfaccia concreta, e sono definiti come delle entità della UI percepibili dall'utente: essi si dividono in una moltitudine di sottotipi che permettono di descrivere vari tipi di interfacce a partire da quelle grafiche in 2D fino ad arrivare a quelle vocali. Tali elementi possono essere contenitori come layout, menu bar, etc., o singoli elementi come button, combo box, text view, etc.. Le cuiRelationship invece, permettono di descrivere le relazioni tra due o più cio, e ne esistono di vari tipi. Ricapitolando, in questa fase di lavoro non si è fatto altro che tradurre i vari componenti e contenitori astratti in componenti e contenitori concreti, e tradurre le relazioni presenti tra di loro in relazioni concrete. Nella terza e ultima fase di lavoro, si è passato dall'interfaccia concreta all'implementazione della GUI: questo passaggio è quasi 1:1, infatti bisogna solamente tradurre il linguaggio del modello CUI nel linguaggio di programmazione scelto.



6.2.1 Progettazione interfaccia astratta

Di seguito alcune osservazioni per facilitare la comprensione del diagramma AUI:

- L'interfaccia astratta è stata descritta in più schemi per una questione di leggibilità.
- Per ogni aio presente nel sistema, è stato specificato se si tratta di abstractContainer o abstractIndividualComponent, inoltre è stata specificato l'attributo "Name", nome dell'aio all'interno dell'interfaccia.
- Negli schemi mostrati sarà presente la relazione di abstractContainment, che viene usata per indicare che un contenitore contiene uno o più abstractContainer e o uno o più abstractIndividualComponent; il tipo specifico di contenimento sarà poi mostrato nella fase di progettazione concreta.

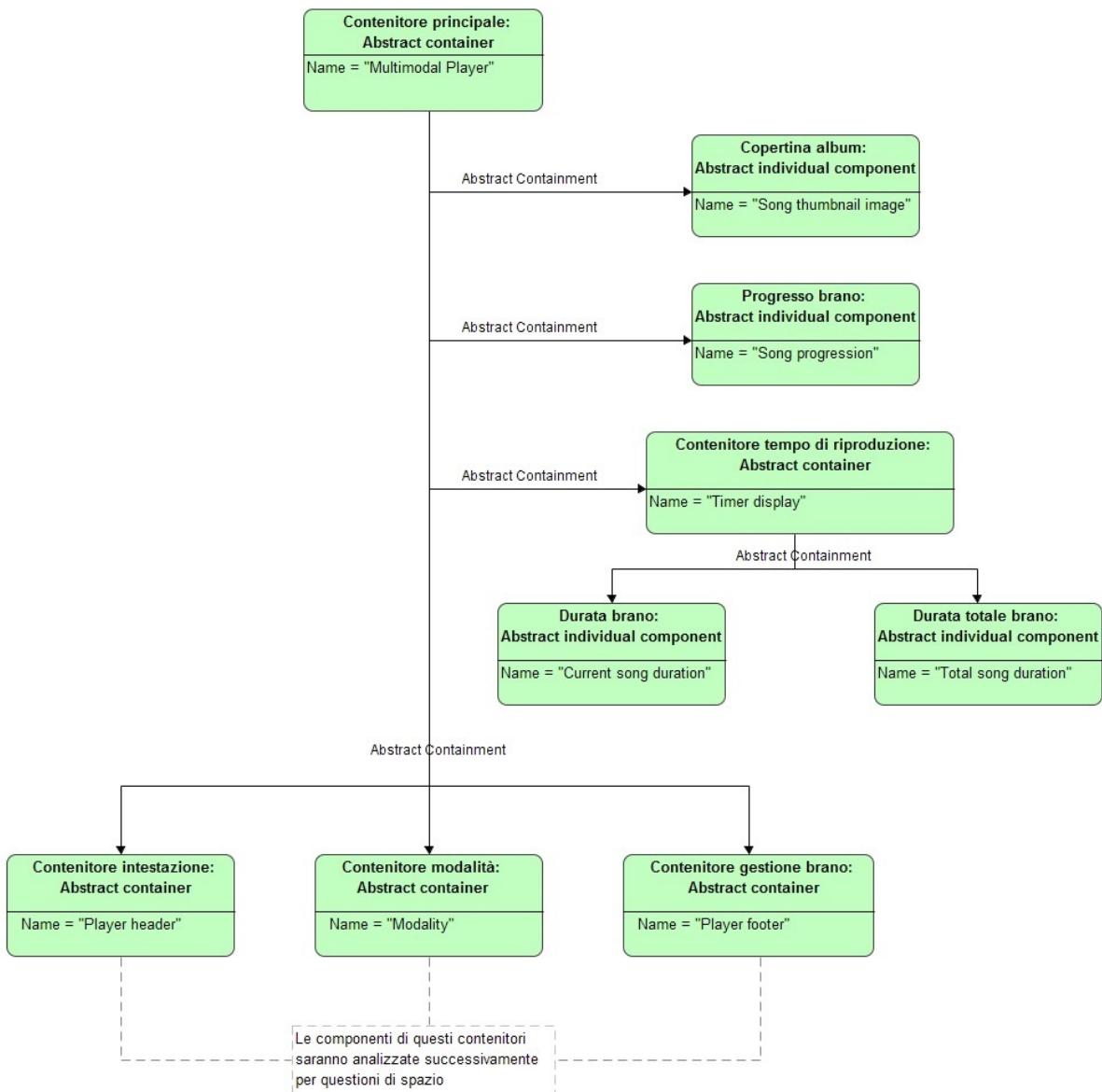


Figura 6.1: Interfaccia astratta - contenitore principale

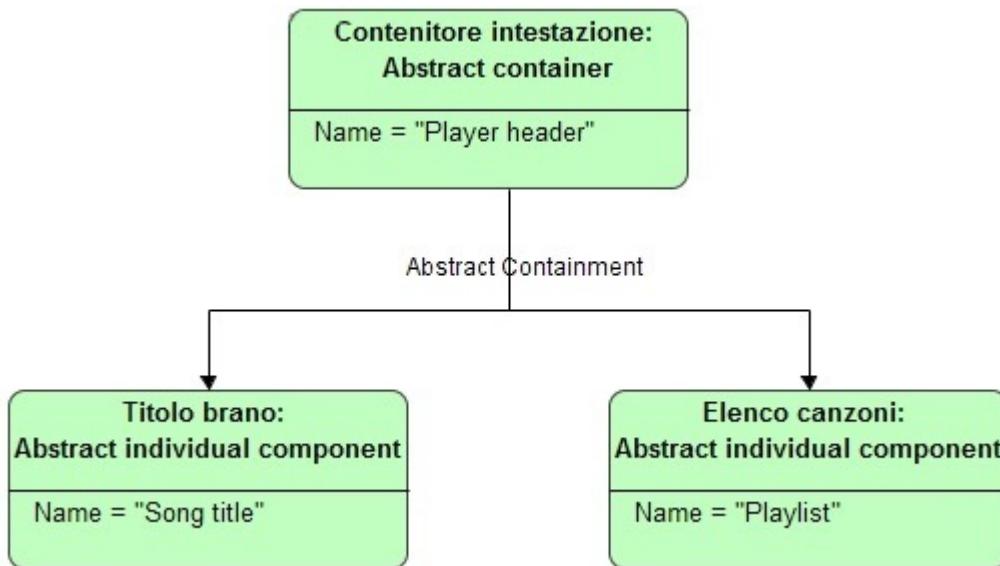


Figura 6.2: Interfaccia astratta - contenitore intestazione

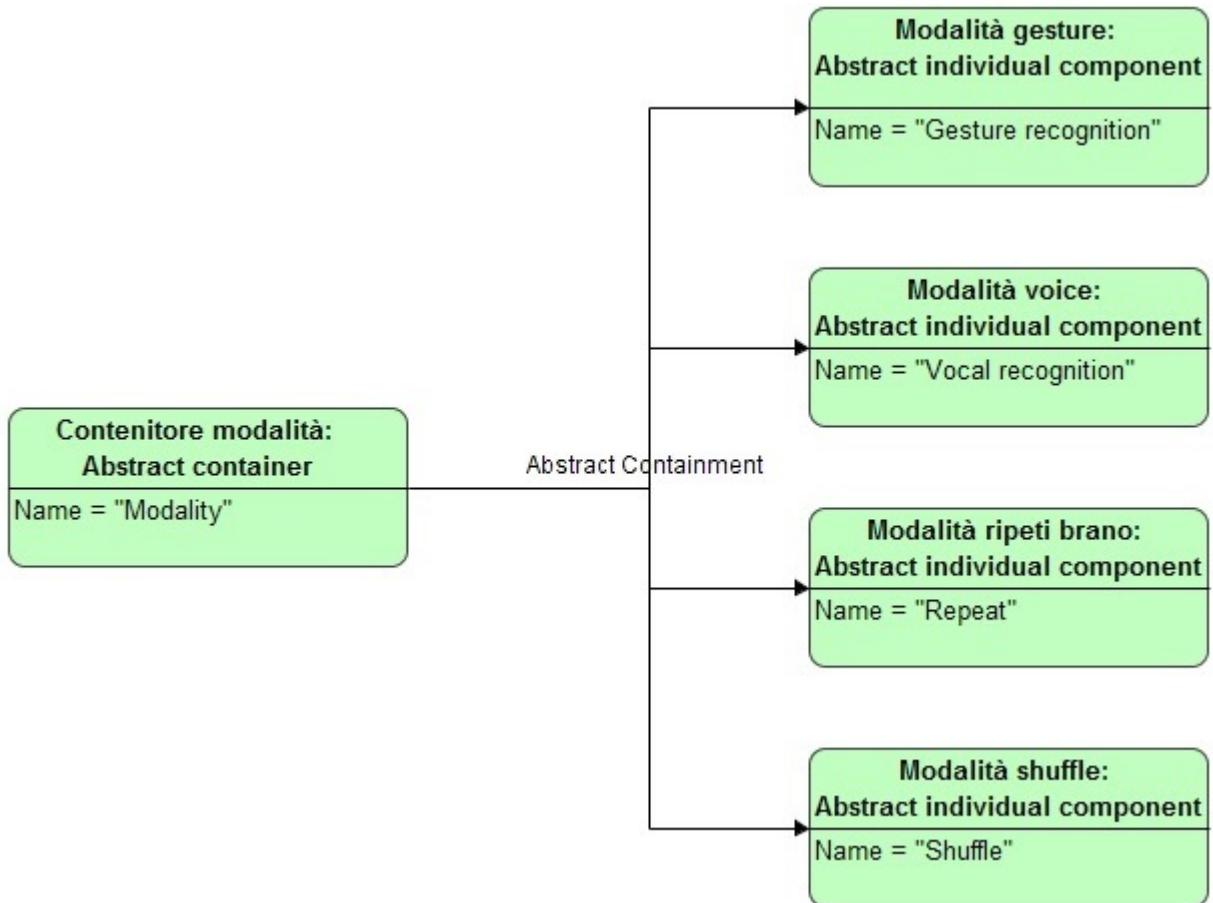


Figura 6.3: Interfaccia astratta - contenitore modalità

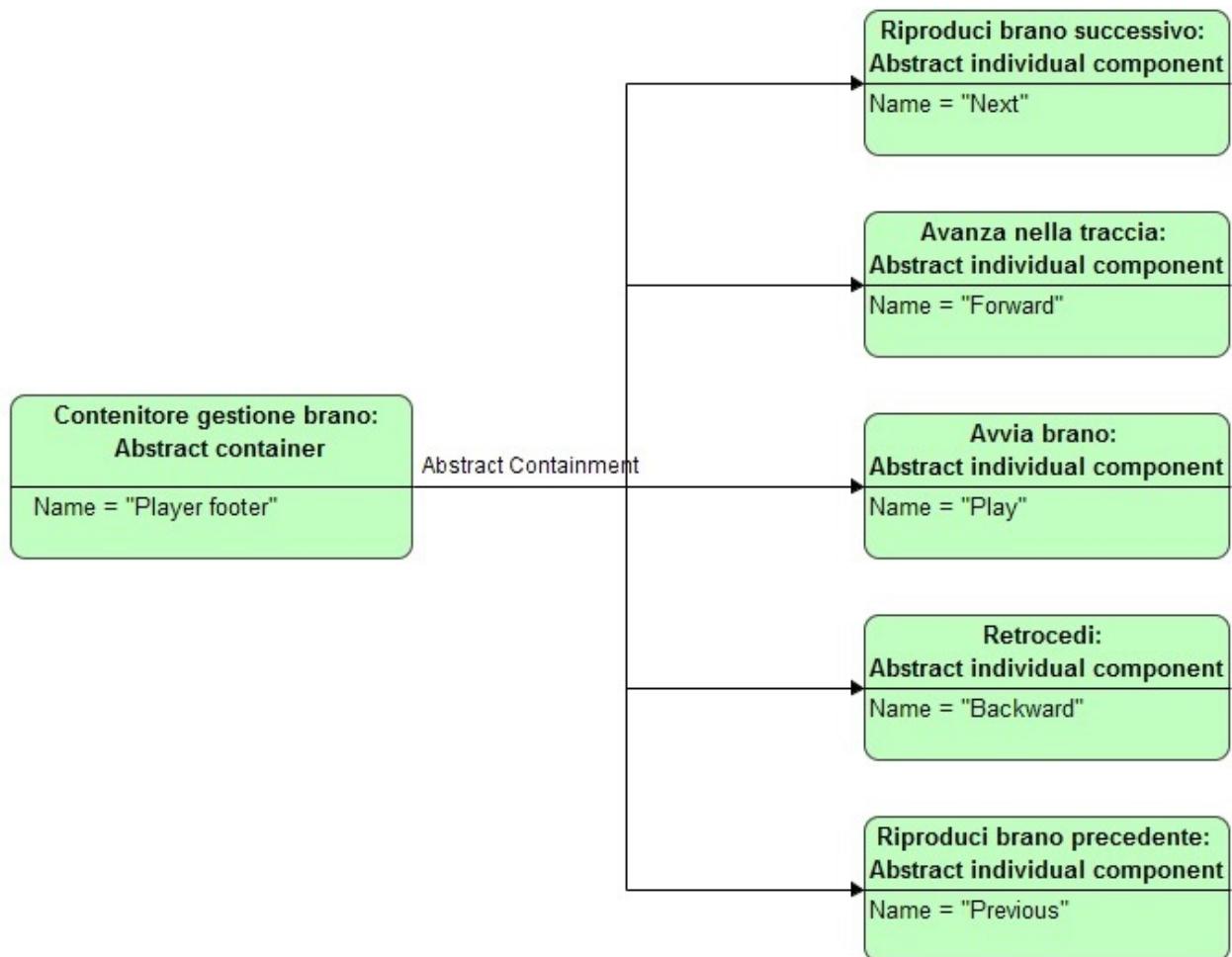


Figura 6.4: Interfaccia astratta: contenitore gestione brano



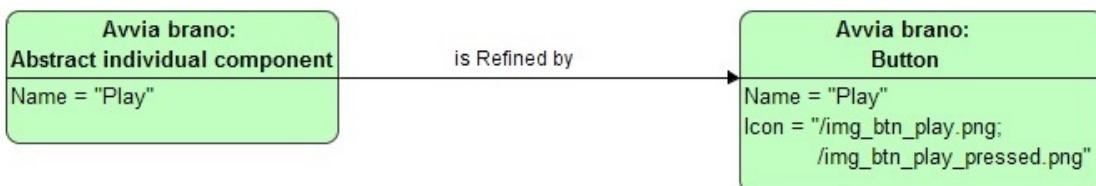
6.2.2 Progettazione interfaccia concreta

Come già detto in precedenza, in questa fase della progettazione si è passato alla concretizzazione dell'interfaccia astratta, ossia sono stati tradotti i vari elementi aio e le relazioni che vi sono tra di essi, in elementi e relazioni ciò descritte dal modello CUI. Tale modello offre una vasta gamma di famiglie di ciò, ognuna adatta a descrivere un particolare tipo di interfaccia, quindi prima di passare alla traduzione si è scelto quale tipologia di ciò utilizzare.

La nostra scelta è ricaduta sui 2DGraphicalCio. Un 2DGraphicalCio è un elemento che compone un'interfaccia grafica, e può essere un contenitore (2DGraphicalContainer) o una componente individuale (2DGraphicallIndividualComponent). Sia i contenitori, sia le componenti individuali si differenziano a loro volta in una serie di elementi concreti, ad esempio layout, menu bar, etc. per i contenitori, e button, textview, etc. per le componenti. Una volta fatta tale scelta non è rimasto altro da fare che concretizzare i vari elementi dell'interfaccia astratta e le relazioni presenti tra di essi, ossia sostituirli rispettivamente con elementi 2DGraphicalCio e con relazioni cuiRelationship. Negli schemi successivi verrà mostrato come i vari elementi aio sono stati tradotti.

Alcune precisazione sugli schemi:

- L'interfaccia concreta è stata descritta in più schemi per una questione di leggibilità.
- Nei diagrammi prodotti con il modello CUI, andrebbero rappresentati l'elemento aio e messo in relazione con l'elemento cui con il quale è stato concretizzato, in questo modo:



Tuttavia, sia per una questione di spazio, sia per una questione di leggibilità, le relazioni "isRefinedBy" saranno omesse. Confrontando gli schemi dell'interfaccia astratta con quelli dell'interfaccia concreta sarà possibile capire come i vari elementi astratti sono stati concretizzati.



- Per ogni componente e contenitore è stato specificato sia il tipo, sia una serie di attributi, tra cui "Name", nome dell'elemento, "Icon" eventuale icona presente sull'elemento.
- Negli schemi mostrati sarà presente la relazione di graphicalContainment, che viene usata per indicare che un contenitore contiene graficamente uno o più 2DGraphicalContainer e o uno o più 2DGraphicallndividualComponent.
- Di seguito viene fornita una legenda delle icone presenti negli schemi dell'interfaccia:

NOME	ICONA
img_btn_play.png	▶
img_btn_play_pressed.png	▶
img_btn_pause.png	⏸
img_btn_pause_pressed.png	⏸
img_btn_next.png	▶
img_btn_previous.png	◀
img_btn_forward.png	▶
img_btn_backward.png	◀
img_btn_playlist.png	⠇
img_btn_shuffle.png	🔀
img_btn_shuffle_pressed.png	🔀
img_btn_repeat.png	🔁
img_btn_repeat_pressed.png	🔁
img_btn_voice.png	🎙
img_btn_voice_pressed.png	🎙
img_btn_gesture.png	✋
img_btn_gesture_pressed.png	✋

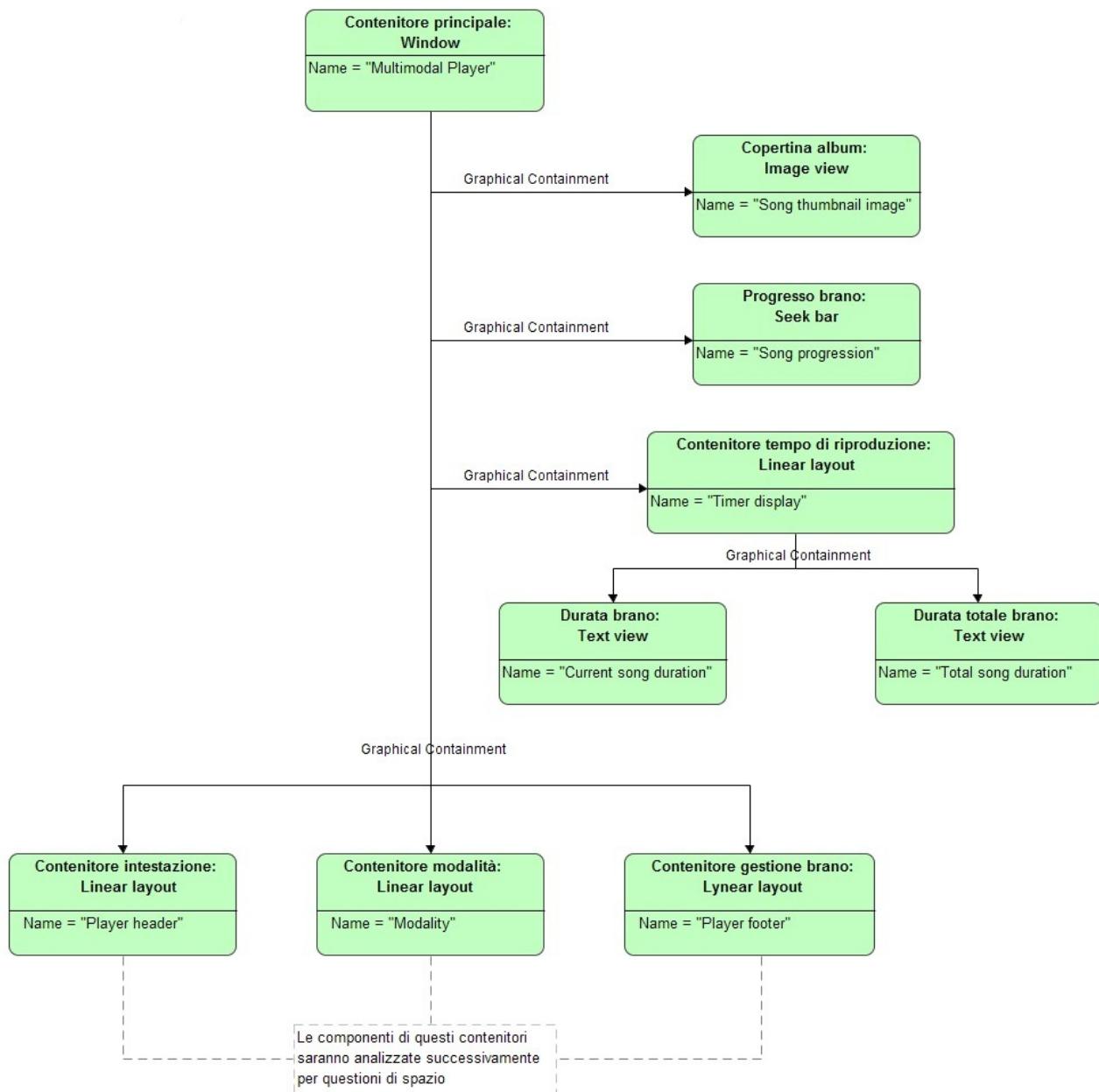


Figura 6.5: Interfaccia concreta - contenitore principale

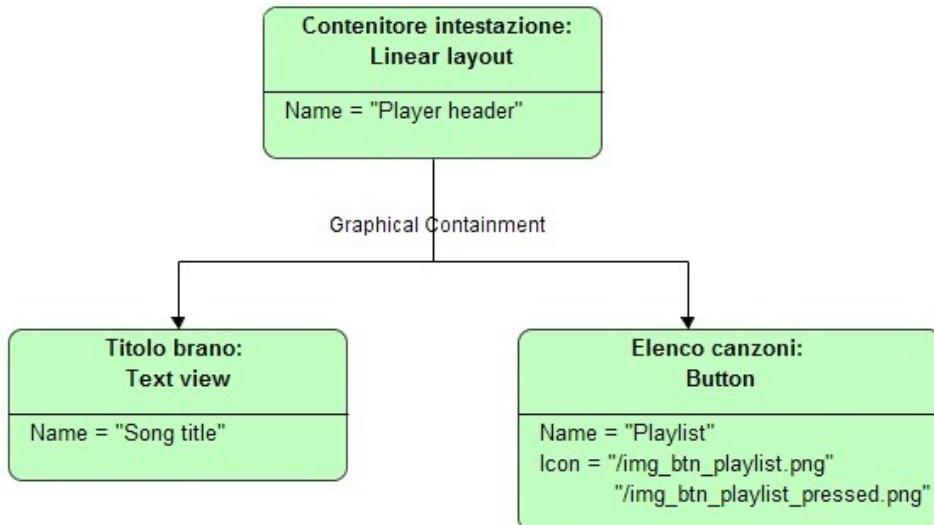


Figura 6.6: Interfaccia concreta - contenitore intestazione

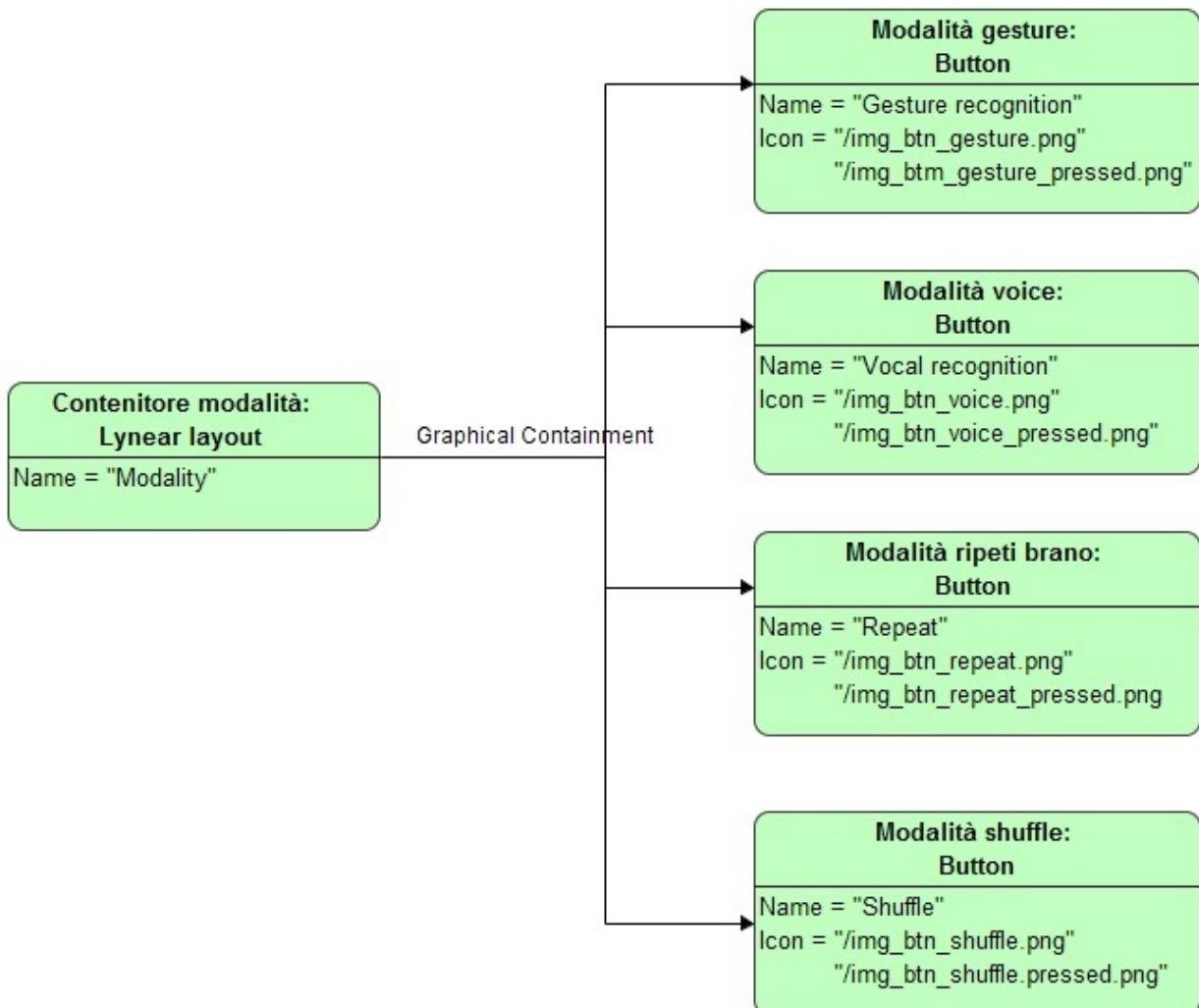


Figura 6.7: Interfaccia concreta - contenitore modalità

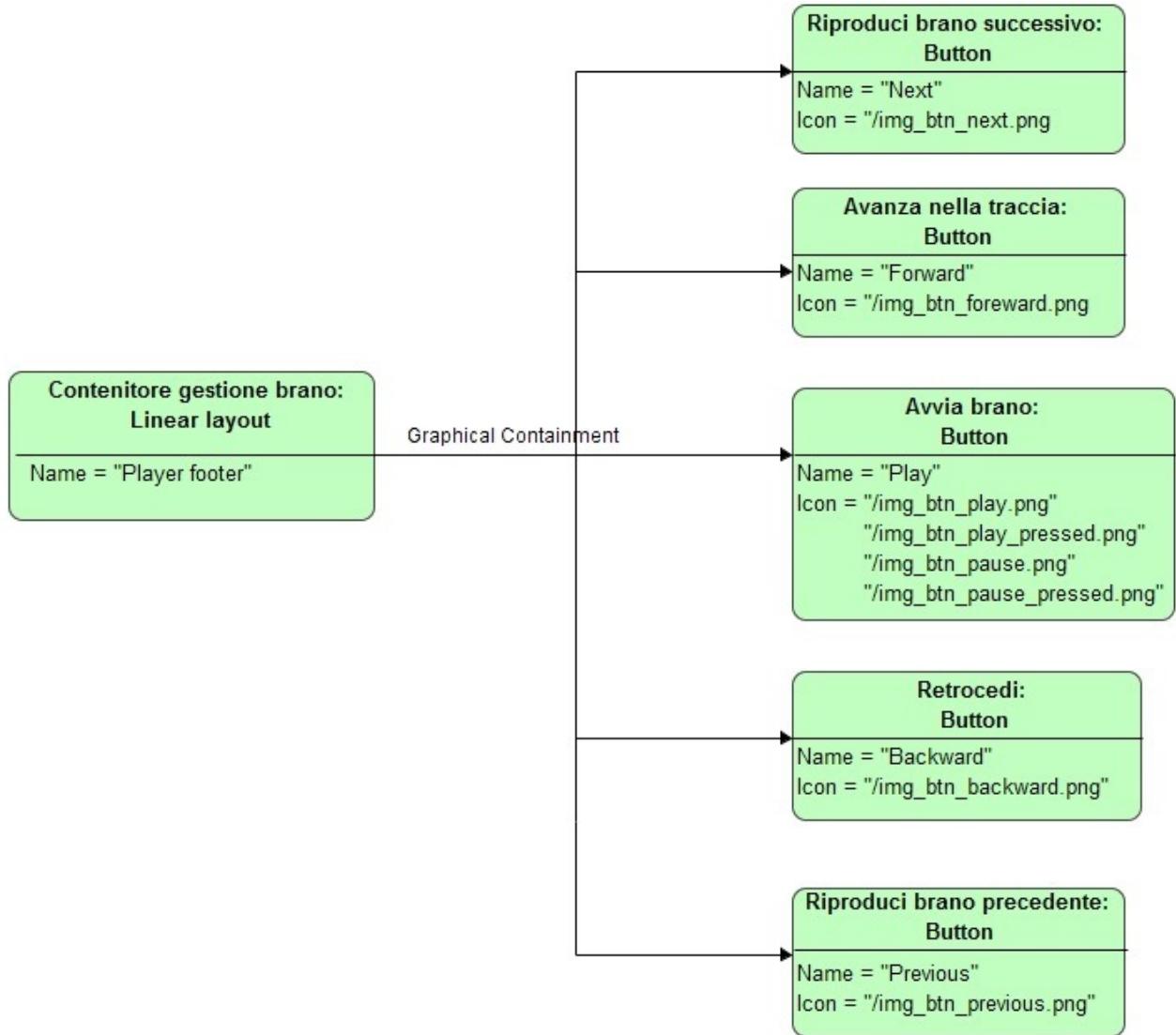


Figura 6.8: Interfaccia concreta – contenitore gestione brano



6.3 Implementazione GUI

Una volta creata l'interfaccia concreta utilizzando il modello CUI, non è rimasto altro da fare che produrre la GUI (Graphical User Interface); per fare ciò, come già detto in precedenza, basta tradurre gli elementi concreti, con il linguaggio di programmazione scelto. Di seguito vengono mostrati alcuni esempi, corredati da screenshot, dell'interfaccia prodotta, mentre nel paragrafo successivo si valuterà l'interfaccia ottenuta tramite le euristiche di Nielsen.

Le due schermate principali dell'applicazione sono la schermata di riproduzione (Figura 6.9) e la schermata di visualizzazione della playlist (Figura 6.10).

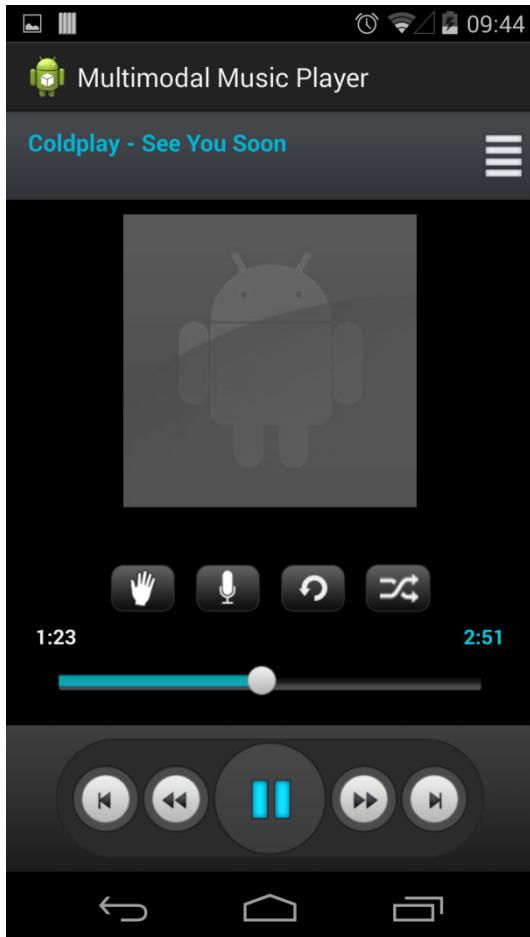


Figura 6.9: schermata di riproduzione

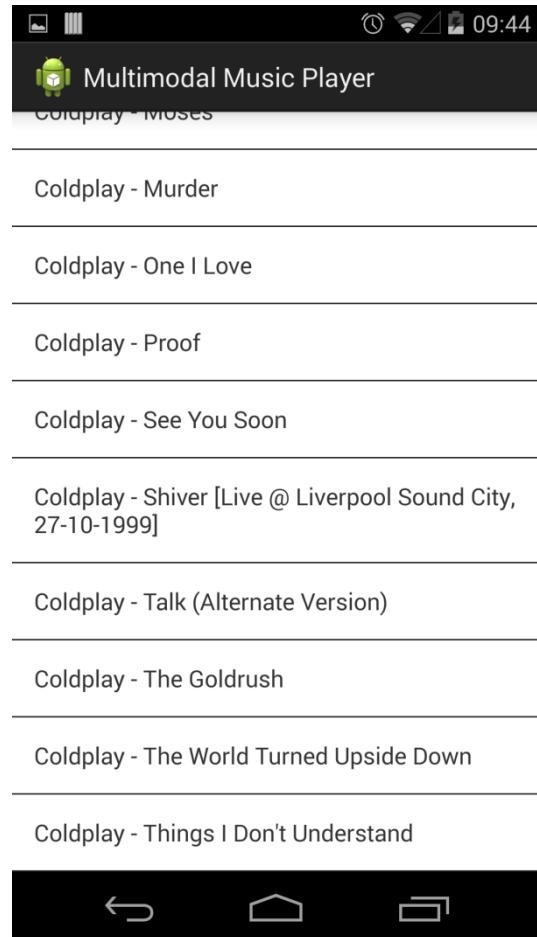


Figura 6.10: schermata della playlist



Di seguito, invece, vengono mostrati alcuni screenshot dell'interfaccia prodotta durante l'utilizzo dell'applicazione.

Nella Figura 11, ad esempio, viene mostrato come si può accedere alla schermata di visualizzazione della playlist: l'utente, dalla schermata di riproduzione, preme il pulsante "playlist" (evidenziato in Figura 11), e il sistema provvede a mostrare la schermata contenente la lista dei brani disponibili per la riproduzione presenti nel dispositivo.

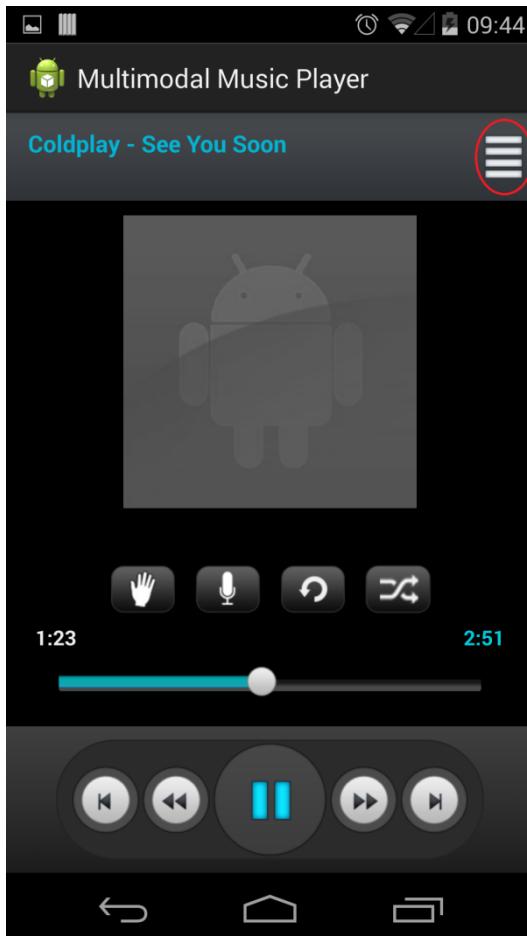


Figura 6.11: accesso alla schermata di visualizzazione della playlist



Nelle Figure 12 e 13, invece, è illustrato come l'utente può sospendere l'esecuzione del brano corrente usando l'interfaccia touch: l'utente tocca l'icona "pausa" (Figura 12) e il sistema, dopo aver sostituito quest'ultima con l'icona "play" (Figura 13), provvede a sospendere la riproduzione del brano corrente.

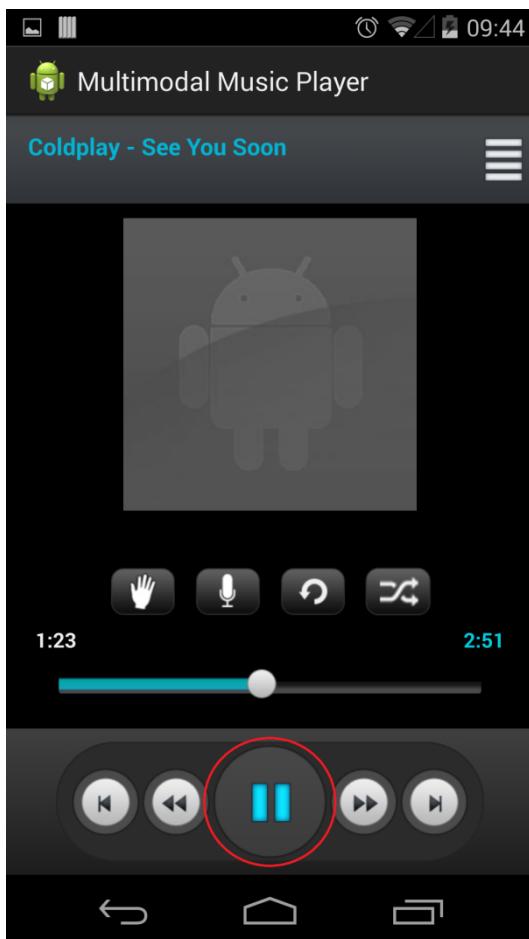


Figura 6.12: pulsante pausa

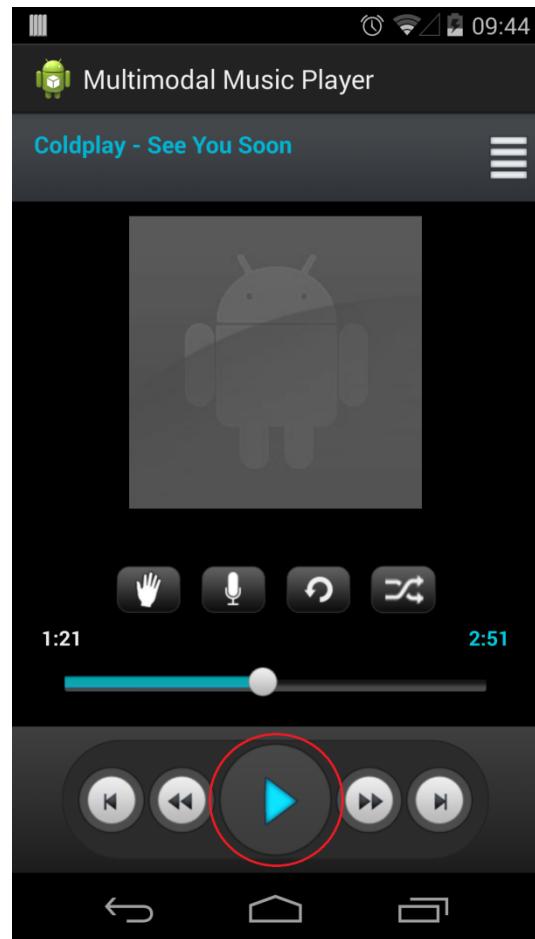


Figura 6.13: pulsante play



Le Figure 6.14 e 6.15 mostrano come avviene la riproduzione del brano successivo, tramite l'interfaccia touch: l'utente preme il pulsante “prossimo brano”, il sistema inizia la riproduzione del brano successivo, e le informazioni disponibili nella schermata di riproduzione riguardanti artista e titolo del brano vengono cambiate, in accordo con quelle del nuovo brano che il lettore musicale sta ora eseguendo.

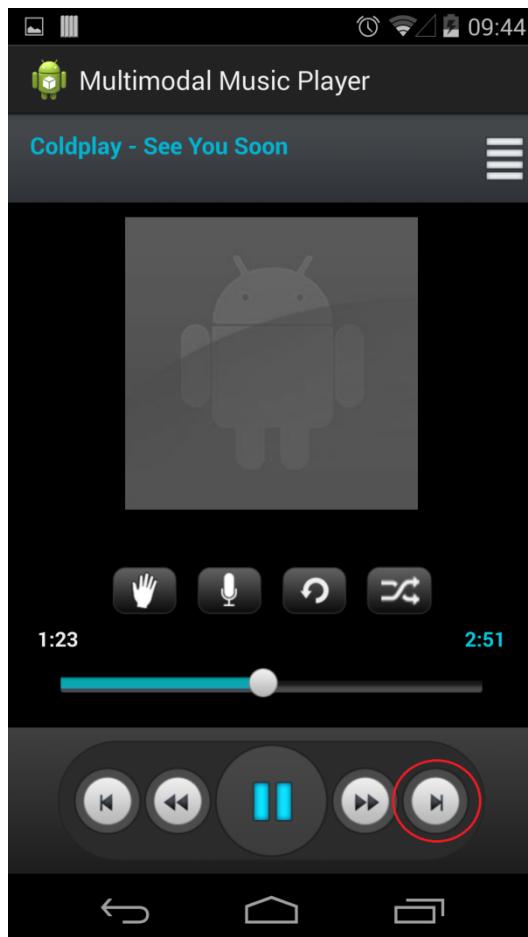


Figura 6.14: Riproduzione del brano successivo (prima)

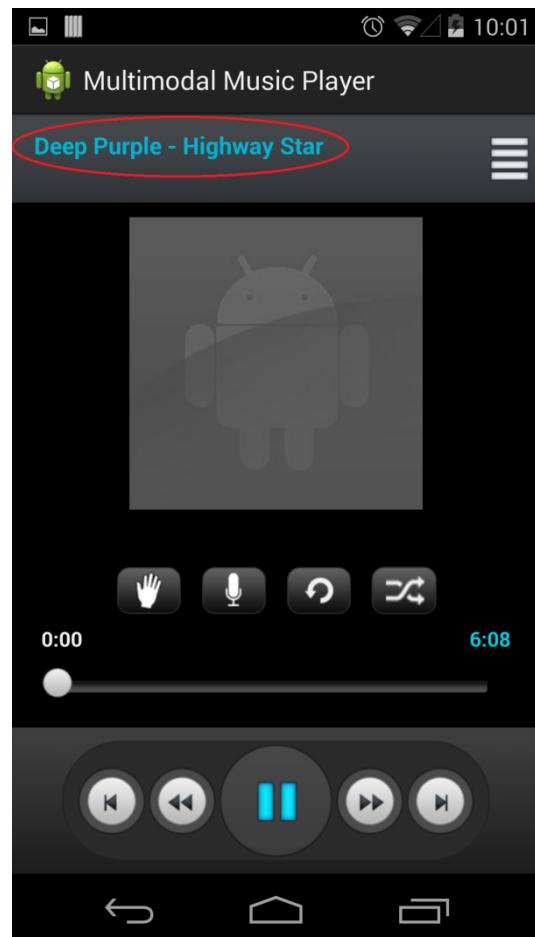


Figura 6.15: Riproduzione del brano successivo (dopo)



Una delle funzionalità del lettore musicale è la “riproduzione causale”. Per attivarla, l’utente tocca l’icona “riproduzione casuale”, la quale contestualmente viene sostituita con l’icona “riproduzione casuale attiva” per segnalare all’utente che la modalità richiesta è stata attivata; allo stesso scopo, un messaggio di avviso è brevemente mostrato sullo schermo del dispositivo (Figura 6.16).

La disattivazione della modalità “riproduzione casuale”, mostrata nella Figura 6.17, avviene in modo analogo: quando tale modalità è attiva, l’utente, premendo il pulsante “riproduzione casuale attiva”, ne richiede la disattivazione; il sistema provvede quindi ad eseguire l’operazione, sostituire la relativa icona con quella di “riproduzione casuale”, e infine mostra un messaggio sullo schermo per avvisare l’utente.

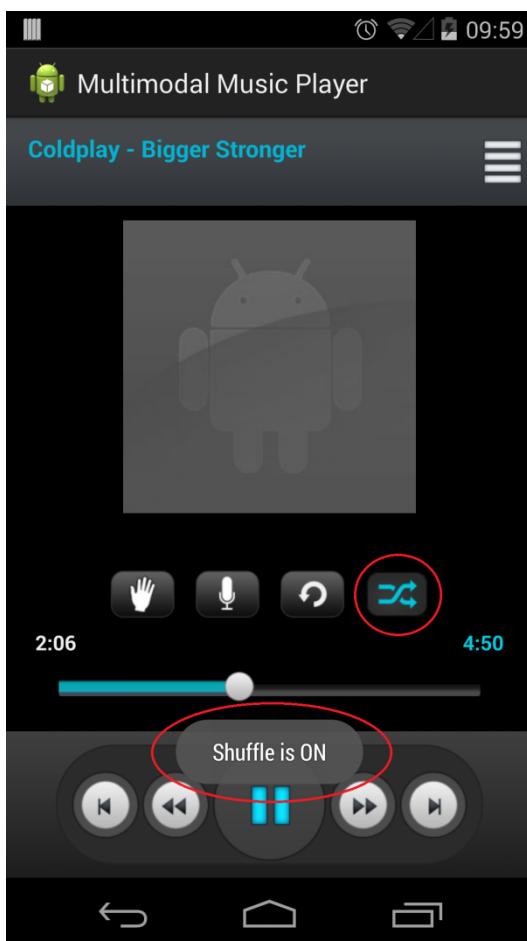


Figura 6.16: attivazione funzionalità
“riproduzione casuale”

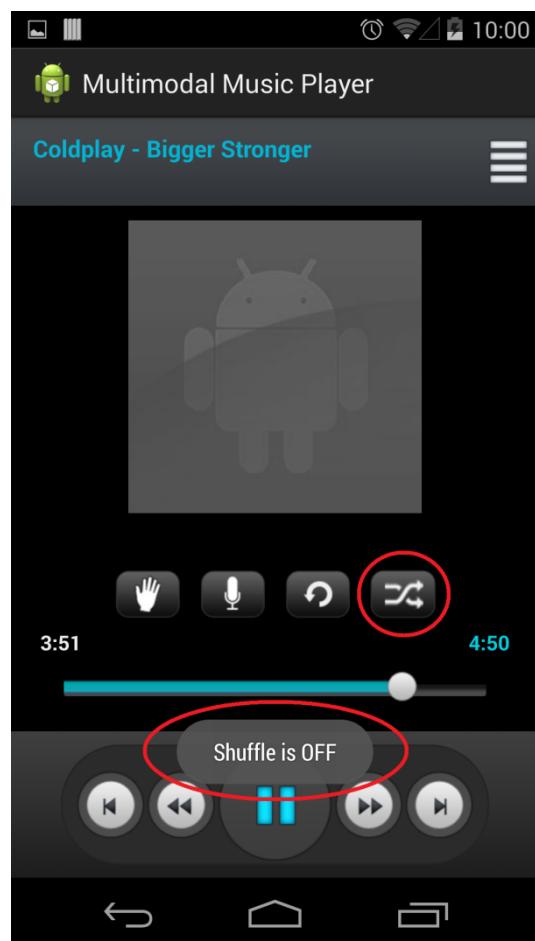


Figura 6.17: disattivazione funzionalità
“riproduzione casuale”



Ci concentriamo ora sull'aspetto multimodale, descrivendo come è stata realizzato il rapporto tra l'utente e le funzionalità multimodali dell'applicazione a livello di interfaccia grafica. Per quanto riguarda il riconoscimento dei gesti, nelle Figure 6.18 e 6.19 possiamo osservare come l'utente può rispettivamente attivare e disattivare tale funzionalità: dopo aver premuto l'icona "riconoscimento gesti", il sistema sostituisce quest'ultima con la nuova icona "riconoscimento gesti attivo", che avvisa l'utente dell'avvenuta attivazione e mostra, a questo proposito, anche un messaggio sullo schermo (Figura 6.18); per la disattivazione, l'utente non deve fare altro che premere il pulsante "riconoscimento gesti attivo", e il sistema provvederà alla sua sostituzione con l'icona "riconoscimento gesti" e a mostrare il messaggio informativo associato (Figura 6.19).

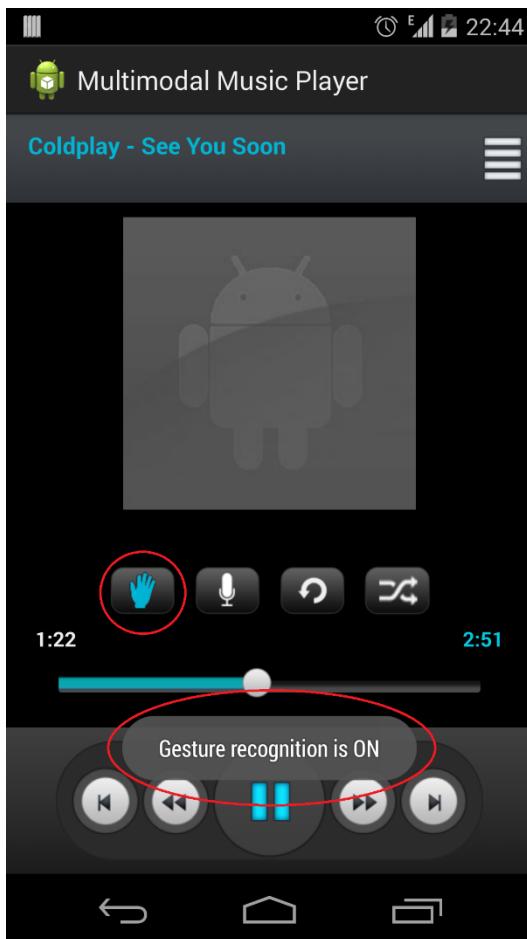


Figura 6.18: attivazione funzionalità "riconoscimento gesti"

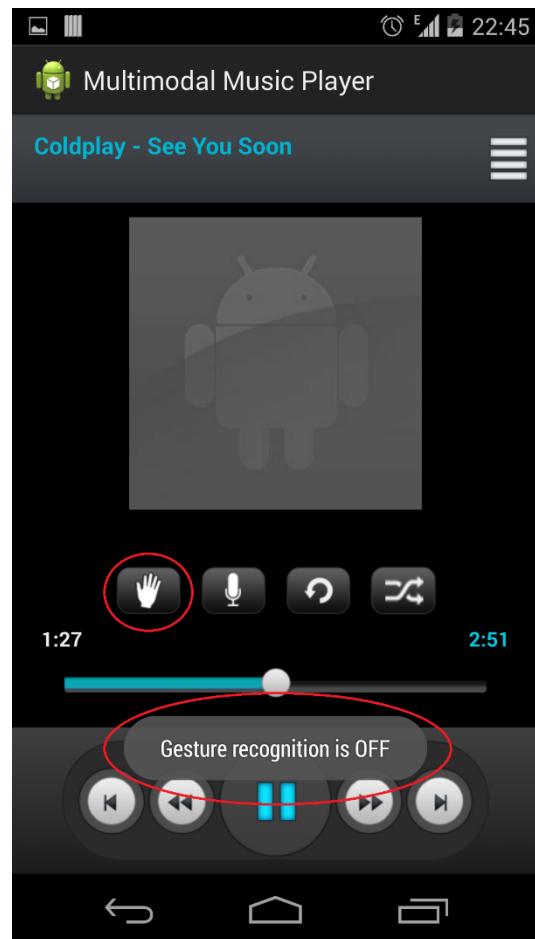


Figura 6.19: disattivazione funzionalità "riconoscimento gesti"



Infine, relativamente all'interazione tra l'utente e la funzionalità di riconoscimento dei comandi vocali, questa è descritta nelle Figure 6.20, 6.21 e 6.22. Quando l'utente preme il pulsante “riconoscimento comandi vocali”, il sistema verifica se si stiano utilizzando o meno delle cuffie: nel secondo, mostra prima di tutto un messaggio di warning sullo schermo per avvisare l'utente che la funzionalità potrebbe non funzionare correttamente (Figura 6.20, per ulteriori approfondimenti a questo proposito si rimanda al capitolo 7). Per il resto il meccanismo di questa funzionalità, come si può vedere nelle Figure 6.21 e 6.22, è del tutto analogo a quanto accennato poco sopra riguardo la funzionalità di riconoscimento dei gesti.

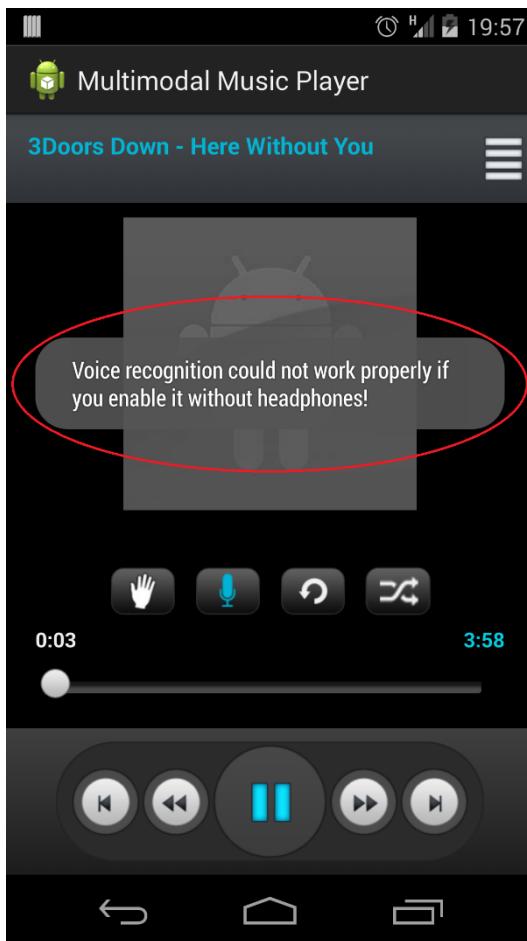


Figura 6.20: messaggio di warning per l'attivazione della funzionalità "riconoscimento comandi vocali"

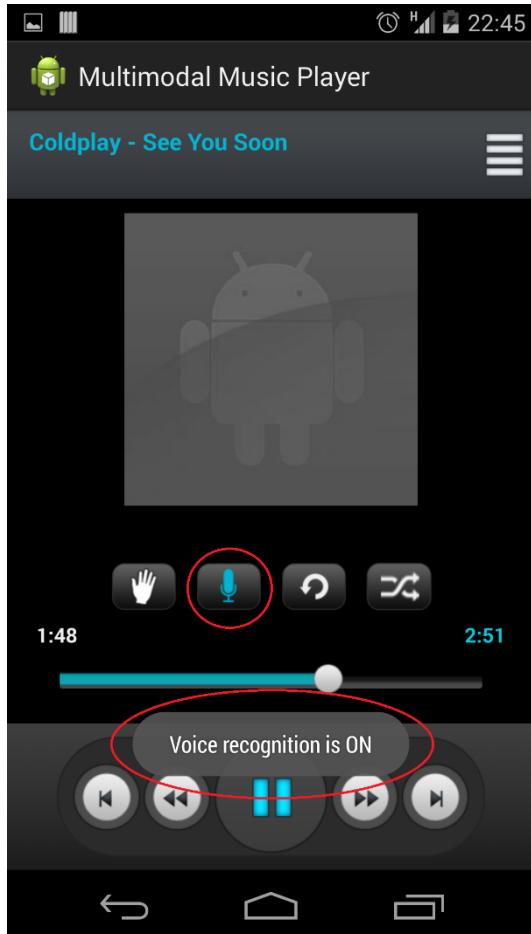


Figura 6.21: attivazione funzionalità
"riconoscimento comandi vocali"

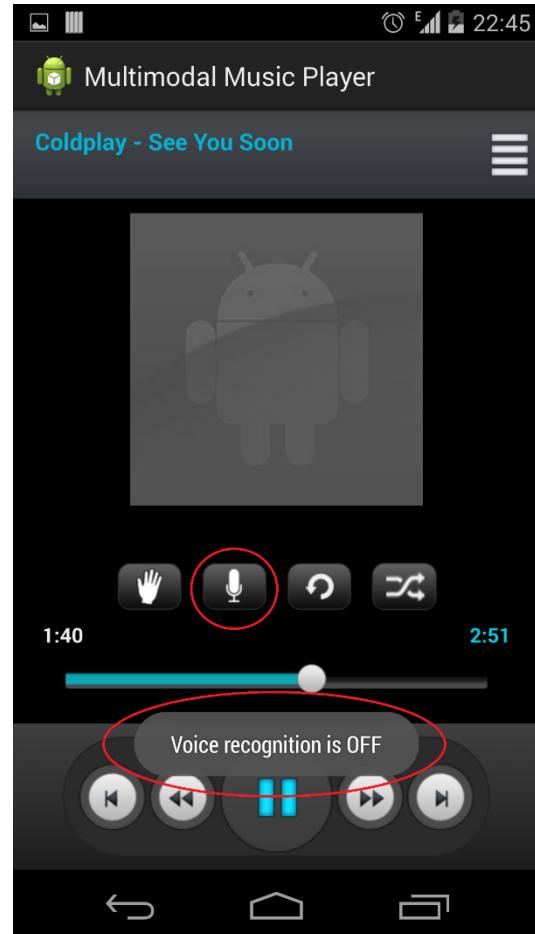


Figura 6.22: disattivazione funzionalità
"riconoscimento comandi vocali"



6.4 Valutazione GUI e euristiche di Nielsen

Visibilità dello stato del sistema

L'utente sa sempre in che stato si trova il sistema. Per quanto riguarda il brano in esecuzione, di cui è sempre visibile artista e titolo (Figura 6.9), l'utente sa in ogni momento quale è il punto di riproduzione del brano, sia grazie alla barra di riproduzione sia alle informazioni sul tempo di riproduzione presenti su di essa. Dopo aver cambiato brano, l'utente vedrà nella schermata di riproduzione artista e titolo del nuovo brano in esecuzione (figura 6.15) . Altri esempi sono costituiti dalle modalità "gesture", "voice", "repeat" e "shuffle", l'utente sa sempre se sono attive o non attive osservando le icone dei tasti che permettono di accedervi. L'icona in bianco indica che la modalità è al momento disattiva, l'icona in azzurro indica che la modalità è momentaneamente attiva; oltre a ciò viene mostrato un messaggio che comunica che la modalità è stata attivata o disattivata(figure 6.16, 6.17, 6.18, 6.19, 6.20, 6.21). Un ultimo esempio è costituito dal tasto "play", l'icona del tasto cambia a seconda se il brano sia in esecuzione o no (figura 6.12, 6.13), segnalando così lo stato di esecuzione del brano.

Corrispondenza tra mondo reale e sistema

I termini utilizzati come playlist, shuffle, repeat ecc. sono termini ben noti all'utente, e comuni a tutti i lettori musicali più diffusi.

Libertà e controllo da parte degli utenti

L'utente non ha nessun vincolo nell'uso dell'applicazione. Può decidere infatti tra tre modalità di interazione differenti. Per esempio il brano può essere cambiato cliccando sull'apposita icona, eseguendo un movimento della mano verso destra o pronunciando la parola "next"; alternativamente il brano può essere messo in pausa sia toccando l'apposita icona sia pronunciando il comando "stop". Per una spiegazione dettagliata delle modalità di interazione voice e gesture si rimanda al capitolo 7.

Consistenza e standard

Le icone utilizzate sono molto comuni e facilmente riconoscibili, ad esempio l'icona delle modalità di "repeat" e "shuffle" sono presenti in tutti i lettori musicali. Anche i tasti per la gestione del brano e per accedere alla playlist sono icone comuni e standard. Le sole icone create ad hoc sono quelle per la modalità "gesture" e "voice", ma comunque sono facilmente riconoscibili ed associabili alla loro funzionalità (tutte le icone sono visibili nella figura 6.9).

Prevenzione degli errori

Le possibilità dell'utente di commettere errori nell'uso dell'applicazione sono estremamente limitate. Al fine di evitare che involontariamente tramite gesti o tramite comandi vocali esegua operazioni non volute, l'utente deve necessariamente attivare le modalità di interazione alternative prima di poterle utilizzare. Per quanto riguarda la



modalità di interazione "voice" se attivata quando non sono inserite le cuffie nel dispositivo, il sistema mostra un apposito messaggio di warning per avvertire l'utente che la musica potrebbe interferire con l'interpretazione dei comandi vocali (figura 6.13).

Riconoscere piuttosto che ricordare

L'utilizzo del sistema non prevede passaggi difficili che l'utente deve memorizzare. L'utente può utilizzare l'applicazione in modo semplice e intuitivo. Inoltre sia le icone, molto comuni e facilmente riconoscibili, sia i messaggi che vengono visualizzati, permettono all'utente di riconoscere le varie funzionalità con estrema facilità.

Flessibilità ed efficienza d'uso

L'utente ha a disposizione ben tre modalità di interazione con il quale è libero di utilizzare il sistema.

Design minimalista

Le schermate (figure 6.9 e 6.10) hanno un design minimale, infatti contengono un numero minimo di elementi, e pulsanti. L'ordine e la posizione de i pulsanti nella schermata di riproduzione non è casuale, ma cerca di seguire i principi delle "Leggi della Gestalt" [6], in particolare la "legge della vicinanza o della prossimità", "gli elementi del campo visivo che sono fra loro vicini vengono percepiti come unità", infatti i pulsanti necessari a svolgere un determinato tipo di attività sono vicini tra di loro proprio per essere percepiti come unità; ad esempio i tasti "gesture", "voice", "repeat" e "shuffle" sono disposti uno vicino all'altro, proprio per essere percepiti come un'unica unità adibita alla attivazione e disattivazione delle varie modalità. Ulteriore esempio sono i tasti per la gestione del brano, che sono posti vicini proprio per essere percepiti come un unico insieme. Infine non sono presenti grafiche particolari che potrebbero confondere l'utente.

Aiutare gli utenti a riconoscere gli errori, diagnosticarli e correggerli

Nel caso l'utente inavvertitamente cambi il brano in esecuzione, può sempre tornare al brano precedente utilizzando gli appositi comandi "touch" o "gesture". Un altro esempio, è nel caso in cui l'utente cambi inavvertitamente il punto di riproduzione della canzone in esecuzione; tramite l'apposita barra o tramite i tasti per lo scorrimento veloce del brano, l'utente può sempre tornare al punto di riproduzione originale.

Guida e documentazione

Per le istruzioni relative al funzionamento dell'applicazione e ai comandi utilizzabili, sarà presente una documentazione nello store di Google.



DOCUMENTO 7

INTERAZIONE MULTIMODALE

7.1 Introduzione

In questo documento vengono spiegate dettagliatamente le modalità per l'interazione tra l'utente e l'applicazione descrivendone i principali motivi della loro utilità, i problemi riscontrati e gli strumenti utilizzati per la loro gestione.



7.2 Modalità di interazione

Oltre alla possibilità di utilizzare il lettore mp3 attraverso il touchscreen come una normale applicazione per smartphone, viene data all'utente la possibilità di attivare l'utilizzo di comandi vocali e di comandi gestuali. Questo approccio offre diversi vantaggi che rendendo l'applicazione più utilizzabile ed efficiente a seconda delle esigenze dell'utente.

7.2.1 Interazione vocale

Questo tipo di interazione rende possibile la comunicazione tra essere umano e macchina attraverso l'utilizzo esclusivo di comandi vocali. Alla base di questa modalità c'è il riconoscimento della voce ottenuto attraverso l'elaborazione di segnali vocali in sequenze di bit che ne rappresentano il contenuto. Nel nostro caso è sufficiente confrontare le sequenze così ottenute con un dizionario in modo da associarle a stringhe che possono essere comprese e riconosciute dalla macchina. I vantaggi principali che si possono ottenere sono innanzitutto un'interazione molto più semplice e immediata: infatti il discorso rappresenta il principale strumento di comunicazione tra esseri umani ed è sicuramente più intuitivo per un utente dare ordini vocali a un dispositivo piuttosto che eseguirli a mano. Inoltre esistono persone con diverse disabilità a cui sarebbe magari molto più difficile o addirittura impossibile eseguire azioni manuali. Basti pensare ad esempio a utenti non vedenti o privi di arti. Il principale problema riscontrato nell'interazione vocale è dovuto ai rumori ambientali che possono disturbare i segnali in input. Oltre alla necessità di inviare i comandi in ambienti poco rumorosi, nel nostro caso è stato riscontrato un ulteriore problema: se l'utente sta ascoltando un brano direttamente dalle casse esterne del suo dispositivo, la musica stessa può disturbare il riconoscimento della voce rendendo nello specifico la richiesta di pausa molto difficile. Per riuscire ad aggirare il problema si è deciso di mettere un messaggio di warning nell'applicazione che consiglia all'ascoltatore di utilizzare questa modalità esclusivamente con degli auricolari in modo da non disturbare il segnale in input.

7.2.2 Interazione gestuale

Come la voce, i gesti rappresentano un altro importante modello di comunicazione. Basti pensare che nella vita di tutti i giorni noi umani tendiamo a comunicare e a rafforzare ciò che vogliamo esprimere attraverso i gesti. Questo spiega come i vantaggi ottenuti dall'interazione gestuale siano notevoli: innanzitutto si possono eseguire comandi molto più intuitivi rispetto a quelli offerti da una normale applicazione. Si ha infatti la possibilità di eseguire le operazioni che si desiderano attraverso dei gesti comuni. Questo è possibile attraverso un semplice processo di riconoscimento: nel nostro caso la fotocamera del dispositivo rileva i diversi movimenti della mano e, attraverso un dizionario dei segni, elabora le immagini e riconosce il comando eseguito dall'utente associandolo alla funzione corrispondente (Figura 7.1). Il problema principale di questa tipologia di interazione è dovuto alla condizione attuale degli algoritmi di riconoscimento dei gesti basati sulla visione. Questi non sono del tutto precisi e hanno bisogno di condizioni di illuminazione ottimali e non soggette a disturbi. Nel nostro caso specifico il sistema di riconoscimento dei gesti è risultato abbastanza buono e preciso.

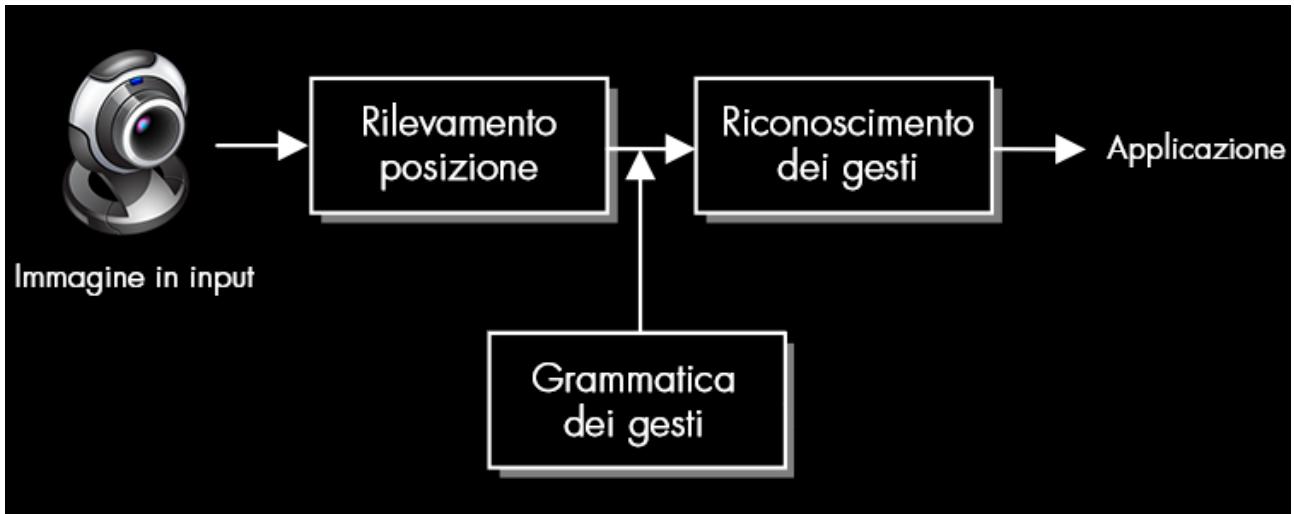


Figura 7.1: schema di rilevamento dei movimenti della mano

7.3 Tecnologie utilizzate

Vengono qui descritte nel dettaglio le diverse scelte implementative e le ragioni di tali decisioni.

7.3.1 Scelta del sistema operativo

Si è preferito rivolgere l'applicazione a un ambiente Android principalmente per facilitarne la pubblicazione e per renderla disponibile su un maggior numero di dispositivi. Questa scelta è stata possibile grazie alla presenza di librerie per l'implementazione sia dell'interazione vocale che gestuale.

7.3.2 Implementazione del riconoscimento vocale

Per le funzioni vocali e per il riconoscimento del parlato si è deciso di servirsi delle API di Android¹. Questa libreria è infatti molto semplice da utilizzare e il riconoscimento si è dimostrato preciso e efficiente: è sufficiente controllare che la voce elaborata corrisponda a delle stringhe costanti che rappresentano i comandi inviati. Si è optato per questa soluzione, nonostante essa richieda la presenza di una connessione a Internet costante per comunicare con i server Google per il riconoscimento vocale, dopo aver inizialmente provato anche ad utilizzare Sphinx, che tuttavia, oltre ad essere poco intuitiva, molto spesso non è riuscita a riconoscere la voce correttamente. Il servizio di riconoscimento vocale implementato nell'applicazione, quindi, funziona nel seguente modo: l'utente, come visto in precedenza, attiva tramite l'apposito pulsante il servizio, che parte in background e utilizza il microfono del dispositivo per restare in ascolto; una volta captati uno o più suoni, li salva in un file audio che provvederà ad inviare a un server Google, il quale procederà con l'analisi del contenuto e restituirà un certo numero di parole riconosciute, ognuna con la propria probabilità di confidenza. A questo punto il servizio confronta l'insieme di parole ottenute con le stringhe costanti che corrispondono ai

¹ <http://developer.android.com/reference/android/speech/package-summary.html>

comandi vocali, alla ricerca di un matching; infine restituisce il risultato della propria ricerca. I comandi vocali implementati nell'applicazione sono quattro:

- *play* per riprendere l'esecuzione di un brano sospeso
- *stop* per fermare la riproduzione del brano attualmente in ascolto
- *next* per passare al brano successivo
- “*previous* per tornare al brano precedente

7.3.3 Implementazione del riconoscimento dei gesti

Per l'invio e la ricezione dei comandi eseguiti con le gesture si è usata l'API Nanogest [7], che abbinano il riconoscimento di alcuni movimenti della mano a caratteristiche quali invarianza rispetto al colore della pelle e all'orientamento del dispositivo, funzionamento in condizioni ambientali non ottimali (scarsa o eccessiva luminosità, presenza di ombre, etc.) e un buon algoritmo per l'eliminazione dello sfondo. Affinché il riconoscimento possa avvenire correttamente, la mano deve essere rilevata per intero, ma allo stesso tempo non deve trovarsi troppo distante dalla fotocamera del dispositivo: la distanza di utilizzo raccomandata è di 20 cm circa. Queste API hanno permesso di gestire quattro movimenti della mano (verso l'alto, verso il basso, a sinistra e a destra) attraverso quattro costanti, che sono state messe in relazione con altrettante operazioni coerenti con il contesto dell'applicazione e intuitivamente associabili al movimento eseguito, così da essere anche facilmente memorizzabili dall'utente:

- **SWIPE_LEFT**: utilizzata per eseguire il brano precedente
- **SWIPE_RIGHT**: utilizzata per eseguire il brano precedente
- **SWIPE_UP**: per incrementare il livello del volume
- **SWIPE_DOWN**: per abbassare il livello del volume

Nel codice dell'applicazione sviluppata, dopo aver implementato l'interfaccia *Nanogest.GestureListener* fornita dalle API di cui sopra, il riconoscimento dei diversi gesti è stato possibile attraverso il metodo

```
public void onGesture(Gesture gesture, double timestamp)
```

dove il *gesture* indica una delle quattro azioni sopra descritte e il *timestamp* rappresenta il momento in cui l'azione è stata eseguita.

DOCUMENTO 8

CONCLUSIONE E SVILUPPI FUTURI



8.1 Risultati ottenuti

Questo progetto ha come obiettivo la creazione di un'applicazione mobile per la riproduzione di file musicali in formato mp3, con cui l'utente possa interagire utilizzando altre modalità oltre al classico touchscreen. A questo proposito si è optato per offrire due nuove opzioni: gesture e comandi vocali. Tutti i comandi più comunemente utilizzati in un'applicazione per la riproduzione di file musicali, quindi, sono stati associati a una delle due nuove modalità, dall'avvio/sospensione della riproduzione alla gestione del livello del volume al passaggio dal brano in ascolto a quello precedente o successivo. I risultati ottenuti sono stati molto buoni e in linea, se non superiori, con le nostre aspettative iniziali sia per quanto riguarda il riconoscimento dei comandi gestuali e vocali dati dall'utente, sia dal punto di vista del tempo di risposta tra il comando e l'esecuzione dell'operazione associata.

Il dispositivo che è stato impiegato nell'esecuzione dei test sulle funzionalità multimodali è un Nexus 5, con sistema operativo Android 4.4.2 (KitKat). Per quanto riguarda il riconoscimento dei gesti, tenendo la mano a una distanza minima di circa 20 cm dallo schermo (e quindi dalla fotocamera frontale), la percentuale di riconoscimento è stata molto elevata nell'individuazione del gesto in sé e nell'associazione tra il movimento della mano effettuato e il gesto che si voleva effettivamente far riconoscere all'applicazione. A titolo di esempio, si è notato che a parità di gesto eseguito e condizioni ambientali, il colore della pelle non influenza il riconoscimento ed è stato inoltre riscontrato che tenere la mano in modi diversi (palmo aperto, palmo chiuso, in posizione laterale rispetto alla fotocamera, etc.) influisce meno di quanto previsto sulla percentuale di successo del rilevamento; quest'ultima osservazione vale anche per i test eseguiti in differenti condizioni ambientali (illuminazione, presenza di ombre, etc.) e di sfondo. Ottimi i risultati per quanto riguarda tempo di risposta che è risultato essere inferiore a 1 secondo.

Relativamente al riconoscimento dei comandi vocali, i test eseguiti hanno dato un buon riscontro. Collegando un paio di cuffie al dispositivo, infatti, il riconoscimento per entrambi i comandi "play" e "stop" è stato in linea con le aspettative; lo stesso dicasì in presenza di rumore ambientale, come musica proveniente da altre sorgenti o persone che dialogano nelle vicinanze, a patto di trovarsi più vicino al microfono del dispositivo. Come evidenziato nel capitolo 7, la difficoltà maggiore nel riconoscimento dei comandi si è avuta nel caso di contemporaneo utilizzo delle casse del dispositivo come output audio in quanto, trovandosi fisicamente vicino al microfono, impediscono spesso l'identificazione del comando "stop". Il tempo di risposta tra il comando e la relativa esecuzione è di poco superiore a 1 secondo, buon valore se si considera l'invio dello stream audio via Internet ai server Google, la sua elaborazione, la ricezione delle parole riconosciute e il successivo matching tra queste e le stringhe che identificano i comandi vocali.

Il risultato ottenuto dal progetto, come si evince quindi anche dall'esito dei test, è un'applicazione che rispecchia l'obiettivo voluto e presenta chiare caratteristiche di multimodalità, funzionanti, efficienti e perfettamente integrate nel contesto di un'applicazione per la riproduzione di musica.



8.2 Sviluppi futuri

Per quanto riguarda i possibili sviluppi futuri dell'applicazione, sono state prese in considerazione diverse eventualità. Innanzitutto la risoluzione dei bug presenti, a mano a mano che essi verranno individuati, è un'attività praticamente obbligatoria; pur avendo sviluppato il sistema dopo un'attenta fase di studio e aver proceduto all'esecuzione di molti test, in particolare, sulle funzionalità multimodali, è pressoché impossibile rilasciare un sistema che sia del tutto privo di problemi. Ad esempio, uno dei bug attualmente non ancora risolti riguarda la modalità stand-by dello schermo: se l'applicazione è in primo piano e lo schermo si spegne passando in stand-by, nel momento in cui l'utente riattiva quest'ultimo la schermata di inserimento della password di sblocco del dispositivo non è immediatamente visualizzata. Un workaround temporaneo per evitare che si presenti questo problema consiste nel portare l'applicazione in background prima che lo schermo passi alla modalità stand-by. Poiché si tratta di un'applicazione per dispositivi mobili, uno dei punti cruciali dell'ottimizzazione nell'utilizzo delle risorse riguarda il consumo della batteria che è probabilmente la problematica maggiore dei più moderni smartphone e tablet. È quindi chiaro che, al di là dello sviluppo di nuove tipologie di batterie con un'efficienza più alta, l'ottimizzazione deve avere un'importanza rilevante in ogni momento dello sviluppo. Per quanto riguarda la nostra applicazione, è evidente che le funzionalità multimodali introdotte siano le più critiche per il consumo della batteria: la modalità di riconoscimento dei gesti, quando attiva, richiede infatti che la fotocamera frontale del dispositivo sia sempre accesa, mentre il riconoscimento vocale, come visto nel capitolo 7, fa uso continuo sia del microfono che della connessione Internet. Una delle aree su cui focalizzare maggiormente l'attenzione nello sviluppo dell'applicazione successivo al rilascio sarà la riduzione del consumo delle risorse e, di conseguenza, della batteria dei dispositivi nei quali è installata. Negli screenshot presenti nel paragrafo riguardante l'implementazione della GUI, è possibile notare, nella zona centrale di quest'ultima, un'immagine standard che rimane sempre fissa durante la riproduzione dei brani. Tale spazio è stato riservato con l'intenzione di visualizzare, al centro dello schermo, un'immagine associata alla canzone in riproduzione come, ad esempio, la copertina dell'album o del singolo da cui essa è tratta oppure una foto dell'autore. Sarebbe quindi interessante sviluppare, in futuro, una funzione che permetta di estrarre dal brano attualmente in ascolto l'immagine solitamente memorizzata nelle informazioni del file mp3 associato e di visualizzarla, quando presente, al posto di quella standard attualmente in uso per tutte le canzoni. Infine, per quanto riguarda la funzionalità di riconoscimento dei comandi vocali, è stato accennato nel capitolo 7 che al momento sono stati implementati quattro comandi: "play", "stop", "next" e "previous". Uno degli sviluppi futuri potrebbe prevedere, a questo proposito, l'inserimento di nuovi comandi, ad esempio "shuffle" e "repeat" per attivare le modalità corrispondenti, oppure "up" e "down" per aumentare e diminuire il volume della riproduzione.

BIBLIOGRAFIA

- [1] Da Wikipedia "http://it.wikipedia.org/wiki/Deployment_Diagram".
- [2] Diagrammi dei componenti UML: linee guida <http://msdn.microsoft.com/it-it/library/dd409393.aspx>
- [3] Université catholique de Louvain (UCL), Louvain School of Management, (UnGit) of Information Systems, Belgian Laboratory of Computer-Human Interaction. UsiXML, User Interface eXtensible Markup Language.
- [4] François Beuvens, Vivian Genaro Motti, Jérémie Melchior, Benoît Michel, Ricardo Tesoriero, Jean Vanderdonckt. UsiXML – General Description.
- [5] <http://www.usixml.org/en/home.html?IDC=221>
- [6] Eleonora Bilotta. La psicologia della Gestalt.
- [7] <http://www.nanocritical.com/nanogest/>