

Relazione Tecnica

Vinyl Collection (Flutter)

Mobile Programming - Università degli Studi di Salerno
Anno Accademico 2024/2025
Biele Pasquale – Cinque Pierluigi – De Stefano Nicolas Pio

Sommario

1. Descrizione Generale	2
2. Funzionalità Principali	2
3. Scelte Tecnologiche	2
4. Struttura del Codice	2
5. Motivazioni delle Scelte	3
6. UI/UX	4
7. Limiti Attuali e Sviluppi Futuri	4

Relazione Tecnica – Vinyl Collection

1. Descrizione Generale

Vinyl Collection è un'applicazione mobile sviluppata in Flutter che consente all'utente di gestire una collezione personale di dischi in vinile. L'app è pensata per collezionisti e appassionati di musica che desiderano organizzare, catalogare e analizzare i propri vinili in modo efficiente e intuitivo.

2. Funzionalità Principali

- **Visualizzazione Home:** panoramica generale con vinili posseduti e suggeriti.
- **Ricerca Avanzata:** filtro dinamico per trovare vinili nella collezione.
- **Gestione Collezione:** aggiunta, modifica e rimozione dei vinili.
- **Categorie e Generi:** organizzazione per generi musicali con aggiunta, modifica e rimozione.
- **Dettagli Vinile:** scheda informativa completa per ogni elemento.
- **Analisi e Statistiche:** dashboard riepilogativa.
- **Integrazione con Discogs:** API esterna per ottenere info e vinili.

3. Scelte Tecnologiche

Flutter:

- Scelto per la sua portabilità cross-platform e ottima UI nativa.
- Permette hot reload e sviluppo rapido.

Dart:

- Linguaggio ufficiale per Flutter, con sintassi moderna e orientata agli oggetti.

SQLite:

- Per la persistenza locale dei dati, ideale per app offline-first.

Discogs API:

- Permette di arricchire automaticamente i dati dei vinili con info reali.

4. Struttura del Codice

Entry Point – main.dart:

- Avvia l'app e imposta il MainScaffold, ovvero la navigazione principale.

Schermate Principali – schermoPrincipali/:

- homepage.dart: schermata iniziale con suggerimenti e overview.
- schermata_collezione.dart: mostra l'intera collezione salvata.
- schermata_ricerca.dart: implementa la ricerca di vinili tramite l'API.
- schermata_categorie.dart: visualizza vinili per genere.

Schermate Secondarie – schermoSecondarie/:

- schermata_aggiungi.dart: form per inserire un nuovo vinile.
- schermata_modifica.dart: modifica dei dati esistenti.
- dettaglio_vinile_collezione.dart: dettaglio scheda vinile presente nella collezione.
- dettaglio_vinile_suggerito.dart: dettaglio scheda vinile suggerito dall'API.
- schermata_analisi_vinili.dart: statistiche e analisi visive.

Componenti Riutilizzabili – componenti/:

- filtro_ricerca_widget.dart: widget per filtri di ricerca.
- suggestion_tile.dart, genere_tile.dart: UI dei singoli elementi.

Modelli – vinile/:

- vinile.dart: classe principale che rappresenta un disco.
- genere.dart, condizione.dart: modelli ausiliari.

Servizi – discogs_service.dart:

- Gestisce le richieste alle API esterne di Discogs.

Utilità – dimensioni_schermo.dart:

- Gestione dinamica del layout rispetto alla dimensione dello schermo.

5. Motivazioni delle Scelte

- **Separazione logica**: cartelle ben organizzate secondo l'architettura Model-View - View Model.
- **Modularità**: componenti UI separati e riutilizzabili.
- **Scalabilità**: codice pronto per futuri miglioramenti.

6. UI/UX

- Interfaccia moderna e coerente grazie a Material 3.
- Navigazione semplice con BottomNavigationBar.
- Colori, padding e icone usati per migliorare l'esperienza utente.
- Responsività garantita da dimensioni_schermo.dart.

7. Limiti Attuali e Possibili Sviluppi Futuri

Limiti:

- Mancanza di autenticazione utente.
- Nessuna sincronizzazione cloud (solo dati locali).

Possibili Sviluppi Futuri:

- Implementazione login (Firebase o OAuth).
- Backup/sync su cloud.
- Dashboard più avanzata con grafici interattivi.