

**Università degli Studi di Salerno**  
Corso di Ingegneria del Software

**SwaGGed**  
**Problem Statement**  
**Versione 1.0**



Data: 13/10/2024

<u>Progetto: SwaGGed</u>	Versione: 1.0
<u>Documento: Problem Statement</u>	Data: 09/10/2024

**Coordinatore del progetto**

Nome	Matricola

**Partecipanti:**

Nome	Matricola
Choaib Goumri	0512118390
Mattia Gallucci	0512116893

<b>Scritto da:</b>	Choaib Goumri
	Mattia Gallucci

**Revision History**

Data	Versione	Descrizione	Autore
13/10/2024	1.0	Problem Statement	Choaib Goumri, Gallucci Mattia

# Indice

1.	DOMINIO DEL PROBLEMA.....	4
1.1.	Condizione Attuale.....	4
1.2.	Soluzioni .....	5
2.	SCENARI .....	6
2.1	Visualizzazione sezione commenti.....	6
2.2	Ricerca di community.....	6
2.3	Creazione di contenuti.....	6
2.4	Visualizzazione post salvati.....	7
2.5	Moderazione dei contenuti.....	7
2.6	Inibire l'accesso agli utenti .....	7
3.	REQUISITI FUNZIONALI.....	8
3.1	Requisiti comuni (Ospite, Registrato, Moderatore) .....	8
3.2	Operazioni per l'Ospite .....	9
3.3	Operazioni per l'Utente registrato .....	9
3.4	Operazioni per il Moderatore.....	11
4.	REQUISITI NON FUNZIONALI.....	12
4.1	Usability .....	12
4.2	Reliability .....	12
4.3	Performance.....	13
4.4	Supportability.....	13
5	TARGET ENVIRONMENT .....	14
6	DELIVERABLE & DEADLINES .....	14

# 1. DOMINIO DEL PROBLEMA

## 1.1. Condizione Attuale

Il mondo della musica rap e hip-hop è cresciuto in maniera esponenziale negli ultimi decenni, con una presenza sempre più forte sui social media e piattaforme di streaming. Tuttavia, nonostante questa espansione globale, il **dominio di problema** che SwaGGed intende affrontare riguarda diverse aree critiche nel rapporto tra la comunità rap, gli artisti emergenti e i fan:

1. **Frammentazione della comunità:** Gli appassionati di rap e gli artisti si ritrovano su diverse piattaforme, come YouTube, Instagram o Twitter, ma manca un vero punto d'incontro dedicato esclusivamente alla cultura rap. Questa frammentazione impedisce una connessione fluida e mirata tra artisti e fan, che cercano un luogo unico dove potersi confrontare su tematiche specifiche legate al rap, senza distrazioni da altri generi o interessi.
2. **Difficoltà per gli artisti emergenti:** Per i nuovi artisti rap, entrare in un mercato musicale sempre più saturo è una sfida. Le piattaforme di streaming attuali, come Spotify o SoundCloud, offrono opportunità di pubblicazione, ma gli artisti emergenti hanno spesso difficoltà a farsi notare tra milioni di altri brani caricati ogni giorno. Non esistono piattaforme che combinino in modo specifico la **promozione musicale** con il **coinvolgimento attivo della comunità**, che possa amplificare la visibilità di nuovi talenti.
3. **Mancanza di interattività:** Molte piattaforme offrono solo consumo passivo di contenuti musicali, senza incentivare il coinvolgimento diretto del pubblico nella creazione, discussione e promozione di nuova musica. La scena rap, tuttavia, ha da sempre una forte dimensione partecipativa, con battaglie freestyle, remix, e collaborazioni. La mancanza di uno spazio che faciliti questa interattività limita le possibilità per gli utenti di connettersi più a fondo con la cultura rap.
4. **Assenza di un ambiente social dedicato al rap:** Le principali piattaforme di social media non forniscono strumenti specifici per sostenere e promuovere la crescita della cultura rap. Mancano strumenti di curatela musicale, gruppi tematici specifici, o una segmentazione precisa di contenuti per artisti e fan del rap.

## 1.2. Soluzioni

SwaGGed è pensato per affrontare direttamente questi problemi, offrendo soluzioni innovative che rispondano alle esigenze della comunità rap, facilitando la crescita e l'interazione all'interno di un ambiente dedicato:

- 1. Piattaforma unica per la comunità rap:** SwaGGed fornisce uno spazio dedicato esclusivamente agli appassionati di rap, eliminando la dispersione tra piattaforme generaliste. In questo modo, gli utenti possono incontrarsi, discutere e interagire in un ambiente completamente orientato alla cultura hip-hop, creando una comunità forte e coesa. Articoli, forum tematici e gruppi social vengono curati per garantire che il focus rimanga esclusivamente su artisti e temi legati al rap.
- 2. Promozione e visibilità per artisti emergenti:** SwaGGed offre agli artisti emergenti uno spazio per caricare la propria musica e promuoverla direttamente alla comunità rap, garantendo un sistema di feedback immediato. Attraverso funzionalità di condivisione e ascolto collettivo, gli artisti possono farsi conoscere e costruire una base di fan che cresce organicamente. Inoltre, la piattaforma facilita la collaborazione tra artisti emergenti e noti, creando opportunità di crescita attraverso sinergie e partnership.
- 3. Interazione e creazione di contenuti:** A differenza delle piattaforme di streaming tradizionali, SwaGGed incoraggia la partecipazione attiva degli utenti. I fan possono non solo ascoltare musica, ma anche interagire direttamente con gli artisti, commentare nuovi brani, e partecipare a discussioni su argomenti specifici come testi, produzione musicale, battaglie di freestyle e remix. Questo rafforza l'engagement della comunità e crea un senso di appartenenza unico.
- 4. Curatela e personalizzazione:** SwaGGed offre strumenti per curare contenuti personalizzati. Gli utenti possono creare playlist condivisibili, seguire artisti e fan con interessi simili e partecipare a gruppi di discussione focalizzati su temi come "rap underground", "trap", o "freestyle battle". Questa curatela facilita un'esperienza immersiva e personalizzata, garantendo agli utenti di essere sempre aggiornati sui loro artisti e tendenze preferite.
- 5. Ecosistema rap completo:** Con la sua vasta gamma di funzionalità, SwaGGed si pone come una soluzione completa per tutto ciò che riguarda la cultura rap. Offre non solo contenuti informativi e social, ma diventa anche uno strumento di networking per artisti, fan e professionisti del settore. La piattaforma mira a diventare il fulcro di nuove collaborazioni, scoperte di talenti e crescita creativa, fornendo uno spazio che unisce sia il mondo musicale che quello sociale.

## 2. SCENARI

### 2.1 *Visualizzazione sezione commenti*

Marco, un utente non registrato, eccede alla piattaforma del social media come ospite. Nella homepage, nota un post intitolato “I 10 migliori album rap del 2023” e, incuriosito, decide di approfondire l’argomento. Clicca sul post e il sito lo reindirizza su una pagina contenente solo quel contenuto.

Marco può scorrere la nuova pagina e vedere una sezione di commenti dove gli utenti discutono le loro opinioni sugli album elencati. Purtroppo, essendo un utente non registrato, non può partecipare alla discussione.

### 2.2 *Ricerca di community*

Laura accede al sito come ospite, desiderosa di esplorare la community dedicata al genere musicale “trap”. Nella parte superiore della homepage, nota una barra di ricerca ben visibile. Con un semplice clic, inizia a digitare la parola “trap” e preme il tasto Invio.

Dopo un breve caricamento, viene reindirizzata a una pagina dei risultati di ricerca. Qui trova diverse categorie musicali, organizzate per facilitare la navigazione: una sezione con i post contenenti la parola “trap”, una dedicata alle communities e un’altra che elenca gli utenti con “trap” nel nome.

Curiosa di scoprire più informazioni, Laura decide di cliccare sulla scheda “Community”. La pagina si aggiorna, mostrando un elenco di communities con “trap” nel titolo.

Una volta identificata la community che risuona di più con i suoi interessi, Laura clicca sul titolo. La nuova pagina si carica rapidamente. Ora ha finalmente l’opportunità di navigare tra i contenuti della comunità trap, esplorando nuovi artisti, condividendo opinioni e scoprendo le ultime tendenze del genere.

### 2.3 *Creazione di contenuti*

Luigi, un utente registrato al social network, decide di pubblicare un nuovo post nella community “Ye”. Si autentica quindi al sito e dalla homepage, nota un pulsante in alto con l’etichetta “Crea”. Cliccandolo, viene reindirizzato a una pagina di creazione contenuti.

Qui seleziona la comunità “Ye” dal menu a discesa. Inserisce il titolo “Miglior artista della storia dell’umanità” e nel corpo del post scrive: “Ye supera di gran lunga qualsiasi altro artista mai esistito, esistente e che mai esisterà.” Fiero del suo contributo, clicca sul pulsante “Pubblica” situato sotto l’editor del post, rendendo così il suo contenuto visibile alla comunità “Ye”.

## **2.4 Visualizzazione post salvati**

Andrea, un utente registrato, si autentica alla piattaforma. Mentre scorre la homepage, si imbatte in un post interessante intitolato “La drill è nata a Chicago?”. Inizia a leggere, ma non avendo abbastanza tempo per finirlo, decide di salvarlo per più tardi.

Sotto al post, nota l'icona del segnalibro e la clicca, salvando il contenuto. Quando ha più tempo, Andrea torna sulla piattaforma e, dalla homepage, clicca sull'icona del suo profilo in alto a destra. Viene reindirizzato a una pagina con diverse sezioni, tra cui “Post”, “Commenti” e “Salvati”.

Seleziona la scheda “Salvati”, dove trova il post che aveva messo da parte. Clicca su di esso e riprende la lettura esattamente da dove l'aveva interrotta.

## **2.5 Moderazione dei contenuti**

Mattia, un amministratore nella moderazione di contenuti, si autentica con le sue credenziali e atterra sulla homepage. Mentre scorre il feed, nota un post il cui titolo contiene una parola offensiva. Essendo responsabile della moderazione, decide di intervenire immediatamente.

In alto a destra del post, trova un pulsante rosso con l'etichetta “Elimina”. Cliccandolo, compare un pop-up di conferma. Dopo aver letto attentamente, Mattia preme “Conferma” e il post, insieme ai relativi commenti, viene rimosso.

La homepage si aggiorna automaticamente, e il contenuto offensivo non è più visibile agli utenti, garantendo un ambiente rispettoso sulla piattaforma.

## **2.6 Inibire l'accesso agli utenti**

Choib, un amministratore del sistema, accede alla piattaforma e si ritrova sulla homepage. Subito nota un post pubblicato da un utente chiamato “ChristianHater”. Accanto al nome, osserva che il numero di segnalazioni ricevute è arrivato a 10. Dato questo alto numero di segnalazioni, Choib sa che al prossimo errore l'utente verrà sospeso dalla piattaforma. Poiché il post contiene contenuti di odio, Choib decide di agire.

Clicca sul nome dell'utente, che lo reindirizza a una nuova pagina contenente il profilo di “ChristianHater”. Qui, accanto al nome dell'utente, trova un pulsante “BAN”. Senza esitazioni, clicca su questo pulsante. Compare un pop-up di conferma e, dopo aver confermato l'azione, viene reindirizzato nuovamente alla homepage. Ora “ChristianHater” è bandito dalla piattaforma, contribuendo così a mantenere un ambiente più sicuro e rispettoso per tutti gli utenti.

### 3. REQUISITI FUNZIONALI

#### 3.1 *Requisiti comuni (Ospite, Registrato, Moderatore)*

- **RF1 – Visualizzare home page:**
  - Permettere all'utente di visualizzare la pagina principale del sistema.
  - **Priorità:** alta.
- **RF2 – Ricercare community:**
  - Permettere la ricerca di community presenti nel sistema.
  - **Priorità:** alta.
- **RF3 – Ricercare post:**
  - Consentire la ricerca di post specifici nel sistema.
  - **Priorità:** alta.
- **RF4 – Ricercare utenti:**
  - Consentire la ricerca di utenti registrati nel sistema.
  - **Priorità:** alta.
- **RF5 – Visualizzare i commenti:**
  - Mostrare i commenti associati ai post.
  - **Priorità:** alta.
- **RF6 – Visualizzare i like:**
  - Permettere la visualizzazione del numero di like di un post o commento.
  - **Priorità:** media.



### 3.2 Operazioni per l'Ospite

- **RF7 – Registrarsi:**
  - Permettere all'utente ospite di creare un account nel sistema.
  - **Priorità:** alta.
- **RF8 – Autenticarsi:**
  - Consentire all'utente ospite di accedere al sistema inserendo le credenziali.
  - **Priorità:** alta.

### 3.3 Operazioni per l'Utente registrato

- **RF9 – Apprezzare i contenuti:**
  - Consentire di mettere like ai post e ai commenti.
  - **Priorità:** alta.
- **RF10 – Commentare i contenuti:**
  - Consentire di aggiungere commenti ai post.
  - **Priorità:** alta.
- **RF11 – Segnalare post, community, commenti, utenti:**
  - Consentire la segnalazione di contenuti o utenti problematici.
  - **Priorità:** alta.
- **RF12 – Seguire utenti:**
  - Permettere di seguire altri utenti per vedere i loro aggiornamenti.
  - **Priorità:** media.

- **RF13 – Smettere di seguire gli utenti:**
  - Permettere di non seguire più gli utenti precedentemente seguiti.
  - **Priorità:** media.
- **RF13 – Iscrivere alle community:**
  - Permettere l'iscrizione a una o più community.
  - **Priorità:** alta.
- **RF14 – Salvare i post:**
  - Consentire di salvare post per una consultazione successiva.
  - **Priorità:** alta.
- **RF15 – Visualizzare i post salvati:**
  - Mostrare all'utente i post che ha salvato.
  - **Priorità:** alta.
- **RF16 – Visualizzare i commenti e post creati:**
  - Mostrare all'utente i commenti e i post che ha creato.
  - **Priorità:** alta.
- **RF17 – Visualizzare il proprio profilo:**
  - Consentire la visualizzazione del profilo personale.
  - **Priorità:** alta.
- **RF19 – Effettuare il log out:**
  - Consentire all'utente di disconnettersi dal sistema.
  - **Priorità:** alta.
- **RF20 – Modificare le proprie credenziali:**
  - Permettere all'utente di aggiornare le informazioni di accesso e profilo.
  - **Priorità:** media.

- **RF21 – Creare community:**
  - Consentire di creare una nuova community.
  - **Priorità:** alta.
- **RF22 – Modificare il nome della community:**
  - Permettere di cambiare il nome di una community creata.
  - **Priorità:** media.
- **RF23 – Eliminare la propria community:**
  - Consentire la rimozione di una community creata dall'utente.
  - **Priorità:** alta.

### ***3.4 Operazioni per il Moderatore***

- **RF24 – Inibire utenti:**
  - Consentire di bloccare o sospendere utenti dal sistema.
  - **Priorità:** alta.
- **RF25 – Eliminare post:**
  - Consentire di cancellare post inappropriati.
  - **Priorità:** alta.
- **RF26 – Eliminare community:**
  - Consentire di eliminare community che violano le regole.
  - **Priorità:** alta.
- **RF27 – Eliminare commenti:**
  - Consentire di rimuovere commenti non conformi alle linee guida.
  - **Priorità:** alta.

## 4. REQUISITI NON FUNZIONALI

### 4.1 *Usability*

#### RNFU\_1:

- Il sistema deve fornire un'interfaccia grafica.
- **Priorità:** alta.

#### RNFU\_2:

- Il sistema deve fornire una navbar con componenti grafiche che permettano di raggiungere le varie sezioni principali del sito in base a ogni ruolo supportato dal sistema. Verrà posta in alto nella pagina e sarà implementato un oggetto apposito che, in base al ruolo corrente dell'utente che sta navigando, mostrerà diverse view.
- **Priorità:** alta.

#### RNFU\_3:

- Il sito deve essere responsive in modo che possa adattarsi ai seguenti dispositivi: PC, smartphone e tablet. Questo requisito verrà implementato con CSS e, se necessario, con l'ausilio di qualche libreria come Bootstrap.
- **Priorità:** media.

### 4.2 *Reliability*

#### RNFR\_1:

- In presenza di input non validi inseriti dall'utente, il sistema deve rispondere con appropriati messaggi di errore. Questi verranno mostrati tramite degli avvisi sulla pagina contenente l'errore e tramite l'evidenziazione dei campi che risultano errati.
- **Priorità:** alta.

#### RNFR\_2:

- Il sistema deve trattare i dati in modo sicuro, garantendo che siano inaccessibili a malintenzionati. Le password degli utenti devono essere criptate nel database. Verrà utilizzato il sistema di crittografia in base 64 per salvare la versione crittografata e non reversibile delle password originarie.
- **Priorità:** alta.

### *4.3 Performance*

#### RNFP\_1:

- Il sistema deve essere in grado di gestire più utenti contemporaneamente.
- **Priorità:** alta.

### *4.4 Supportability*

#### RNFS\_1:

- Il sistema software dovrà avere un'architettura three-tier con un client che implementa il livello di presentazione, un server che implementa la logica applicativa e un DBMS per la gestione dei dati. In questo modo, il modello e la sua visualizzazione saranno nettamente separati, favorendo le operazioni di manutenzione.
- **Priorità:** alta.

## 5 TARGET ENVIRONMENT

Tecnologie utilizzate per lo sviluppo del sistema

- Apache Tomcat 10
- Java Server Pages, Java Servlets
- JavaScript, AJAX
- MySQL, MySQL Workbench, JDBC

## 6 DELIVERABLE & DEADLINES

- 14 Ottobre 2024: Rilascio Problem Statement
- 28 Ottobre 2024: Requisiti e casi d'uso
- 11 Novembre 2024: Requirements Analysis Document
- 25 Novembre 2024: System Design Document
- 8 Dicembre 2024: Prima versione del software
- 16 Dicembre 2024: Piano di test e specifica interfacce dei moduli del sistema