Analisi dei requisiti

Gruppo 02

December 15, 2024

# Introduzione

Il programma implementa una rubrica per gestire i contatti telefonici e/o e-mail. Il programma consente agli utenti registrati di inserire, eliminare o modificare dei contatti all’interno della rubrica. Permette di effettuare un login con email e password e a nuovi utenti di registrarsi.

# Requisiti

## Requisiti funzionali

• **Funzionalità individuali:**

**IF-1** Il sistema deve permettere agli utenti non registrati di registrarsi inserendo un’e-mail e una password.

**IF-2** Il sistema deve permettere agli utenti registrati di accedere tramite e-mail e password. Dopo l’accesso vengono visualizzati i contatti salvati in precedenza. **IF-3** Il sistema deve permettere agli utenti registrati di inserire un nuovo contatto.

**IF-4** Il sistema notifica con un messaggio se l’utente tenta di aggiungere un contatto duplicato. Due contatti sono uguali se hanno lo stesso nome e cognome. **IF-5** Il sistema deve permettere agli utenti registrati di eliminare un contatto selezionato.

**IF-6** Il sistema deve permettere agli utenti di modificare le informazioni di un contatto selezionato.

**IF-7** Il sistema deve permettere di ricercare un contatto tramite una sottostringa inserita in una barra di ricerca.

**IF-8** Il sistema consente di aggiungere tag ai contatti.

**IF-9** Il sistema consente di filtrare i contatti tramite la selezione di un tag. **IF-10** Il sistema deve permettere di ordinare le righe della tabella in base a nome o cognome e con un ordinamento che può essere crescente o decrescente. **IF-11** Il sistema deve permettere agli utenti registrati di importare i contatti da un file o esportare i contatti in un file.

* **Esigenze di Dati e informazioni:**

**DF-1** Dati del profilo utente:

* + e-mail
  + password

**DF-2** Dati dei contatti:

* + nome e cognome (almeno uno dei due campi deve essere non vuoto)
  + da zero a tre numeri di telefono
  + da zero a tre indirizzi e-mail
  + da zero a tre tag
* **Interfaccia utente:**

**UI-1** Pagina di login e registrazione dove inserire l’e-mail e la password per il login o l’e-mail e la password (e conferma password) per la registrazione. **UI-2** La rubrica viene visualizzata in forma tabellare. Ogni colonna della tabella corrisponde a un campo del generico contatto. Ogni riga della tabella corrisponde a un singolo contatto.

**UI-3** La selezione di un contatto avviene cliccando la corrispondente riga con il mouse.

**UI-4** Nella pagina principale della rubrica sono presenti diversi bottoni e menu per eseguire le operazioni.

**UI-5** Messaggi di errore: in caso di errore nelle informazioni inserite dall’utente l’interfaccia deve indicare quale informazione è errata ed eventuali vincoli non rispettati.

**UI-6** Se l’utente prova a inserire un contatto già esistente compare un messaggio di notifica.

• **Interfacce con sistemi esterni:**

**IS-1** Database per salvare le informazioni.

# Vincoli di progettazione

1. Il sistema deve essere implementato tramite un’applicazione Java compatibile con Maven.
2. La consegna del progetto è prevista per il 15/12/2024.

# Casi d’uso

## Attori

* Utente registrato: può effettuare il login; una volta effettuato il login, visualizza la pagina principale della rubrica contenente i contatti salvati.
* Utente: può registrarsi inserendo i dati richiesti.
* Database: gestisce il servizio di autenticazione e di memorizzazione dei contatti.

## Descrizione dei casi d’uso REGISTRAZIONE

**Attori partecipanti:** utente, database

**Precondizioni:**

* L’utente apre l’applicazione

**Postcondizioni:**

* L’utente ha un account e diventa un utente registrato

**Flusso di eventi:**

1. L’utente inserisce e-mail e password (e conferma password)
2. Verifica della correttezza dei dati inseriti
3. I dati vengono inseriti nel database
4. Registrazione completata

**Flusso di eventi alternativo:**

* 1. L’indirizzo e-mail è già presente nel database. L’utente deve effettuare il login.
  2. La password inserita non rispetta le condizioni di sicurezza consigliate; si richiede l’inserimento di una password differente.
  3. Le password non corrispondono.

**LOGIN**

**Attori partecipanti:** utente registrato, database

**Precondizioni:** • L’utente deve essere registrato

**Postcondizioni:**

• L’utente ha accesso alla homepage della rubrica

**Flusso di eventi:**

1. L’utente registrato inserisce e-mail e password.
2. Verifica delle credenziali (nel database).
3. Verifica andata a buon fine.

**Flusso di eventi alternativo:**

* 1. E-mail non presente nel database → Registrazione.
  2. Credenziali errate

**LOGOUT**

**Attori partecipanti:** utente registrato, database

**Precondizioni:** • L’utente ha effettuato il login

**Postcondizioni:**

• L’utente torna alla schermata iniziale

**Flusso di eventi:**

1. L’utente preme sul bottone per uscire dall’account.
2. Viene chiusa la connessione col database.

**OPERAZIONI RUBRICA**

**Attori partecipanti:** utente registrato

**Precondizioni:**

• L’utente ha eseguito il login

**Flusso di eventi:**

1. L’utente visualizza la pagina principale della rubrica e sceglie quale operazione effettuare

**Flusso di eventi alternativo:**

1.a L’utente seleziona un contatto e l’operazione da effettuare su di esso.

**AGGIUNGERE CONTATTO**

**Attori partecipanti:** utente registrato

**Precondizioni:**

* L’utente preme sul bottone per l’aggiunta di un contatto

**Postcondizioni:**

* Un contatto viene aggiunto alla rubrica

**Flusso di eventi:**

1. L’utente specifica le informazioni da associare al nuovo contatto.
2. Le modifiche vengono salvate.

**Flusso di eventi alternativo:**

* 1. I campi nome e cognome sono entrambi vuoti; il contatto non viene inserito in rubrica.
  2. Contatto duplicato.
  3. L’utente annulla l’operazione.

**RIMUOVERE CONTATTO**

**Attori partecipanti:** utente registrato

**Precondizioni:**

* Nella rubrica è presente almeno un contatto.
* L’utente indica che vuole rimuovere un contatto tramite l’apposito bottone.

**Postcondizioni:**

* Il contatto selezionato è eliminato dalla rubrica

**Flusso di eventi:**

1. Il contatto viene rimosso dalla rubrica.
2. Le modifiche vengono salvate.

**MODIFICARE INFORMAZIONI PER UN CONTATTO GIA ES-`**

**ISTENTE**

**Attori partecipanti:** utente registrato

**Precondizioni:**

* Nella rubrica è presente almeno un contatto
* L’utente indica di voler modificare un contatto tramite l’apposito bottone

**Postcondizioni:**

* Le informazioni del contatto selezionato sono modificate

**Flusso di eventi:**

1. L’utente modifica le informazioni del contatto selezionato.
2. Le modifiche vengono salvate.

**Flusso di eventi alternativo:**

* 1. Non è possibile salvare la modifica effettuata perché sono stati eliminati sia il nome che il cognome.
  2. Contatto duplicato.
  3. L’utente annulla l’operazione.

**RICERCA CONTATTI**

**Attori partecipanti:** utente registrato

**Precondizioni:**

* L’utente seleziona la barra di ricerca o un tag dal menu

**Postcondizioni:**

* L’utente visualizza solo i contatti corrispondenti ai criteri di ricerca.

**Flusso di eventi:**

1. L’utente inserisce la sottostringa da ricercare e/o seleziona il/i tag.

**IMPORTARE CONTATTI**

**Attori partecipanti:** utente registrato

**Precondizioni:**

* L’utente preme il tasto per importare i contatti.

**Postcondizioni:**

* I contatti da importare sono aggiunti alla rubrica.

**Flusso di eventi:**

1. L’utente seleziona il file contenente i contatti da importare.
2. I contatti vengono aggiunti.

**Flusso di eventi alternativo:**

* 1. Errore di importazione: messaggio d’errore.
  2. L’utente annulla l’operazione.

**ESPORTARE CONTATTI**

**Attori partecipanti:** utente registrato

**Precondizioni:**

* L’utente preme il tasto per esportare i contatti.

**Postcondizioni:**

* I contatti presenti in rubrica vengono esportati nel file indicato.

**Flusso di eventi:**

1. L’utente seleziona il file in cui esportare i contatti.
2. I contatti vengono esportati.

**Flusso di eventi alternativo:**

* 1. Errore di esportazione: messaggio d’errore.
  2. L’utente annulla l’operazione.

**ORDINARE CONTATTI**

**Attori partecipanti:** utente registrato

**Precondizioni:**

* L’utente indica un criterio di ordinamento cliccando su uno dei campi della tabella

**Postcondizioni:**

* I contatti della rubrica sono visualizzati secondo il criterio scelto indicato

**Flusso di eventi:**

1. L’ordinamento è stato effettuato correttamente.

**CONTATTO DUPLICATO**

**Attori partecipanti:** utente registrato

**Precondizioni:**

* L’utente sta aggiungendo/modificando un contatto con nome e cognome già associati a un altro contatto.

**Postcondizioni:**

* La rubrica viene aggiornata.

**Flusso di eventi:**

1. Compare un messaggio di scelta.
2. Viene aggiunto/modificato un contatto.

**Flusso di eventi alternativo:**

2.a.1 L’utente torna a modificare le informazioni del contatto.

2.a.2 L’utente salva le modifiche.

2.b L’utente annulla l’operazione.

**SALVATAGGIO**

**Attori partecipanti:** utente registrato, database

**Precondizioni:**

* L’utente indica di voler salvare una modifica.

**Postcondizioni:**

* Le modifiche vengono salvate.

**Flusso di eventi:**

1. La rubrica aggiornata viene salvata.

# Descrizione dei requisiti

|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | **Stato del requisito** | **Priorità** | **Rischio**  **Tecnico** |
| IF-1 | Accettato per questa release | Alta | Medio |
| IF-2 | Accettato per questa release | Alta | Basso |
| IF-3 | Accettato per questa release | Alta | Basso |
| IF-4 | Accettato per questa release | Bassa | Basso |
| IF-5 | Accettato per questa release | Alta | Basso |
| IF-6 | Accettato per questa release | Media | Basso |
| IF-7 | Accettato per questa release | Media | Basso |
| IF-8 | Accettato per questa release | Bassa | Basso |
| IF-9 | Accettato per questa release | Bassa | Basso |
| IF-10 | Accettato per questa release | Media | Basso |
| IF-11 | Accettato per questa release | Medio | Medio |
| DF-1 | Accettato per questa release | Alta | Basso |
| DF-2 | Accettato per questa release | Alta | Basso |
| UI-1 | Accettato per questa release | Alta | Basso |
| UI-2 | Accettato per questa release | Alta | Basso |
| UI-3 | Accettato per questa release | Alta | Basso |
| UI-4 | Accettato per questa release | Alta | Basso |
| UI-5 | Accettato per questa release | Media | Basso |
| UI-6 | Accettato per questa release | Bassa | Basso |
| IS-1 | Accettato per questa release | Alta | Alto |

*Tabella dei requisiti*

# Diagramma dei casi d’uso

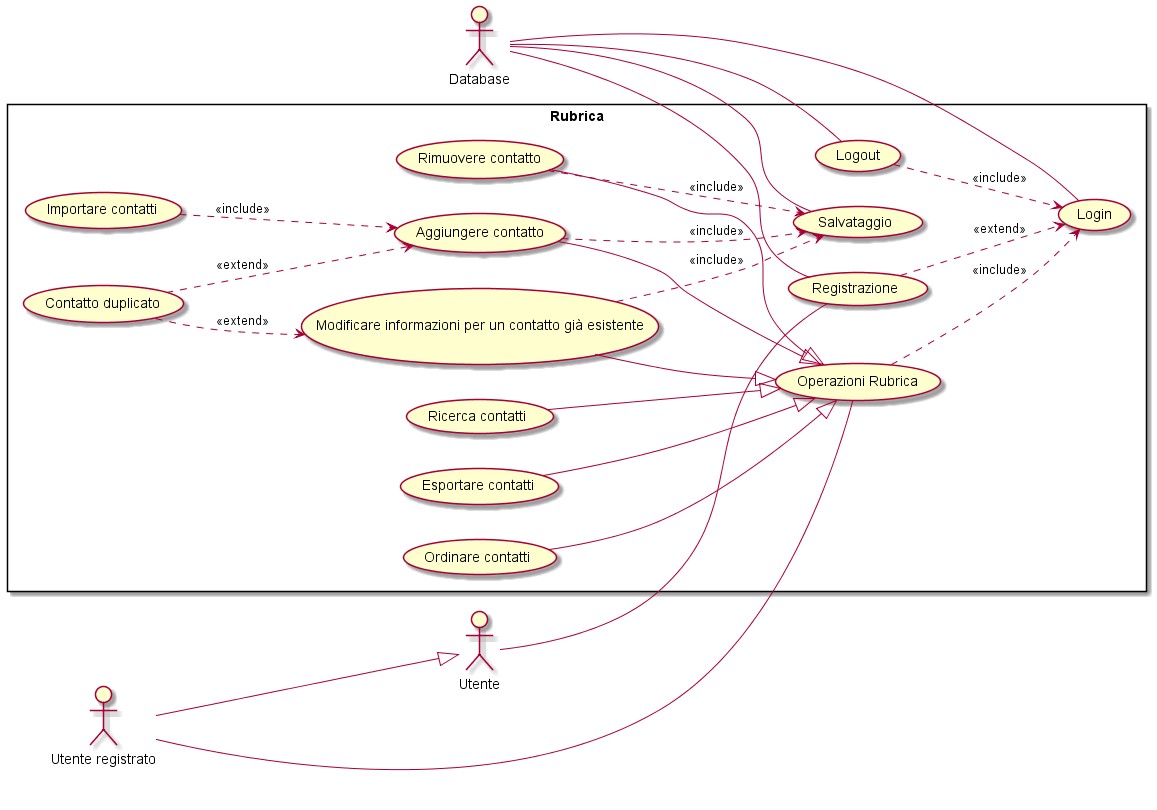


Figure 1: Diagramma UML

Design

# Attributi di qualità

## Esterni

* **Sicurezza**: Il sistema nell’inserimento dei dati sul database effettua una verifica che rende difficile l’attacco tramite session hijacking. Le password degli utenti salvate su database sono crittografate.
* **Scalabilità**: Il sistema è stato progettato per supportare l’espansione senza degradare significativamente le prestazioni; infatti, l’utilizzo del database permette la restituzione di più record in tempi ottimali. Nota: Attualmente, il prototipo è ospitato su un server con capacità limitate. Tuttavia, la progettazione del sistema prevede la possibilità di migrare a server ad alte prestazioni e distribuiti in futuro, garantendo così una gestione efficiente dell’aumento del carico di lavoro.)
* **Usabilità**: Il programma dispone di un’interfaccia grafica che consente agli utenti di accedere facilmente a tutte le funzionalità. I contatti possono essere classificati in tre categorie: job, university e home. Tramite un apposito menu`, è possibile visualizzare solo i contatti appartenenti ai tag selezionati. Inoltre, una barra di ricerca permette di filtrare i contatti, mostrando solo quelli che contengono la parola inserita in uno qualsiasi dei campi.
* **Disponibilità**: l’utente per accedere ai dati della rubrica può collegarsi alla rete Internet, avviare l’applicazione su qualsiasi PC, o laptop, effettuare l’accesso e consultare i propri contatti.

## Interni

* **Manutenibilità**: Il sistema organizza i dati in un database facilitando l’implementazione di future ed eventuali nuove funzioni (Ad esempio la creazione di una web app che permetterà agli utenti di collegarsi tramite browser). Il sistema è facilmente manutenibile grazie ai livelli di coesione, che sono quasi tutti funzionali ad eccezione di uno e ai livelli di accoppiamento, che sono alternativamente per dati o per comunicazione.
* **Portabilità**: L’applicazione è stata sviluppata in Java, garantendo la compatibilità con tutti i principali sistemi operativi (Windows, MacOS, Linux, ecc.). Grazie alla Java Virtual Machine (JVM), il codice può essere eseguito senza modifiche su qualsiasi piattaforma che la supporti.

# Diagrammi delle classi

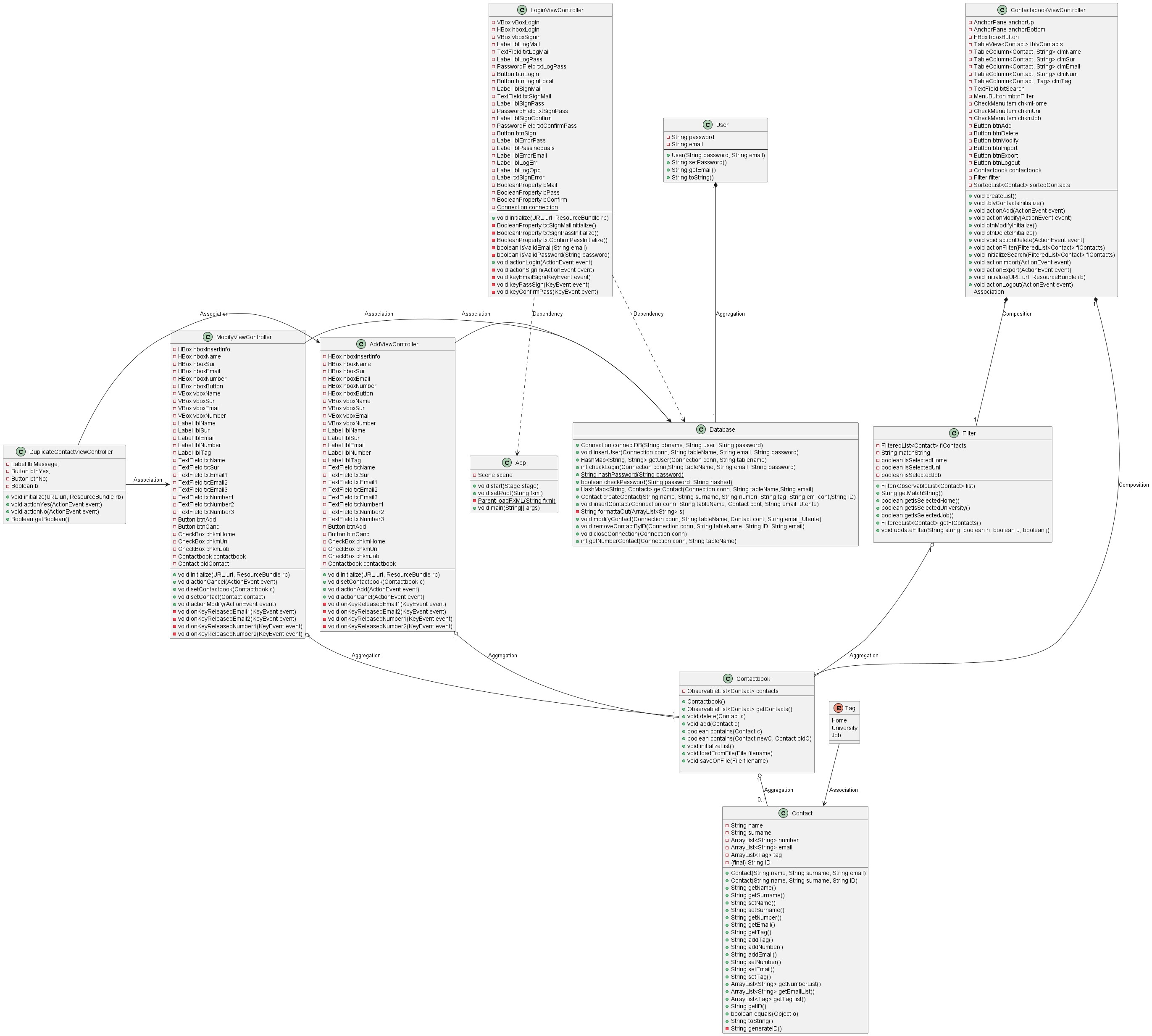


Figure 2: Diagramma delle classi

Il diagramma completo delle classi è molto complesso e contiene informazioni sulla struttura dell’interfaccia grafica (HBox, VBox, ...) che ne rendono più difficile la lettura, senza apportare informazioni rilevanti. Di seguito è quindi proposta una versione snellita del diagramma delle classi dove non sono presenti tali attributi.

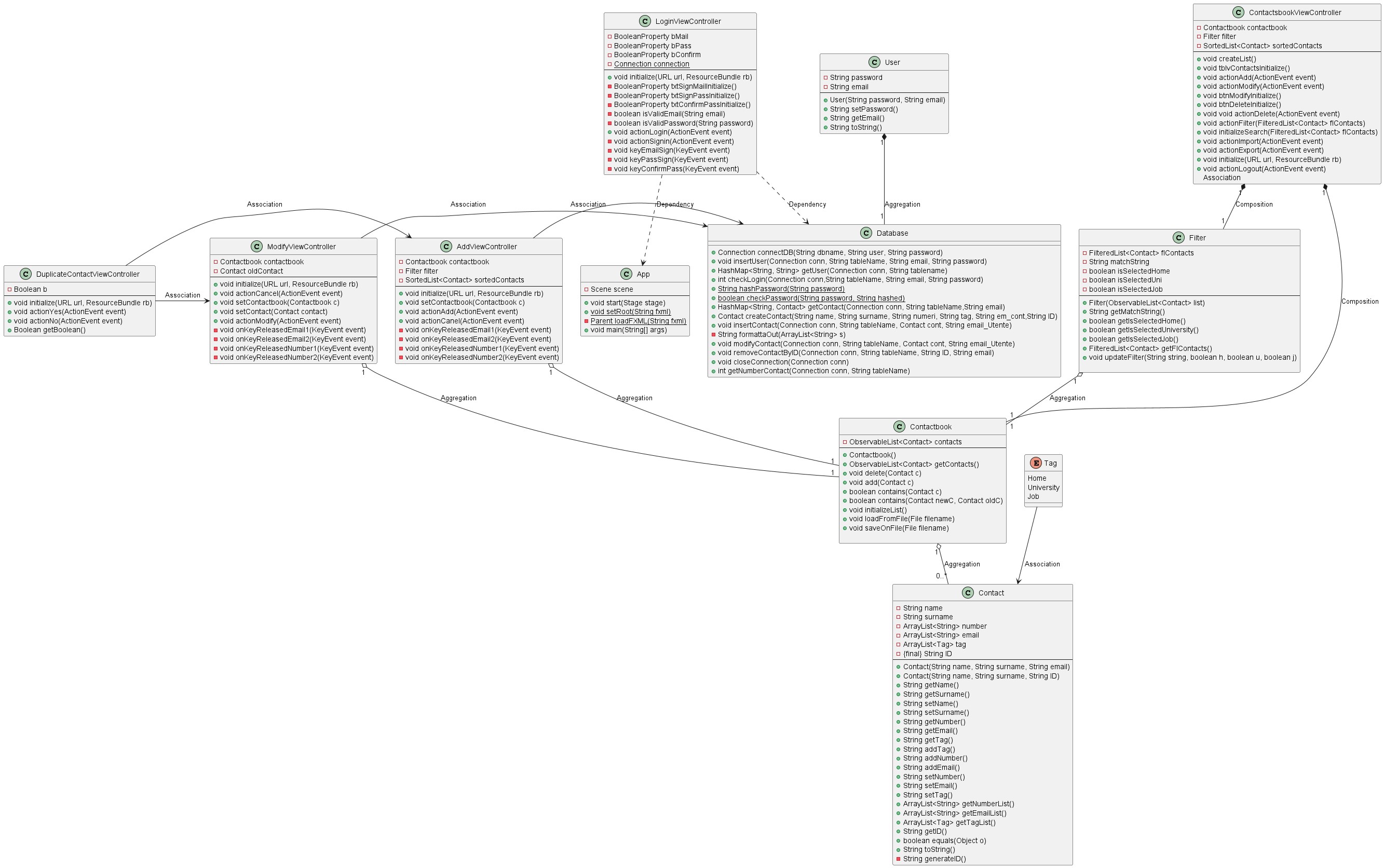


Figure 3: Diagramma delle classi in versione alternativa

# Diagrammi delle sequenze

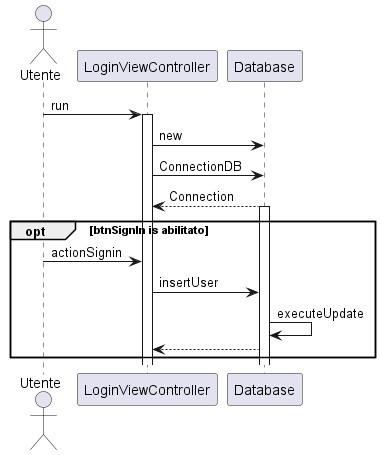


Figure 4: Diagramma delle sequenze relativo alla registrazione di un nuovo utente.

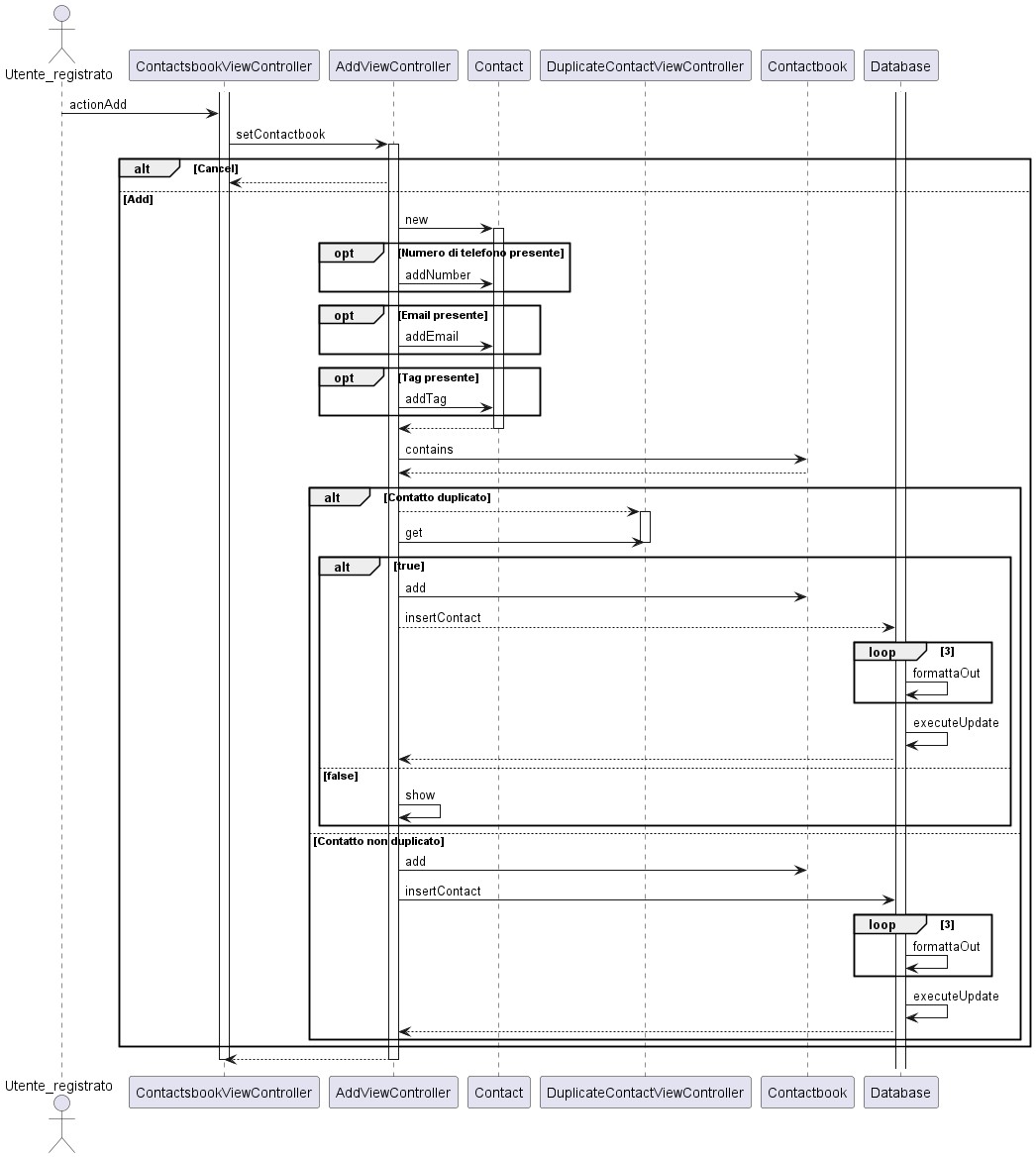


Figure 5: Diagramma delle sequenze relativo al login.

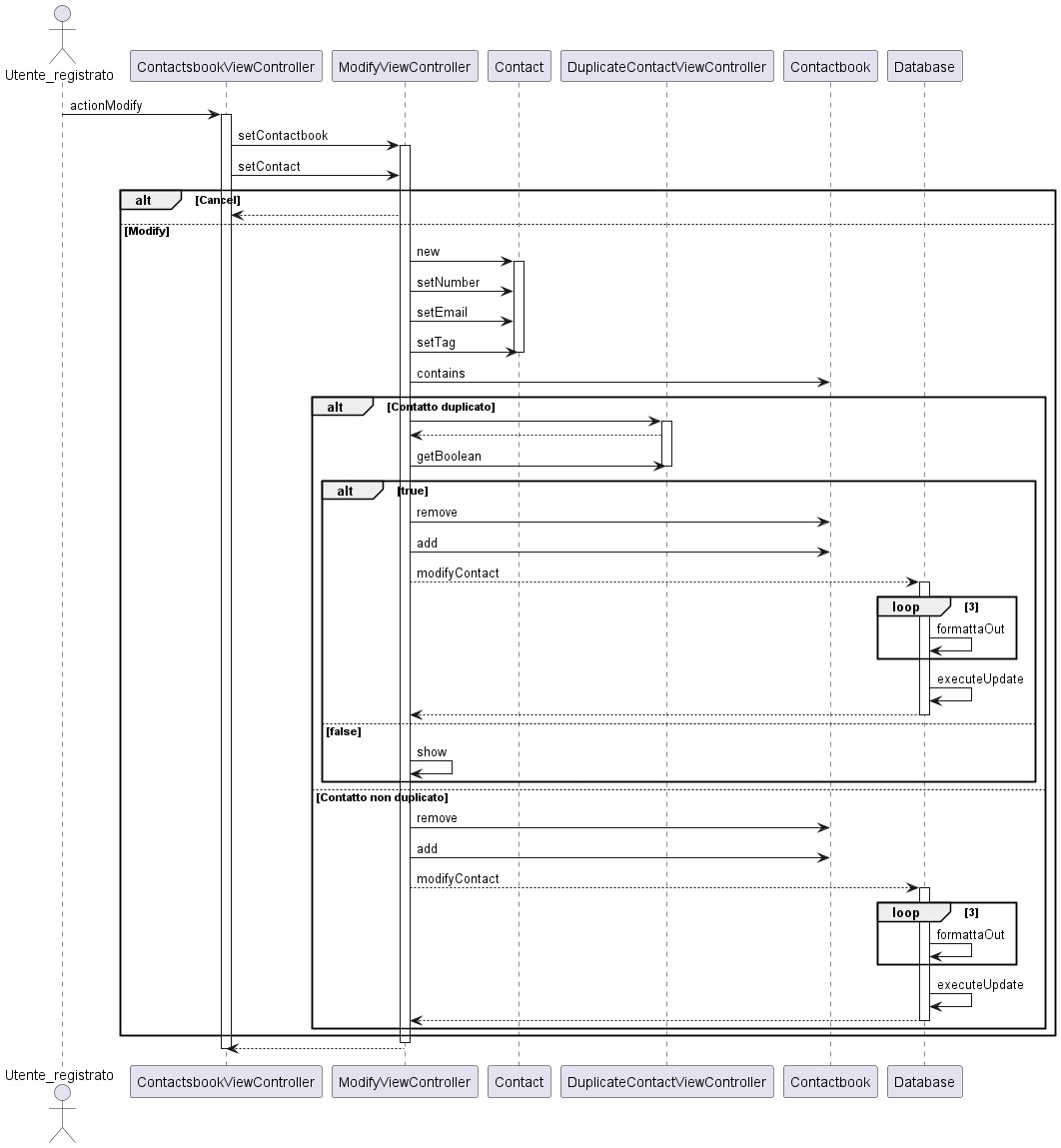


Figure 6: Diagramma delle sequenze relativo alla modifica di un contatto già esistente.

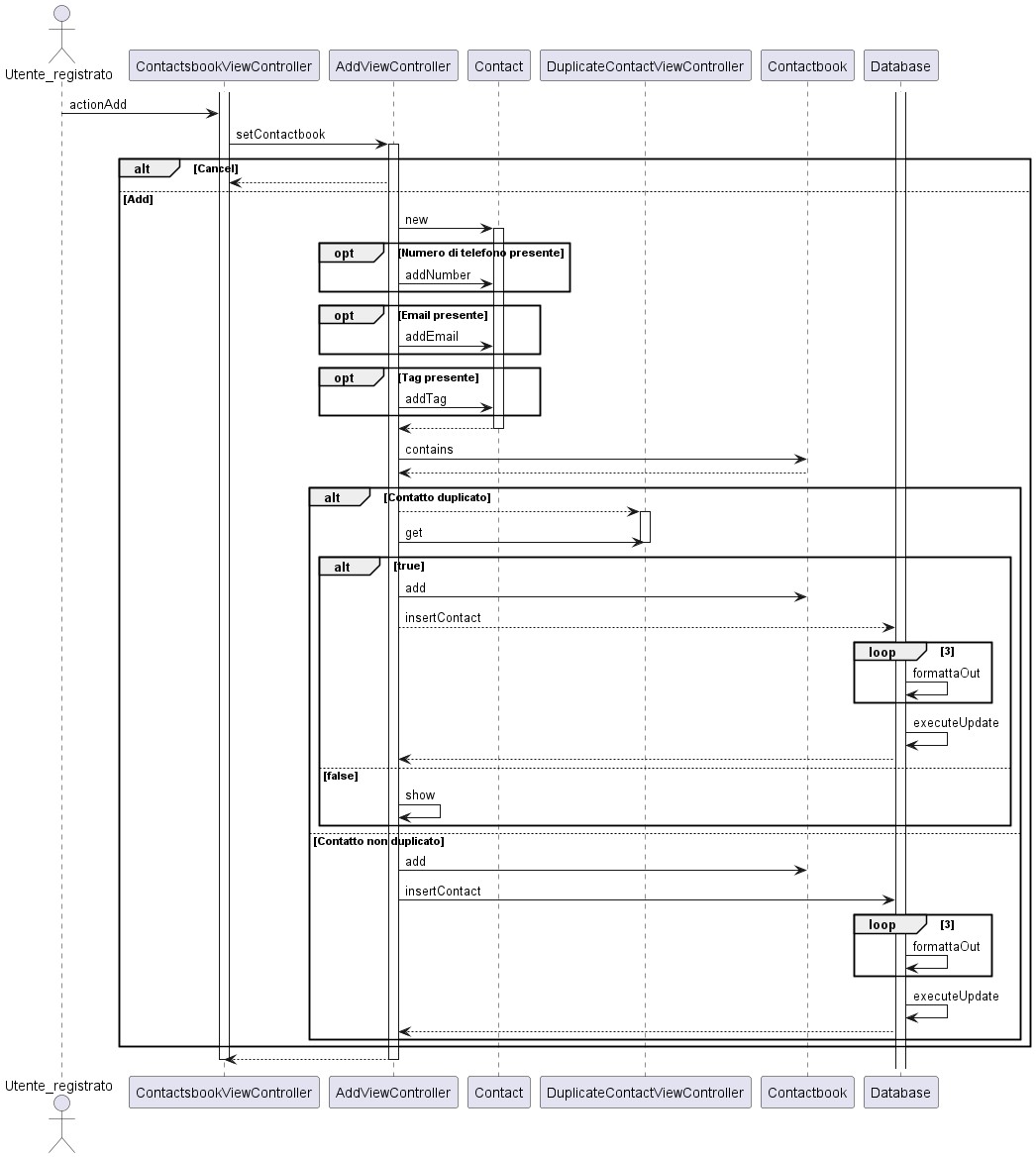


Figure 7: Diagramma delle sequenze relativo all’aggiunta di un nuovo contatto.

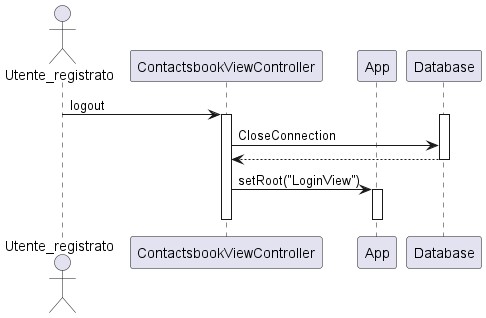
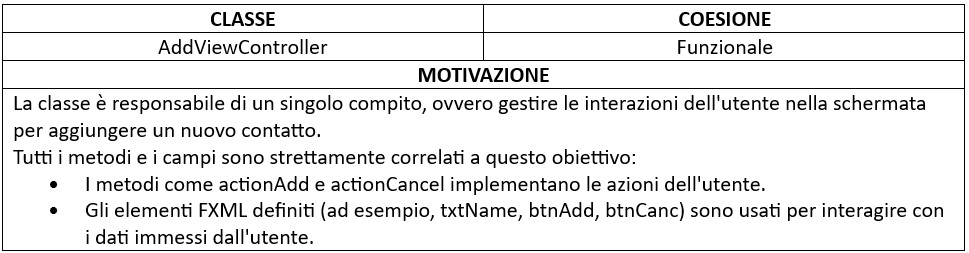
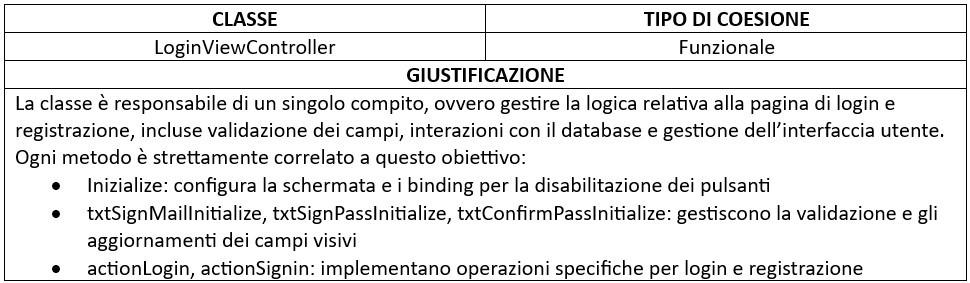


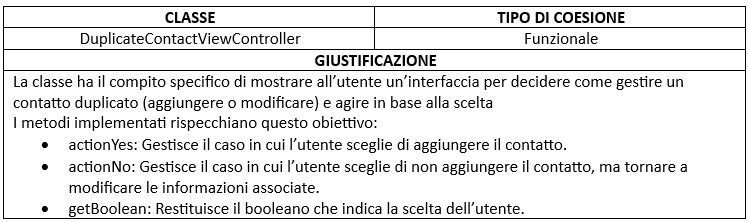
Figure 8: Diagramma delle sequenze relativo al logout.

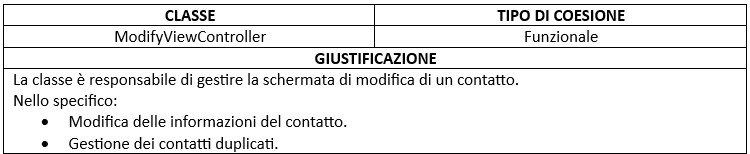
# Analisi di coesione e accoppiamento

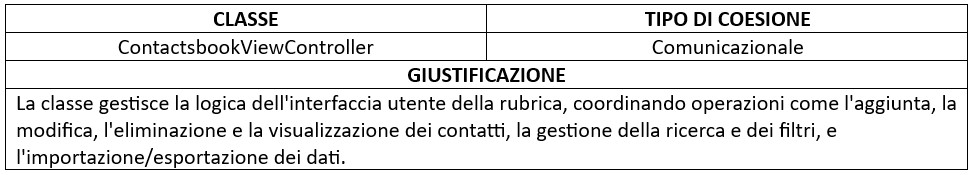
Di seguito sono presenti le tabelle che analizzano coesione e accoppiamento di varie classi.

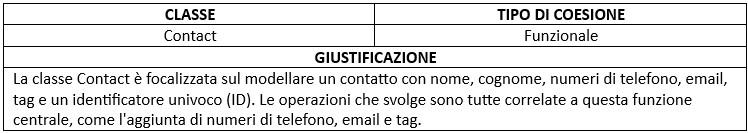


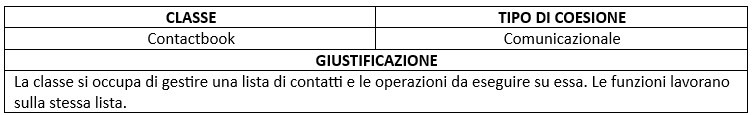


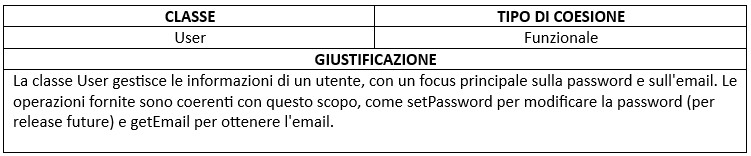


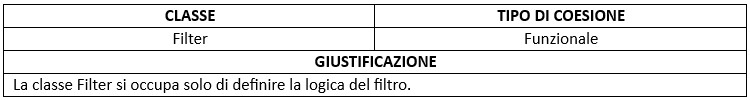


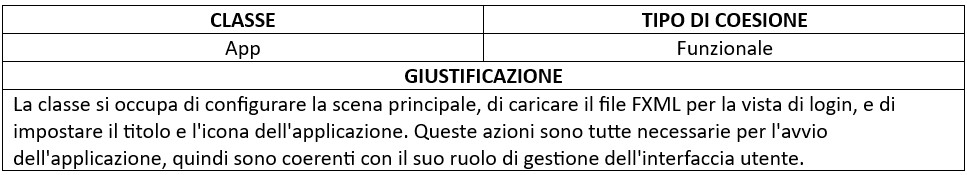


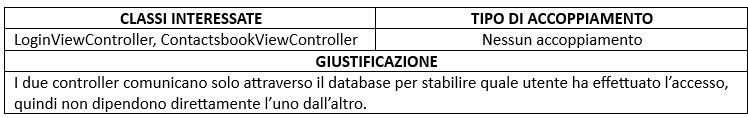


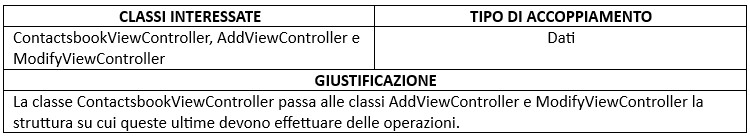


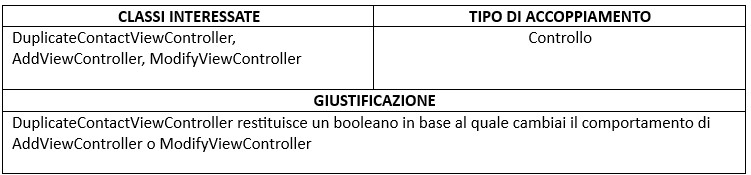




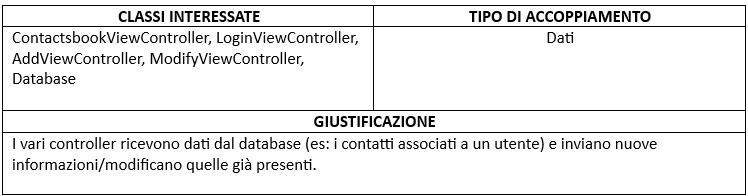












# Convenzioni

Sono state stabilite alcune convenzioni sui nomi da dare ai componenti grafici dell’interfaccia:

* clm per indicare le colonne;
* txt per indicare i campi di testo
* lbl per indicarele etichette
* btn per indicare i bottoni
* mbtn per indicare i menu button
* tblv per indicare la table view
* tm per indicare i radio menu
* chkm per indicare i check menu