

Traccia d'esame - Progetto gestione digitale delle carriere musicali Music Career

Daniele Visone 0334000130

Il progetto Music Career ha come obiettivo la realizzazione di un sistema informativo per la gestione delle carriere musicali di artisti e manager. L'applicativo web è stato sviluppato utilizzando Django, un framework web basato su Python, che consente un'implementazione sicura, scalabile e facilmente mantenibile di sistemi informativi relazionali. Il sistema consente di: gestire rapporti contrattuali tra manager e artisti, registrare e visualizzare brani pubblicati e concerti organizzati, per i manager la possibilità di monitorare i propri artisti e, se necessario, azzerare le statistiche.

L'applicazione è stata sviluppata utilizzando per tutto il ciclo di vita della progettazione Django, un framework web basato su Python, che consente in modo sicuro di implementare sistemi informativi relazionali, nel nostro caso il database a cui fa affidamento Django è SQLite3. Il front-end dell'applicazione è realizzato con HTML5 e CSS.

Analisi e progettazione concettuale

Il dominio applicativo riguarda la gestione delle carriere artistiche musicali. Ogni Utente può registrarsi come Artista o Manager. Gli artisti pubblicano brani e partecipano a concerti, mentre i manager stipulano contratti di collaborazione con gli artisti per gestirne le loro attività.

Entità individuabili nel progetto

1) Utente: gestisce l'autenticazione e i dati base dell'account (username, password ed email).

- 2) **Artista**: contiene le informazioni aggiuntive sull'artista. È collegato a un utente e include genere musicale, guadagni totali e riferimenti ai brani e ai concerti.
- 3) **Manager**: Rappresenta l'utente supervisore dell'artista che gestisce contratti e artisti.
- 4) **Contratto**: Relazione tra artista e un manager. Ha come attributi la data di inizio e fine del rapporto lavorativo.
- 5) **Traccia**: Rappresenta un brano pubblicato da un artista. Contiene informazioni su titolo, data di pubblicazione, genere musicale, numero di stream Spotify (concettuale, non implementato con API) e ricavi derivanti.
- 6) **Concerto**: Evento dal vivo organizzato per un artista. Contiene data, luogo e guadagno ottenuto.

Relazioni tra entità

Entità 1	Entità 2	Tipo di relazioni
Utente<->Artista	1:1	Ogni artista corrisponde ad un utente
Utente<->Manager	1:1	Ogni manager corrisponde a un utente
Artista<->Traccia	N:N	Un artista può avere più manager e viceversa
Artista<->Traccia	1:N	Un artista può pubblicare più tracce
Artista<->Concerto	1:N	Un artista può partecipare a più concerti

Modello E-R (entità relazioni)

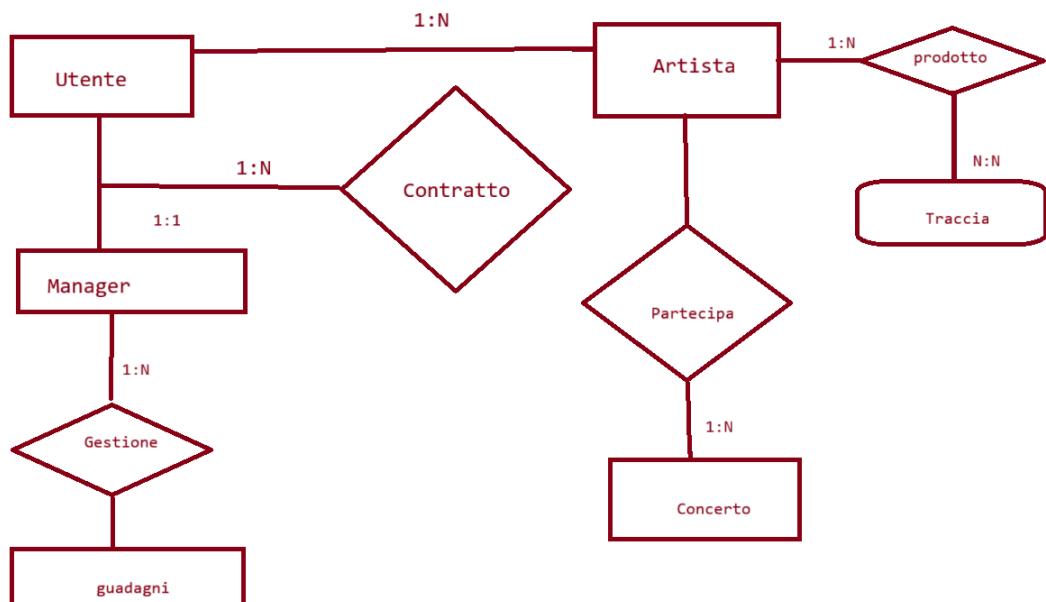
Il modello ER è uno strumento grafico e concettuale utilizzato per progettare e rappresentare la struttura logica di una base di dati.

Serve per descrivere quali dati devono essere memorizzati, quali dati sono collegati tra loro.

Il modello ER è formato da:

Le entità rappresentano gli oggetti o concetti nel mondo reale che si vogliono memorizzare e si rappresentano con rettangoli.

Le relazioni descrivono i legami tra le entità e sono rappresentate tramite rombi.



Progettazione logica - Struttura tabelle principali

Tabella Utente:

Campo	Tipo	Chiave	Descrizione
id	INT	PK	identificativo univoco
username	VARCHAR		nome utente
password	VARCHAR		password
email	VARCHAR		email dell'utente

Tabella Artista:

Campo	Tipo	Chiave	Descrizione
id	INT	PK	identificativo artista
id_utente	INT	FK	collegamento esterno alla tabella Utente
genere	VARCHAR		genere musicale
guadagni_totali	DECIMAL		somma dei guadagni totali

Tabella: Manager

Campo	Tipo	Chiave	Descrizione
id	INT	PK	identificativo manager
id_utente	INT	FK	collegamento esterno alla tabella Utente

Tabella: Contratto

Campo	Tipo	Chiave	Descrizione
id	INT	PK	identificativo contratto
id_artista	INT	FK	collegamento esterno artista
inizio_data	DATE		data inizio contratto
fine_data	DATE		data fine

			contratto
--	--	--	-----------

Tabella Traccia:

Campo	Tipo	Chiave	Descrizione
id	INT	PK	identificativo brano
id_artista	INT	FK	artista autore del brano
titolo	VARCHAR		titolo del brano
data_pubblicazione	DATE		data pubblicazione
stream_spotify	INT		Numero di stream
guadagni	DECIMAL		guadagno totale

Tabella: Concerto

Campo	Tipo	Chiave	Descrizione
id	INT	PK	identificativo concerto
artist_id	INT	FK	artista protagonista
luogo	VARCHAR		luogo del concerto
guadagno	DECIMAL		guadagno ottenuto

Implementazione del sistema informativo

Il sistema è sviluppato con Django che fornisce:

1) un'architettura Model-View-Template (MVT paradigma)

- 2) gestione integrata di utenti, sessioni e autenticazione (hashing delle password bcrypt integrato)
- 3) interfaccia amministrativa per la gestione dei dati.

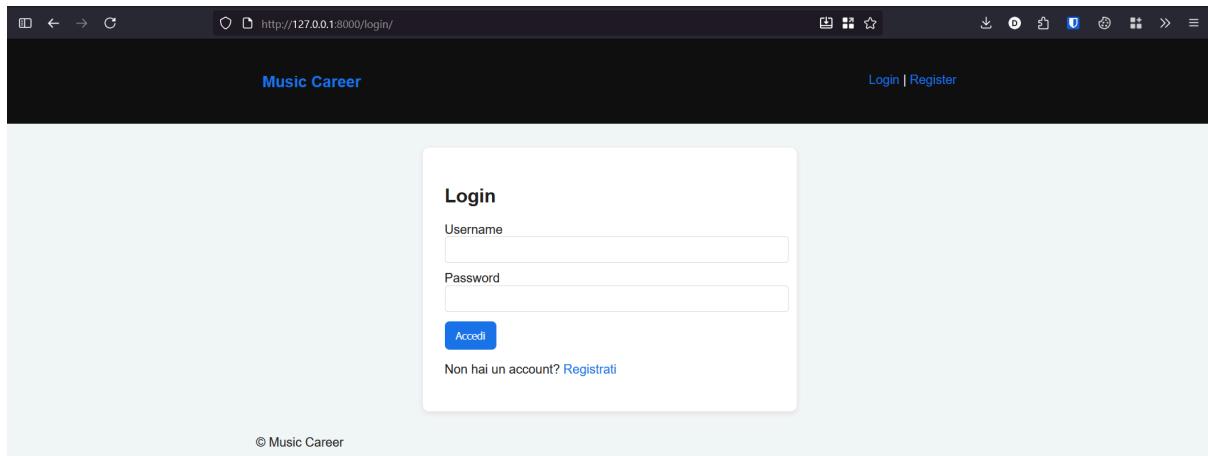
In particolare l'artista può compiere le seguenti funzionalità

- Registrazione tramite form dedicato
- Accesso alla dashboard personale
- Visualizzazione di: contratti attivi (data inizio e fine), brani pubblicati e ricavi per ciascuno, concerti organizzati e relativi guadagni, somma totale dei guadagni derivanti da streaming e concerti.

Per quanto riguarda il manager questo può:

- Visualizzare la lista di artisti gestiti.
- Accesso ai dati relativi ai contratti, brani e concerti.
- Possibilità di azzerare le statistiche di un artista (ricavi, concerti, stream ecc)

In allegato alcune foto dell'applicativo:



Landing page del sito

Music Career

Ciao firstartist [Logout](#)

Aggiungi brano

Title:

Release date:

Spotify streams:

Spotify earnings:

Salva

Inserimento dati artista

The screenshot shows a web browser window with the URL <http://127.0.0.1:8000>. The page title is "Music Career". On the right, it says "Ciao firstartist" and has a "Logout" link. The main content area has a yellow header bar with the text "Brano aggiunto". Below it, the heading "Artist: firstartist" is followed by "Genere: Nessun contratto". There are two blue buttons: "+ Aggiungi brano" and "+ Aggiungi concerto". A section titled "Brani" contains a single item: "• X Agosto — Streams: 20 — Guadagni: 0,05". A section titled "Concerti" contains the message "• Nessun concerto". At the bottom, the total earnings are displayed as "Guadagni totali: 0,05000000000000000". The footer says "© Music Career".

Dashboard artista aggiornata

The screenshot shows a web browser window with the following details:

- Address Bar:** http://127.0.0.1:8000
- Page Title:** Music Career
- User Information:** Ciao firstmanager [Logout](#)
- Section Header:** Manager: firstmanager
- Section Subtitle:** Artisti
- List of Artists:**
 - io — Genere: — Guadagni: 0,00 | [Vedi](#) | [Azzerare e cancella dati](#)
 - manager1 — Genere: — Guadagni: 0,00 | [Vedi](#) | [Azzerare e cancella dati](#)
 - firstartist — Genere: — Guadagni: 0,0500000000000000 | [Vedi](#) | [Azzerare e cancella dati](#)
- Page Footer:** © Music Career

Dashboard manager con la possibilità di azzerare tutte le statistiche di tutti gli artisti.

Possibili sviluppi futuri

Integrazione API Spotify per ottenere in tempo reale statistiche sugli stream, gestione pagamenti e royalties e dashboard analitiche con grafici di performance per artista.