

**POORIFY**  
**System Design Document**  
Versione 1.1 / 03.12.2022



# Poorify

Anna Linda Brenga	0512109669
Alessandro Farina	0512109741
Christian Mascolo	0512111139
Jessica Zampetti	0512109894

# INDICE

- 1. INTRODUZIONE.....3
  - 1.1 SCOPO DEL SISTEMA .....3
  - 1.2 OBIETTIVI DI DESIGN .....3
  - 1.3 DEFINIZIONI, ACRONIMI E ABBREVIAZIONI .....4
  - 1.4 RIFERIMENTI .....4
  - 1.5 OVERVIEW .....4
- 2. SISTEMA ATTUALE .....5
- 3. SISTEMA PROPOSTO .....6
  - 3.1 PANORAMICA .....6
  - 3.3 HARDWARE/SOFTWARE MAPPING.....7
  - 3.4 PERSISTENT DATA MANAGEMENT.....8
  - 3.5 ACCESS CONTROL AND SECURITY .....9
  - 3.6 GLOBAL SOFTWARE CONTROL.....10
  - 3.7 BOUNDARY CONDITIONS.....10
    - 3.7.1 USE CASES BOUNDARY CONDITIONS .....11
- 4. SUBSYSTEM SERVICES .....14

# 1. INTRODUZIONE

## 1.1 SCOPO DEL SISTEMA

Poorify è una piattaforma di streaming musicale che si pone come obiettivo quello di semplificare la fruizione di opere musicali attraverso un sistema gratuito e user-friendly. In particolare, Poorify punta a fornire agli utenti uno strumento che permetta loro di riprodurre liberamente brani presenti, creare playlist e interagire con altri iscritti, e agli artisti una piattaforma dove eseguire l'upload dei propri lavori al fine di accrescere la propria fama e costruire una fanbase.

## 1.2 OBIETTIVI DI DESIGN

Label	Criteria Group	Design Criterion	Description	Origin
DG1	Performance	Memory	In base alle stime effettuate il server utilizzato deve avere una capacità di 250GB.	RNF4, 5, 6 & 7
DG2	Dependability	Robustness	Il sistema deve notificare l'utente in caso di input errato e gestire propriamente le possibili eccezioni.	Dominio Applicativo
DG3	Dependability	Reliability	L'output delle operazioni deve coincidere con quanto specificato negli Use Cases e quindi con le aspettative degli utenti.	Dominio Applicativo
DG4	Dependability	Security	Il sistema non deve permettere agli utenti di accedere a pagine per cui non possiedono l'autorizzazione tramite dei filtri.	RNF2
DG5	Maintenance	Extensibility	Il sistema deve facilitare la futura aggiunta di nuove funzionalità.	Dominio Applicativo
DG6	Maintenance	Readability	Il codice deve essere modulare e commentato in modo da semplificarne la lettura e la comprensione.	RNF8
DG7	End User	Usability	Il sistema deve notificare l'utente circa l'esito delle operazioni in caso di errore con degli opportuni messaggi.	RNF1

Trade-Off	Rational
Space vs Speed	Il sistema deve garantire l'accesso a una grande quantità di contenuti, questo potrebbe causare dei rallentamenti durante la ricerca o la fruizione degli stessi.
Delivery Time vs Functionality	È richiesto di sviluppare l'intero sistema proposto, implementando tutti gli Use Cases, entro i tempi previsti per le deadlines.
Delivery Time vs Usability	Per vincoli di tempo si è deciso di non rendere la grafica della piattaforma responsive.

### 1.3 DEFINIZIONI, ACRONIMI E ABBREVIAZIONI

- Piattaforma: Applicazione Web (Poorify)
- RAD: Requirements Analysis Document
- UCBC: Use Case Boundary Condition
- RNF: Requisito Non Funzionale
- DG: Design Goal

### 1.4 RIFERIMENTI

Nel corso del documento facciamo riferimento ai Requisiti Non Funzionali, agli Use Cases e al Class Diagram presenti nel documento RAD.

### 1.5 OVERVIEW

Nei paragrafi successivi presentiamo:

- L'analisi dell'architettura di una piattaforma concorrente a Poorify.
- La decomposizione del sistema in sottosistemi.
- La strategia di deployment su hardware.
- La strategia adottata per la gestione dei dati persistenti.
- Le varie Boundary Condition del sistema.
- I servizi offerti dai vari sottosistemi.

## **2. SISTEMA ATTUALE**

Il principale concorrente di Poorify è il noto servizio di streaming Spotify. La piattaforma del colosso svedese si basa fundamentalmente su un'architettura orientata ai servizi con un backend scritto prevalentemente in Python e Java. I messaggi vengono scambiati tramite protocollo Hermes, che sostituisce la più classica soluzione enterprise di HTTP e XML. Il principale strumento di persistenza impiegato è PostgreSQL, un DBMS a oggetti, mentre le tracce audio sono salvate su server Amazon S3. La piattaforma sfrutta inoltre machine learning e IA per fornire contenuti personalizzati in base ai gusti dell'utente.

## 3. SISTEMA PROPOSTO

### 3.1 PANORAMICA

Poorify è una piattaforma browser distribuita.

Nei paragrafi che seguono illustriamo l'architettura di massima dell'intero sistema. Elenchiamo brevemente le componenti principali dell'applicazione:

- Un front-end web dinamico sviluppato con JSP, JS e CSS.
- Un back-end per la logica di business basato su Servlet Java.
- Un insieme di classi DAO che collegano la logica alla persistenza tramite JDBC.
- Database relazionale SQL e File System su server in locale per la persistenza dei dati.
- Una serie di terminali Python per le operazioni esterne dell'amministratore.

### 3.2 SUBSYSTEM DECOMPOSITION

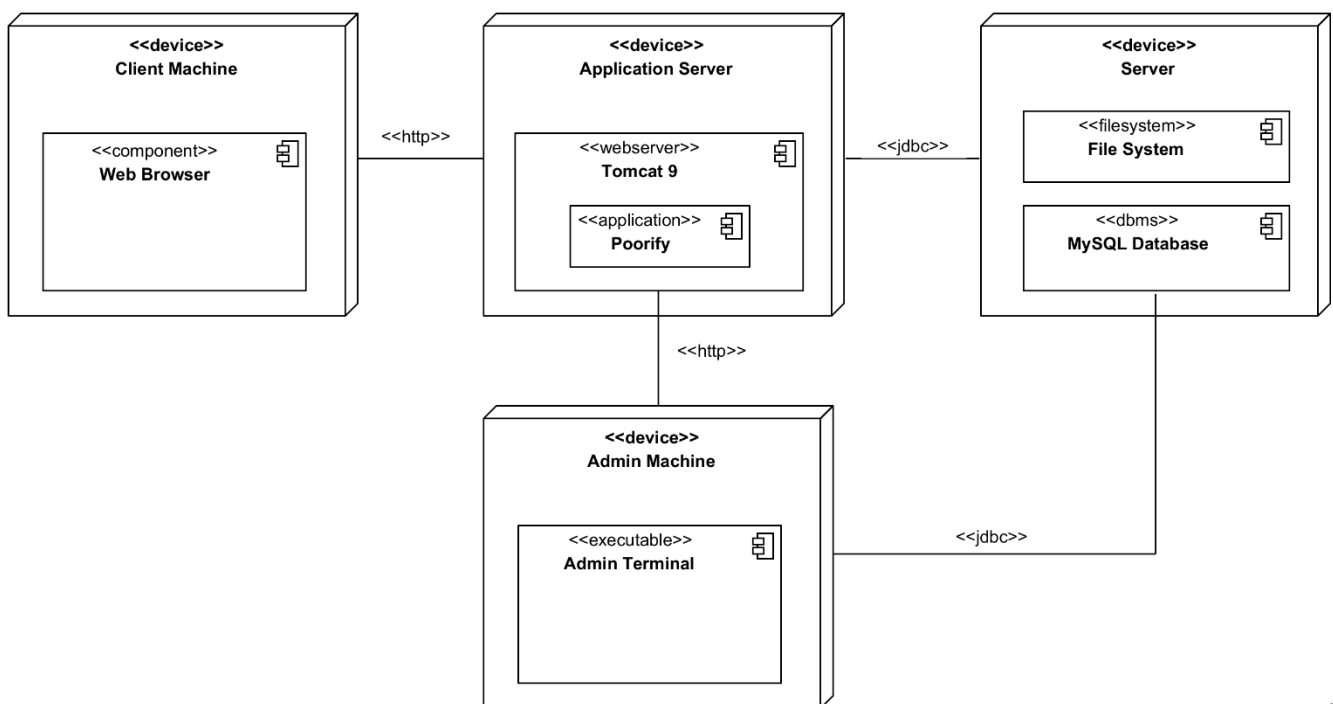
La piattaforma Poorify sfrutta una architettura web three-tier:

- **Presentation layer:** Interfaccia impiegata dagli utenti finali per interagire con la piattaforma tramite web browser dalle loro macchine client.
- **Logic layer:** La logica di business che gestisce le operazioni del sistema. Dividiamo tale strato in cinque sottosistemi/partizioni:
  - *Gestione Utente:* per la registrazione, l'autenticazione, la modifica e l'eliminazione del profilo dei fruitori.
  - *Gestione Playlist:* per la creazione, l'eliminazione, aggiunta/rimozione di brani e la modifica delle playlist (inclusa la possibilità di rendere la playlist pubblica/privata e singola/collaborativa).
  - *Gestione Album:* per l'upload e l'eliminazione degli album.
  - *Gestione Navigazione:* per la ricerca dei contenuti e la navigazione tra le varie pagine della piattaforma.
  - *Gestione Riproduzione:* per la riproduzione dei brani e la gestione della coda di ascolti.
- **Data layer:** Lo strato software che permette di effettuare le operazioni su database necessarie al funzionamento della piattaforma.

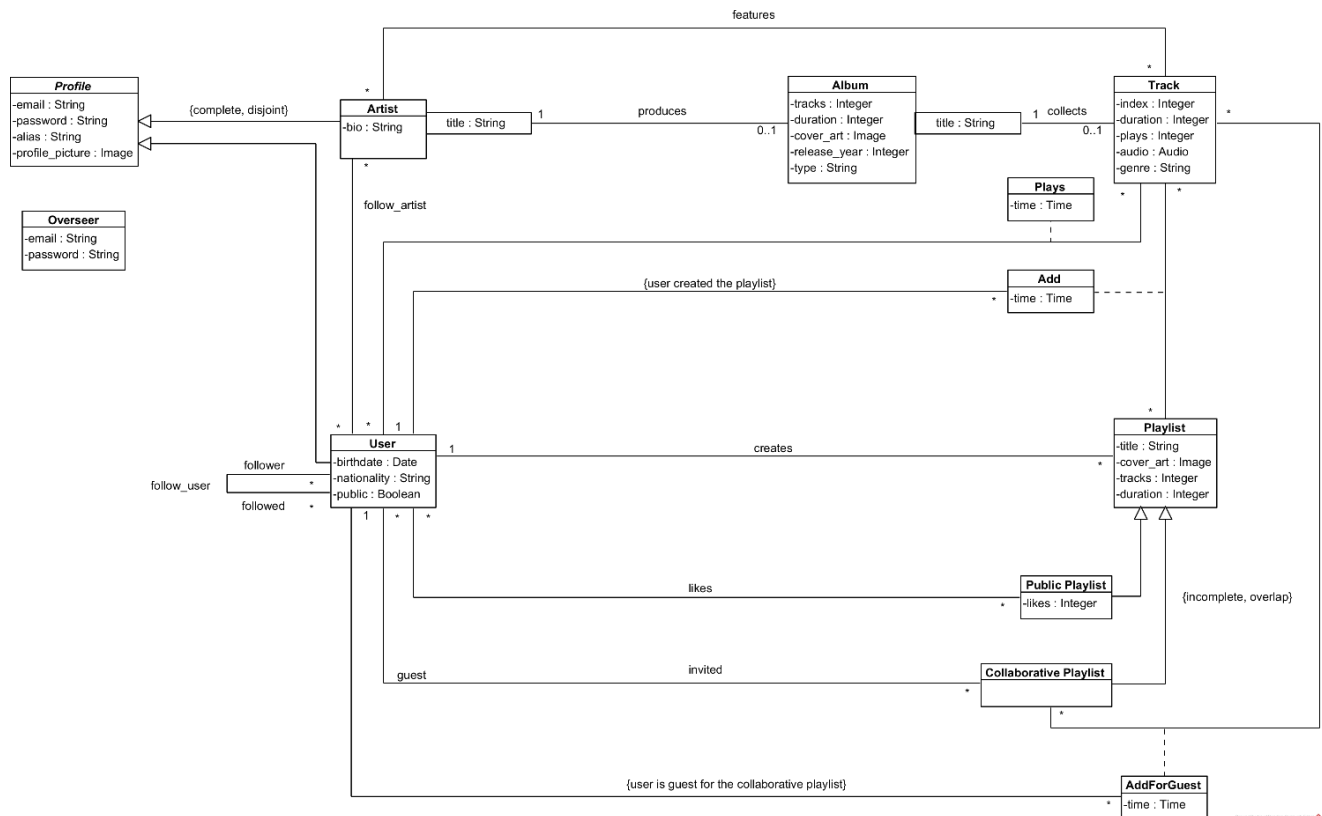
### 3.3 HARDWARE/SOFTWARE MAPPING

Gli utenti accedono alla piattaforma Poorify tramite web browser dalla propria macchina client stabilendo una connessione http con un server. L'applicazione viene eseguita nel web container Tomcat che interagisce con un database relazionale SQL, potenzialmente distribuito su molteplici macchine server.

Gli amministratori del sistema possono interagire sia con Tomcat per effettuare startup e shutdown del sistema, sia operare direttamente su database tramite un terminale dedicato per aggiungere nuovi impiegati supervisor o lanciare delle query per l'aggiornamento o la creazione di contenuti.



### 3.4 PERSISTENT DATA MANAGEMENT



Facciamo riferimento al class diagram per selezionare quali attributi memorizzare su database relazionale e per quali invece impiegare il file system. In linea generale preferiamo il file system per dati voluminosi, come immagini e tracce audio.

I file multimediali sono acceduti tramite un path relativo a partire da una cartella di default all'interno del server in locale, è quindi possibile far migrare il sistema su altre macchine server senza dover cambiare tali riferimenti.

La root del nostro file system presenta le seguenti sottocartelle:

- **/Profile**: per *profile\_picture* della classe *Profile*, e di conseguenza *User* e *Artist*.
- **/Album**: per *cover\_art* della classe *Album*.
- **/Playlist**: per *cover\_art* della classe *Playlist*.
- **/Track**: per *audio* della classe *Track*.

I file salvati sono denominati *x.estensione* dove *x* è l'id associato alla entry corrispondente nel database. Ad esempio *profile\_picture* dell'utente/artista con id 27 è salvata come 27.png nella sottocartella */Profile* mentre l'attributo *audio* del brano con id 45 è salvato come 45.mp3 nella sottocartella */Track*.

Le restanti informazioni presenti nel class diagram, essendo meno complesse e perlopiù testuali o numeriche, sono rese persistenti su database.



### 3.5 ACCESS CONTROL AND SECURITY

Illustriamo nella seguente tabella le operazioni che ogni tipologia di utente può eseguire all'interno della piattaforma.

	Utente Non Registrato	Utente	Artista	Supervisore
Gestione Utente	Registrazione	Autenticazione Disconnessione Modifica Profilo Visualizza Profilo Eliminazione Profilo Seguire (Smettere di seguire) Artista o Utente Pubblico	Autenticazione Disconnessione Modifica Profilo Visualizza Profilo Eliminazione Profilo	Autenticazione Disconnessione Eliminazione Profilo Altri
Gestione Playlist		Creazione Playlist Rendere Singola o Collaborativa Rendere Pubblica o Privata Invitare Guest per Collaborativa Aggiungere un Brano Eliminare un Brano Lasciare Like Modifica Playlist Cambio Ordine Brani Eliminazione Playlist		Eliminazione Playlist
Gestione Album			Upload Album Eliminazione Album	Eliminazione Album
Gestione Navigazione		Ricerca Contenuti Visualizza Pagina Artista Visualizza Pagina Album Visualizza Pagina Playlist Visualizza Pagina Utente Navigazione Pagine Generazione Feed Utente	Visualizza Pagina Album Navigazione Pagine	Ricerca Contenuti Visualizza Pagina Artista Visualizza Pagina Album Visualizza Pagina Playlist Visualizza Pagina Utente Navigazione Pagine
Gestione Riproduzione		Riproduzione Brano Riproduzione Playlist/Album Aggiungere Brano alla Coda Saltare Brano in Coda		Riproduzione Brano

La piattaforma sfrutta dei filtri per garantire la sicurezza impedendo agli utenti di eseguire operazioni e accedere a pagine per cui non possiedono l'autorizzazione. Inoltre i supervisori hanno il compito di eliminare i contenuti ritenuti offensivi o inopportuni e nel caso eliminare il profilo del responsabile.

### 3.6 GLOBAL SOFTWARE CONTROL

L'architettura della piattaforma Poorify è intrinsecamente Event-driven in quanto le servlet reagiscono agli eventi generati dall'input utente, ciò permette di aggiungere nuove funzionalità in maniera semplice senza dover modificare il codice già scritto. L'applicazione è anche distribuita e quindi alcune operazioni su database possono causare delle indesiderate race conditions. Risulta pertanto necessario gestire la concorrenza tramite thread e applicare dei meccanismi di sincronizzazione per garantire la coerenza delle risorse della piattaforma.

### 3.7 BOUNDARY CONDITIONS

Startup e shutdown del sistema sono gestiti tramite due eseguibili Python.

- *startup.py* effettua la configurazione del sistema. In questa fase viene creato e popolato, previo inserimento delle credenziali dell'amministratore del sistema, il database MySQL impiegato dalla piattaforma, e viene eseguito lo startup del web container Tomcat.
- *overseer.py* permette di gestire (aggiungere/cambiare/eliminare) i supervisori della piattaforma tramite alcune query predefinite.
- *wrapped.py* per generare per ciascun utente una playlist con i suoi brani più ascoltati durante l'anno trascorso.
- *charts.py* per generare delle classifiche dei brani più ascoltati in ogni nazione dove è disponibile il servizio.
- *shutdown.py* effettua lo shutdown del web container Tomcat, la piattaforma risulterà quindi offline.

Il sistema deve risultare robusto nella gestione dell'input utente, ad esempio interrompendo l'operazione sul nascere. In caso di eccezioni o fallimenti non previsti però non si potrà garantire il corretto funzionamento della piattaforma e potrebbe risultare necessario un riavvio.

### 3.7.1 USE CASES BOUNDARY CONDITIONS

#### UCBC-1: STARTUP

USE CASE NAME	Startup
PARTICIPATING ACTORS	Amministratore del Sistema
ENTRY CONDITION	<ul style="list-style-type: none"><li>L'amministratore esegue <i>startup.py</i>.</li></ul>
FLOW OF EVENTS	<ol style="list-style-type: none"><li>L'amministratore fornisce username e password del database locale.</li><li>Il programma provvede a creare e popolare il database per applicazione tramite i file <i>creazione.sql</i> e <i>popolamento.sql</i>.</li><li>Il programma provvede a effettuare il deploy del file <i>Poorify.war</i> sul web container Tomcat.</li><li>Il programma effettua lo startup di Tomcat.</li></ol>
EXIT CONDITION	<ul style="list-style-type: none"><li>La piattaforma Poorify è online.</li></ul>
EXCEPTIONS	<ul style="list-style-type: none"><li>Se al punto 1 le credenziali fornite sono errate, l'amministratore viene notificato con un messaggio di errore e può ritentare l'autenticazione. (UCBC-1.1: CredenzialiErrate)</li><li>Se al punto 2 startup.py non trova creazione.sql o popolamento.sql il programma termina con un messaggio di errore. (UCBC-1.2: FileSQLNonTrovati)</li><li>Se al punto 3 startup.py non trova Poorify.war il programma termina con un messaggio di errore. (UCBC-1.3: FileWarNonTrovato)</li><li>Se al punto 4 startup.py non riesce a avviare Tomcat il programma termina con un messaggio di errore. (UCBC-1.4: TomcatNonAvviato)</li></ul>

#### UCBC-2: SHUTDOWN

USE CASE NAME	Shutdown
PARTICIPATING ACTORS	Amministratore del Sistema
ENTRY CONDITION	<ul style="list-style-type: none"><li>L'amministratore esegue <i>shutdown.py</i>.</li></ul>
FLOW OF EVENTS	<ol style="list-style-type: none"><li>L'amministratore conferma di voler eseguire lo shutdown del sistema.</li><li>Il programma effettua lo shutdown del web container Tomcat.</li></ol>
EXIT CONDITION	<ul style="list-style-type: none"><li>La piattaforma Poorify è offline.</li></ul>
EXCEPTIONS	<ul style="list-style-type: none"><li>Se al punto 1 l'amministratore non conferma lo shutdown, il programma termina normalmente e la piattaforma è ancora online. (UCBC-2.1: ShutdownRimandato)</li></ul>

### UCBC-3: ADD OVERSEER

USE CASE NAME	Add Overseer
PARTICIPATING ACTORS	Amministratore del Sistema
ENTRY CONDITION	<ul style="list-style-type: none"><li>L'amministratore esegue <i>overseer.py</i> selezionando l'opzione aggiungi.</li></ul>
FLOW OF EVENTS	<ol style="list-style-type: none"><li>L'amministratore fornisce username e password del database locale.</li><li>L'amministratore inserisce <i>email</i> e <i>password</i> del nuovo supervisore.</li><li>Il programma verifica che i campi rispettino il formato richiesto.</li><li>Il programma verifica che l'email non sia già utilizzata.</li><li>Il programma salva i dati su database.</li></ol>
EXIT CONDITION	<ul style="list-style-type: none"><li>Il supervisore è stato aggiunto con successo.</li></ul>
EXCEPTIONS	<ul style="list-style-type: none"><li>Se al punto 1 le credenziali fornite sono errate, l'amministratore viene notificato con un messaggio di errore e può ritentare l'autenticazione. (UCBC-1.1: CredenzialiErrate)</li><li>Se al punto 3 uno dei campi non rispetta il formato richiesto, il programma mostra un messaggio di errore e l'amministratore può ritentare l'inserimento. (UCBC-3.1: FormatoNonValido)</li><li>Se al punto 4 l'email inserita risulta già utilizzata, il programma mostra un messaggio di errore e l'amministratore può ritentare l'inserimento. (UCBC-3.2: EmailGiàInUso)</li></ul>

### UCBC-4: REMOVE OVERSEER

USE CASE NAME	Remove Overseer
PARTICIPATING ACTORS	Amministratore del Sistema
ENTRY CONDITION	<ul style="list-style-type: none"><li>L'amministratore esegue <i>overseer.py</i> selezionando l'opzione rimuovi.</li></ul>
FLOW OF EVENTS	<ol style="list-style-type: none"><li>L'amministratore fornisce username e password del database locale.</li><li>L'amministratore inserisce l'<i>email</i> del supervisore da rimuovere.</li><li>Il programma verifica l'esistenza del supervisore.</li><li>Il programma chiede conferma all'amministratore per procedere.</li><li>L'amministratore conferma la propria decisione.</li><li>Il programma elimina i dati del supervisore dal database.</li></ol>
EXIT CONDITION	<ul style="list-style-type: none"><li>Il supervisore è stato rimosso con successo.</li></ul>
EXCEPTIONS	<ul style="list-style-type: none"><li>Se al punto 1 le credenziali fornite sono errate, l'amministratore viene notificato con un messaggio di errore e può ritentare l'autenticazione. (UCBC-1.1: CredenzialiErrate)</li><li>Se al punto 3 non viene trovata l'email fornita, il programma mostra un messaggio di notifica, "Email Non Trovata" per e l'amministratore può ritentare l'inserimento. (UCBC-4.1: EmailNonTrovata)</li><li>Se al punto 5 l'amministratore non effettua la conferma il programma termina normalmente. (UCBC-4.2: RimozioneRimandata)</li></ul>

#### UCBC-5: UPDATE OVERSEER

USE CASE NAME	Update Overseer
PARTICIPATING ACTORS	Amministratore del Sistema
ENTRY CONDITION	<ul style="list-style-type: none"><li>L'amministratore esegue <i>overseer.py</i> selezionando l'opzione cambia.</li></ul>
FLOW OF EVENTS	<ol style="list-style-type: none"><li>L'amministratore fornisce username e password del database locale.</li><li>L'amministratore inserisce l'<i>email</i> del supervisore da rimuovere.</li><li>Il programma verifica l'esistenza del supervisore.</li><li>L'amministratore inserisce la nuova <i>password</i>.</li><li>Il programma verifica che i campi rispettino il formato richiesto.</li><li>Il programma aggiorna i dati del supervisore sul database.</li></ol>
EXIT CONDITION	<ul style="list-style-type: none"><li>Il supervisore è stato aggiornato con successo.</li></ul>
EXCEPTIONS	<ul style="list-style-type: none"><li>Se al punto 1 le credenziali fornite sono errate, l'amministratore viene notificato con un messaggio di errore e può ritentare l'autenticazione. (UCBC-1.1: CredenzialiErrate)</li><li>Se al punto 3 non viene trovata l'email fornita, il programma mostra un messaggio di notifica, "Email Non Trovata" per e l'amministratore può ritentare l'inserimento. (UCBC-4.1: EmailNonTrovata)</li><li>Se al punto 5 uno dei campi non rispetta il formato richiesto, il programma mostra un messaggio di errore e l'amministratore può ritentare l'inserimento. (UCBC-3.1: FormatoNonValido)</li></ul>

#### UCBC-6: CHARTS OR WRAPPED

USE CASE NAME	Charts or Wrapped
PARTICIPATING ACTORS	Amministratore del Sistema
ENTRY CONDITION	<ul style="list-style-type: none"><li>L'amministratore esegue <i>charts.py</i> o analogamente <i>wrapped.py</i>.</li></ul>
FLOW OF EVENTS	<ol style="list-style-type: none"><li>L'amministratore fornisce username e password del database locale.</li><li>Il programma esegue lo script SQL associato.</li></ol>
EXIT CONDITION	<ul style="list-style-type: none"><li>Lo script è stato eseguito con successo.</li></ul>
EXCEPTIONS	<ul style="list-style-type: none"><li>Se al punto 1 le credenziali fornite sono errate, l'amministratore viene notificato con un messaggio di errore e può ritentare l'autenticazione. (UCBC-1.1: CredenzialiErrate)</li></ul>

## 4. SUBSYSTEM SERVICES

Sottosistemi	Servizi	Operazioni
Gestione Utente	Registrazione Autenticazione Modifica Profilo Eliminazione Profilo Funzioni Social	Registrazione Autenticazione, Disconnessione Modifica Profilo Eliminazione Profilo Seguire/Smettere di seguire Artisti/Utenti Pubblici
Gestione Playlist	Creazione Playlist Gestione Brani Modifica Attributi Playlist Cambiamento Tipologia Eliminazione Playlist Feedback Cambio Ordine Brani Invito Utenti Collaborativa	Creazione Playlist Aggiunta Brano, Rimozione Brano Modifica Playlist Rendere Singola/Collaborativa, Privata/Pubblica Eliminazione Playlist Lasciare Like, Togliere Like Cambio Ordine Brani Invito Utenti Collaborativa
Gestione Album	Upload Album Eliminazione Album	Upload Album Eliminazione Album
Gestione Navigazione	Ricerca Contenuti Navigazione Pagine Visualizzazione Pagine	Ricerca Contenuti Navigazione Pagine Visualizza Pagina Utente/Artista/Album/Playlist
Gestione Riproduzione	Riproduzione Brano Riproduzione Generale Gestione Coda di Ascolti	Riproduzione Brano Riproduzione Playlist, Riproduzione Album Aggiunta Brano In coda, Saltare Brano Corrente