

UNIVERSITÀ DEGLI STUDI DI NAPOLI FEDERICO II

SCUOLA POLITECNICA E DELLE SCIENZE DI BASE

DIPARTIMENTO DI INGEGNERIA ELETTRICA E TECNOLOGIE DELL'INFORMAZIONE



CineMate20

Corso di laurea in Informatica Anno Accademico
2020/2021

Committenti

Sergio di Martino
Luigi Libero Lucio Starace

Responsabili

Carmine Testa N86002676
Dmytro Lozyak N86002756





Indice

| | |
|---|------------|
| 1 DOCUMENTO DEI REQUISITI SOFTWARE | 5 |
| 1.1 MODELLO FUNZIONALE: APPLICAZIONE ANDROID | 5 |
| 1.1.1 REQUISITI FUNZIONALI | 5 |
| 1.1.2 REQUISITI NON FUNZIONALI..... | 9 |
| 1.1.3 REQUISITI DI DOMINIO | 10 |
| 1.1.4 MODELLAZIONE USE CASE DIAGRAM (ANDROID)..... | 11 |
| 1.1.5 TABELLE DI COCKBURN (ANDROID) | 14 |
| 1.1.6 MOCK-UP (ANDROID)..... | 41 |
| 1.2 MODELLO FUNZIONALE: APPLICAZIONE DESKTOP | 73 |
| 1.2.1 REQUISITI FUNZIONALI | 73 |
| 1.2.2 REQUISITI NON FUNZIONALI..... | 74 |
| 1.2.3 REQUISITI DI DOMINIO | 74 |
| 1.2.4 MODELLAZIONE USE CASE DIAGRAM (DESKTOP) | 75 |
| 1.2.5 TABELLE DI COCKBURN (DESKTOP)..... | 77 |
| 1.2.6 MOCK-UP (DESKTOP) | 82 |
| 1.3 GLOSSARIO | 87 |
| 1.4 MODELLO DI DOMINIO: APPLICAZIONE ANDROID | 88 |
| 1.4.1 MODELLO A OGGETTI DI ANALISI..... | 88 |
| 1.4.2 DIAGRAMMI DI SEQUENZA DI ANALISI | 113 |
| 1.4.3 DIAGRAMMI DI STATO DI ANALISI..... | 127 |
| 1.4.4 DIAGRAMMI DI ATTIVITÀ | 128 |
| 1.4.5 DIAGRAMMA DI GIANT..... | 129 |
| 1.5 MODELLO DI DOMINIO: APPLICAZIONE DESKTOP | 130 |
| 1.5.1 MODELLO A OGGETTI DI ANALISI..... | 130 |
| 1.5.2 DIAGRAMMI DI SEQUENZA DI ANALISI | 136 |
| 1.6 PROPOSTA DI IMPEGNO CON DIAGRAMMA DI GANTT | 140 |
| 2 DOCUMENTO DI DESIGN DEL SISTEMA | 141 |
| 2.1 ANALISI DELL'ARCHITETTURA | 141 |
| 2.1.1 ARCHITETTURA BACK-END | 142 |
| 2.1.2 ARCHITETTURA FRONT-END | 144 |
| 2.2 DIAGRAMMA DELLE CLASSI DI DESIGN..... | 146 |
| 2.2.1 ANDROID | 147 |
| 2.2.2 DESKTOP..... | 166 |



| | |
|---|------------|
| 2.3 CRC CARDS..... | 171 |
| 2.3.1 ANDROID | 171 |
| 2.3.2 DESKTOP..... | 181 |
| 2.4 DIAGRAMMA DI STATO DI DESIGN | 184 |
| 2.4.1 ANDROID | 184 |
| 2.4.2 DESKTOP..... | 186 |
| 2.5 DIAGRAMMA DI SEQUENZA DI DESIGN | 187 |
| 2.5.1 ANDROID | 187 |
| 2.5.2 DESKTOP..... | 192 |
| 4 DOCUMENTO DI TESTING DEL SISTEMA | 194 |
| 4.1 TEST PLAN PER SYSTEM TESTING APPLICAZIONE ANDROID | 194 |
| 4.1.1 ACCESSO AL SISTEMA | 194 |
| 4.1.2 REGISTRAZIONE AL SISTEMA | 196 |
| 4.1.3 RECUPERO PASSWORD DIMENTICATA..... | 198 |
| 4.1.4 EFFETTUARE LA RICERCA DI FILM..... | 199 |
| 4.1.5 CREAZIONE DI UNA LISTA DI FILM PERSONALIZZATA | 200 |
| 4.1.6 RIMOZIONE DI UNA LISTA DI FILM PERSONALIZZATA..... | 201 |
| 4.1.7 AGGIUNTA/RIMOZIONE DI UN FILM AD/DA UNA LISTA | 202 |
| 4.1.8 RIMOZIONE AMICIZIA | 203 |
| 4.1.9 EFFETTUARE LE RICERCHE DI UTENTI | 204 |
| 4.1.10 INVIARE LE RICHIESTE DI AMICIZIA | 205 |
| 4.1.12 RIFIUTARE LA RICHIESTA DI AMICIZIA | 206 |
| 4.1.13 VISUALIZZAZIONE DELLA SCHEDA DI UN FILM | 207 |
| 4.1.14 VISUALIZZAZIONE DELLA SCHEDA UTENTE | 208 |
| 4.1.15 VISUALIZZAZIONE DI UNA LISTA | 209 |
| 4.1.16 VISUALIZZAZIONE DEI FILM IN COMUNE CON AMICO | 210 |
| 4.1.17 VISUALIZZAZIONE DELLE NOTIFICHE | 211 |
| 4.1.18 EFFETTUARE UNA SEGNALAZIONE..... | 212 |
| 4.1.19 CAMBIARE LA PASSWORD | 213 |
| 4.1.20 CAMBIARE IMMAGINE DI PROFILO | 214 |
| 4.1.21 USCIRE DAL PROFILO | 215 |
| 4.1.22 MODIFICARE LA DESCRIZIONE DI UNA LISTA..... | 216 |
| 4.1.23 VISUALIZZAZIONE DEL PROPRIO PROFILO | 217 |
| 4.2 TEST PLAN PER SYSTEM TESTING APPLICAZIONE DESKTOP | 218 |
| 4.2.1 ACCESSO AL SISTEMA | 218 |
| 4.2.2 VISUALIZZARE UNA SEGNALAZIONE..... | 219 |
| 4.2.3 GESTIRE UNA SEGNALAZIONE..... | 220 |
| 4.2.4 USCIRE DAL SISTEMA..... | 221 |
| 4.2.5 VISUALIZZARE IL REGOLAMENTO..... | 222 |



| | |
|---|-----|
| 4.3 CODICE XUNIT PER UNIT TESTING | 223 |
| 4.3.1 UNIT TESTING PER IL METODO <i>CALCULATESECRETHASH</i> | 223 |
| 4.3.2 UNITE TESTING PER IL METODO <i>CHECKCAMPIVALIDI</i> | 226 |
| 4.3.3 UNIT TESTING PER IL METODO <i>ORDINALISTA</i> | 229 |
| 4.3.4 UNIT TESTING PER IL METODO <i>GETUTENTEJSONOBJECTCONCURRENTSTATUS</i> | 235 |



1 Documento dei Requisiti Software

1.1 Modello Funzionale: applicazione Android

1.1.1 Requisiti Funzionali

| | |
|--------------------|--|
| ID | AND_RF01 |
| Nome | Registrazione nel sistema. |
| Descrizione | Il sistema permette all'utente ancora non registrato di effettuare la registrazione, indicando Username, Password e E-mail. |
| ID | AND_RF02 |
| Nome | Accesso al sistema. |
| Descrizione | Il sistema permette all'utente già registrato di accedere all'applicazione, indicando correttamente E-mail e Password oppure utilizzando un account di Google o Facebook. |
| ID | AND_RF03 |
| Nome | Recupero password dimenticata. |
| Descrizione | Il sistema permette all'utente che ha smarrito la password di recuperarla, specificando l'e-mail legata al proprio account. Il sistema deve inviare un codice di conferma, permettendo all'utente di cambiare la propria password. |
| ID | AND_RF04 |
| Nome | Effettuare ricerche di film. |
| Descrizione | Il sistema permette di effettuare le ricerche di film specificando il nome del film Il sistema mostrerà una lista di risultati per query effettuata. |



| | |
|--------------------|--|
| ID | AND_RF05 |
| Nome | Creazione/ Rimozione di liste di film personalizzate. |
| Descrizione | Il sistema permette di creare/rimuovere le liste di film personalizzate, la creazione è caratterizzata da un titolo e da una descrizione (opzionale) personalizzata ed è pubblica. |

| | |
|--------------------|---|
| ID | AND_RF06 |
| Nome | Aggiunta/Rimozione di film dalla propria lista. |
| Descrizione | Il sistema permette di aggiungere/rimuovere un film alla/dalla lista personale. |

| | |
|--------------------|---|
| ID | AND_RF07 |
| Nome | Effettuare le ricerche di utenti. |
| Descrizione | Il sistema permette di ricercare gli altri utenti registrati sulla piattaforma, il sistema effettuerà la ricerca riferendosi all'attributo username degli utenti. |

| | |
|--------------------|---|
| ID | AND_RF08 |
| Nome | Inviare/Ricevere le richieste di amicizia a/da altri utenti. |
| Descrizione | Il sistema permette di Inviare/Ricevere le richieste di amicizia a/da altri utenti. |

| | |
|--------------------|--|
| ID | AND_RF09 |
| Nome | Accettare/Rifiutare le richieste di amicizia a/da altri utenti. |
| Descrizione | Il sistema permette di Accettare o Rifiutare le richieste di amicizia a/da altri utenti. |



| | |
|--------------------|---|
| ID | AND_RF10 |
| Nome | Visualizzazione della scheda di un film. |
| Descrizione | <p>Il sistema permette di visualizzare i dettagli di un film, mostrando:</p> <ul style="list-style-type: none"> • Copertina del film • Titolo • Durata • Genere • Valutazione • Descrizione |

| | |
|--------------------|--|
| ID | AND_RF11 |
| Nome | Visualizzazione di un utente. |
| Descrizione | <p>Il sistema permette di visualizzare le informazioni di un utente, in particolare:</p> <ul style="list-style-type: none"> • Username • Immagine di profilo • Elenco delle liste • Relazione di amicizia • Elenco film in comune (solo nel caso utente sia un amico) |

| | |
|--------------------|--|
| ID | AND_RF12 |
| Nome | Visualizzazione di una lista. |
| Descrizione | <p>Il sistema permette di visualizzare il contenuto di una particolare lista, in particolare:</p> <ul style="list-style-type: none"> • Titolo della lista • Descrizione di una lista • Elenco di film che la compongono |

| | |
|--------------------|--|
| ID | AND_RF13 |
| Nome | Visualizzazione dei film in comune con un amico. |
| Descrizione | Il sistema permette di visualizzare i film in comune con un particolare amico sottoforma di una lista. |



| | |
|--------------------|--|
| ID | AND_RF14 |
| Nome | Visualizzazione delle notifiche. |
| Descrizione | <p>Il sistema permette di visualizzare le notifiche, quali possono essere:</p> <ul style="list-style-type: none"> • Ricezione di una richiesta di amicizia da parte un altro utente. • Accettazione/Rifiuto di una richiesta di amicizia da parte di un altro utente. • Il provvedimento preso in seguito ad una segnalazione inviata/ricevuta. |

| | |
|--------------------|--|
| ID | AND_RF15 |
| Nome | Effettuare una segnalazione. |
| Descrizione | <p>Il sistema permette di segnalare una lista di altri utenti, specificando il motivo della segnalazione tra:</p> <ul style="list-style-type: none"> • Spoiler • Contenuti violenti o ripugnanti • Contenuti offensivi od oltraggiosi |

| | |
|--------------------|---|
| ID | AND_RF16 |
| Nome | Cambiare la Password. |
| Descrizione | Il sistema permette ad un utente loggato può cambiare la Password di accesso. |

| | |
|--------------------|---|
| ID | AND_RF17 |
| Nome | Cambiare immagine di profilo. |
| Descrizione | Il sistema permette ad un utente loggato cambiare la propria immagine di profilo. |

| | |
|--------------------|---|
| ID | AND_RF18 |
| Nome | Uscire dal profilo. |
| Descrizione | Il sistema permette ad un utente loggato di uscire dal proprio account, tramite tasto Logout. |



| | |
|--------------------|--|
| ID | AND_RF19 |
| Nome | Modificare la descrizione di una lista. |
| Descrizione | Il sistema permette all'utente di modificare la descrizione di una lista di sua proprietà. |

| | |
|--------------------|--|
| ID | AND_RF20 |
| Nome | Visualizzazione del proprio profilo. |
| Descrizione | Il sistema permette all'utente di visualizzare il proprio profilo. |

| | |
|--------------------|--|
| ID | AND_RF21 |
| Nome | Rimozione dell'amicizia |
| Descrizione | Il sistema permette all'utente di eliminare utenti dalla propria lista di amici. |

1.1.2 Requisiti non Funzionali

| | |
|--------------------|--|
| ID | AND_RNF01 |
| Nome | Performance di ricerca |
| Descrizione | Il sistema deve mostrare i risultati di ricerca per un film o un utente entro 5 secondi dall'invio della query nel 95% dei casi. |

| | |
|--------------------|---|
| ID | AND_RNF02 |
| Nome | Usabilità del sistema |
| Descrizione | Un utente medio deve riuscire a comprendere le funzionalità dell'applicazione ed essere in grado di utilizzarle entro 1 ora dell'inizio del suo utilizzo. |

| | |
|--------------------|--|
| ID | AND_RNF03 |
| Nome | Limitazione numero segnalazioni |
| Descrizione | Un utente loggato potrà effettuare al più una volta segnalazione per ogni lista. |



| | |
|--------------------|---|
| ID | AND_RNF04 |
| Nome | Validità password |
| Descrizione | <p>Un account deve possedere una password che rispetta i criteri di sicurezza del sistema, ovvero:</p> <ul style="list-style-type: none"> • Lunghezza minima di 8 caratteri • Lunghezza massima di 30 caratteri |

| | |
|--------------------|---|
| ID | AND_RNF05 |
| Nome | Validità username |
| Descrizione | <p>Un account deve possedere uno username che rispetta i criteri del sistema ovvero:</p> <ul style="list-style-type: none"> • Non è possibile utilizzare username già presenti (Unicità) |

| | |
|--------------------|---|
| ID | AND_RNF06 |
| Nome | Validità titolo e descrizione per una lista personalizzata |
| Descrizione | <p>Una lista personalizzata deve rispettare i criteri del sistema ovvero:</p> <ul style="list-style-type: none"> • Limite di lunghezza a 30 caratteri per il titolo e 200 caratteri per la descrizione |

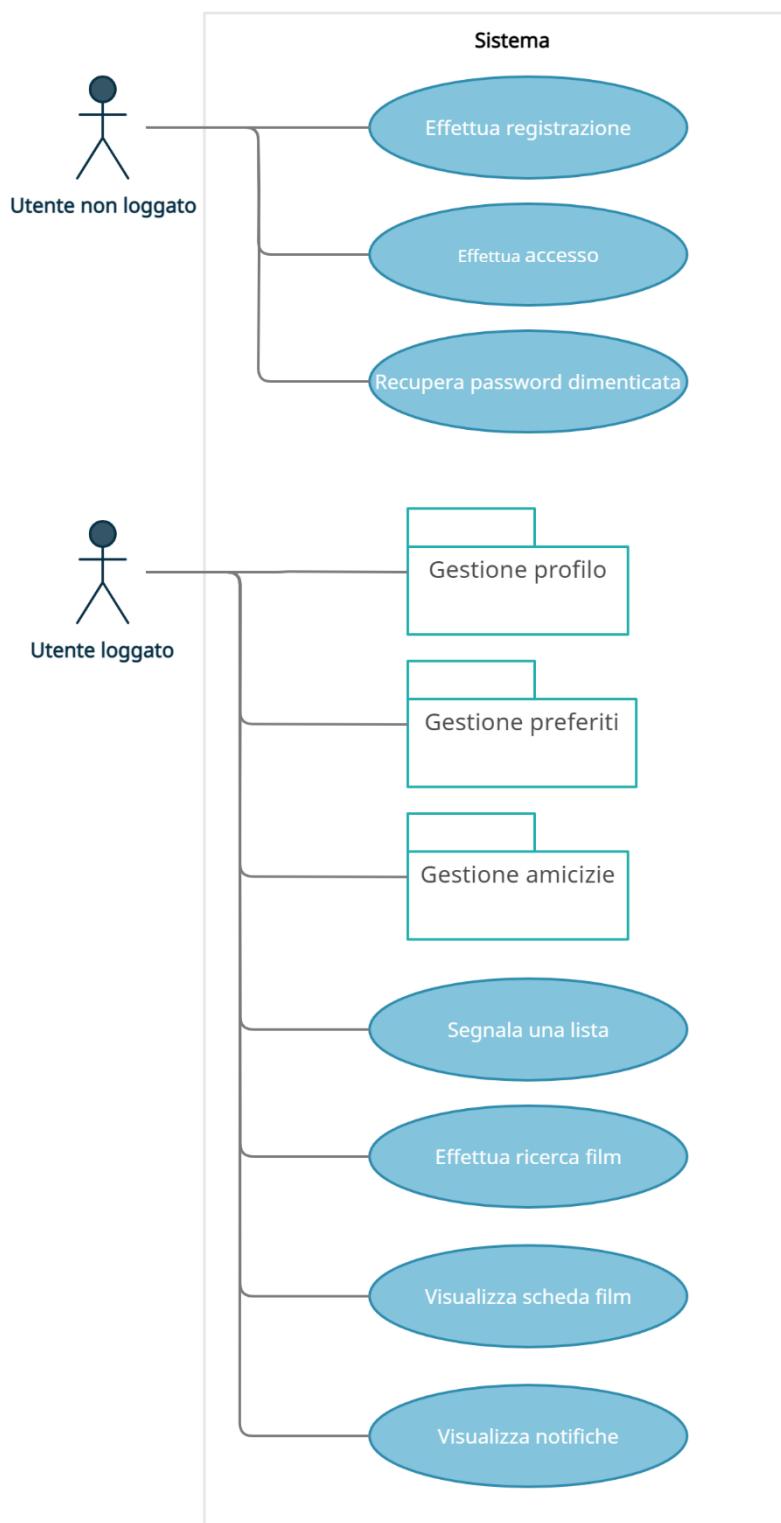
| | |
|--------------------|---|
| ID | AND_RNF07 |
| Nome | Affidabilità del sistema |
| Descrizione | <p>Un utente che perde la connessione, successivamente al ripristino della connessione deve poter riconnettersi senza reinserire i campi username e password.</p> |

1.1.3 Requisiti di dominio

| | |
|--------------------|--------------------|
| ID | AND_RD01 |
| Nome | ISO/IEC 27018:2020 |
| Descrizione | |

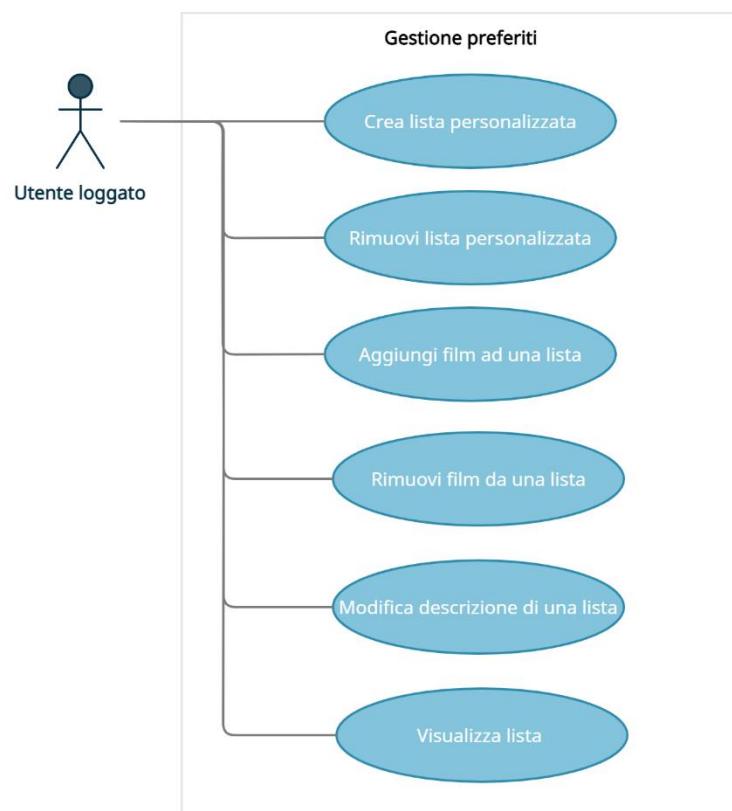


1.1.4 Modellazione use case diagram (Android)

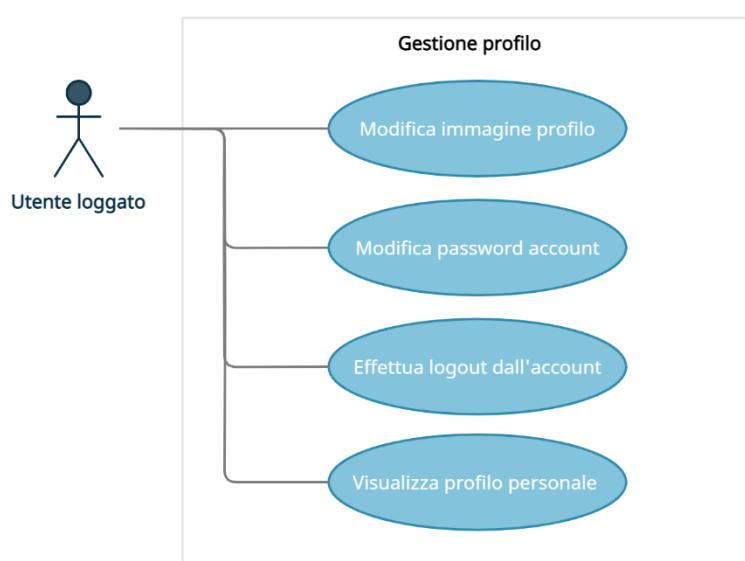


Use case diagram 1 – Sistema principale



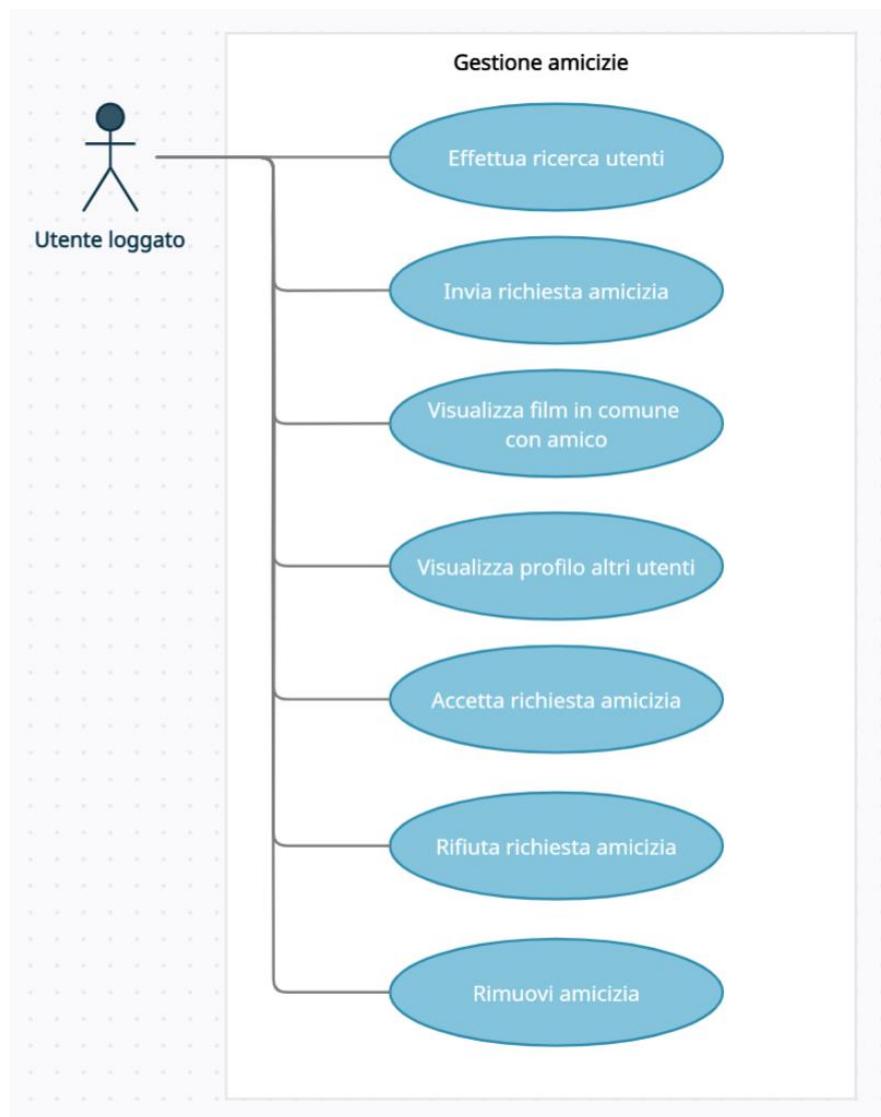


Use case diagram 2 – Gestione preferiti



Use case diagram 3 – Gestione profilo





Use case diagram 4 – Gestione amicizie



1.1.5 Tabelle di Cockburn (Android)

1.1.5.1 – Accesso al sistema

| USER CASE #1 | Accesso al sistema | | |
|--|---|---|---|
| Goal in Context | L'utente vuole effettua accesso al sistema | | |
| Preconditions | L'utente non è autentificato | | |
| Success End Condition | L'utente effettua accesso al sistema | | |
| Failer End Condition | L'utente non riesce effettuare accesso al sistema | | |
| Primary Actor | Utente non loggato | | |
| Trigger | Utente apre applicazione CineMate20 | | |
| DESCRIPTION | STEP | UTENTE NON LOGGATO | SISTEMA |
| EXTENSION #1 Utente decide di loggare con Google | 1 | Inserisce Email e Password e preme il pulsante "Login" nella schermata Login AND_MK01 | |
| | 2 | | Mostra la schermata Home AND_MK03 |
| EXTENSION #2 Utente decide di loggare con Facebook | STEP | UTENTE NON LOGGATO | SISTEMA |
| | 1a | Preme il pulsante "Sign in with Google" | |
| | 2a | | Mostra la schermata di accesso Google |
| | 3a | Seleziona account Google per loggare | |
| | 4a | | Torna allo step 2 dello scenario principale |
| EXTENSION #2 Utente decide di loggare con Facebook | STEP | UTENTE NON LOGGATO | SISTEMA |
| | 1b | Preme il pulsante "Sign in with Facebook" | |
| | 2b | | Mostra la schermata di accesso Facebook |
| | 3b | Seleziona account Facebook per loggare | |
| | 4b | | Torna allo step 2 dello scenario principale |



| EXTENSION #3 Utente compila i campi con i dati non validi o mancanti: - E-mail non valida - Password errata | STEP | UTENTE NON LOGGATO | SISTEMA |
|---|-------------|---------------------------|---|
| | 2c | | Mostra un messaggio di errore AND_MSG08 |
| | 3c | | Torna allo step 1 dello scenario principale |



1.1.5.2 – Registrazione al sistema

| USER CASE #2 | | | | Registrazione al sistema |
|---|-------------|--|----------------|---|
| Goal in Context | | | | I'utente vuole registrarsi al sistema |
| Preconditions | | | | L'utente non è autentificato |
| Success End Condition | | | | I'utente effettua la registrazione al sistema |
| Failer End Condition | | | | I'utente non riesce effettuare la registrazione al sistema |
| Primary Actor | | | | Utente non loggato |
| Trigger | | | | L'utente preme il pulsante "Registrazione" nella schermata di Login |
| DESCRIPTION | STEP | UTENTE NON LOGGATO | SISTEMA | |
| Utente non compila tutti i campi o non rispetta i requisiti di compilazione | 1 | L'utente preme il pulsante "Registrazione" nella schermata di Login AND_MK01 | | |
| | 2 | | | Mostra schermata di Registrazione AND_MK02 |
| | 3 | Compila i campi e preme il pulsante "Registrati" | | |
| | 4 | | | Mostra la schermata Login AND_MK01 |
| EXTENSION #1 Utente non compila tutti i campi o non rispetta i requisiti di compilazione | STEP | UTENTE NON LOGGATO | SISTEMA | |
| | 4a | | | Mostra un messaggio di errore compilazione campi AND_MSG10 |
| | 5a | | | Ritorna allo step 2 dello scenario principale |
| EXTENSION #2 Utente sbaglia confermare la password | STEP | UTENTE NON LOGGATO | SISTEMA | |
| | 4b | | | Mostra un messaggio di errore "conferma password" AND_MSG03 |
| | 5b | | | Ritorna allo step 2 dello scenario principale |
| EXTENSION #3 Utente utilizza Username già registrato | STEP | UTENTE NON LOGGATO | SISTEMA | |
| | 4c | | | Mostra un messaggio di errore "Username esistente" AND_MSG01 |
| | 5c | | | Ritorna allo step 2 dello scenario principale |
| EXTENSION #4 Utente utilizza E-mail già registrata | STEP | UTENTE NON LOGGATO | SISTEMA | |
| | 4d | | | Mostra un messaggio di errore "E-mail già utilizzata" AND_MSG02 |
| | 5d | | | Ritorna allo step 2 dello scenario principale |



| EXTENSION #5 Utente compila i campi e preme "Annulla" | STEP | UTENTE NON LOGGATO | SISTEMA |
|---|------|-----------------------------|--|
| | 3e | Preme il pulsante "Annulla" | |
| | 4e | | Torna alla schermata di Login AND_MK01 |



1.1.5.3 – Recupero password dimenticata

| USER CASE #3 | Recupero password dimenticata | | |
|--|--|--|--|
| Goal in Context | L'utente ha smarrito la password del suo account e la vuole recuperare | | |
| Preconditions | L'utente non è autentificato | | |
| Success End Condition | L'utente riesce a recuperare la password cambiandola | | |
| Failer End Condition | L'utente non riesce a recuperare la password dimenticata | | |
| Primary Actor | Utente non loggato | | |
| Trigger | L'utente preme "Password Dimenticata" | | |
| DESCRIPTION | STEP | UTENTE NON LOGGATO | SISTEMA |
| EXTENSION #1 Utente sbaglia inserire e-mail | 1 | L'utente preme il pulsante "Password Dimenticata" nella schermata AND_MK01 | |
| | 2 | | Mostra schermata di Password Dimenticata AND_MK33 |
| | 3 | Compila i campi | |
| | 4 | Preme il pulsante "Invia codice di recupero" | |
| | 5 | | Mostra la schermata inserire il codice di conferma e nuova password AND_MK34 |
| | 6 | Inserisce codice di conferma e la nuova password | |
| | 7 | Preme pulsante "Salva" | |
| | 8 | | Mostra la schermata login AND_MK01 e il messaggio AND_MSG12 |
| EXTENSION #1 Utente sbaglia inserire e-mail | STEP | UTENTE NON LOGGATO | SISTEMA |
| | 5a | | Mostra un messaggio di errore compilazione campi AND_MSG10 |
| | 6a | | Ritorna alla schermata login AND_MSG01 |



| | STEP | UTENTE NON LOGGATO | SISTEMA |
|---|--------------|------------------------------------|---|
| EXTENSION #2 Utente sbaglia confermare il codice | 8b | | Mostra un messaggio di errore AND_MSG10 |
| | 9b | | Ritorna alla schermata login AND_MSG01 |
| EXTENSION #3 Utente non compila tutti i campi o li compila in maniera errata | 8c | | Mostra un messaggio di errore AND_MSG10 |
| | 9c | | Ritorna alla schermata login AND_MSG01 |
| EXTENSION #4 Utente annulla l'operazione | STEP | UTENTE NON LOGGATO | SISTEMA |
| | 4d - 7d - | Utente preme il pulsante "Annulla" | |
| | 5d - 8d - | | Mostra la schermata login AND_MSG01 |



1.1.5.4 – Effettuare la ricerca di film

| USER CASE #4 | | | | Effettuare la ricerca di film |
|---|-------------|--|----------------|---|
| Goal in Context | | | | L'utente vuole trovare un film |
| Preconditions | | | | L'utente è autentificato |
| Success End Condition | | | | L'utente trova film ricercato |
| Failer End Condition | | | | L'utente non riesce trovare il film ricercato |
| Primary Actor | | | | Utente loggato |
| Trigger | | | | Utente preme il pulsante "Ricerca" |
| DESCRIPTION | STEP | UTENTE LOGGATO | SISTEMA | |
| | 1 | Preme il pulsante "Ricerca" nella schermata home AND_MK03 | | |
| | 2 | | | Mostra la schermata di ricerca film AND_MK10 |
| | 3 | Inserire le parole di ricerca e preme invio sulla tastiera | | |
| | 4 | | | Mostra la schermata con film trovati AND_MK12 |
| | 5 | Clicca sul film interessato | | |
| | 6 | | | Mostra la scheda di un film AND_MK13 |
| EXTENSION #1 Non ci sono i risultati per le parole di ricerca | STEP | UTENTE LOGGATO | SISTEMA | |
| | 4a | | | Mostra messaggio MSG_19 |



1.1.5.5 – Creazione di una lista di film personalizzata

| USER CASE #5 | | | | Creazione di una lista di film personalizzata |
|------------------------------|--|---|----------------|---|
| Goal in Context | | | | L'utente vuole creare una lista personalizzata |
| Preconditions | | | | L'utente è autentificato |
| Success End Condition | | | | L'utente ha creato una nuova lista personalizzata |
| Failer End Condition | | | | Nessuna |
| Primary Actor | | | | Utente loggato |
| Trigger | Utente preme il pulsante "Nuova lista" nella schermata Home o nella schermata Menu a tendina liste | | | |
| DESCRIPTION | STEP | UTENTE LOGGATO | SISTEMA | |
| | 1 | Preme il pulsante "Nuova lista" nella schermata Home AND_MK03 o nelle schermate Menu a tendina liste AND_MK25 | | |
| | 2 | | | Mostra la schermata creazione di una lista AND_MK26 |
| | 3 | Inserisce il Titolo e la descrizione | | |
| | 4 | Preme il pulsante "Salva" | | |
| | 5 | | | Mostra il messaggio associato alla creazione di una nuova lista AND_MSG04 |
| EXTENSIONE #1 | | | | |
| Utente annulla l'operazione | STEP | UTENTE LOGGATO | SISTEMA | |
| | 3a-4b | Utente preme il tasto annulla | | |
| | 4a-5b | | | Mostra la schermata Home AND_MK03 o qualunque schermata dalla quale è stato invocato menu a tendina. Termina caso d'uso |



1.1.5.6 – Rimozione di una lista di film personalizzata

| USER CASE #6 | Rimozione di una lista di film personalizzata | | |
|--|---|---|--|
| Goal in Context | L'utente vuole rimuovere una lista personalizzata | | |
| Preconditions | L'utente è autenticato e ha almeno una lista di film personalizzata | | |
| Success End Condition | L'utente ha cancellato una lista di lista personalizzata | | |
| Failer End Condition | Nessuna | | |
| Primary Actor | Utente loggato | | |
| Trigger | Utente preme il tasto "Cancella lista" dal menu a tendina nella schermata Visualizzazione di tutte le liste | | |
| DESCRIPTION | STEP | UTENTE LOGGATO | SISTEMA |
| EXTENSIONE #1 Utente annulla l'operazione | 1 | Preme il pulsante "Cancella lista" dal menu a tendina nella schermata Visualizzazione di tutte le liste AND_MK07 | |
| | 2 | | Mostra la tendina AND_MK31 |
| | 3 | Preme il pulsante "Cancella lista" | |
| | 4 | | Mostra il messaggio della conferma di una lista cancellata AND_MSG06 |
| EXTENSIONE #1 Utente annulla l'operazione | STEP | UTENTE LOGGATO | SISTEMA |
| | 3a | Effettua operazione di annullamento dell'operazione | |
| | | | Mostra schermata <Visualizzazione di tutte le liste AND_MK07. Termina caso d'uso |



1.1.5.7 – Aggiunta/Rimozione di un film ad una lista

| USER CASE #7 | Aggiunta/Rimozione un film ad/da una lista | | |
|---|---|--|--|
| Goal in Context | L'utente vuole aggiungere/rimuovere un film in una propria lista personalizzata | | |
| Preconditions | L'utente è autentificato e possiede almeno una lista | | |
| Success End Condition | L'utente riesce ad aggiungere/rimuovere un film in una lista personalizzata scelta tra quelle possedute dall'utente | | |
| Failer End Condition | L'utente non riesce ad aggiungere/rimuovere il film ad una lista | | |
| Primary Actor | Utente loggato | | |
| Trigger | Utente preme il tasto "+" dalla schermata scheda film o schermata Elenco film ricerca | | |
| DESCRIPTION | STEP | UTENTE LOGGATO | SISTEMA |
| | 1 | Preme il tasto "+" dalla schermata scheda film AND_MK13 o schermata Elenco film ricerca AND_MK12 | |
| | 2 | | Mostra la schermata Menu a tendina liste AND_MK25 |
| | 3 | Seleziona/deseleziona, tramite la checkbox, le liste in cui si vuole aggiungere il film | |
| | 4 | Premere Fatto | |
| | 5 | | Mostra il messaggio di conferma aggiunta film in una lista AND_MSG15 |
| EXTENSION #1 L'utente preme Fatto senza selezionare una lista | STEP | UTENTE LOGGATO | SISTEMA |
| | 5a | | Mostra il messaggio di conferma aggiunta film in una lista AND_MSG15 |
| | | | |
| EXTENSION #2 L'utente aggiunge/rimuove il film ad una lista tramite l'elenco dei film in una lista | STEP | UTENTE LOGGATO | SISTEMA |
| | 1c | Utente preme sul menù al lato di un film nella schermata elenco film in lista AND_MK08/AND_MK21 | |
| | 2c | | Mostra Menu a tendina opzioni film AND_MK28/AND_MK32 |
| | 3c | Utente preme "salva in una lista" dal menu a tendina AND_MK28/AND_MK32 | |
| | 4c | | Torna al punto 2 dello scenario principale |



| EXTENSION #3 | STEP | UTENTE LOGGATO | SISTEMA |
|---|------|----------------------------|---|
| L'utente invece di premere fatto torna indietro, premendo tasto Annulla | 4d | Utente preme tasto Annulla | |
| | 5d | | Mostra la schermata Scheda Film AND_MK13 o Elenco film ricerca AND_MK12 |



1.1.5.8 – Rimozione amicizia

| USER CASE #8 | Rimozione amicizia | | |
|-----------------------|---|--|---|
| Goal in Context | L'utente vuole rimuovere un amico | | |
| Preconditions | L'utente è autentificato e possiede almeno un amico | | |
| Success End Condition | L'utente riesce a rimuovere un amico dalla propria lista di amici | | |
| Failer End Condition | L'utente non riesce a rimuovere un amico | | |
| Primary Actor | Utente loggato | | |
| Trigger | Utente preme il tasto "Rimuovi amico" dalla schermata ricerca utenti, schermata altro utente, schermata Amici | | |
| DESCRIPTION | STEP | UTENTE LOGGATO | SISTEMA |
| | 1 | Utente preme il tasto "Rimuovi amico" dalla schermata ricerca utenti AND_MK16, schermata altro utente AND_MK18, schermata Amici AND_MK04 | |
| | 2 | | Mostra il messaggio di conferma AND_MSG20 |



1.1.5.9 – Effettuare le ricerche di utenti

| USER CASE #9 | Effettuare le ricerche di utenti | | |
|-----------------------|--|--|---|
| Goal in Context | L'utente vuole trovare dei utenti | | |
| Preconditions | L'utente è autenticato ed esistono altri utenti registrati nel sistema | | |
| Success End Condition | L'utente ha trovato utenti ricercati | | |
| Failer End Condition | L'utente non ha trovato utenti ricercati | | |
| Primary Actor | Utente loggato | | |
| Trigger | Utente preme il pulsante "Aggiungi nuovo amico" nella schermata Amici | | |
| DESCRIPTION | STEP | UTENTE LOGGATO | SISTEMA |
| | 1 | Preme il pulsante "Aggiungi nuovo amico" nella schermata Amici AND_04 | |
| | 2 | | Mostra la schermata di ricerca prima di query AND_MK15 |
| | 3 | Inserire username di ricerca e preme invio | |
| | 4 | | Mostra la schermata del risultato di ricerca AND_MK16 |



1.1.5.10 – Inviare richieste di amicizia

| USER CASE #10 | | | | Inviare richieste di amicizia |
|------------------------------|-------------|---|--|--|
| Goal in Context | | | | L'utente vuole inviare una richiesta di amicizia ad un utente |
| Preconditions | | | | L'utente è autentificato |
| Success End Condition | | | | L'utente riesce ad inviare una richiesta di collegamento ad un altro utente |
| Failer End Condition | | | | Nessuna |
| Primary Actor | | | | Utente loggato |
| Trigger | | | | Utente preme il simbolo "Aggiungi come amico" nella scheda utente o nell'elenco utenti ricercati |
| DESCRIPTION | STEP | UTENTE LOGGATO | | SISTEMA |
| | 1 | preme il simbolo "Aggiungi come amico" nella scheda utente AND_MK17 o nell'elenco utenti ricercati AND_MK16 | | |
| | 2 | | | Mostra il messaggio di conferma invio amicizia AND_MSG16 |



1.1.5.11 – Accettare richiesta amicizia

| USER CASE #11 | | | | Accettare richiesta amicizia |
|------------------------------|-------------|---|--|---|
| Goal in Context | | | | L'utente vuole accettare una richiesta di amicizia |
| Preconditions | | | | L'utente è autentificato |
| Success End Condition | | | | L'utente riesce ad accettare una richiesta di amicizia |
| Failer End Condition | | | | Nessuna |
| Primary Actor | | | | Utente loggato |
| Trigger | | | | Utente preme il simbolo "Accetta" nella schermata amicizie o nella schermata tutte richieste amicizie |
| DESCRIPTION | STEP | UTENTE LOGGATO | | SISTEMA |
| | 1 | preme il simbolo "Accetta" nella schermata amicizie AND_MK04 o nella schermata tutte richieste amicizie AND_MK14 o nella schermata scheda di un altro utente AND_MK19 | | |
| | 2 | | | Mostra il messaggio di conferma amicizia accettata AND_MSG17 |



1.1.5.12 – Rifiutare richiesta amicizia

| USER CASE #12 | | | | Rifiutare richiesta amicizia |
|------------------------------|-------------|---|--|---|
| Goal in Context | | | | L'utente vuole rifiutare una richiesta di amicizia |
| Preconditions | | | | L'utente è autentificato |
| Success End Condition | | | | L'utente riesce a rifiutare una richiesta di amicizia |
| Failer End Condition | | | | Nessuna |
| Primary Actor | | | | Utente loggato |
| Trigger | | | | Utente preme il simbolo "Rifiuta" nella schermata amicizie o nella schermata tutte richieste amicizie |
| DESCRIPTION | STEP | UTENTE LOGGATO | | SISTEMA |
| | 1 | preme il simbolo "Rifiuta" nella schermata amicizie AND_MK04 o nella schermata tutte richieste amicizie AND_MK14 o nella schermata scheda di un altro utente AND_MK19 | | |
| | 2 | | | Mostra il messaggio di conferma amicizia rifiutata AND_MSG18 |



1.1.5.13 – Visualizzazione della scheda di un film

| USER CASE #13 | Visualizzazione della scheda di un film | | |
|---|--|---|--|
| Goal in Context | L'utente vuole una scheda di un film | | |
| Preconditions | L'utente è autentificato e esiste un film nel database | | |
| Success End Condition | L'utente visualizza la scheda di un film | | |
| Failer End Condition | Nessuna | | |
| Primary Actor | Utente loggato | | |
| Trigger | Utente clicca sull'icona o titolo del film nella schermata di Ricerca o dentro una lista | | |
| DESCRIPTION | STEP | UTENTE LOGGATO | SISTEMA |
| EXTENSION #1 Utente preme pulsante "Back" | 1 | Clicca sull'icona o sul titolo del film nella schermata di Ricerca AND_MK12 o dentro una lista AND_MK08, AND_21 | |
| | 2 | | Mostra la schermata di un film AND_MK13 |
| EXTENSION #1 Utente preme pulsante "Back" | STEP | UTENTE LOGGATO | SISTEMA |
| | 1a | Utente preme il pulsante "Back" | |
| | 2a | | Mostra la schermata precedente. Termina caso d'uso |



1.1.5.14 – Visualizzazione della scheda utente

| USER CASE #14 | | | | Visualizzazione della scheda utente |
|------------------------------|-------------|---|----------------|---|
| Goal in Context | | | | L'utente vuole visualizzare la scheda di un utente |
| Preconditions | | | | L'utente è autenticato e l'utente da visualizzare è registrato sulla piattaforma |
| Success End Condition | | | | L'utente riesce a visualizzare la scheda di un utente |
| Failer End Condition | | | | Nessuna |
| Primary Actor | | | | Utente loggato |
| Trigger | | | | Utente seleziona un utente dalle schermate: schermata amicizie, schermata utenti ricercati e schermata tutte richieste amicizia |
| DESCRIPTION | STEP | UTENTE LOGGATO | SISTEMA | |
| | 1 | Seleziona un utente dalle schermate: schermata amicizie AND_MK04, schermata utenti ricercati AND_MK16 e schermata tutte richieste amicizia AND_MK14 | | |
| | 2 | | | Mostra la schermata scheda utente AND_MK17, AND_MK18, AND_MK19 |



1.1.5.15 – Visualizzazione di una lista

| USER CASE #15 | Visualizzazione di una lista | | |
|-----------------------|---|--|---|
| Goal in Context | L'utente vuole visualizzare una lista | | |
| Preconditions | L'utente è autenticato e ci sono delle liste create | | |
| Success End Condition | L'utente visualizza una lista | | |
| Failer End Condition | Nessuna | | |
| Primary Actor | Utente loggato | | |
| Trigger | Utente clicca sull'icona o sul titolo della lista nella schermata di Visualizzazione tutte le liste o sull'Icona della lista nella schermata Home | | |
| DESCRIPTION | STEP | UTENTE LOGGATO | SISTEMA |
| | 1 | Clicca sull'icona o sul titolo della lista nella schermata di Visualizzazione tutte le liste AND_MK07, AND_MK22 o sull'Icona della lista nella schermata Home AND_MK03 | |
| | 2 | | Mostra il contenuto di una lista AND_MK08 o AND_MK_21 |



1.1.5.16 – Visualizzazione dei film in comune con amico

| USER CASE #16 | | | | Visualizzazione dei film in comune con amico |
|------------------------------|-------------|---|----------------|--|
| Goal in Context | | | | Visualizzazione film in comune con amico |
| Preconditions | | | | L'utente è autentificato |
| Success End Condition | | | | L'utente riesce a visualizzare film in comune con amico |
| Failer End Condition | | | | Nessuno |
| Primary Actor | | | | Utente loggato |
| Trigger | | | | Utente preme su film in comune nella schermata scheda utente amico |
| DESCRIPTION | STEP | UTENTE LOGGATO | SISTEMA | |
| | 1 | Preme film in comune nella schermata scheda utente amico AND_MK18 | | |
| | 2 | | | Mostra elenco film in comune AND_MK23 |



1.1.5.17 – Visualizzazione delle notifiche

| USER CASE #17 | | | | Visualizzazione delle notifiche |
|------------------------------|-------------|--|----------------|---|
| Goal in Context | | | | L'utente vuole visualizzare le notifiche |
| Preconditions | | | | L'utente è autentificato |
| Success End Condition | | | | L'utente visualizza le proprie notifiche |
| Failer End Condition | | | | Nessuna |
| Primary Actor | | | | Utente loggato |
| Trigger | | | | Utente preme l'icona delle notifiche nella schermata Home |
| DESCRIPTION | STEP | UTENTE LOGGATO | SISTEMA | |
| | 1 | Utente preme il pulsante "Notifiche" nella schermata Home AND_MK03 | | |
| | 2 | | | Mostra la schermata delle notifiche AND_MK09 |



1.1.5.18 – Effettuare una segnalazione

| USER CASE #18 | | | | Effettuare una segnalazione |
|--|-------------|--|----------------|---|
| Goal in Context | | | | L'utente vuole segnalare una lista |
| Preconditions | | | | L'utente è autentificato |
| Success End Condition | | | | L'utente segnalare una lista |
| Failer End Condition | | | | Nessuna |
| Primary Actor | | | | Utente loggato |
| Trigger | | | | Utente nella visualizzazione della lista di un altro utente preme il pulsante " ! " |
| DESCRIPTION | STEP | UTENTE LOGGATO | SISTEMA | |
| EXTENTIONS #1 Utente preme il pulsante "Annulla" | 1 | Nella visualizzazione di una lista di un altro utente AND_MK21 preme il pulsante " ! " | | |
| | 2 | | | Mostra la schermata della segnalazione AND_MK36 |
| | 3 | Sceglie il motivo della segnalazione e preme il pulsante "Fatto" | | |
| | 4 | | | Mostra il messaggio della segnalazione effettuata AND_MSG09 |
| EXTENTIONS #1 Utente preme il pulsante "Annulla" | STEP | UTENTE LOGGATO | SISTEMA | |
| | 3a | Preme il pulsante "Annulla" | | |
| | 4a | | | Mostra la schermata della visualizzazione di una lista di un altro utente AND_MK21. Termina caso d'uso |



1.1.5.19 – Cambiare la password

| USER CASE #19 | | | | Cambiare la Password |
|--|-------------|--|--|--|
| Goal in Context | | | | L'utente vuole cambiare la password |
| Preconditions | | | | L'utente è autentificato |
| Success End Condition | | | | L'utente cambia la password |
| Failer End Condition | | | | L'utente non riesce cambiare la password |
| Primary Actor | | | | Utente loggato |
| Trigger | | | | Utente preme il pulsante "cambia password" nella schermata del Profilo |
| DESCRIPTION | STEP | UTENTE LOGGATO | SISTEMA | |
| EXTENTIONS #1 Utente inserisce nuova password che non rispetta i requisiti | 1 | Preme "cambia password" nella schermata del Profilo AND_MK06 | | |
| | 2 | | Mostra la schermata del cambiamento della password AND_MK37 | |
| | 3 | Inserisce nuova password e preme il pulsante "Salva" | | |
| | 4 | | Mostra il messaggio di successo cambiamento della password AND_MSG12 | |
| EXTENTIONS #1 Utente inserisce nuova password che non rispetta i requisiti | STEP | UTENTE LOGGATO | SISTEMA | |
| | 4b | | Mostra messaggio di errore AND_MSG13 | |
| | | | | |



1.1.5.20 – Cambiare immagine di profilo

| USER CASE #20 | | Cambiare immagine di profilo | |
|------------------------------|-------------|--|--|
| Goal in Context | | L'utente vuole cambiare immagine di profilo | |
| Preconditions | | L'utente è autentificato | |
| Success End Condition | | L'utente cambia immagine di profilo | |
| Failer End Condition | | Nessuna | |
| Primary Actor | | Utente loggato | |
| Trigger | | Utente preme "cambia immagine" nella schermata <Profilo> | |
| DESCRIPTION | STEP | UTENTE LOGGATO | SISTEMA |
| | 1 | Preme "cambia immagine" nella schermata AND_MK06 | |
| | 2 | | Mostra la schermata selezione immagine (gestito dal sistema Android) |
| | 3 | Sceglie immagine locale | |
| | 4 | | Mostra messaggio di conferma AND_MSG14 |



1.1.5.21 – Uscire dal profilo

| USER CASE #21 | | | | Uscire dal profilo |
|------------------------------|-------------|---|----------------|--|
| Goal in Context | | | | L'utente vuole uscire dal profilo |
| Preconditions | | | | L'utente è autentificato |
| Success End Condition | | | | L'utente esce dal profilo |
| Failer End Condition | | | | Nessuna |
| Primary Actor | | | | Utente loggato |
| Trigger | | | | Utente preme il pulsante "Logout" nella schermata Menu premuto |
| DESCRIPTION | STEP | UTENTE LOGGATO | SISTEMA | |
| | 1 | Utente preme il pulsante "Logout" nella schermata Menu premuto AND_MK05 | | |
| | 2 | | | Mostra la schermata di Login AND_MK01 |



1.1.5.22 – Modificare la descrizione di una lista

| USER CASE #22 | | | | Modificare la descrizione di una lista |
|---|-------------|--|---|--|
| Goal in Context | | | | L'utente vuole modificare la descrizione di una lista |
| Preconditions | | | | L'utente è autentificato ed ha una lista personalizzare |
| Success End Condition | | | | L'utente modifica la descrizione di una lista |
| Failer End Condition | | | | Nessuna |
| Primary Actor | | | | Utente loggato |
| Trigger | | | | Utente nella visualizzazione di una lista preme pulsate modifica a fianco alla descrizione |
| DESCRIPTION | STEP | UTENTE LOGGATO | SISTEMA | |
| | 1 | Nella visualizzazione di una lista AND_MK08 preme il pulsante "Modifica" a fianco alla descrizione | | |
| | 2 | | Mostra la schermata della modifica della descrizione AND_MK24 | |
| | 3 | Cambia la descrizione e preme "Ok" | | |
| | 4 | | Mostra la schermata AND_MK08 | |
| EXTENTION #1 Utente preme "Annulla" | STEP | UTENTE LOGGATO | SISTEMA | |
| | 3a | Preme tasto annulla nella schermata modifica della descrizione | | |
| | 4a | | Ritorna allo step 4 dello scenario principale | |



1.1.5.23 – Visualizzazione del proprio profilo

| USER CASE #23 | Visualizzazione del proprio profilo | | |
|------------------------------|---|---|--|
| Goal in Context | L'utente vuole visualizzare proprio profilo | | |
| Preconditions | L'utente è autentificato | | |
| Success End Condition | L'utente visualizza proprio profilo | | |
| Failer End Condition | Nessuna | | |
| Primary Actor | Utente loggato | | |
| Trigger | Utente preme il pulsante Profilo nella schermata Menu premuto | | |
| DESCRIPTION | STEP | UTENTE LOGGATO | SISTEMA |
| | 1 | Preme il pulsante "Profilo" nella schermata Menu premuto AND_MK05 | |
| | 2 | | Mostra la schermata del profilo AND_MK06 |



1.1.6 Mock-up (Android)

Di seguito vengono i mostrati Mock-up principali con la descrizione dei componenti in ogni schermata.

Nei seguenti Mock-up sono state adottate le seguenti convenzioni:

- La freccia indietro riportata in figura rappresenta il “back” di android, la quale riporterà alla schermata precedente. Essendo una funzionalità di Android non ne è stato specificato il comportamento nella descrizione dei mock-up.



- I pulsanti presenti nella barra inferiore rappresentanti in figura, nonostante questi ultimi vengono riportati in numerose schermate sono stati descritti solo nella schermata MK_03 per comodità e per evitare ridondanze.



1.1.6.1 – Schermata: Accesso al sistema

| AND_MK01 | |
|----------|---|
| | Componente 1 |
| | <p>Tipo Pulsante</p> <p>Funzionalità Apre la schermata accesso di Google</p> |
| | Componente 2 |
| | <p>Tipo Pulsante</p> <p>Funzionalità Apre la schermata accesso di Facebook</p> |
| | Componente 3 |
| | <p>Tipo Campo testuale</p> <p>Funzionalità Campo per l'inserimento del username</p> |
| | Componente 4 |
| | <p>Tipo Campo testuale</p> <p>Funzionalità Campo per l'inserimento della password</p> |
| | Componente 5 |
| | <p>Tipo Pulsante</p> <p>Funzionalità Apre la schermata AND_MK33</p> |
| | Componente 6 |
| | <p>Tipo Pulsante</p> <p>Funzionalità Avvia il tentativo di accesso, mostra AND_MK03 nel caso di successo e AND_MSG08 altrimenti</p> |
| | Componente 7 |
| | <p>Tipo Pulsante</p> <p>Funzionalità Mostra schermata registrazione AND_MK02</p> |



1.1.6.2 – Schermata: Registrazione al sistema

| AND_MK02 | |
|----------|---|
| | Componente 1 |
| | Tipo Campo testuale Funzionalità Campo per l'inserimento dello username |
| | Componente 2 |
| | Tipo Campo testuale Funzionalità Campo per l'inserimento della password |
| | Componente 3 |
| | Tipo Campo testuale Funzionalità Campo per l'inserimento della conferma password |
| | Componente 4 |
| | Tipo Campo testuale Funzionalità Campo per l'inserimento della e-mail |
| | Componente 5 |
| | Tipo Pulsante Funzionalità Pulsante che avvia la registrazione al sistema |
| | Componente 6 |
| | Tipo Pulsante Funzionalità Pulsante che annulla la registrazione al sistema |

The screenshot shows a mobile application interface for registration. At the top is a logo of a movie camera with film reels. Below it is the text 'CineMates20'. There are four input fields labeled 'Username' (1), 'Password' (2), 'Conferma password' (3), and 'E-mail' (4). At the bottom are two buttons: 'Annulla' (6) on the left and 'Registrati' (5) on the right.



1.1.6.3 – Schermata: Home

| AND_MK03 | |
|----------|--|
| | Componente 1 |
| | <p>Tipo Pulsante</p> <p>Funzionalità Apre Menu laterale AND_MK05</p> |
| | Componente 2 |
| | <p>Tipo Pulsante</p> <p>Funzionalità Apre la schermata Notifiche AND_MK09</p> |
| | Componente 3 - 6 |
| | <p>Tipo Pulsante</p> <p>Funzionalità Apre la schermata Visualizzazione lista AND_MK08</p> |
| | Componente 4 |
| | <p>Tipo Pulsante</p> <p>Funzionalità Apre la schermata Visualizzazione film AND_MK13</p> |
| | Componente 5 |
| | <p>Tipo Pulsante</p> <p>Funzionalità Apre la schermata Visualizzazione le liste AND_MK07</p> |
| | Componente 7 |
| | <p>Tipo Pulsante</p> <p>Funzionalità Apre menu creazione lista AND_MK26</p> |
| | Componente 8 |
| | <p>Tipo Pulsante</p> <p>Funzionalità Mostra schermata Home AND_MK03</p> |
| | Componente 9 |
| | <p>Tipo Pulsante</p> <p>Funzionalità Mostra schermata Ricerca AND_MK10</p> |
| | Componente 10 |
| | <p>Tipo Pulsante</p> <p>Funzionalità Mostra schermata Amici AND_MK04</p> |



1.1.6.4 – Schermata: Amici

| AND_MK04 | | |
|---------------------|---|---|
| | Componente 1 | |
| | Tipo | Pulsante |
| | Funzionalità | Apre la schermata di tutte le richieste AND_MK14 |
| | Componente 2 | |
| | Tipo | Pulsante |
| | Funzionalità | Accetta la richiesta di amicizia, mostra il messaggio AND_MSG17 |
| | Componente 3 | |
| Tipo | Pulsante | |
| Funzionalità | Rifiuta la richiesta di amicizia, mostra il messaggio AND_MSG18 | |
| Componente 4 | | |
| Tipo | Pulsante | |
| Funzionalità | Effettua la ricerca del nuovo amico, apre AND_MK15 | |
| Componente 5 | | |
| Tipo | Campo di ricerca | |
| Funzionalità | Effettua la ricerca tra gli amici | |
| Componente 6 | | |
| Tipo | Pulsante | |
| Funzionalità | Apre la schermata dell'utente AND_MK18 | |
| Componente 7 | | |
| Tipo | Pulsante | |
| Funzionalità | Rimuove utente dai amici | |



1.1.6.5 – Schermata: Menu laterale premuto

| AND_MK05 | |
|----------|---|
| | Componente 1 |
| | Tipo Pulsante |
| | Funzionalità Apre la schermata Profilo AND_MK06 |
| | Componente 2 |
| | Tipo Pulsante |
| | Funzionalità Apre la schermata informativa |
| | Componente 3 |
| | Tipo Pulsante |
| | Funzionalità Effettua logout, apre la schermata AND_MK01 |

A smartphone screen displaying a user profile interface. At the top left is a user icon and the text "NomeUtente". To the right is a bell icon. Below this, there are two main menu items: "Profilo" (with a person icon) and "Info" (with a question mark icon). Both of these items have a small blue circle with the number "1" in the bottom right corner. At the bottom of the screen, there is a "Logout" button with a cross icon and a small blue circle with the number "3" in the bottom right corner. The background of the phone screen shows some blurred text and icons.

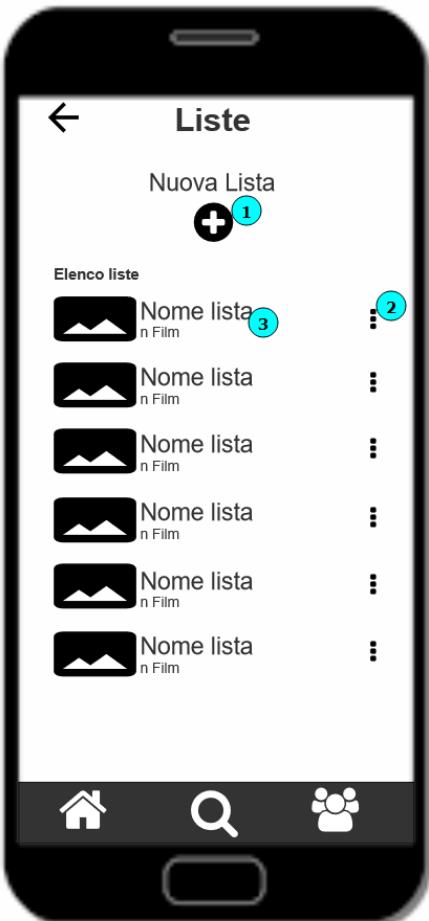
1.1.6.6 – Schermata: Profilo

| AND_MK06 | |
|----------|--|
| | Componente 1 |
| | Tipo Pulsante Funzionalità Apre la schermata modifica immagine AND_MK38 |
| | Componente 2 |
| | Tipo Pulsante Funzionalità Visualizza la password |
| | Componente 3 |
| | Tipo Pulsante Funzionalità Apre la schermata modifica password AND_MK37 |



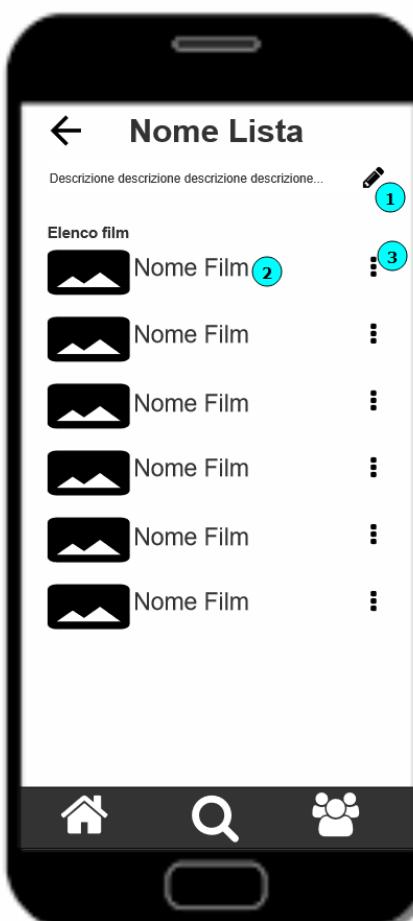

1.1.6.7 – Schermata: Tutte le liste

| AND_MK07 | |
|---------------------|--|
| | Componente 1 |
| Tipo | Pulsante |
| Funzionalità | Apre la schermata creazione nuova lista AND_MK26 |
| | Componente 2 |
| Tipo | Pulsante |
| Funzionalità | Apre menu a tendina AND_MK31 |
| | Componente 3 |
| Tipo | Pulsante |
| Funzionalità | Apre la schermata contenuto lista AND_MK08 |




1.1.6.8 – Schermata: Contenuto lista

| AND_MK08 | |
|---------------------|---|
| | Componente 1 |
| Tipo | Pulsante |
| Funzionalità | Apre la schermata modifica descrizione AND_MK24 |
| | Componente 2 |
| Tipo | Pulsante |
| Funzionalità | Apre la schermata scheda film AND_MK13 |
| | Componente 3 |
| Tipo | Pulsante |
| Funzionalità | Apre menu a tendina AND_MK28 |



The smartphone screen displays a list titled "Nome Lista" with the following details:

- Header: "Nome Lista" with a back arrow icon.
- Text input field: "Descrizione descrizione descrizione descrizione..." with a pencil icon.
- Section: "Elenco film" with a camera icon.
- List items (6 rows):
 - Icon: Camera icon. Text: "Nome Film" (with a circled "2"). Action: Edit icon (circled "1")
 - Icon: Camera icon. Text: "Nome Film". Action: More options icon (circled "3")
 - Icon: Camera icon. Text: "Nome Film". Action: More options icon
 - Icon: Camera icon. Text: "Nome Film". Action: More options icon
 - Icon: Camera icon. Text: "Nome Film". Action: More options icon
 - Icon: Camera icon. Text: "Nome Film". Action: More options icon
- Bottom navigation bar: Home, Search, and User icons.



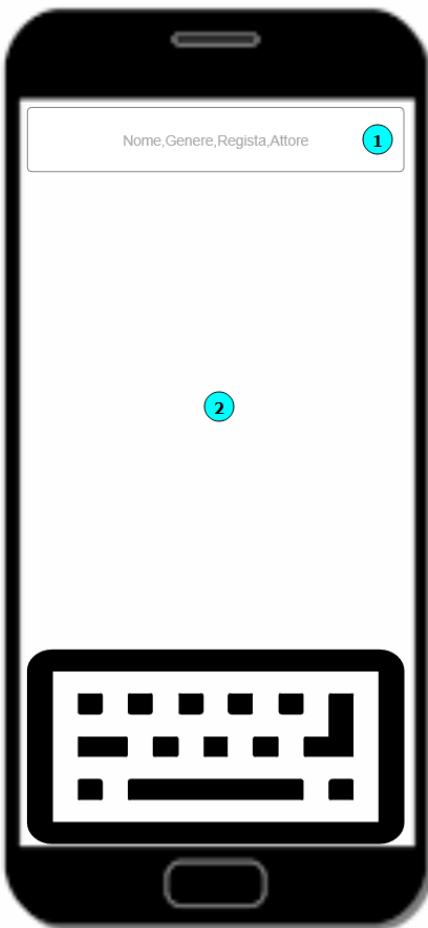
1.1.6.9 – Schermata: Notifiche

| AND_MK09 | |
|---|--------------------------------------|
| Componente 1 | |
| Tipos | Campo testuale |
| Funzionalità | Contiene il messaggio della notifica |
|  <p>The smartphone screen displays a list of notifications under the heading "Notifiche". The notifications are:</p> <ul style="list-style-type: none">Tizio ti ha mandato 1 richiesta di amiciziaLa tua segnalazione sul commento dell'utente <Nome Utente> ha portato alla cancellazione del...La tua segnalazione sul commento dell'utente <Nome Utente> ha portato alla cancellazione del...La tua segnalazione sul commento dell'utente <Nome Utente> è stata respinta. <p>The bottom navigation bar includes icons for Home, Search, and Groups.</p> | |



1.1.6.10 – Schermata: Ricerca Film

| AND_MK10 | |
|----------|---|
| | Componente 1 |
| | Tipo Campo testuale |
| | Funzionalità Campo di ricerca di film nel caso di successo mostra schermata AND_MK12 |
| | Componente 2 |
| | Tipo Campo cliccabile |
| | Funzionalità Esce dalla modalità di inserimento |

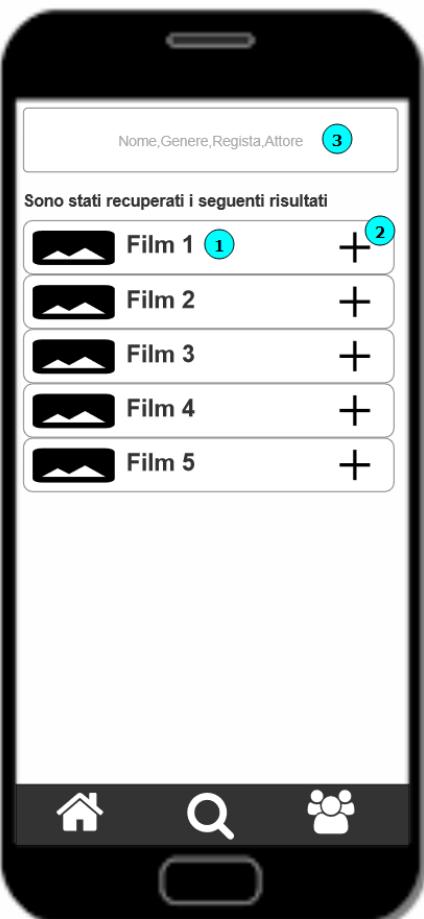


The diagram shows a smartphone interface. At the top, there is a search bar containing the placeholder text "Nome,Genere,Regista,Attore". To the right of the search bar is a blue circular button with the number "1". Below the search bar is a virtual keyboard. Two small cyan circles, labeled "1" and "2", are overlaid on the screen: circle "1" is positioned near the top right corner of the search bar area, and circle "2" is positioned below the keyboard.



1.1.6.11 – Schermata: Risultato Ricerca Film

| AND_MK12 | |
|---------------------|--|
| | Componente 1 |
| Tipo | Pulsante |
| Funzionalità | Apre la schermata scheda film AND_MK13 |
| Componente 2 | |
| Tipo | Pulsante |
| Funzionalità | Apre menu a tendina AND_MK25 |
| Componente 3 | |
| Tipo | Campo testuale |
| Funzionalità | Campo di ricerca di film |




1.1.6.12 – Schermata: Scheda Film

| AND_MK13 | |
|---------------------|---------------------------------------|
| | Componente 1 |
| Tipo | Pulsante |
| Funzionalità | Apre menu creazione lista AND_MK26 |
| | Componente 2 |
| Tipo | Pulsante |
| Funzionalità | Apre menu a tendina AND_MK32 |
| | Componente 3 |
| Tipo | Campo testuale |
| Funzionalità | Campo di ricerca di film |



The smartphone screen displays a movie card. At the top is a blue mountain icon with a sun. Below it is a title input field labeled "Titolo" with a placeholder "Descrizione". Underneath the title is a section with "Durata : hh:mm", "Genere : tipo", and "Valutazione: X.X". A blue plus sign button with a "1" badge is located below this section. At the bottom is a navigation bar with icons for Home, Search, and User.



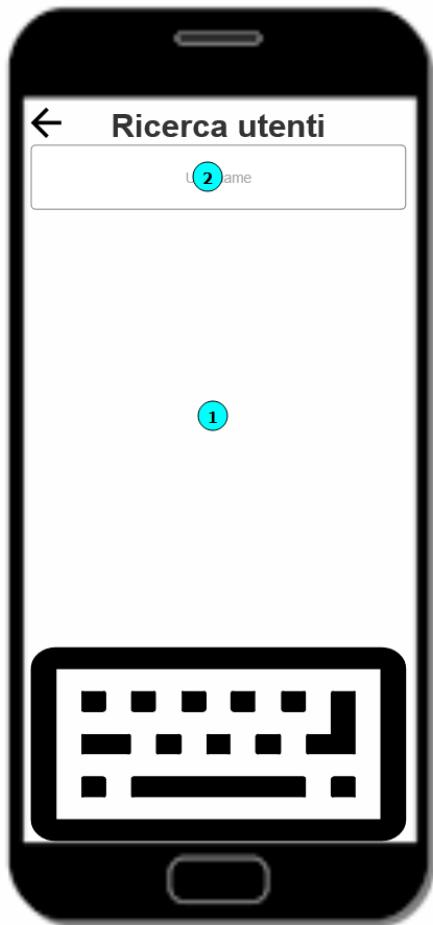
1.1.6.13 – Schermata: Tutte le richieste di Amicizia

| AND_MK14 | |
|----------|--|
| | Componente 1 |
| | <p>Tipo Pulsante</p> <p>Funzionalità Accetta la richiesta di amicizia, mostra il messaggio AND_MSG17</p> |
| | Componente 2 |
| | <p>Tipo Pulsante</p> <p>Funzionalità Rifiuta la richiesta di amicizia, mostra il messaggio AND_MSG18</p> |
| | Componente 3 |
| | <p>Tipo Pulsante</p> <p>Funzionalità Mostra la schermata Profilo altro utente AND_MK19</p> |




1.1.6.14 – Schermata: cerca utenti

| AND_MK15 | |
|---------------------|---|
| | Componente 1 |
| Tipo | Campo cliccabile |
| Funzionalità | Esce dalla modalità di inserimento |
| | Componente 2 |
| Tipo | Campo testuale |
| Funzionalità | Campo della ricerca degli utenti, nel caso di successo mostra la schermata AND_MK16 |

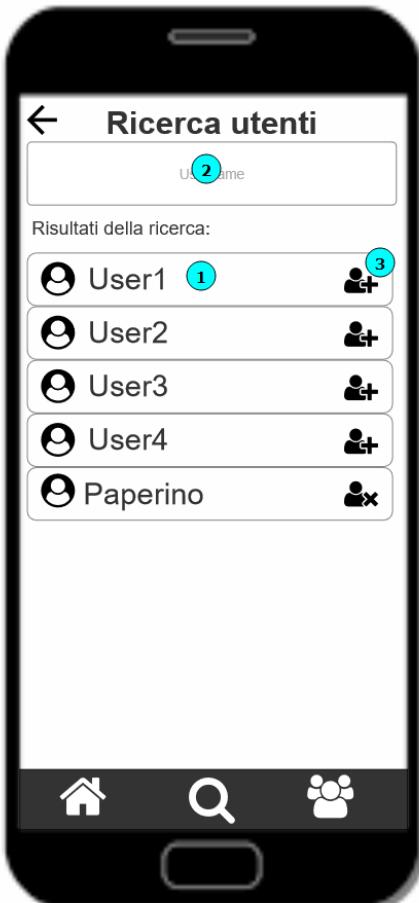


The image shows a smartphone screen displaying a search interface. The title bar says "Ricerca utenti". Below it is a text input field with the placeholder "Nome" and a blue circular icon containing the number "2". At the bottom of the screen is a virtual keyboard.



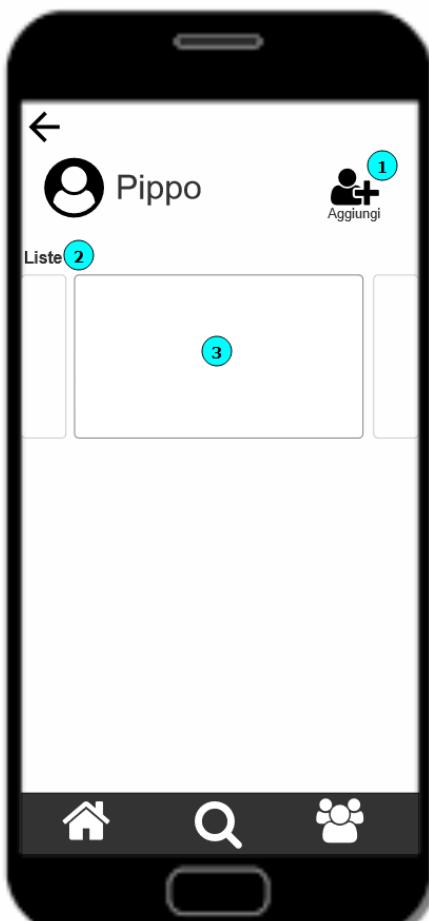
1.1.6.15 – Schermata: Ricerca utenti dopo Query

| AND_MK16 | |
|--------------|---|
| | Componente 1 |
| | <p>Tipo Pulsante</p> <p>Funzionalità Mostra la schermata Profilo altro utente AND_MK17</p> |
| | Componente 2 |
| | <p>Tipo Campo testuale</p> <p>Funzionalità Campo della ricerca degli utenti</p> |
| Componente 3 | |
| | <p>Tipo Pulsante</p> <p>Funzionalità Invia richiesta di amicizia, mostra il messaggio AND_MSG16</p> |




1.1.6.16 – Schermata: Profilo altro utente non Amico

| AND_MK17 | |
|----------|---|
| | Componente 1 |
| | <p>Tipo Pulsante</p> <p>Funzionalità Invia richiesta di amicizia, mostra il messaggio AND_MSG16</p> |
| | Componente 2 |
| | <p>Tipo Pulsante</p> <p>Funzionalità Mostra la schermata le liste di altri utenti AND_MK22</p> |
| | Componente 3 |
| | <p>Tipo Pulsante</p> <p>Funzionalità Mostra la schermata lista di altri utenti AND_MK21</p> |

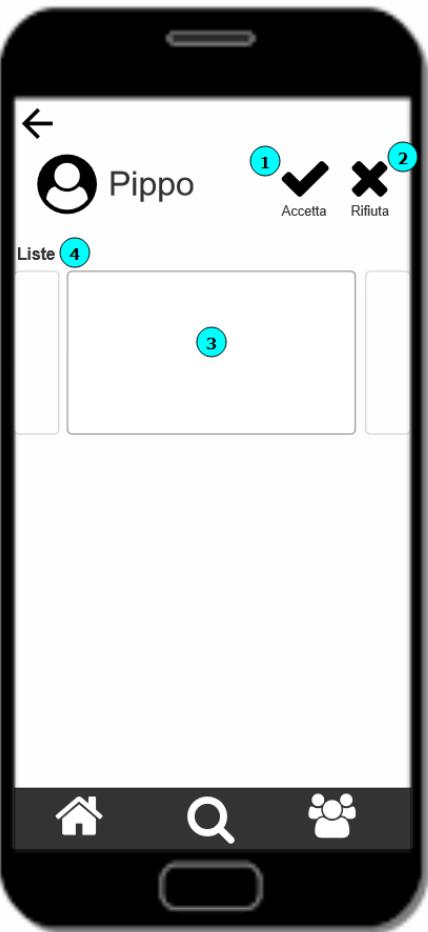



1.1.6.17 – Schermata: Profilo altro utente Amico

| AND_MK18 | |
|----------|---|
| | Componente 1 |
| | <p>Tipo Pulsante</p> <p>Funzionalità Rimuove utente dagli amici</p> |
| | Componente 2 |
| | <p>Tipo Pulsante</p> <p>Funzionalità Mostra la schermata scheda film AND_MK23</p> |
| | Componente 3 |
| | <p>Tipo Pulsante</p> <p>Funzionalità Mostra la schermata film in comune AND_MK12</p> |
| | Componente 4 |
| | <p>Tipo Pulsante</p> <p>Funzionalità Mostra la schermata liste di altri utenti AND_MK22</p> |
| | Componente 5 |
| | <p>Tipo Pulsante</p> <p>Funzionalità Mostra la schermata lista di altri utenti AND_MK21</p> |

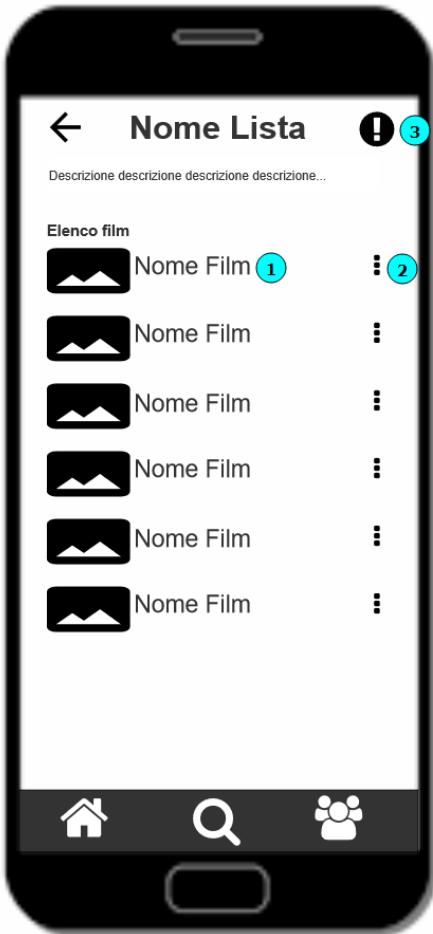


1.1.6.18 – Schermata: Profilo utente che ha richiesto amicizia

| AND_MK19 | |
|--|---|
| | Componente 1 |
| | <p>Tipo Pulsante</p> <p>Funzionalità Accetta richiesta, mostra il messaggio AND_MSG17</p> |
| | Componente 2 |
| | <p>Tipo Pulsante</p> <p>Funzionalità Rifiuta richiesta, mostra il messaggio AND_MSG18</p> |
| | Componente 3 |
| | <p>Tipo Pulsante</p> <p>Funzionalità Mostra la schermata lista di altri utenti AND_MK21</p> |
| | Componente 4 |
| | <p>Tipo Pulsante</p> <p>Funzionalità Mostra la schermata liste di altri utenti AND_MK22</p> |
|  | |



1.1.6.19 – Schermata: Lista di altro utente

| AND_MK21 | |
|--|---|
| | Componente 1 |
| Tipo | Pulsante |
| Funzionalità | Mostra la schermata scheda Film AND_MK13 |
| | Componente 2 |
| Tipo | Pulsante |
| Funzionalità | Menu a tendina AND_MK32 |
| | Componente 3 |
| Tipo | Pulsante |
| Funzionalità | Effettua segnalazione della lista, apre la schermata AND_MK36 |
| | |
|  | |

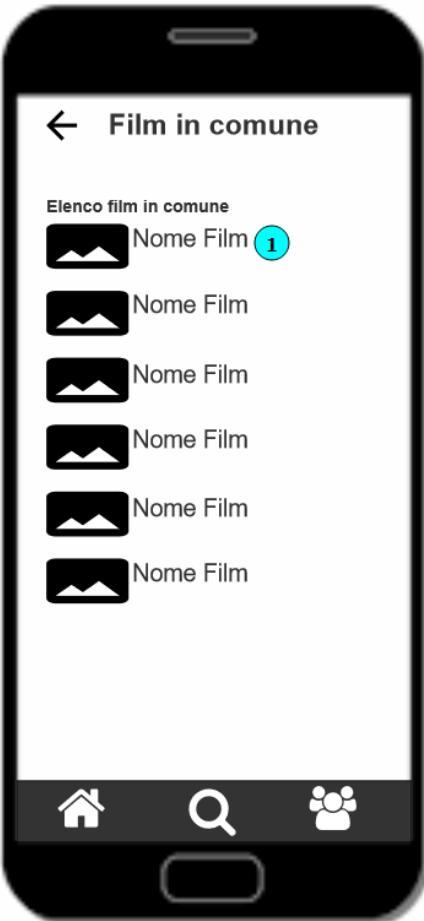


1.1.6.20 – Schermata: Le liste di un altro utente

| AND_MK022 | |
|--------------|--|
| Componente 1 | |
| Tipo | Pulsante |
| Funzionalità | Apre la schermata Lista di un altro utente AND_MK21 |
| | |



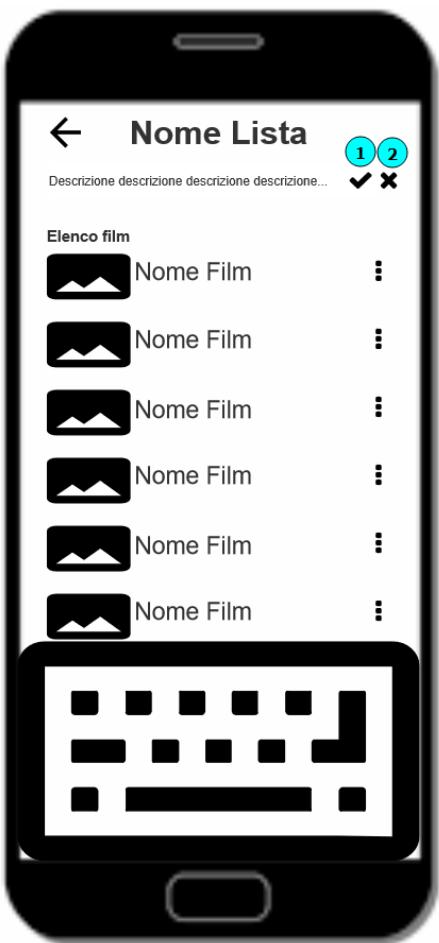
1.1.6.21 – Schermata: Lista Film in comune con un Amico

| AND_MK23 | |
|--|---|
| Componente 1 | |
| Tipo | Pulsante |
| Funzionalità | Apre la schermata scheda Film AND_MK13 |
|  | |



1.1.6.22 – Schermata: Modifica descrizione lista

| AND_MK24 | |
|---------------------|--------------------------------|
| | Componente 1 |
| Tipo | Pulsante |
| Funzionalità | Accetta la modifica effettuata |
| | Componente 2 |
| Tipo | Pulsante |
| Funzionalità | Rifiuta la modifica effettuata |



The smartphone screen displays a list titled "Nome Lista" with six items, each showing a movie icon and the text "Nome Film". Above the list are two circular buttons labeled 1 and 2, with a checkmark and an X respectively. Below the list is a virtual keyboard.



1.1.6.23 – Schermata: Salva film in una lista

| AND_MK25 | |
|---|--|
| Salva in... + Crea nuova lista 1 | Componente 1 |
| ■ Preferiti 4 | Tipo Pulsante |
| ■ Da guardare dopo | Funzionalità Apre la schermata creazione lista AND_MK26 |
| ■ Lista 1 | Componente 2 |
| ■ Lista 2 | Tipo Pulsante |
| ■ Lista 3 | Funzionalità Annulla operazione, ritorna alla schermata di partenza |
| ■ Nome Lista | Componente 3 |
| | Tipo Pulsante |
| | Funzionalità Salva le modifiche, ritorna alla schermata di partenza |
| | Componente 4 |
| | Tipo Casella selezione |
| | Funzionalità Aggiunge/Rimuove un film dalla particolare lista |

1.1.6.24 – Schermata: Crea nuova lista

| AND_MK26 | |
|--|--|
| Nuova lista | Componente 1 |
| Titolo 1 | Tipo Campo testuale |
| Aggiungi una descrizione (opzionale) | Funzionalità Campo riservato al titolo della Lista |
| Descrizione 2 | Componente 2 |
| | Tipo Campo testuale |
| | Funzionalità Campo riservato alla descrizione della nuova lista |
| | Componente 3 |
| | Tipo Pulsante |
| | Funzionalità Annulla la creazione della nuova lista |
| | Componente 4 |
| | Tipo Pulsante |
| | Funzionalità Crea nuova lista, in caso di successo mostra messaggio AND_MSG04 |



1.1.6.25 – Schermata: Menu a tendina film

| AND_MK28 | |
|----------|--|
| | Componente 1 |
| | Tipo Pulsante Funzionalità Apre menu AND_MK25 |
| | Componente 2 |
| | Tipo Pulsante Funzionalità Rimuove film dalla lista, mostra messaggio AND_MSG07 |

1.1.6.26 – Schermata: Opzioni lista propria

| AND_MK31 | |
|----------|---|
| | Componente 1 |
| | Tipo Pulsante Funzionalità Cancella lista interamente, mostra il messaggio AND_MSG06 |

1.1.6.27 – Schermata: Opzioni lista dell’altro utente

| AND_MK32 | |
|----------|---|
| | Componente 1 |
| | Tipo Pulsante Funzionalità Apre la schermata Salva film in una lista AND_MK25 |



1.1.6.28 – Schermata: Recupero password passo Inserire E-mail

| AND_MK33 | |
|--------------|---|
| | Componente 1 |
| | Tipo Campo testuale Funzionalità Campo riservato all'email dell'account da recuperare |
| | Componente 2 |
| | Tipo Pulsante Funzionalità Annulla operazione |
| Componente 3 | |
| | Tipo Pulsante Funzionalità Nel caso di successo (Email rispetta i requisiti) invia il codice di recupero all'e-mail inserita |

1.1.6.29 – Schermata: Recupero password passo conferma Codice

| AND_MK34 | |
|--------------|---|
| | Componente 1 |
| | Tipo Campo testuale Funzionalità Campo riservato al codice inviato sull'e-mail |
| Componente 2 | |
| | Tipo Campo testuale Funzionalità Campo riservato alla nuova password |
| Componente 3 | |
| | Tipo Pulsante Funzionalità Annulla operazione |
| Componente 4 | |
| | Tipo Pulsante Funzionalità Conferma operazione e salva la nuova password. Mostra la schermata Login AND_MK01 |



1.1.6.31 – Schermata: Segnalazione

| AND_MK36 | |
|---------------------|---|
| | Componente 1 |
| Tipo | Campi selezionabili esclusivi |
| Funzionalità | Seleziona una tra le motivazioni |
| Componente 2 | |
| Tipo | Pulsante |
| Funzionalità | Annulla operazione |
| Componente 3 | |
| Tipo | Pulsante |
| Funzionalità | Conferma operazione, mostra messaggio AND_MSG09 |



1.1.6.32 – Schermata: Cambio password

| AND_MK37 | |
|----------|---|
| | Componente 1 |
| | Tipo Campo testuale Funzionalità Campo testuale riservato alla vecchia password |
| | Componente 2 |
| | Tipo Campo testuale Funzionalità Campo testuale riservato alla nuova password |
| | Componente 3 |
| | Tipo Pulsante Funzionalità Annulla operazione, mostra schermata precedente AND_MK06 |
| | Componente 4 |
| | Tipo Pulsante Funzionalità Conferma operazione, mostra schermata precedente AND_MK06 |




1.1.6.35 – Messaggio: Username non disponibile

AND_MSG01

Username non disponibile

1.1.6.36 – Messaggio: E-mail già utilizzata

AND_MSG02

E-mail già utilizzata

1.1.6.37 – Messaggio: Conferma password non corrisponde

AND_MSG03

Password e conferma password non corrispondono

1.1.6.38 – Messaggio: Conferma lista crea

AND_MSG04

Lista <Nome Lista> è stata
creata

1.1.6.39 – Messaggio: Conferma eliminazione lista

AND_MSG06

La lista <Nome Lista> è stata eliminata
con successo

1.1.6.40 – Messaggio: Conferma rimozione elemento dalla lista

AND_MSG07

Elemento cancellato dalla lista



1.1.6.41 – Messaggio: Username o Password non validi

AND_MSG08

Username o Password non
validi

1.1.6.42 – Messaggio: Conferma avvenuta segnalazione

AND_MSG09

Segnalazione avvenuta con successo

1.1.6.43 – Messaggio: Campi mancanti o invalidi

AND_MSG10

Campi mancanti o requisiti
non rispettati

1.1.6.44 – Messaggio: Codice errato

AND_MSG11

Codice errato

1.1.6.45 – Messaggio: Conferma cambiamento password

AND_MSG12

PASSWORD CAMBIATA CON SUCCESSO

1.1.6.46 – Messaggio: Password non conforme ai requisiti

AND_MSG13

PASSWORD NON RISPETTA REQUISITI



1.1.6.47 – Messaggio: Conferma cambio immagine del profilo

| |
|-----------|
| AND_MSG14 |
|-----------|

Immagine aggiornata

1.1.6.48 – Messaggio: Conferma aggiunta film in lista

| |
|-----------|
| AND_MSG15 |
|-----------|

Cambiamenti salvati

1.1.6.49 – Messaggio: Conferma invio amicizia

| |
|-----------|
| AND_MSG16 |
|-----------|

Amicizia inviata con successo

1.1.6.50 – Messaggio: Conferma amicizia accettata

| |
|-----------|
| AND_MSG17 |
|-----------|

Amicizia accettata

1.1.6.51 – Messaggio: Conferma amicizia rifiutata

| |
|-----------|
| AND_MSG18 |
|-----------|

Amicizia rifiutata

1.1.6.51 – Messaggio: Conferma amicizia rifiutata

| |
|-----------|
| AND_MSG19 |
|-----------|

Nessun risultato trovato



1.1.6.52 – Messaggio: Conferma amicizia rimossa

AND_MSG20

Amicizia rimossa con successo



1.2 Modello Funzionale: applicazione Desktop

1.2.1 Requisiti Funzionali

| | |
|--------------------|---|
| ID | DSK_RF01 |
| Nome | Accesso alla piattaforma. |
| Descrizione | Il sistema permette di effettuare accesso sulla piattaforma Desktop tramite Codice identificativo e Password. |

| | |
|--------------------|--|
| ID | DSK_RF02 |
| Nome | Visualizzare una segnalazione. |
| Descrizione | Il sistema permette all'amministratore di visualizzare i dettagli di una segnalazione. |

| | |
|--------------------|--|
| ID | DSK_RF03 |
| Nome | Gestire una segnalazione. |
| Descrizione | Il sistema permette all'amministratore di gestire una segnalazione di una particolare lista, scegliendo tra: <ul style="list-style-type: none">• Respingere una segnalazione• Cancellare la lista segnalata• Oscurare descrizione e nome della lista segnalata |

| | |
|--------------------|---|
| ID | DSK_RF04 |
| Nome | Uscire dal sistema. |
| Descrizione | Il sistema permette all'amministratore loggato di uscire dal sistema, tramite il pulsante Logout. |

| | |
|--------------------|---|
| ID | DSK_RF05 |
| Nome | Consultazione del regolamento. |
| Descrizione | Il sistema deve consentire ad un amministratore di consultare il regolamento della piattaforma. |



1.2.2 Requisiti non Funzionali

| | |
|--------------------|---|
| ID | DSK_RNF01 |
| Nome | Usabilità del sistema. |
| Descrizione | Un amministratore deve poter utilizzare a pieno il sistema dopo un utilizzo di 30 minuti. |

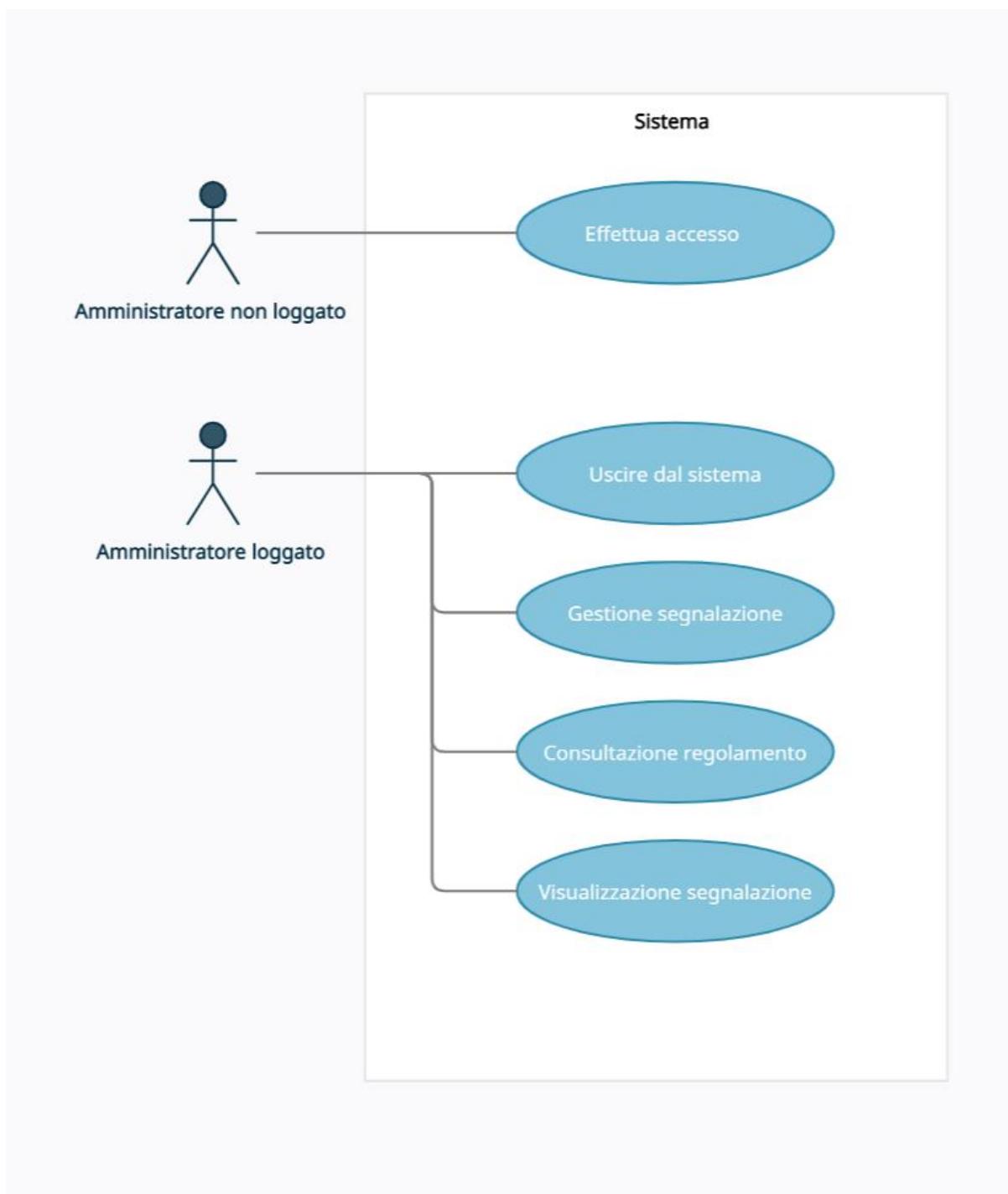
| | |
|--------------------|---|
| ID | DSK_RNF02 |
| Nome | Gestione concorrenze dei amministratori. |
| Descrizione | Il sistema deve vietare di gestire (completare la pratica della gestione di una segnalazione) una stessa segnalazione a più amministratori. |

1.2.3 Requisiti di Dominio

| | |
|--------------------|--------------------|
| ID | DSK_RD01 |
| Nome | ISO/IEC 27018:2020 |
| Descrizione | |



1.2.4 Modellazione use case diagram (Desktop)



Use case diagram 5 – Sistema generico



1.2.5 Tabelle di Cockburn (Desktop)

1.2.5.1 Accesso al sistema

| USER CASE #1 | Accesso alla piattaforma | | |
|--|--|---|---|
| Goal in Context | L'amministratore vuole entrare nella piattaforma | | |
| Preconditions | L'amministratore non è autenticato | | |
| Success End Condition | L'amministratore entra nella piattaforma | | |
| Failer End Condition | Amministratore non riesce ad entrare nella piattaforma | | |
| Primary Actor | Amministratore non loggato | | |
| Trigger | Amministratore avvia applicazione Desktop | | |
| DESCRIPTION | STEP | AMMINISTRATORE NON LOGGATO | SISTEMA |
| EXTENTION #1 Amministratore non compila i campi o compila con i valori non validi | 1 | Avvia applicazione Desktop | |
| | 2 | | Mostra la schermata Login DKS_MK01 |
| | 3 | Inserire il Codice Identificativo e la Password e preme il pulsante Login | |
| | 4 | | Mostra la schermata principale DKS_MK02 |
| EXTENTION #1 | STEP | AMMINISTRATORE NON LOGGATO | SISTEMA |
| Amministratore non compila i campi o compila con i valori non validi | 4a | | Mostra il messaggio di errore DKS_MSG01 |
| | 5a | | Torna allo step 2 dello scenario principale |



1.2.5.2 Visualizzare una segnalazione

| USER CASE #2 | Visualizzare una segnalazione | | |
|-----------------------|--|--|---|
| Goal in Context | L'amministratore vuole visualizzare i dettagli di una segnalazione | | |
| Preconditions | L'amministratore è autenticato ed esiste una segnalazione | | |
| Success End Condition | L'amministratore visualizza una segnalazione | | |
| Failer End Condition | Nessuna | | |
| Primary Actor | Amministratore loggato | | |
| Trigger | Amministratore preme sulla segnalazione interessata | | |
| DESCRIPTION | STEP | AMMINISTRATORE LOGGATO | SISTEMA |
| | 1 | Preme sulla segnalazione interessata nella schermata principale DSK_MK02 | |
| | 2 | | Mostra la schermata contenente i dettagli della segnalazione DSK_MK03 |



1.2.5.3 Gestione di una segnalazione

| USER CASE #3 | | | | Gestire una segnalazione |
|--|---|---|--|---------------------------------|
| Goal in Context | L'amministratore vuole gestire una segnalazione | | | |
| Preconditions | L'amministratore è autenticato ed esiste una segnalazione | | | |
| Success End Condition | L'amministratore visualizza gestisce una segnalazione | | | |
| Failer End Condition | Nessuna | | | |
| Primary Actor | Amministratore loggato | | | |
| Trigger | Amministratore sceglie la segnalazione nella schermata principale | | | |
| DESCRIPTION | STEP | AMMINISTRATORE LOGGATO | SISTEMA | |
| | 1 | Sceglie la segnalazione nella schermata principale DSK_MK02 | | |
| | 2 | | Apre una nuova schermata contenente i dettagli della segnalazione DSK_MK03 | |
| | 3 | Preme su uno dei pulsanti tra: Cancella, Oscura o Respingi | | |
| | 4 | Preme sul pulsante Conferma | | |
| | 5 | | Mostra il messaggio di conferma DKS_MSG02 e chiude la schermata DSK_MK03 | |
| EXTENTION #1 Amministratore preme "Annulla" | STEP | AMMINISTRATORE LOGGATO | SISTEMA | |
| | 3a-4b | Amministratore preme il pulsante "Annulla" | | |
| | 4a-5b | | Chiude la finestra DKS_MK03. Termina caso d'uso | |



1.2.5.4 Uscire dal sistema

| USER CASE #4 | | | | Uscire dal sistema |
|------------------------------|-------------|--|--|--|
| Goal in Context | | | | L'amministratore vuole uscire dal sistema |
| Preconditions | | | | L'amministratore è autentificato |
| Success End Condition | | | | L'amministratore esce dal sistema |
| Failer End Condition | | | | Nessuna |
| Primary Actor | | | | Amministratore loggato |
| Trigger | | | | Amministratore preme il pulsante Logout nella schermata principale |
| DESCRIPTION | STEP | AMMINISTRATORE LOGGATO | | SISTEMA |
| | 1 | Preme il pulsante Logout nella schermata principale DSK_MK02 | | |
| | 2 | | | Mostra la schermata di Login DSK_MK01 |



1.2.5.6 Consultare il regolamento

| USER CASE #5 | | | | Consultazione del regolamento |
|------------------------------|-------------|--|--|---|
| Goal in Context | | | | L'amministratore vuole consultare il regolamento |
| Preconditions | | | | L'amministratore è autentificato |
| Success End Condition | | | | L'amministratore consulta il regolamento |
| Failer End Condition | | | | Nessuna |
| Primary Actor | | | | Amministratore loggato |
| Trigger | | | | Amministratore preme il pulsante consulta il regolamento nella schermata principale |
| DESCRIPTION | STEP | AMMINISTRATORE LOGGATO | | SISTEMA |
| | 1 | Preme il pulsante consulta il regolamento nella schermata principale DSK_MK02 | | |
| | 2 | | | Apre una nuova finestra contenente il regolamento DSK_MK04 |



1.2.6 Mock-up (Desktop)

1.2.6.1 Schermata: Login

DSK_MK01

CineMate

Codice identificativo 1

Password 2

Login 3

| Componente 1 | |
|--------------|--|
| Tipo | Campo testuale |
| Descrizione | Campo in cui inserire il codice identificativo relativo all'account amministratore |

| Componente 2 | |
|--------------|---|
| Tipo | Campo testuale |
| Descrizione | Campo in cui inserire la password relativa all'account amministratore |

| Componente 3 | |
|--------------|--|
| Tipo | Pulsante |
| Descrizione | Pulsante che consente di accedere alla schermata "DSK_MK02" nel caso in cui i dati inseriti risultino corretti |



1.2.6.2 Schermata: Home

Gestione nuove segnalazioni

Consulta il regolamento (3)

Tutte le segnalazioni

numero segnalazione (4)

| Contenuto | Utente segnalatore | Utente segnalato | Orario | Numero segnalazione |
|---|--------------------|------------------|------------------|---------------------|
| Segnalazione spoiler <Nome Lista> (1) | utente1 | utente2 | 13.09.11/03/2020 | SC001 |
| Segnalazione contenuti violenti o ripugnanti <Nome Lista> | utente2 | utente3 | 13.32.11/03/2020 | SU001 |
| | | | | |
| | | | | |
| | | | | |

CineMates20

Logout (2)

Componente 1

| Tipo | Pulsante |
|--------------|--|
| Descrizione | Pulsante che porta alla schermata della “Gestione di una segnalazione” DSK_MK03 |
| Componente 2 | |
| Tipo | Pulsante |
| Descrizione | Pulsante di Logout, permette di uscire dall’account tornando alla schermata DSK_MK01 |

Componente 2

| Tipo | Pulsante |
|-------------|--|
| Descrizione | Pulsante di Logout, permette di uscire dall'account tornando alla schermata DSK_MK01 |

Componente 3

| Tipo | Pulsante |
|-------------|--|
| Descrizione | Pulsante che permette di visualizzare il regolamento con la schermata DSK_MK04 |

Componente 4

| Tipo | Campo testuale |
|-------------|--|
| Descrizione | Campo testuale in cui è possibile effettuare una ricerca tra le varie segnalazioni |



1.2.6.3 Schermata: Gestione segnalazione

| DSK_MK03 | |
|---|---|
| Gestione segnalazione | |
| Utente segnalato | Orario segnalazione |
| utente1 | 13:05 03/05/2021 |
| Utente segnalatore | |
| utente2 | |
| Contenuto segnalato | |
| Commento utente2 ore <hh/mm dd/mm/yy> film <NomeFilm>: Maria alla fine tradirà Antonio. | |
| <input type="button" value="Respingi"/> <input type="button" value="Cancella contenuto"/> <input type="button" value="Oscura contenuto"/> | |
| 1 2 3 4 5 | |
| <input type="button" value="Annulla"/> <input type="button" value="Conferma"/> | |
| Componente 1 | |
| Tipo | Pulsante |
| Descrizione | Pulsante che rappresenta la volontà dall'amministratore di respingere la segnalazione. |
| Componente 2 | |
| Tipo | Pulsante |
| Descrizione | Pulsante che rappresenta la volontà dall'amministratore di cancellare il contenuto segnalato. |
| Componente 3 | |
| Tipo | Pulsante |
| Descrizione | Pulsante che rappresenta la volontà dall'amministratore di oscurare il contenuto segnalato. |
| Componente 4 | |
| Tipo | Pulsante |
| Descrizione | Pulsante per annullare tutti i provvedimenti presi. |
| Componente 5 | |
| Tipo | Pulsante |
| Descrizione | Pulsante per confermare tutti i provvedimenti presi. |



1.2.6.4 Schermata: Regolamento

DSK_MK04

Regolamento

Gli utenti possono effettuare delle segnalazione di liste, le cause della segnalazione sono esclusivamente legati ai contenuti Spoiler, violenti o ripugnanti e contenuti offensivi od Oltraggiosi.

Un amministratore deve gestire le segnalazione, selezionando tra le tre opzioni: Cancellare il contenuto segnalato, in questo caso la lista segnalata verrà eliminata, Oscurare il contenuto, in questo caso il Titolo e la descrizione della lista verranno oscurati in modo che non siano leggibili, Respingere la segnalazione, in questo caso la lista segnalata non sarà modificata in alcun modo.

Al fine di rendere la minima l'intrepretazione, di seguito verranno fornite le spiegazioni e gli esempi per ogni tipo di segnalazione.

Spoiler
Descrizione: Un contenuto è considerato spoiler nel caso in cui contenga alcune informazioni riguardanti gli avvenimenti contenuti nel film.
Descrizione di una scena o descrizione del finale del film, anche sottoforma implicita.
Esempio: "Nella scena in cui George ruba il porfoglio si vede che il portafoglio non è dell'epoca...."
Come gestire: In genere Spoiler sono da oscurare ma in casi in cui è evidente che la lista è stata creata ai fini di spoilerare si può anche eliminare la lista stessa.

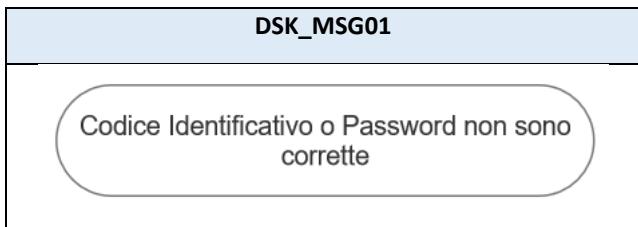
Contenuti violenti o ripugnanti
Descrizione: Incitamento di altri a commettere atti di violenza contro individui o un gruppo definito di persone.
Contenuti che riguardano incidenti stradali, disastri naturali, conseguenze di guerre o attacchi terroristici, risse di strada, attacchi fisici, violenze sessuali, sacrifici, torture, cadaveri, proteste o rivolte, rapine, procedure mediche o altri scenari simili, pubblicati allo scopo di sconvolgere o disgustare gli spettatori
Esempio: "Nel giorno xxxxx invito tutti ad uscire incontrarsi nel posto "xxxx" per vendicare accaduto "xxxx".
Come gestire: In genere questo tipo di contenuto deve essere cancellato

Contenuti offensivi od oltraggiosi
Descrizione: Titoli o descrizioni che contengono le forme di offesa personale o diretta ad un gruppo di persone
Esempio: "il popolo "XXX" non è degno di essere rappresentato in questo film"
Come gestire: In genere questo tipo di contenuto deve essere cancellato.

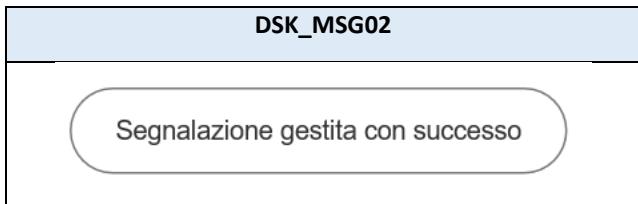
Note
 Si tenga presente che un utente può segnalare anche i contenuti che rispettano tutti i requisiti della community, quindi indipendentemente dal tipo di segnalazione che arriva amministratore deve analizzare il contenuto con estrema accuratezza al fine di prendere un giusta scelta.



1.2.6.5 Messaggio: Errore login



1.2.6.6 Messaggio: Segnalazione gestita



1.3 Glossario

| | Termine | Descrizione |
|----------|-----------------------------|--|
| 1 | Standard ISO/IEC 27018:2020 | <p>Questo Standard stabilisce obiettivi di controllo e linee guida mirati alla protezione delle informazioni personali (PII) in linea con i principi sulla privacy della ISO/IEC 29100 per l'ambiente cloud.</p> <p>In particolare, questo documento specifica le linee guida basate sulla ISO/IEC 27002, prendendo in considerazione i requisiti normativi per la protezione delle informazioni personali (PII) che possono essere applicabili nel contesto dell'ambiente di rischio per la sicurezza delle informazioni di un fornitore di servizi di cloud.</p> |
| 2 | Altro utente | Inteso un utente <i>registrato</i> e che sia diverso dal utente attuale. |
| 3 | Pulsante | Inteso un qualsiasi oggetto in cliccabile che effettua qualche operazione |
| 4 | | |
| 5 | | |
| 6 | | |



1.4 Modello di Dominio: applicazione Android

1.4.1 Modello a Oggetti di Analisi

Di seguito verranno rappresentati i Class Diagram con opportune relazioni tra le classi al fine di mostrare visualmente i concetti principali relativi al dominio.

Il metodo usato al fine di rintracciare gli elementi principali è quello dell'euristica Three-Object-Type, cioè pattern *Entity-Boundary-Control* (EBC), le informazioni riguardo questo pattern sono reperibili sul sito di Wikipedia (<https://en.wikipedia.org/wiki/Entity-control-boundary>) o su qualunque altro sito Internet.

In sintesi, pattern è costituito da tre oggetti:

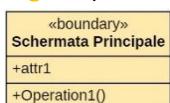
- *Entity* – modellano l'informazione persistente
- *Boundary* - modellano le interazioni tra gli attori e il sistema
- *Control* - modellano la logica che si occupa di realizzare gli Use case

Le possibili relazioni tra questi tre oggetti sono rappresentate dalla seguente tabella:

| Relazione tra gli oggetti | | | | |
|---------------------------|-------|------------|---------|--------|
| Ruolo | Actor | Boundary | Control | Entity |
| Actor | Si | Si | No | No |
| Boundary | Si | Parz/Pieno | Si | No |
| Control | No | Si | Si | Si |
| Entity | No | No | Si | Si |

Al fine di facilitare la lettura sono stati assegnati i colori a ogni oggetto dentro il Class Diagram

- Boundary (Colore giallo)



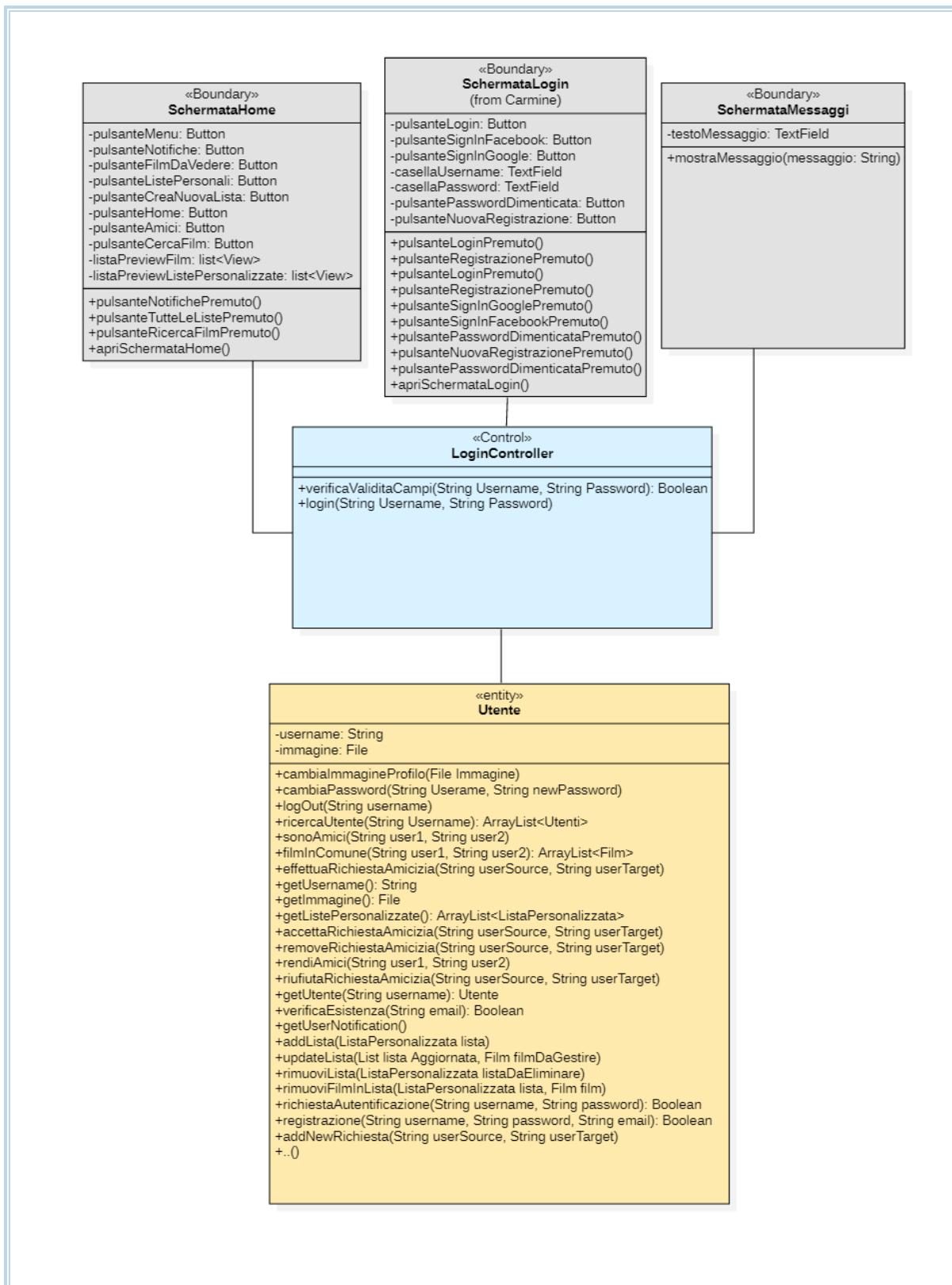
- Control (Colore grigio)



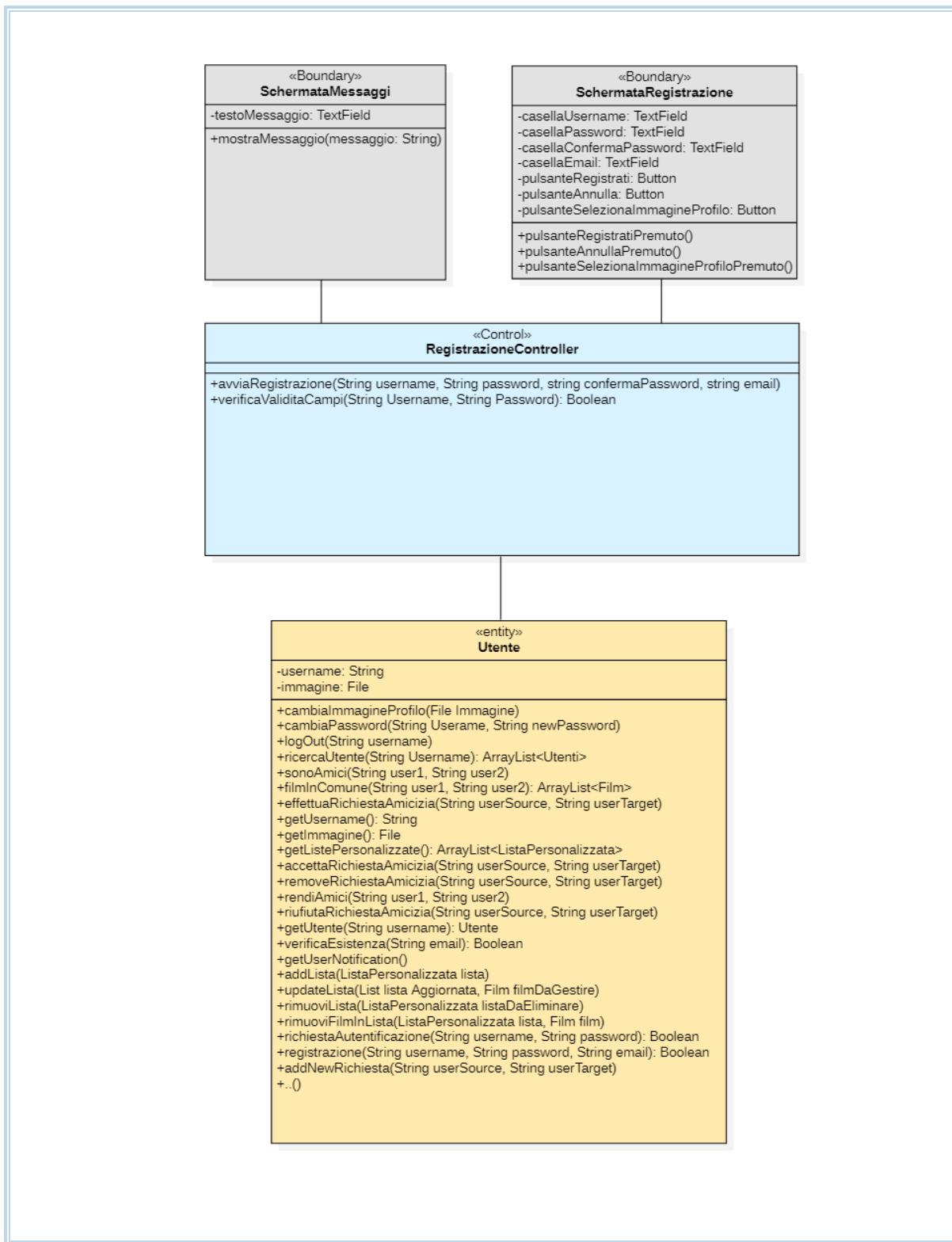
- Utente (Colore azzurro)



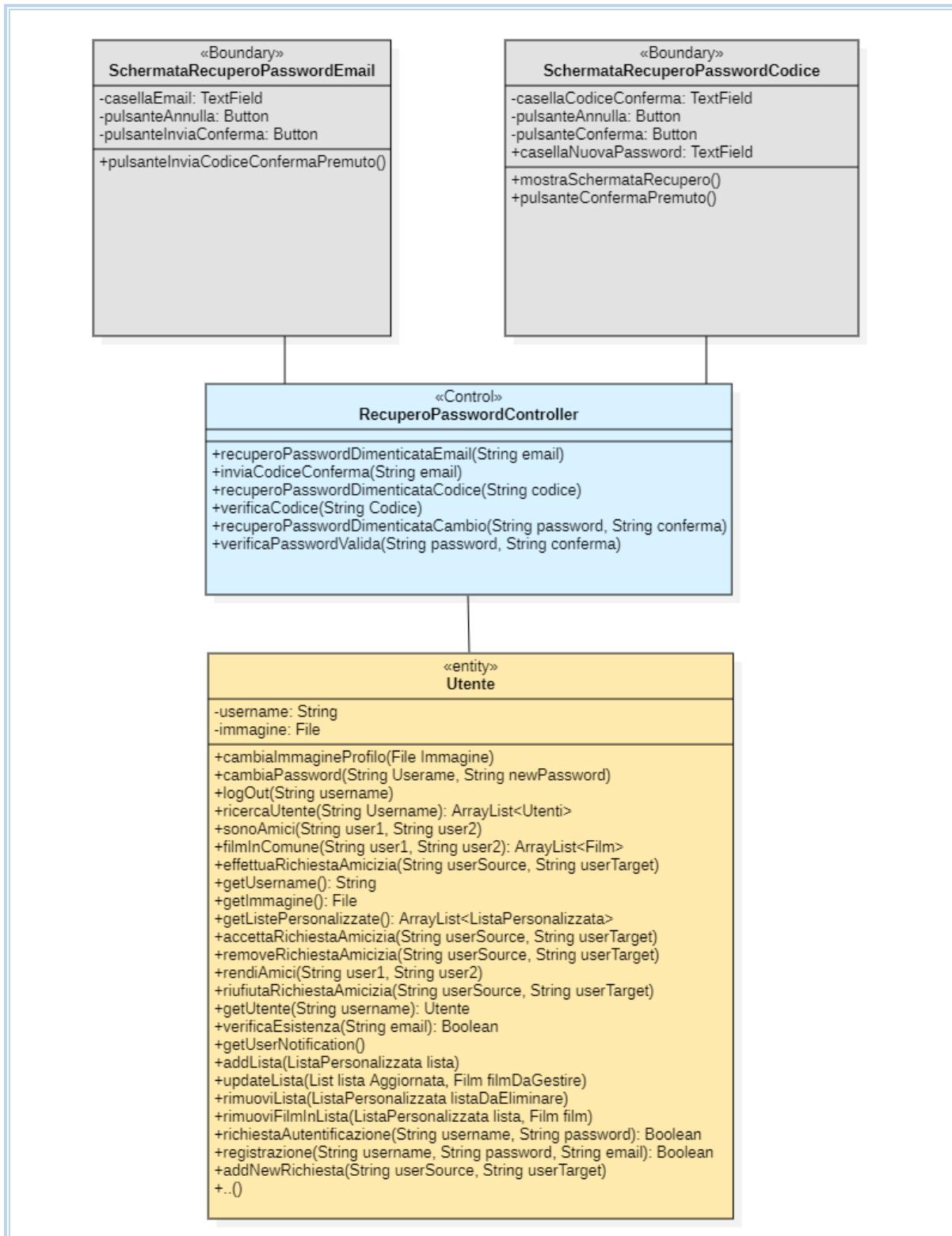
1.4.1.1 Accesso al sistema



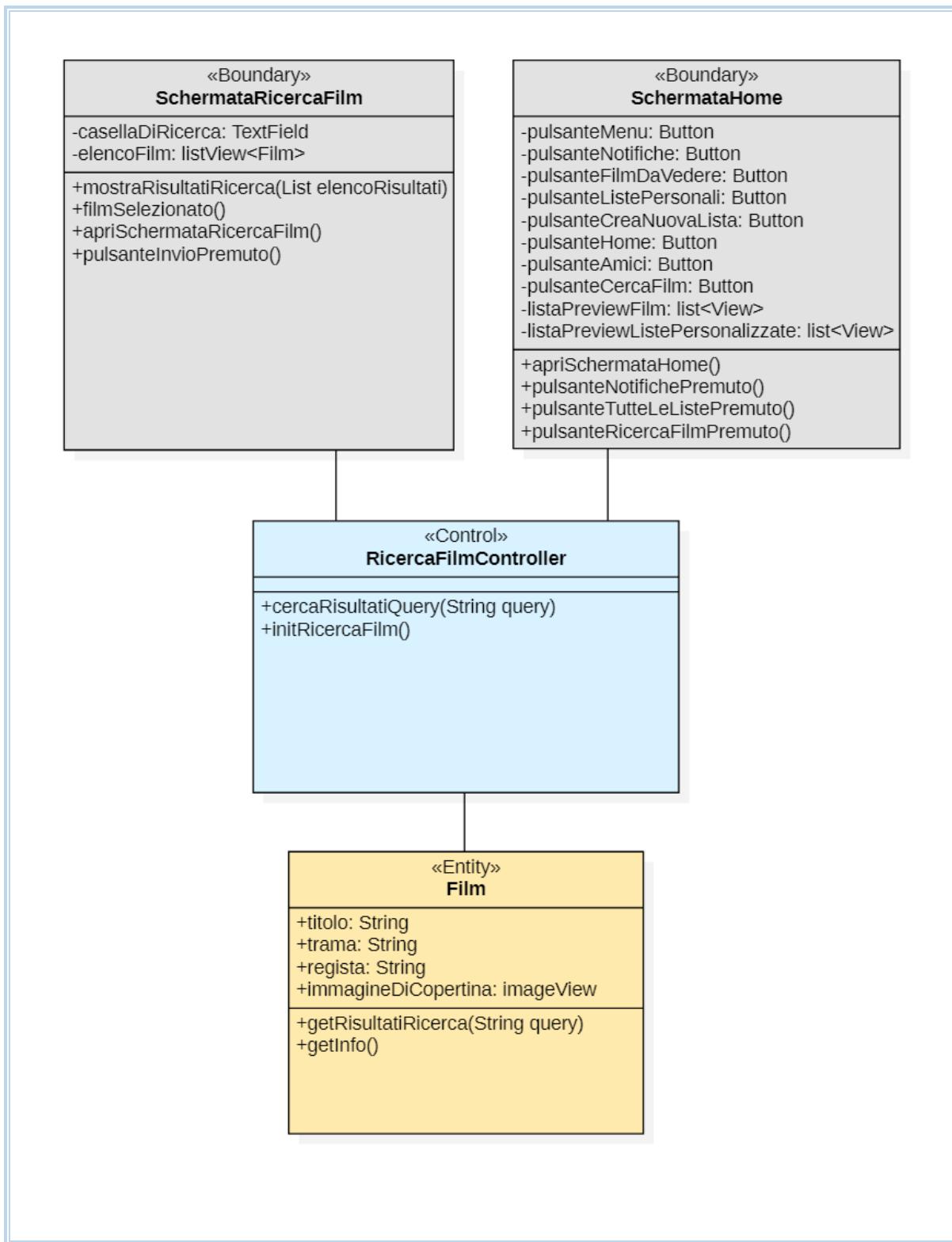
1.4.1.2 Registrazione al sistema



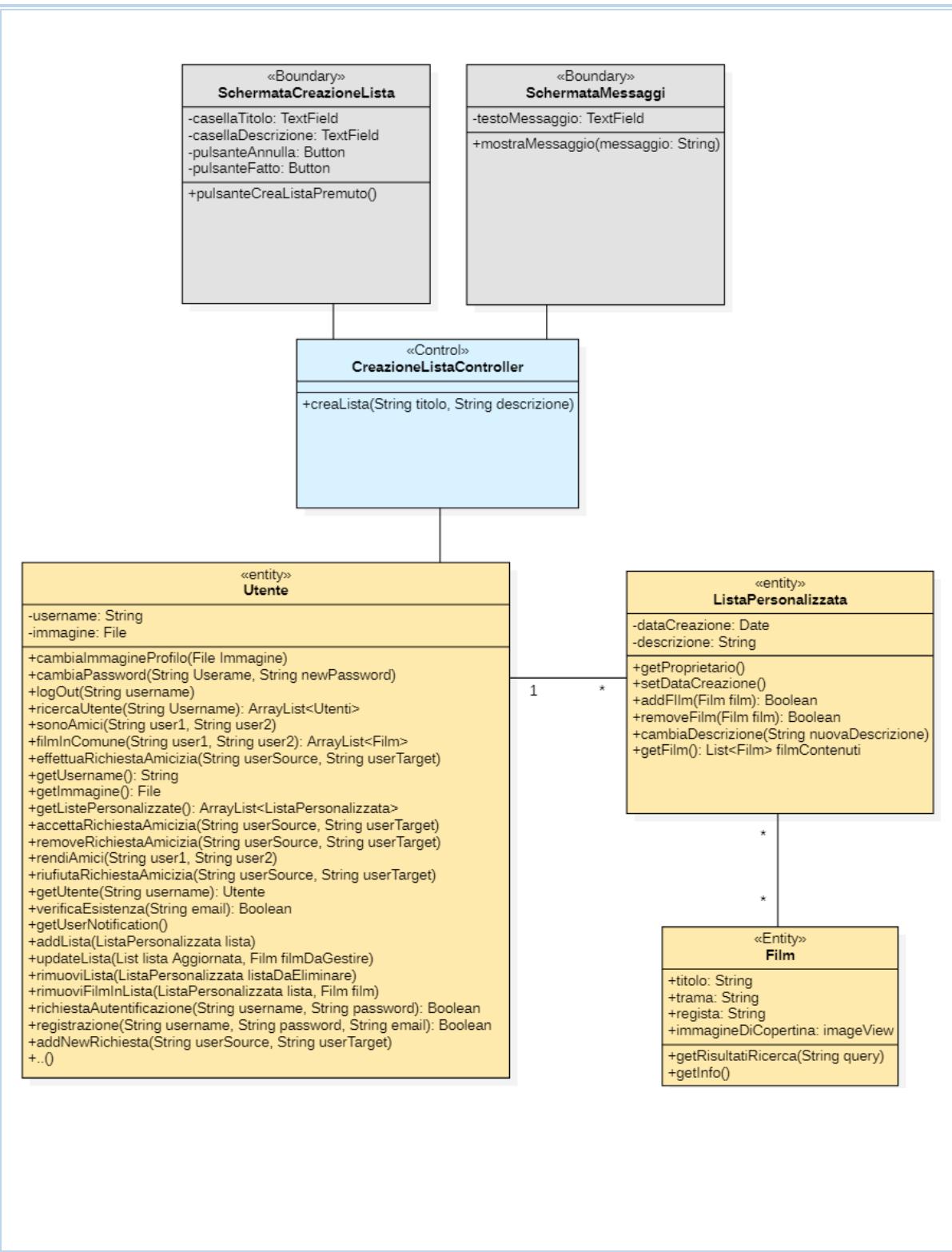
1.4.1.3 Recupero password dimenticata



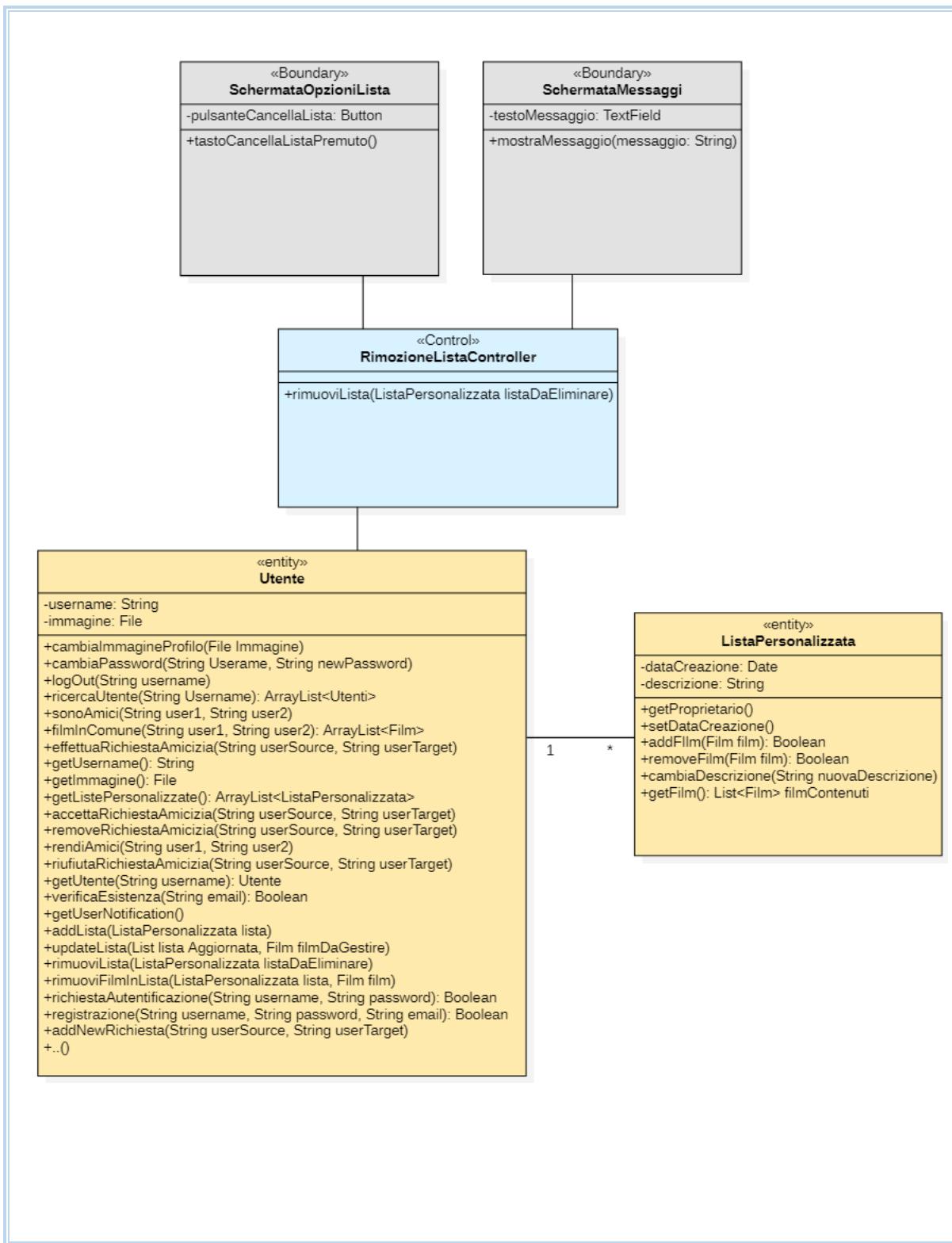
1.4.1.4 Effettuare ricerche film



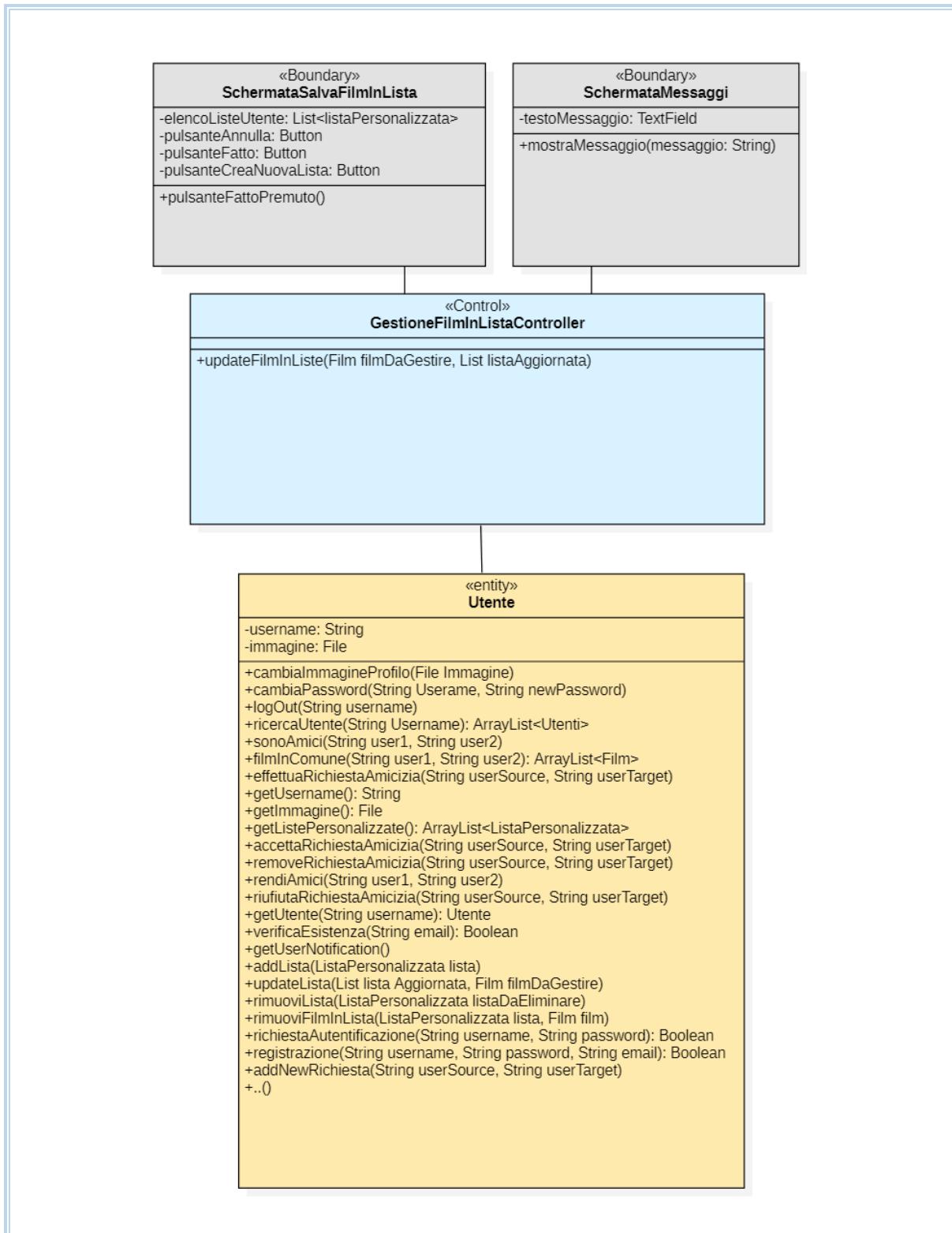
1.4.1.5 Creazione di una lista di film personalizzata



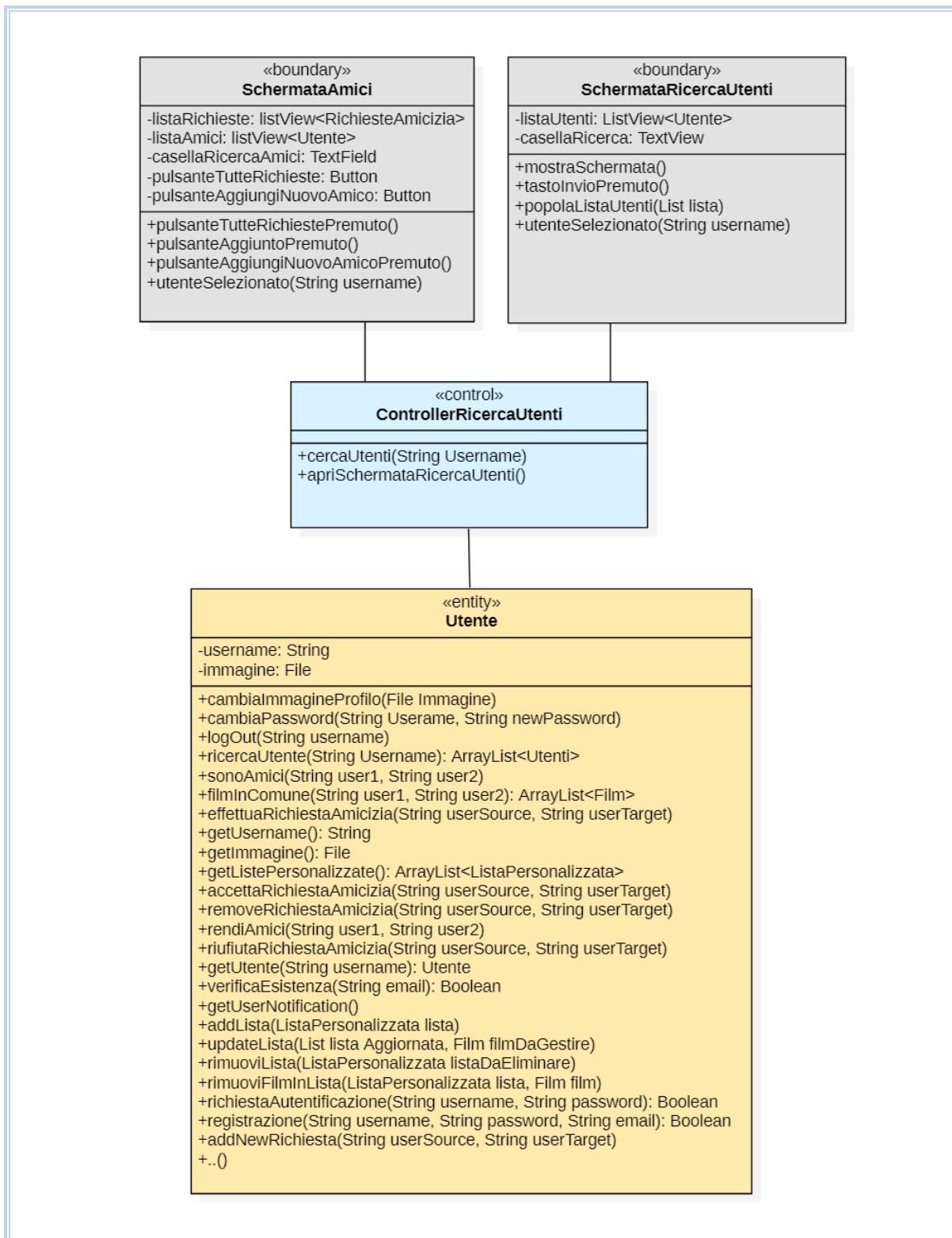
1.4.1.6 Rimozione di una lista di film personalizzata



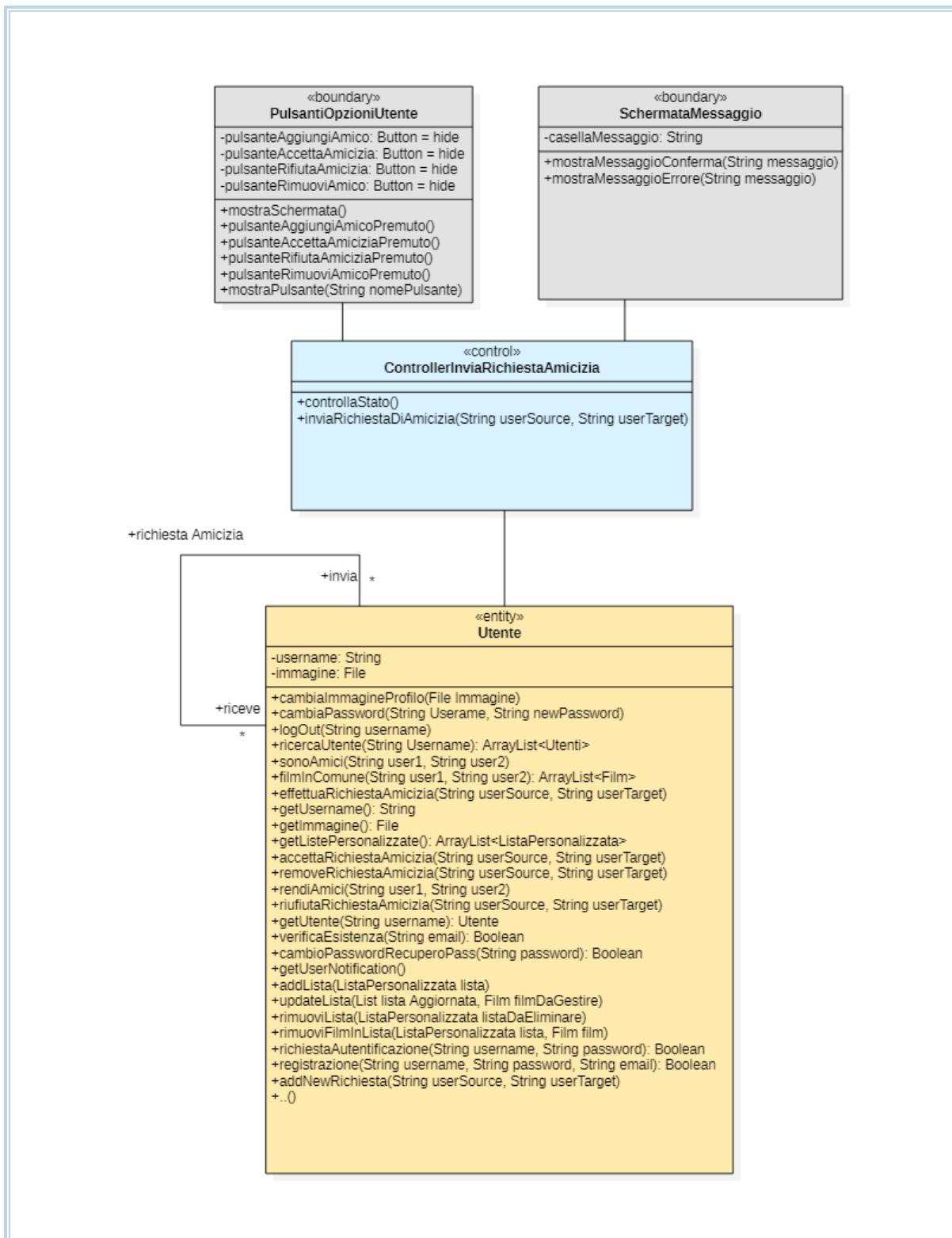
1.4.1.7 Aggiungere e rimuovere film da una lista di film personalizzata



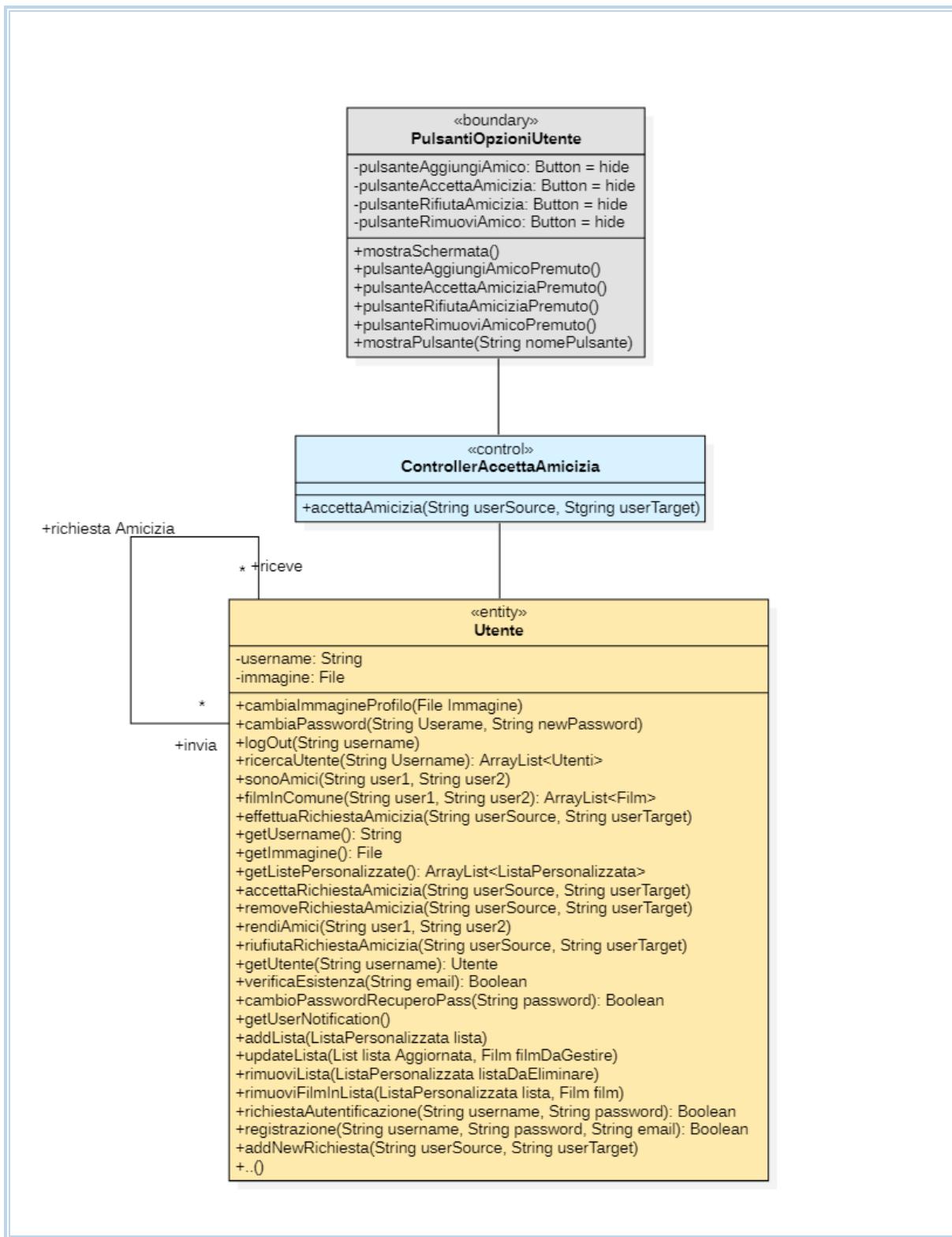
1.4.1.8 Effettuare ricerche di altri utenti



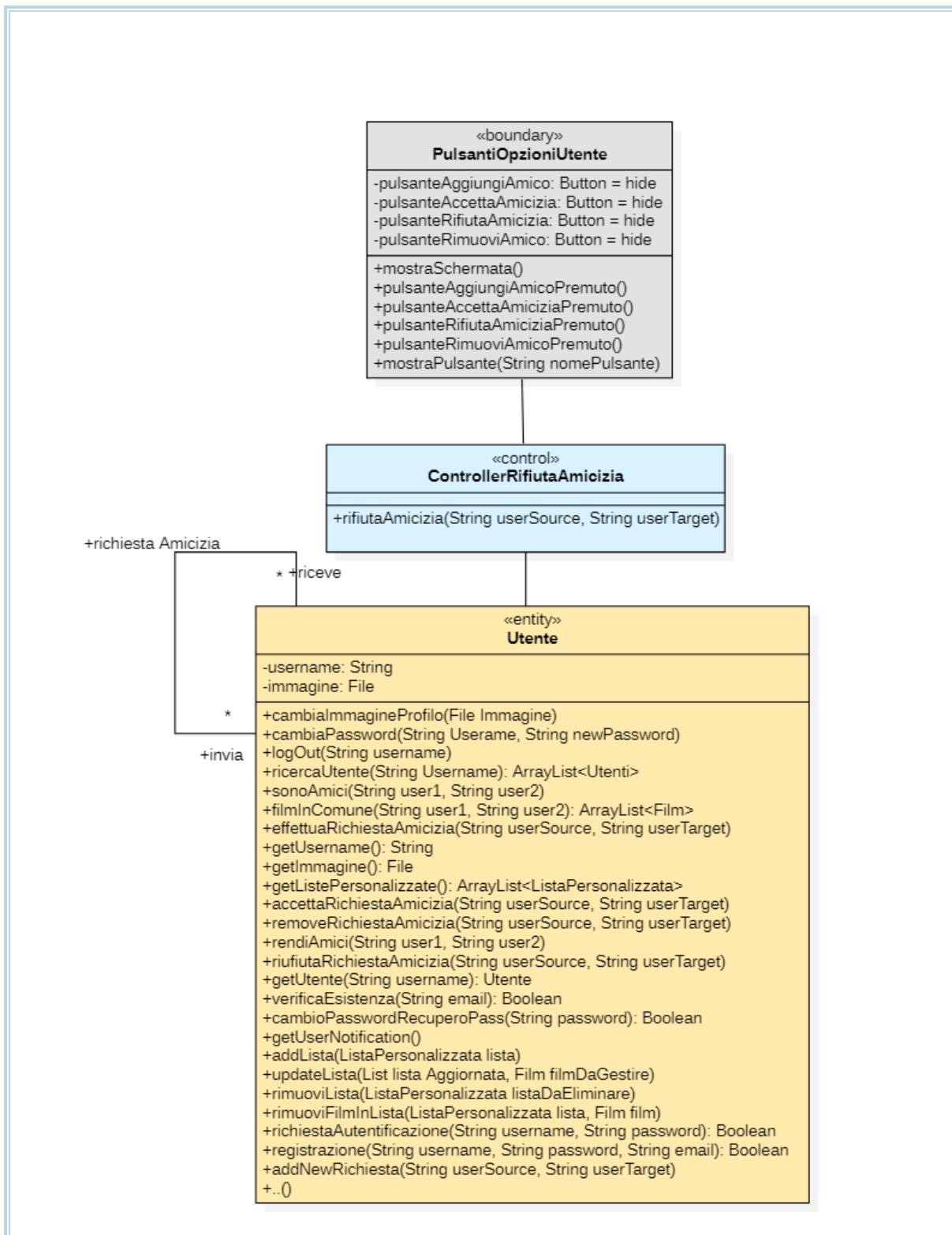
1.4.1.9 Inviare richieste di amicizia



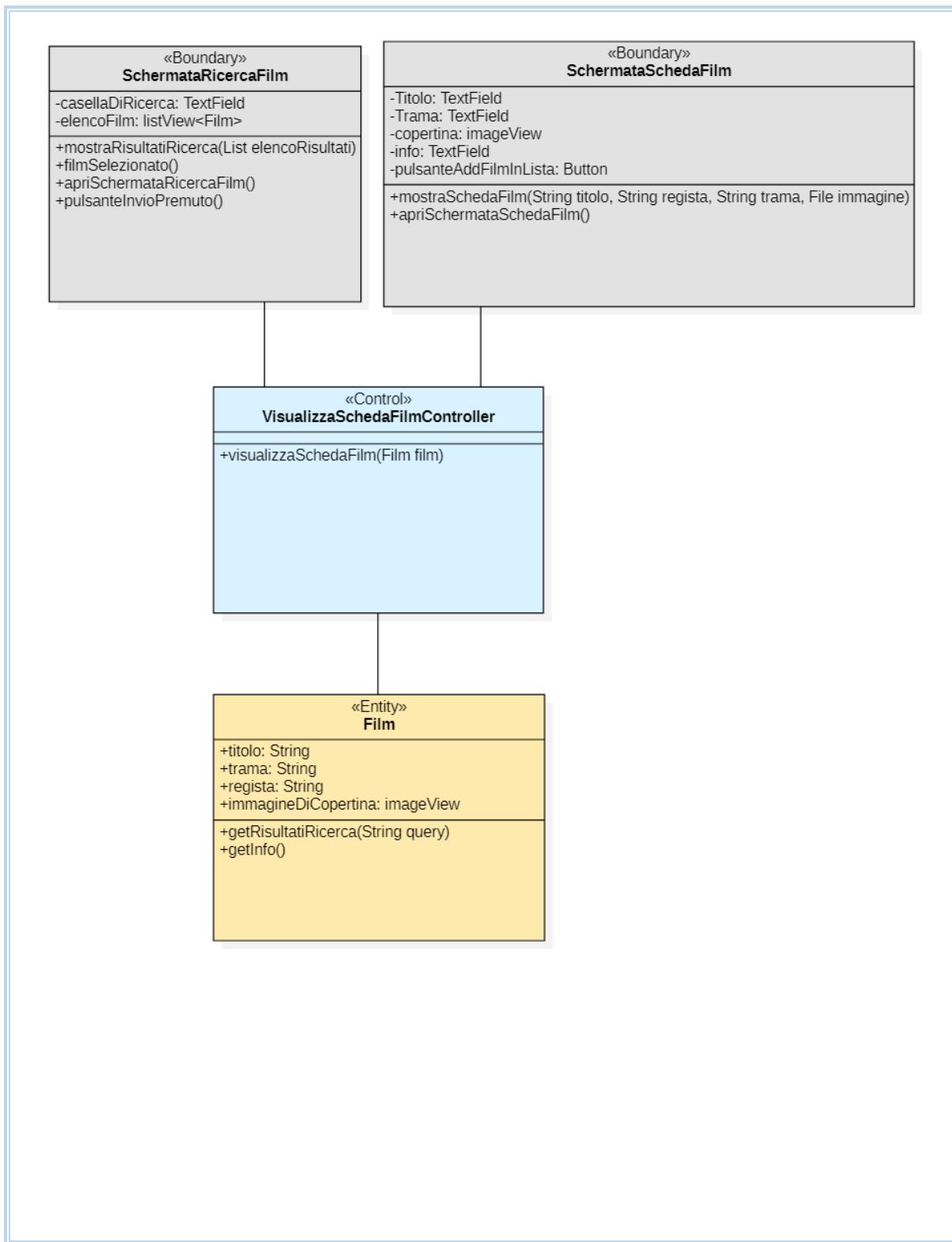
1.4.1.10 Accettare richieste di amicizia



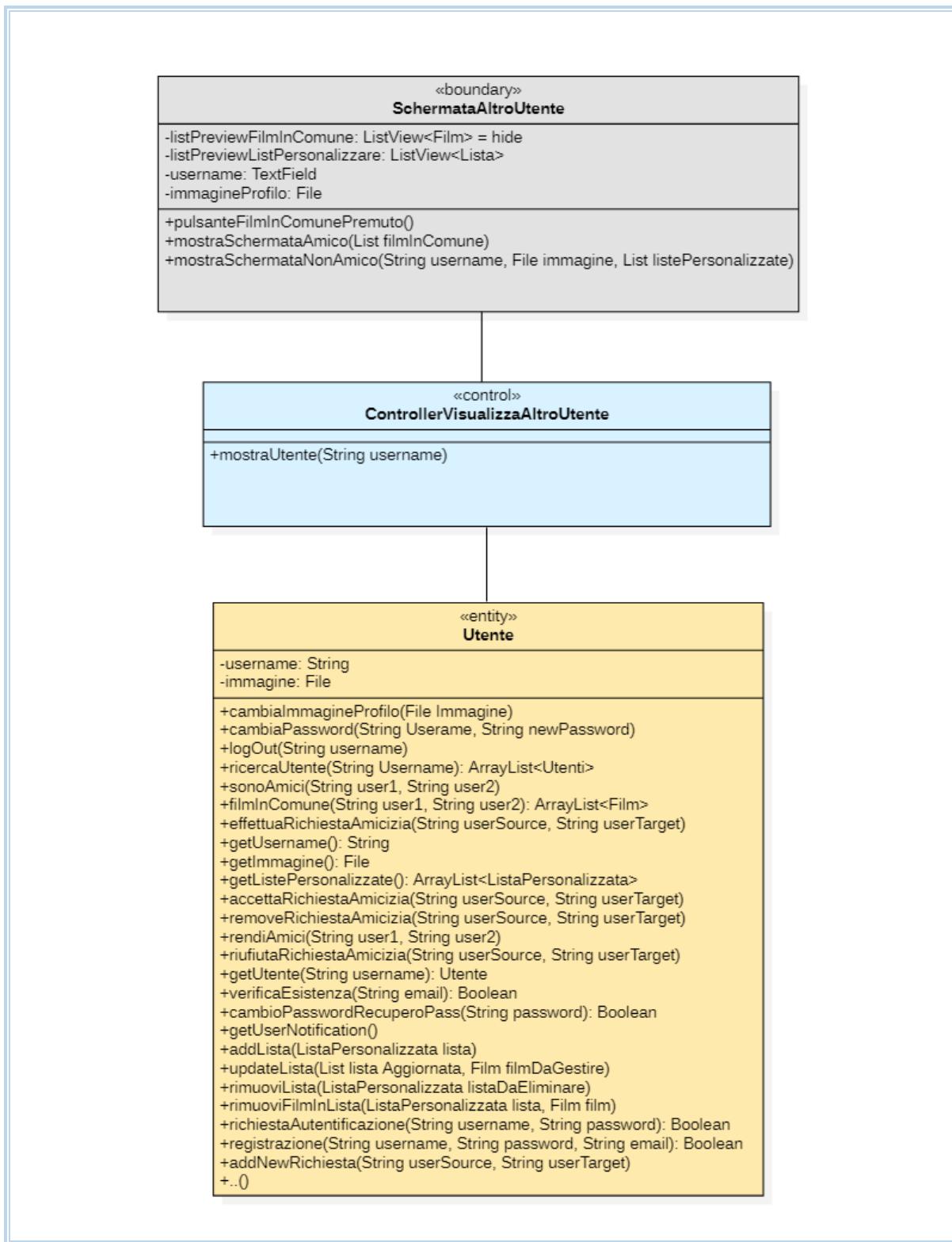
1.4.1.11 Rifiutare richieste di amicizia



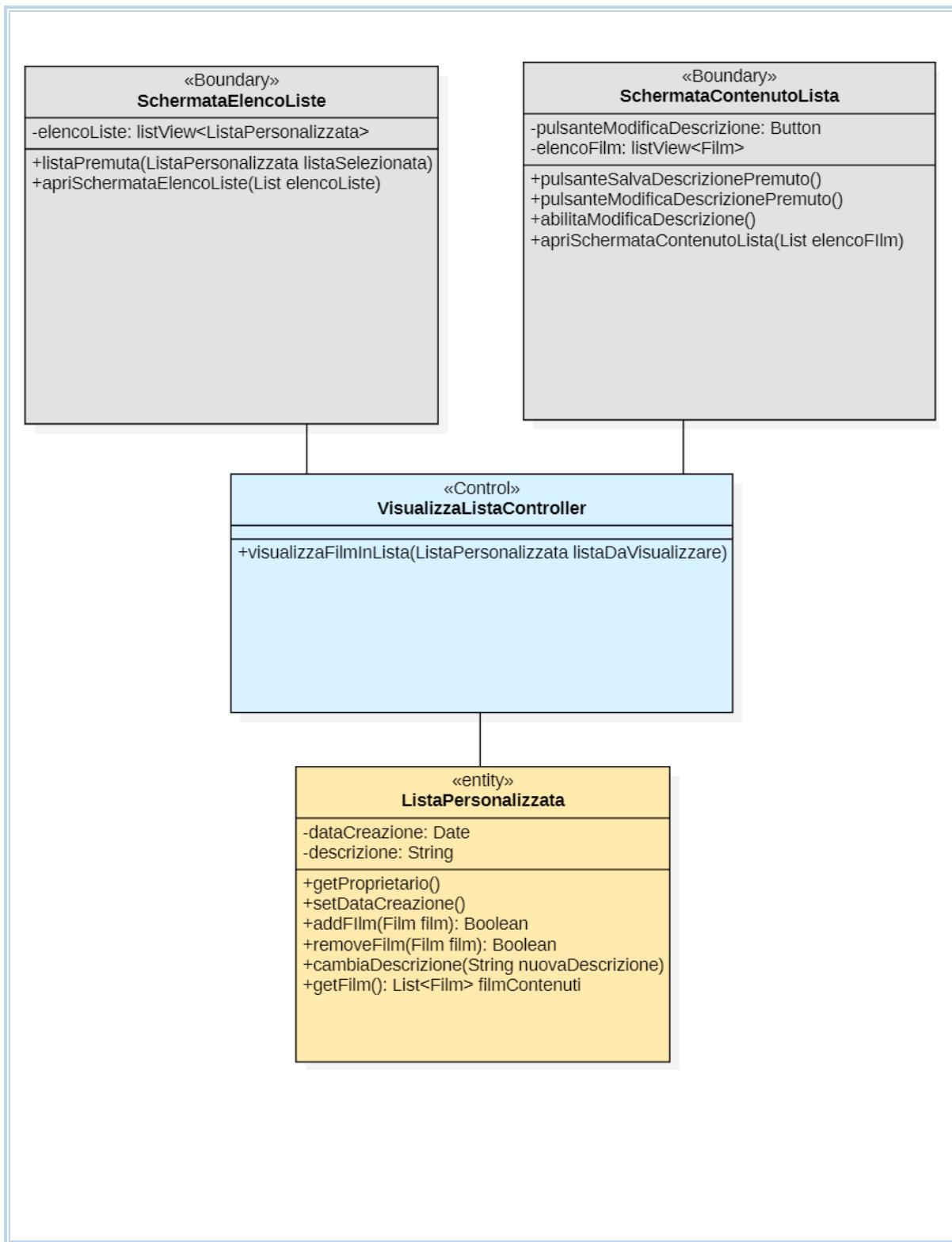
1.4.1.12 Visualizzazione della scheda di un film



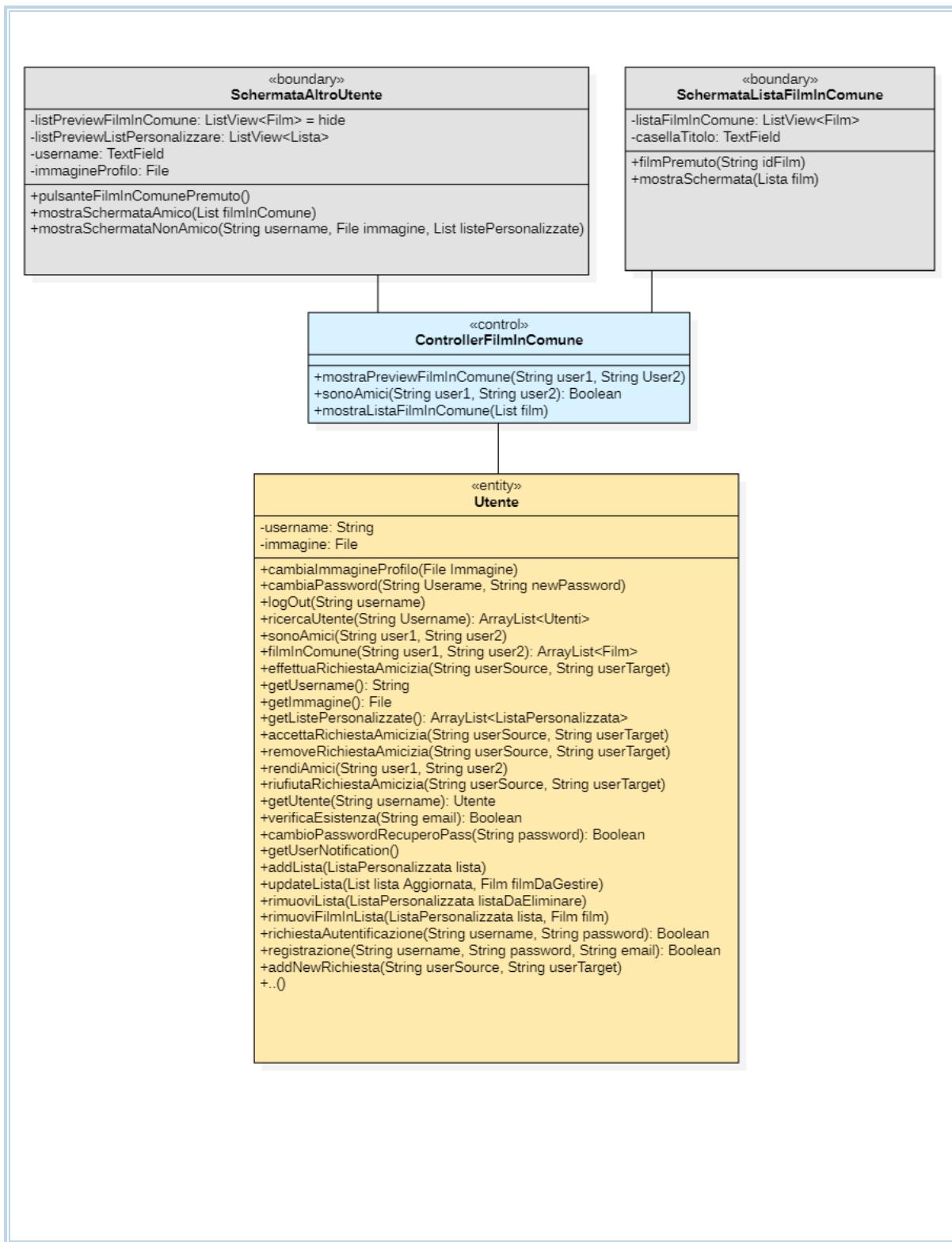
1.4.1.13 Visualizzazione della scheda utente



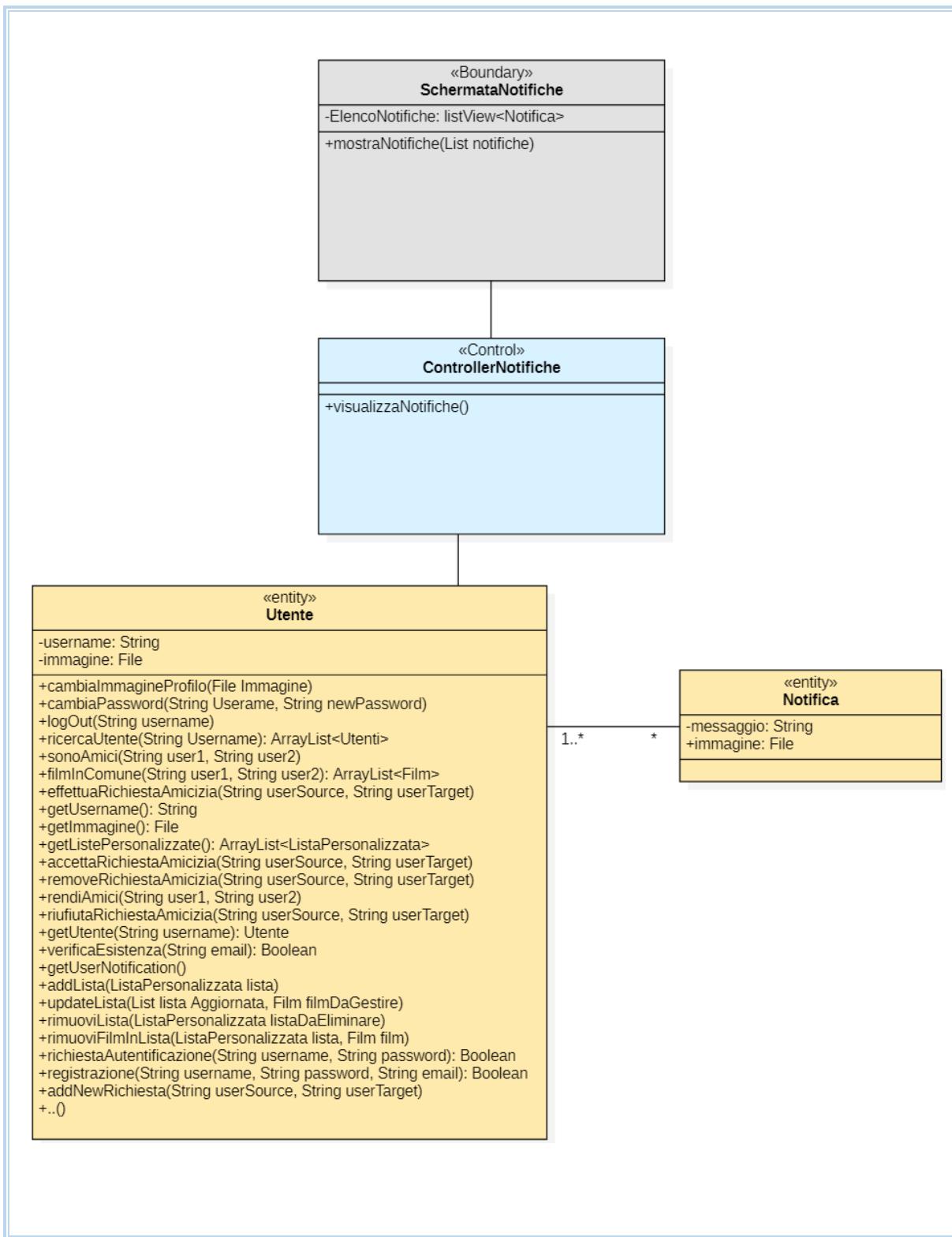
1.4.1.14 Visualizzazione di una lista



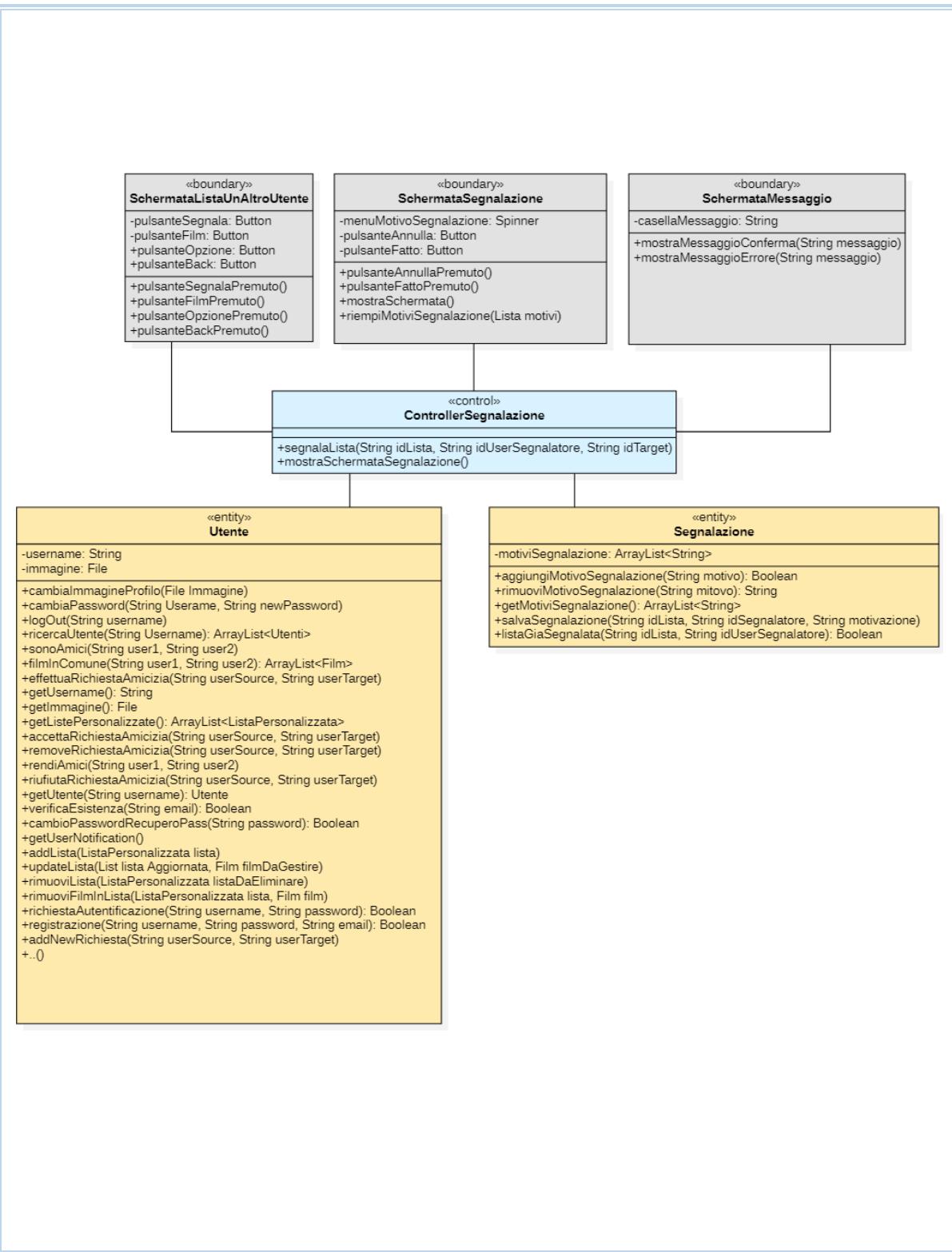
1.4.1.15 Visualizzazione dei film in comune con un amico



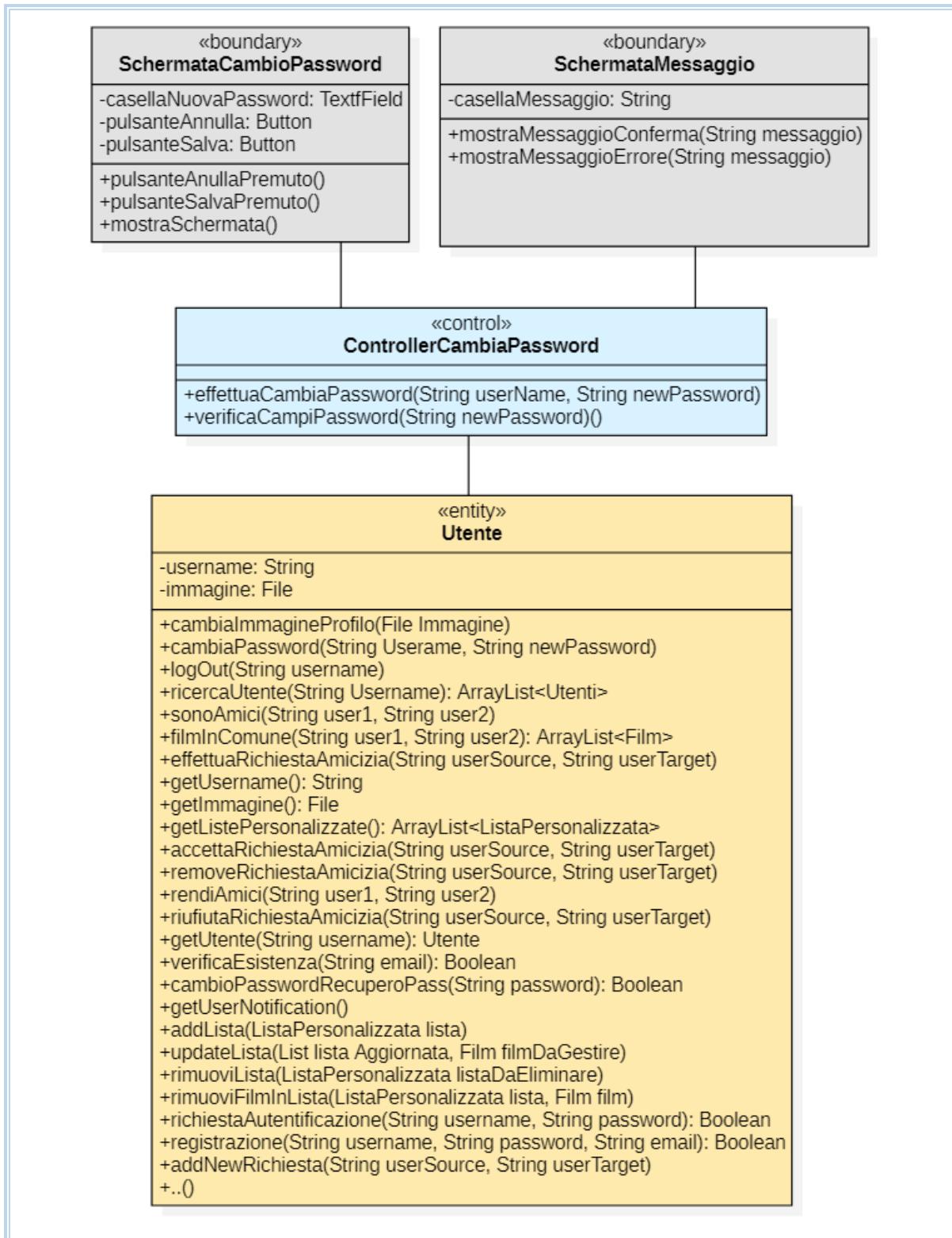
1.4.1.16 Visualizzazione delle notifiche



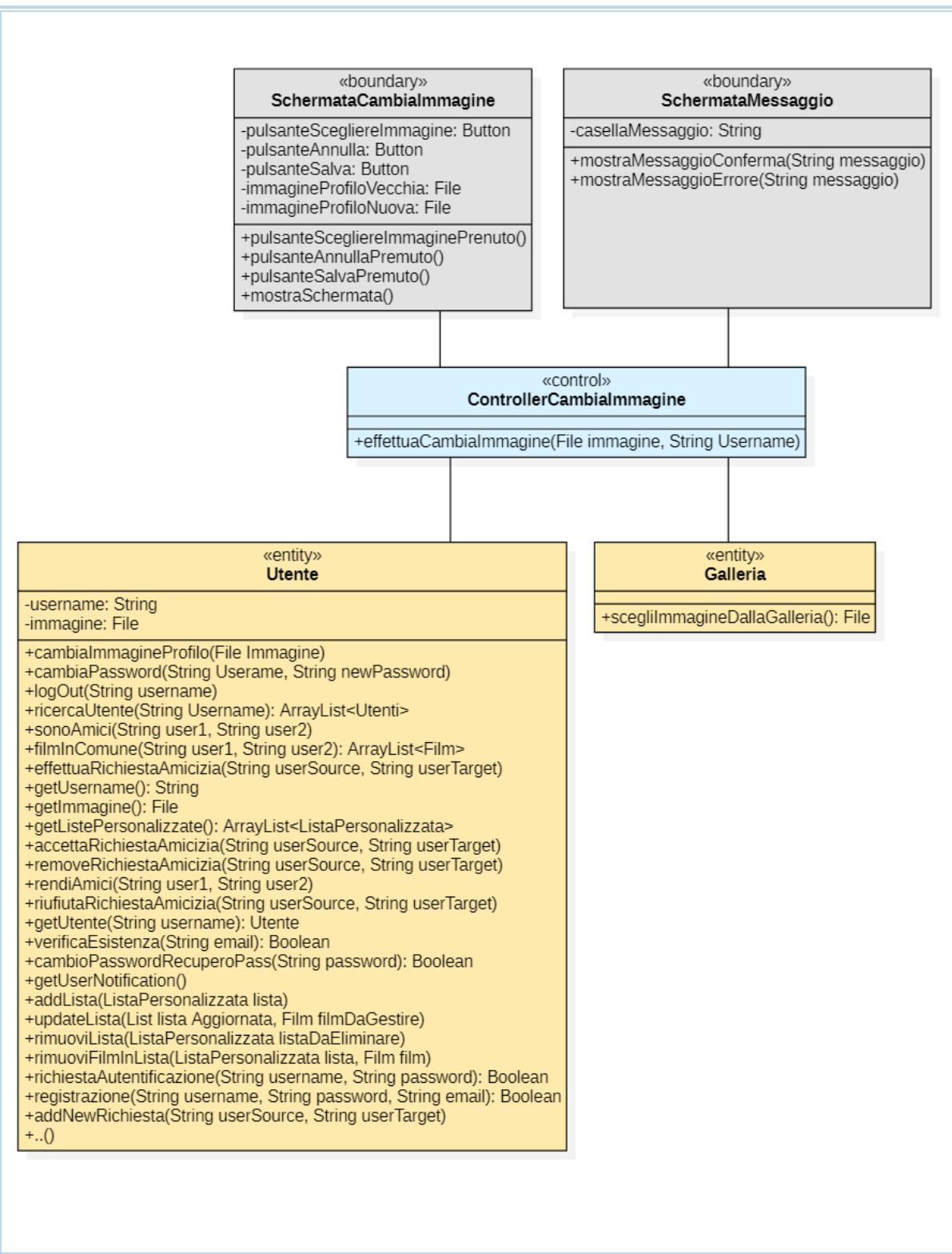
1.4.1.17 Effettuare una segnalazione



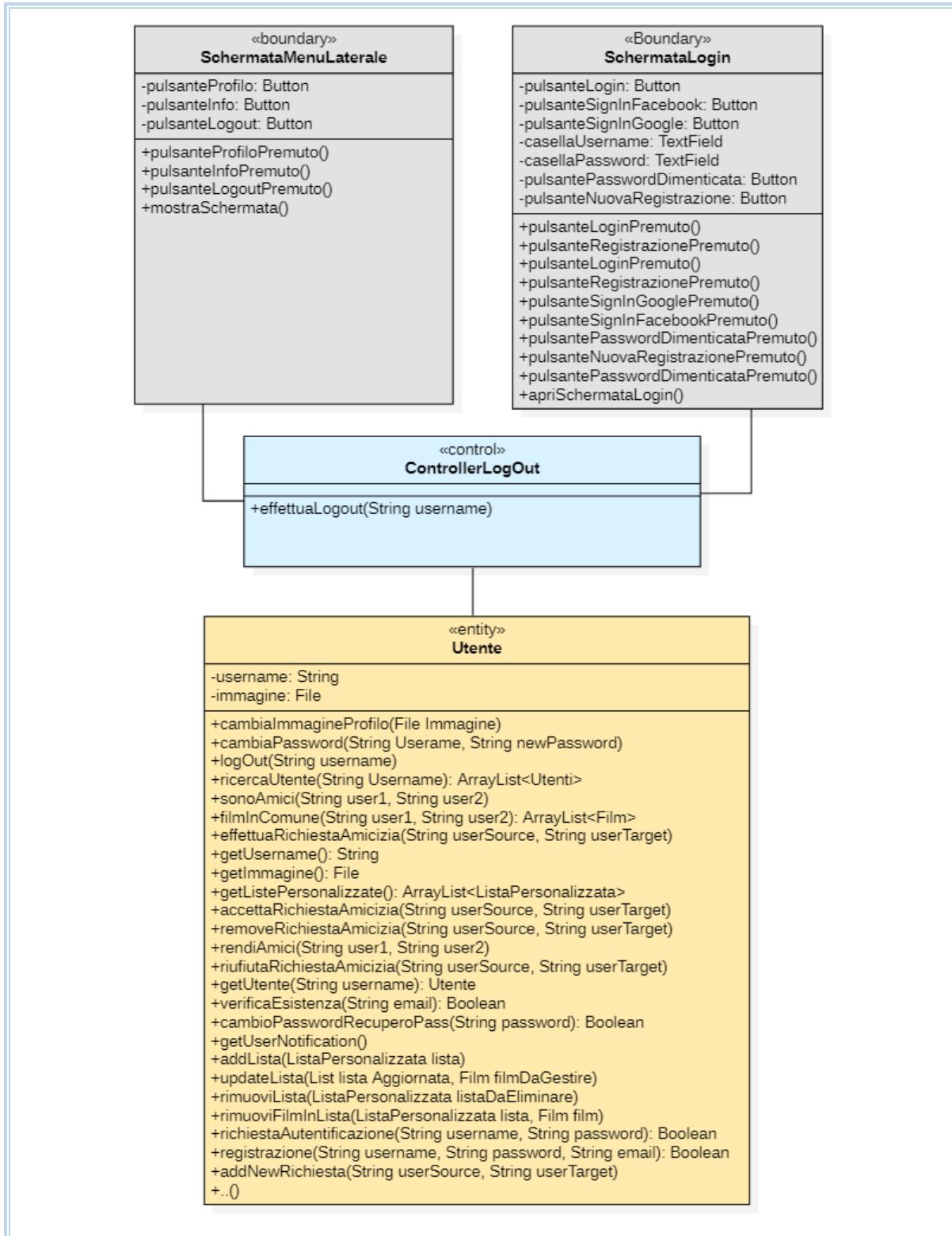
1.4.1.18 Cambiare la password



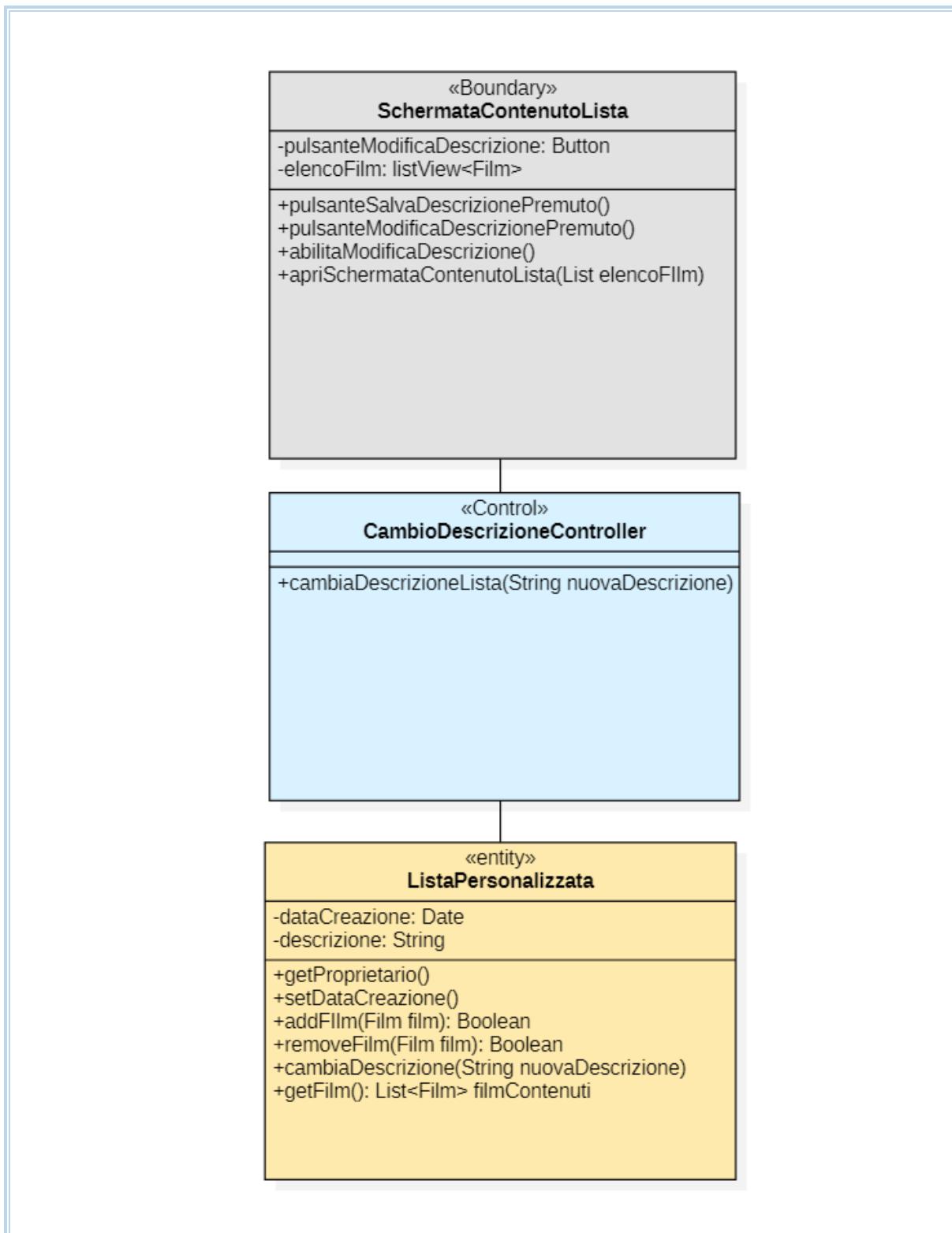
1.4.1.19 Cambiare immagine di profilo



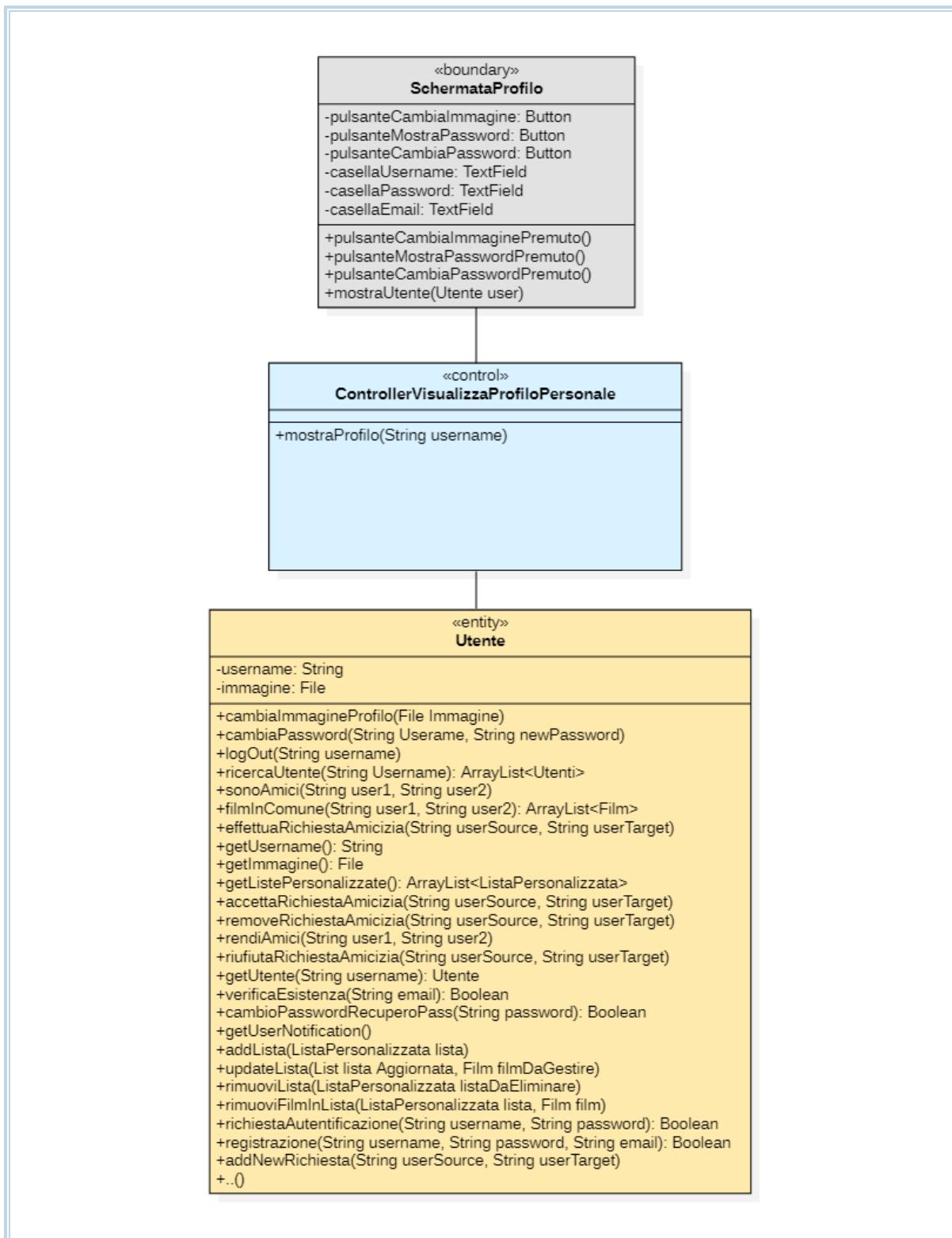
1.4.1.20 Uscire dal profilo



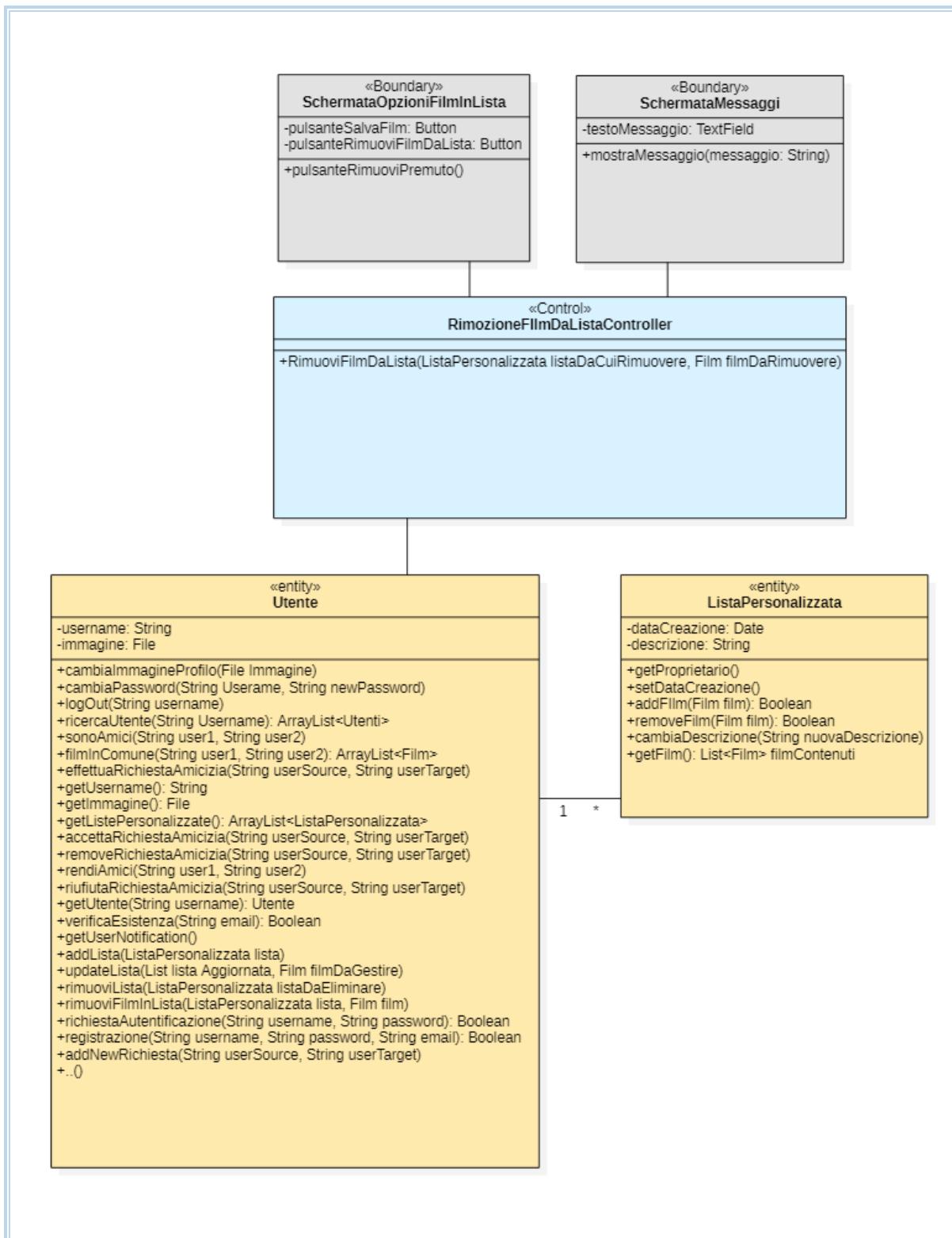
1.4.1.21 Modificare la descrizione di una lista



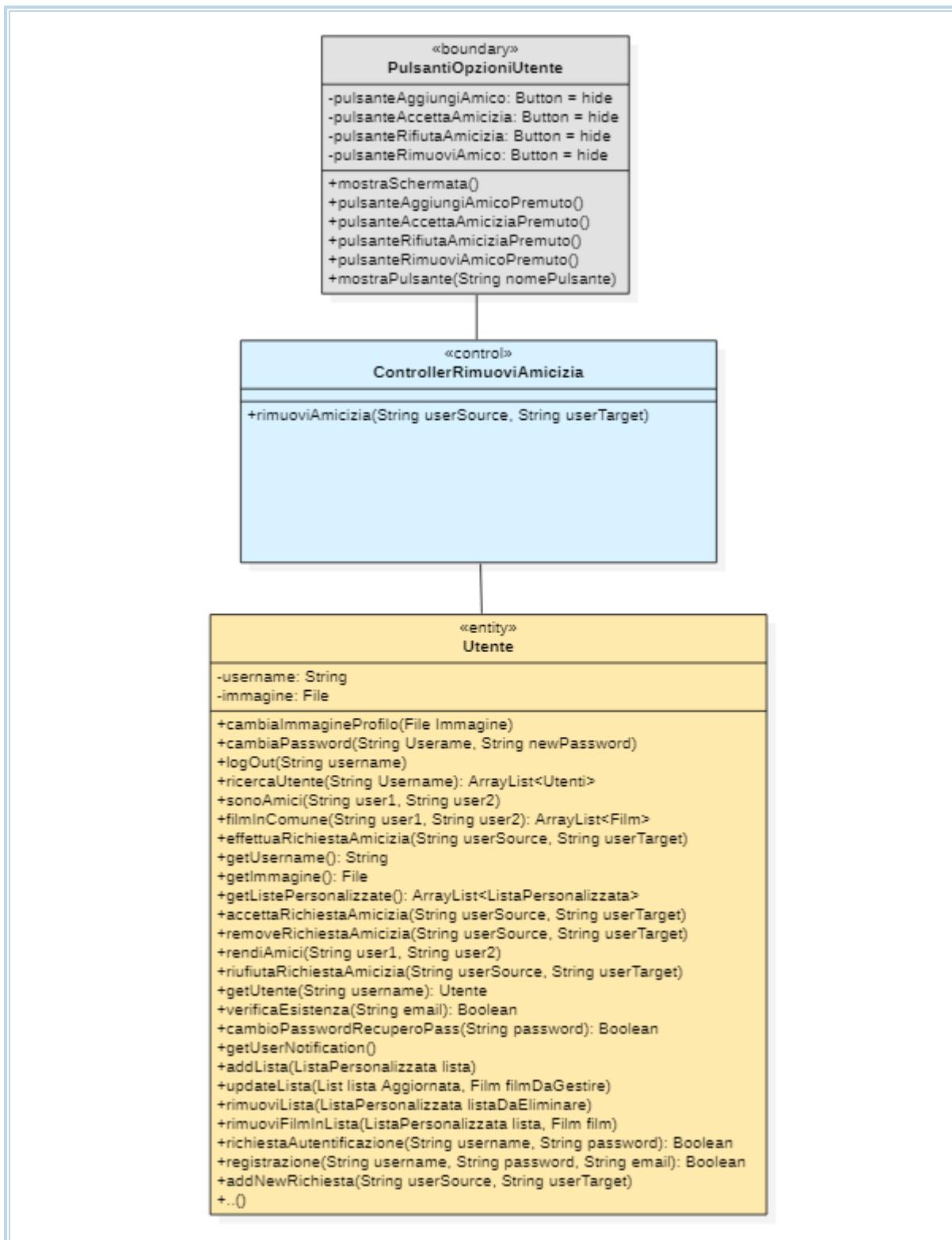
1.4.1.22 Visualizzazione del proprio profilo



1.4.1.23 Rimozione di un film da una lista (2° modo)



1.4.1.24 Rimozione dell'amicizia

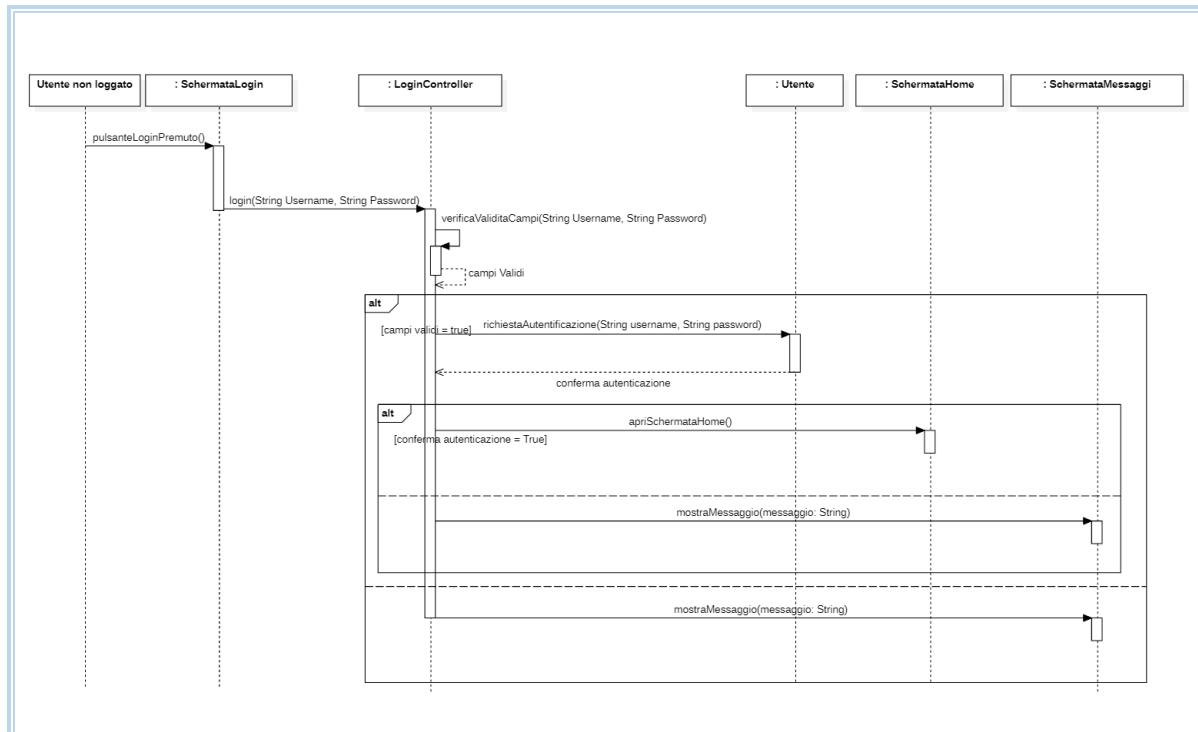


1.4.2 Diagrammi di sequenza di analisi

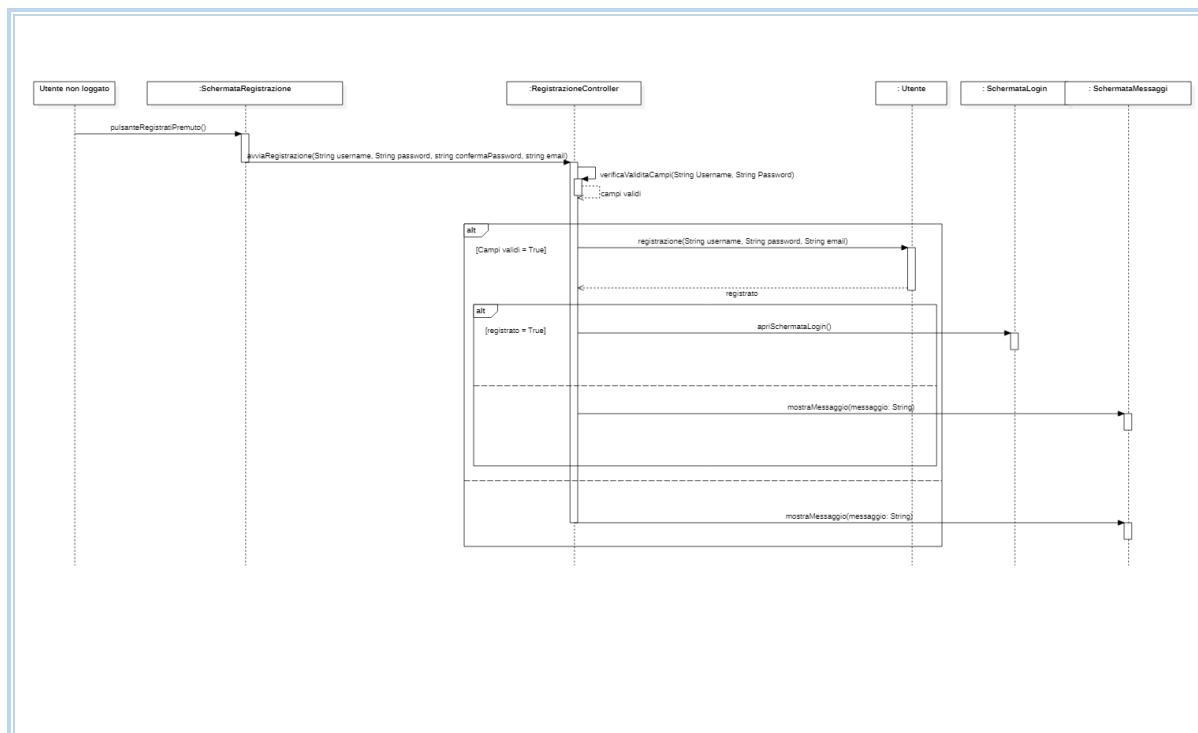
Di seguito sono riportati i diagrammi di sequenza di analisi, il loro funzionamento è quello di mostrare la relazione che intercorre tra le classi mostrate nei class diagram della sezione precedente.



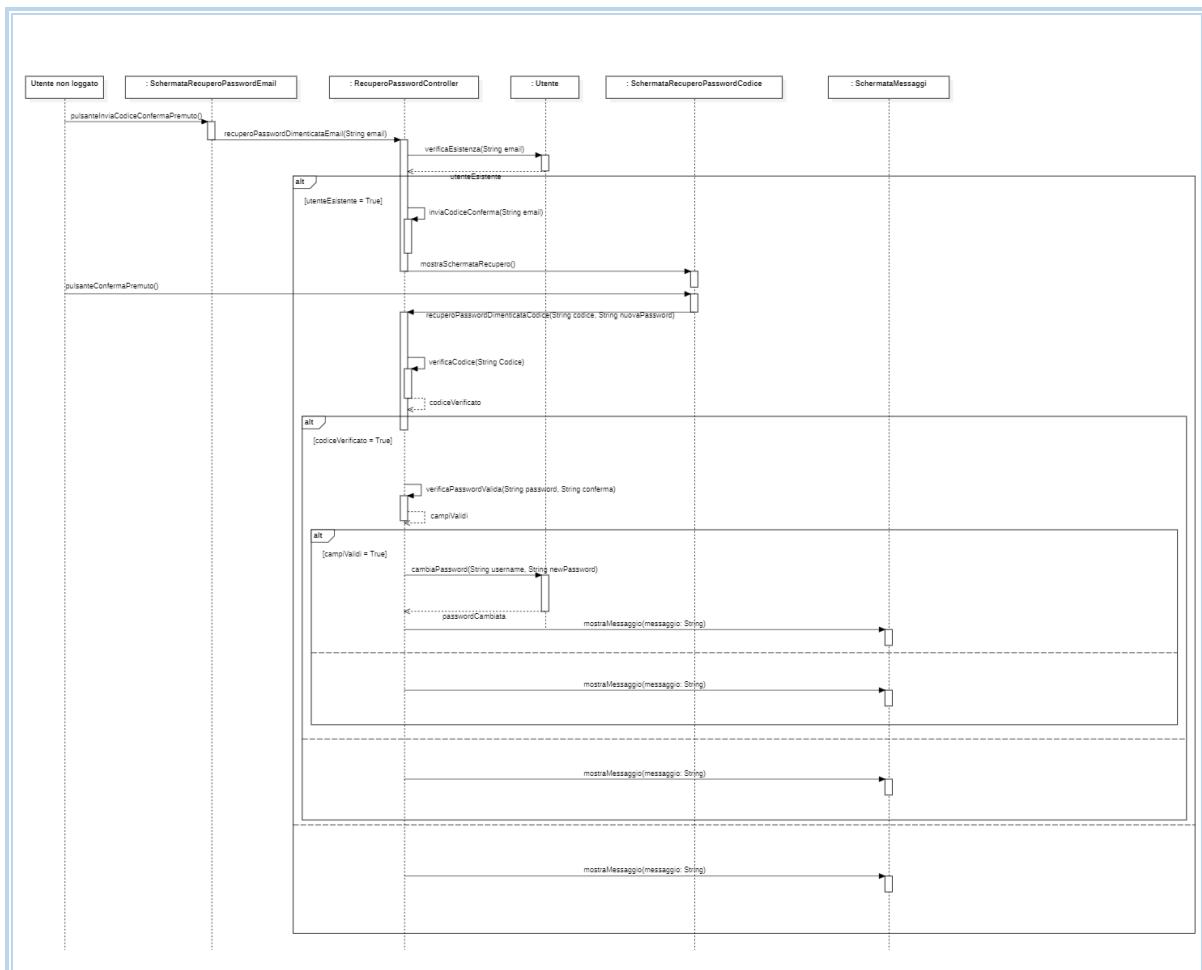
1.4.2.1 Accesso al sistema



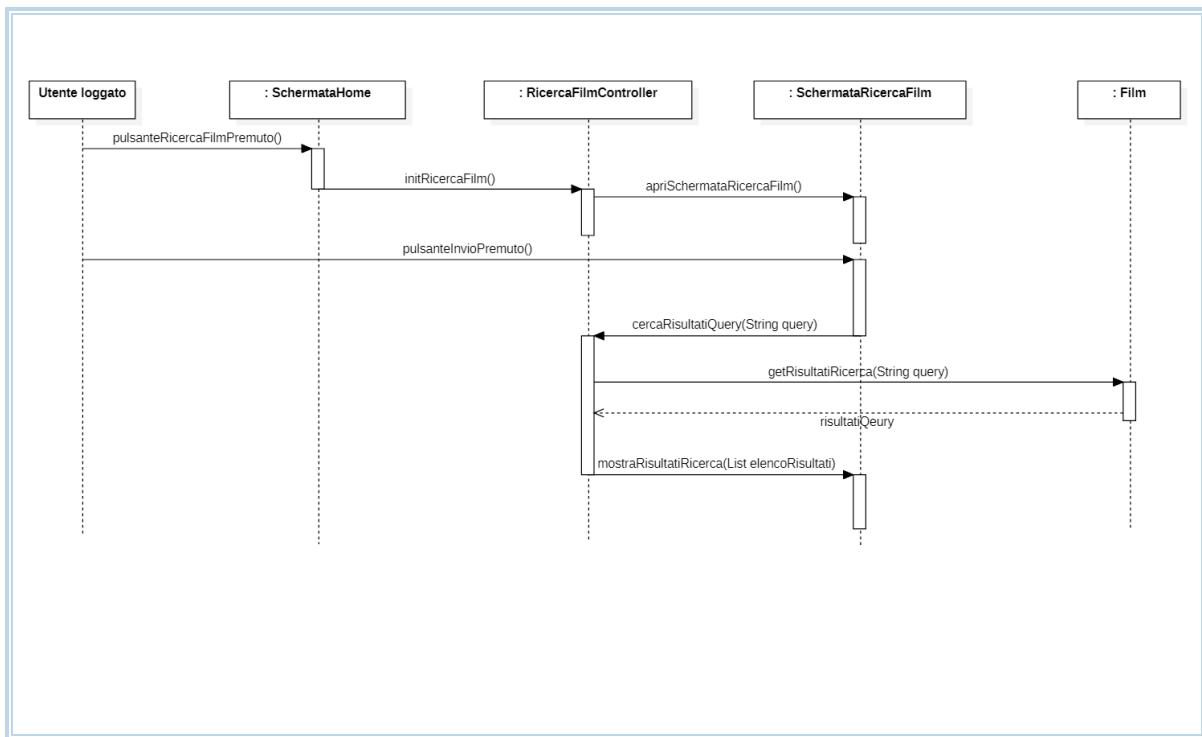
1.4.2.2 Registrazione al sistema



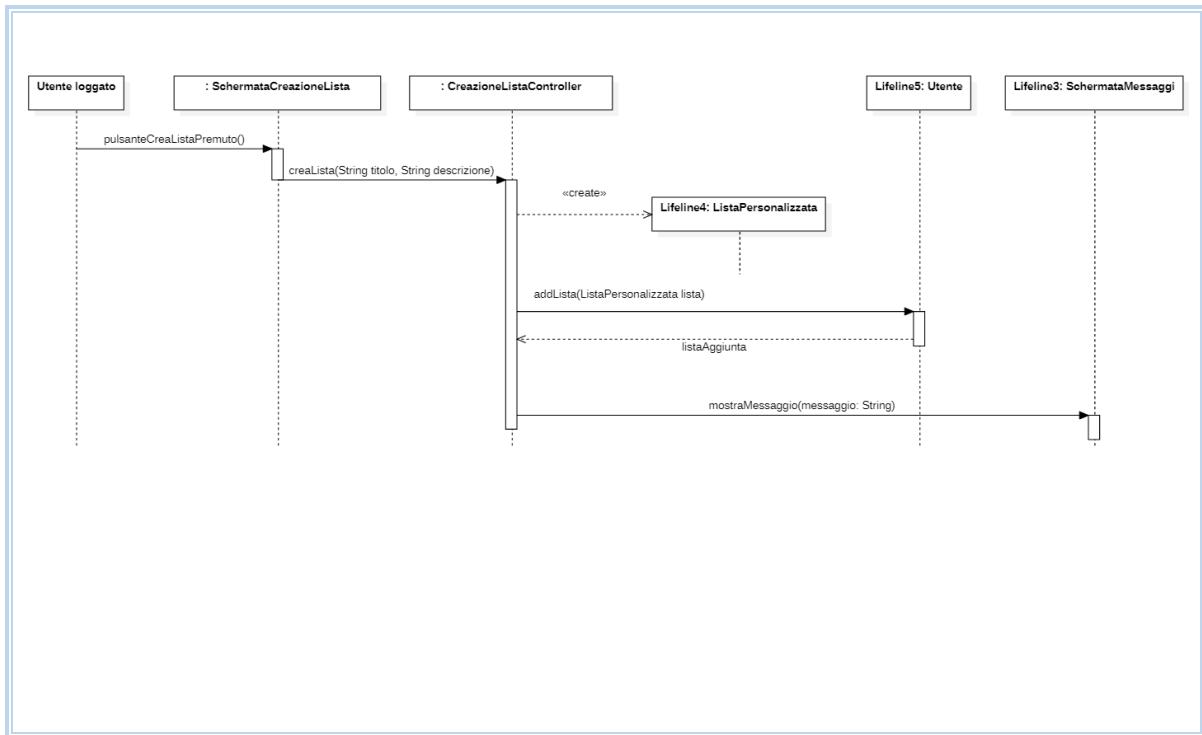
1.4.2.3 Recupero password dimenticata



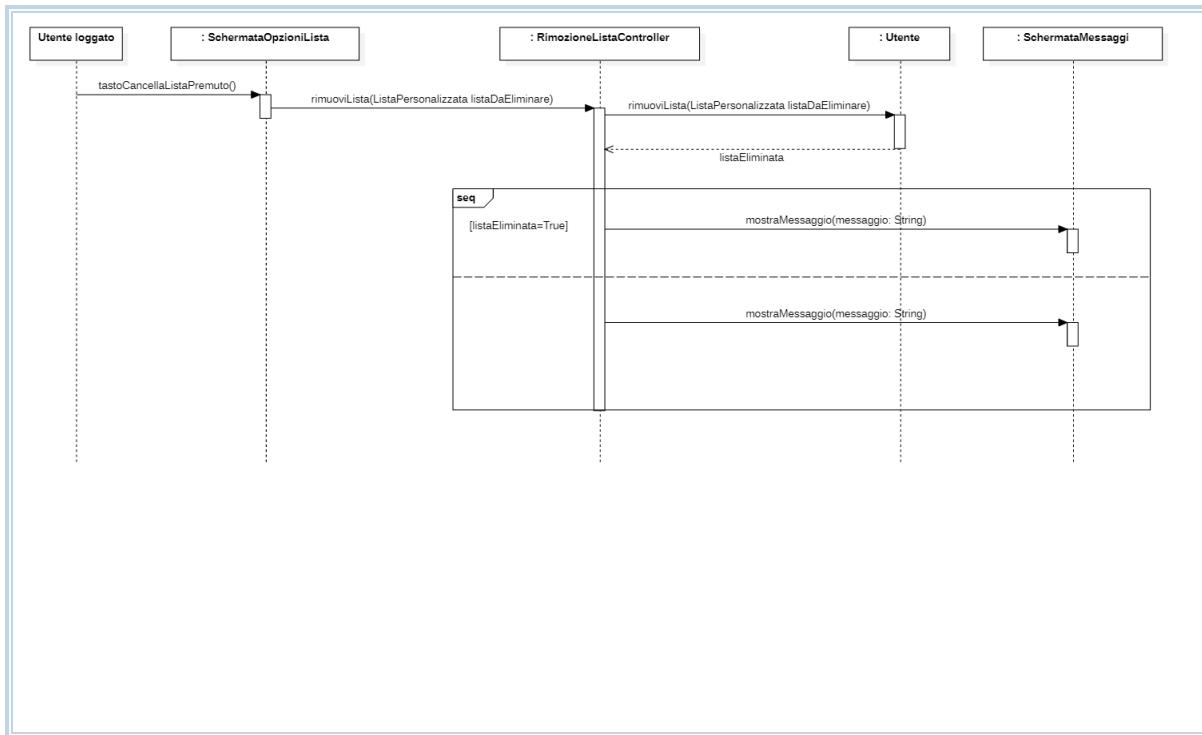
1.4.2.4 Effettuare ricerche film



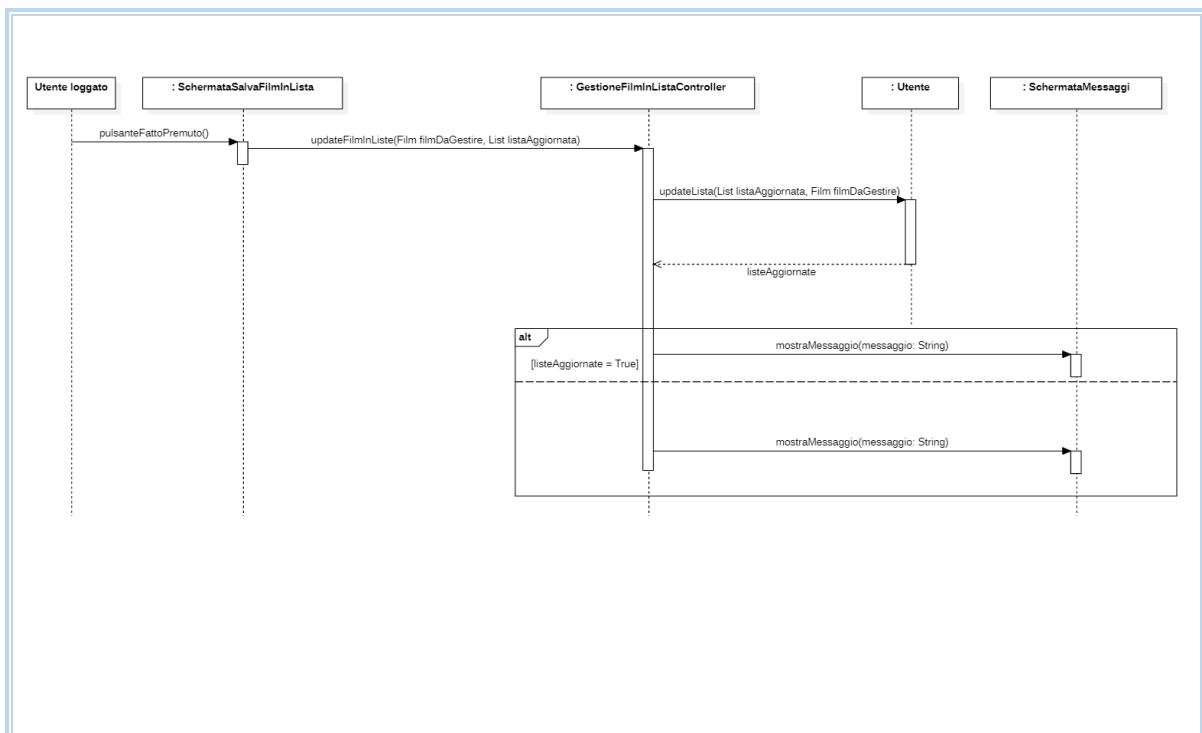
1.4.2.5 Creazione di una lista personalizzata



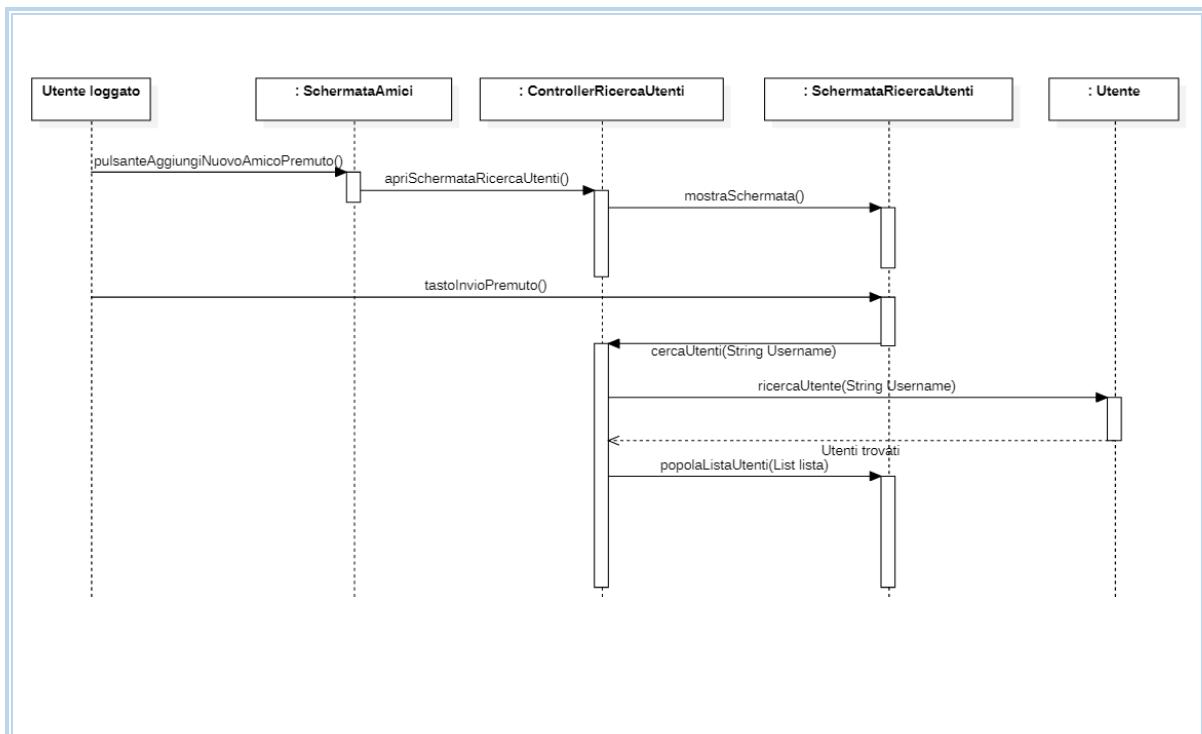
1.4.2.6 Rimozione di una lista personalizzata



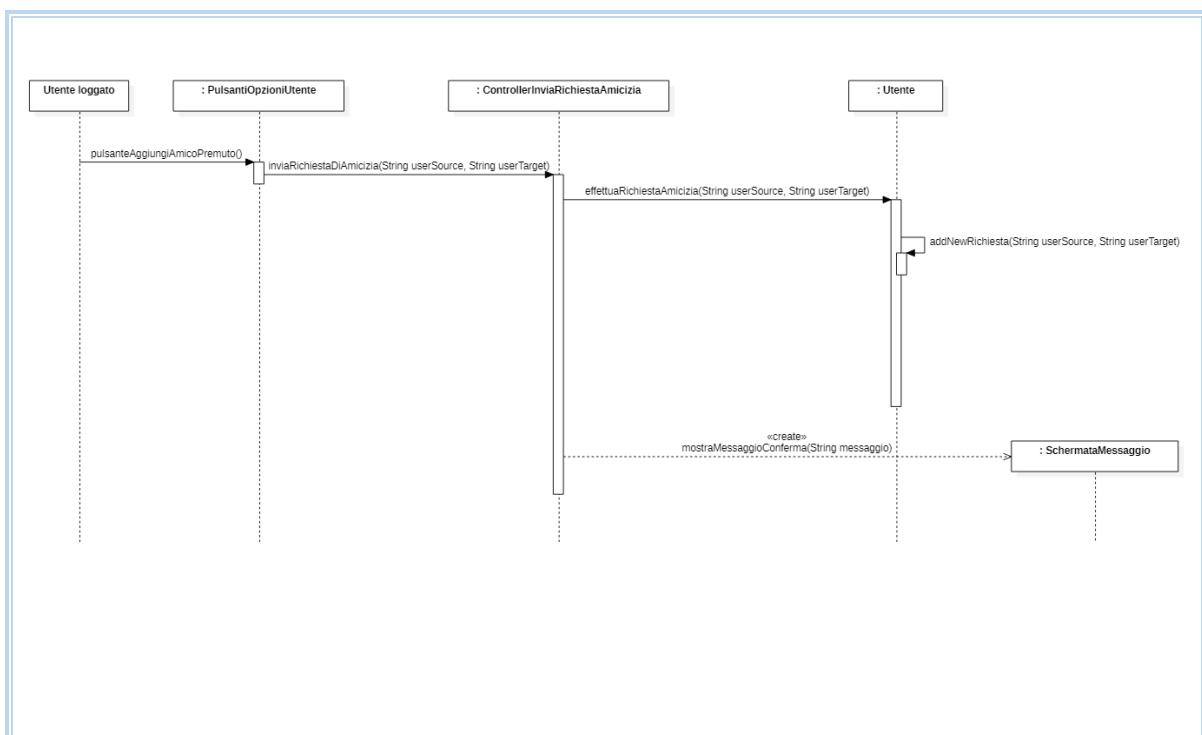
1.4.2.7 Aggiungere e rimuovere film da una lista personalizzata



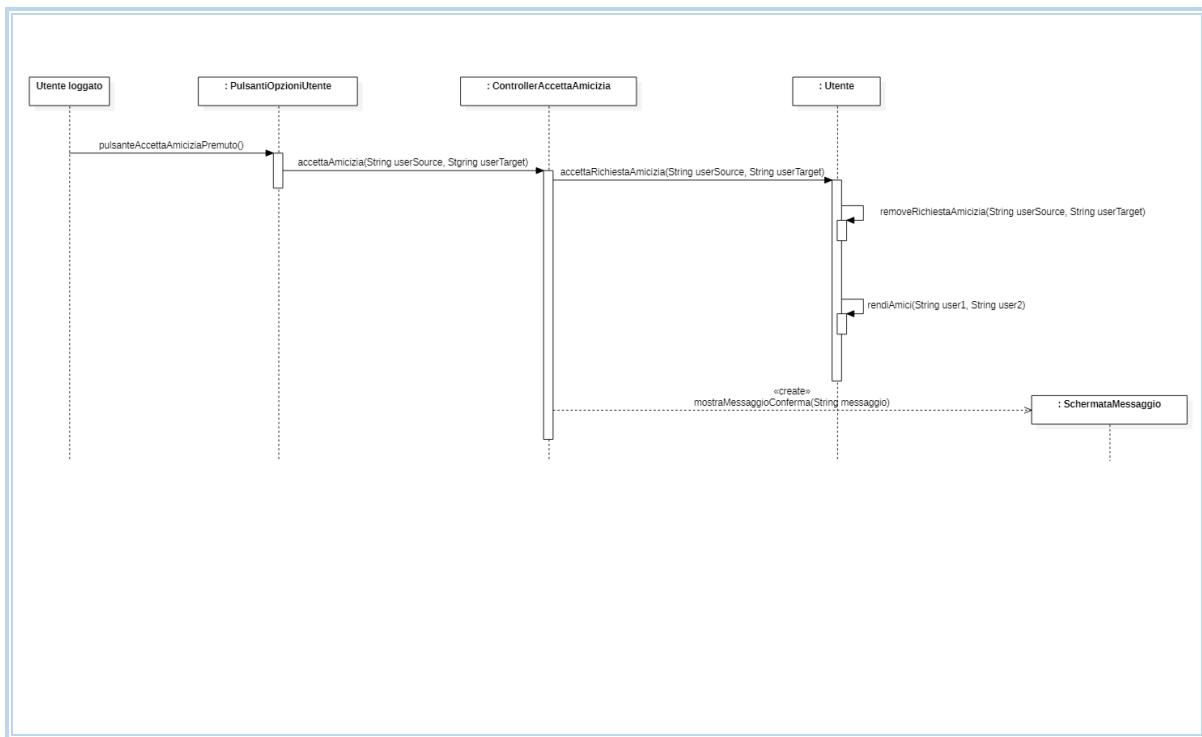
1.4.2.8 Effettuare ricerche di altri utenti



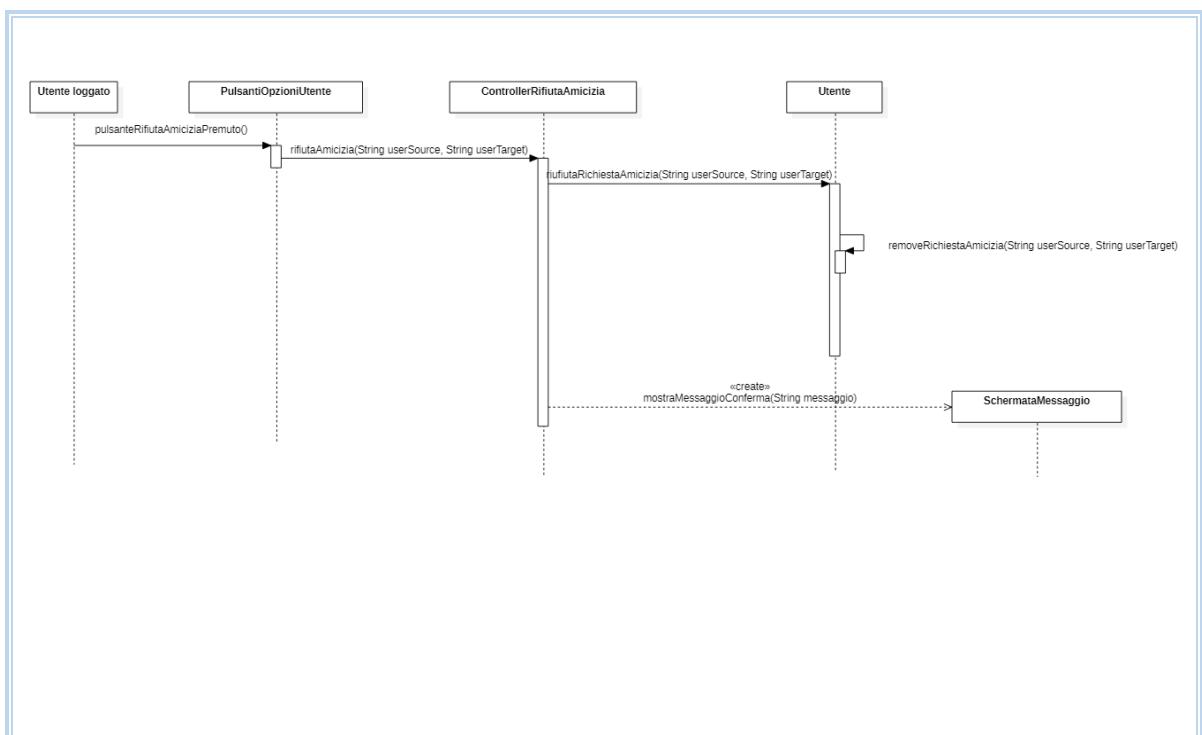
1.4.2.9 Inviare richieste di amicizia



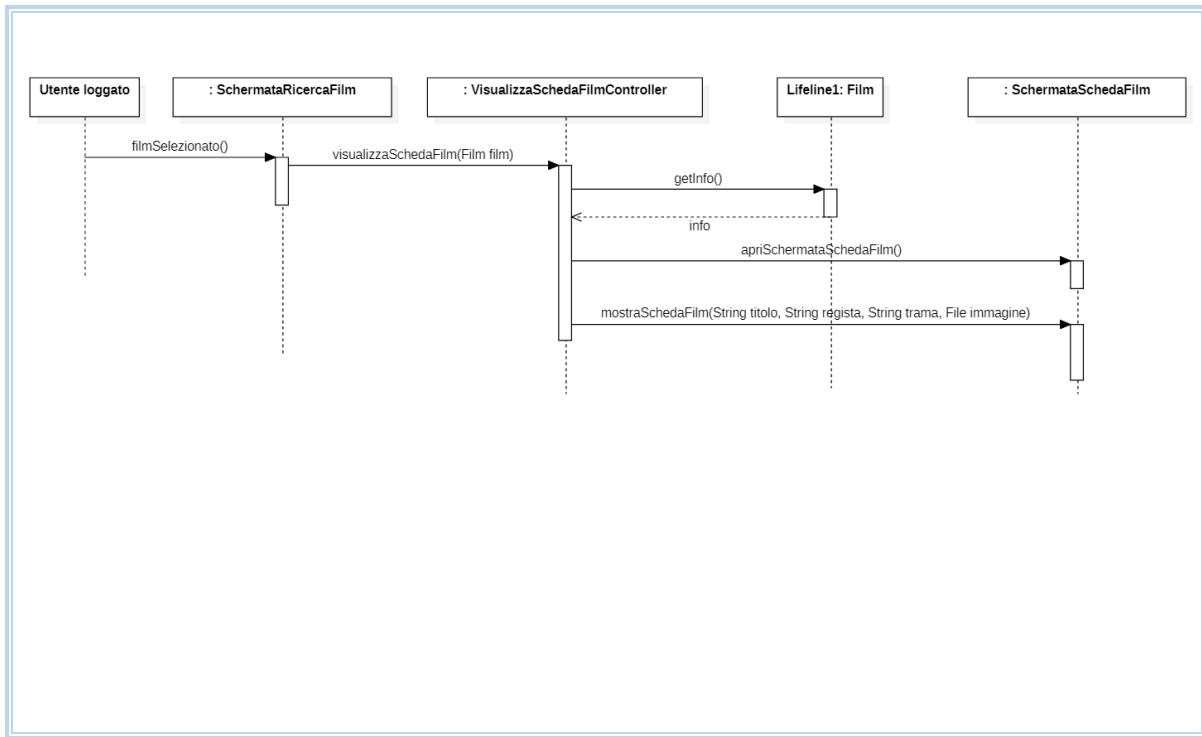
1.4.2.10 Accettare richieste di amicizia



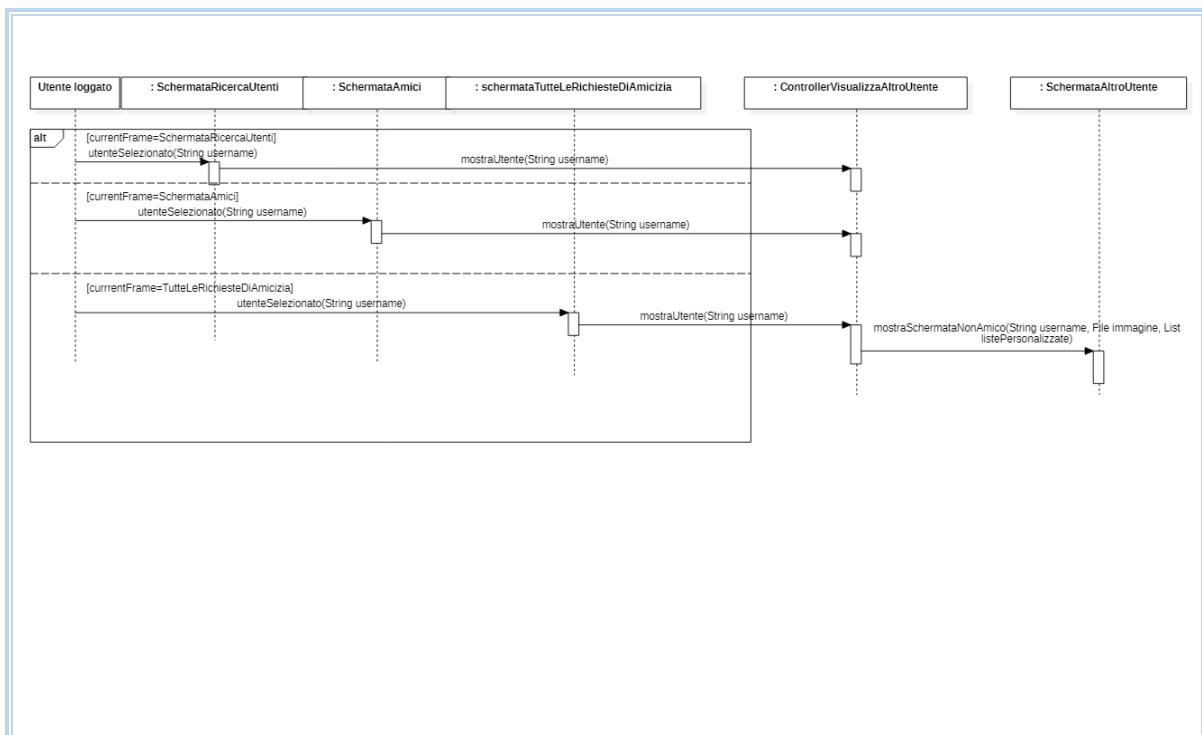
1.4.2.11 Rifiutare richieste di amicizia



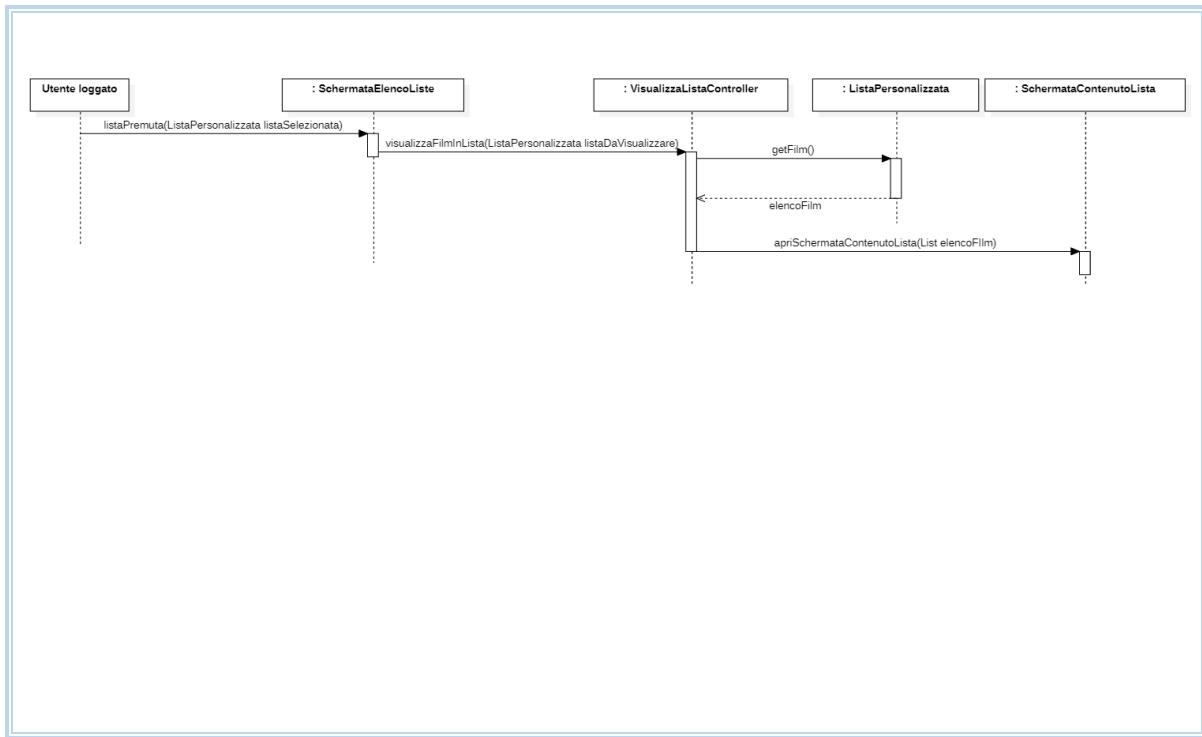
1.4.2.12 Visualizzazione della scheda di un film



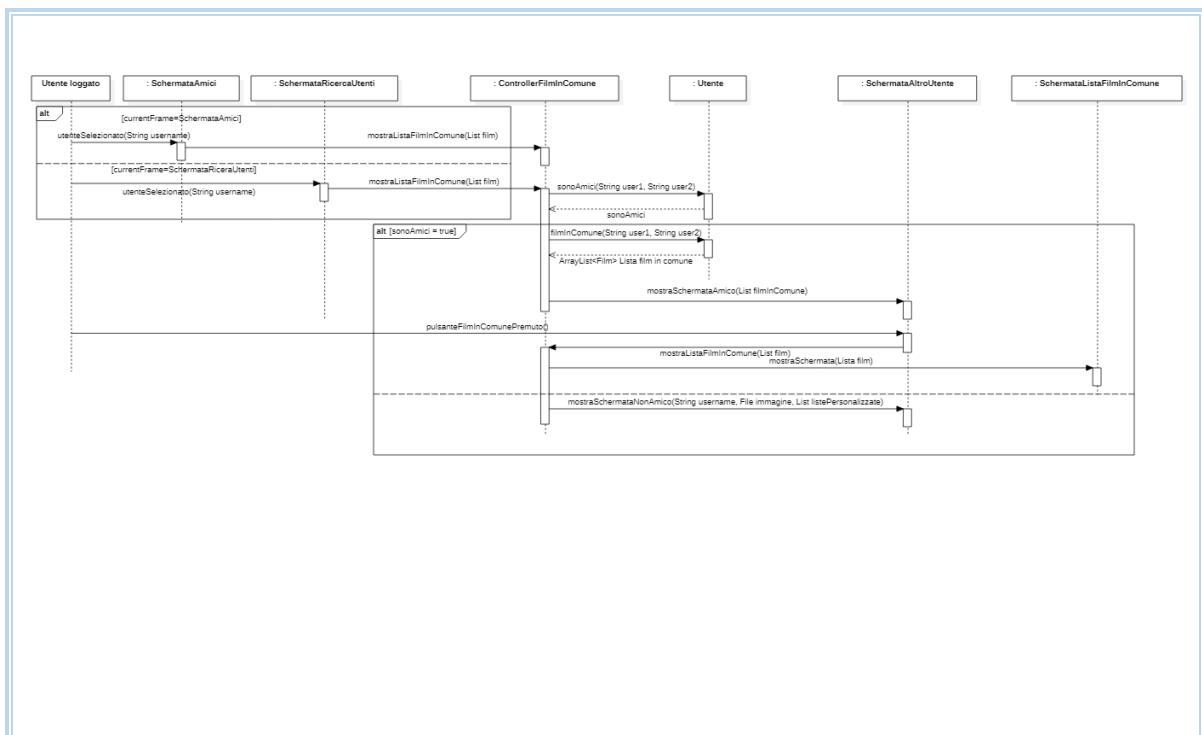
1.4.2.13 Visualizzazione della scheda utente



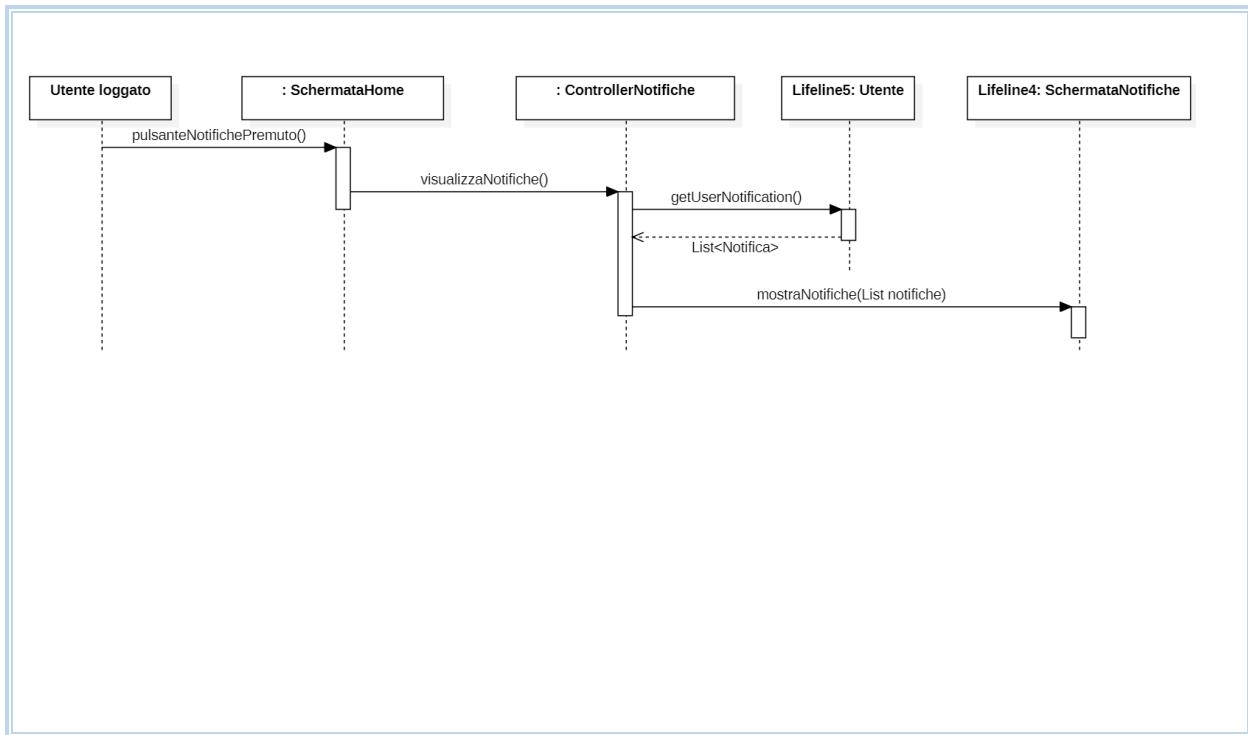
1.4.2.14 Visualizzazione di una lista



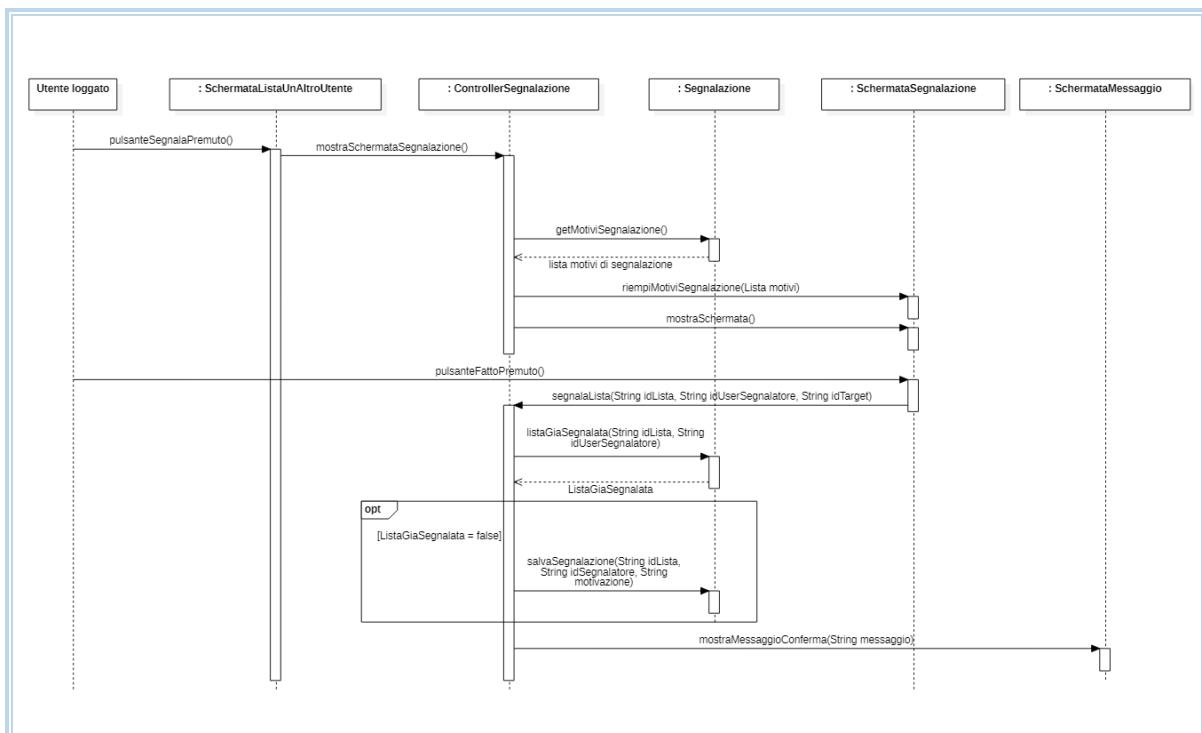
1.4.2.15 Visualizzazione dei film in comune con un amico



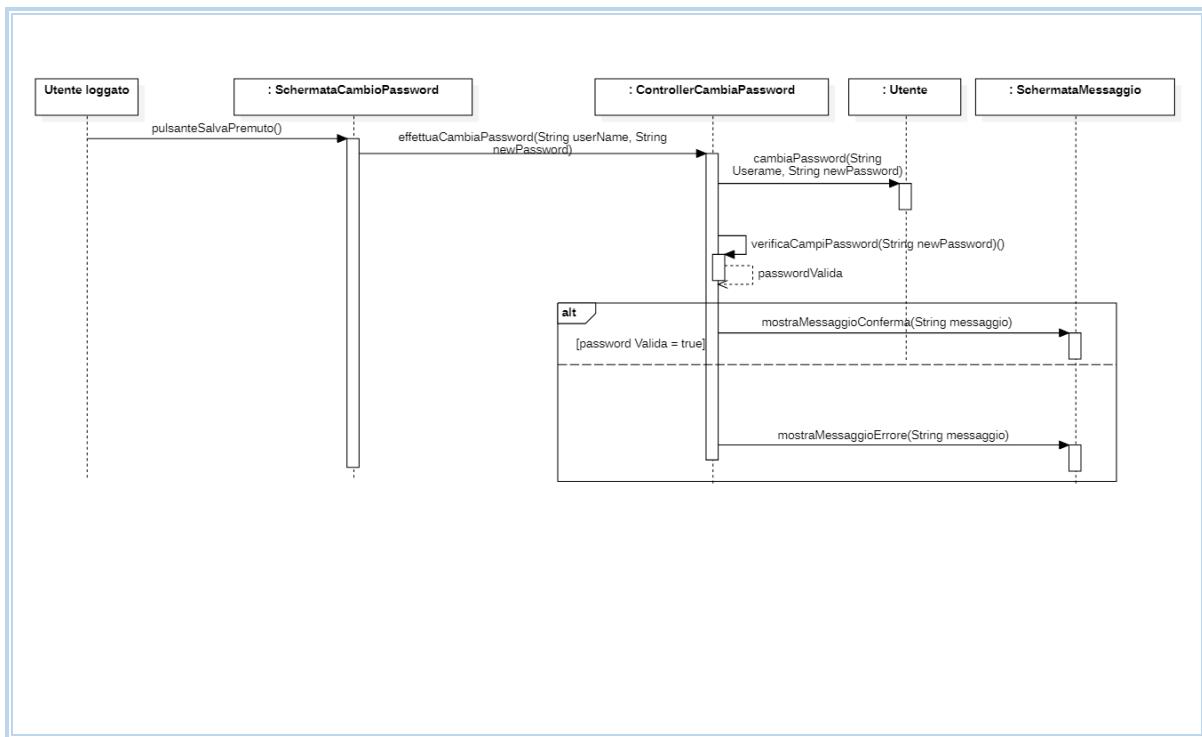
1.4.2.16 Visualizzazione delle notifiche



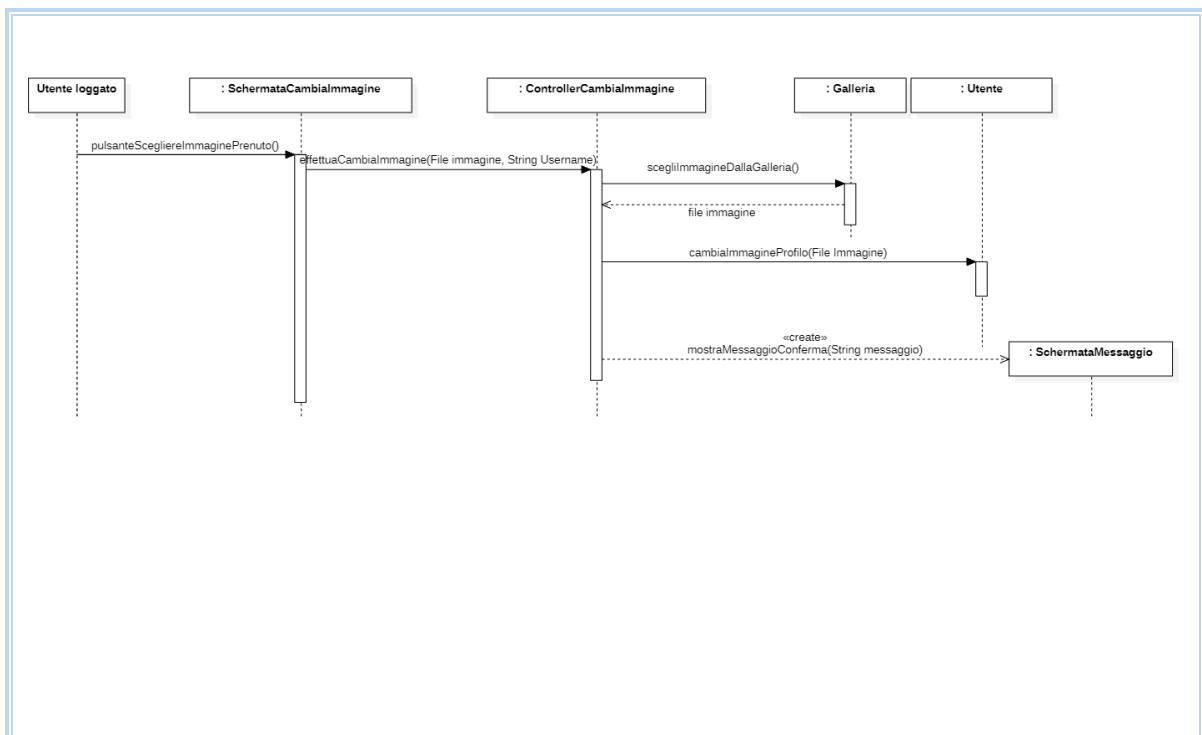
1.4.2.17 Effettuare una segnalazione



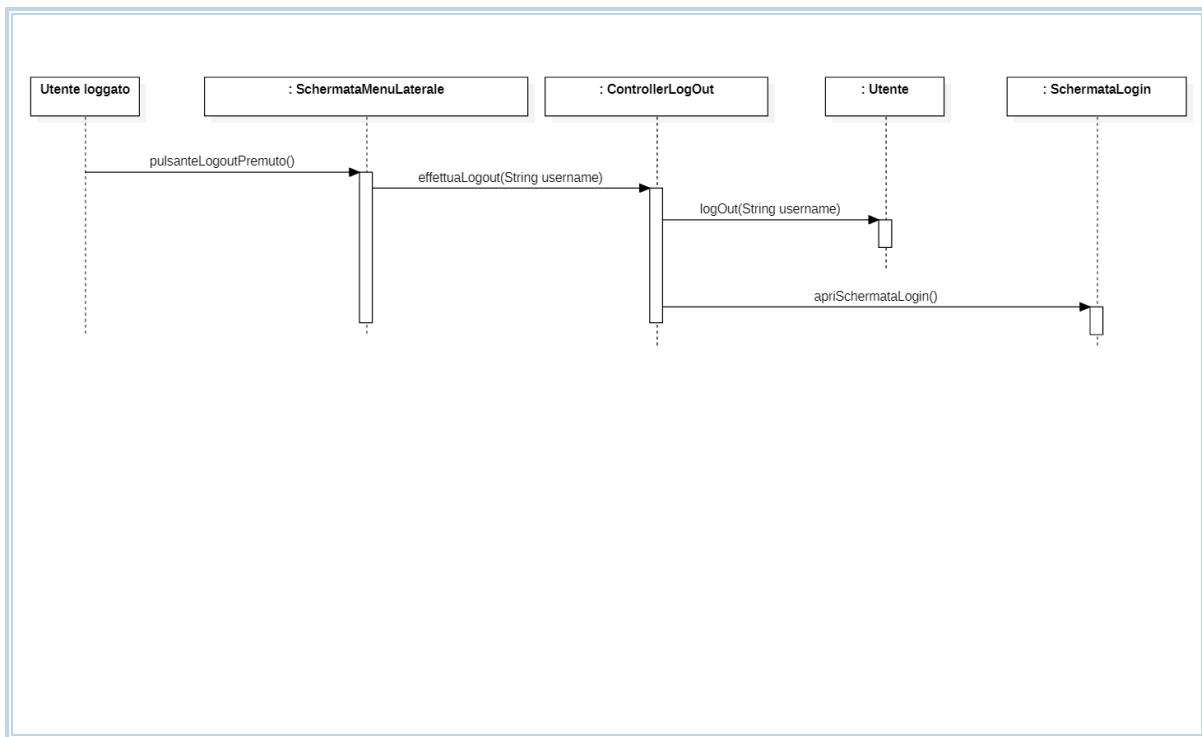
1.4.2.18 Cambiare password



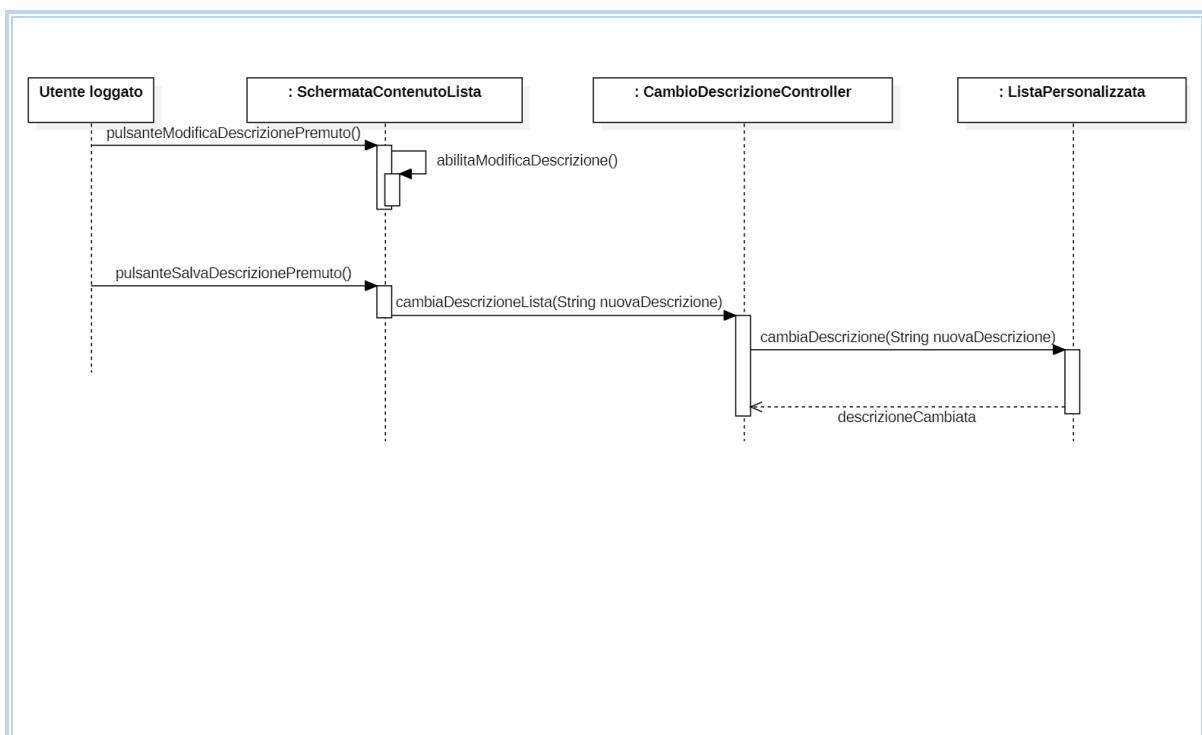
1.4.2.19 Cambiare immagine di profilo



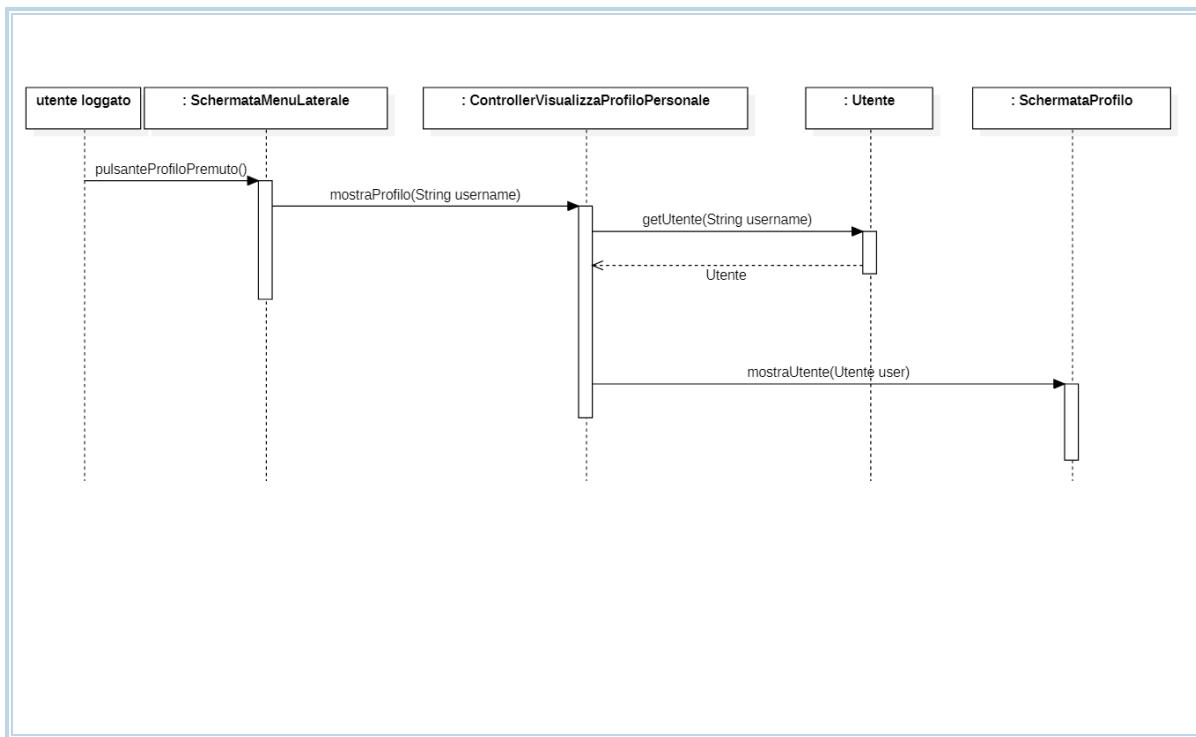
1.4.2.20 Uscire dal profilo



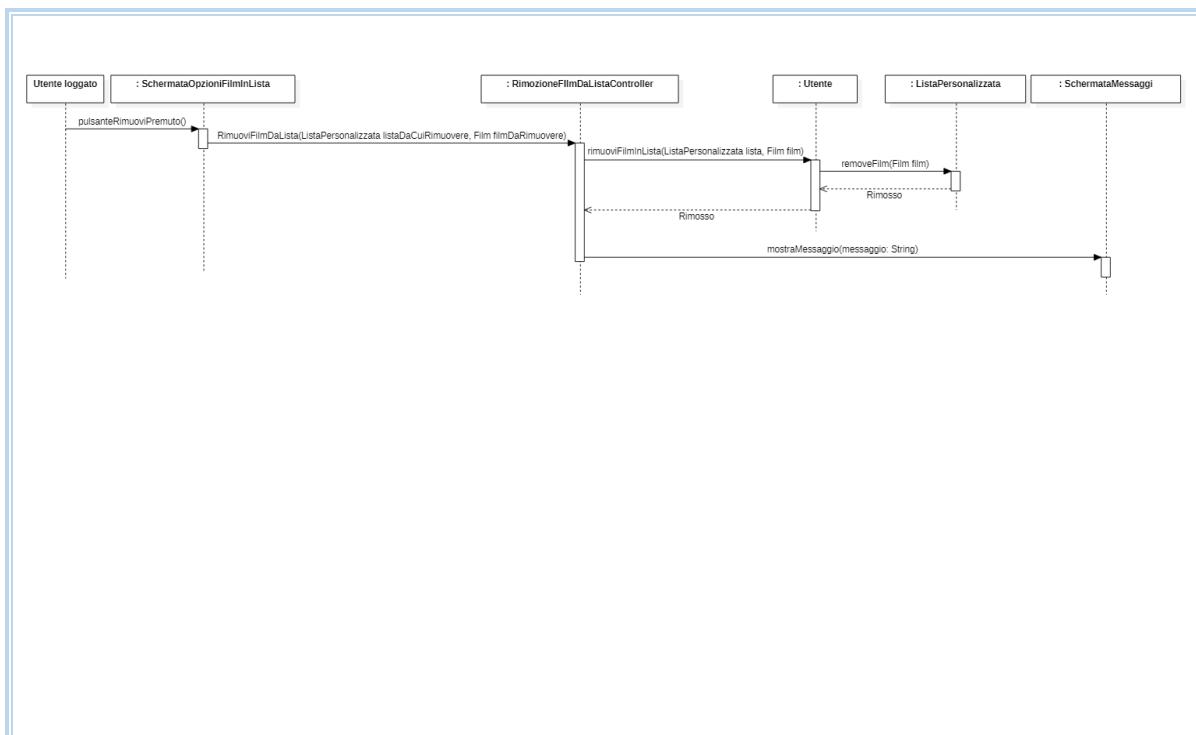
1.4.2.21 Modificare la descrizione di una lista



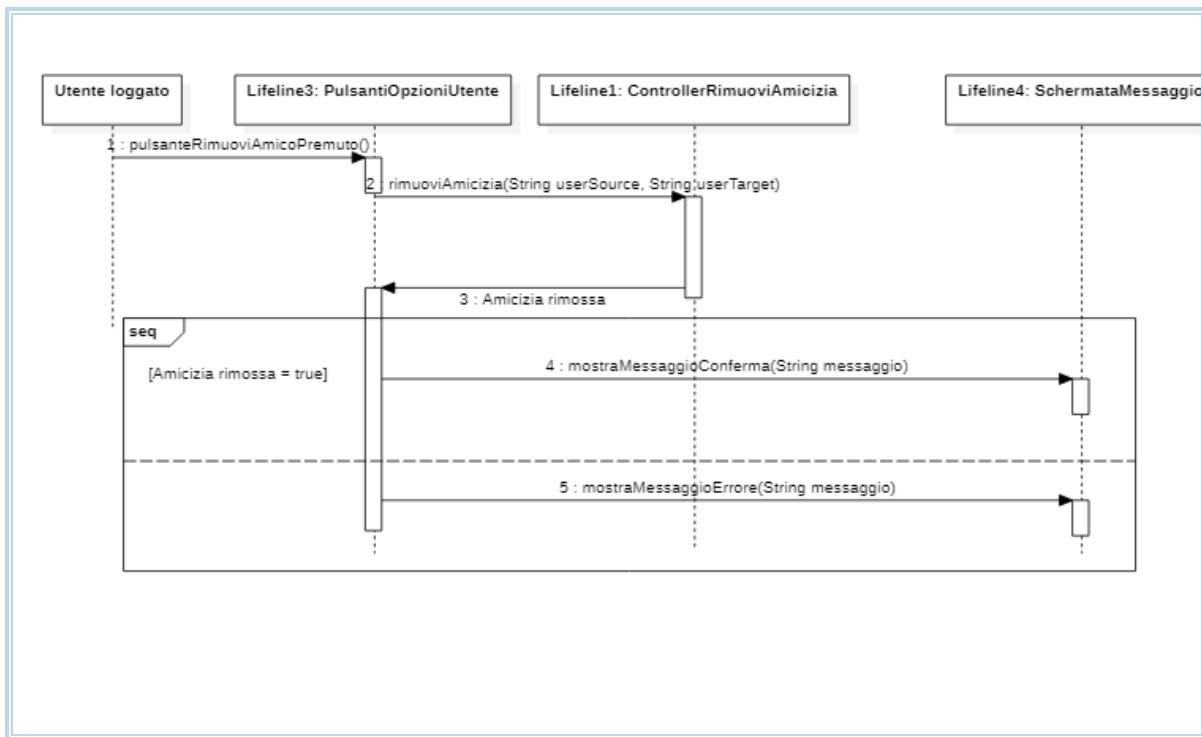
1.4.2.22 Visualizzazione del proprio profilo



1.4.2.23 Rimozione di un film da una lista (2° modo)

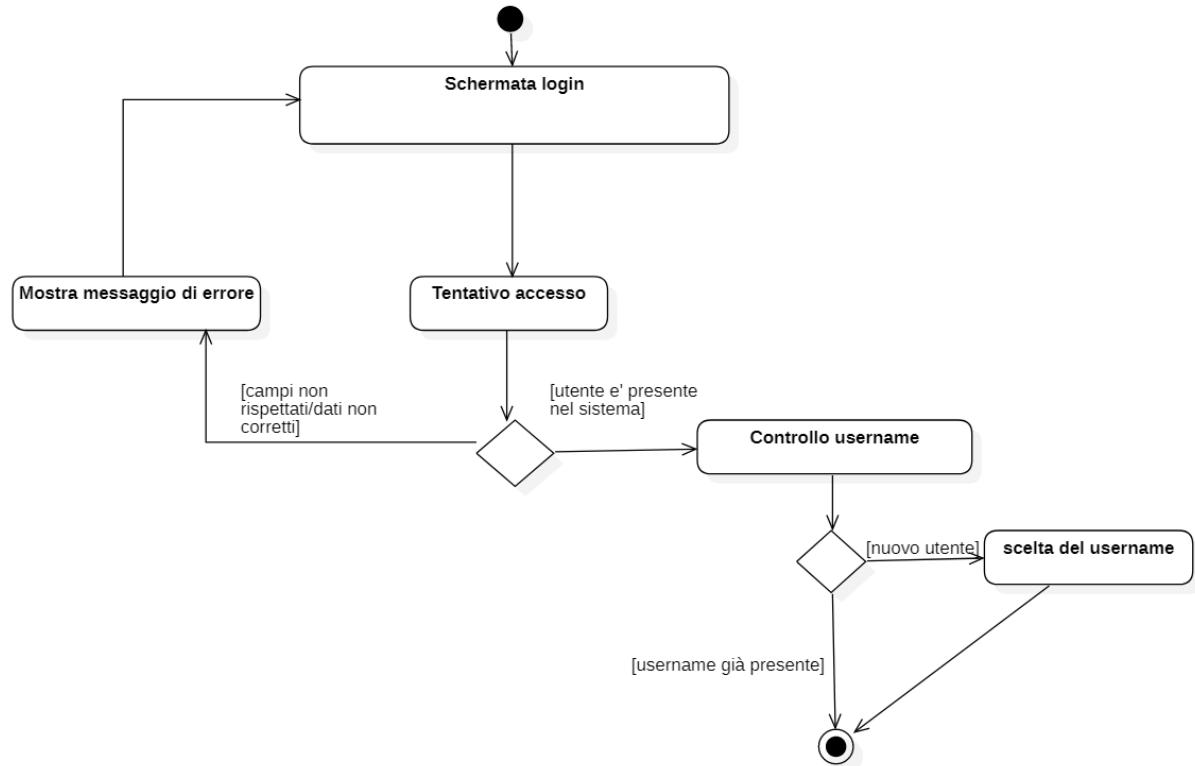


1.4.2.24 Rimozione dell'amicizia



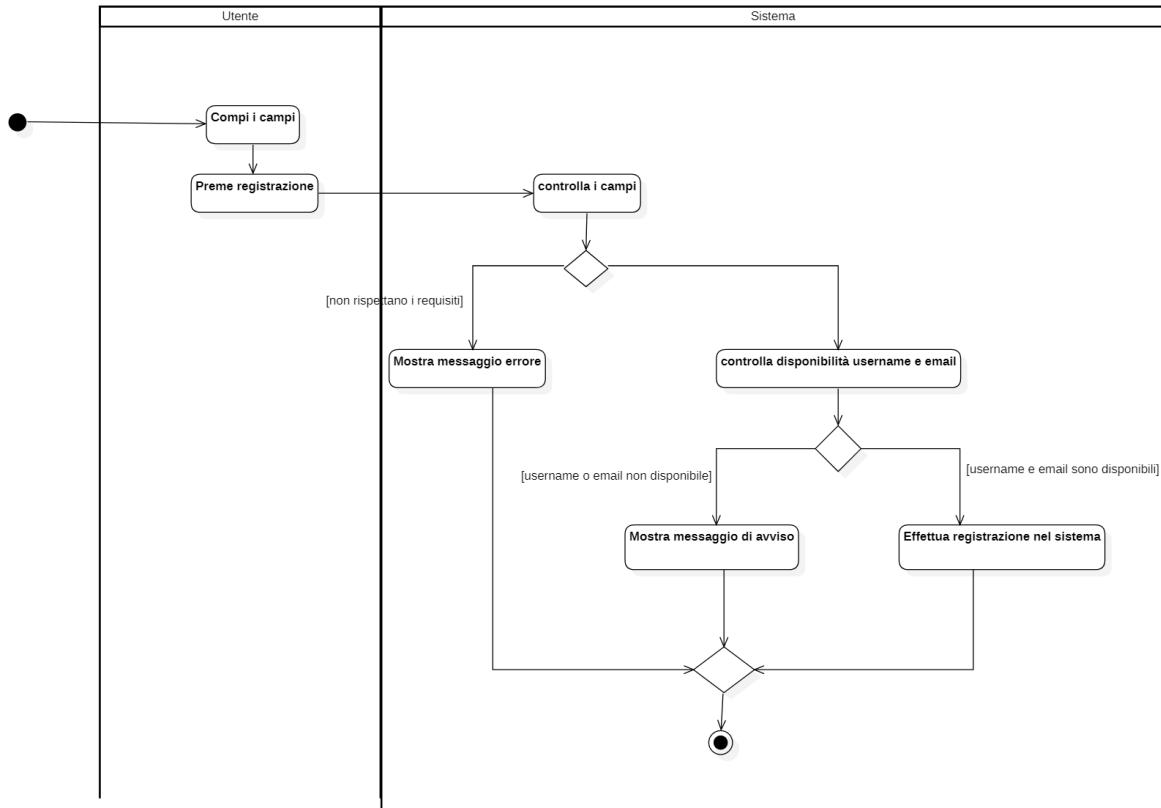
1.4.3 Diagrammi di stato di analisi

1.4.3.1 Accesso al sistema



1.4.4 Diagrammi di attività

1.4.4.1 Registrazione al sistema



1.4.5 Diagramma di Giant



1.5 Modello di Dominio: applicazione Desktop

1.5.1 Modello a Oggetti di Analisi

Di seguito verranno rappresentati i Class Diagram con opportune relazioni tra le classi al fine di mostrare visualmente i concetti principali relativi al dominio.

Il metodo usato al fine di rintracciare gli elementi principali è quello dell'euristica Three-Object-Type, cioè pattern *Entity-Boundary-Control* (EBC), le informazioni riguardo questo pattern sono reperibili sul sito di Wikipedia (<https://en.wikipedia.org/wiki/Entity-control-boundary>) o su qualunque altro sito Internet.

In sintesi, pattern è costituito da tre oggetti:

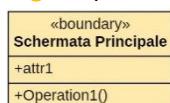
- *Entity* – modellano l'informazione persistente
- *Boundary* - modellano le interazioni tra gli attori e il sistema
- *Control* - modellano la logica che si occupa di realizzare gli Use case

Le possibili relazioni tra questi tre oggetti sono rappresentate dalla seguente tabella:

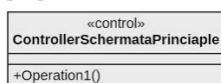
| Relazione tra gli oggetti | | | | |
|---------------------------|-------|------------|---------|--------|
| Ruolo | Actor | Boundary | Control | Entity |
| Actor | Si | Si | No | No |
| Boundary | Si | Parz/Pieno | Si | No |
| Control | No | Si | Si | Si |
| Entity | No | No | Si | Si |

Al fine di facilitare la lettura sono stati assegnati i colori a ogni oggetto dentro il Class Diagram

- Boundary (Colore giallo)



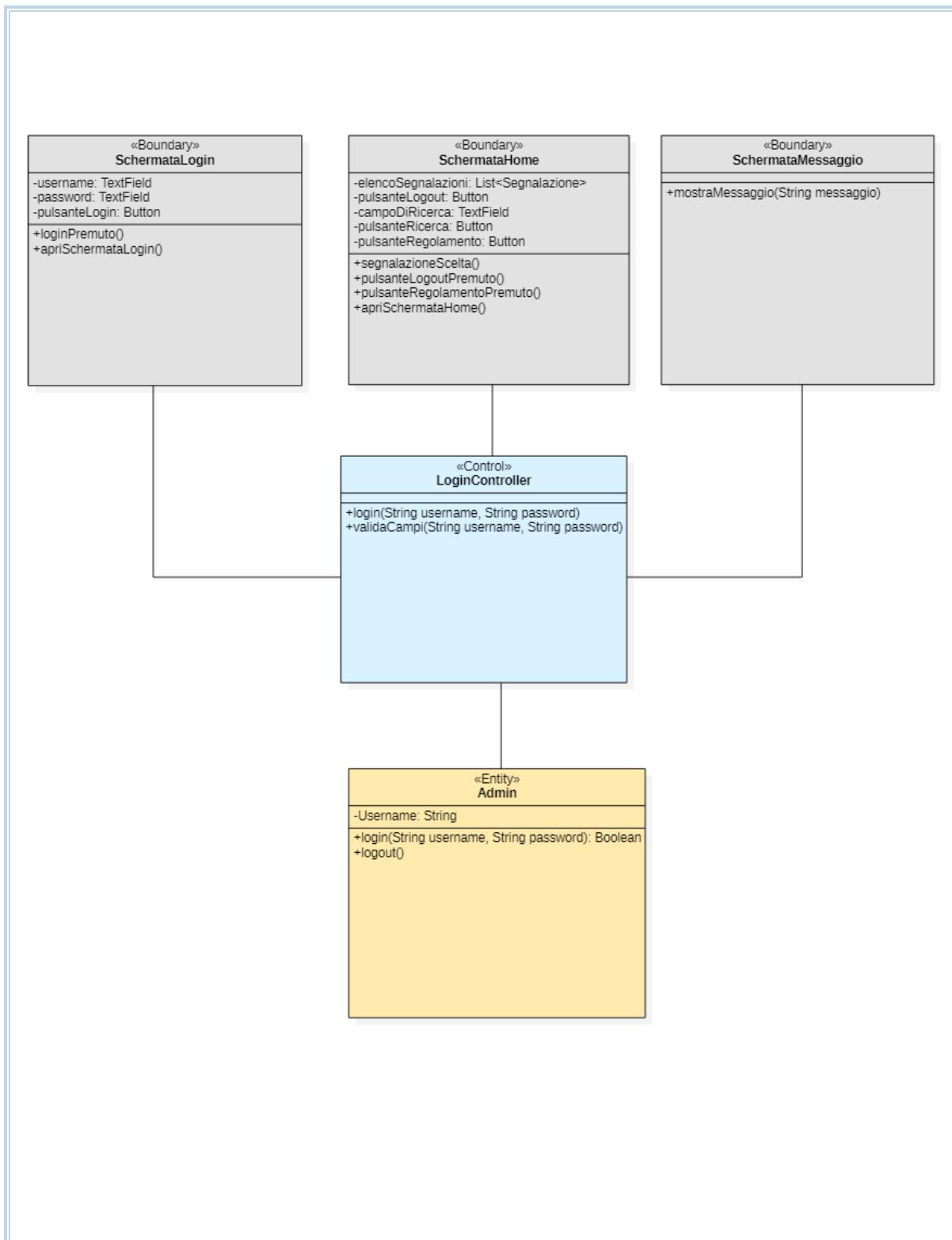
- Control (Colore grigio)



- Utente (Colore azzurro)

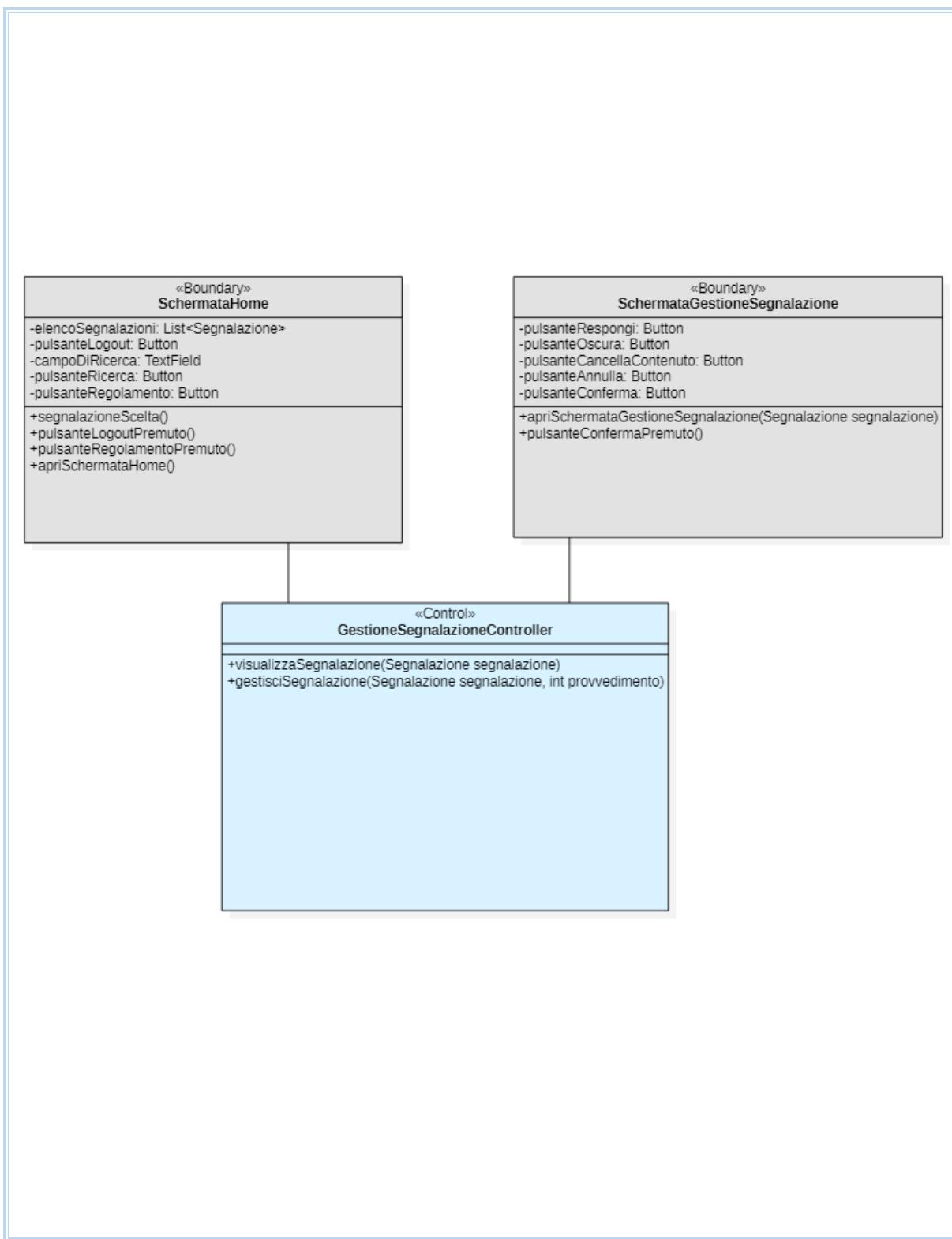


1.5.1.1 Accesso al sistema



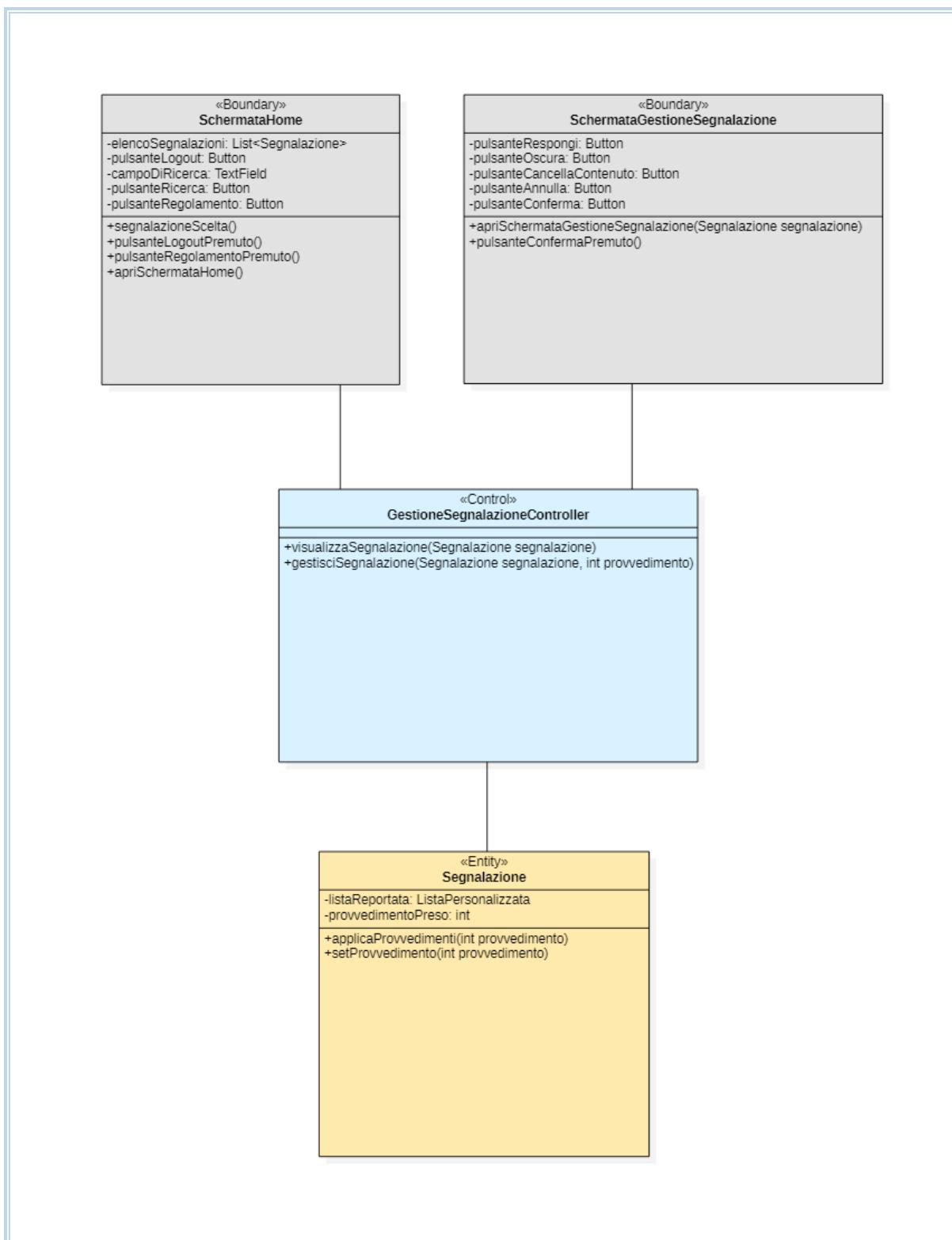
1.5.1.2 Visualizzare una segnalazione





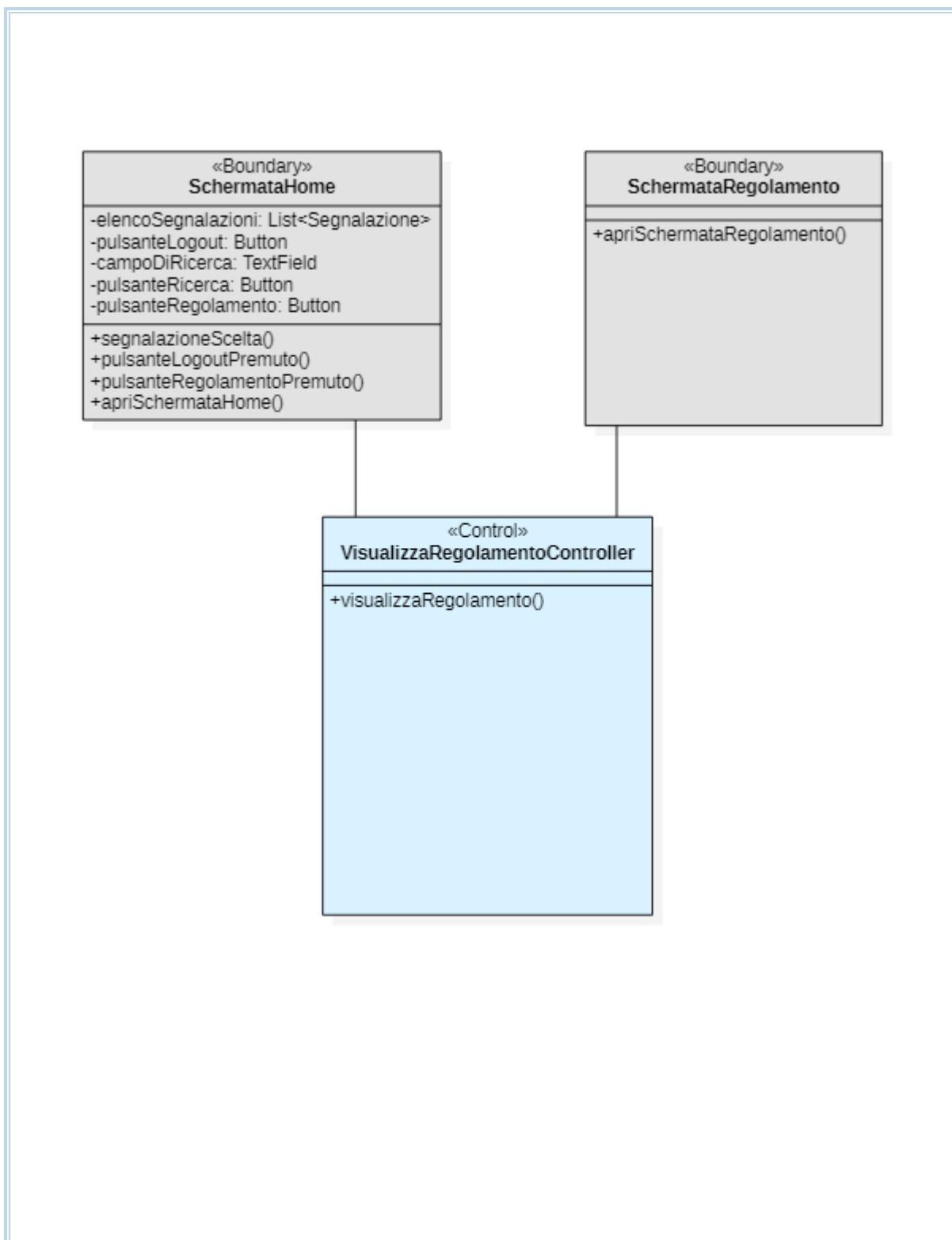
1.5.1.3 Gestire una segnalazione





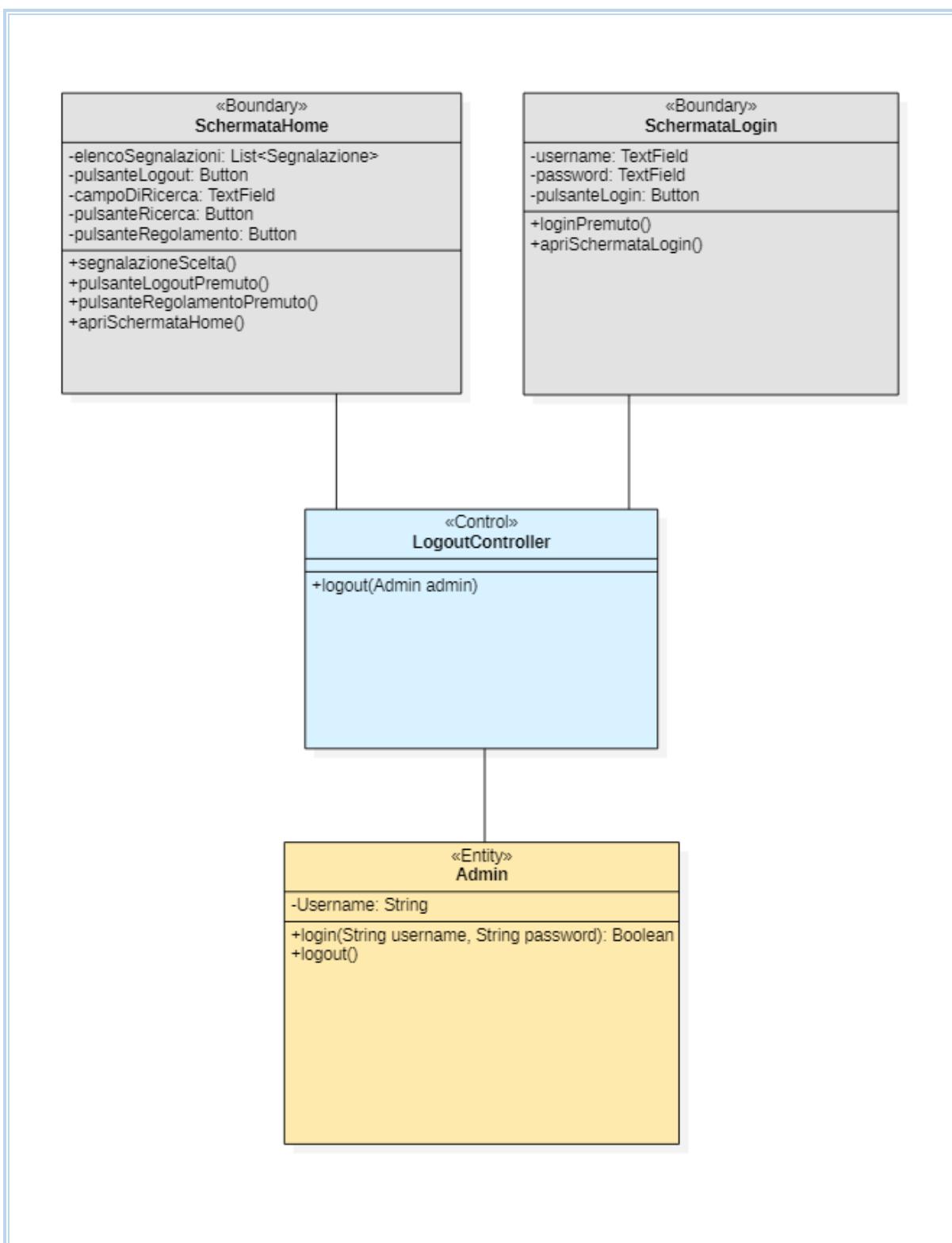
1.5.1.4 Visualizzare il regolamento





1.5.1.5 Effettuate logout



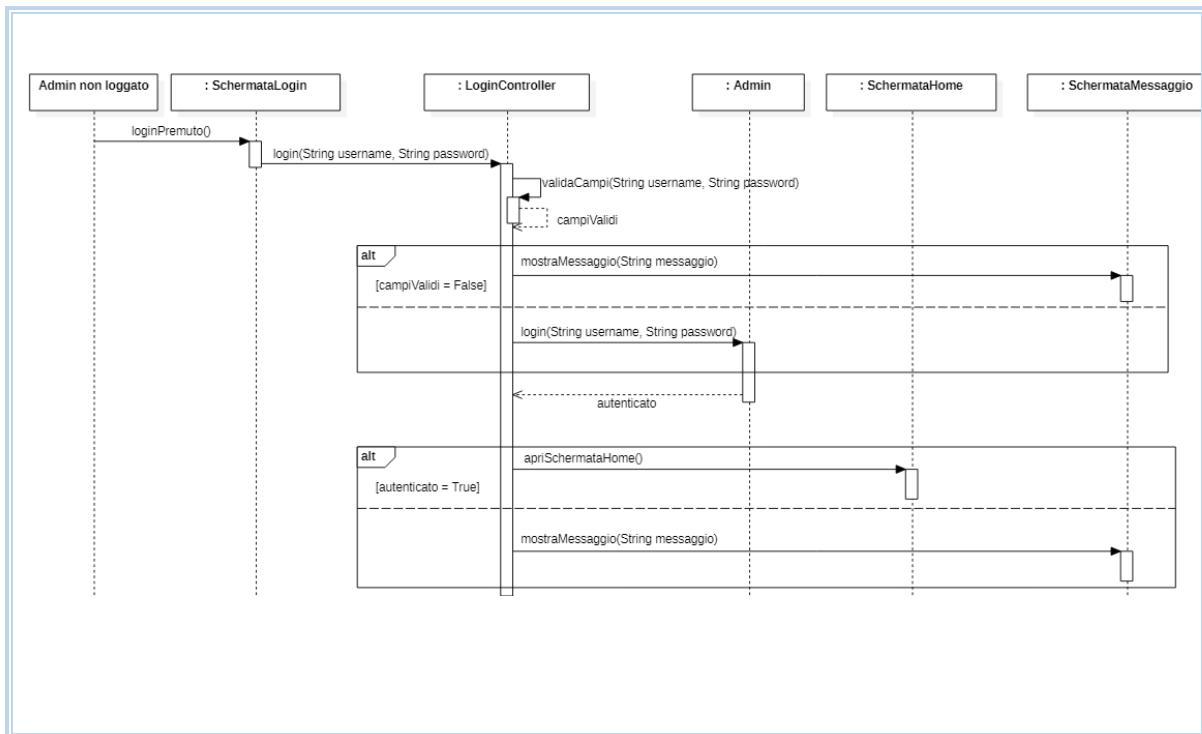


1.5.2 Diagrammi di sequenza di analisi

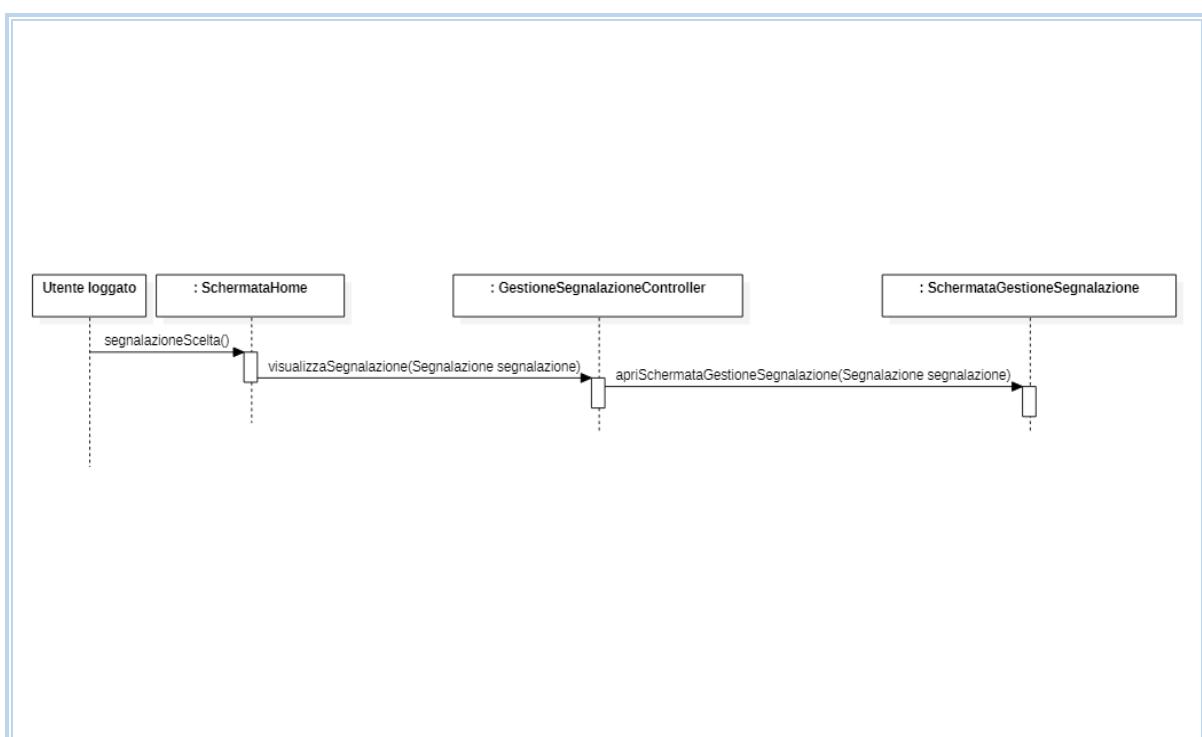
Di seguito sono riportati i diagrammi di sequenza di analisi, il loro funzionamento è quello di mostrare la relazione che intercorre tra le classi mostrate nei class diagram della sezione precedente.



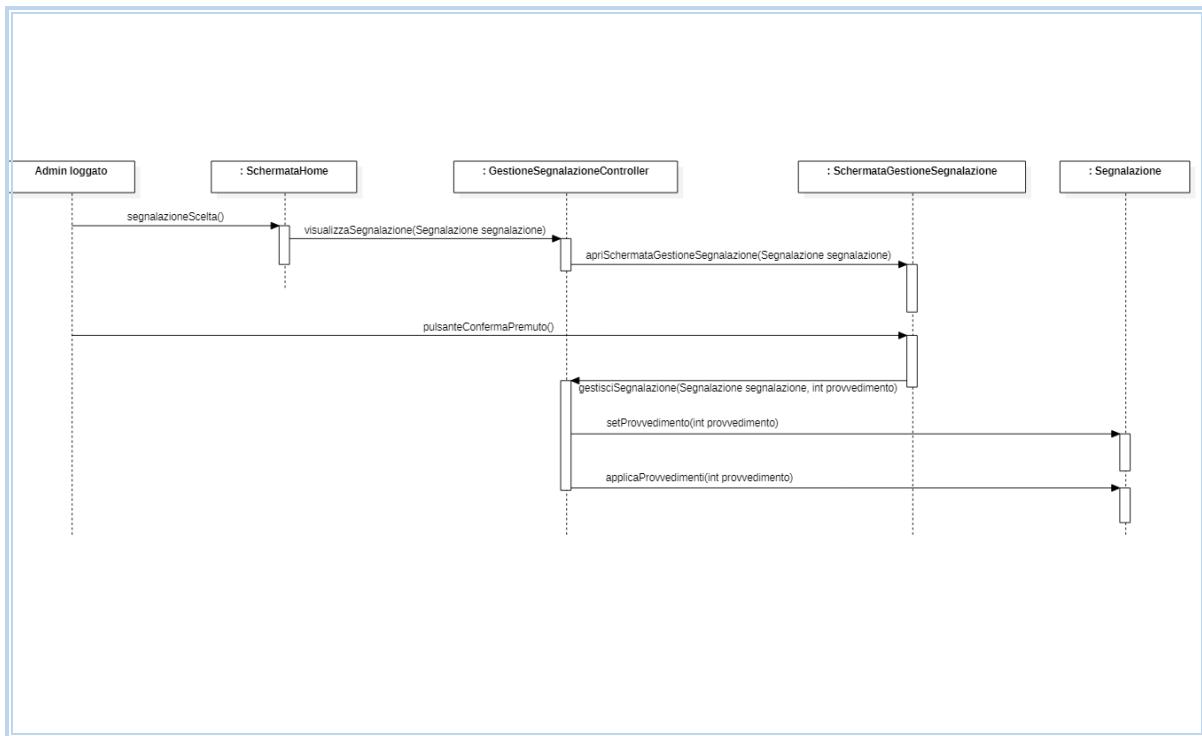
1.5.2.1 Accessi al sistema



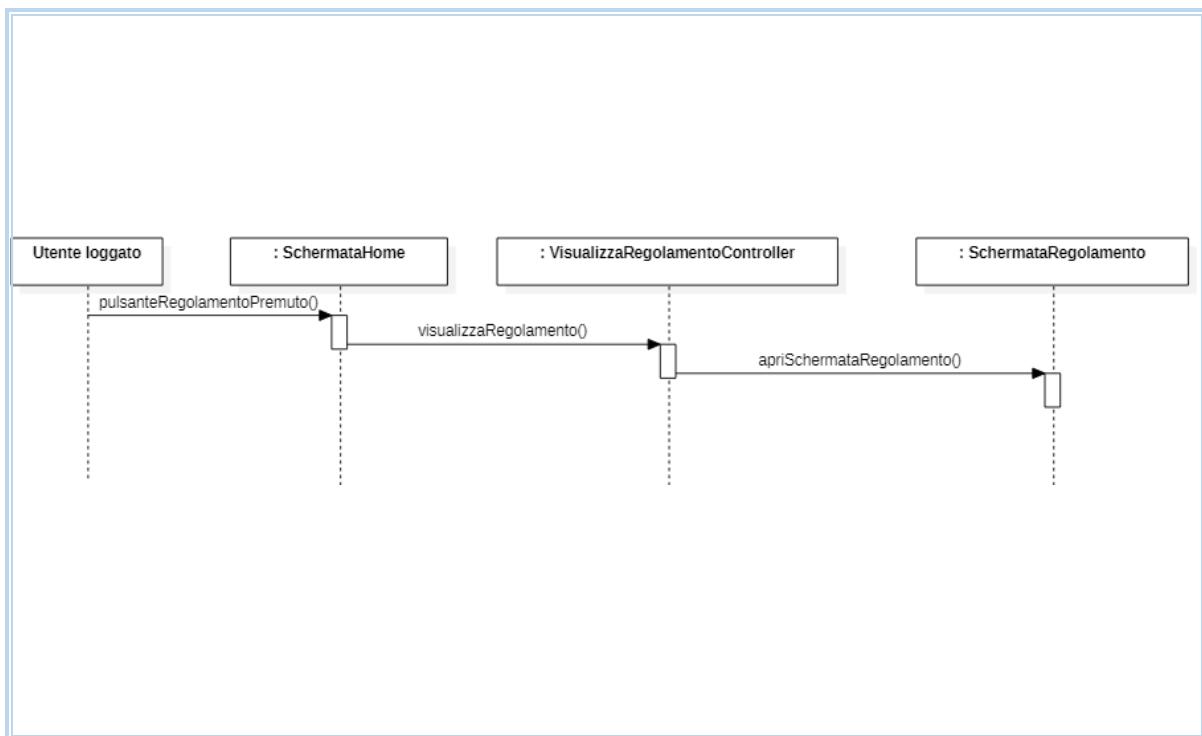
1.5.2.2 Visualizzare una segnalazione



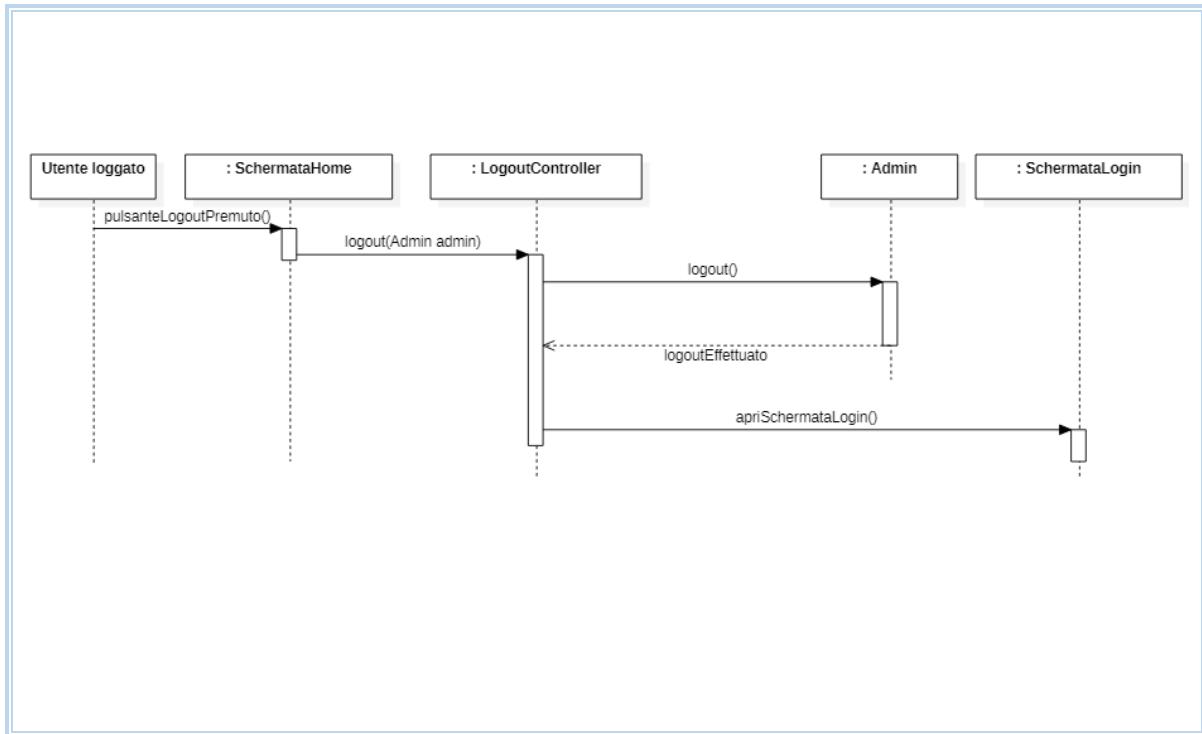
1.5.2.3 Gestire una segnalazione



1.5.2.4 Visualizzare il regolamento

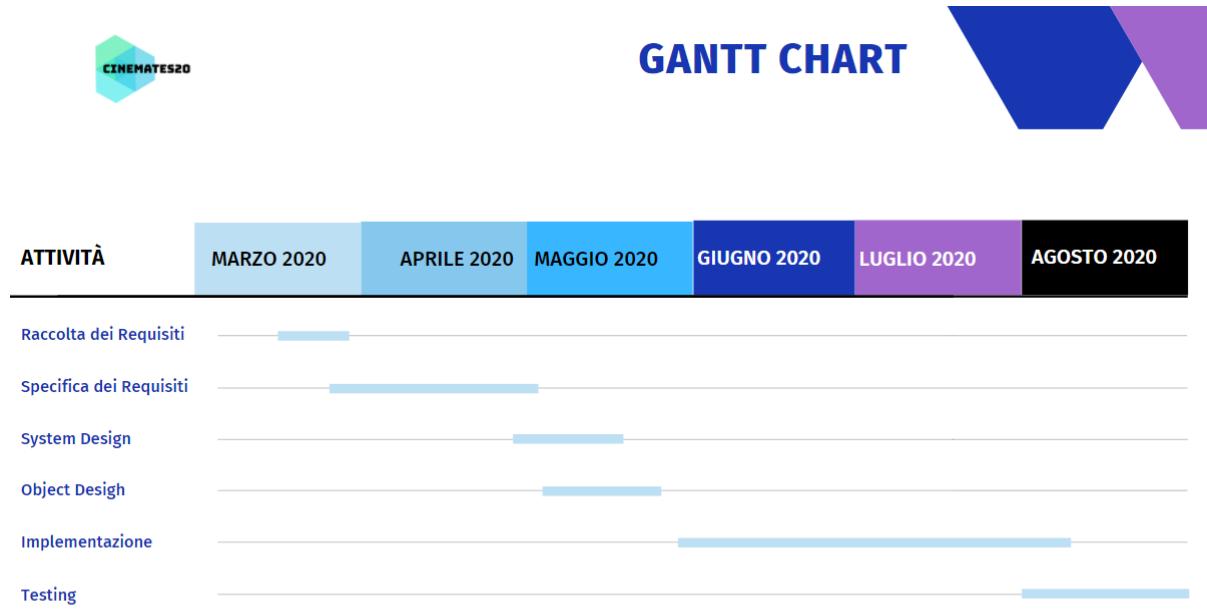


1.5.2.5 Effettuate logout



1.6 Proposta di impegno con diagramma di Gantt

Di seguito verrà illustrata la proposta di impegno delle risorse con la pianificazione delle attività espresse nell’arco di tempo.



Le risorse impegnate in questo caso sono esclusivamente i studenti Dmytro Lozyak e Carmine Testa. Tutte le fasi sono state discusse all’interno del Team che ha deciso di non stabilire i rigidi limiti per ogni delle fasi in modo di avere più flessibilità in caso dei lavori non previsti.

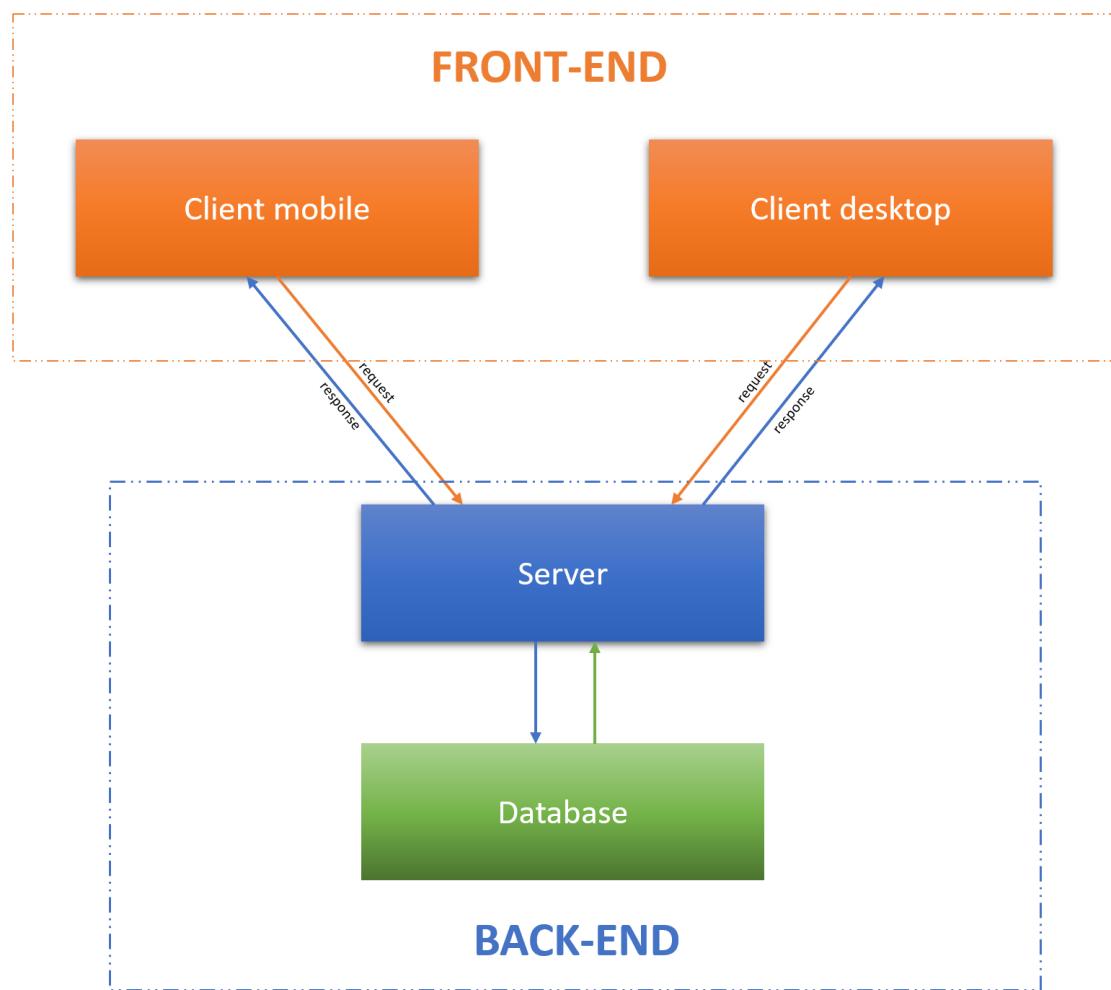
Il modello utilizzato per la programmazione delle fasi è quello a cascata, che nonostante ha un forte legame tra le diversi fasi di sviluppo, si adatta perfettamente all’intento del progetto che bisogna sviluppare e soprattutto il suo utilizzo è estremamente facile da maneggiare (possibile e facile la realizzazione con diversi strumenti) e comprendere (molto esplicativo nel mostrare l’ordine delle fasi e l’arco temporale utilizzato per ogni singola fase di sviluppo).



2 Documento di Design del sistema

2.1 Analisi dell'architettura

L'architettura dell'intero sistema è rappresentata da 3 livelli successivamente rappresentati:



I due clienti, in questo caso Client mobile (applicazione Android) e cliente desktop (applicazione windows) faranno le richieste al server, il quale una volta certificato l'utente e controllato la sua richiesta, sarà incaricato a effettuare delle richieste a database per poi, una volta filtrato le informazioni necessarie, inviarle al client che l'ha chiesto.

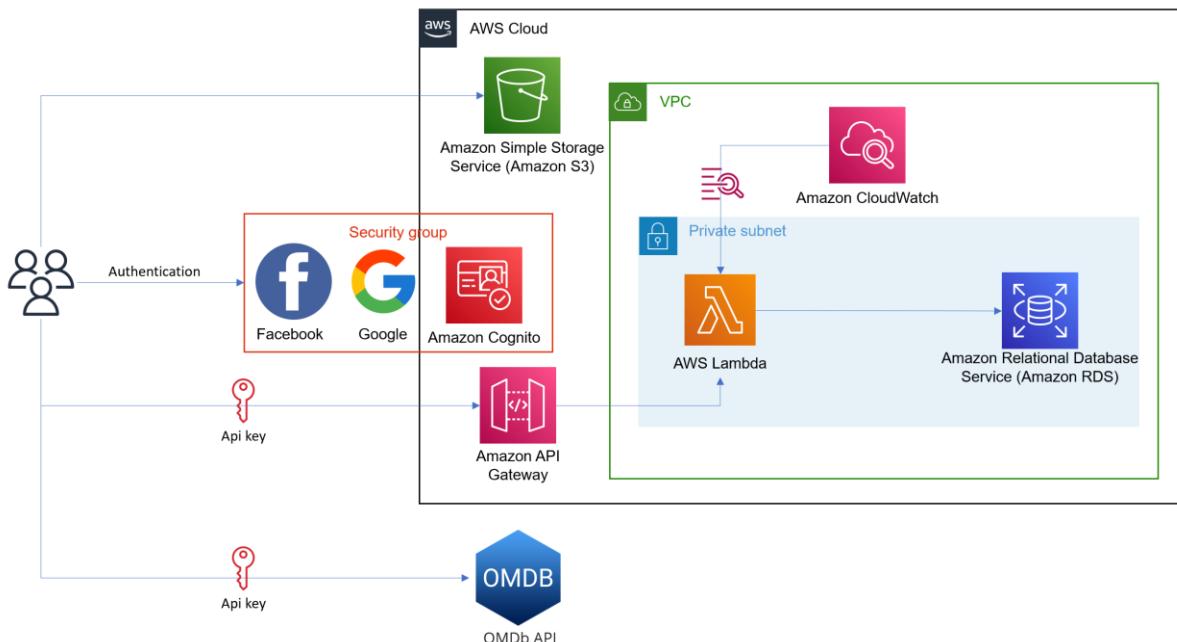


2.1.1 Architettura back-end

Per quanto riguarda il back-end è stato scelto di utilizzare una tecnologia serverless. I vantaggi ottenuti da tale scelta sono riassunti nei seguenti punti:

- Non è richiesta la gestione di una struttura adeguata all'hosting del server, non sarà più necessario allocare, ricalibrare e mantenere server per eseguire applicazioni, database e sistemi di storage.
- Ci si può concentrare esclusivamente sul prodotto senza badare al server.
- Si ha un servizio sempre attivo con possibilità future per espandere il prodotto in maniera semplice e intuitiva.

Per la realizzazione della suddetta architettura è stato scelto di utilizzare i servizi della piattaforma Amazon Web Services (AWS).



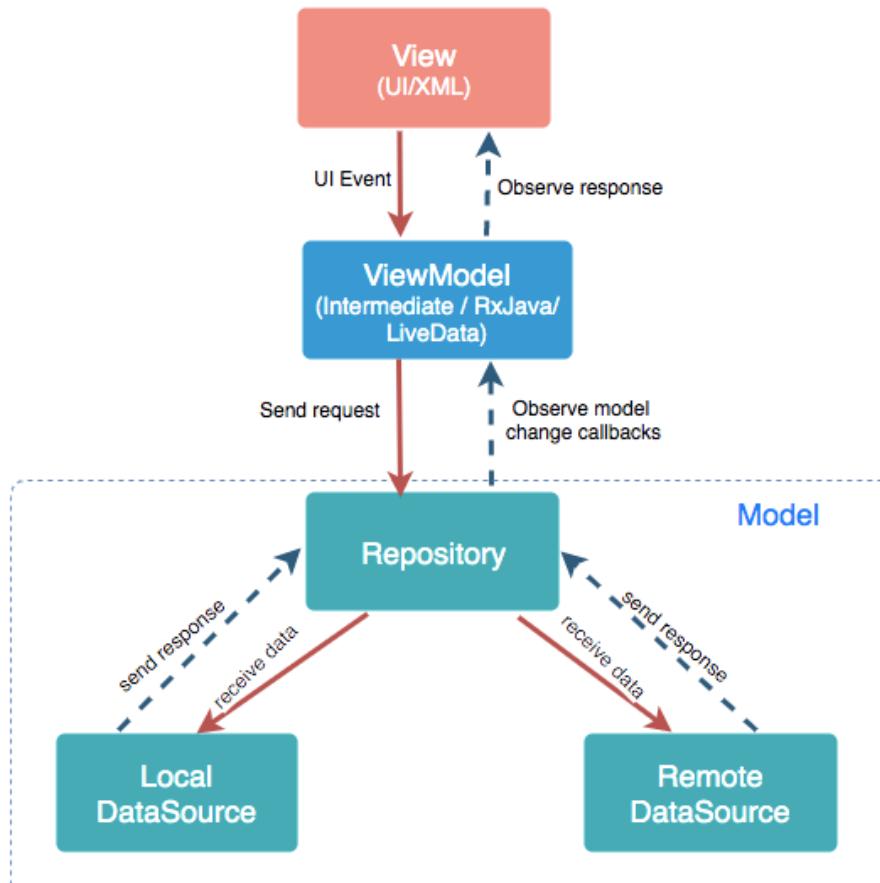
Verranno dunque elencati i servizi scelti per l'uso e una breve descrizione riportata dalla documentazione ufficiale.

- **Amazon Simple Storage Service (Amazon S3):** è un servizio di storage di oggetti che offre scalabilità, disponibilità dei dati, sicurezza e prestazioni all'avanguardia nel settore. Ciò significa che le industrie di ogni dimensione possono utilizzarlo per archiviare e proteggere una qualsiasi quantità di dati per una vasta gamma di casi d'uso. Amazon S3 offre caratteristiche di gestione semplici da utilizzare che ti consentono di organizzare i dati e di configurare controlli di accesso ottimizzati per soddisfare requisiti aziendali, di pianificazione e di conformità specifici.
- **Amazon Cognito:** permette di aggiungere strumenti di registrazione degli utenti, accesso e controllo degli accessi alle app Web e per dispositivi mobili, in modo rapido e semplice.
- **AWS Lambda:** è un servizio di elaborazione serverless che ti permette di eseguire il codice senza effettuare il provisioning o gestire i server, creare una logica di dimensionamento dei cluster in funzione dei carichi di lavoro, mantenere integrazioni degli eventi o gestire i runtime.
- **Amazon Relational Database Service (Amazon RDS):** semplifica l'impostazione, il funzionamento e il dimensionamento di database relazionali nel cloud. Questo servizio fornisce una capacità ridimensionabile a un costo conveniente, automatizzando al tempo stesso le attività di amministrazione del database più dispendiose in termini di tempo, quali il provisioning di hardware, l'impostazione di database, gli aggiornamenti e i backup. Consente di concentrarsi sulle proprie applicazioni per fornire le prestazioni ottimali, la disponibilità elevata, la sicurezza e la compatibilità di cui hanno bisogno.
- **Amazon CloudWatch:** è un servizio di monitoraggio e osservabilità creato per ingegneri, sviluppatori, ingegneri responsabili dell'affidabilità del sito (SRE) e manager IT DevOps. CloudWatch fornisce dati e analisi concrete per monitorare le applicazioni, rispondere ai cambiamenti di prestazioni a livello di sistema, ottimizzare l'utilizzo delle risorse e ottenere una visualizzazione unificata dello stato di integrità operativa.
- **Amazon API Gateway:** è un servizio completamente gestito che semplifica per gli sviluppatori la creazione, la pubblicazione, la manutenzione, il monitoraggio e la protezione delle API su qualsiasi scala. Le API fungono da "porta di entrata" per consentire l'accesso delle applicazioni ai dati, alla logica aziendale o alle funzionalità dai servizi back-end. API Gateway consente di creare API RESTful e WebSocket che rendono possibili applicazioni di comunicazione bidirezionale in tempo reale. API Gateway supporta carichi di lavoro containerizzati e senza server, oltre che applicazioni Web.



2.1.2 Architettura front-end

Per l'architettura client è stato scelto di adottare il pattern MVVM (Model-View-ViewModel).



L'architettura, come suggerisce la figura e anche il nome, consiste nel suddividere l'applicativo in 3 strati nettamente separati tra loro:

- **View:** si occupa della presentazione delle schermate che l'utente andrà a visualizzare. Inoltre, dovrà segnalare al **ViewModel** eventuali eventi che si presentano quando un utente interagisce con l'interfaccia.
- **ViewModel:** si occupa della logica per l'interfaccia utente, contiene le informazioni (**LiveData**) che la **view** deve rappresentare, ha la funzione di intermediario tra **view** e **model**.
- **Model:** si occupa della logica dell'intero sistema. Comunicando con i servizi esterni, li interroga, legge le loro risposte, le elabora e le restituisce in maniera leggibile al **ViewModel**.



Struttura interna

Per ogni Fragment/Activity sarà associata una classe ViewModel che gestirà la logica della View stessa e fornirà i dati da visualizzare. Ogni ViewModel avrà un riferimento al repository che sarà un'interfaccia sicura con i vari servizi utilizzati. Da sottolineare come il ViewModel non è conoscenza della View alla quale fornisce i dati, e come il Model non è a conoscenza dei ViewModel ai quali fornisce le risposte ottenute dalle richieste verso l'esterno.

Comunicazione

Il metodo con cui il ViewModel segnalerà alla View un cambiamento nei dati, sarà effettuato tramite Observer e MutableLiveData. Quando una View verrà creata, tramite opportuni Observer si metterà in ascolto sui MutableLiveData presenti nel ViewModel, in questo modo quando questi ultimi cambieranno la view potrà aggiornarsi di conseguenza. Dal lato del ViewModel invece, una volta effettuata una richiesta e dopo aver ottenuto le risposte desiderate, apporteranno delle modifiche sui valori conservati nei MutableLiveData locali in modo da segnalare alla View che i dati sono cambiati.

Vantaggi

I vantaggi provenienti da quest'architettura sono innumerevoli tra cui la possibilità di riusare il codice, essendo le schermate separate dalla logica sarà possibile riutilizzarle senza troppi problemi, allo stesso modo sarà possibile modifica la schermata senza preoccuparsi di riscrivere il codice per la logica essendo esso sempre presente all'interno del ViewModel. Dalla suddetta divisione tra sezione logica e quella di visualizzazione è facile intuire la facilità con cui si possono trovare eventuali errori nel codice; quindi, c'è un'ottima gestione dell'intero sistema. Sono inoltre facilitati i test, dato che durante i test sarà possibile interrogare i metodi di logica solo all'interno nel ViewModel senza interpellare la View.



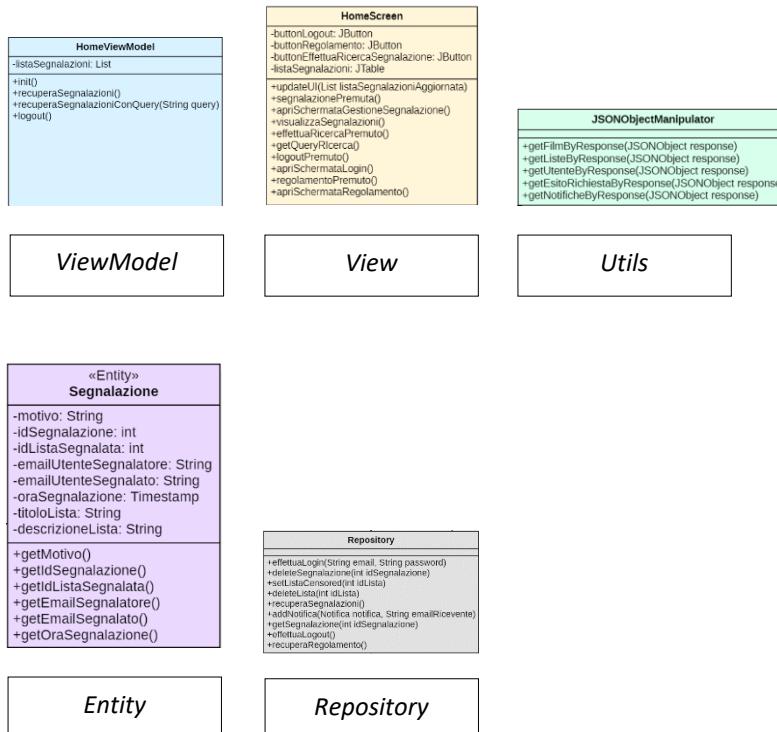
2.2 Diagramma delle classi di design

In questo paragrafo verranno mostrati i diagrammi delle classi di design, dove vengono mostrate, in base alle scelte architetturali descritte nel paragrafo precedente, come le classi sono connesse tra loro. Si può notare come nei diagrammi è stato adottato come collegamento tra le classi la “Directed Association” in quanto la relazione tra le classi è solo navigabile in un verso.

Ad esempio, ogni viewModel nell'applicativo ha un collegamento verso il repository che gli fornisce i servizi, ma non ha alcun riferimento alla View, è dunque più naturale in questo modo leggere il diagramma.

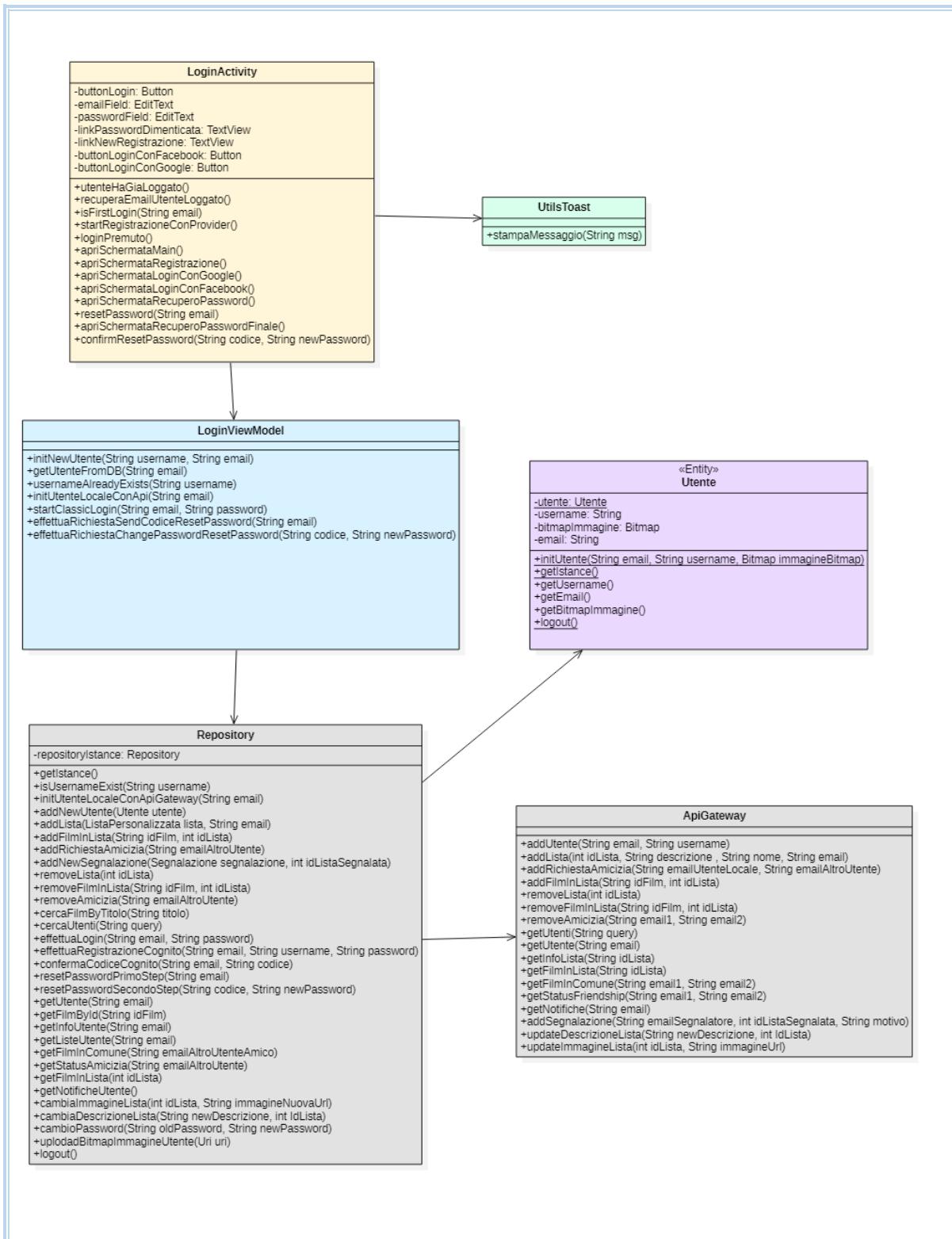
Sono stati adottati alcuni colori per identificare visivamente meglio la categoria della classe a cui appartiene:

- **VIEW:** color giallo
- **VIEWMODEL:** colore azzurro
- **REPOSITORY-API:** colore grigio
- **ENTITY:** colore viola
- **UTILS:** colore verde

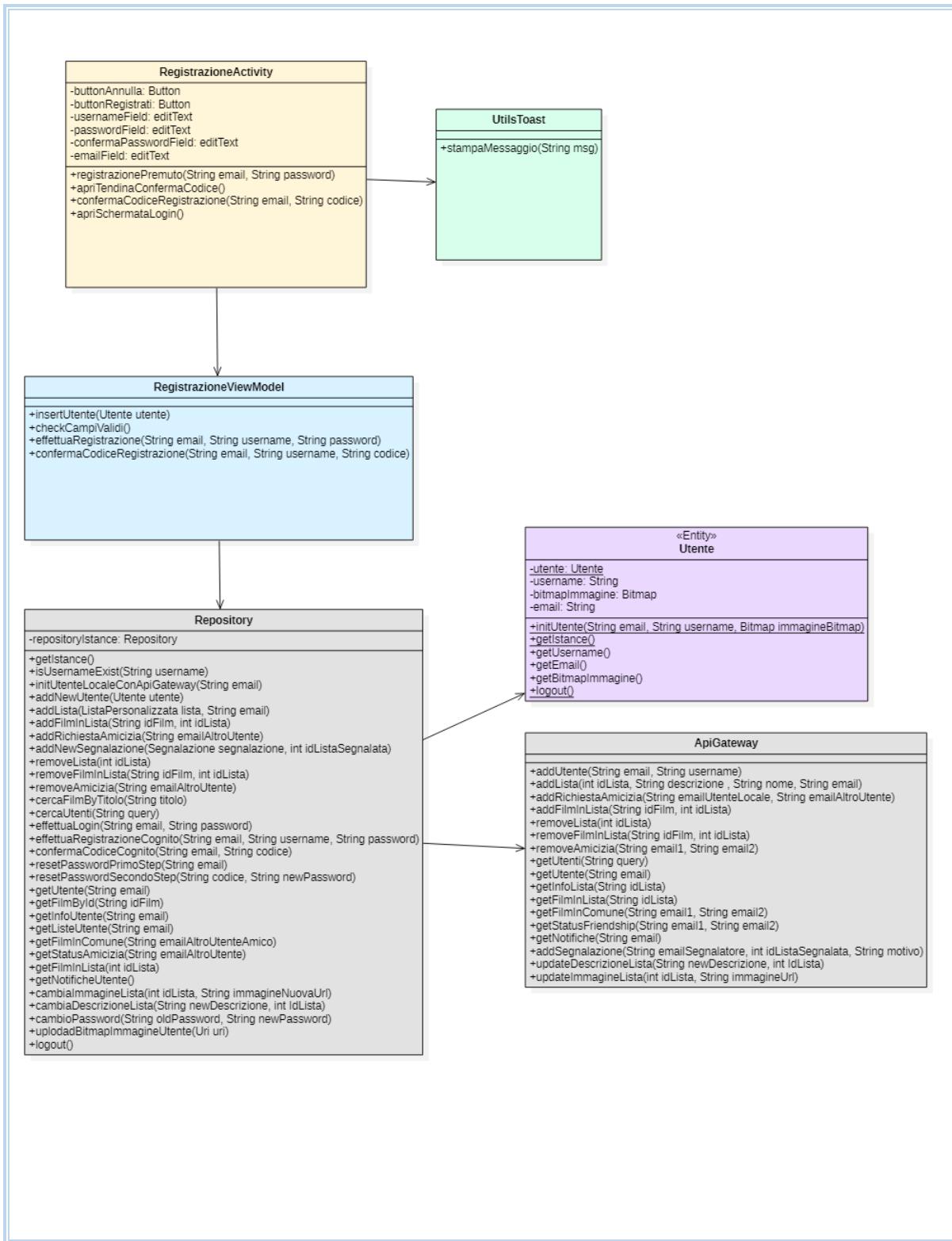


2.2.1 Android

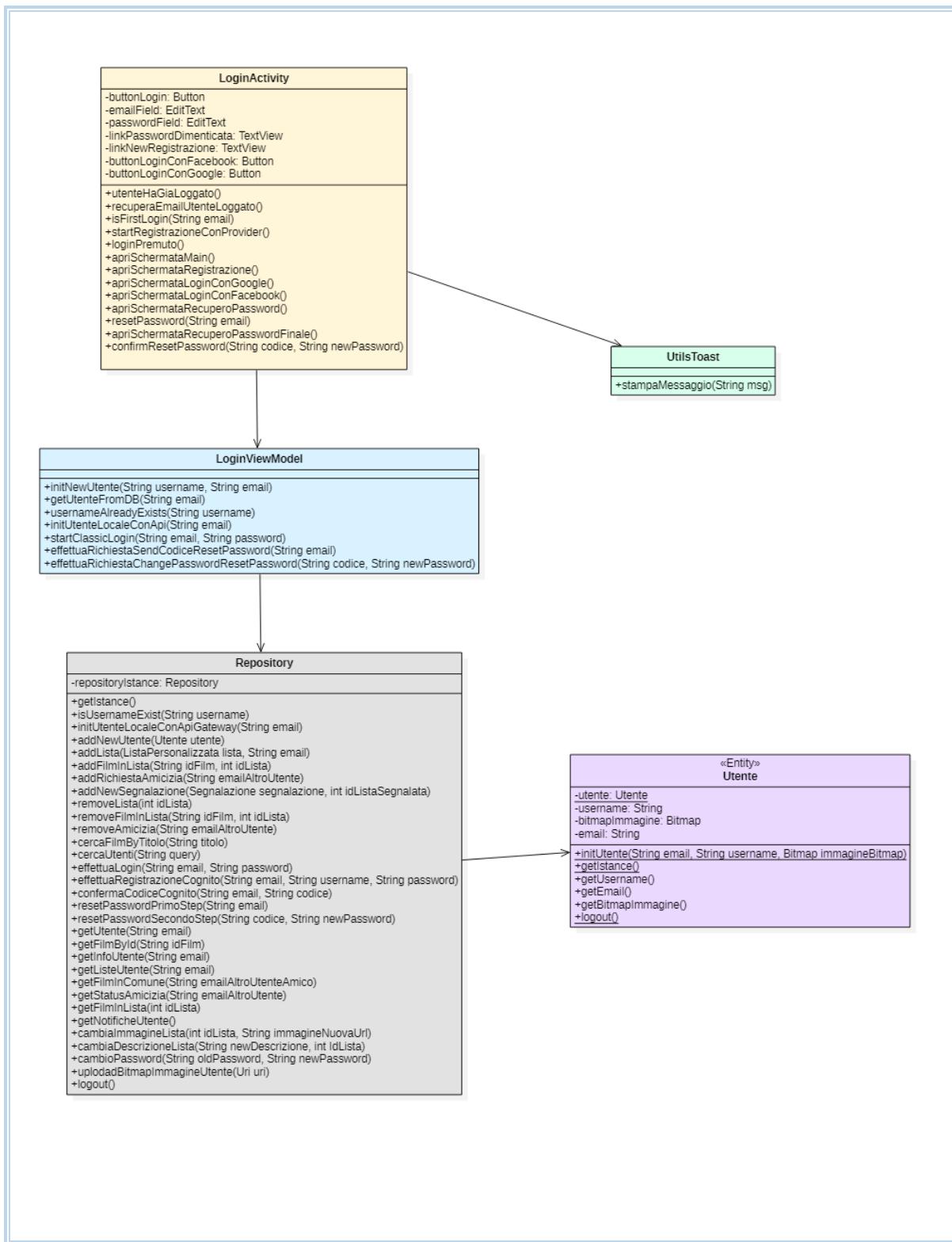
2.2.1.1 Accesso al sistema



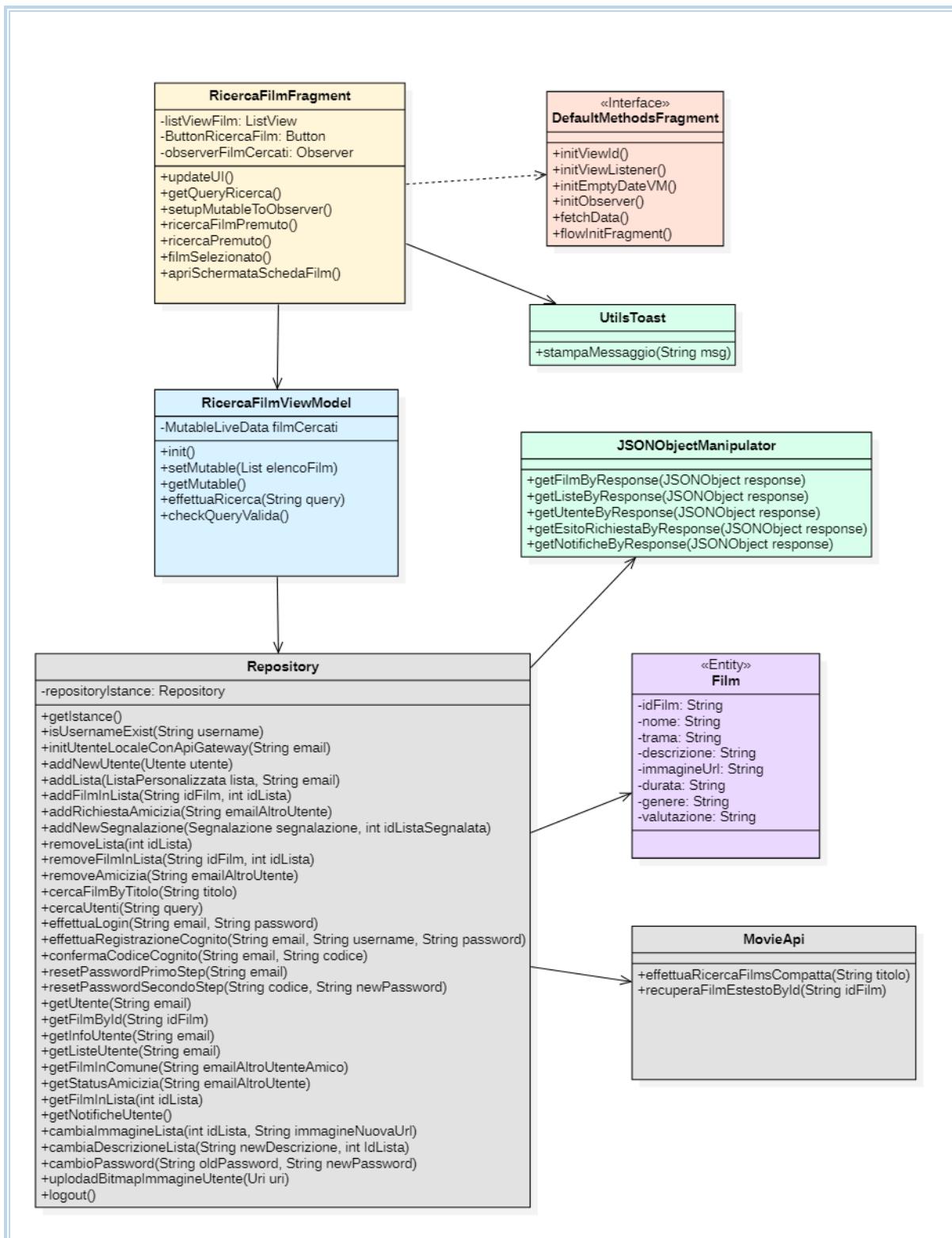
2.2.1.2 Registrazione al sistema



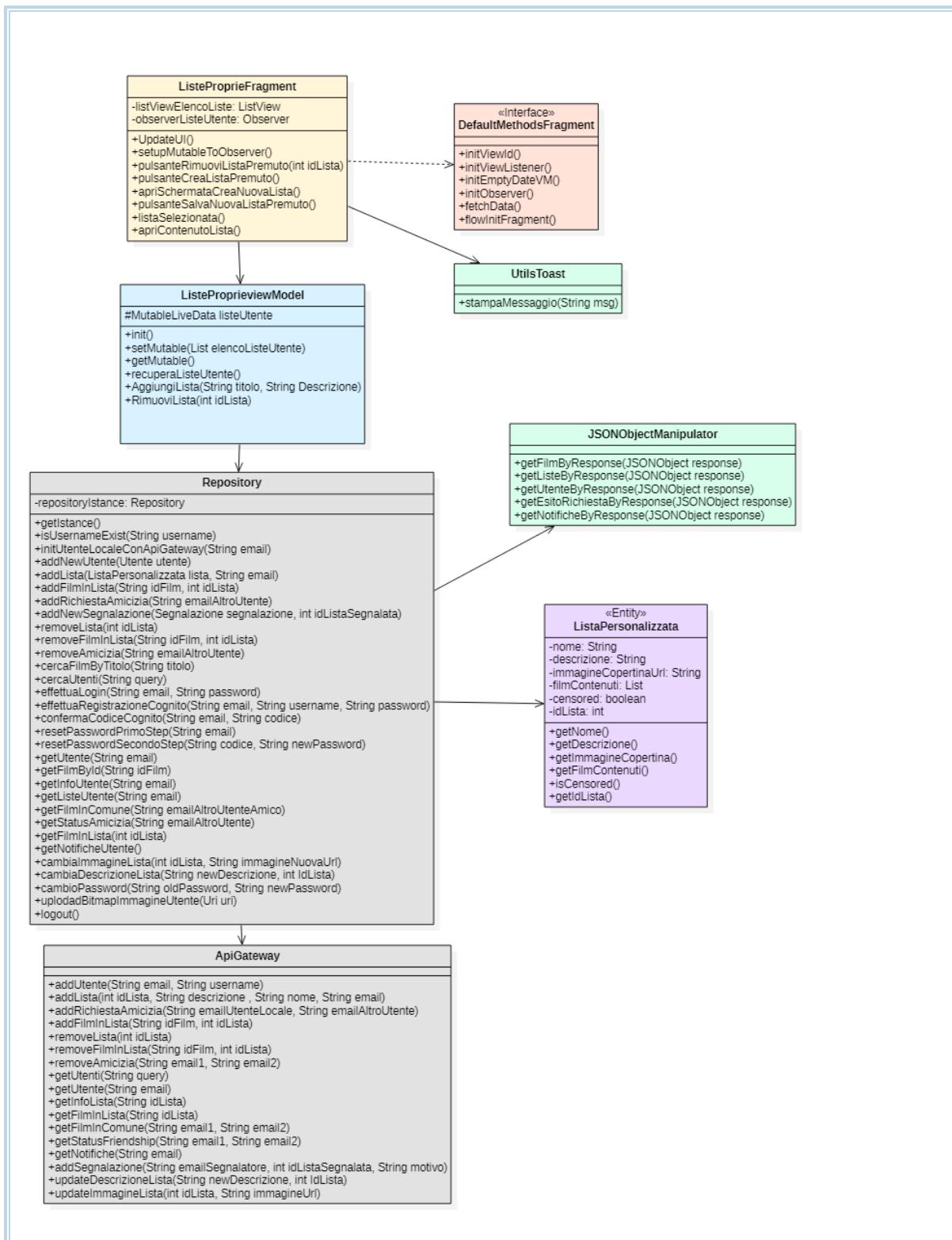
2.2.1.3 Recupero password dimenticata



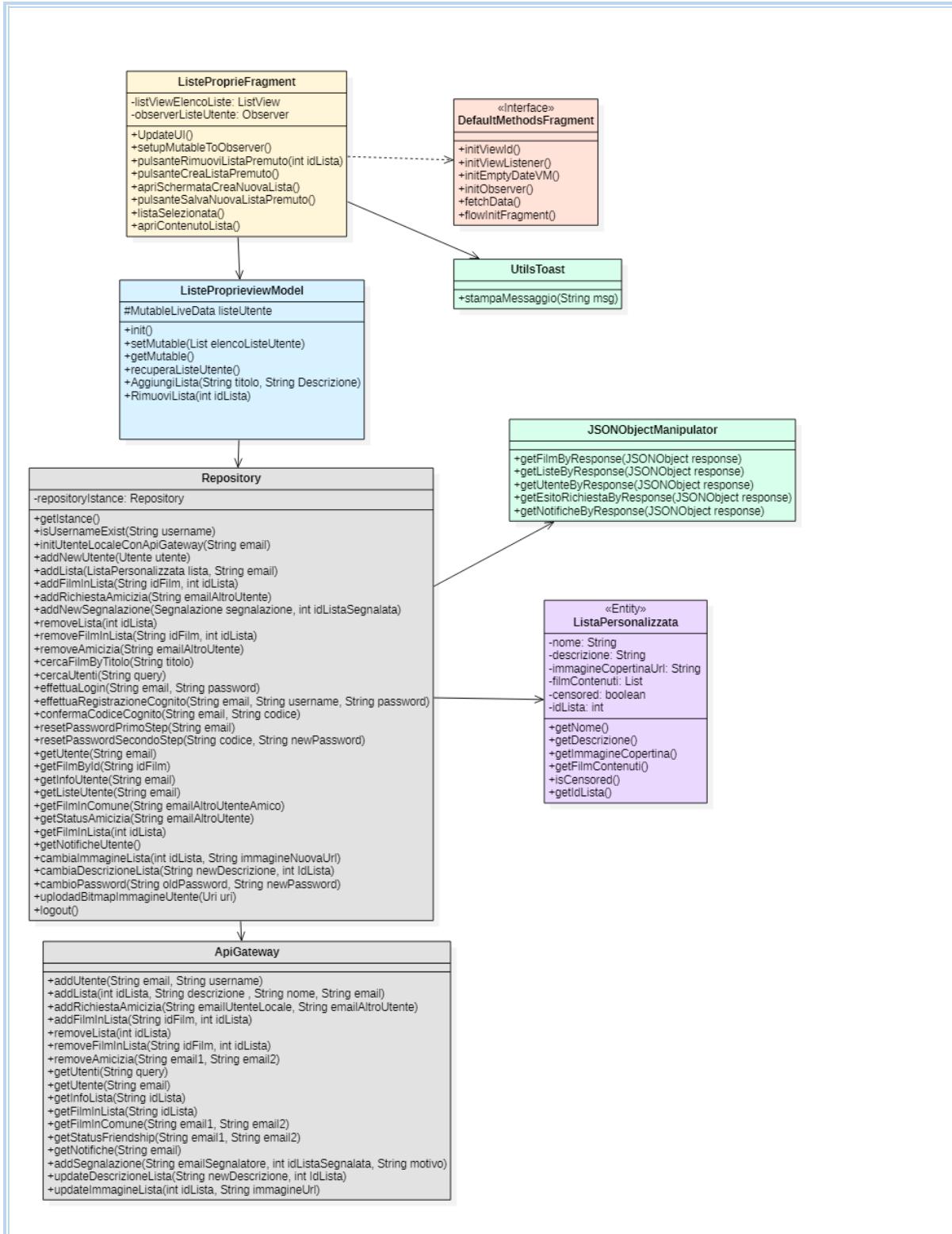
2.2.1.4 Effettuare ricerche film



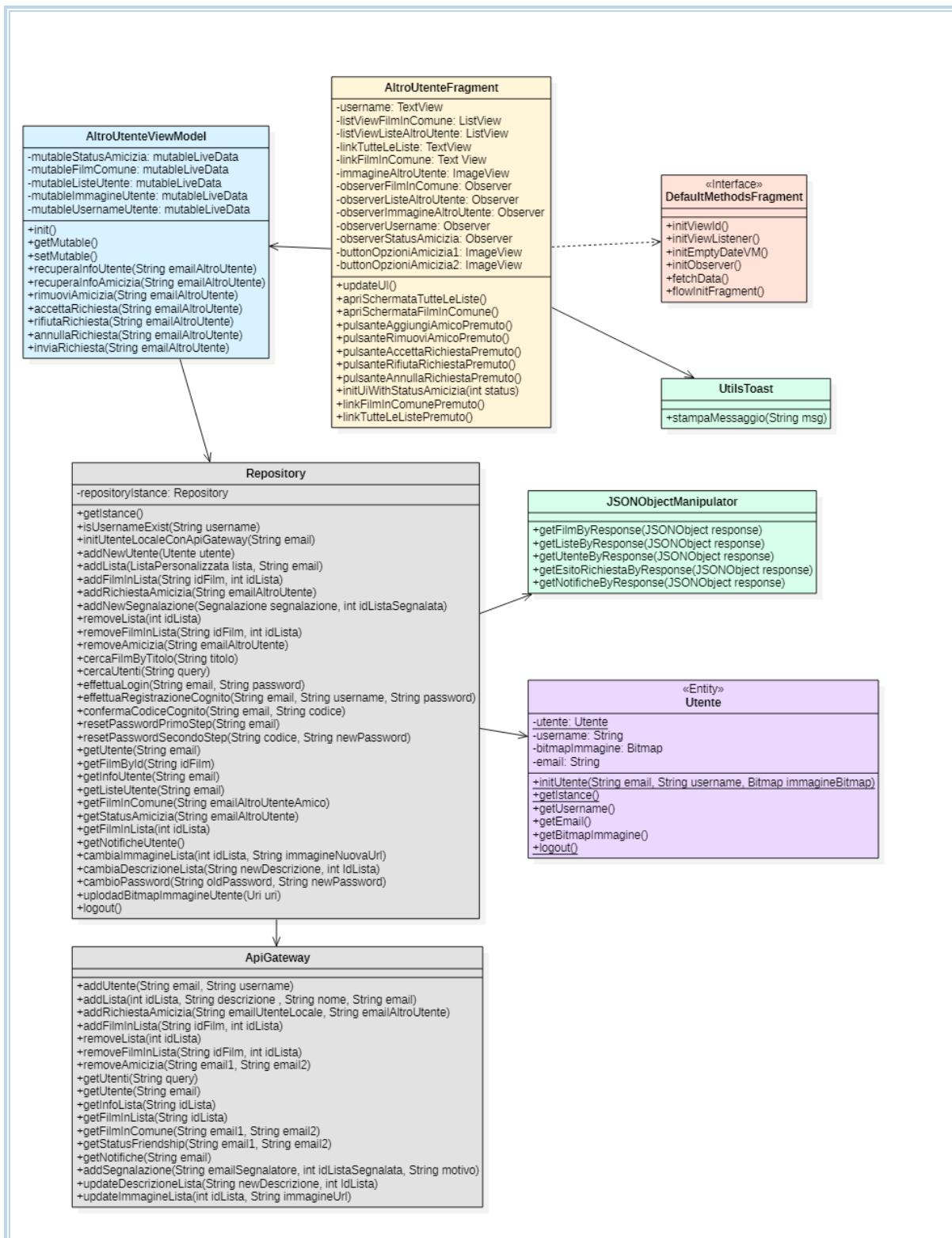
2.2.1.5 Creazione di una lista di film personalizzata



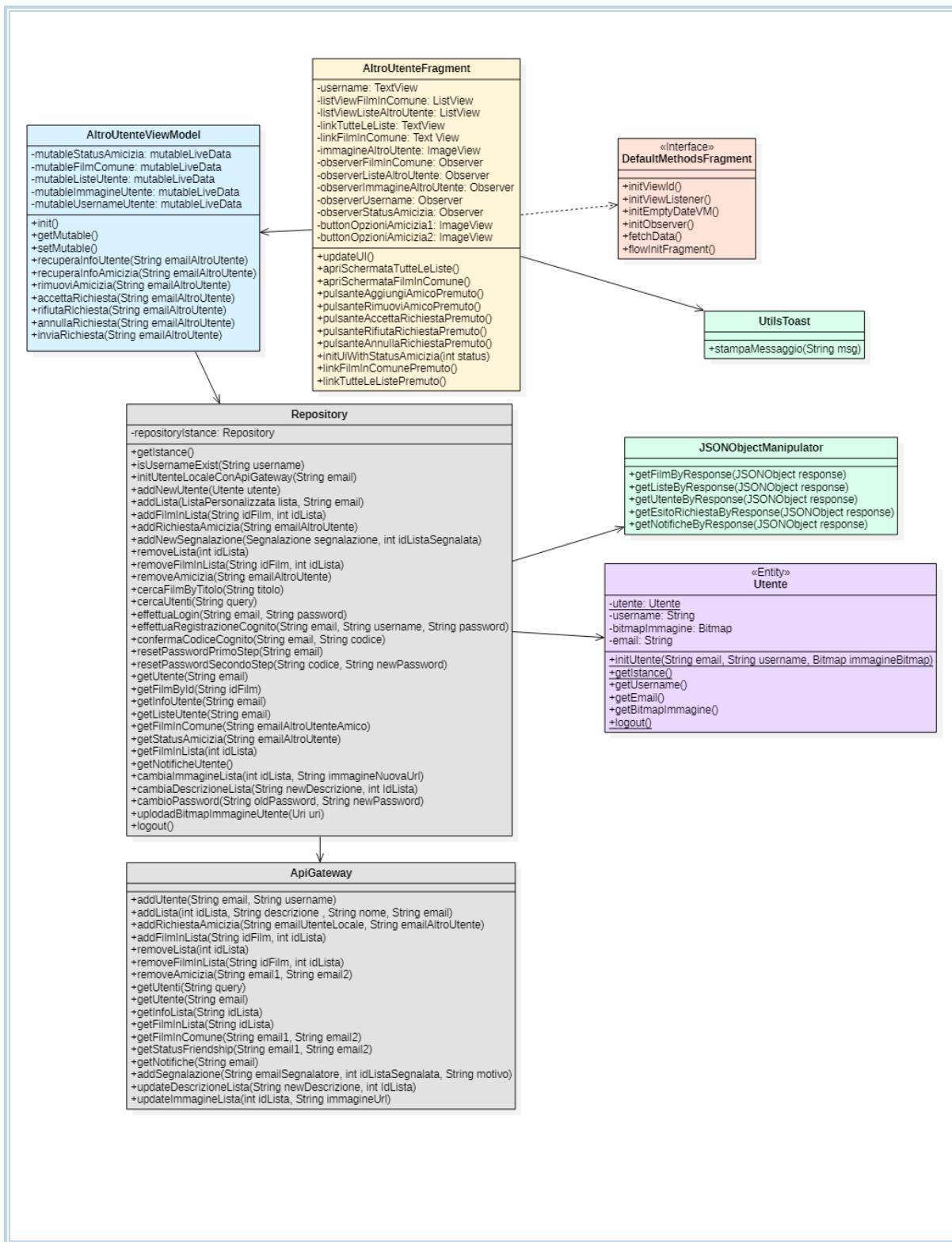
2.2.1.6 Rimozione di una lista di film personalizzata



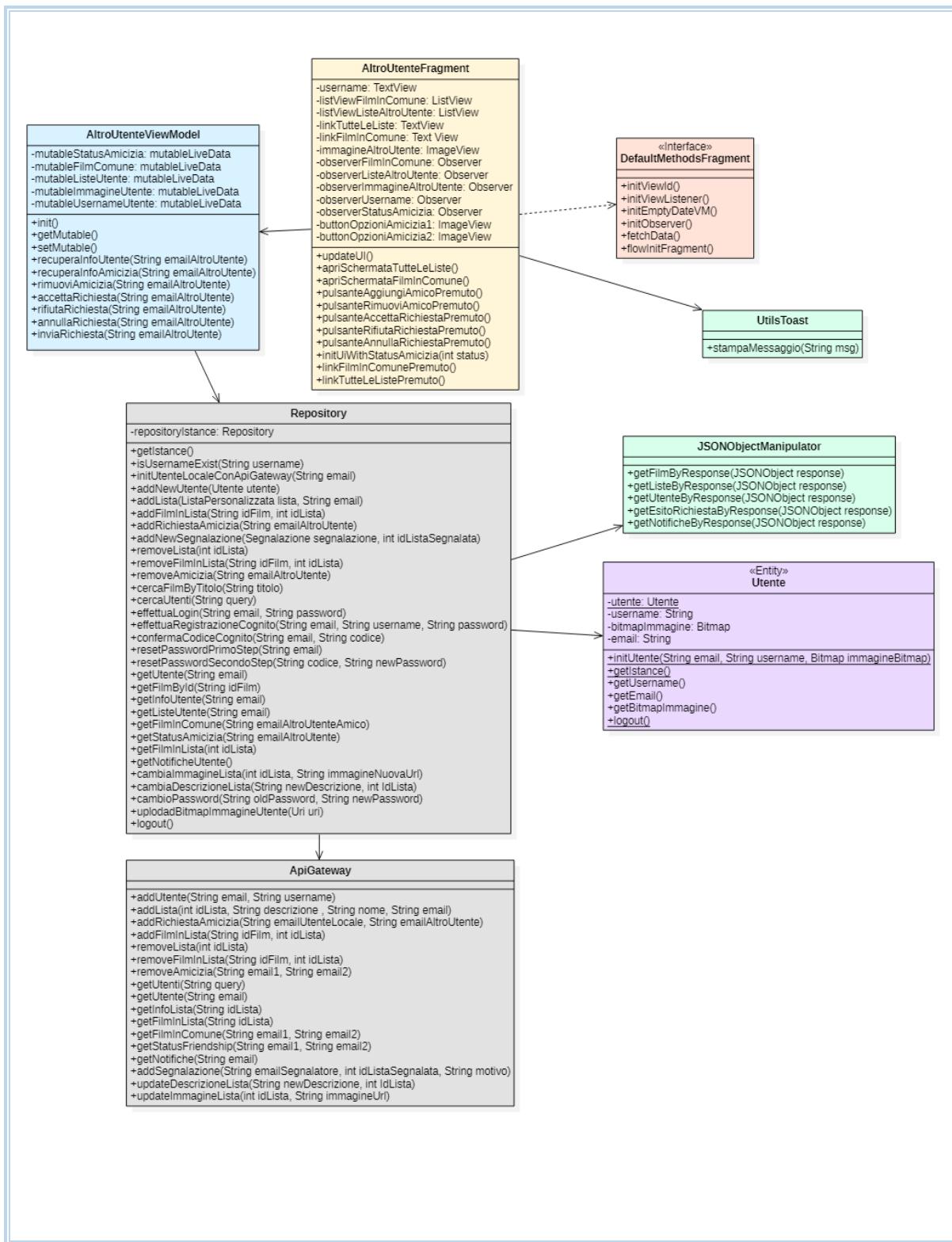
2.2.1.7 Inviare/Ricevere richieste di amicizia



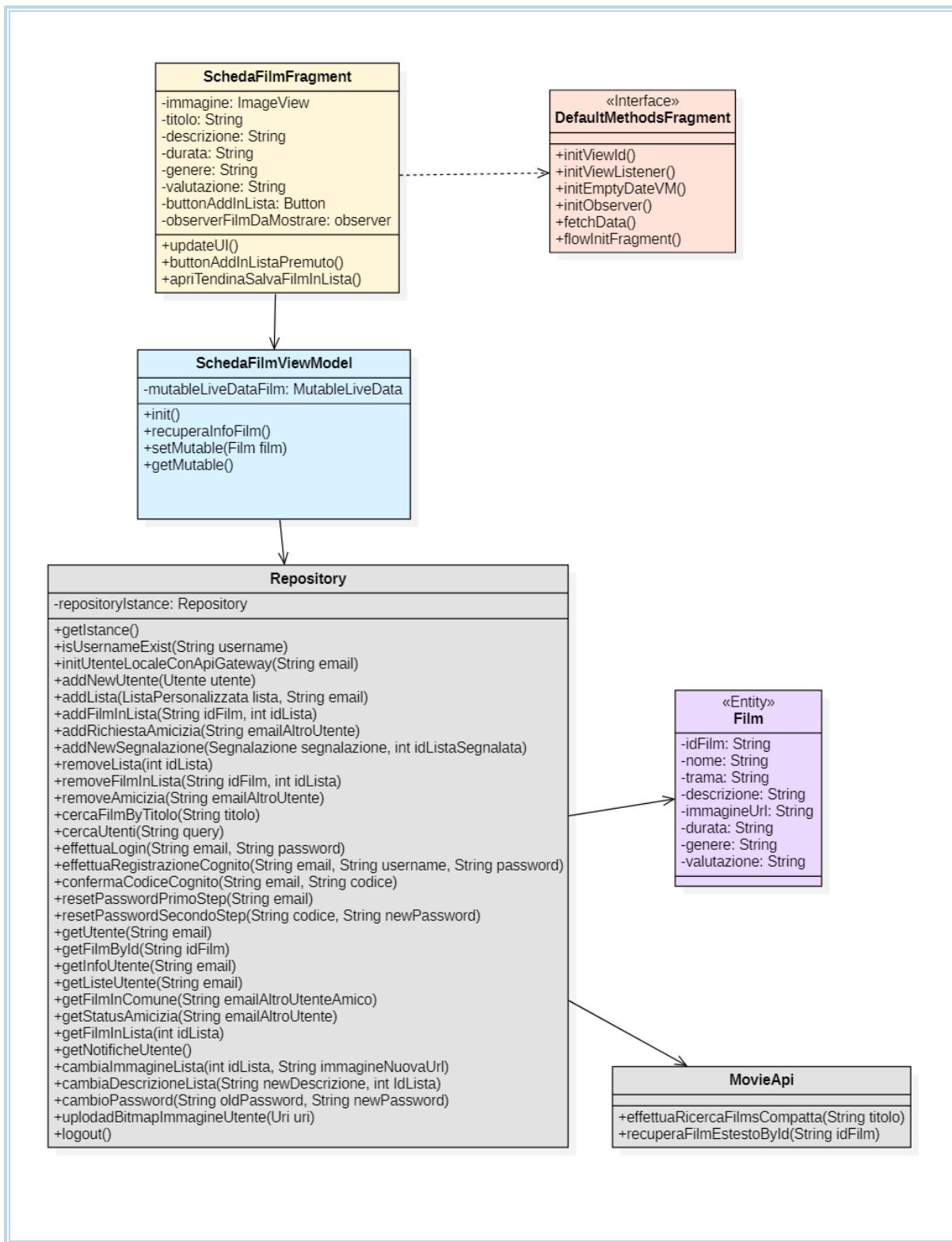
2.2.1.8 Accettare-Rifiutare richieste di amicizia



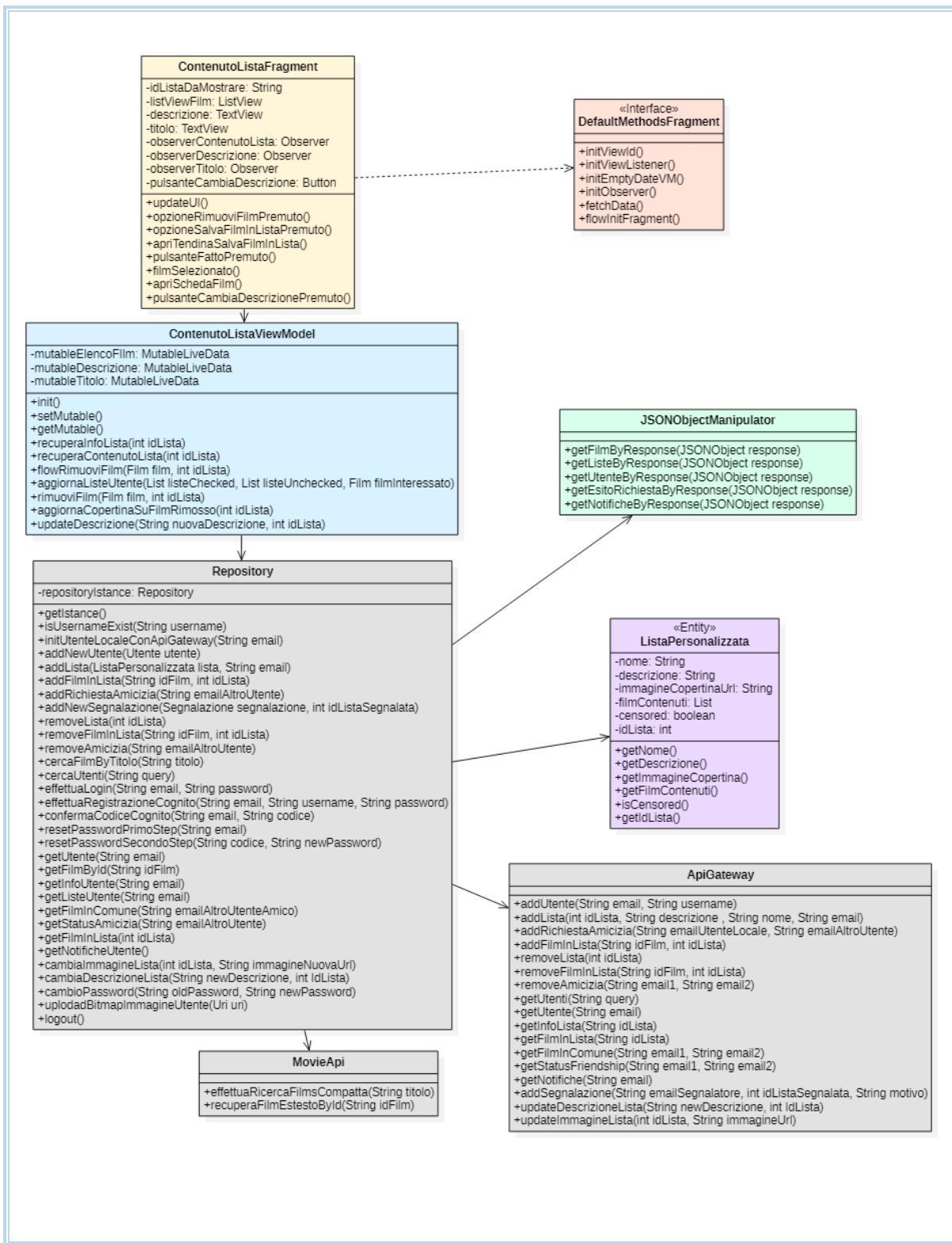
2.2.1.9 Visualizzazione scheda altro utente



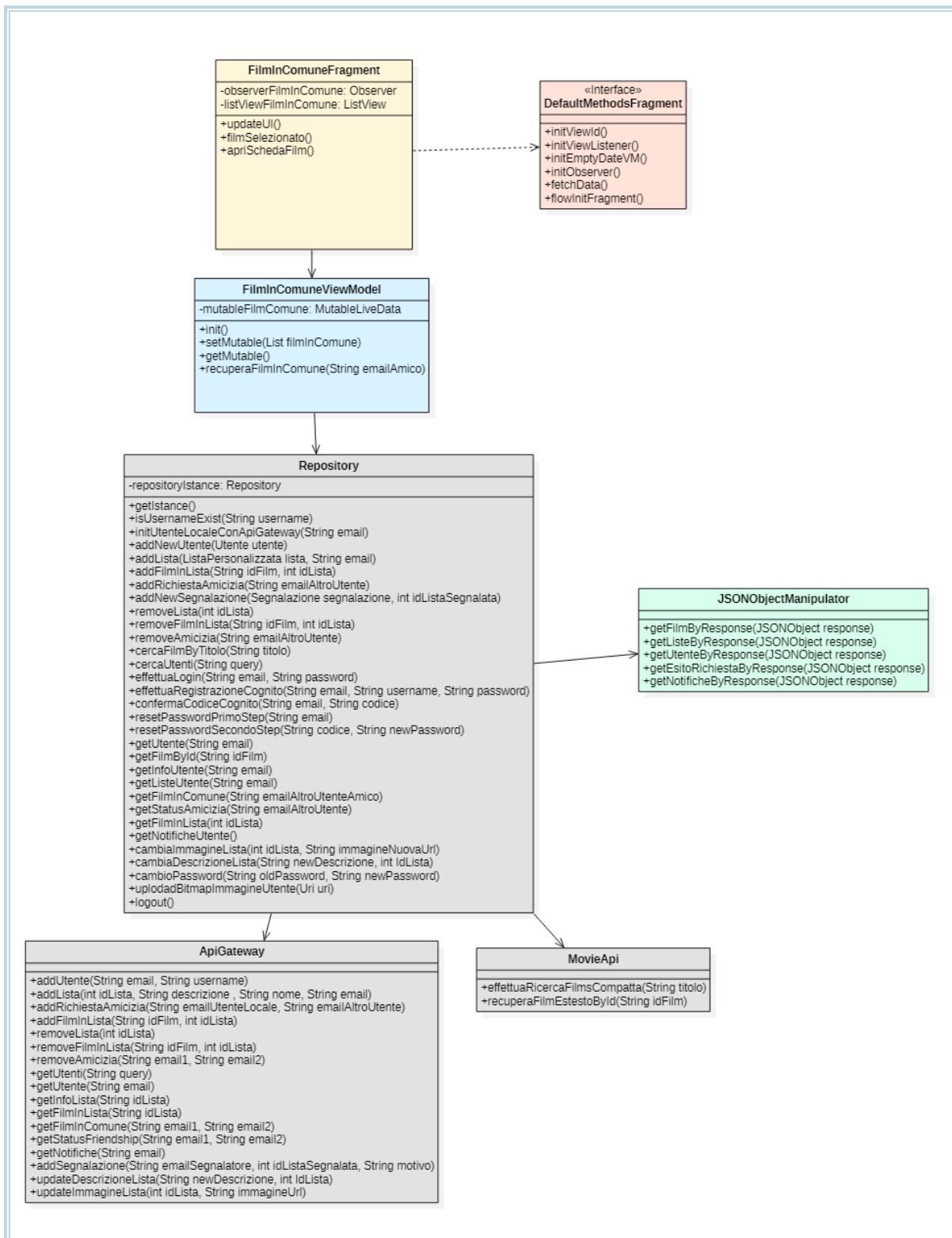
2.2.1.10 Visualizzazione scheda film



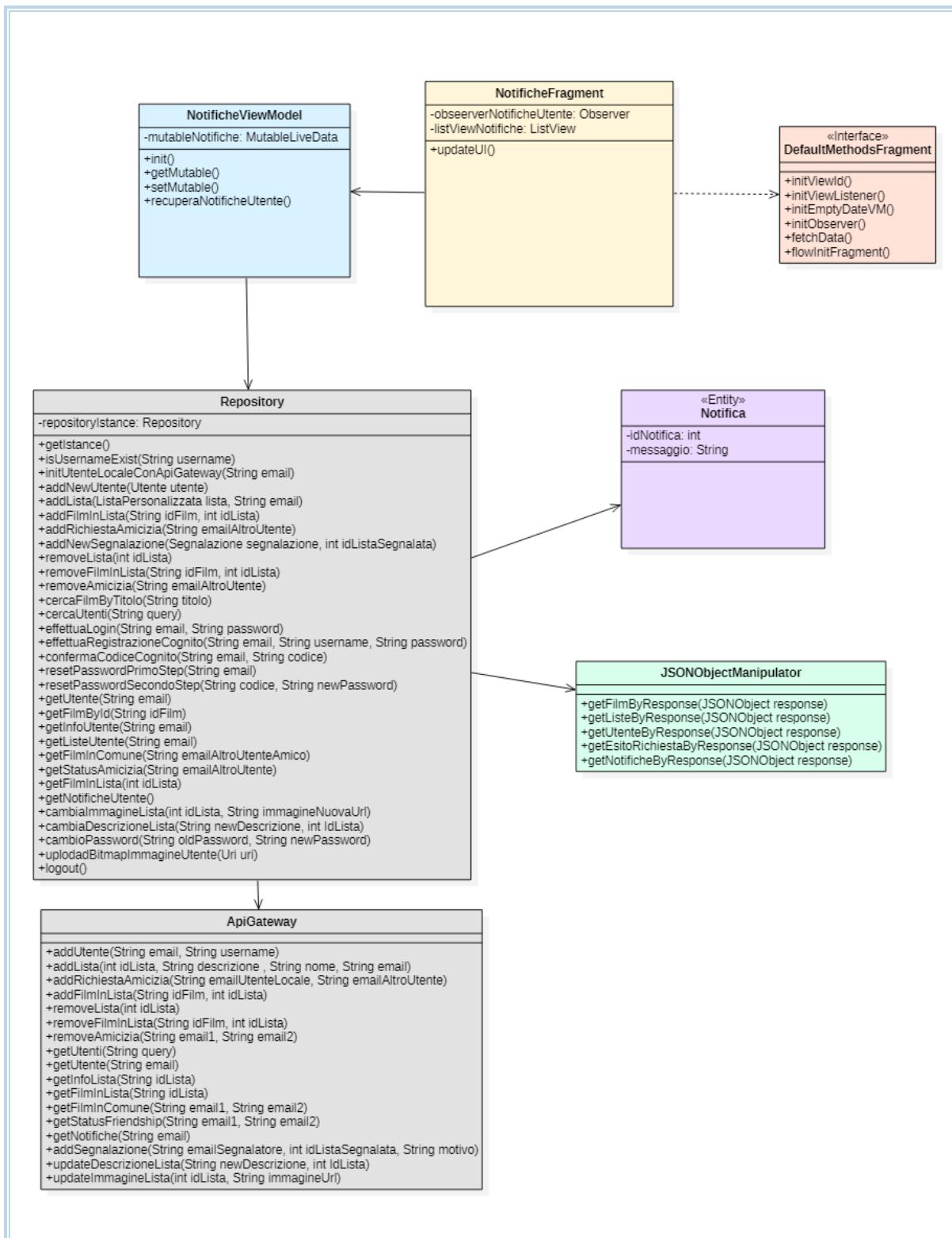
2.2.1.11 Visualizzazione lista propria



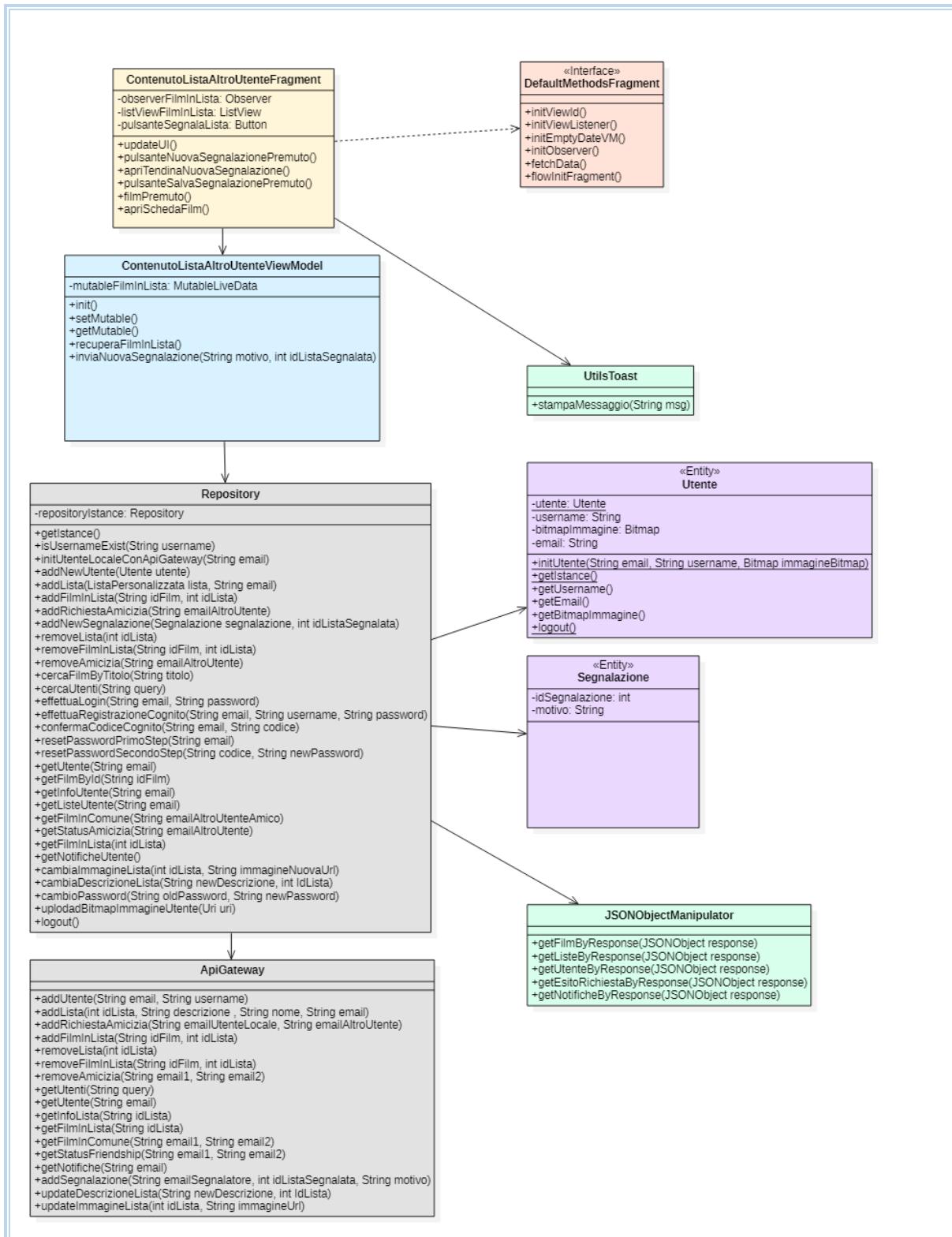
2.2.1.12 Visualizzazione film in comune



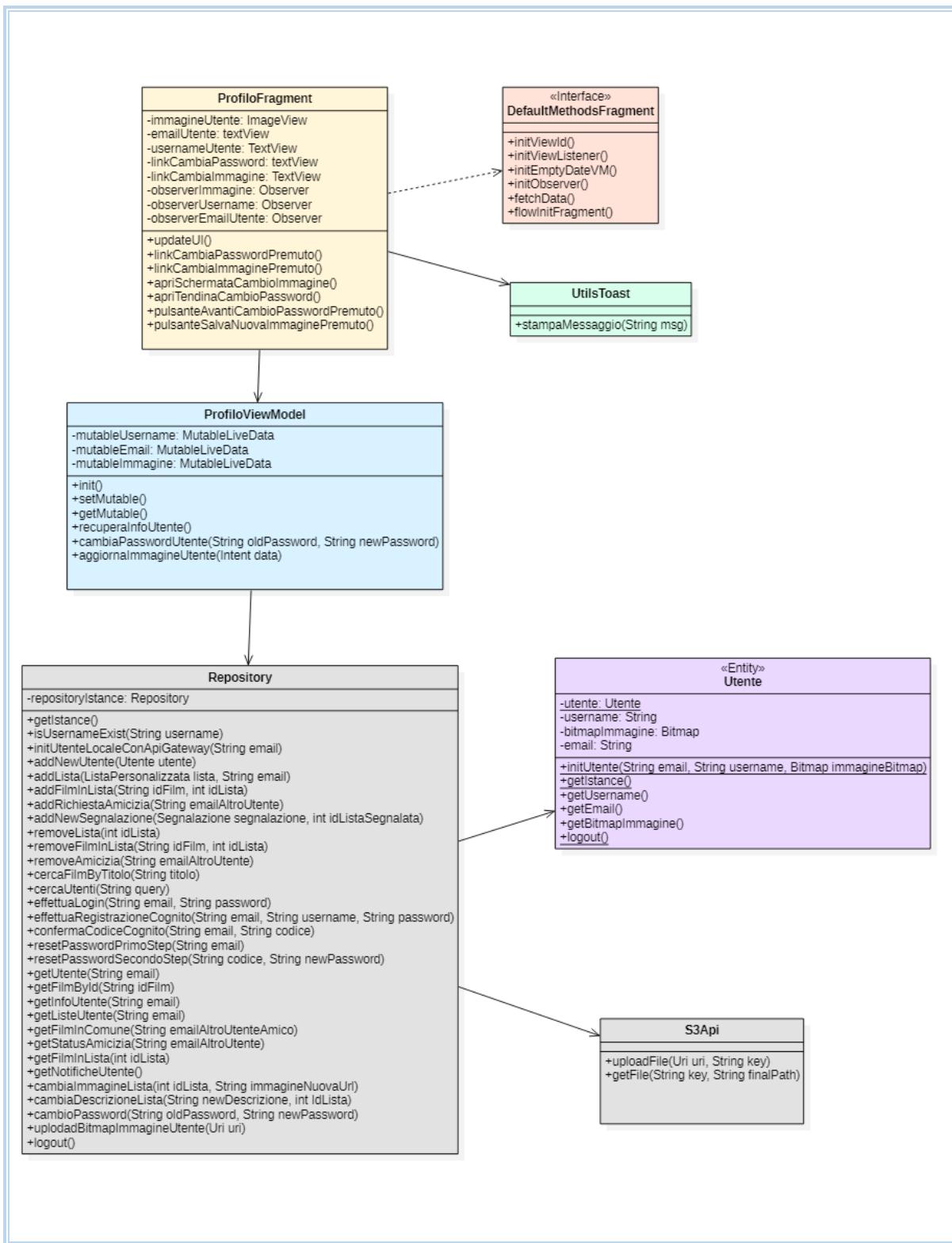
2.2.1.13 Visualizzazione notifiche



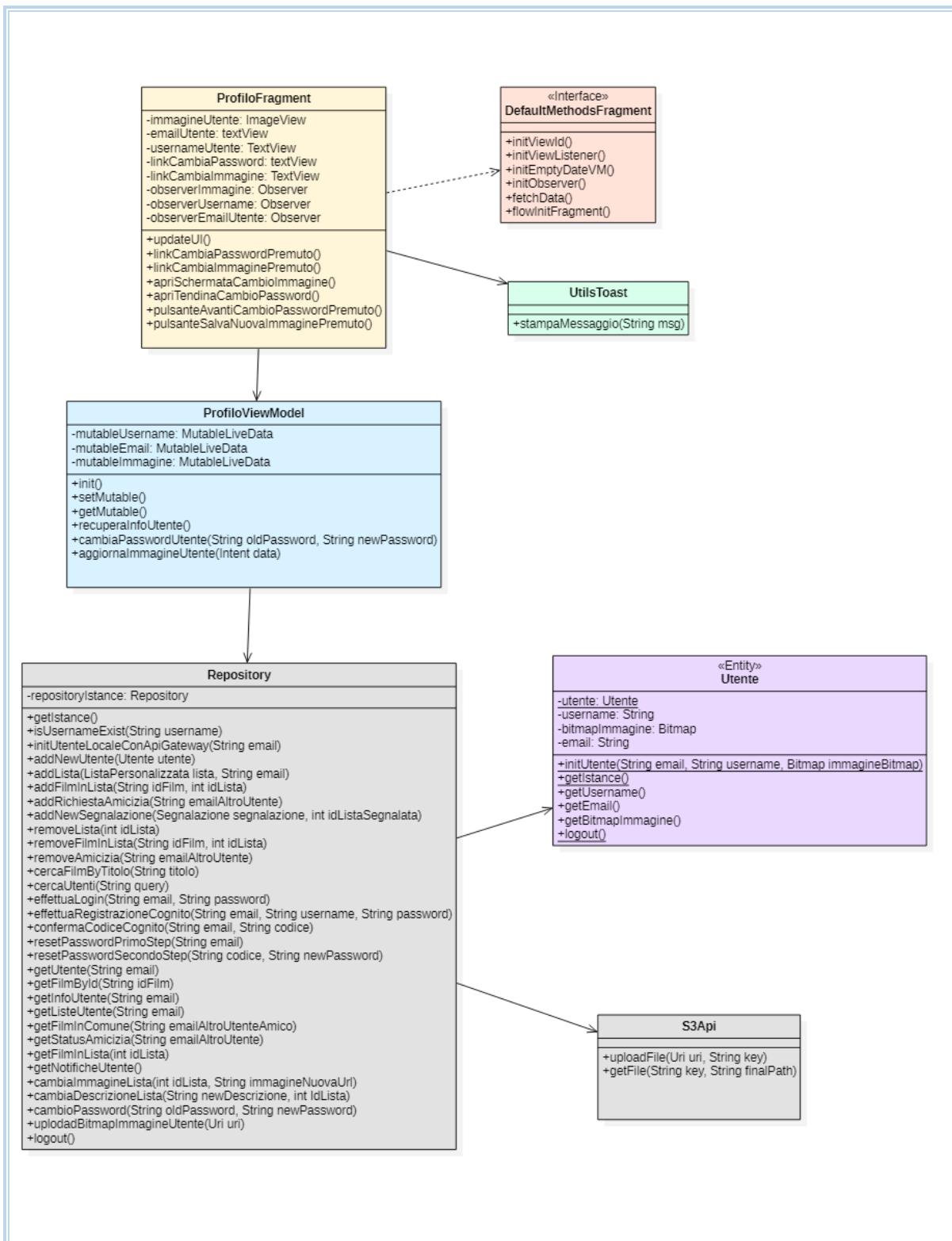
2.2.1.14 Effettuare segnalazione



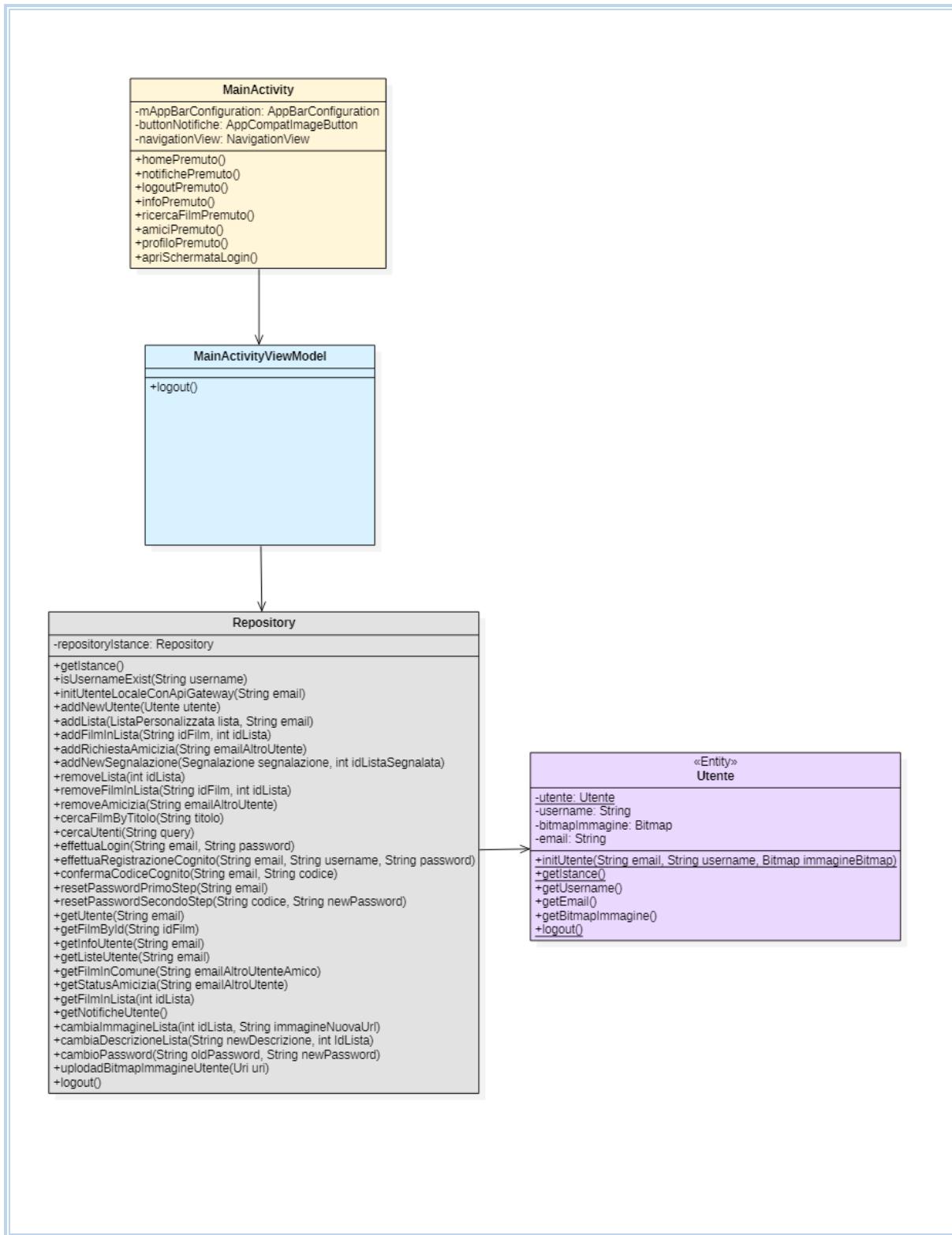
2.2.1.15 Cambio password/Cambio immagine profilo



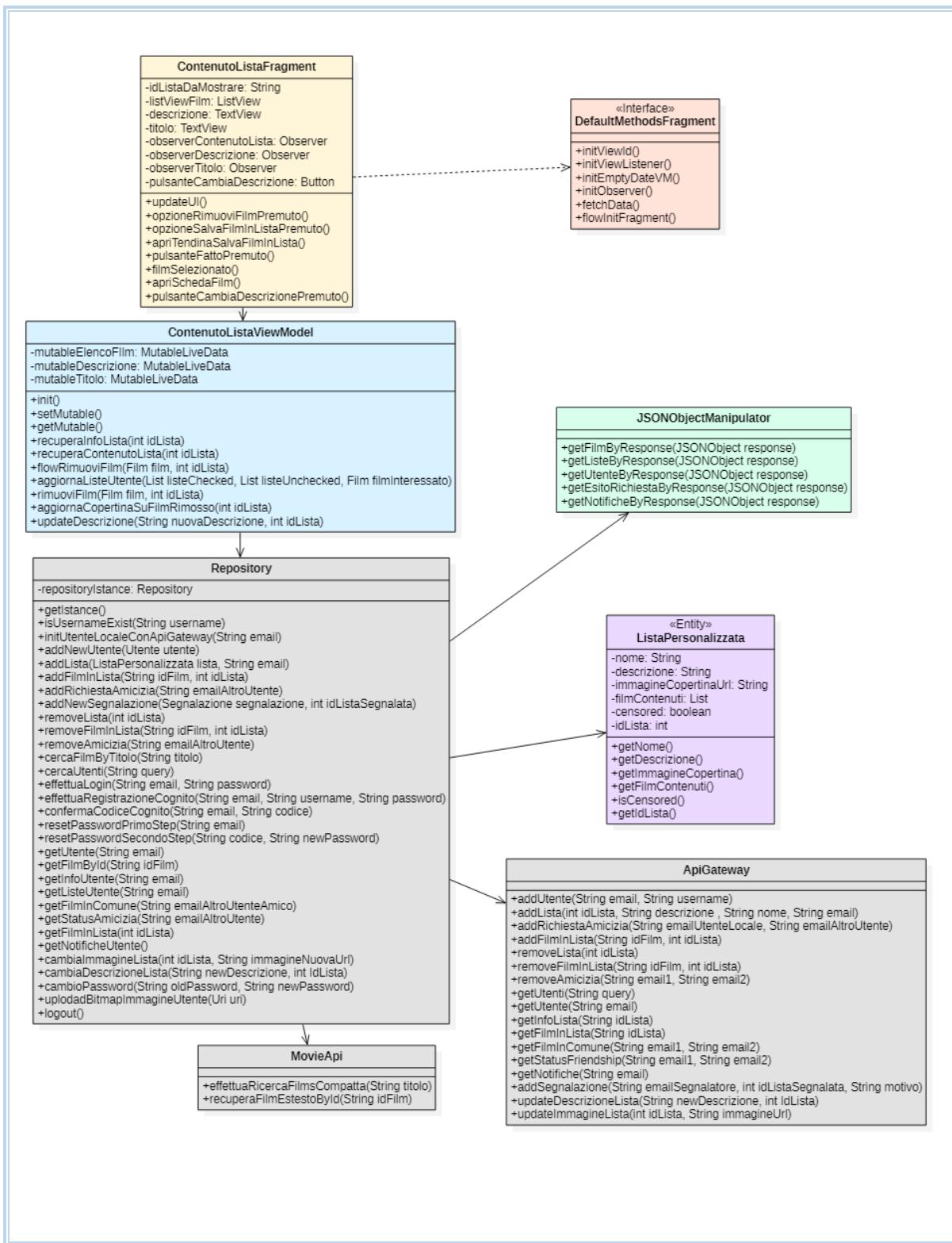
2.2.1.16 Visualizzazione proprio profilo



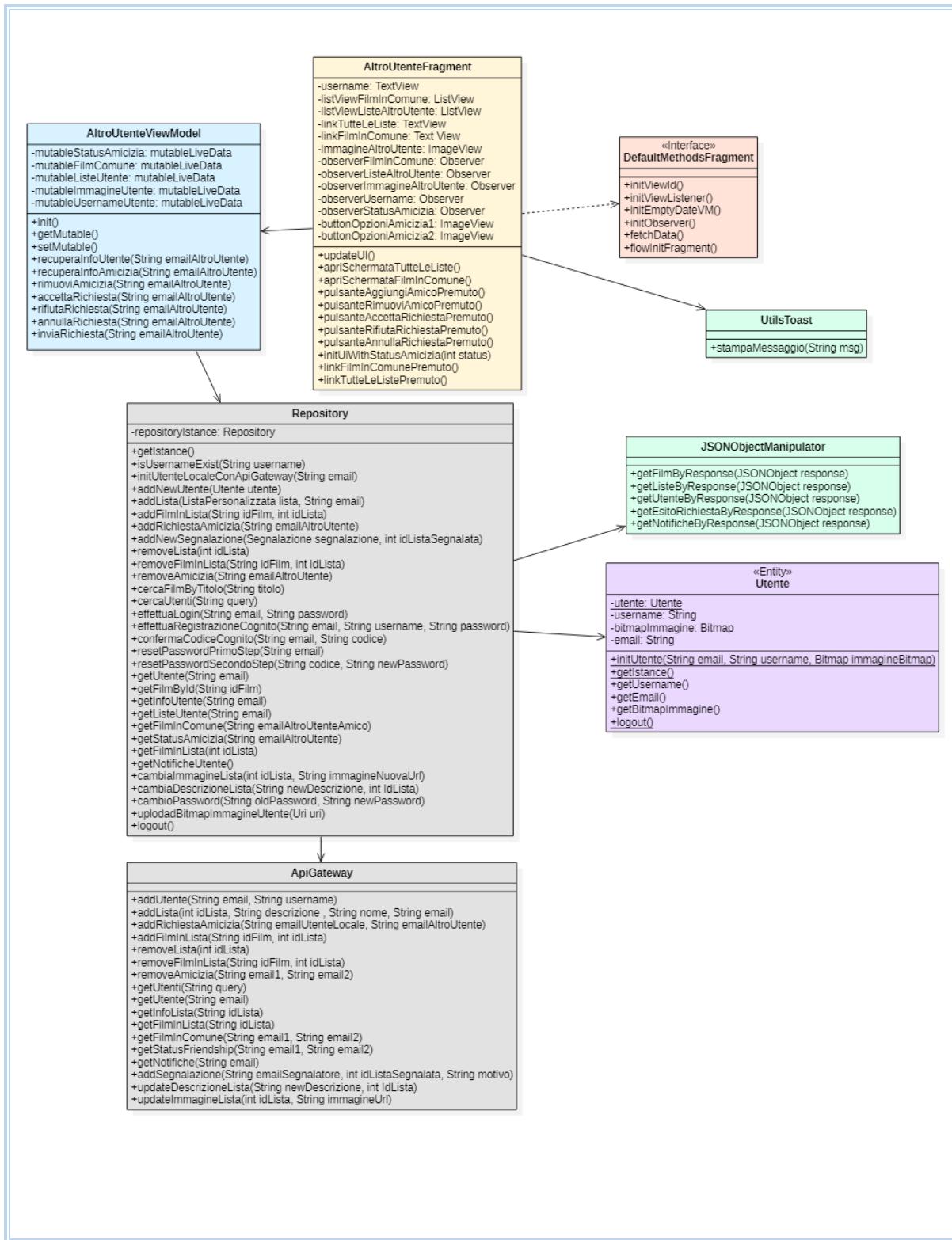
2.2.1.17 Logout



2.2.1.18 Modifica della descrizione di una lista

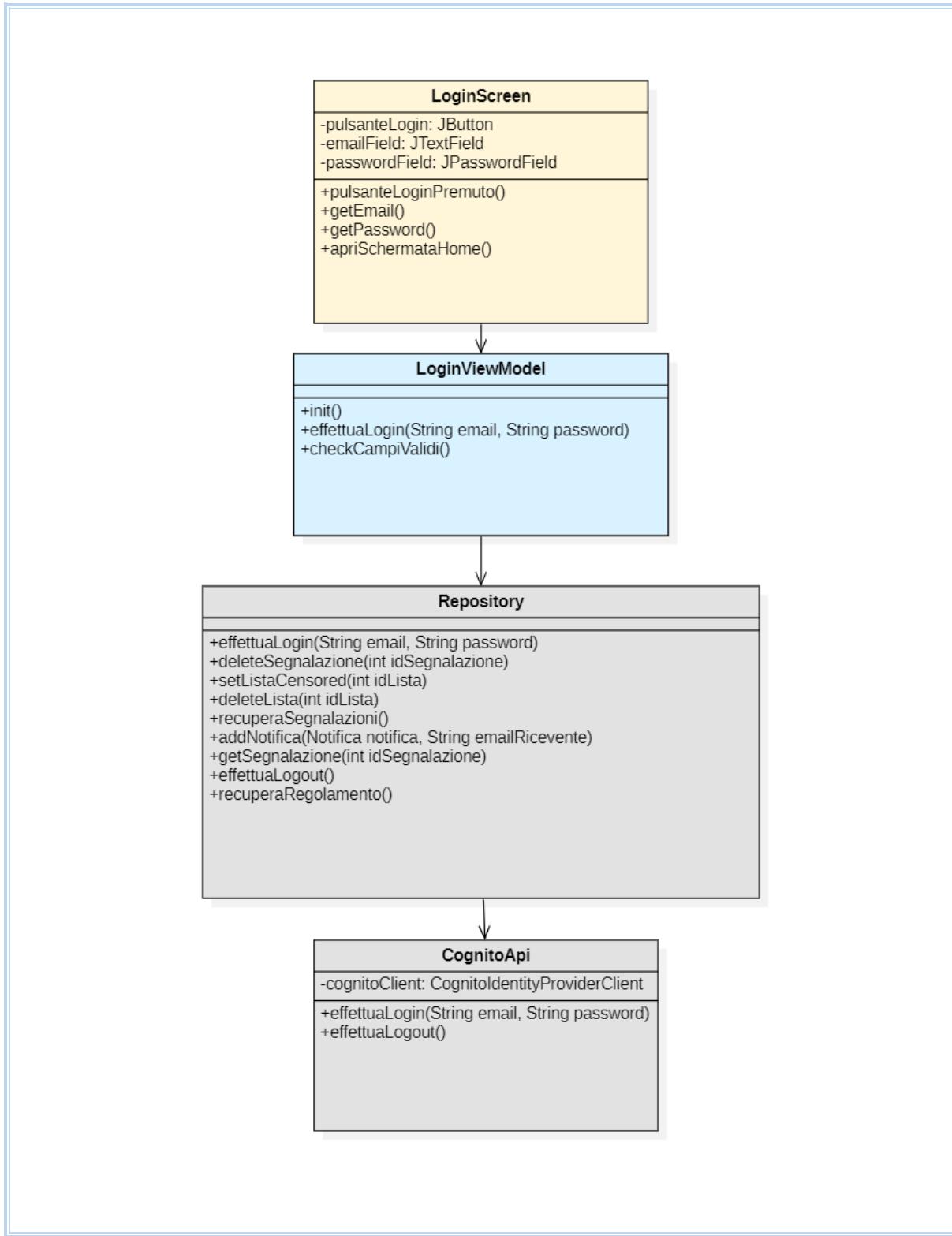


2.2.1.19 Rimuovere amicizia

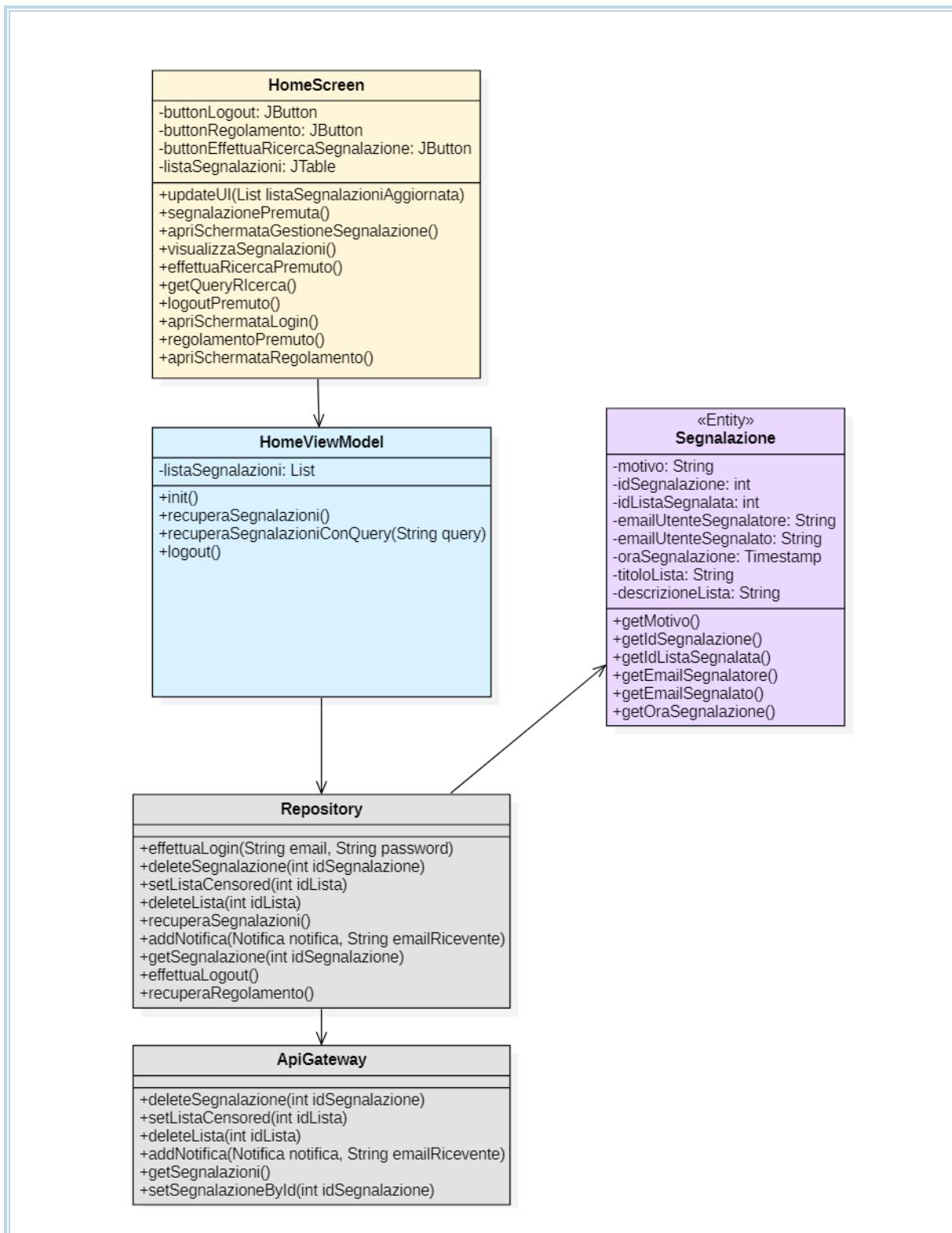


2.2.2 Desktop

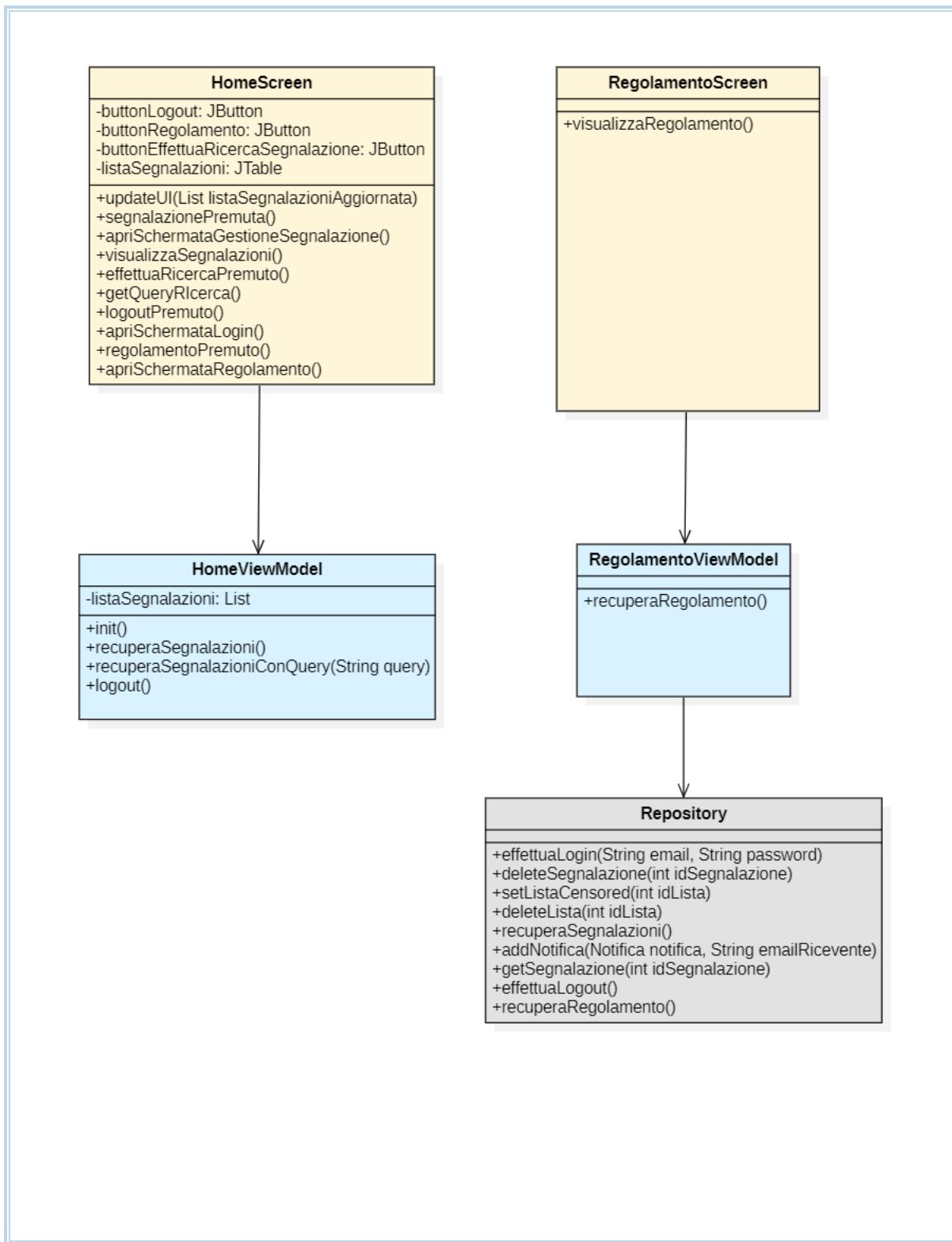
2.2.2.1 Accesso al sistema



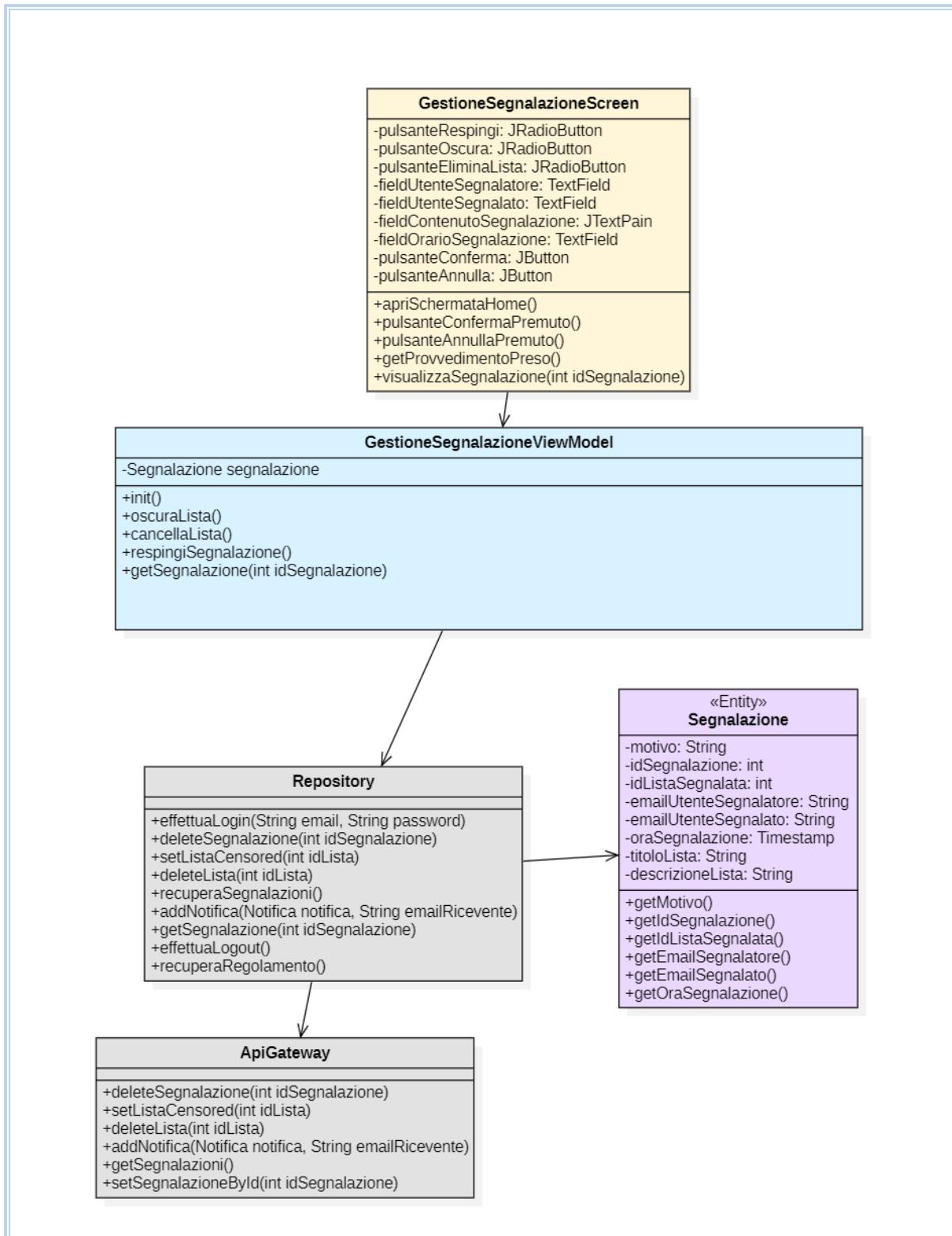
2.2.2.2 Visualizza una segnalazione



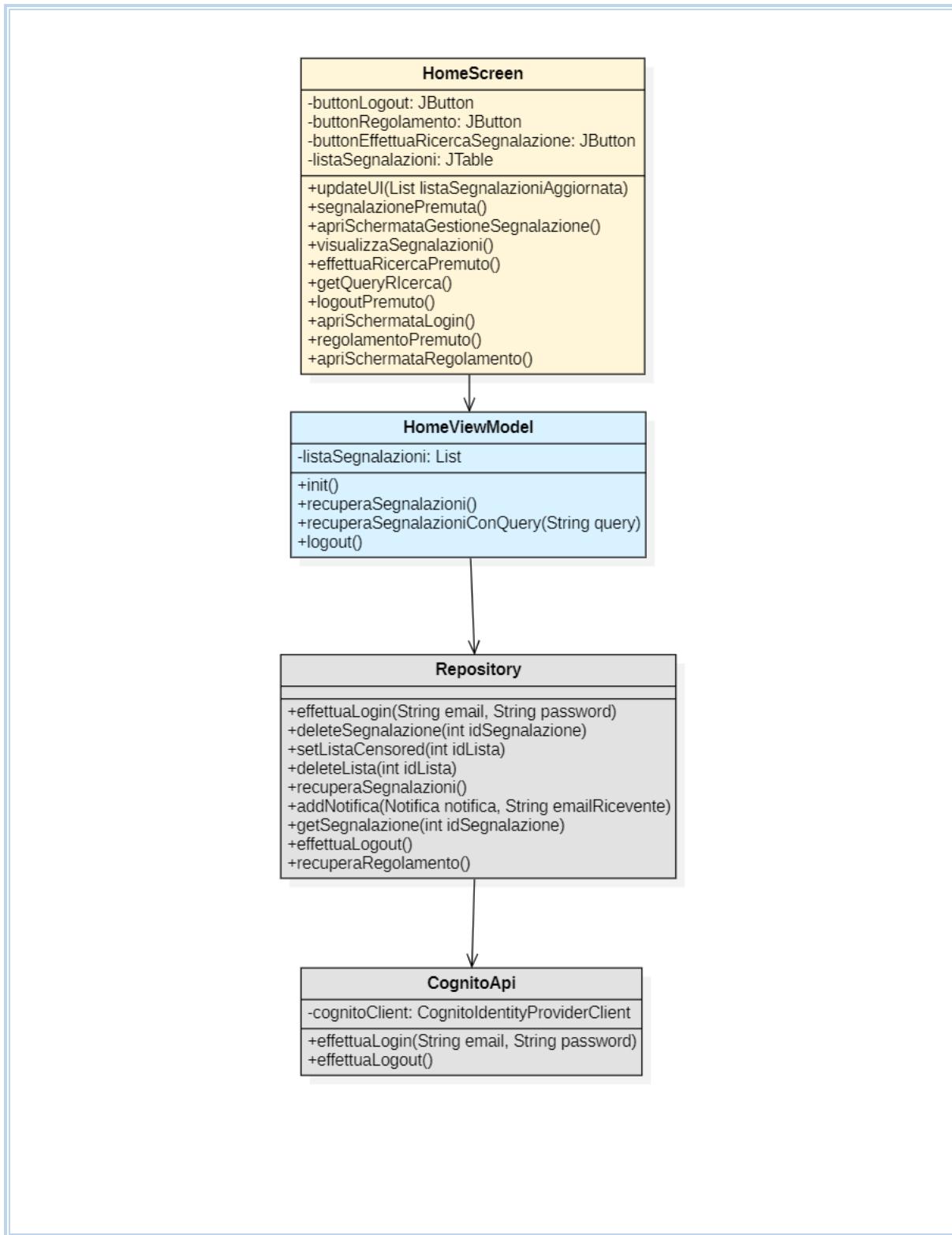
2.2.2.3 Visualizza il regolamento



2.2.2.4 Gestione segnalazione



2.2.2.5 Logout



2.3 CRC Cards

In questo paragrafo verranno mostrate tramite le CRC le classi, il loro funzionamento, le loro responsabilità e i legami che avranno con le altre classi.

Per pura comodità verranno rappresentate solo le cards non di interfaccia, dato che queste ultime avranno come unica responsabilità quella di rappresentare le informazioni per consentire la visualizzazione all'utente.

Per le classi “ViewModel”, verranno descritte tramite un'unica CRC card generica che generalizza il funzionamento di un generico “ViewModel”. La suddetta CRC card è applicabile a tutte le classi “ViewModel”.

2.3.1 Android

2.3.1.1 SchermataViewModelGenerico

| | |
|---|---------------------------------|
| <i>Nome classe</i> SchermataViewModelGenerico | <i>Superclasse</i> ViewModel |
| Sottoclassi Nessuna... | |
| Responsabilità La classe dovrà contenere le informazioni che verranno poi mostrate dalla view. La classe dovrà tenere la view aggiornata dei cambiamenti sui dati. | |
| Collaborazioni La view (Fragment o Activity) associata al view model. Repository. | |



2.3.1.2 Repository

| | |
|--|-----------------------------------|
| <p><i>Nome classe</i> Repository</p> | <p><i>Superclasse</i> ...</p> |
| <p>Sottoclassi Nessuna...</p> | |
| <p>Responsabilità La classe repository fornisce una comoda risorsa per interfacciarsi con i servizi esterni. Si occupa di leggere le risposte che arrivano dall'esterno, decodificarle e successivamente formulare una risposta leggibile per il viewModel richiedente. La classe si occupa delle operazioni per l'account come la modifica della password e il recupero della stessa. Inoltre gestisce le cache del sistema.</p> | |
| <p>Collaborazioni Con tutte le classi “API” Con JSONObjectManipulator (Interprete per le risposte JSONObject) Con tutte le classi “Entity” CacheLocali per Film, ListePersonalizzate, BitmapDecodificati.</p> | |



2.3.1.3 S3Class

| | |
|--|---------------------------|
| <i>Nome classe</i> S3Class | <i>Superclasse</i> ... |
| Sottoclassi Nessuna... | |
| Responsabilità La classe offre la possibilità al client di utilizzare il servizio di AmazonAwsS3 il quale consente di caricare e scaricare file dal cloud tramite un codice di riconoscimento. | |
| <i>Collaborazioni</i> Repository | |

2.3.1.4 ApiGateway

| | |
|--|---------------------------|
| <i>Nome classe</i> ApiGateway | <i>Superclasse</i> ... |
| Sottoclassi Nessuna... | |
| Responsabilità La classe offre la possibilità al client di utilizzare Amazon AWS Lambda, il quale è un servizio di elaborazione serverless che ti permette di eseguire il codice senza effettuare il provisioning o gestire i server. Si occupa inoltre di costruire in modo corretto gli url per effettuare le richieste lambda. | |
| <i>Collaborazioni</i> UrlStringBuilder Repository | |



2.3.1.5 UrlStringBuilder

| | |
|---|---------------------------|
| <i>Nome classe</i> UrlStringBuilder | <i>Superclasse</i> ... |
| Sottoclassi Nessuna... | |
| Responsabilità La classe si occupa di costruire gli url per effettuare le richieste Lambda. | |
| Collaborazioni ApiGateway | |

2.3.1.6 MovieApi

| | |
|--|---------------------------|
| <i>Nome classe</i> MovieApi | <i>Superclasse</i> ... |
| Sottoclassi Nessuna... | |
| Responsabilità La classe offre la possibilità al client di utilizzare il servizio offerto da “The Open Movie Database”. La classe si occuperà di leggere le risposte del servizio e tramite opportune trasformazioni fornire risultati leggibili per poter essere visualizzati. | |
| Collaborazioni Repository | |



2.3.1.7 Utente

| | |
|---|--------------------------------------|
| <p><i>Nome classe</i></p> <p>Utente</p> | <p><i>Superclasse</i></p> <p>...</p> |
| <p>Sottoclassi</p> <p>Nessuna...</p> | |
| <p>Responsabilità</p> <p>La classe rappresenta le informazioni di un utente e in particolare le informazioni dell'utente attualmente collegato.</p> <p>Offre la possibilità di effettuare il Logout e di stampare in console le informazioni di un utente.</p> | |
| <p>Collaborazioni</p> <p>Amplify, GoogleSignIn, Profile (Sono servizi esterni per fare Logout)</p> <p>NotificheService</p> | |

2.3.1.8 ListaPersonalizzata

| | |
|--|--------------------------------------|
| <p><i>Nome classe</i></p> <p>ListaPersonalizzata</p> | <p><i>Superclasse</i></p> <p>...</p> |
| <p>Sottoclassi</p> <p>Nessuna...</p> | |
| <p>Responsabilità</p> <p>La classe rappresenta le informazioni di una lista personalizzata.</p> <p>Offre la possibilità di stampare in console le informazioni di una lista personalizzata.</p> | |
| <p>Collaborazioni</p> <p>Nessuna...</p> | |



2.3.1.9 Film

| | |
|---|---------------------------|
| <i>Nome classe</i> Film | <i>Superclasse</i> ... |
| Sottoclassi Nessuna... | |
| Responsabilità La classe rappresenta le informazioni di un film. Offre la possibilità di stampare in console le informazioni di un film. | |
| Collaborazioni Nessuna... | |

2.3.1.10 Notifica

| | |
|---|---------------------------|
| <i>Nome classe</i> Notifica | <i>Superclasse</i> ... |
| Sottoclassi Nessuna... | |
| Responsabilità La classe rappresenta le informazioni di una notifica. Offre la possibilità di stampare in console le informazioni di una notifica. | |
| Collaborazioni Nessuna... | |



2.3.1.11 Segnalazione

| | |
|---|---------------------------|
| <i>Nome classe</i> Segnalazione | <i>Superclasse</i> ... |
| Sottoclassi <i>Nessuna...</i> | |
| Responsabilità La classe rappresenta le informazioni di una segnalazione. Offre la possibilità di stampare in console le informazioni di una segnalazione. | |
| Collaborazioni <i>Nessuna...</i> | |

2.3.1.12 BackgroundFilmCache

| | |
|--|---------------------------|
| <i>Nome classe</i> BackgroundFilmCache | <i>Superclasse</i> ... |
| Sottoclassi <i>Nessuna...</i> | |
| Responsabilità La classe rappresenta la cache per il film. | |
| Collaborazioni Repository | |



2.3.1.13 BackgroundBitmapCache

| | |
|---|---------------------------|
| <i>Nome classe</i> BackgroundBitmapCache | <i>Superclasse</i> ... |
| Sottoclassi Nessuna... | |
| Responsabilità La classe rappresenta la cache per i bitmap scaricati dal servizio S3. | |
| <i>Collaborazioni</i> Repository | |

2.3.1.14 CacheLocali

| | |
|---|---------------------------|
| <i>Nome classe</i> CacheLocali | <i>Superclasse</i> ... |
| Sottoclassi Nessuna... | |
| Responsabilità La classe rappresenta la cache per le liste Personalizzate dell'utente locale. | |
| <i>Collaborazioni</i> Repository | |



2.3.1.15 JSONObjectManipulator

| | |
|---|---------------------------|
| <i>Nome classe</i> JSONObjectManipulator | <i>Superclasse</i> ... |
| Sottoclassi Nessuna... | |
| Responsabilità La classe si occupa di trasformare le risposte dal formato JSONObject ad un formato rappresentabile. | |
| Collaborazioni Repository | |

2.3.1.16 DefaulMethodInterface

| | |
|---|---------------------------|
| <i>Nome classe</i> DefaulMethodInterface | <i>Superclasse</i> ... |
| Sottoclassi Nessuna... | |
| Responsabilità La classe rappresenta un'interfaccia che indica i contratti dei metodi standard che un qualunque fragment dovrebbe implementare per rispettare il pattern MVVM, indica inoltre l'ordine consigliato con cui dovrebbero essere invocati i metodi. | |
| Collaborazioni Tutte le classi "Fragment" | |



2.3.1.17 NotificheService

| | |
|--|---------------------------|
| <i>Nome classe</i> NotificheService | <i>Superclasse</i> ... |
| Sottoclassi Nessuna... | |
| Responsabilità La classe è responsabile del controllo per le notifiche utente. | |
| Collaborazioni Repository | |



2.3.2 Desktop

2.3.2.1 Repository

| | |
|---|---------------------------|
| <i>Nome classe</i> Repository | <i>Superclasse</i> ... |
| Sottoclassi Nessuna... | |
| Responsabilità La classe repository fornisce una comoda risorsa per interfacciarsi con i servizi esterni. Si occupa di leggere le risposte che arrivano dall'esterno, fornendole in maniera leggibile al viewModel. | |
| Collaborazioni Con tutte le classi “API” Con tutte le classi “Entity” | |

2.3.2.2 Notifica

| | |
|---|---------------------------|
| <i>Nome classe</i> Notifica | <i>Superclasse</i> ... |
| Sottoclassi Nessuna... | |
| Responsabilità La classe fa parte del gruppo delle entity per cui rappresenta una notifica che gli utenti ricevono in caso di provvedimento preso nei loro confronti. | |
| Collaborazioni Repository, GestioneSegnalazioneViewModel | |



2.3.2.3 Segnalazione

| | |
|--|---------------------------|
| <i>Nome classe</i> Segnalazione | <i>Superclasse</i> ... |
| Sottoclassi Nessuna... | |
| Responsabilità La classe fa parte del gruppo delle entity per cui rappresenta una segnalazione effettuata da parte di un utente nei confronti di una lista posseduta da un altro utente. | |
| <i>Collaborazioni</i> Repository | |

2.3.2.4 CognitoApi

| | |
|--|---------------------------|
| <i>Nome classe</i> CognitoApi | <i>Superclasse</i> ... |
| Sottoclassi Nessuna... | |
| Responsabilità La classe fa parte del gruppo delle API per cui offre all'esterno dei metodi che consentono di interagire direttamente con un provider di servizi, in questo caso con il servizio di Cognito. | |
| <i>Collaborazioni</i> Repository | |



2.3.2.1 ApiGateway

| | |
|---|---------------------------|
| <i>Nome classe</i> ApiGateway | <i>Superclasse</i> ... |
| Sottoclassi Nessuna... | |
| Responsabilità La classe fa parte del gruppo delle API per cui offre all'esterno dei metodi che consentono di interagire direttamente con un provider di servizi, in questo caso con il servizio di ApiGateway. Si occupa anche di fornire risposte leggibili per i richiedenti. | |
| Collaborazioni Repository, CineMatesClient | |

2.3.2.1 CineMatesClient

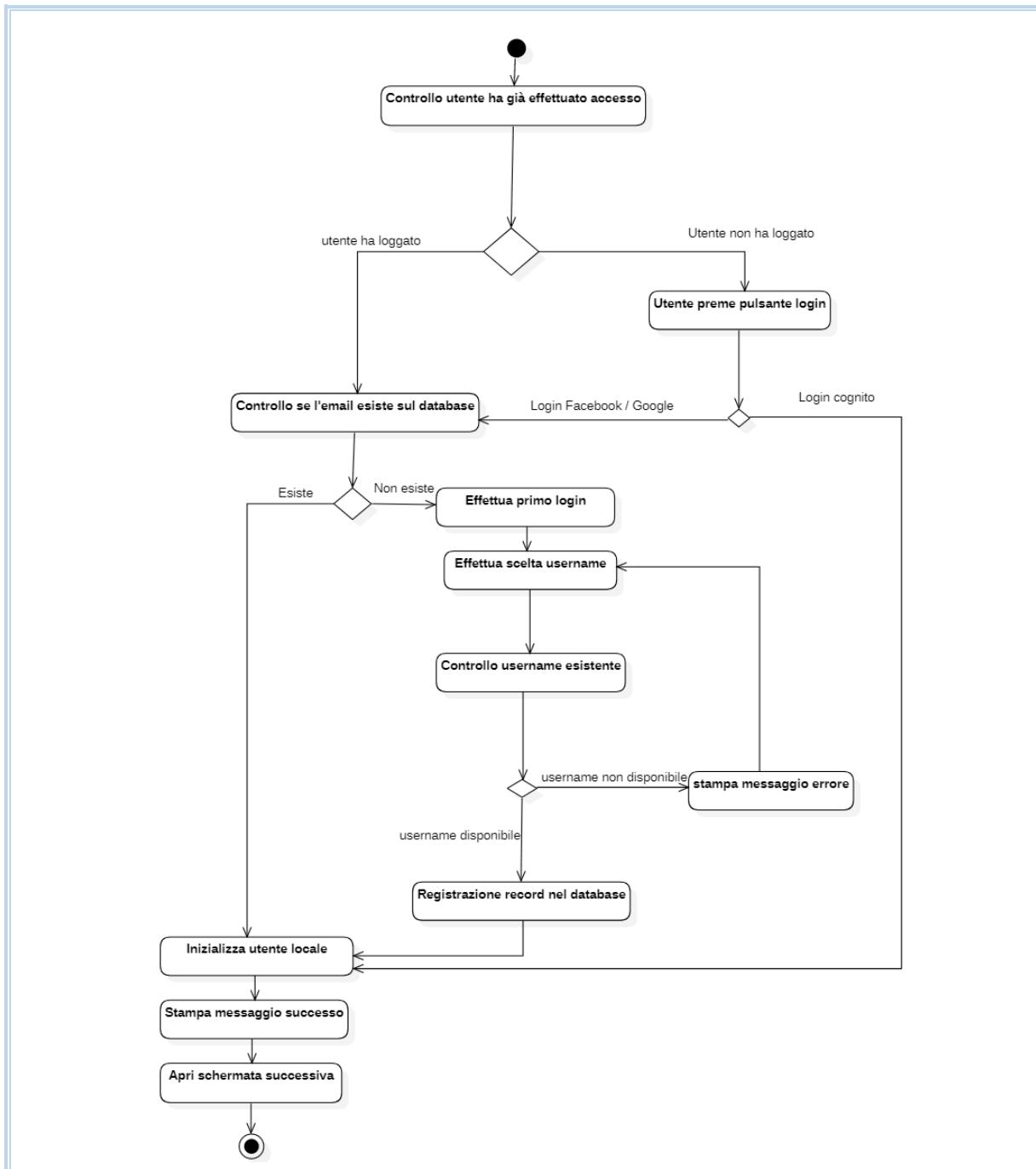
| | |
|--|---------------------------|
| <i>Nome classe</i> CineMatesClient | <i>Superclasse</i> ... |
| Sottoclassi Nessuna... | |
| Responsabilità La classe si occupa di costruire correttamente le richieste da effettuare per il servizio Lambda, inoltre traduce le risposte in maniera che sia comprensibili. | |
| Collaborazioni ApiGateway | |



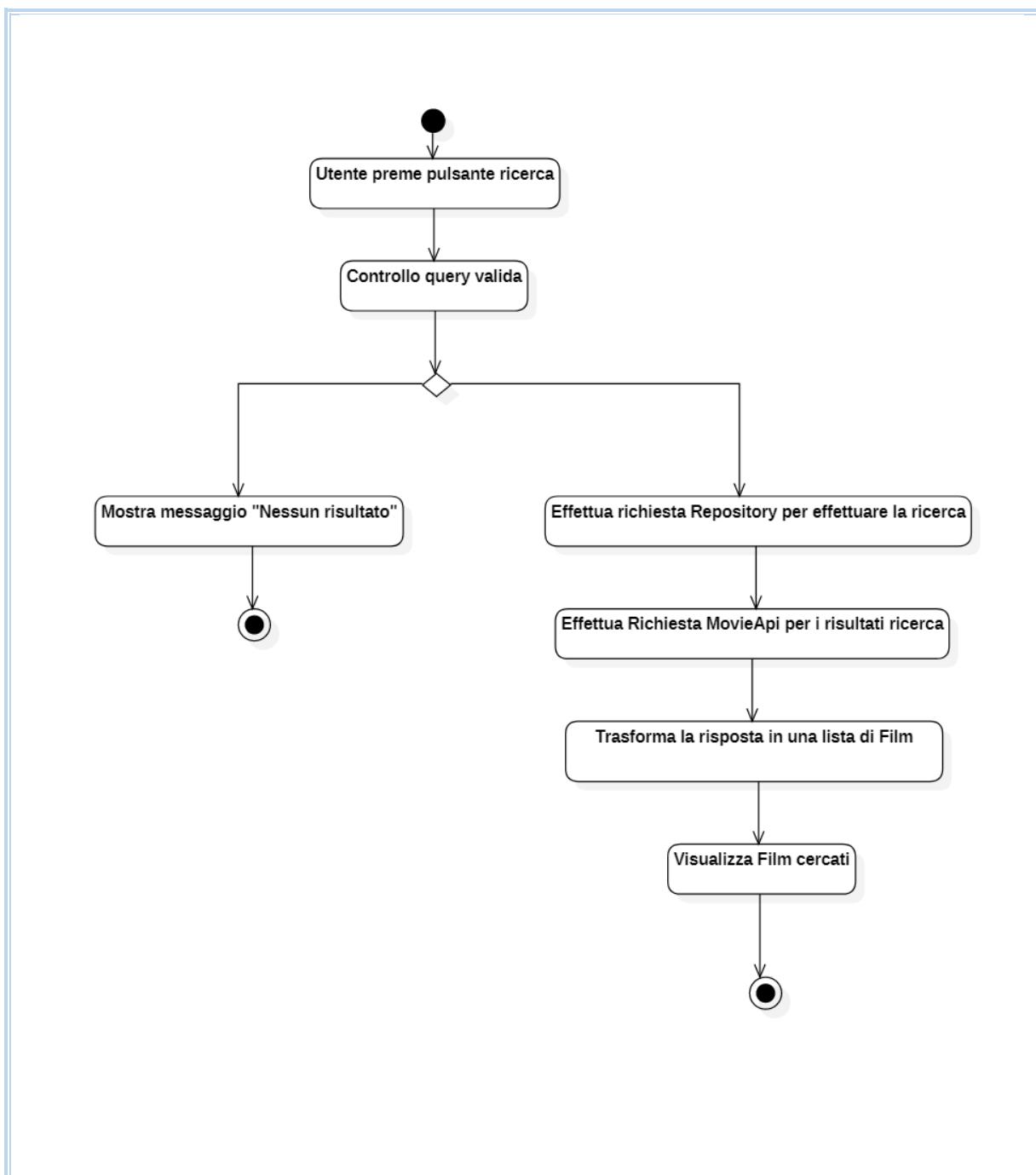
2.4 Diagramma di stato di design

2.4.1 Android

2.4.1 Login

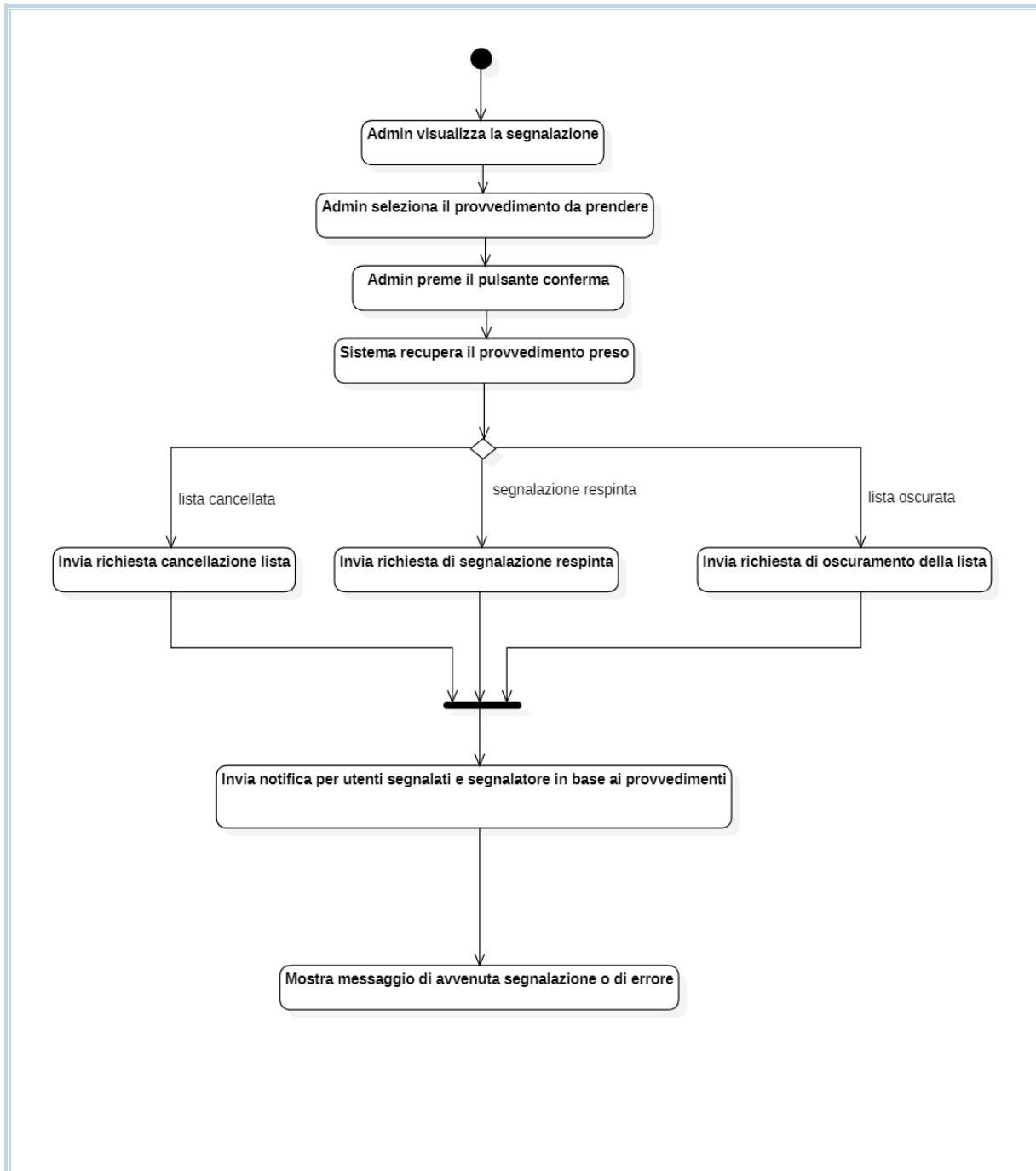


2.4.2 Ricerche film



2.4.2 Desktop

2.4.2.1 Gestione segnalazione

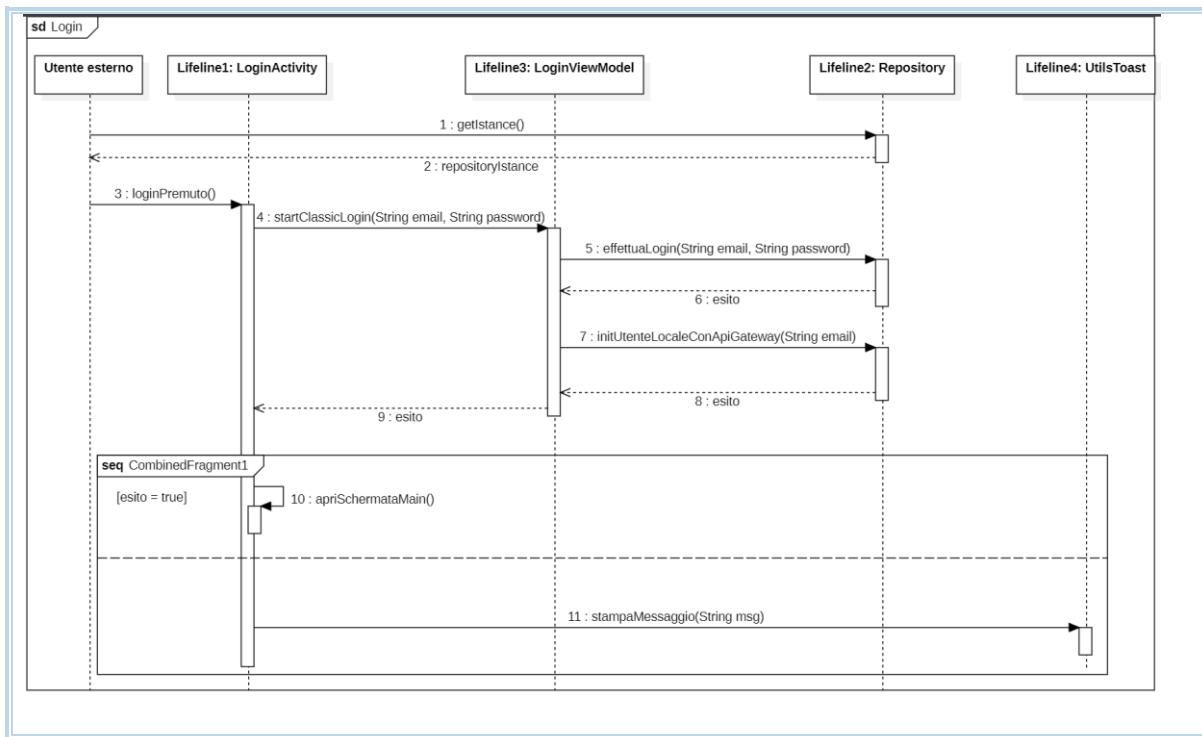


2.5 Diagramma di sequenza di design

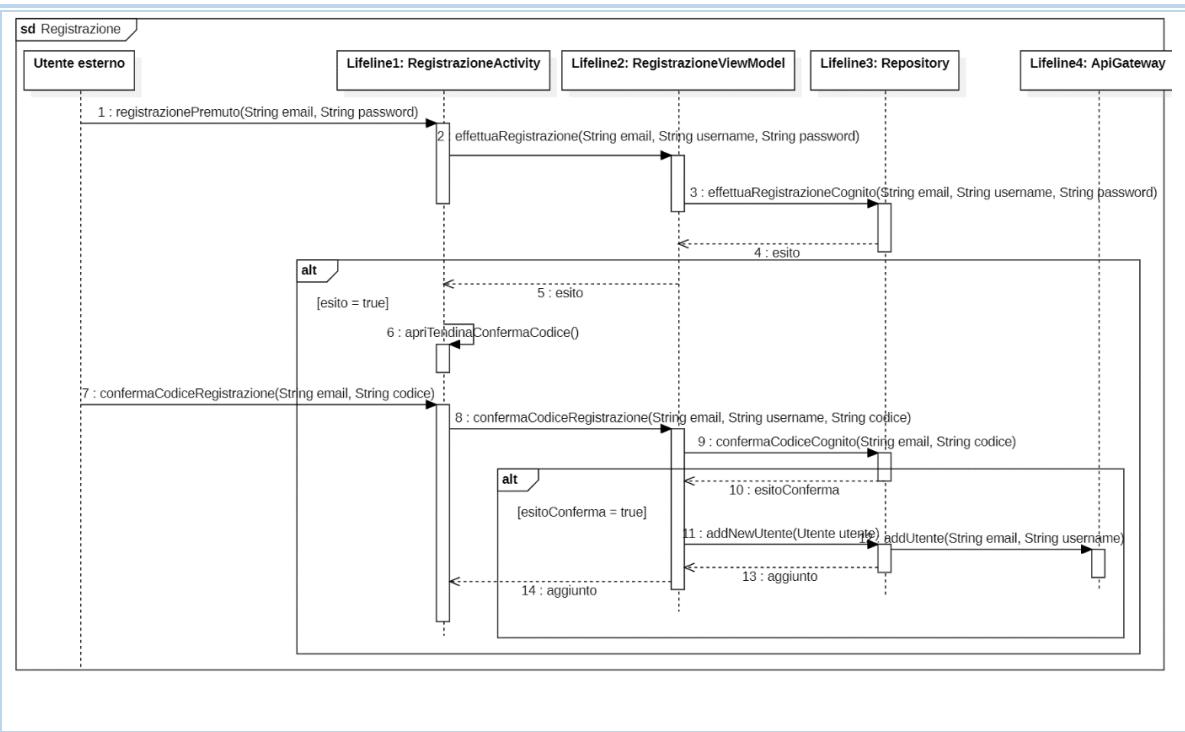
CONVENZIONE ADOTTATA: in alcuni diagrammi è stata volontariamente omessa la gestione degli eventi nel caso in cui la risposta ad una funzione risultava essere FALSE per motivi di leggibilità del diagramma stesso. Ad ogni modo, ogni qualvolta si presenti un esito == False verrà mostrato un messaggio di errore all'utente anche se non specificato.

2.5.1 Android

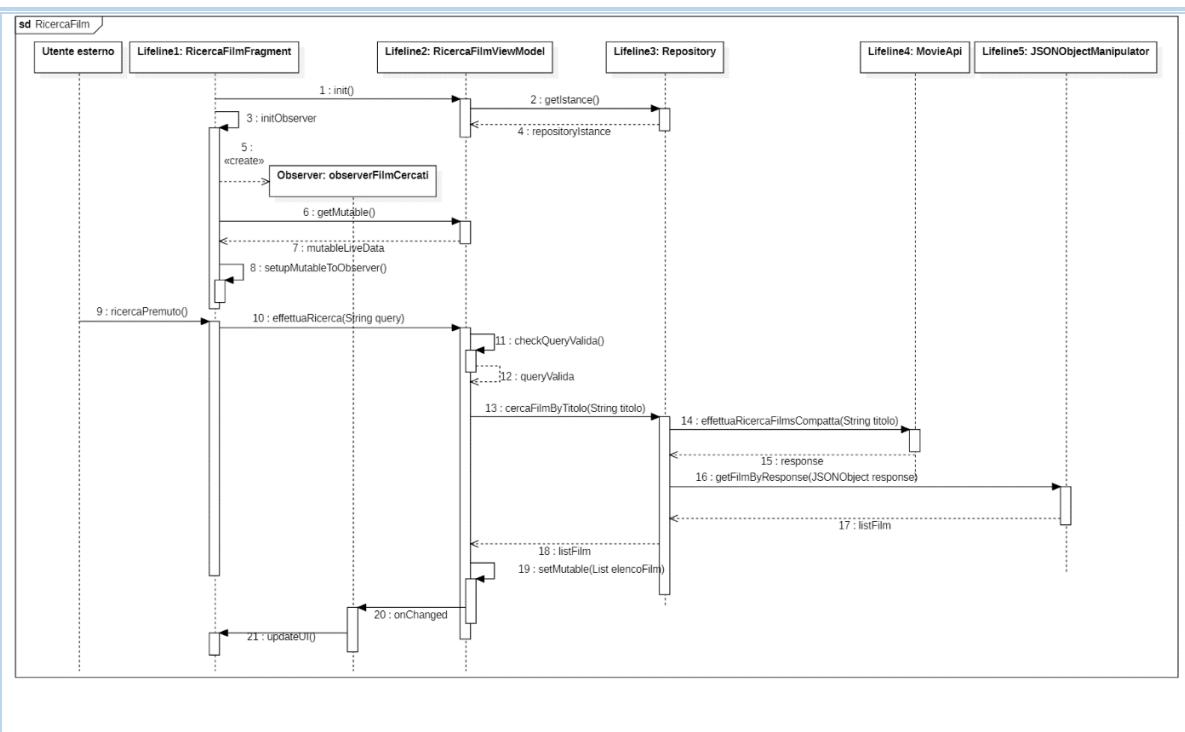
2.5.1 Login



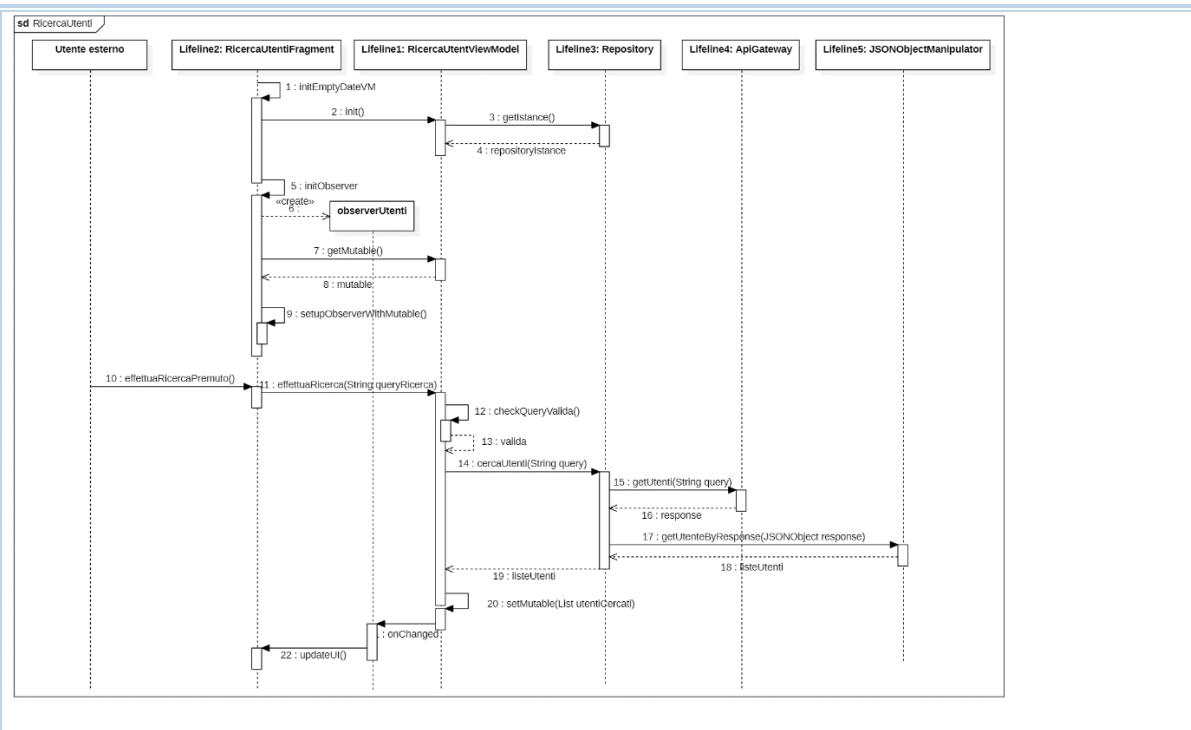
2.5.2 Registrazione



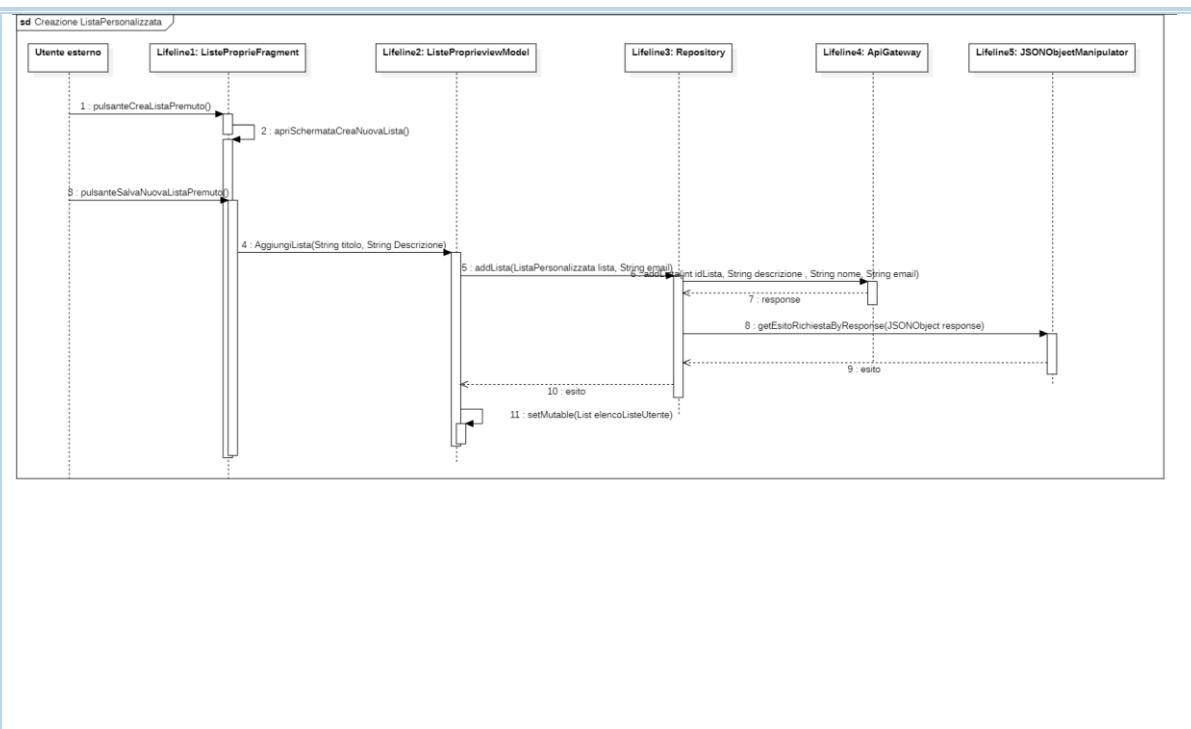
2.5.3 Ricerca film



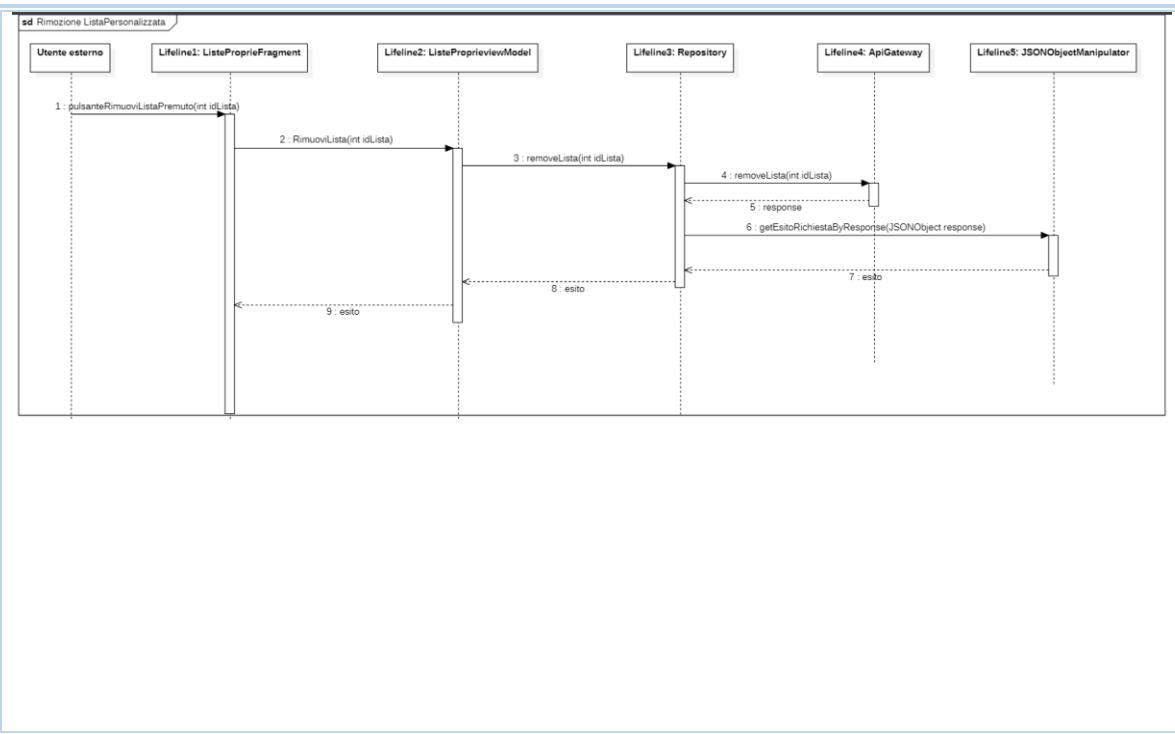
2.5.4 Ricerca utenti



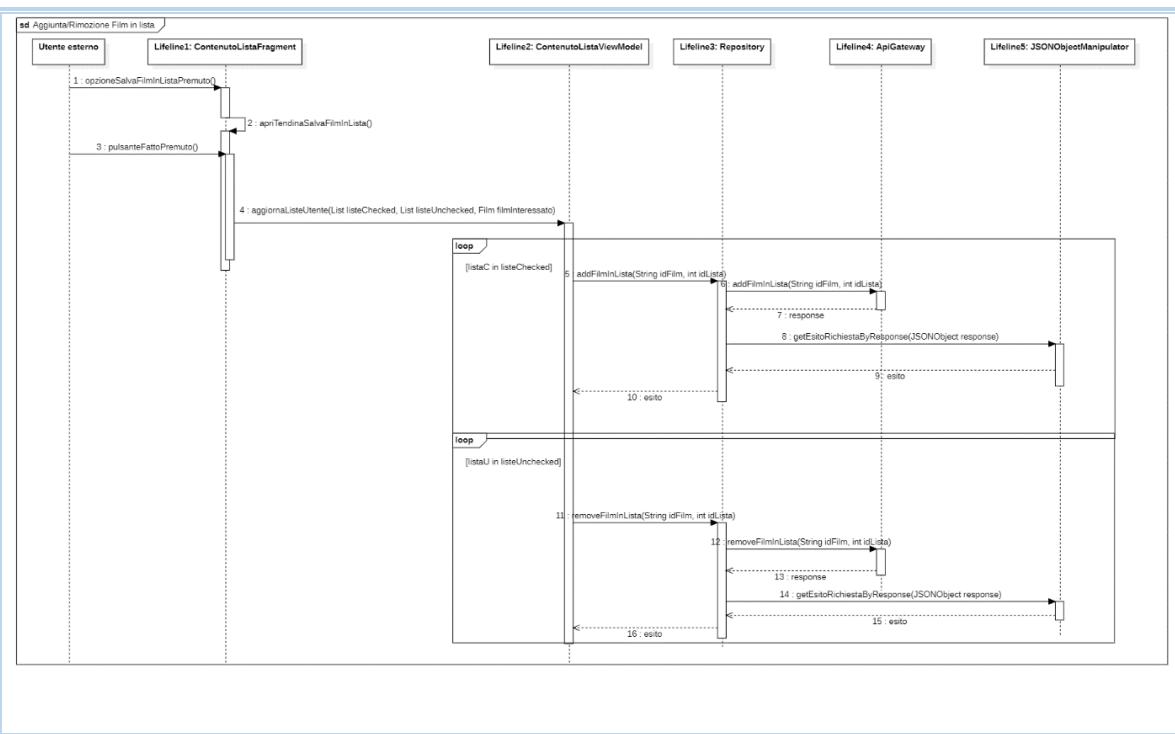
2.5.5 Crea lista personalizzata



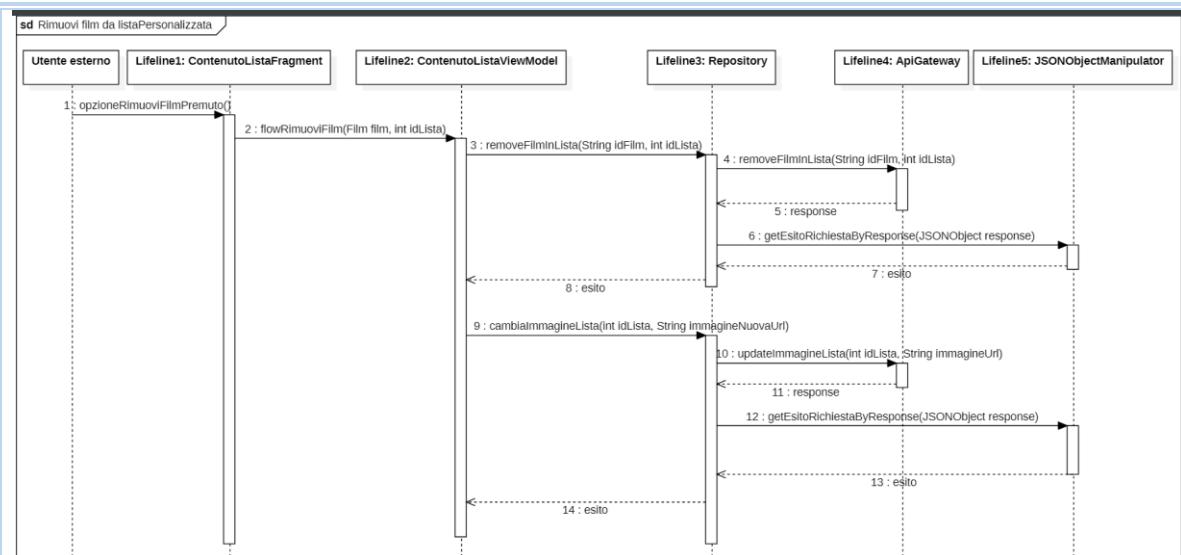
2.5.6 Rimozione lista personalizzata



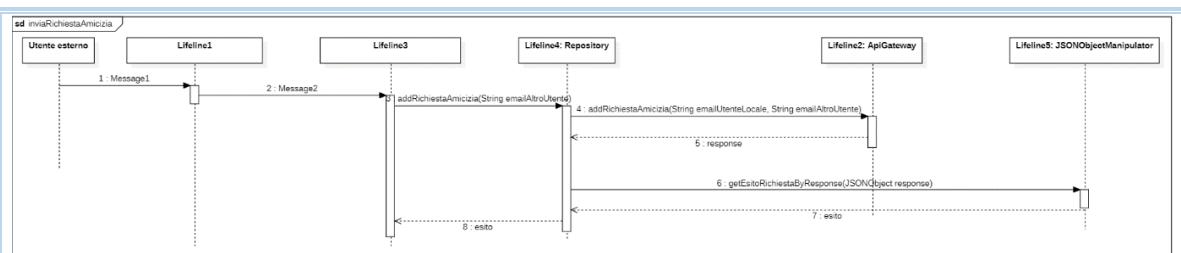
2.5.7 Aggiunta-Rimozione film da lista



2.5.8 Rimozione film da lista (2 metodo)

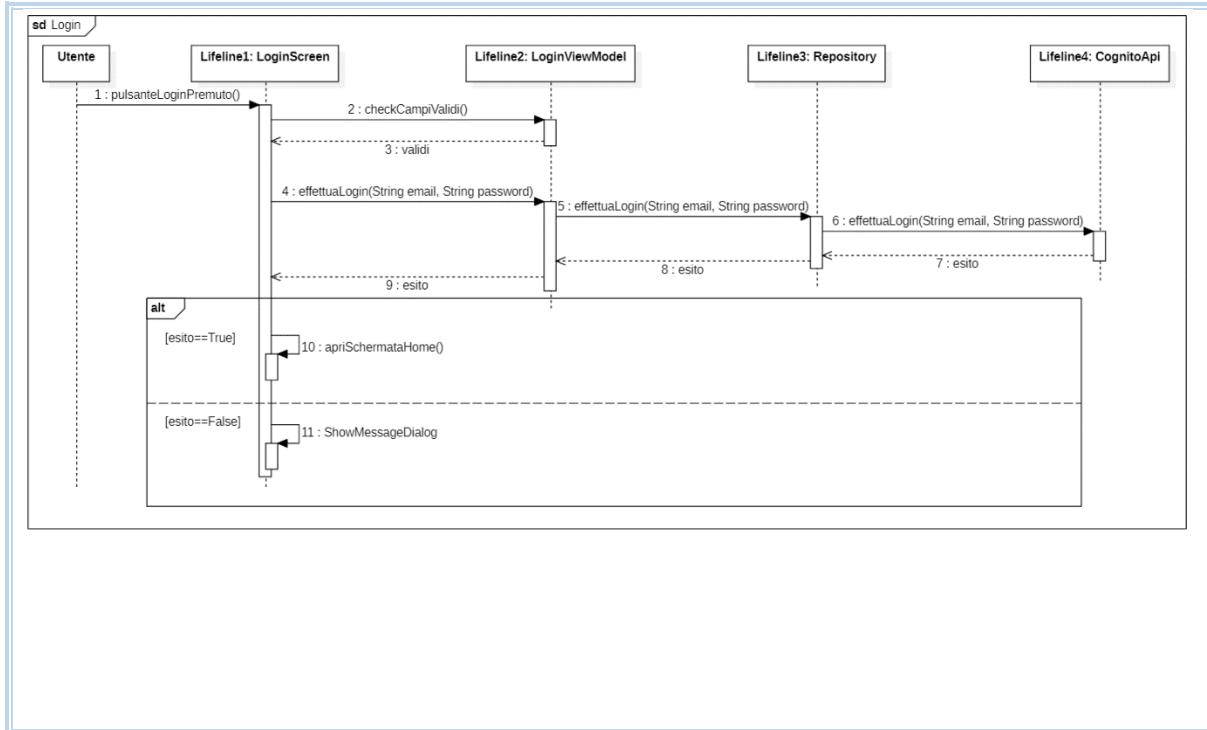


2.5.9 Invia richiesta amici

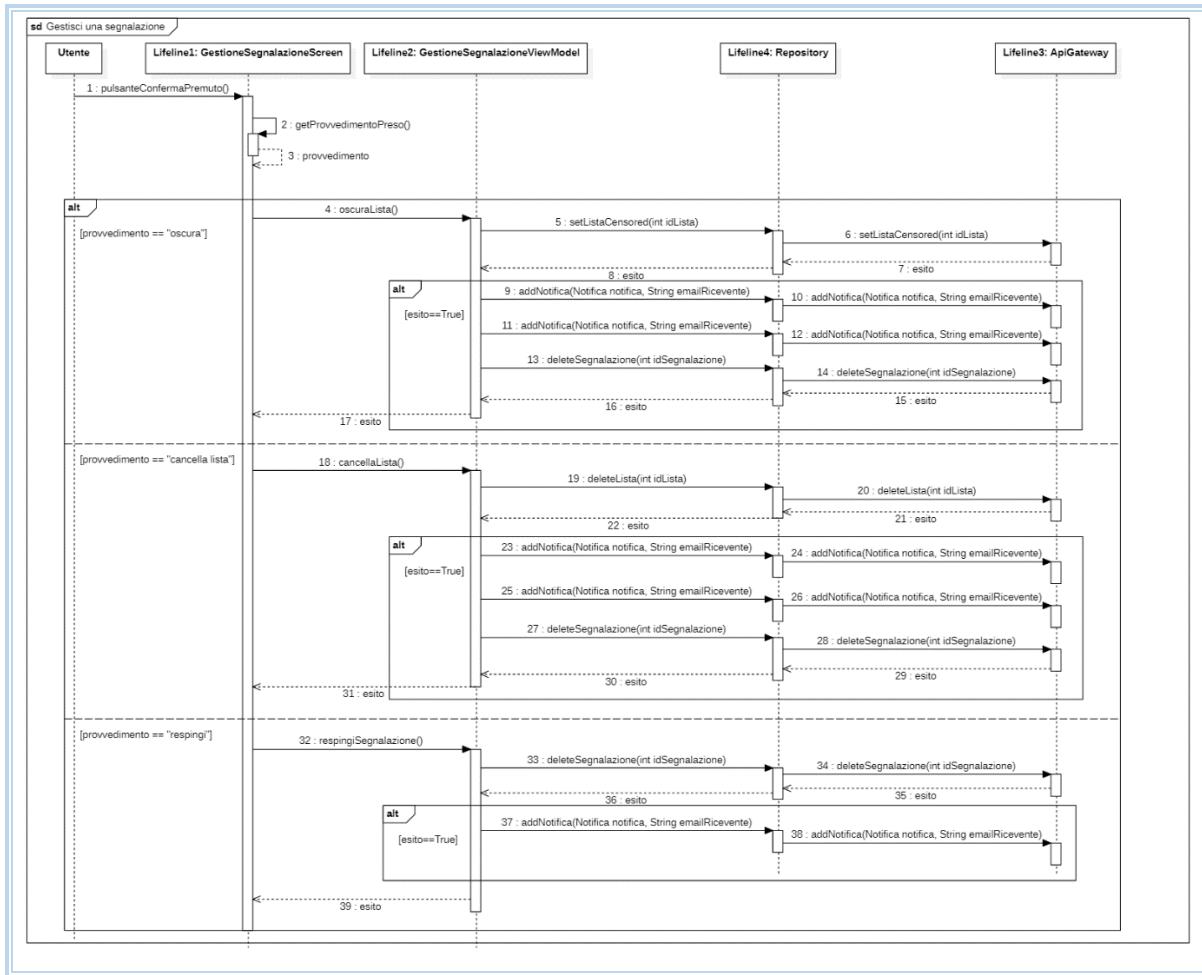


2.5.2 Desktop

2.5.2.1 Login



2.5.2.2 Gestione segnalazione



4 Documento di Testing del sistema

4.1 Test plan per System Testing applicazione Android

4.1.1 Accesso al sistema

| Test ID | AND_TEST01 | |
|---|--|-----------|
| Test Name | Test di accesso al sistema | |
| Test Description | Obiettivo è di verificare il funzionamento corretto dell'accesso al sistema | |
| Input | Risultato desiderato | Risultato |
| Scelgo autenticazione con Google per prima volta, scegliendo un account Google “infoemfworkshop@gmail.com”. | Accesso al sistema riuscito. Mostra schermata Dialog per scelta di username. | Positivo |
| Scelgo autenticazione con Google, scegliendo un account Google “infoemfworkshop@gmail.com”, avendo già scelto username precedentemente. | Accesso al sistema riuscito. Mostra schermata Home. | Positivo |
| Scelgo autenticazione con Facebook, scegliendo un account Facebook “esempio003@hotmail.com” | Accesso al sistema riuscito. Mostra schermata Dialog per scelta di username. | Positivo |
| Scelgo autenticazione con Facebook, scegliendo un account Facebook “esempio003@hotmail.com”, avendo già scelto username precedentemente | Accesso al sistema riuscito. Mostra schermata Home. | Positivo |
| Compilo campo email con “dimalozyak1997@gmail.com” e la password “qawsedrf” | Accesso al sistema riuscito. Mostra schermata Home. | Positivo |



| | | |
|--|--|----------|
| Compilo campo email con "dimalozyak1997@gmail.com" e la password " qawsedrft " | Accesso al sistema fallito. Messaggio mostrato "Email o Password non validi" | Positivo |
| Compilo campo email con "dimalozyak1997@gmail.com" e la password vuota | Accesso al sistema fallito. Messaggio mostrato "Email o Password non validi" | Positivo |
| Compilo campo email con spazio vuoto e la password " qawsedrf " | Accesso al sistema fallito. Messaggio mostrato "Email o Password non validi" | Positivo |



4.1.2 Registrazione al sistema

| Test ID | AND_TEST02 | |
|--|--|-----------|
| Test Name | Test di registrazione al sistema | |
| Test Description | Obiettivo è di verificare il funzionamento corretto della registrazione al sistema | |
| Input | Risultato desiderato | Risultato |
| <p>Effettuo la prima registrazione compilando i campi con:</p> <p>Username: cineMates20 password: qawsedrf conferma password: qawsedrf email: dimalozyak1997@gmail.com</p> | Registrazione riuscita, viene inviato il codice di conferma all'email e viene visualizzato il Dialog per la conferma del codice. | Positivo |
| Conferma la registrazione usando il codice ricevuto sull'email | Mostra il messaggio "Registrazione effettuata con il successo" e mostra la schermata Login. | Positivo |
| Conferma la registrazione usando il codice diverso da quello ricevuto sull'email. | Mostra il messaggio "Registrazione non confermata" | Positivo |
| <p>Effettuo la registrazione compilando con i campi:</p> <p>Username: cineMates20 password: qawsedrf conferma password: qawsedrf email: dimalozyakxxx@gmail.com</p> | Mostra il messaggio "Username non è disponibile" | Positivo |
| <p>Effettuo la registrazione compilando con i campi:</p> <p>Username: cineMatesNew password: qawsedrf conferma password: qawsedrf email: dimalozyak1997@gmail.com</p> | Mostra il messaggio "Email è già utilizzata" | Positivo |
| <p>Effettuo la registrazione compilando con i campi:</p> <p>Username: cinemaPassione password: qawsedrf conferma password: qawsedrfs email: cinemaPassione@gmail.com</p> | Mostra il messaggio "Password e conferma Password non corrispondono" | Positivo |



| | | |
|--|---|----------|
| Effettuo la registrazione compilando con i campi: Username: cinemaPassione password: (vuoto) conferma password: (vuoto) email: cinemaPassione@gmail.com | Mostra i messaggi: “Campi mancanti o requisiti non rispettati” “Lunghezza password deve essere compresa tra 8 e 30 caratteri” | Positivo |
| | | |



4.1.3 Recupero password dimenticata

| Test ID | AND_TEST03 | |
|--|--|-----------|
| Test Name | Test di recupero password dimenticata | |
| Test Description | Obiettivo è di verificare il funzionamento corretto di recupero password dimenticata | |
| Input | Risultato desiderato | Risultato |
| Compilo campo E-mail con "dimalozyak1997@gmail.com" e premo invia il codice di recupero. | Viene inviato il codice di conferma sull'email e mostrato Dialog successivo | Positivo |
| Compilo il campo conferma codice con il codice ricevuto e campo nuova password con "qwertasdf" e premo Salva | Mostra il messaggio "Password cambiata con successo" | Positivo |
| Compilo il campo conferma codice con il codice ricevuto e campo nuova password con (vuoto) e premo Salva | Mostra il messaggio "Campi mancanti o requisiti non rispettati" | Positivo |
| Compilo il campo E-mail con "qwertasdf" | Mostra il messaggio "Campi mancanti o requisiti non rispettati" | Positivo |



4.1.4 Effettuare la ricerca di film

| Test ID | AND_TEST04 | |
|---|---|-----------|
| Test Name | Test ricerca film | |
| Test Description | Obiettivo è di verificare il funzionamento corretto di ricerca film | |
| Input | Risultato desiderato | Risultato |
| Compilo il campo ricerca con il titolo "Venom" | Mostra la lista di film contenenti la parola "Venom" nel titolo | Positivo |
| Compilo il campo di ricerca con il titolo (vuoto) o con stringa casuale "asdaddjkjad" | Mostra il messaggio "Nessun risultato trovato" | Positivo |



4.1.5 Creazione di una lista di film personalizzata

| Test ID | AND_TEST05 | |
|--|--|-----------|
| Test Name | Test di creazione di una lista di film personalizzata | |
| Test Description | Obiettivo è di verificare il funzionamento corretto di creazione di una lista di film personalizzata | |
| Input | Risultato desiderato | Risultato |
| Premo il pulsante “ + nuova lista ” nella schermata Home | Mostra la schermata Dialog per la creazione di una nuova lista | Positivo |
| Compilo il campo Titolo con “Nuova lista” e campo descrizione con “la mia nuova lista” e premo salva | Crea nuova lista e mostra il messaggio “la nuova lista è stata creata con successo” | Positivo |
| Premo il pulsante “ + ” nella schermata Liste | Mostra la schermata Dialog per la creazione di una nuova lista | Positivo |
| Compilo il campo Titolo con “Nuova lista 2” e campo descrizione con (vuoto) ” e premo salva | Crea nuova lista e mostra il messaggio “la nuova lista 2 è stata creata con successo” | Positivo |
| Premo “ + crea nuova lista ” nella Dialog “Salva film in una lista” | Mostra la schermata Dialog per la creazione di una nuova lista | Positivo |
| Compilo il campo Titolo con “Nuova lista 3” e campo descrizione con (vuoto) ” e premo salva | Crea nuova lista e mostra il messaggio “la nuova lista 3 è stata creata con successo” | Positivo |



4.1.6 Rimozione di una lista di film personalizzata

| Test ID | AND_TEST06 | |
|---|--|-----------|
| Test Name | Test di rimozione di una lista di film personalizzata | |
| Test Description | Obiettivo è di verificare il funzionamento corretto di rimozione di una lista di film personalizzata | |
| Input | Risultato desiderato | Risultato |
| Scelgo opzione “Cancella lista” della lista “Nuova lista” nella schermata Liste proprie | Cancella la lista e mostra il messaggio “Nuova lista” è stata eliminata con successo | Positivo |



4.1.7 Aggiunta/Rimozione di un film ad/da una lista

| Test ID | AND_TEST07 | |
|---|--|-----------|
| Test Name | Test di Aggiunta/Rimozione di un film ad/da una lista | |
| Test Description | Obiettivo è di verificare il funzionamento corretto di Aggiunta/Rimozione di un film ad/da una lista | |
| Input | Risultato desiderato | Risultato |
| Nel Dialog “Salva film in una lista” seleziono la casella della lista “Nuova lista” e premo Fatto | Salva il film dentro la lista “Nuova lista” e stampa il messaggio “Cambiamenti salvati” | Positivo |
| Nel Dialog “Salva film in una lista” deselecto la casella della lista “Nuova lista” e premo Fatto | Rimuove il film dalla lista “Nuova lista” e stampa il messaggio “Cambiamenti salvati” | Positivo |
| Dentro il contenuto della lista “Nuova lista” apro il menu opzioni film e seleziono “Rimuovi dalla lista” | Rimuove il film dalla lista “Nuova lista” e stampa il messaggio “Elemento cancellato dalla lista” | Positivo |



4.1.8 Rimozione amicizia

| Test ID | AND_TEST08 | |
|--|---|-----------|
| Test Name | Test di rimozione amicizia | |
| Test Description | Obiettivo è di verificare il funzionamento corretto di rimozione amicizia | |
| Input | Risultato desiderato | Risultato |
| Clicco il pulsante “Rimuovi amicizia” sull’utente “carmine” nella lista | Rimuove l’amicizia con utente “carmine” e stampa il messaggio “Amicizia rimossa con successo” | Positivo |
| Clicco il pulsante “Rimuovi amicizia” sull’utente “carmine” nella schermata “Altro utente” | Rimuove l’amicizia con utente “carmine” e stampa il messaggio “Amicizia rimossa con successo” | Positivo |



4.1.9 Effettuare le ricerche di utenti

| Test ID | AND_TEST09 | |
|---|--|-----------|
| Test Name | Test per effettuare le ricerche di utenti | |
| Test Description | Obiettivo è di verificare il funzionamento corretto per effettuare le ricerche di utenti | |
| Input | Risultato desiderato | Risultato |
| Compilo il campo “Username” con testo “carmine” e premo il pulsante ricerca | Mostra la lista di utenti che contengono la parola “carmine” dentro username | Positivo |
| Lascio il campo “Username” vuoto e premo il pulsante ricerca | Mostra la lista contenente tutti gli utenti registrati al sistema | Positivo |



4.1.10 Inviare le richieste di amicizia

| Test ID | AND_TEST10 | |
|---|--|-----------|
| Test Name | Test per inviare le richieste di amicizia | |
| Test Description | Obiettivo è di verificare il funzionamento corretto per inviare le richieste di amicizia | |
| Input | Risultato desiderato | Risultato |
| Clicco il pulsante “Inviare richiesta di amicizia” sull’utente “carmine” nella lista | Invia richiesta di amicizia all’utente “carmine” e stampa il messaggio “Amicizia inviata con successo” | Positivo |
| Clicco il pulsante “Inviare richiesta di amicizia” sull’utente “carmine” nella schermata “Altro utente” | Invia richiesta di amicizia all’utente “carmine” e stampa il messaggio “Amicizia inviata con successo” | Positivo |



4.1.12 Rifiutare la richiesta di amicizia

| Test ID | AND_TEST12 | |
|--|--|-----------|
| Test Name | Test per Rifiutare la richiesta di amicizia | |
| Test Description | Obiettivo è di verificare il funzionamento corretto per rifiutare la richiesta di amicizia | |
| Input | Risultato desiderato | Risultato |
| Clicco il pulsante “Rifiuta la richiesta di amicizia” sull’utente “carmine” nella lista | Invia richiesta di amicizia all’utente “carmine” e stampa il messaggio “Richiesta amicizia rifiutata con successo” | Positivo |
| Clicco il pulsante “Rifiuta la richiesta di amicizia” sull’utente “carmine” nella schermata “Altro utente” | Invia richiesta di amicizia all’utente “carmine” e stampa il messaggio “Richiesta amicizia rifiutata con successo” | Positivo |



4.1.13 Visualizzazione della scheda di un film

| Test ID | AND_TEST13 | |
|---|--|-----------|
| Test Name | Test per la visualizzazione della scheda di un film | |
| Test Description | Obiettivo è di verificare il funzionamento corretto per la visualizzazione della scheda di un film | |
| Input | Risultato desiderato | Risultato |
| Premo sul film "Venom" dentro una lista | Mostra schermata "Scheda film" con il poster e tutte le informazioni inerenti al film | Positivo |
| Premo sul film "Venom" nella schermata ricerca film | Mostra schermata "Scheda film" con il poster e tutte le informazioni inerenti al film | Positivo |



4.1.14 Visualizzazione della scheda utente

| Test ID | AND_TEST14 | |
|---|---|-----------|
| Test Name | Test per la visualizzazione della scheda utente | |
| Test Description | Obiettivo è di verificare il funzionamento corretto per la visualizzazione della scheda utente | |
| Input | Risultato desiderato | Risultato |
| Premo sul utente "carmine" dentro una lista | Mostra la schermata "Altro Utente" contenente, immagine profilo, username, le liste dell'utente e liste film in comune. | Positivo |
| Premo sul utente "carmine" nella schermata ricerca utenti | Mostra la schermata "Altro Utente" contenente: immagine profilo, username, le liste dell'utente e liste film in comune. | Positivo |
| Premo sul utente "alberto" nella schermata ricerca utenti | Mostra la schermata "Altro Utente" contenente: immagine profilo, username, le liste dell'utente. | Positivo |



4.1.15 Visualizzazione di una lista

| Test ID | AND_TEST15 | |
|--|---|-----------|
| Test Name | Test per la visualizzazione di una lista | |
| Test Description | Obiettivo è di verificare il funzionamento corretto per la visualizzazione di una lista | |
| Input | Risultato desiderato | Risultato |
| Dentro elenco liste premo sulla lista “Nuova lista 2” | Mostra la schermata “Contenuta lista” con il nome lista, descrizione lista e la lista di film contenuti | Positivo |
| Nella schermata “home” dentro la lista Rotante selezione la lista “Nuova lista 2” | Mostra la schermata “Contenuta lista” con il nome lista, descrizione lista e la lista di film contenuti | Positivo |
| Nella schermata “Altro utente” dentro la lista Rotante selezione la lista “Film da guardare” | Mostra la schermata “Contenuta lista” con il nome lista, descrizione lista e la lista di film contenuti | Positivo |



4.1.16 Visualizzazione dei film in comune con amico

| Test ID | AND_TEST16 | |
|---|---|-----------|
| Test Name | Test per la visualizzazione dei film in comune con amico | |
| Test Description | Obiettivo è di verificare il funzionamento corretto per la visualizzazione dei film in comune con amico | |
| Input | Risultato desiderato | Risultato |
| Nella schermata “Altro utente” dell’utente amico “carmine” premo sul “Film in comune” | Mostra la schermata “Lista Film in comune” contenente elenco di film che gli utenti hanno in comune | Positivo |



4.1.17 Visualizzazione delle notifiche

| Test ID | AND_TEST17 | |
|---|--|-----------|
| Test Name | Test per la visualizzazione delle notifiche | |
| Test Description | Obiettivo è di verificare il funzionamento corretto per la visualizzazione delle notifiche | |
| Input | Risultato desiderato | Risultato |
| Premo sul tasto “Notifiche” nella barra superiore dell’applicazione | Mostra la schermata “Notifiche” contenente tutte le notifiche ricevute | Positivo |
| Nella barra delle notifiche, premo sulla notifica ricevuta dall’applicazione con il messaggio “Utente carmine ti ha inviato la richiesta di amicizia” | Mostra la schermata “Notifiche” contenente tutte le notifiche ricevute | Positivo |



4.1.18 Effettuare una segnalazione

| Test ID | AND_TEST18 | |
|---|--|-----------|
| Test Name | Test di per effettuare una segnalazione | |
| Test Description | Obiettivo è di verificare il funzionamento corretto per effettuare una segnalazione | |
| Input | Risultato desiderato | Risultato |
| Premo pulsante segnalazione nella lista “Film da guardare”, scelgo il motivo “Contenuti offensivi od oltraggiosi” | Viene creata la segnalazione sulla lista e stampato il messaggio “Segnalazione effettuata con successo” | Positivo |
| Premo pulsante segnalazione nella lista “Film da guardare”, scelgo il motivo “Spoiler” | Non viene creata la segnalazione sulla lista ma stampa il messaggio “Segnalazione effettuata con successo” | Positivo |
| Premo pulsante segnalazione nella lista “Film da guardare”, scelgo il motivo “Contenuti violenti o ripugnanti” | Non viene creata la segnalazione sulla lista ma stampa il messaggio “Segnalazione effettuata con successo” | Positivo |
| Premo pulsante segnalazione nella lista “preferiti”, scelgo il motivo “Spoiler” | Viene creata la segnalazione sulla lista e stampato il messaggio “Segnalazione effettuata con successo” | Positivo |
| Premo pulsante segnalazione nella lista “nuovi”, scelgo il motivo “Contenuti violenti o ripugnanti” | Viene creata la segnalazione sulla lista e stampato il messaggio “Segnalazione effettuata con successo” | Positivo |



4.1.19 Cambiare la password

| Test ID | AND_TEST19 | |
|---|---|-----------|
| Test Name | Test di per cambiare la password | |
| Test Description | Obiettivo è di verificare il funzionamento corretto per cambiare la password | |
| Input | Risultato desiderato | Risultato |
| Compilo il campo vecchia password con " qawsedrf ", il campo nuova password con "password1234 " e premo "Salva" | Viene cambiata la password e mostrato il messaggio " Password cambiata con successo " | Positivo |
| Compilo il campo vecchia password con "password1234 ", il campo nuova password con "pass " e premo "Salva" | Mostra il messaggio " Password non rispetta i requisiti " | Positivo |
| Lascio i campi vuoti e premo "Salva" | Mostra il messaggio " Password non rispetta i requisiti " | Positivo |
| Compilo il campo vecchia password in modo errato, il campo nuova password con "qawsedrf1234 " e premo "Salva" | Mostra il messaggio " Password non rispetta i requisiti " | Positivo |



4.1.20 Cambiare immagine di profilo

| Test ID | AND_TEST20 | |
|---|--|-----------|
| Test Name | Test di per cambiare immagine di profilo | |
| Test Description | Obiettivo è di verificare il funzionamento corretto per cambiare immagine di profilo | |
| Input | Risultato desiderato | Risultato |
| Premo cambia immagine profilo | Viene proposto dal sistema applicazione per ricercare immagine | Positivo |
| Selezione applicazione Galleria e scelgo una nuova immagine | Cambia immagine e mostra il messaggio “ Immagine cambiata ” | Positivo |



4.1.21 Uscire dal profilo

| Test ID | AND_TEST21 | |
|---|--|-----------|
| Test Name | Test di per uscire dal profilo | |
| Test Description | Obiettivo è di verificare il funzionamento corretto per uscire dal profilo | |
| Input | Risultato desiderato | Risultato |
| Loggato con provider Google premo Logout | Viene fatto logout dal account e mostra la schermata Login | Positivo |
| Loggato con provider Facebook premo Logout | Viene fatto logout dal account e mostra la schermata Login | Positivo |
| Loggato con Cognito premo Logout | Viene fatto logout dal account e mostra la schermata Login | Positivo |



4.1.22 Modificare la descrizione di una lista

| Test ID | AND_TEST21 | |
|---|--|-----------|
| Test Name | Test di per modificare la descrizione di una lista | |
| Test Description | Obiettivo è di verificare il funzionamento corretto per modificare la descrizione di una lista | |
| Input | Risultato desiderato | Risultato |
| Premo il pulsante “ modifica ” | Viene visualizzata interfaccia per il cambiamento della lista | Positivo |
| Cambio la descrizione della lista “Nuova lista” da (vuoto) a “ Nuova descrizione ” e premo tasto “Conferma” | Viene cambiata la descrizione della lista e mostrato il messaggio “ Descrizione cambiata ” | Positivo |
| Provo cambiare la descrizione inserendo più di 200 caratteri | Il sistema blocca ogni successivo inserimento oltre 200 caratteri permessi | Positivo |



4.1.23 Visualizzazione del proprio profilo

| Test ID | AND_TEST21 | |
|-------------------------------------|---|-----------|
| Test Name | Test di per visualizzare il proprio profilo | |
| Test Description | Obiettivo è di verificare il funzionamento corretto per visualizzare il proprio profilo | |
| Input | Risultato desiderato | Risultato |
| Premo sul profilo nel menu laterale | Viene mostrato il Profilo del utente con personale immagine profilo, username e e-mail utilizzata | Positivo |



4.2 Test plan per System Testing applicazione Desktop

4.2.1 Accesso al sistema

| Test ID | DESK_TEST1 | |
|---|---|-----------|
| Test Name | Test di per verificare accesso al sistema | |
| Test Description | Obiettivo è di verificare il funzionamento corretto per verificare accesso al sistema | |
| Input | Risultato desiderato | Risultato |
| Compilo campo Codice speciale con “qaw” e campo Password con “qweras” | Mostra il messaggio di errore “Codice identificativo o Password non sono corretti” | Positivo |
| Compilo campo Codice speciale con “testprova”e campo Password con “qawsedrf” | Mostra il messaggio di errore “Codice identificativo o Password non sono corretti” | Positivo |
| Compilo campo Codice speciale con “adminmytro”e campo Password con “qawsedrf” | Effettua login e mostra la schermata home | Positivo |



4.2.2 Visualizzare una segnalazione

| Test ID | DESK_TEST2 | |
|---|---|-----------|
| Test Name | Test di per visualizzare la segnalazione | |
| Test Description | Obiettivo è di verificare il funzionamento corretto per verificare accesso al sistema | |
| Input | Risultato desiderato | Risultato |
| Nella schermata home seleziona la segnalazione con codice identificativo "31669945" | Mostra la schermata con informazione dettagliate sulla segnalazione | Positivo |



4.2.3 Gestire una segnalazione

| Test ID | DESK_TEST3 | |
|---|--|-----------|
| Test Name | Test di per gestire una segnalazione | |
| Test Description | Obiettivo è di verificare il funzionamento corretto per gestire una segnalazione | |
| Input | Risultato desiderato | Risultato |
| Visualizzando la segnalazione con codice identificativo “31669945” seleziono Oscura il contenuto e premo Conforma | Viene oscurata la lista segnalata, la segnalazione viene eliminata e verrà inviata una notifica sia all’utente che l’ha mandata sia al proprietario della lista | Positivo |
| Visualizzando la segnalazione con codice identificativo “1872324244” seleziono Cancellla contenuto e premo Conforma | Viene eliminata la lista segnalata, la segnalazione viene eliminata e verrà inviata una notifica sia all’utente che l’ha mandata sia al proprietario della lista | Positivo |
| Visualizzando la segnalazione con codice identificativo “233345611” seleziono Respingi e premo Conforma | Viene eliminata la segnalazione e utente che l’ha inviata verrà informato tramite una notifica | Positivo |



4.2.4 Uscire dal sistema

| Test ID | DESK_TEST4 | |
|---|--|-----------|
| Test Name | Test di per uscire dal sistema | |
| Test Description | Obiettivo è di verificare il funzionamento corretto per uscire dal sistema | |
| Input | Risultato desiderato | Risultato |
| Amministratore "admidmytro" nella schermata preme Logout. | Viene fatto logout dell'amministratore e visualizzata di nuovo la schermata di login | Positivo |



4.2.5 Visualizzare il regolamento

| Test ID | DESK_TEST5 | |
|---|---|-----------|
| Test Name | Test di per visualizzare il regolamento | |
| Test Description | Obiettivo è di verificare il funzionamento corretto per visualizzare il regolamento | |
| Input | Risultato desiderato | Risultato |
| Premo consulta il regolamento nella schermata home | Viene caricato e visualizzato il regolamento | Positivo |



4.3 Codice xUnit per unit testing

4.3.1 Unit testing per il metodo *calculateSecretHash*

Il metodo *calculateSecretHash* è usato per generare un codice (*String*) univoco per la richiesta di connessione al pool di utenti di Cognito.

Il metodo è usato internamente all'applicazione Desktop.

Il metodo ritorna:

- *Stringa SecretHash* che AWS richiede per inviare la richiesta di login tramite Cognito.

Il metodo richiede tre parametri:

- *userPoolClientId* id dell'applicazione client
- *userPoolClientSecret* la chiave secreta dell'applicazione client
- *userName* identificativo dell'utente che cerca loggare

Implementazione del metodo:

```

1  /**
2   * @param userPoolClientId id del client app
3   * @param userPoolClientSecret il codice secreto dell'app client
4   * @param userName userName usato per l'utente che cerca loggare
5   * @return SecretHash che AWS richiede per inviare la richiesta di login tramite Cognito
6   * @throws IllegalArgumentException nel caso in cui almeno uno dei parametri e' null, userPool e' vuoto o ClientSecret e' vuoto
7   */
8
9  public static String calculateSecretHash(@NotNull String userPoolClientId, @NotNull String userPoolClientSecret, @NotNull String userName) throws IllegalArgumentException {
10    final String HMAC_SHA256_ALGORITHM = "HmacSHA256";
11    if( userPoolClientId.equals("") || userPoolClientSecret.equals("") ){
12      throw new IllegalArgumentException("Empty key");
13    }
14    SecretKeySpec signingKey = new SecretKeySpec(
15      userPoolClientSecret.getBytes(StandardCharsets.UTF_8),
16      HMAC_SHA256_ALGORITHM);
17    try {
18      Mac mac = Mac.getInstance(HMAC_SHA256_ALGORITHM);
19      mac.init(signingKey);
20      mac.update(userName.getBytes(StandardCharsets.UTF_8));
21      byte[] rawMac = mac.doFinal(userPoolClientId.getBytes(StandardCharsets.UTF_8));
22      return Base64.getEncoder().encodeToString(rawMac);
23    } catch (Exception e) {
24      throw new RuntimeException("Error while calculating ");
25    }
26  }

```

4.3.1.1 – Strategia Black Box

Per la realizzazione della strategia Black Box sono state definite le seguenti classi di equivalenza:

| Nome CE | Parametro | Insieme dei valori | Risultato |
|---------|-----------------------------|--|------------|
| CE1 | <i>userPoolClientId</i> | Insieme degli elementi Stringa di caratteri | Valido |
| CE2 | <i>userPoolClientId</i> | Stringa vuota | Non valido |
| CE3 | <i>userPoolClientSecret</i> | Insieme degli elementi Stringa di caratteri | Valido |
| CE4 | <i>userPoolClientSecret</i> | Stringa vuota | Non valido |



| | | | |
|-----|-----------------|--|--------|
| CE5 | <i>userName</i> | Insieme degli elementi Stringa di caratteri | Valido |
| CE6 | <i>userName</i> | Stringa vuota | Valido |

Seguendo la strategia di SECT (Strong Equivalence Class Testing) abbiamo individuato seguenti test da effettuare:

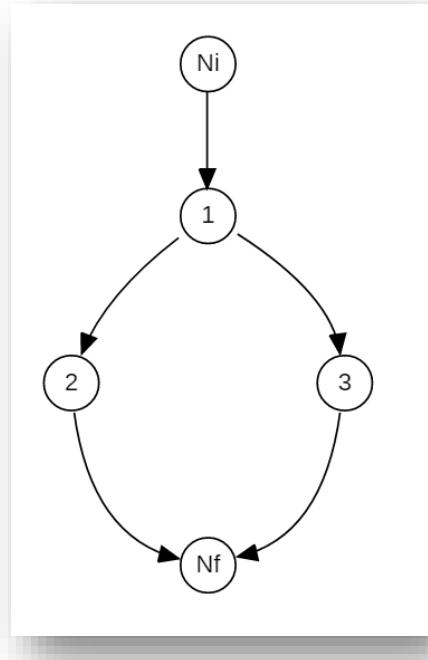
```

1  public class CognitoApiTest {
2      @Test
3      public void calculateSecretHash(){
4          String userPoolClientId = "dfsdfds";
5          String userPoolClientSecret = "dfsdfds";
6          String userName = "dfsdfds";
7          String risultato = CognitoApi.calculateSecretHash(userPoolClientId, userPoolClientSecret, userName);
8          Assert.assertEquals("KjyJGxmUCWq2q5ZEJw/04scUgaC3DcibYOpBMNpLXMK=", risultato);
9      }
10     @Test (expected = IllegalArgumentException.class)
11     public void calculateSecretHashCampiVuoti(){
12         String userPoolClientId = "";
13         String userPoolClientSecret = "";
14         String userName = "";
15         CognitoApi.calculateSecretHash(userPoolClientId, userPoolClientSecret, userName);
16     }
17     @Test
18     public void calculateSecretHashUserNameVuoto(){
19         String userPoolClientId = "dfsdfds";
20         String userPoolClientSecret = "dfsdfds";
21         String userName = "";
22         String risultato = CognitoApi.calculateSecretHash(userPoolClientId, userPoolClientSecret, userName);
23         Assert.assertEquals("i5H/zUHs/0c2vQx5oI0s3AkK9BfrNdUbg7j9WeQqLSA=", risultato);
24     }
25     @Test (expected = IllegalArgumentException.class)
26     public void calculateSecretHashConSecretVuoto(){
27         String userPoolClientId = "dfsdfds";
28         String userPoolClientSecret = "";
29         String userName = "dfsdfds";
30         CognitoApi.calculateSecretHash(userPoolClientId, userPoolClientSecret, userName);
31     }
32     @Test (expected = IllegalArgumentException.class)
33     public void calculateSecretHashConUserPoolVuoto(){
34         String userPoolClientId = "";
35         String userPoolClientSecret = "dfsdfds";
36         String userName = "dfsdfds";
37         CognitoApi.calculateSecretHash(userPoolClientId, userPoolClientSecret, userName);
38     }
39     @Test (expected = IllegalArgumentException.class)
40     public void calculateSecretHashConUserSecretVuotoConUserVuoto(){
41         String userPoolClientId = "dfsdfds";
42         String userPoolClientSecret = "";
43         String userName = "";
44         CognitoApi.calculateSecretHash(userPoolClientId, userPoolClientSecret, userName);
45     }
46     @Test (expected = IllegalArgumentException.class)
47     public void calculateSecretHashConClientVuoto(){
48         String userPoolClientId = "";
49         String userPoolClientSecret = "dfsdfds";
50         String userName = "dfsdfds";
51         CognitoApi.calculateSecretHash(userPoolClientId, userPoolClientSecret, userName);
52     }
53     @Test (expected = IllegalArgumentException.class)
54     public void calculateSecretHashConClientVuotoConUserSecretVuoto(){
55         String userPoolClientId = "";
56         String userPoolClientSecret = "";
57         String userName = "dfsdfds";
58         CognitoApi.calculateSecretHash(userPoolClientId, userPoolClientSecret, userName);
59     }
60 }
61 }
62 }
```



4.3.1.2 – Strategia White Box

La strategia White Box viene fornita con il grafo che rappresenta ogni possibile percorso di esecuzione della funzione:



Da grafo mostrato possiamo ricavare i due metodi i quali basteranno per ottenere *Branch Coverage*:

- calculateSecretHash_branch_1_2()
- calculateSecretHash_branch_1_3()

```

1  @Test (expected = IllegalArgumentException.class)
2  public void calculateSecretHash_branch_1_2(){
3      String userPoolClientId = "";
4      String userPoolClientSecret = "";
5      String userName = "dfsdfds";
6      CognitoApi.calculateSecretHash(userPoolClientId, userPoolClientSecret, userName);
7  }
8  @Test
9  public void calculateSecretHash_branch_1_3(){
10     String userPoolClientId = "dfsdfds";
11     String userPoolClientSecret = "dfsdfds";
12     String userName = "dfsdfds";
13     String risultato = CognitoApi.calculateSecretHash(userPoolClientId, userPoolClientSecret, userName);
14     Assert.assertEquals("KjyJGxmUCWq2q5ZEJw/04scUgaC3DcibYOpBMNpLXMK=", risultato);
15
16 }
```



4.3.2 Unite testing per il metodo checkCampiValidi

Il metodo checkCampiValidi è usato per effettuare un controllo sui dati inseriti per login sull'applicazione Desktop.

Il metodo ritorna:

- *Boolean* che sarà true nel caso in cui campi sono validi, false altrimenti

Il metodo richiede tre parametri:

- *username* la stringa identificativa dell'utente
- *password* la chiave secreta dell'utente

Implementazione del metodo:

```

1  public boolean checkCampiValidi(@NotNull String username, @NotNull String password) throws IllegalArgumentException{
2      return username.length() >= 4 && username.length() <= 32 && password.length() >= 8 && password.length() <= 32;
3  };
4

```

4.3.2.1 – Strategia Black Box

Per la realizzazione della strategia Black Box sono state definite le seguenti classi di equivalenza:

| Nome CE | Parametro | Insieme dei valori | Risultato |
|---------|-----------------|--|------------|
| CE1 | <i>username</i> | Insieme degli elementi Stringa di caratteri | Valido |
| CE2 | <i>username</i> | Stringa vuota | Non validi |
| CE3 | <i>password</i> | Insieme degli elementi Stringa di caratteri | Valido |
| CE4 | <i>password</i> | Stringa vuota | Non validi |



Seguendo la strategia di Boundary testing abbiamo individuato seguenti test da effettuare:

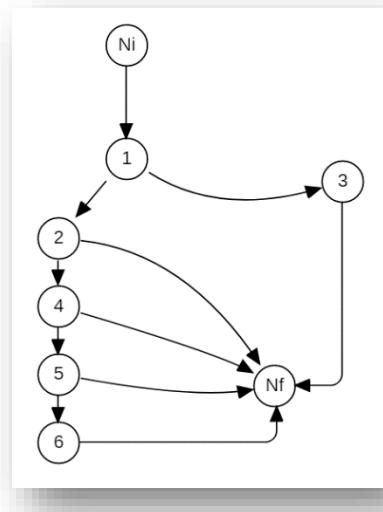
```

1  @Test
2  public void checkCampiValidiLunghezzaMinima() {
3      LoginViewModel viewModel = new LoginViewModel();
4      String username = "abcd";
5      String password = "qawsedrf";
6      Assert.assertTrue(viewModel.checkCampiValidi(username, password));
7  }
8  @Test
9  public void checkCampiValidiLunghezzaMinimaPlus() {
10     LoginViewModel viewModel = new LoginViewModel();
11     String username = "abcdx";
12     String password = "qawsedrfx";
13     Assert.assertTrue(viewModel.checkCampiValidi(username, password));
14 }
15 @Test
16 public void checkCampiValidiLunghezzaMassima() {
17     LoginViewModel viewModel = new LoginViewModel();
18     String username = "abcdabcdabcdabcdabcdabcdabcd";
19     String password = "abcdabcdabcdabcdabcdabcdabcd";
20     Assert.assertTrue(viewModel.checkCampiValidi(username, password));
21 }
22 @Test
23 public void checkCampiValidiLunghezzaMassimaMinus() {
24     LoginViewModel viewModel = new LoginViewModel();
25     String username = "abcdabcdabcdabcdabcdabcdabc";
26     String password = "abcdabcdabcdabcdabcdabcdabc";
27     Assert.assertTrue(viewModel.checkCampiValidi(username, password));
28 }
29 @Test
30 public void checkCampiValidiLunghezzaNormal() {
31     LoginViewModel viewModel = new LoginViewModel();
32     String username = "abcdabcdabc";
33     String password = "abcdabcdabcdabcdab";
34     Assert.assertTrue(viewModel.checkCampiValidi(username, password));
35 }
```



4.3.2.2 – Strategia White Box

La strategia White Box viene fornita con il grafo che rappresenta ogni possibile percorso di esecuzione della funzione:



Da grafo mostrato possiamo ricavare i cinque metodi i quali basteranno per ottenere *Branch Coverage*:

- checkCampiValidi_branch_1_3()
- checkCampiValidi_branch_1_2()
- checkCampiValidi_branch_1_2_4()
- checkCampiValidi_branch_1_2_4_5()
- checkCampiValidi_branch_1_2_4_5_6()

```

1  @Test
2  public void checkCampiValidi_branch_1_3() {
3      LoginViewModel viewModel = new LoginViewModel();
4      String username = "ab";
5      String password = "abcdabcdabcdabcdab";
6      Assert.assertFalse(viewModel.checkCampiValidi(username, password));
7  }
8  @Test
9  public void checkCampiValidi_branch_1_2() {
10     LoginViewModel viewModel = new LoginViewModel();
11     String username = "abcdabcdabcdabcdabcdabcdabcdX";
12     String password = "abcdabcdabcdabcdab";
13     Assert.assertFalse(viewModel.checkCampiValidi(username, password));
14 }
15 }
16
17 @Test
18 public void checkCampiValidi_branch_1_2_4() {
19     LoginViewModel viewModel = new LoginViewModel();
20     String username = "abcdabcdabc";
21     String password = "abcd";
22     Assert.assertFalse(viewModel.checkCampiValidi(username, password));
23 }
24
25 @Test
26 public void checkCampiValidi_branch_1_2_4_5() {
27     LoginViewModel viewModel = new LoginViewModel();
28     String username = "abcdabcdabc";
29     String password = "abcdabcdabcdabcdabcdabcdabcdX";
30     Assert.assertFalse(viewModel.checkCampiValidi(username, password));
31 }
32
33 @Test
34 public void checkCampiValidi_branch_1_2_4_5_6() {
35     LoginViewModel viewModel = new LoginViewModel();
36     String username = "abcdabcd";
37     String password = "abcdabcdabcdab";
38     Assert.assertTrue(viewModel.checkCampiValidi(username, password));
39 }
  
```



4.3.3 Unit testing per il metodo *ordinaLista*

Il metodo *ordinaLista* è un metodo utilizzato per riordinare le liste recuperate da più thread che lavorano in parallelo. I thread appena citati recuperano le liste in maniera casuale; quindi, il metodo le riordina seguendo la risposta in *JSONObject* ricevuta dall'esterno.

Il metodo è usato internamente nell'applicazione Android.

Il metodo ritorna:

- *List<Film>* una lista di film ordinata.

Il metodo richiede due parametri:

- *JSONArray* che contiene dei *JSONObject*, ogni *JSONObject* rappresenta un film tramite il campo “*Imdbid*”.
- *List<Film>* che rappresenta la lista non ordinata popolata dai thread sopra citati.

Implementazione del metodo:

```

1  public List<Film> ordinaLista(JSONArray jsonArray, List<Film> listaNonOrdinata) throws JSONException {
2      List<String> listaIdFilmOrdinati = JSONObjectManipulator.getListeIdFilmByJSONArray(jsonArray);
3      List<Film> listaFilmOrdinata = new ArrayList<>();
4
5      if(listaNonOrdinata != null && !listaNonOrdinata.isEmpty()){
6          if(listaNonOrdinata.contains(null))
7              throw new NullPointerException();
8
9          for(String id : listaIdFilmOrdinati){
10             Film film = listaNonOrdinata.stream().filter(o -> o.getIdFilm().equals(id)).findAny().orElse(null);
11             listaFilmOrdinata.add(film);
12         }
13     }
14
15     return listaFilmOrdinata;
16 }
17
18 }
```



4.3.3.1 – Strategia Black Box

Per la realizzazione della strategia Black Box sono state definite le seguenti classi di equivalenza:

| Nome CE | Parametro | Insieme dei valori | Risultato |
|---------|-------------------------|---|------------|
| A1 | <i>jsonArray</i> | Con JSONObject istanziati correttamente | Valido |
| A2 | <i>jsonArray</i> | Senza JSONObject all'interno (Vuoto) | Valido |
| A3 | <i>jsonArray</i> | Riferimento a NULL | Valido |
| A4 | <i>jsonArray</i> | Con JSONObject istanziati con campi sbagliati | Non valido |
| B1 | <i>listaNonOrdinata</i> | Con Film istanziati correttamente | Valido |
| B2 | <i>listaNonOrdinata</i> | Senza Film all'interno (Vuoto) | Valido |
| B3 | <i>listaNonOrdinata</i> | Riferimento a NULL | Valido |
| B4 | <i>listaNonOrdinata</i> | Con Film = NULL al suo interno | Non valido |

Seguendo la strategia di SECT (Strong Equivalence Class Testing) abbiamo individuato seguenti test da effettuare.

1. A1_B1
2. A1_B2
3. A1_B3
4. A1_B4
5. A2_B1
6. A2_B2
7. A2_B3
8. A2_B4
9. A3_B1
10. A3_B2
11. A3_B3
12. A3_B4
13. A4_B1
14. A4_B2
15. A4_B3
16. A4_B4



Configurazione iniziale

Nelle seguenti immagini è possibile visualizzare gli oggetti utilizzati durante il test.

```
public void configureJSONObject() throws JSONException {
    jsonObjectFilm1 = new JSONObject();
    jsonObjectFilm2 = new JSONObject();
    jsonObjectFilm3 = new JSONObject();
    jsonObjectFilm4 = new JSONObject();
    jsonObjectFilmWrongField = new JSONObject();

    jsonObjectFilm1.put("name: "imdbid", value: "100");
    jsonObjectFilm2.put("name: "imdbid", value: "200");
    jsonObjectFilm3.put("name: "imdbid", value: "300");
    jsonObjectFilm4.put("name: "imdbid", value: "400");
    jsonObjectFilmWrongField.put("name: "idWrong", value: "500");
}

public void configureFilm(){
    filmTest1 = new Film("NameTest", "100", "immagineCopertinaURL: "urlTest1");
    filmTest2 = new Film("NameTest2", "200", "immagineCopertinaURL: "urlTest2");
    filmTest3 = new Film("NameTest3", "300", "immagineCopertinaURL: "urlTest3");
    filmTest4 = new Film("NameTest4", "400", "immagineCopertinaURL: "urlTest4");
}
```

```
public void configureJsonArray(){
    jsonArrayTestClassic = new JSONArray();

    jsonArrayTestClassic.put(jsonObjectFilm1);
    jsonArrayTestClassic.put(jsonObjectFilm2);
    jsonArrayTestClassic.put(jsonObjectFilm3);
    jsonArrayTestClassic.put(jsonObjectFilm4);

    jsonArrayTestEmpty = new JSONArray();

    jsonArrayTestNull = null;

    jsonArrayTestWithWrongField = new JSONArray();
    jsonArrayTestWithWrongField.put(jsonObjectFilmWrongField);
}

public void configureListFilm(){
    listaFilmTestClassic= new ArrayList<>();
    listaFilmTestClassic.add(filmTest4);
    listaFilmTestClassic.add(filmTest3);
    listaFilmTestClassic.add(filmTest2);
    listaFilmTestClassic.add(filmTest1);

    listaFilmTestEmpty= new ArrayList<>();

    listaFilmTestNull=null;

    listaFilmTestWithNullReference=new ArrayList<>();
    listaFilmTestWithNullReference.add(filmTest1);
    listaFilmTestWithNullReference.add(null);
    listaFilmTestWithNullReference.add(filmTest2);
}
```

```
public void setupTest() throws JSONException {
    configureJSONObject();
    configureFilm();
    configureJsonArray();
    configureListFilm();
}
```



Test effettuati

```

@Test
public void ordinalistaA1_B1() throws JSONException {
    setupTest();

    List<Film> listaOttenuta = repository.ordinalista(jsonArrayTestClassic, listaFilmTestClassic);
    assertEquals( expected: "100", listaOttenuta.get(0).getIdFilm());
    assertEquals( expected: "200", listaOttenuta.get(1).getIdFilm());
    assertEquals( expected: "300", listaOttenuta.get(2).getIdFilm());
    assertEquals( expected: "400", listaOttenuta.get(3).getIdFilm());
}

@Test
public void ordinalistaA1_B2() throws JSONException {
    setupTest();
    List<Film> listaOttenuta = repository.ordinalista(jsonArrayTestClassic, listaFilmTestEmpty);
    assertTrue(listaOttenuta.isEmpty());
}

@Test
public void ordinalistaA1_B3() throws JSONException {
    setupTest();
    List<Film> listaOttenuta = repository.ordinalista(jsonArrayTestNull, listaFilmTestNull);
    assertTrue(listaOttenuta.isEmpty());
}

@Test (expected = NullPointerException.class)
public void ordinalistaA1_B4() throws JSONException {
    setupTest();
    repository.ordinalista(jsonArrayTestClassic, listaFilmTestWithNullReference);
}

```

```

@Test
public void ordinalistaA2_B1() throws JSONException {
    setupTest();
    List<Film> listaOttenuta = repository.ordinalista(jsonArrayTestEmpty, listaFilmTestClassic);
    assertTrue(listaOttenuta.isEmpty());
}

@Test
public void ordinalistaA2_B2() throws JSONException {
    setupTest();
    List<Film> listaOttenuta = repository.ordinalista(jsonArrayTestEmpty, listaFilmTestEmpty);
    assertTrue(listaOttenuta.isEmpty());
}

@Test
public void ordinalistaA2_B3() throws JSONException {
    setupTest();
    List<Film> listaOttenuta = repository.ordinalista(jsonArrayTestEmpty, listaFilmTestNull);
    assertTrue(listaOttenuta.isEmpty());
}

@Test(expected = NullPointerException.class)
public void ordinalistaA2_B4() throws JSONException {
    setupTest();
    repository.ordinalista(jsonArrayTestEmpty, listaFilmTestWithNullReference);
}

```



```

@Test
public void ordinalistaA3_B1() throws JSONException {
    setupTest();
    List<Film> listaOttenuta = repository.ordinalista(jsonArrayTestNull, listaFilmTestClassic);
    assertTrue(listaOttenuta.isEmpty());
}

@Test
public void ordinalistaA3_B2() throws JSONException {
    setupTest();
    List<Film> listaOttenuta = repository.ordinalista(jsonArrayTestNull, listaFilmTestEmpty);
    assertTrue(listaOttenuta.isEmpty());
}
@Test
public void ordinalistaA3_B3() throws JSONException {
    setupTest();
    List<Film> listaOttenuta = repository.ordinalista(jsonArrayTestNull, listaFilmTestNull);
    assertTrue(listaOttenuta.isEmpty());
}
@Test (expected = NullPointerException.class)
public void ordinalistaA3_B4() throws JSONException {
    setupTest();
    repository.ordinalista(jsonArrayTestNull, listaFilmTestWithNullReference );
}

```

```

@Test (expected = JSONException.class)
public void ordinalistaA4_B1() throws JSONException {
    setupTest();
    repository.ordinalista(jsonArrayTestWithWrongField,listaFilmTestClassic);
}

@Test (expected = JSONException.class)
public void ordinalistaA4_B2() throws JSONException {
    setupTest();
    repository.ordinalista(jsonArrayTestWithWrongField,listaFilmTestEmpty);
}

@Test (expected = JSONException.class)
public void ordinalistaA4_B3() throws JSONException {
    setupTest();
    repository.ordinalista(jsonArrayTestWithWrongField,listaFilmTestNull);
}

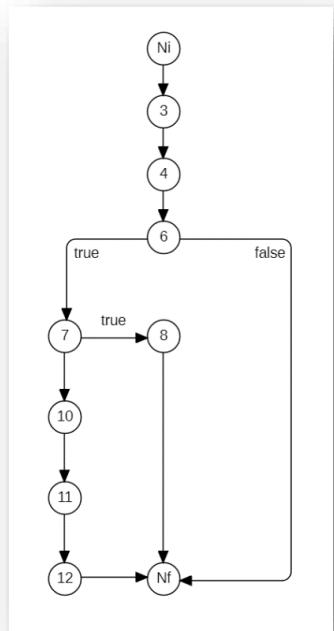
@Test (expected = JSONException.class)
public void ordinalistaA4_B4() throws JSONException {
    setupTest();
    repository.ordinalista(jsonArrayTestWithWrongField,listaFilmTestWithNullReference);
}

```



4.3.3.2 – Strategia White Box

La strategia White Box viene fornita con il grafo che rappresenta ogni possibile percorso di esecuzione della funzione:



Da grafo mostrato possiamo ricavare i 3 metodi i quali basteranno per ottenere *Branch Coverage*:

- `ordinaLista_branch_3_4_6 ()`
- `ordinaLista_branch_3_4_6_7_8 ()`
- `ordinaLista_branch_3_4_6_7_10_11_12 ()`

Test effettuati

```

@Test
public void ordinaLista_branch_3_4_6 () throws JSONException {
    setupTest();
    List<Film> listaOttenuta = repository.ordinaLista(jsonArrayTestClassic,listaFilmTestNull);
    Assert.assertTrue(listaOttenuta.isEmpty());
}

@Test (expected = NullPointerException.class)
public void ordinaLista_branch_3_4_6_7_8 () throws JSONException {
    setupTest();
    repository.ordinaLista(jsonArrayTestClassic,listaFilmTestWithNullReference);
}

@Test
public void ordinaLista_branch_3_4_6_7_10_11_12 ()throws JSONException {
    setupTest();
    List<Film> listaOttenuta = repository.ordinaLista(jsonArrayTestClassic,listaFilmTestClassic);
    assertEquals( expected: "100", listaOttenuta.get(0).getIdFilm());
    assertEquals( expected: "200", listaOttenuta.get(1).getIdFilm());
    assertEquals( expected: "300", listaOttenuta.get(2).getIdFilm());
    assertEquals( expected: "400", listaOttenuta.get(3).getIdFilm());
}
  
```



4.3.4 Unit testing per il metodo getUtenteDaJSONObjectConCurrentStatus

Il seguente metodo è utilizzato per trasformare un JSONObject in un utente. Il metodo legge i campi all'interno del JSONObject e costruisce l'oggetto di tipo Utente con i rispettivi valori
Il metodo è usato internamente nell'applicazione Android.

Il metodo ritorna:

- Un oggetto di tipo utente

Il metodo richiede due parametri:

- Un JSONObject con gli attributi di un utente

Implementazione del metodo:

```
1  public static Utente getUtenteDaJSONObjectConCurrentStatus(JSONObject jsonObject) throws JSONException {  
2      Utente utente = null;  
3  
4      if(jsonObject != null && jsonObject.length()>0){  
5          String username = jsonObject.getString("username");  
6          String email = jsonObject.getString("email");  
7          int CURRENT_STATE = jsonObject.getInt("CURRENT_STATE");  
8  
9          utente = new Utente(username,email, CURRENT_STATE);  
10     }  
11     return utente;  
12 }
```



4.3.4.1 – Strategia Black Box

Per la realizzazione della strategia Black Box sono state definite le seguenti classi di equivalenza:

| Nome CE | Parametro | Insieme dei valori | Risultato |
|---------|-------------------|--------------------------------------|------------|
| A1 | <i>jsonObject</i> | Istanziato correttamente | Valido |
| A2 | <i>jsonObject</i> | Vuoto | Valido |
| A3 | <i>jsonObject</i> | Riferimento a NULL | Valido |
| A4 | <i>jsonObject</i> | Istanziato con riferimenti sbagliati | Non valido |

Seguendo la strategia di SECT (Strong Equivalence Class Testing) abbiamo individuato seguenti test da effettuare.

- 1 A1
- 2 A2
- 3 A3
- 4 A4

Configurazione iniziale

Nel' immagine sotto riportata è possibile visualizzare gli oggetti utilizzati durante il test.

```
public void configureJSONObject() throws JSONException {
    jsonObjectClassic = new JSONObject();
    jsonObjectClassic.put("name: "username",usernameExpected);
    jsonObjectClassic.put("name: "email",emailExpected);
    jsonObjectClassic.put("name: "CURRENT_STATE",currentStatusExpected);

    jsonObjectEmpty = new JSONObject();

    jsonObjectNull = null;

    jsonObjectWithWrongField = new JSONObject();
    jsonObjectWithWrongField.put("name: "usernameleko",usernameExpected);
    jsonObjectWithWrongField.put("name: "emaillefefeh",emailExpected);
    jsonObjectWithWrongField.put("name: "oeoeoeoeoe",currentStatusExpected);

}
```



Test effettuati

```
@Test
public void getUtenteDaJSONObjectConCurrentStatusA1() throws JSONException {
    configureJSONObject();
    Utente utente = JSONObjectManipulator.getUtenteDaJSONObjectConCurrentStatus(jsonObjectClassic);

    Assert.assertEquals(usernameExpected,utente.getUsername());
    Assert.assertEquals(emailExpected,utente.getEmail());
    Assert.assertEquals(currentStatusExpected,utente.getCURRENT_STATE());
}

@Test
public void getUtenteDaJSONObjectConCurrentStatusA2() throws JSONException {
    configureJSONObject();
    Assert.assertNull(JSONObjectManipulator.getUtenteDaJSONObjectConCurrentStatus(jsonObjectEmpty));
}

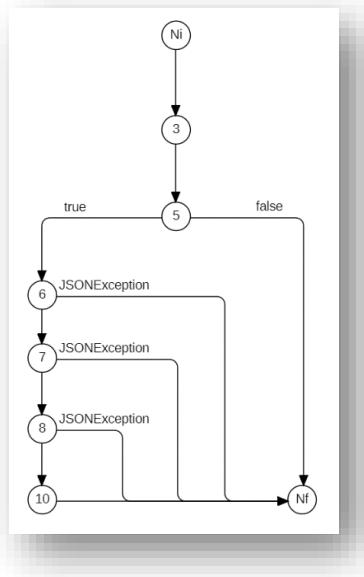
@Test
public void getUtenteDaJSONObjectConCurrentStatusA3() throws JSONException {
    configureJSONObject();
    Assert.assertNull(JSONObjectManipulator.getUtenteDaJSONObjectConCurrentStatus(jsonObjectNull));
}

@Test(expected = JSONException.class)
public void getUtenteDaJSONObjectConCurrentStatusA4() throws JSONException {
    configureJSONObject();
    Assert.assertNull(JSONObjectManipulator.getUtenteDaJSONObjectConCurrentStatus(jsonObjectWithWrongField));
}
```



4.3.4.2 – Strategia White Box

La strategia White Box viene fornita con il grafo che rappresenta ogni possibile percorso di esecuzione della funzione:



Da grafo mostrato possiamo ricavare 5 metodi i quali basteranno per ottenere *Branch Coverage*:

- getUtenteDaJSON_branch_3_5()
- getUtenteDaJSON_branch_3_5_6()
- getUtenteDaJSON_branch_3_5_6_7()
- getUtenteDaJSON_branch_3_5_6_7_8()
- getUtenteDaJSON_branch_3_5_6_7_10()

Configurazione iniziale

```

public void configureWrongJSONObject() throws JSONException {
    jsonObjectWithWrongFieldEmail = new JSONObject();
    jsonObjectWithWrongFieldEmail.put("name: "username", usernameExpected);
    jsonObjectWithWrongFieldEmail.put("name: "emailWRONG", emailExpected);
    jsonObjectWithWrongFieldEmail.put("name: "CURRENT_STATE", currentStatusExpected);

    jsonObjectWithWrongFieldCurrentStatus = new JSONObject();
    jsonObjectWithWrongFieldCurrentStatus.put("name: "username", usernameExpected);
    jsonObjectWithWrongFieldCurrentStatus.put("name: "email", emailExpected);
    jsonObjectWithWrongFieldCurrentStatus.put("name: "CURRENT_STATE_WRONG", currentStatusExpected);

    jsonObjectWithWrongFieldUsername = new JSONObject();
    jsonObjectWithWrongFieldUsername.put("name: "UsernameWrong", usernameExpected);
    jsonObjectWithWrongFieldUsername.put("name: "email", emailExpected);
    jsonObjectWithWrongFieldUsername.put("name: "CURRENT_STATE", currentStatusExpected);
}
  
```



Test effettuati

```
@Test
public void getUtenteDaJSON_branch_3_5() throws JSONException {
    configureJSONObject();
    Assert.assertNull(JSONObjectManipulator.getUtenteDaJSONObjectConCurrentStatus(jsonObjectNull));
}

@Test(expected = JSONException.class)
public void getUtenteDaJSON_branch_3_5_6() throws JSONException {
    configureWrongJSONObject();
    JSONObjectManipulator.getUtenteDaJSONObjectConCurrentStatus(jsonObjectWithWrongFieldUsername);
}

@Test(expected = JSONException.class)
public void getUtenteDaJSON_branch_3_5_6_7() throws JSONException {
    configureWrongJSONObject();
    JSONObjectManipulator.getUtenteDaJSONObjectConCurrentStatus(jsonObjectWithWrongFieldEmail);
}

@Test(expected = JSONException.class)
public void getUtenteDaJSON_branch_3_5_6_7_8() throws JSONException {
    configureWrongJSONObject();
    JSONObjectManipulator.getUtenteDaJSONObjectConCurrentStatus(jsonObjectWithWrongFieldCurrentStatus);
}

@Test
public void getUtenteDaJSON_branch_3_5_6_7_10() throws JSONException {
    configureJSONObject();
    Utente utente = JSONObjectManipulator.getUtenteDaJSONObjectConCurrentStatus(jsonObjectClassic);

    Assert.assertEquals(usernameExpected,utente.getUsername());
    Assert.assertEquals(emailExpected,utente.getEmail());
    Assert.assertEquals(currentStatusExpected,utente.getCURRENT_STATE());
}
```

