Università di Verona

Corso di Laurea in Informatica

Documentazione Progetto

Ingegneria del Software 2025

MusicHub

Gestione Brani Musicali

Studenti:
Bortolaso Mattia - VR500026
Cheng Jiashuo - VR501311
Colombo Matteo - VR500130

Docente:
Carlo Combi

Indice

1 Requisiti ed interazioni utente-sistema

Il sistema è un *catalogo di brani e concerti modificabile* che consente a utenti registrati di inserire, consultare, commentare e gestire:

- Spartiti,
- Testi,
- Audio (formato MP3),
- Video (formato MP4 o link da YouTube),
- Altri materiali digitali relativi a brani musicali.

Il sistema permette inoltre di gestire concerti presenti su YouTube, indicando quali brani sono contenuti al loro interno.

Ogni utente può:

- Caricare nuovi brani e concerti,
- Contribuire con contenuti,
- Lasciare commenti,
- Segmentare video di concerti,
- Inserire annotazioni e note sui file multimediali caricati.

L'amministratore del sistema è responsabile della **gestione degli utenti** e della **moderazione dei contenuti inappropriati**.

1.1 Diagrammi di attività

Nota: I diagrammi che seguono rappresentano singole interazioni tra l'utente e il sistema, modellando un'unica esecuzione dell'attività illustrata. Per ragioni di chiarezza e semplificazione, non è stata esplicitata la possibilità di ripetere l'operazione più volte all'interno della stessa sessione. Tale comportamento ciclico è comunque previsto funzionalmente dal sistema, ma è stato omesso nella rappresentazione UML per evitare un'eccessiva complessità visiva. Si assume che l'operazione possa essere rieseguita solo dopo un eventuale riavvio dell'applicazione, dove applicabile.

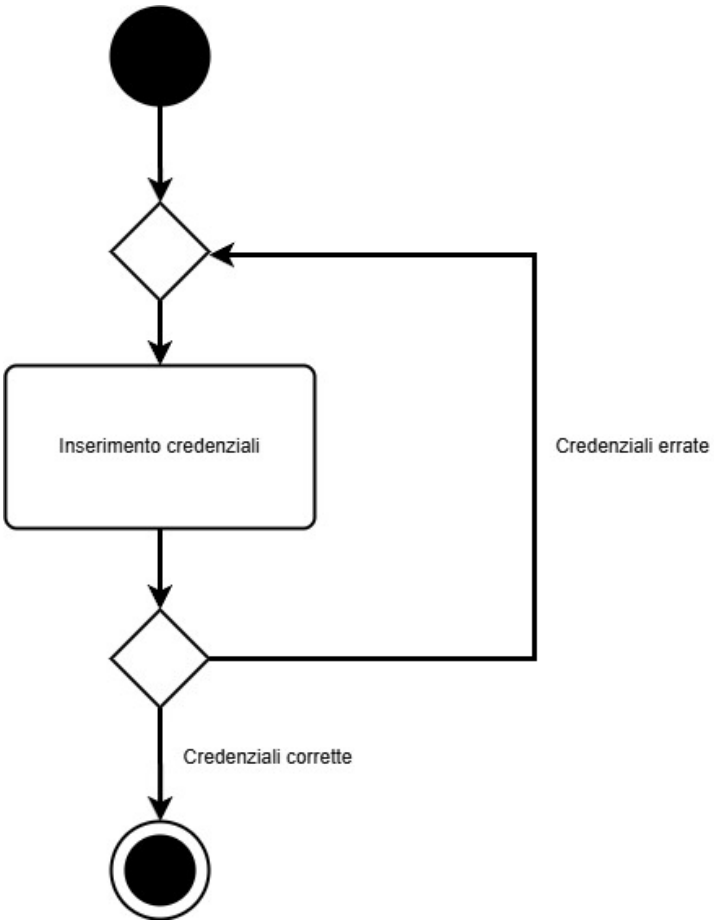

Figura 1: Diagramma di attività del login

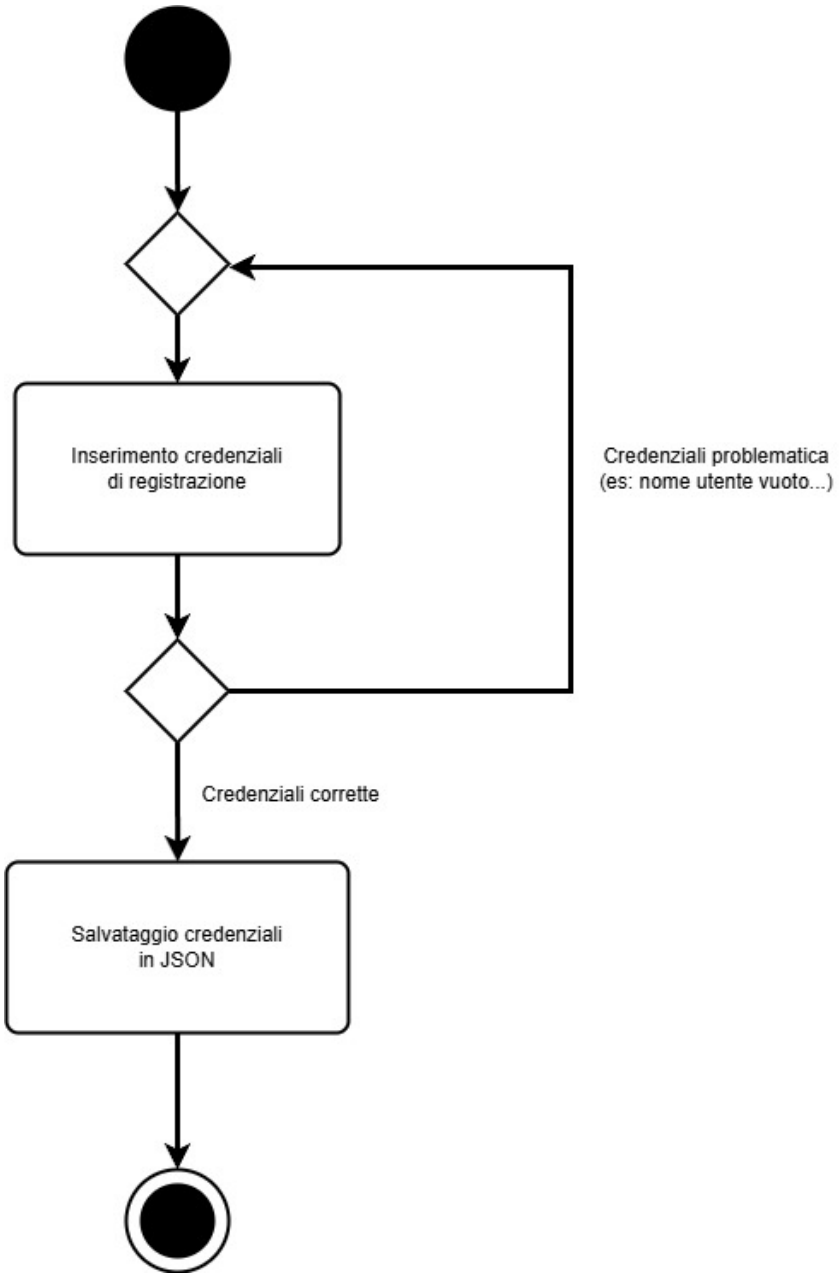

Figura 2: Diagramma di attività della registrazione

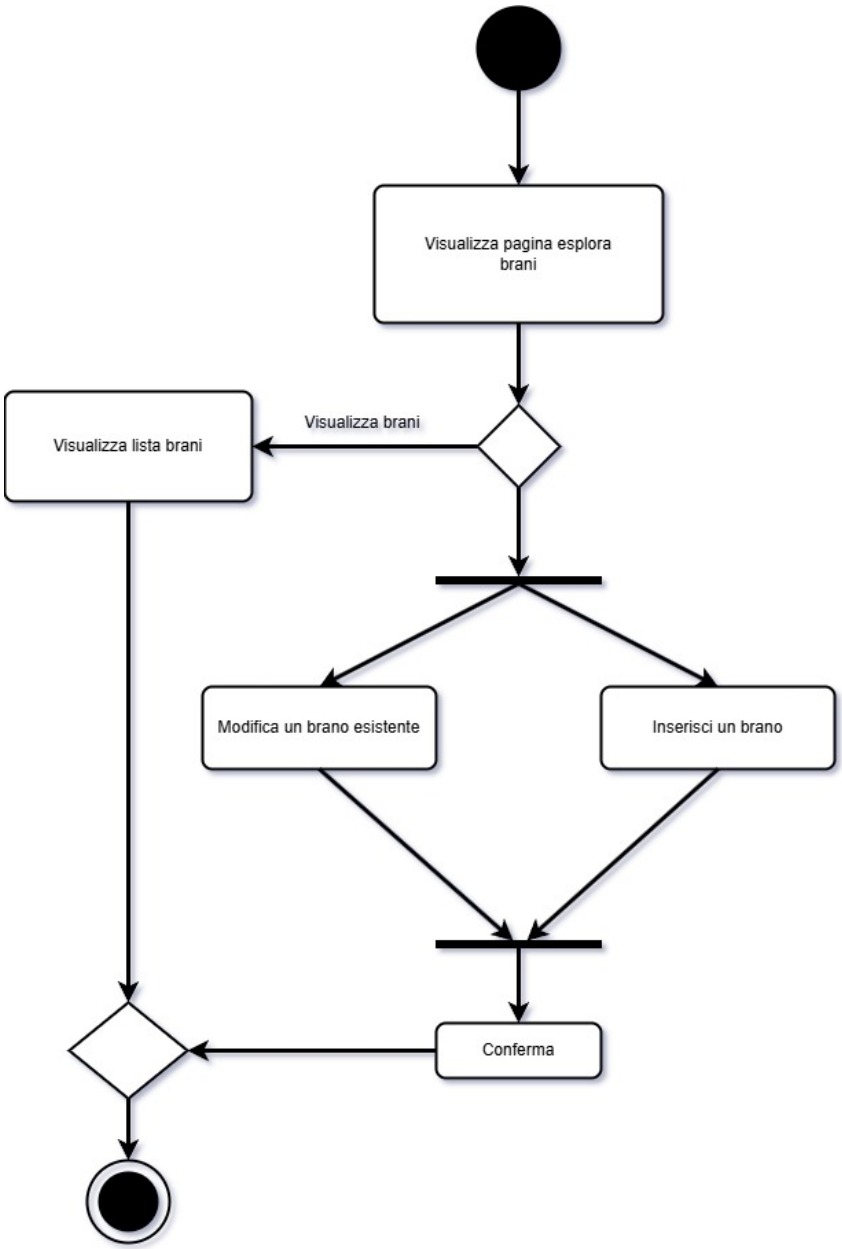

Figura 3: Diagramma di attività del brano

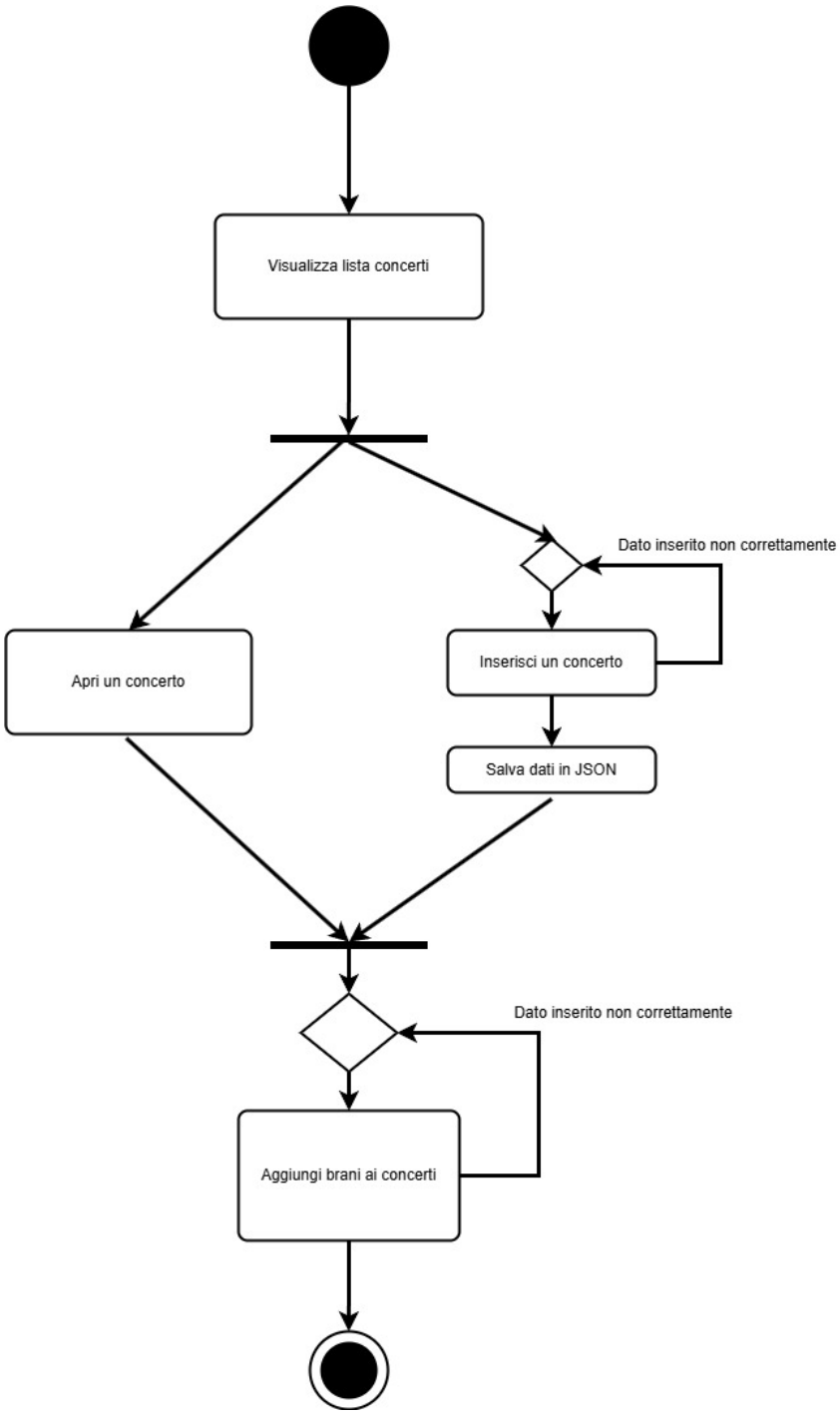

Figura 4: Diagramma di attività del concerto

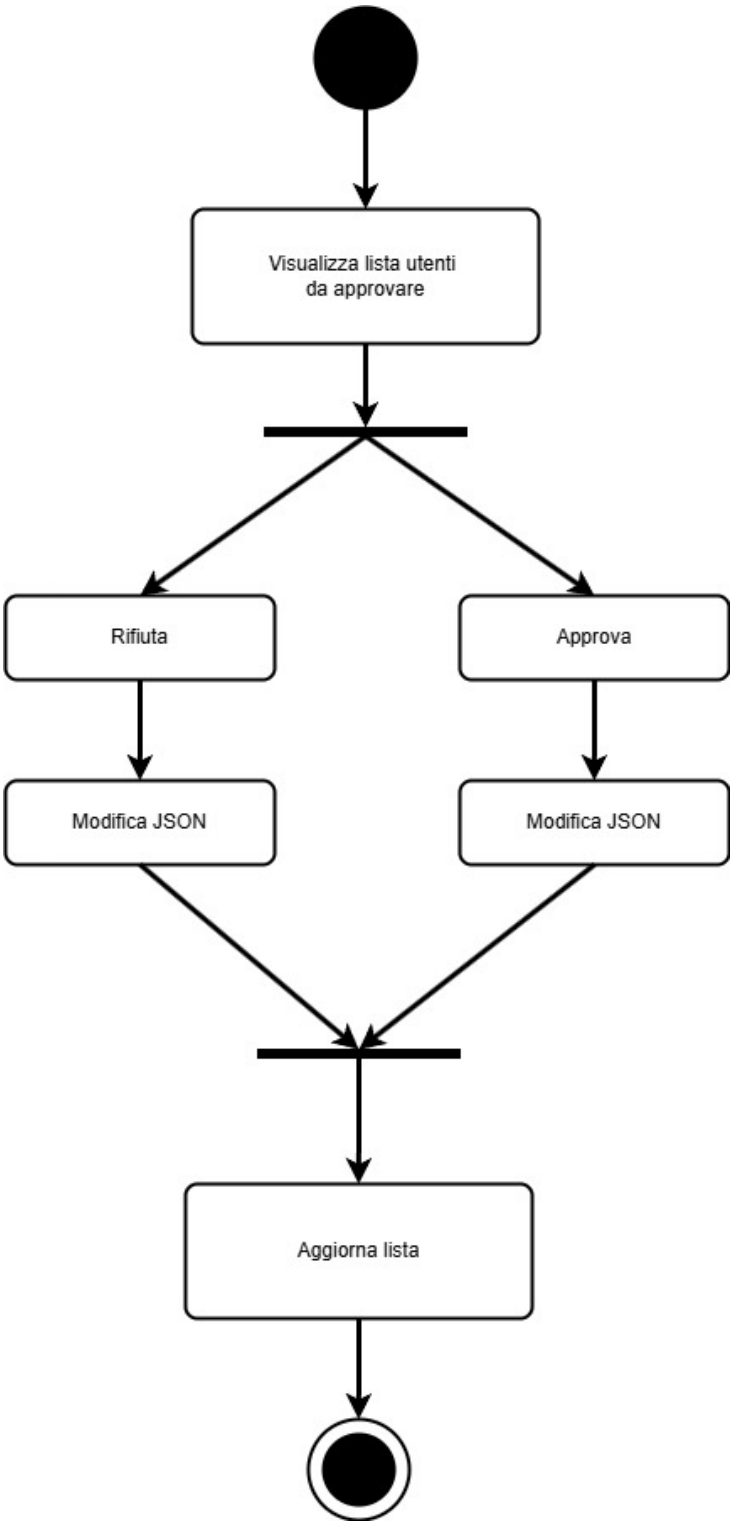

Figura 5: Diagramma di attività dell'admin

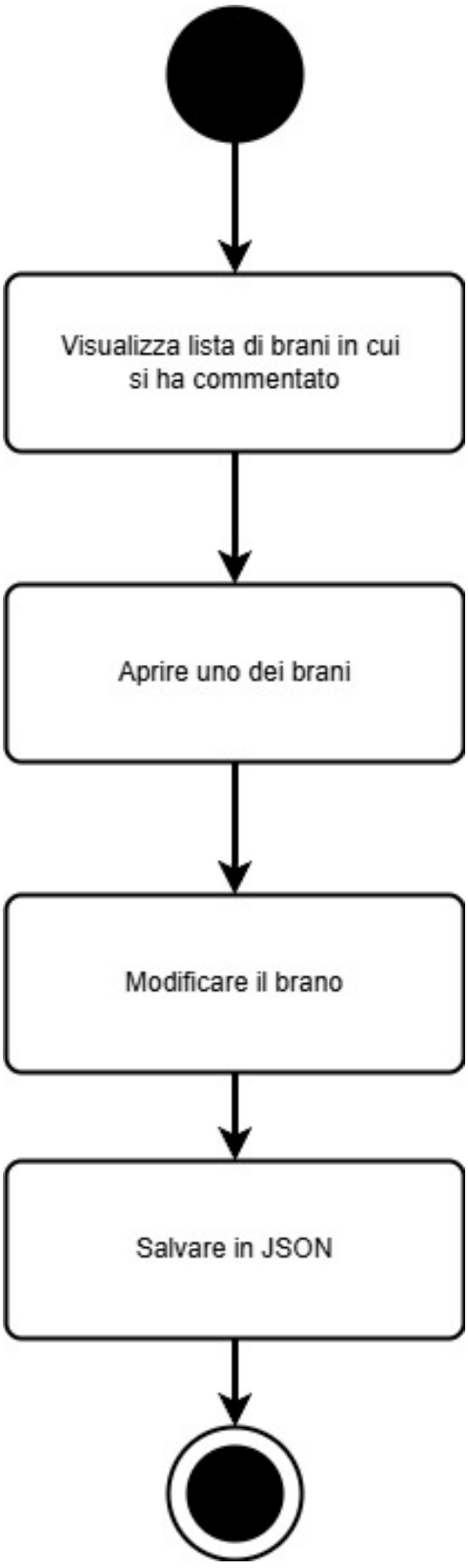

Figura 6: Diagramma di attività della cronologia

2 Specifiche e casi d'uso

Note generali: Il sistema proposto supporta l'utilizzo da parte di *utenti registrati* e *amministratori*. L'amministratore è un utente normale a tutti gli effetti ma con l'autorizzazione ad approvare o eliminare richieste di registrazione al sistema da parte di nuovi utenti e con la possibilità di cancellare qualsiasi commento lasciato dagli utenti. L'amministratore, quindi può effettuare tutte le operazioni che un normale utente può fare nel sistema

2.1 UC1 - Registrazione

Attori: *Utente*

Precondizioni: *Utente non registrato*

Passi:

1. *L'utente compila il form e registra l'account*
2. *L'amministratore approva o rifiuta l'utente*

Postcondizioni: *Utente attivo o rifiutato*

2.2 UC2 - Login

Attori: *Utente, Admin*
Precondizioni: *Utente registrato e approvato*
Passi:

1. *L'utente compila il form ed effettua il login*

Postcondizioni: *Sessione utente avviata*

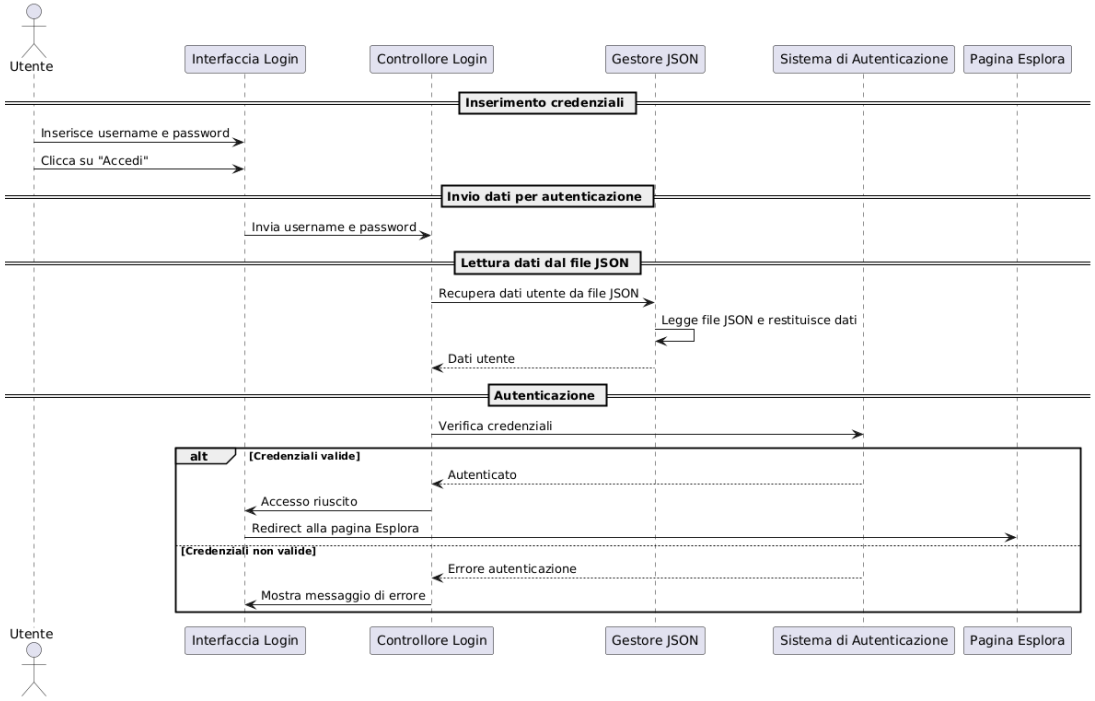

Figura 7: Login

Dopo l'avvenuta corretta autenticazione l'utente verrà reindirizzato alla pagina di Esplora brani

2.3 UC3 - Inserimento brano

Gli utenti possono inserire nuovi brani, per questo è disponibile l'interfaccia interattiva dedicata con autocompletamento grazie all'utilizzo di API Spotify ed Itunes per il recupero di metadati del brano inserito.

Attori: *Utente, Admin*

Precondizioni: *Utente loggato*

Passi:

1. *L'utente accede all'interfaccia di carica brano tramite pulsante nella topbar*
2. *L'utente inserisce Titolo e Autore nel form*
3. *L'utente attende l'autocompletamento di Titolo, Autore, Anno di pubblicazione, Genere e la visualizzazione della copertina dell'album*
4. *L'utente può decidere se compilare i campi Esecutori, Strumenti Musicali, Link YouTube*
5. *L'utente può decidere se caricare file allegati*
6. *Per confermare il caricamento del brano l'utente clicca sul pulsante "Carica"*

Postcondizioni: *Brano presente in sistema*

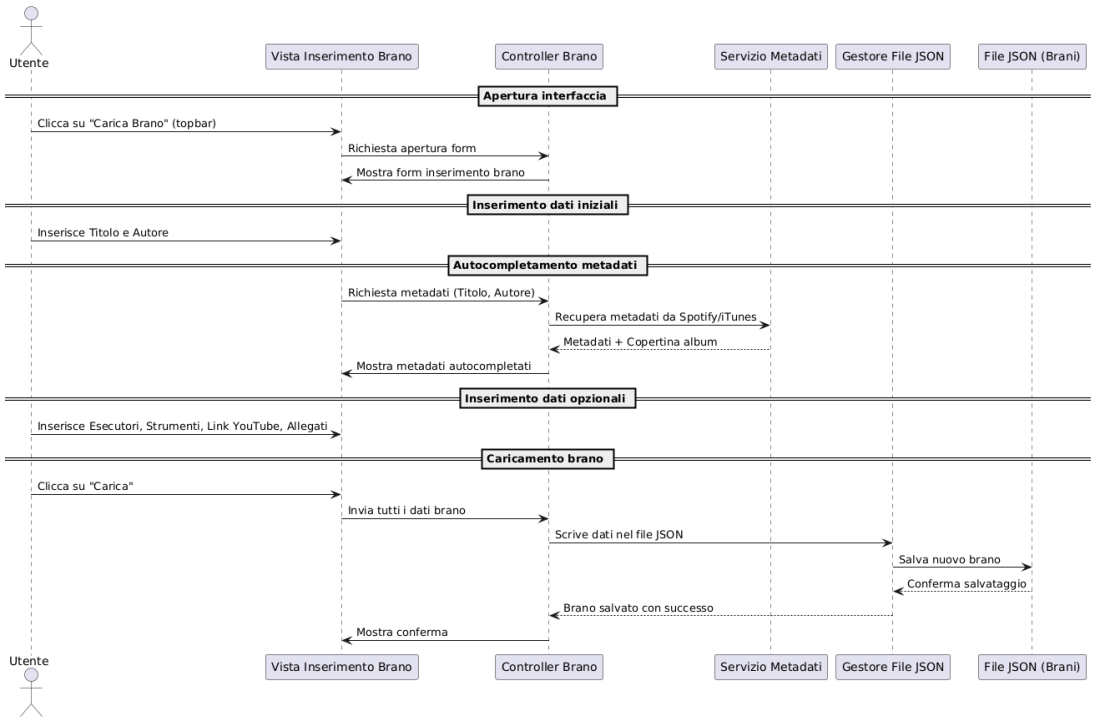

Figura 8: Diagramma di attività della cronologia

2.4 UC4 - Inserimento concerto

Procedura simile all'inserimento di un brano, ma viene chiesto di inserire unicamente il link del concerto YouTube

Attori: *Utente, Admin*

Precondizioni: *Utente loggato*

Passi:

1. *L'utente accede all'interfaccia di carica concerto tramite pulsante nella topbar*
2. *L'utente inserisce il Link YouTube del concerto*
3. *Per confermare il caricamento del concerto l'utente clicca sul pulsante "Carica"*

Postcondizioni: *Concerto presente nel sistema*

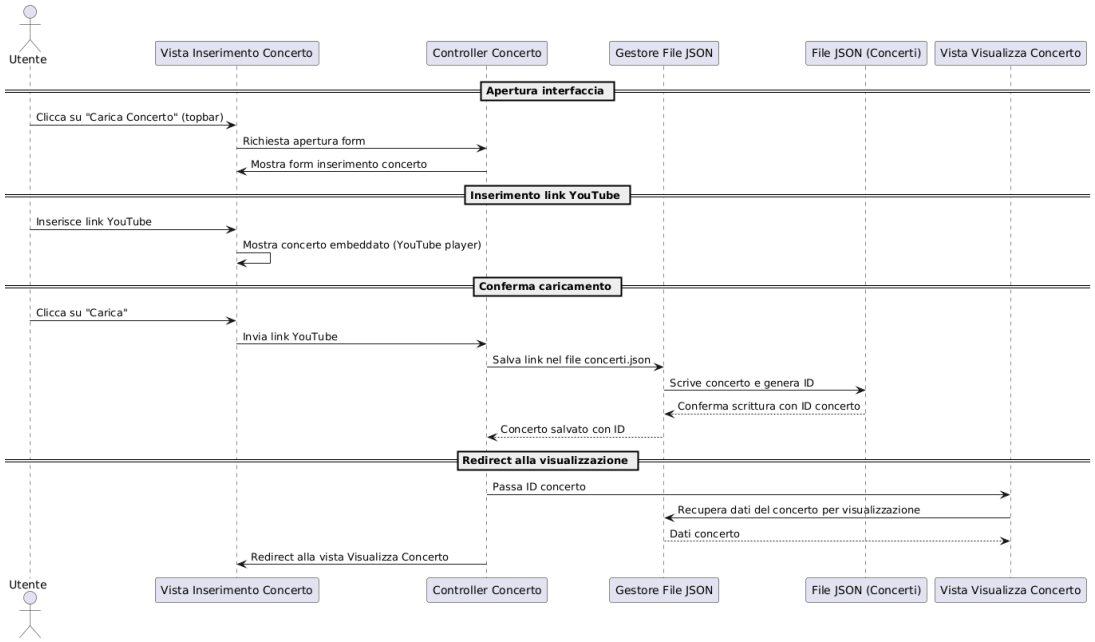

Figura 9: Diagramma di attività della cronologia

2.5 UC5 - Esplorazione contenuti

Per esplorare i brani o concerti contenuti nel sistema, è presente una interfaccia ad hoc con la possibilità di filtrare i brani per Autori, Generi ed Esecutori.

Attori: *Utente, Admin*

Descrizione: *Esplora i brani / concerti nel sistema potendo filtrare i brani per autore, genere ed esecutori*

Precondizioni: *Utente loggato*

Passi:

1. *L'utente accede alla pagina Esplora brani o concerti*
2. *L'utente può filtrare i brani in base ad Autore, Genere, Esecutore*

Postcondizioni: *Utente esplora contenuti*

2.6 UC6 - Caricare File

Attori: *Utente, Admin*

Precondizioni: *Brano già presente nel sistema*

Passi:

1. *L'utente clicca sul tasto per caricare un allegato (solo documenti e file multimediali)*
2. *L'utente seleziona i file che vuole caricare*
3. *L'utente carica i file*

Postcondizioni: *File associato al brano e visibile*

2.7 UC7 - Aggiungere commento o nota

Attori: *Utente, Admin*

Precondizioni: *Brano già presente nel sistema*

Passi:

1. *L'utente accede alla pagina di visualizzazione del brano dalla pagina esplora*
2. *L'utente scrive nella casella di testo del commento/nota*
3. *L'utente clicca invia per caricare il commento*

Postcondizioni: *Commento/nota caricato nel sistema e associato al brano*

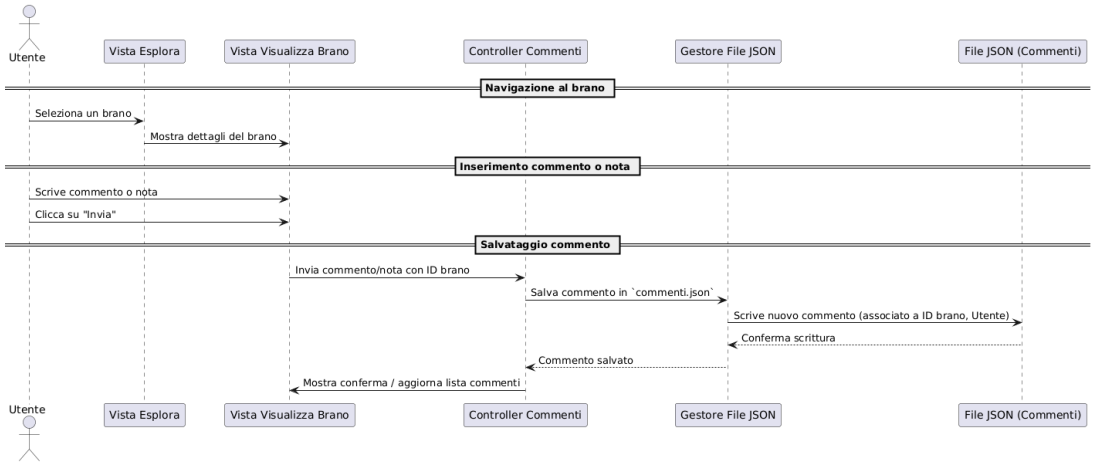

Figura 10: Diagramma di attività della cronologia

2.8 UC8 - Rispondere a commenti

Attori: *Utente, Admin*
Precondizioni: *Commento a cui rispondere già presente nel sistema*
Passi:

1. *L'utente si trova nella pagina di visualizzazione di un brano*
2. *L'utente clicca il tasto rispondi di un commento*
3. *L'utente inserisce la risposta nella casella di testo*
4. *L'utente clicca il tasto invia per caricare la risposta nel sistema*

Postcondizioni: *Risposta al commento caricata nel sistema e associata al brano*

2.9 UC9 - Cancella commenti

Attori: *Admin / Utente (autore contenuto) / Utente (autore proprio commento)*
Precondizioni: *Commento da eliminare presente nel sistema*
Passi:

1. *L'utente, se abilitato, visualizza nel commento il tasto di cancellazione*
2. *L'utente preme il tasto di cancellazione*

Postcondizioni: *Commento cancellato dal sistema*

2.10 UC10 - Ricerca

Attori: *Utente, Admin*
Precondizioni: *L'utente si trova nella pagina esplora*
Passi:

1. *L'utente digita l'elemento da cercare nella barra di ricerca*
2. *L'utente clicca la lente di ingrandimento per avviare la ricerca*

Postcondizioni: *Restituita la lista di elementi che corrisponde alla ricerca*

2.11 UC11 - Visualizza brano

Attori: *Utente, Admin*

Precondizioni: *Brano esistente nel sistema*

Passi:

1. *L'utente seleziona un brano da visualizzare dalla pagina Esplora*
2. *Viene visualizzata l'interfaccia di visualizzazione del brano*

Postcondizioni: *L'utente può navigare nell'interfaccia per effettuare modifiche al brano, visualizzare allegati e commentare*

2.12 UC12 - Visualizza concerto

Attori: *Utente, Admin*

Precondizioni: *Concerto esistente nel sistema*

Passi:

1. *L'utente seleziona un concerto da visualizzare dalla pagina Concerti*
2. *Viene visualizzata l'interfaccia di visualizzazione del concerto*

Postcondizioni: *L'utente può navigare nell'interfaccia per effettuare modifiche al concerto, come segmentarlo*

2.13 UC13 - Segmenta concerto

Attori: *Utente, Admin*

Precondizioni: *Concerto esistente nel sistema*

Passi:

1. *L'utente seleziona un brano da aggiungere al concerto dall'elenco a tendina*
2. *L'utente inserisce il timestamp del brano da aggiungere*
3. *L'utente clicca sul pulsante "Carica" per associare il brano selezionato al concerto nei timestamp inseriti*

Postcondizioni: *Brano associato al concerto*

2.14 UC14 - Amministrazione utenti

Attori: *Admin*

Precondizioni: *Admin deve essere autenticato*

Passi:

1. *Admin accede alla pagina di amministrazione*
2. *Admin visualizza gli utenti non ancora approvati con tasto di approvazione o cancellazione*

Postcondizioni: *Utente registrato è approvato o rimosso dal sistema*

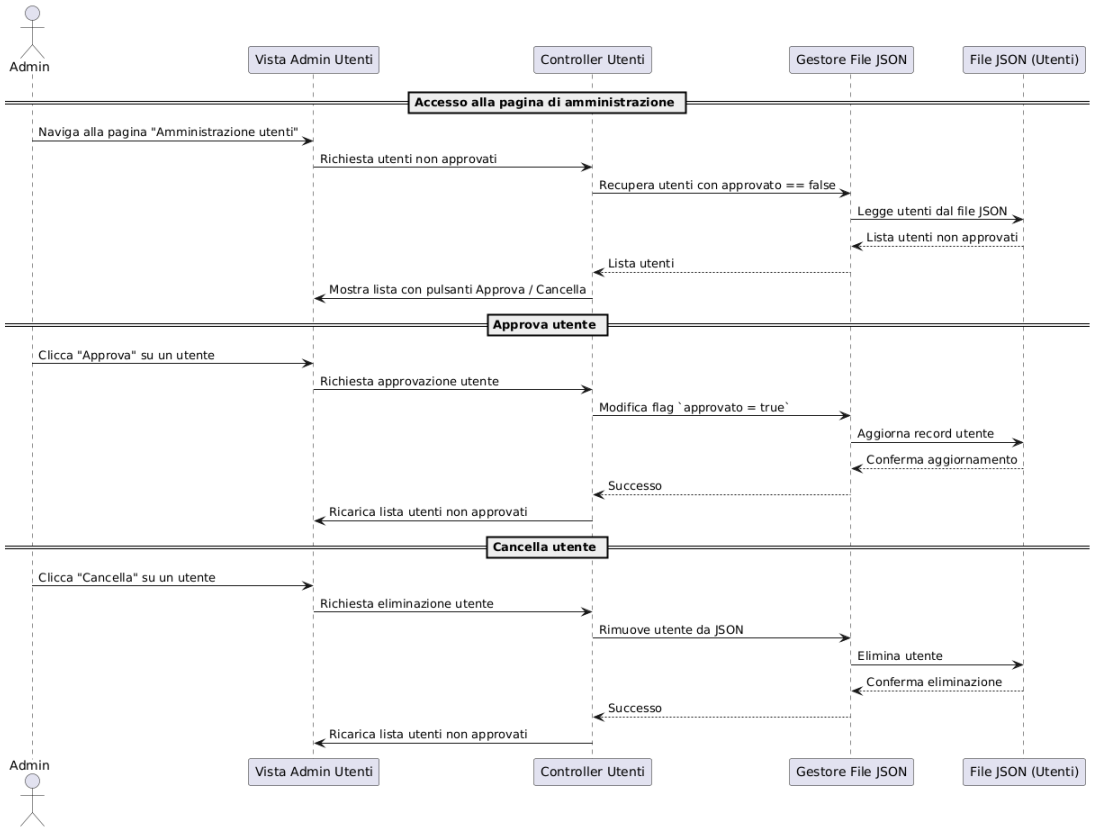

Figura 11: Diagramma di attività della cronologia

3 Implementazione e sviluppo software

3.1 Processo sviluppo

Il processo di sviluppo seguito è stato principalmente di tipo Agile e incrementale. Nonostante ciò, si è cercato, per quanto possibile, di mantenere una certa sequenzialità tra le fasi di progettazione, implementazione e validazione, pur all'interno di un ciclo iterativo. Questo approccio è stato scelto per assicurarsi che ogni fase di sviluppo fosse preceduta da una fase di progettazione solida, con l'obiettivo di lavorare sempre su basi ragionate e stabili.

Al termine di ogni modifica rilevante (ossia a ogni versione del software), è stata svolta una breve fase di verifica e test. Va sottolineato che il flusso di lavoro non è stato perfettamente lineare: durante lo sviluppo si è fatto spesso refactoring su componenti già esistenti, per migliorarne qualità e struttura.

Parallelamente a queste attività, in ogni ciclo è stata curata anche la documentazione, che ha incluso il consolidamento dei diagrammi UML prodotti in fase di progettazione e test (soprattutto diagrammi delle classi), nonché la redazione di ulteriori modelli UML descrittivi, come i diagrammi di sequenza, realizzati solitamente dopo l'implementazione dei relativi metodi.

Prima dell'avvio del ciclo di sviluppo vero e proprio, è stata condotta una fase di analisi dei requisiti, durante la quale sono stati definiti gli use case e i diagrammi di attività. Inoltre, si è svolta una progettazione architetturale preliminare, semplice ma sufficiente per garantire una base strutturale chiara e coerente.

Per quanto riguarda l'implementazione, non è stato seguito un piano di sviluppo rigido o suddiviso in moduli predefiniti. Si è invece proceduto seguendo una logica basata sulle priorità, andando a realizzare le funzionalità ritenute di volta in volta più urgenti o fondamentali.

3.2 Progettazione e pattern

Il sistema è stato progettato utilizzando un approccio basato sulla programmazione ad oggetti, sfruttando le relative tecniche di modellazione. A livello architetturale, si è scelto di adottare il pattern MVC (Model-View-Controller).

La scelta è stata dettata dalla naturale compatibilità di JavaFX con il modello MVC, che ben si adatta alla separazione delle responsabilità tra interfaccia grafica, logica applicativa e gestione dei dati. Questo ha reso l'adozione del pattern sia funzionale che coerente con le tecnologie utilizzate.

Il pattern MVC ha consentito di mantenere una chiara distinzione tra le tre componenti principali, sia dal punto di vista concettuale che nell'organizzazione del codice. Ciascuna parte è stata collocata in un package separato, secondo la seguente struttura:

Modello (Model): contiene la rappresentazione dei dati e delle entità gestite dal sistema. Definisce la struttura delle informazioni persistenti e si occupa della loro gestione.

Vista (View): rappresenta l'interfaccia grafica, realizzata interamente con JavaFX. È responsabile della presentazione dei dati all'utente e dell'interazione visiva.

Controllore (Controller): implementa la logica applicativa e gestisce le interazioni dell'utente. Riceve gli input dalla vista tramite eventi (come `onAction` o `setOnMouseClicked`), li elabora, modifica il modello e aggiorna di conseguenza la vista.

Le azioni dell'utente vengono intercettate dai listener definiti nei controller. Questi ultimi reagiscono modificando lo stato del modello, che a sua volta può notificare la vista o essere visualizzato in modo aggiornato.

Non sono state introdotte suddivisioni ulteriori: modello, vista e controllore sono stati mantenuti come componenti monolitiche, in quanto le loro responsabilità risultano ben distinte e gestibili all'interno di questa struttura.

Nella documentazione seguente sono riportati lo schema architetturale del sistema e i diagrammi UML delle classi che compongono il modello e la vista-controllore. Le interazioni principali tra le componenti MVC sono invece illustrate nei diagrammi di sequenza, presenti nella sezione relativa all'implementazione.

3.3 Class Diagram

La class diagram illustra la struttura statica del modello dati del sistema, mettendo in evidenza le principali classi, i loro attributi e le relazioni esistenti tra esse. Tale diagramma è stato progettato tenendo conto della logica del dominio applicativo, e riflette le entità coinvolte nella gestione dei brani musicali, dei concerti, dei commenti e dell'autenticazione utente.

UTENTE Rappresenta un utilizzatore del sistema, con attributi come username, password, email, admin e approved. Gli oggetti Utente sono collegati a contenuti che l'utente ha caricato o commentato.

BRANO Modella un brano musicale. Contiene attributi come idBrano, titolo, autori, strumenti, generi, anno, esecutori, linkYT, documenti(path), idCommenti, e mantiene un riferimento all'oggetto Utente che lo ha caricato.

CONCERTO Rappresenta un concerto(tramite link youtube). Contiene attributi come titolo, data, id. Anche in questo caso, è presente un riferimento all'Utente creatore.

COMMENTO Contiene il testo del commento, l'id, un flag per distinguere tra nota e commento, un riferimento all'Utente autore e le eventuali risposte, oltre all'id del brano a cui si riferisce.

DOCUMENTO Rappresenta un file associato a un Brano. Contiene l'Utente proprietario, il nome del file e il percorso.

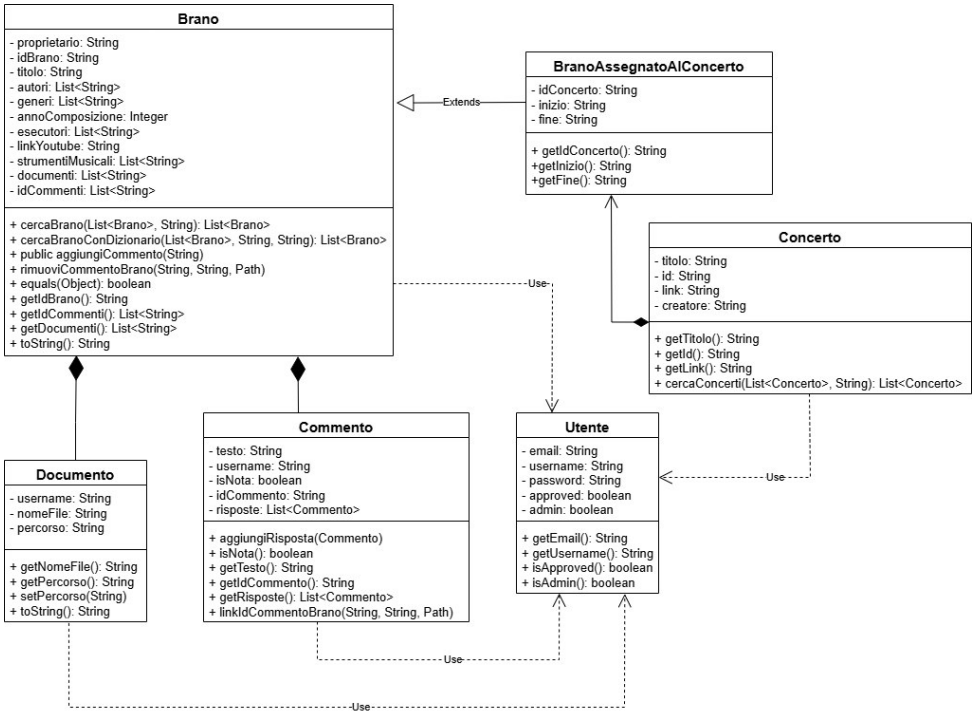

Figura 12: Class Diagram

3.4 Diagrammi di sequenza del software implementato

Seguono diagrammi di sequenza che mostrano le dinamiche di alcune interazioni tra classi particolarmente complesse od interessanti. Sono stati rappresentati soltanto gli scambi salienti.

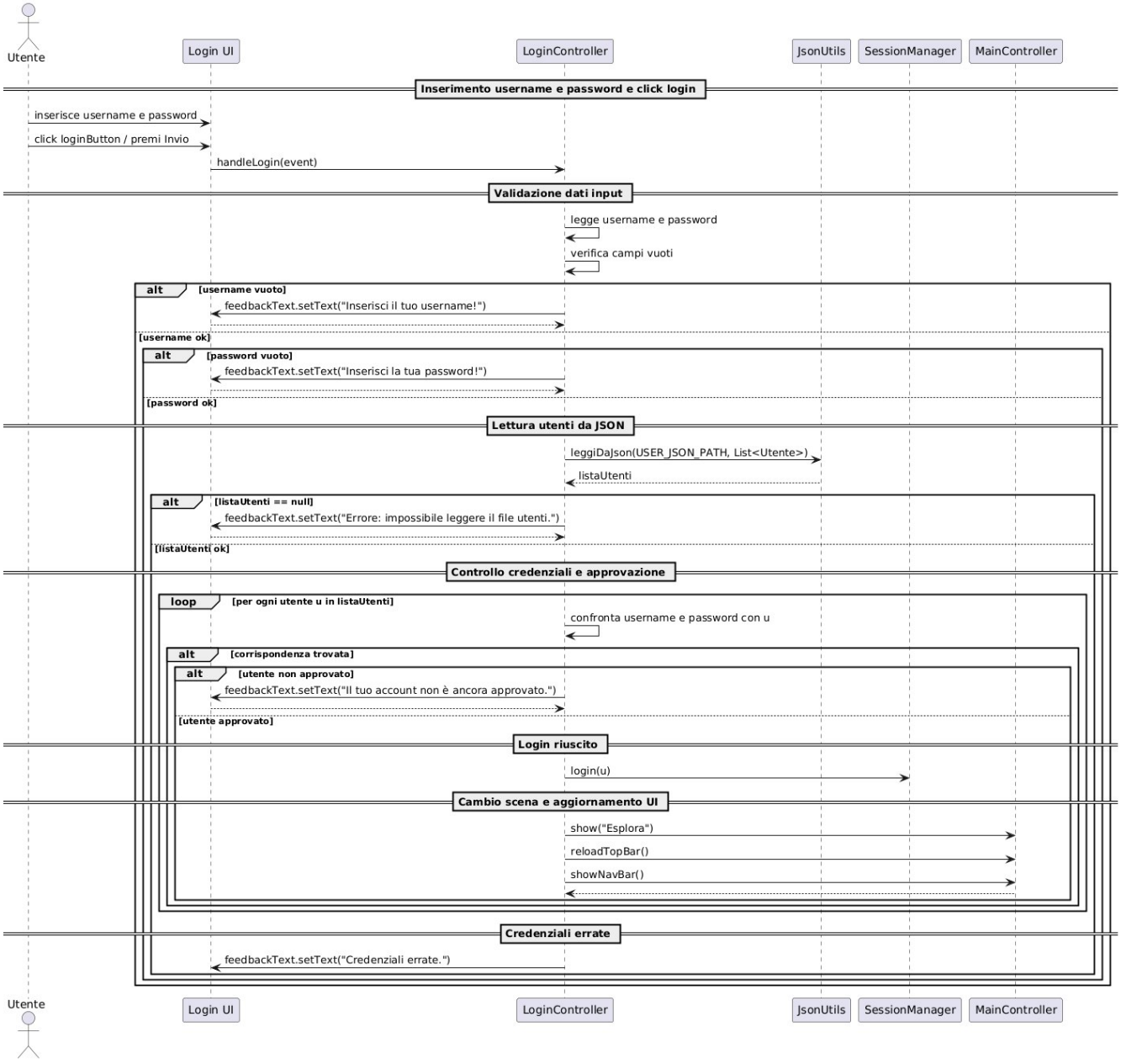

Figura 13: Login

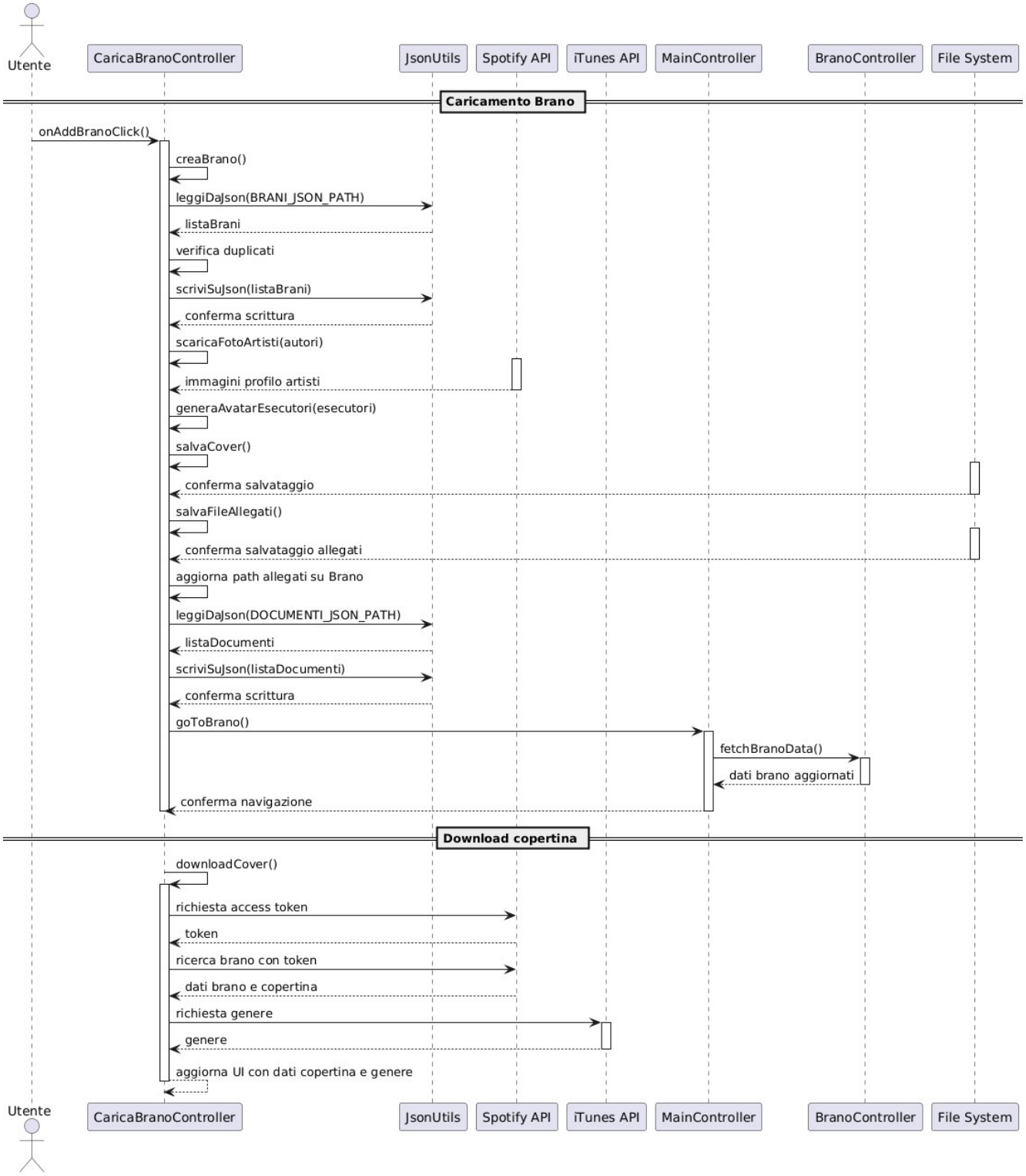

Figura 14: Carica Brano

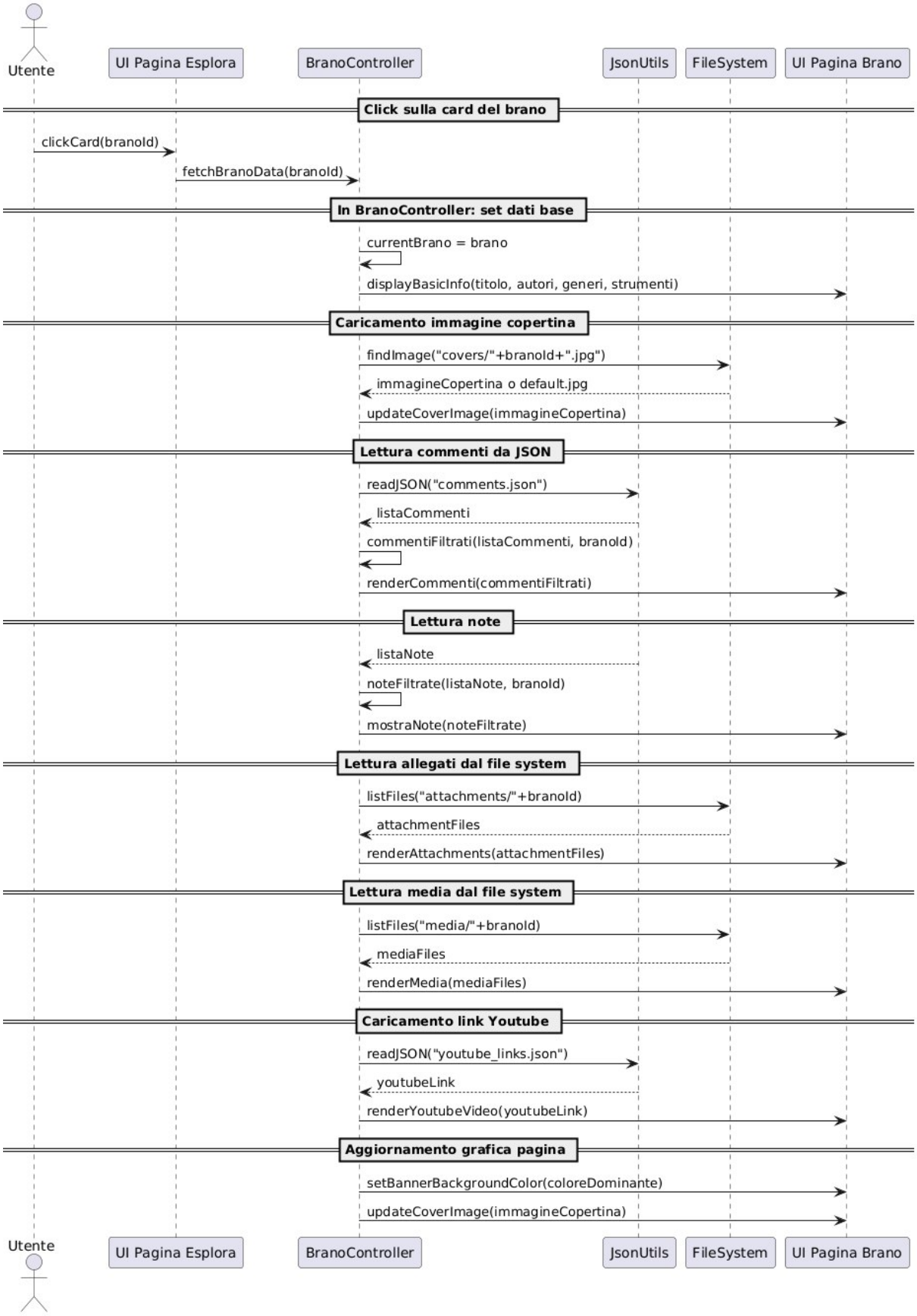

Figura 15: Visualizza Brano

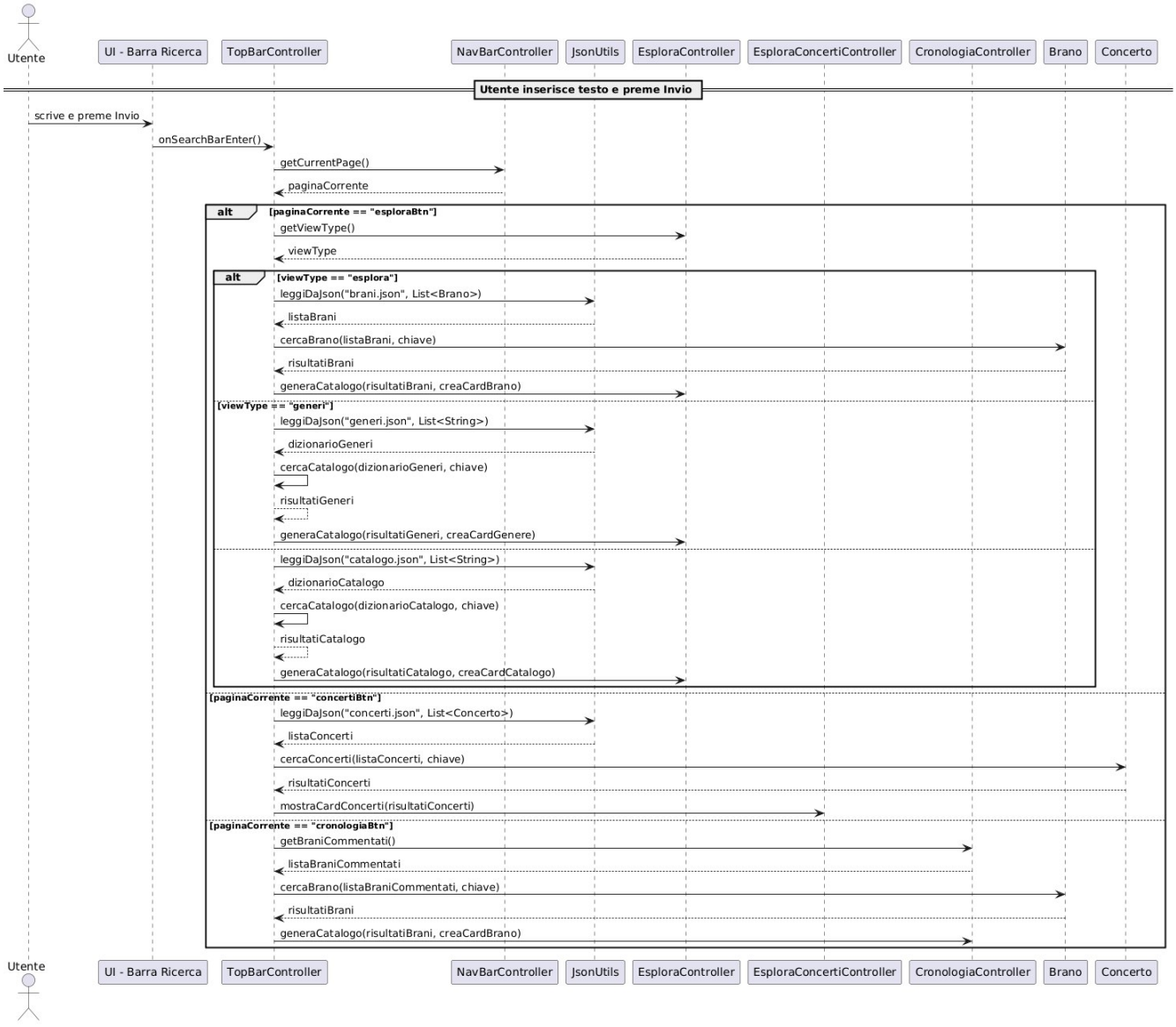

Figura 16: Ricerca

4 Attività di Test e Validazione

4.1 Attività di test e validazione

Al fine di garantire la qualità, l'affidabilità e la conformità del sistema sviluppato rispetto ai requisiti previsti, sono state condotte diverse attività di test e validazione, articolate in più fasi. Queste fasi hanno incluso sia verifiche statiche (analisi del codice, documentazione, diagrammi), sia verifiche dinamiche (test funzionali, test utente).

Tali attività si sono concentrate su:

- Controllo di coerenza tra specifiche, diagrammi UML e implementazione.
- Revisione del codice sorgente, con attenzione alla correttezza logica e all'aderenza ai pattern architetturali.
- Test eseguiti dagli sviluppatori, focalizzati su funzionalità critiche e casi limite.
- Test utente con profili generici, per valutare la fruibilità e l'usabilità del sistema in condizioni reali.

4.2 Test Funzionali degli Sviluppatori

Successivamente, il team di sviluppo ha condotto una serie di test funzionali sul software, verificando che ogni componente implementata fosse aderente alle specifiche richieste.

Per ciascuna funzionalità principale sono stati effettuati:

- Test di verifica del comportamento atteso, secondo i requisiti descritti nel documento iniziale.
- Test di regressione, per garantire che modifiche successive non compromettessero funzionalità già sviluppate.
- Test su scenari limite o imprevisti, con l'obiettivo di individuare eventuali comportamenti anomali o mal gestiti.

Le funzionalità testate nel dettaglio includono:

- Login e Registrazione: verifica dell'autenticazione e della corretta gestione delle credenziali, compresi i messaggi di errore.
- Gestione utenti da parte dell'Admin: approvazione, rimozione e visualizzazione degli utenti non approvati.
- Pagina "Esplora": corretta visualizzazione e accesso ai contenuti (brani e concerti).
- Creazione di un brano: inserimento dati, autocompletamento tramite API (Spotify/iTunes), salvataggio su JSON.
- Pagina del brano: visualizzazione dettagli, gestione dei file allegati, inserimento di commenti e note.
- Visualizzazione concerti: corretta rappresentazione dei concerti registrati, inclusi i dati essenziali e gli embed YouTube.

- Creazione di un concerto: inserimento e salvataggio dei dati nel file concerti.json, e redirect alla pagina dedicata.
- Associazione brani ai concerti: verifica dei controlli su compatibilità e corretta assegnazione.
- Pagina cronologia: conferma della tracciabilità dei brani su cui l'utente ha interagito (commentato o annotato).

Ogni componente è stata testata da più membri del team in momenti distinti, al fine di coprire casi d'uso differenti e simulare un utilizzo più realistico.

4.3 Test Utente Generico (User Acceptance Testing)

Nell'ultima fase di validazione è stato condotto un User Acceptance Test (UAT) coinvolgendo persone esterne al team di sviluppo, anche senza competenze tecniche o informatiche approfondite.

L'obiettivo di questa fase è stato duplice:

- Valutare l'usabilità e l'intuitività dell'interfaccia grafica, realizzata in JavaFX.
- Individuare margini di miglioramento nell'esperienza utente, attraverso feedback diretti.

Durante il test, agli utenti è stato chiesto di:

- Eseguire un accesso e navigare tra le principali sezioni.
- Creare un brano e un concerto.
- Visualizzare contenuti, aggiungere commenti o note.
- Utilizzare la funzione di cronologia.

I feedback raccolti sono stati particolarmente utili per identificare piccoli miglioramenti dell'interfaccia e potenziali nuove funzionalità, oltre a confermare la stabilità delle componenti principali.

Conclusioni

L'intero processo di testing ha permesso di:

- Verificare l'aderenza alle specifiche funzionali.
- Assicurare la robustezza e affidabilità del sistema.
- Migliorare la qualità del codice tramite revisione continua.
- Ottimizzare l'esperienza utente grazie a feedback concreti.

Il software è stato validato sia dal punto di vista tecnico che da quello dell'usabilità, risultando conforme agli obiettivi iniziali del progetto.