IS

USE CASE

| Campo | Descrizione |
|---|---|
| **Nome** | Nome descrittivo del caso d'uso (es: "Inserimento di un brano") |
| **Attori** | Chi partecipa? (es: Utente, Admin) |
| **Descrizione** | Cos'è lo scopo di questo caso d'uso? |
| **Pre-condizioni** | Cosa deve essere vero prima di iniziare? (es: utente loggato) |
| **Post-condizioni** | Cosa deve essere vero alla fine? |
| **Flusso base** | I passi principali per portare a termine il caso d'uso |
| **Alternative / Estensioni** | Cosa succede se qualcosa va storto? o ci sono varianti? |

---

use case:

ACCESS

| Campo | Descrizione |
|---|---|
| **Nome** | register |
| **Attori** | User opt sys 2fa |
| **Descrizione** | Accedere al servizio |
| **Pre-condizioni** | NULL |
| **Post-condizioni** | Utente registrato e connesso al servizio che ne salva i dati |
| **Flusso base** | Inserimento username && pswd && mail → verifica cod mail |
| **Alternative / Estensioni** | if → already user → login<br>If → pswd.wrong == 10 → block service for 30 min + alert root<br>if → forgot pswd \|\| username → recovery mail<br>if → existing username → repeat<br>if → invalid mail → notify user |

MODIFY

| Campo | Descrizione |
|---|---|
| **Nome** | Personalizzazione profilo |
| **Attori** | User sys |
| **Descrizione** | Aggiungere informazioni visibili agli altri user |
| **Pre-condizioni** | Utente deve essere registrato e loggato |
| **Post-condizioni** | La modifica deve venire salvata |
| **Flusso base** | Login → “modifica” → riempimento finestra → “save” |
| **Alternative / Estensioni** | If → !changes.saving → keeps old data && notify user |

LOAD

| Campo | Descrizione |
|---|---|
| **Nome** | Load shit |
| **Attori** | User sys |
| **Descrizione** | Carica brani, testi, spartiti |
| **Pre-condizioni** | Utente loggato && ha un file valido da caricare |
| **Post-condizioni** | A seconda del file si carica nel formato e nel posto giusto |
| **Flusso base** | Login → profile → load() → name (comment) → save |
| **Alternative / Estensioni** | Load → pubblic \|\| privite<br>if !save → notify user<br>else update profile add change |

VIEW

| Campo | Descrizione |
|---|---|
| **Nome** | view |
| **Attori** | User, sys, player |
| **Descrizione** | Permette di vedere un contenuto |
| **Pre-condizioni** | Utente loggato → utente nel feed \|\| nel profilo desiderato |
| **Post-condizioni** | La riproduzione del file dev'essere completata |
| **Flusso base** | Login → feed → select → load → view<br>login → search → select → load → view |
| **Alternative / Estensioni** | If !found → send report to user<br>if error playing → retry<br>if failed to play → notify user |

SEARCH

| Campo | Descrizione |
|---|---|
| **Nome** | Ricerca di profili \|\| tag \|\| file |
| **Attori** | User sys |
| **Descrizione** | Ricerca efficiente degli elementi (tipo ig o youtube) |
| **Pre-condizioni** | ACCESS deve essere vero |
| **Post-condizioni** | Result → shown on the GUI |
| **Flusso base** | ACCESS → "serach" → general \|\| filtered → output |
| **Alternative / Estensioni** | If !search → notify user<br>if input = empty → suggestions<br>if search → violate policy → notify root<br>if search → show results |

COMMENT

| Campo | Descrizione |
|---|---|
| **Nome** | Permette di commentare i file |
| **Attori** | User sys |
| **Descrizione** | Inserire commenti sotto al file o legati a sezioni specifiche |
| **Pre-condizioni** | VIEW deve essere in corso con tutti i suoi requisiti |
| **Post-condizioni** | Il commento deve essere visibile e salvato |
| **Flusso base** | Login → play → “comment” → general \|\| section → “save” |
| **Alternative / Estensioni** | If !comment → notify user<br>if comment = contro.politiche → remove && notify user and root<br>if comment → notify intrested people<br>if comment → tree && likes\dislikes && pinned |

sequence diagram
register user

```
@startuml
actor Utente
participant "Interfaccia Grafica (GUI)" as GUI
participant "Controller Registrazione" as Controller
participant "Servizio Email" as Email
participant "Database" as DB

Utente -> GUI : apre schermata registrazione
GUI -> Utente : mostra form registrazione
Utente -> GUI : inserisce username, password, email
GUI -> Controller : inviaDatiRegistrazione(dati)

alt Dati validi
   Controller -> DB : verificaDisponibilita(username, email)
   DB --> Controller : disponibili
   Controller -> Email : inviaCodiceVerifica(email)
   Email --> Controller : conferma invio
   Controller -> GUI : richiedi codice verifica
   Utente -> GUI : inserisce codice verifica
   GUI -> Controller : inviaCodice(codice)
   alt Codice corretto
      Controller -> DB : creaUtente(dati)
      DB --> Controller : utente creato
      Controller -> GUI : notifica registrazione avvenuta
      GUI -> Utente : accesso al sistema
   else Codice errato
      Controller -> GUI : errore codice
      GUI -> Utente : richiedi reinserimento o reinvio
   end
else Username/email non disponibili
   Controller -> GUI : errore disponibilità
   GUI -> Utente : chiedi nuovi dati
end
@enduml
```

modify
@startuml
actor Utente
participant "Interfaccia Grafica (GUI)" as GUI
participant "Controller Profilo" as Controller
participant "Database Utenti" as DB

Utente -> GUI : accede al profilo
Utente -> GUI : clicca su "Modifica profilo"
GUI -> Controller : richiediDatiProfilo()
Controller -> DB : getDatiProfilo(utenteID)
DB --> Controller : dati correnti
Controller -> GUI : mostraDati()

Utente -> GUI : modifica campi
Utente -> GUI : clicca "Salva"
GUI -> Controller : inviaDatiModificati()

alt Dati validi
    Controller -> DB : aggiornaProfilo(dati)
    DB --> Controller : conferma aggiornamento
    Controller -> GUI : mostra conferma salvataggio
    GUI -> Utente : profilo aggiornato
else Errore nei dati
    Controller -> GUI : mostra errore
    GUI -> Utente : mantieni dati precedenti
end
@enduml

load

```
@startuml
actor Utente
participant "Interfaccia Grafica (GUI)" as GUI
participant "Controller Upload" as Controller
participant "Servizio FileSystem / Media" as Media
participant "Database" as DB

== Inizio caricamento ==
Utente -> GUI : accede al profilo o a un brano
Utente -> GUI : clicca su "Carica contenuto"
GUI -> Controller : richiediDettagliUpload()
Controller -> GUI : mostra form upload

Utente -> GUI : seleziona file/link, aggiunge commento e visibilità
GUI -> Controller : inviaDatiUpload()

alt File valido
   Controller -> Media : salvaFile(file)
   Media --> Controller : restituisce path file
   Controller -> DB : salvaMetadati(brano, tipo, path/link, commento, visibilità)
   DB --> Controller : conferma salvataggio
   Controller -> GUI : upload riuscito
   GUI -> Utente : contenuto caricato con successo
else File invalido
   Controller -> GUI : errore (formato non supportato / file corrotto)
   GUI -> Utente : richiedi nuovo file
end
@enduml
```

view
@startuml
actor Utente
participant "Interfaccia Grafica (GUI)" as GUI
participant "Controller Contenuti" as Controller
participant "Database" as DB
participant "MediaPlayer / Browser" as Player

== Selezione e visualizzazione ==
Utente -> GUI : accede al feed / cerca brano
Utente -> GUI : seleziona contenuto
GUI -> Controller : richiediDettagliContenuto(id)

alt Contenuto disponibile e accessibile
    Controller -> DB : getContenuto(id)
    DB --> Controller : dati contenuto (tipo, path, visibilità)

    alt Contenuto interno
        Controller -> GUI : carica contenuto
        GUI -> Player : apri contenuto (pdf, mp3, mp4...)
        Player -> Utente : visualizzazione / ascolto
    else Contenuto esterno (es. YouTube)
        Controller -> GUI : apri link esterno
        GUI -> Player : embed o browser
        Player -> Utente : riproduzione via web
    end
else Contenuto non trovato o accesso negato
    Controller -> GUI : mostra errore
    GUI -> Utente : "Contenuto non disponibile"
end
@enduml

search

```
@startuml
actor Utente
participant "Interfaccia Grafica (GUI)" as GUI
participant "Controller Ricerca" as Controller
participant "Database" as DB

== Inizio ricerca ==
Utente -> GUI : digita query o seleziona filtri
GUI -> Controller : inviaRichiestaRicerca(query, filtri)

alt Input valido
   Controller -> DB : eseguiQueryRicerca(query, filtri)
   DB --> Controller : risultati
   Controller -> GUI : mostra risultati
   GUI -> Utente : visualizza risultati
else Input invalido o vuoto
   Controller -> GUI : mostra errore o suggerimenti
   GUI -> Utente : messaggio "Nessun risultato / input non valido"
end
@enduml
```

comment

```
@startuml
actor Utente
participant "Interfaccia Grafica (GUI)" as GUI
participant "Controller Upload" as Controller
participant "Servizio FileSystem" as FS
participant "Database" as DB

Utente -> GUI : click su "Carica file"
GUI -> GUI : mostra finestra upload
Utente -> GUI : inserisce titolo, file, commento, visibilità
GUI -> Controller : submitUpload(datiFile)

alt File valido
   Controller -> FS : salvaFile(datiFile)
   FS --> Controller : path file salvato
   Controller -> DB : salvaMetadati(titolo, autore, path, visibilità)
   DB --> Controller : conferma salvataggio
   Controller -> GUI : notifica "Caricamento completato"
   GUI -> Utente : mostra conferma
else File non valido
   Controller -> GUI : mostra errore
   GUI -> Utente : "Formato non supportato"
end
@enduml
```