

~ Myosotis

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?

@startuml



@enduml

~ Myosotis

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 if → already user → login If → pswd.wrong == 10 → block service for 30 min + alert root
Alternative / Estensioni	if → forgot pswd username → recovery mail if → existing username → repeat if → invalid mail → notify user

```
@startuml
```

```
== REGISTRAZIONE ==
```

```
actor Utente
```

```
boundary GUI_Login
```

```
control LoginController
```

```
database DB
```

```
entity Bot
```

```
Utente -> GUI_Login : Inserisce dati
GUI_Login -> LoginController : Invia dati registrazione
LoginController -> DB : Crea nuovo utente
DB --> LoginController : Conferma registrazione
LoginController --> GUI_Login : Mostra conferma
GUI_Login --> Utente : Registrazione completata
Utente -> GUI_Login : login
GUI_Login --> DB : confronta credenziali
DB --> GUI_Login : restituisce la risposta
alt login fallito
    GUI_Login -> Bot : show assistance
    Bot --> GUI_Login
end
GUI_Login --> Utente : login compiuto
```

```
@enduml
```

~ Myosotis

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

@startuml

== MODIFY ==

actor Utente

boundary GUI_change

control Moderating as Mod

control bot

database DB

Utente -> GUI_change : avvia richiesta

GUI_change -> bot : richiesta old data

bot -> DB : fetch

DB --> bot : return old data

bot --> GUI_change : show old data

GUI_change -> Mod : submitting changes

Mod -> bot : load data

bot -> DB : aggiorna informazioni

DB --> bot : conferma aggiornamento

bot --> GUI_change : aggiorna interfaccia

GUI_change --> Utente : mostra esito

@enduml

~ Myosotis

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 Load → public private
Alternative / Estensioni	if !save → notify user else update profile add change

```
@startuml
```

```
== LOAD DATA ==
```

```
actor Utente
boundary GUI_change
control Moderating as Mod
entity pack
control bot
database DB
```

```
Utente -> GUI_change : avvia modifica
GUI_change -> Mod : controllo rapido contenuti
Mod --> pack : carica dati
pack -> bot : invia dati modificati
bot -> DB : aggiorna informazioni
DB --> bot : conferma aggiornamento
bot -> GUI_change : aggiorna interfaccia
GUI_change -> Utente : mostra esito
```

```
@enduml
```

~ Myosotis

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 login → search → select → load → view If !found → send report to user
Alternative / Estensioni	if error playing → retry if failed to play → notify user

```
@startuml
```

```
== VIEW ==
```

```
actor Utente
```

```
boundary GUI_page
```

```
control bot
```

```
actor external_player as ep
```

```
Utente -> GUI_page : clicca sul contenuto
```

```
GUI_page -> bot : richiesta connessione
```

```
bot -> ep : contenuto
```

```
alt Request_error
```

```
    bot -> ep : ask error type
```

```
    ep --> bot : return error type
```

```
    bot --> GUI_page : error message
```

```
end
```

```
ep --> GUI_page : show content
```

```
@enduml
```

~ Myosotis

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 If !search → notify user
Alternative / Estensioni	if input = empty → suggestions if search → violate policy → notify root if search → show results

```
@startuml
```

```
== SERACH ==
```

```
actor Utente
```

```
boundary GUI_page
```

```
control bot
```

```
database DB
```

```
Utente -> GUI_page : clicca su barra di ricerca
```

```
GUI_page -> bot : richiesta confronto
```

```
bot -> DB : ricerca contenuto
```

```
alt 404
```

```
DB --> bot : return error type
```

```
bot --> GUI_page : return msg
```

```
else found
```

```
DB -> bot : contenuto
```

```
bot --> GUI_page : show content
```

```
end
```

```
@enduml
```

~ Myosotis

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” If !comment → notify user if comment = contro.politiche → remove && notify user and
Alternative / Estensioni	root if comment → notify intrested people if comment → tree && likes\dislikes && pinned

```
@startuml
```

```
== COMMENT ==
```

```
actor Utente
```

```
boundary GUI
```

```
control Moderating as Mod
```

```
control bot
```

```
database DB
```

```
Utente -> GUI : avvia richiesta
```

```
GUI -> Mod : richiesta autorizzazione
```

```
alt denied
```

```
Mod --> GUI : commento bloccato
```

```
else consentito
```

```
Mod --> GUI : consentito
```

```
GUI -> bot : richiesta commenti
```

```
bot -> DB : fetch
```

```
alt 404
```

```
DB --> bot : nessun commento
```

```
bot --> GUI : nessun commento
```

```
else commenti trovati
```

```
DB --> bot : struttura commenti
```

```
bot --> GUI : mostra struttura
```

```
end
```

```
end
```

```
GUI -> Mod : aggiungi commento
```

```
Mod -> bot : invia dati
```

```
bot -> DB : aggiorna
```

```
DB --> bot : conferma
```

```
bot --> GUI : aggiorna GUI
```

```
GUI --> Utente : mostra conferma
```

```
@enduml
```