

Design

Il progetto è strutturato secondo il modello MVC:

MODEL:

Contact: classe utilizzata per la gestione di un singolo contatto.

ContactBook: utilizzata per la gestione della lista principale.

EmergencyList: utilizzata per la gestione dei contatti d'emergenza.

Bin: utilizzata per la gestione del cestino.

PhoneBook: raggruppa tutti gli elementi della rubrica, utilizzata principalmente per l'import/export dei dati.

ContactList: per eliminare ripetitività, è stata aggiunta una classe astratta che contiene dei metodi comuni a tutte le liste di contatti.

EXCEPTION:

Le **exception** sono state inserite in un package apposito, per una organizzazione più efficace.

InvalidContactException

FullGroupException

VIEW:

PhoneBookView: è stata scelta un'unica view per una gestione semplice ed efficace.

CONTROLLER:

PhoneBookController: un controller per svolgere le funzionalità del model tramite GUI.

La funzionalità di ogni componente viene esplicitata nei diagrammi di sequenza.

OBIETTIVI:

- Alta coesione
- Basso Livello di accoppiamento
- Principi di buona progettazione rispettati

Coesione

La coesione si riferisce al grado in cui i metodi e gli attributi di una classe lavorano insieme per soddisfare un unico scopo ben definito. Nel progetto si è cercato di mantenere un'alta coesione e pertanto ogni modulo è stato progettato con l'obiettivo di ottenere una coesione di tipo funzionale, che garantisca la collaborazione tra le componenti del modulo, oppure comunicazionale con funzionalità che all'interno del modulo operassero sugli stessi dati.

1. **ContactList:**

- Fornisce metodi generici per gestire una lista di contatti, come aggiungere, rimuovere e cercare contatti. Questi metodi sono ben separati ed ognuno svolge un'operazione diversa.

2. **ContactBook, EmergencyList e Bin:**

- Queste sottoclassi, avendo alcuni metodi in comune, ereditano da **ContactList** e aggiungono comportamenti specifici. dato che ogni classe ha un obiettivo ben definito:
 - **ContactBook** gestisce i contatti standard.
 - **EmergencyList** limita il numero massimo di contatti.
 - **Bin** si occupa del recupero o della cancellazione definitiva.

3. **PhoneBook:**

- Si occupa della gestione della rubrica nel suo complesso e il suo caricamento/salvataggio dei dati su file.

4. **Contact:**

- Si occupa della gestione delle informazioni relative a un singolo contatto.

Principi di buona progettazione

1. Principio di Singola Responsabilità (SRP): Ogni classe ha una singola responsabilità precisa e rappresenta una singola entità del progetto.
2. Principio aperto/chiuso: Rispettato attraverso incapsulamento, ereditarietà ed Override
3. Principio di segregazione dell'interfaccia: La progettazione è stata implementata prediligendo una classe astratta rispetto ad un'interfaccia poiché contiene dei metodi già implementati comuni alle sue sottoclassi.
4. Principio di inversione delle dipendenze: Le classi che manipolano contatti non si preoccupano della gestione del file di salvataggio o dell'interfaccia utente, delegando questi compiti a classi separate.

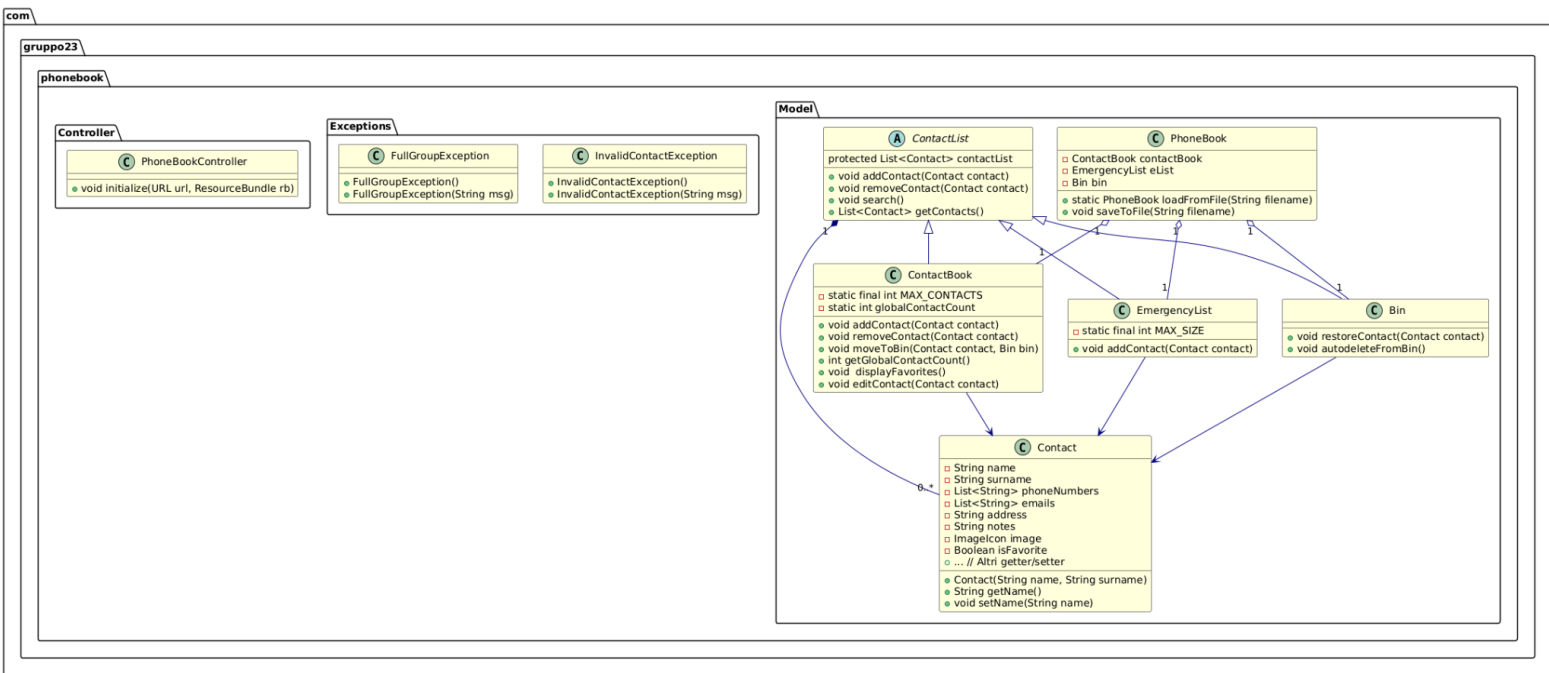
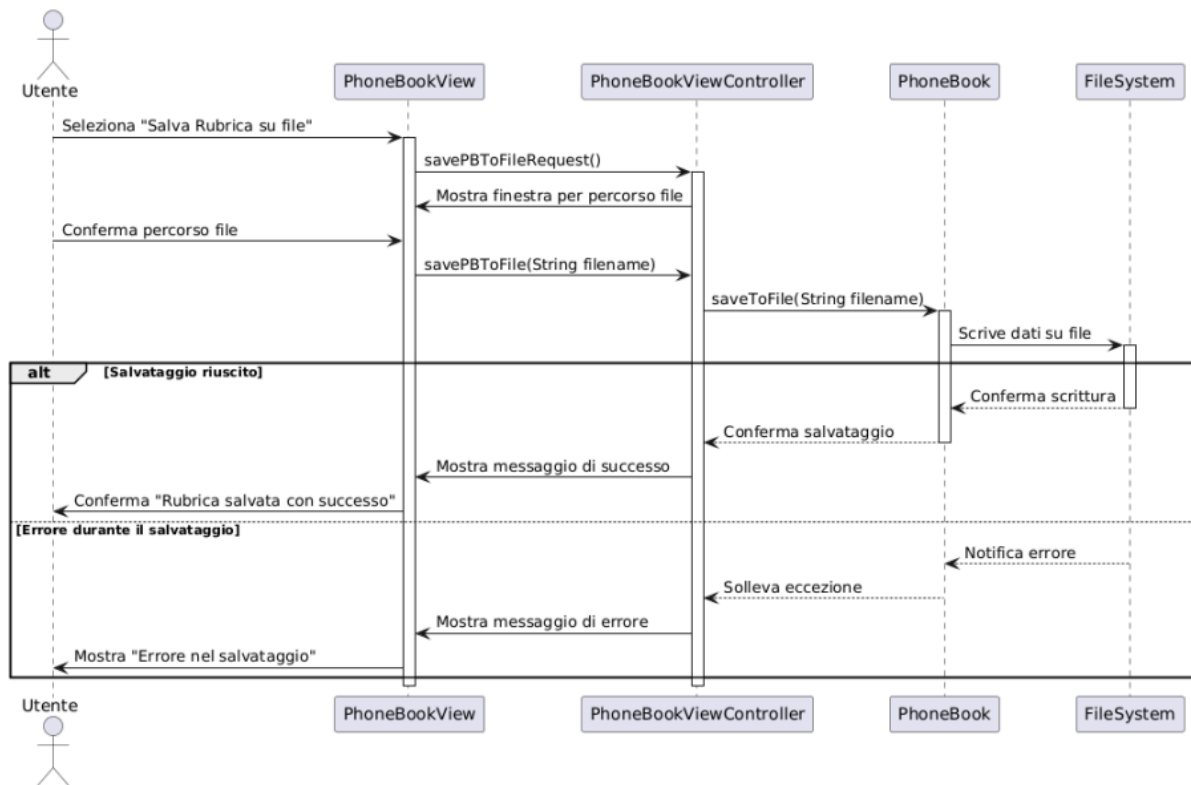


DIAGRAMMA DELLE CLASSI: il diagramma delle classi è stato ideato per essere unito con il diagramma dei package. Le classi inserite hanno attributi e metodi e specificano la loro visibilità e i parametri. I metodi sovrascritti con override sono riportati anche nella sottoclasse, mentre quelli ereditati e non sovrascritti non sono stati inseriti.

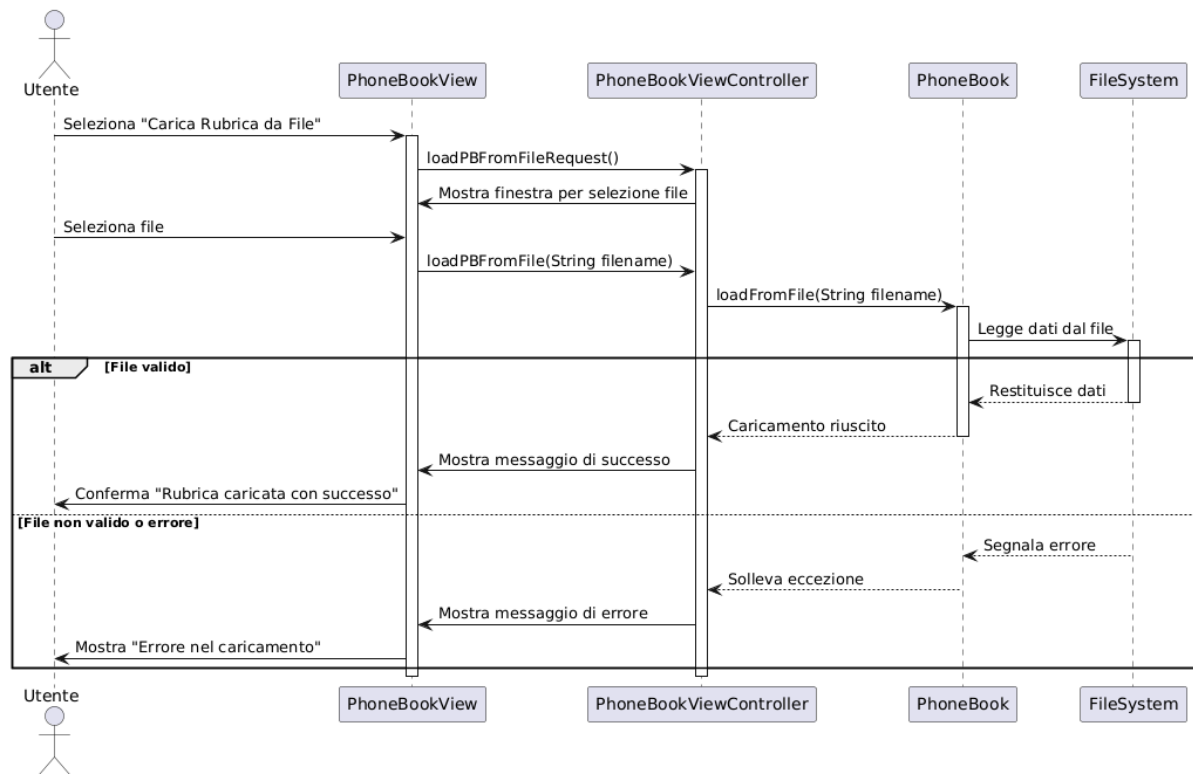
DIAGRAMMI DI SEQUENZA

I diagrammi di sequenza sono stati implementati pensando a come il sistema avrebbe dovuto rispondere a determinati input. Vengono richiamati i metodi del controllore o delle classi per un ordine con il codice sorgente e per una semplice implementazione. Viene inserito l'attore (Utente) e i partecipanti. Il diagramma mostra anche l'interazione con la GUI.

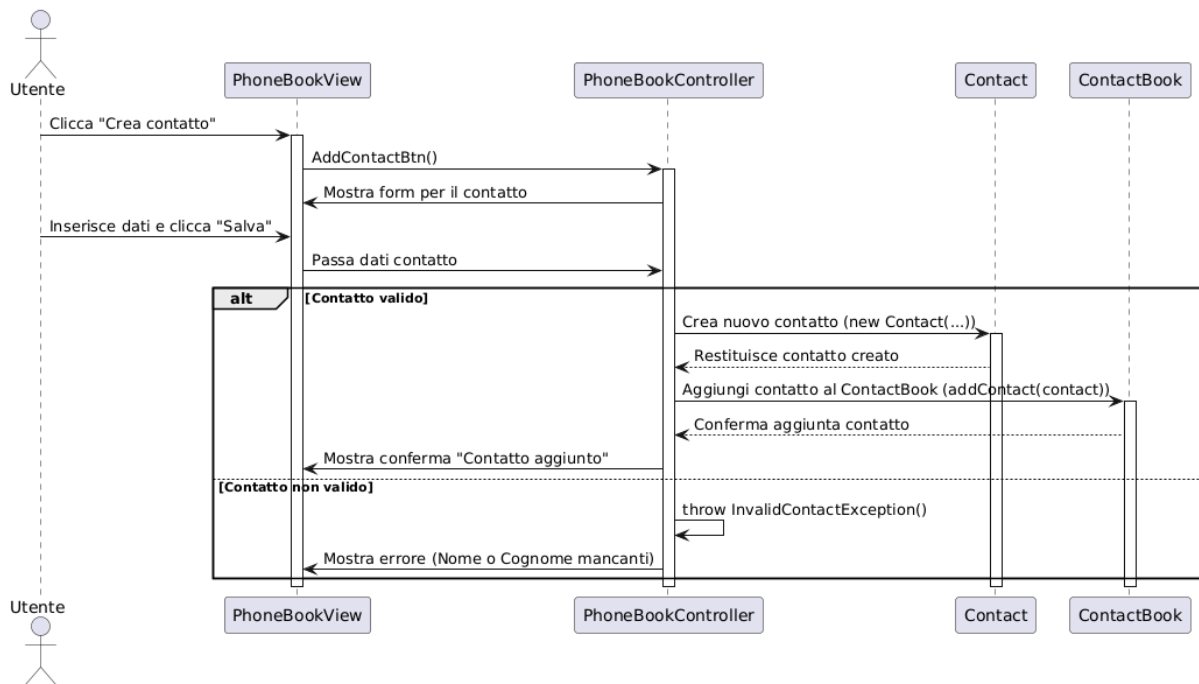
Salva Rubrica su file



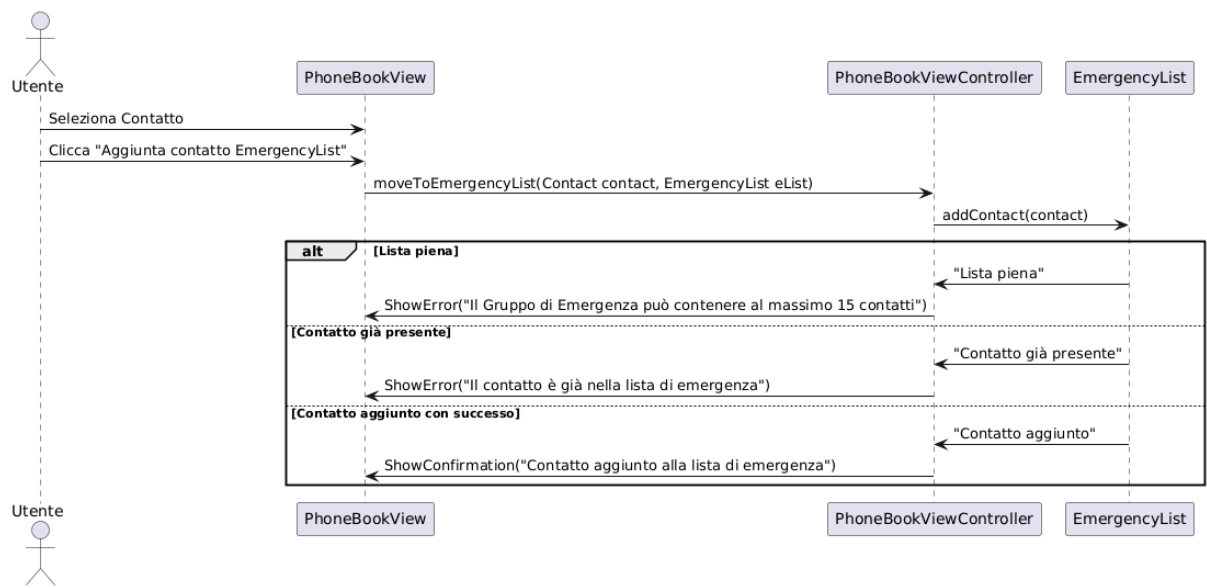
Carica Rubrica da file



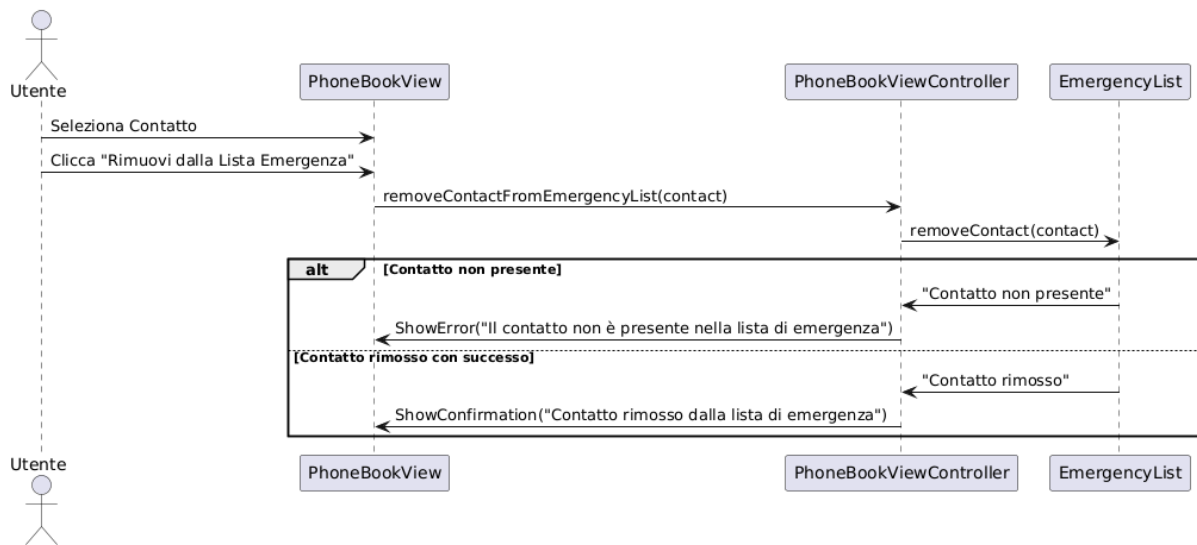
Crea contatto



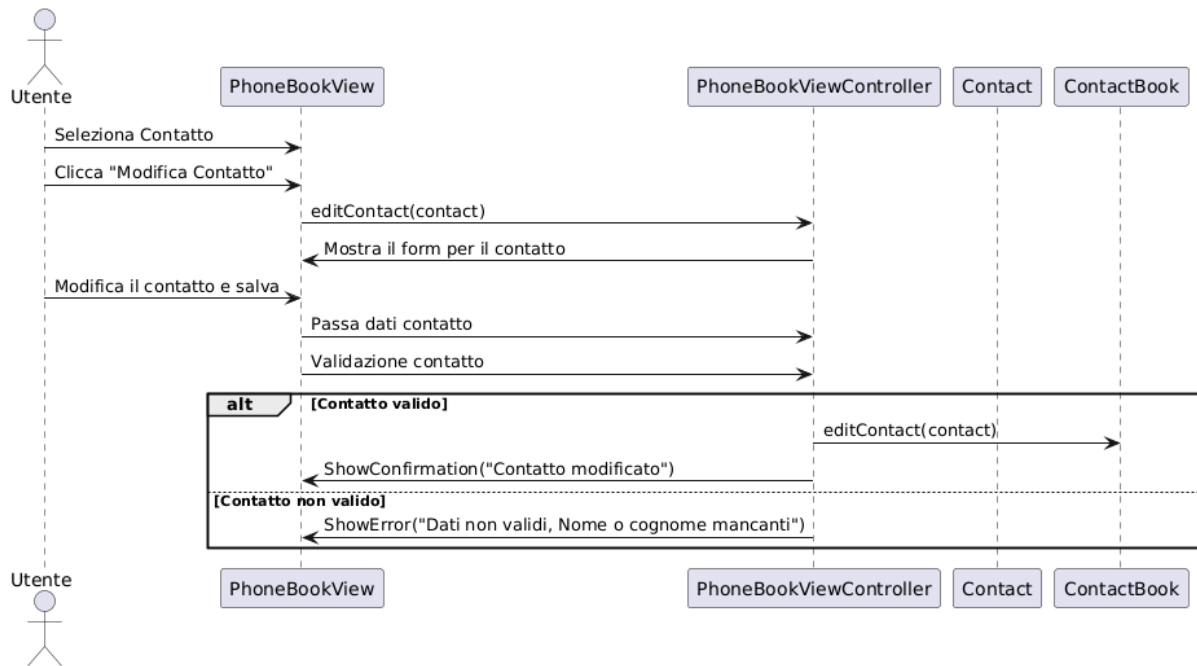
Aggiunta contatto EmergencyList



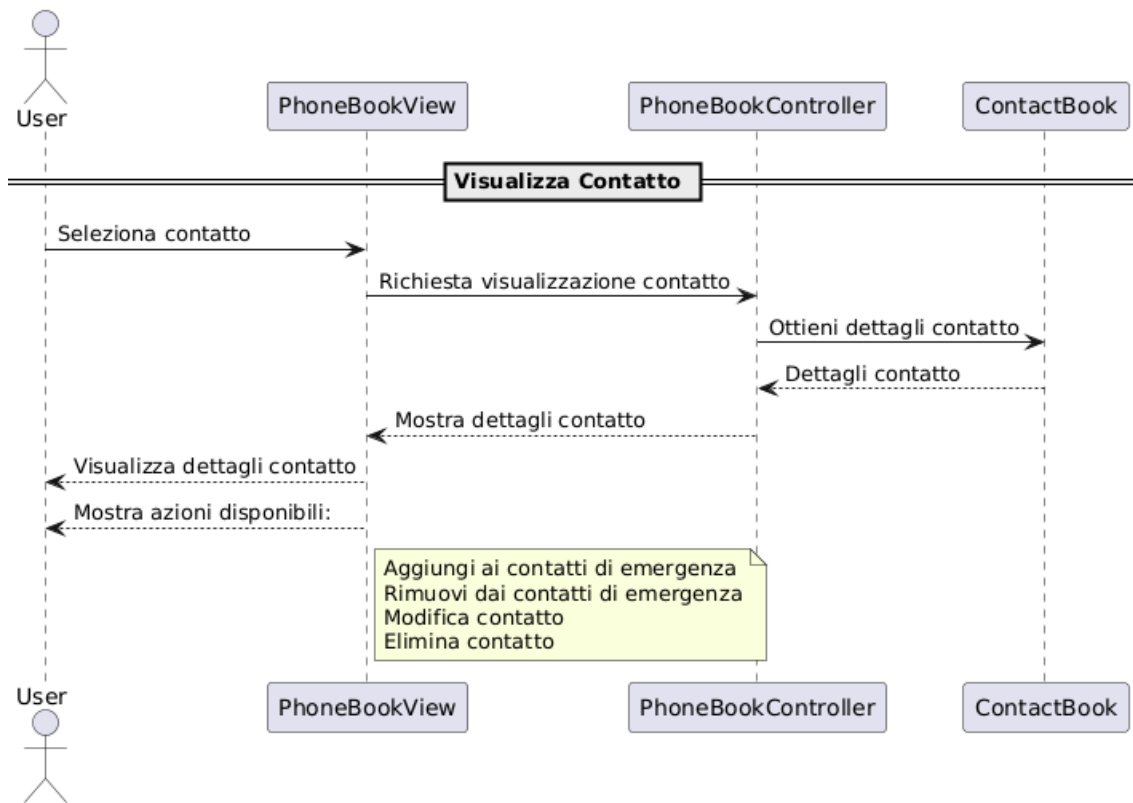
Rimozione contatto EmergencyList



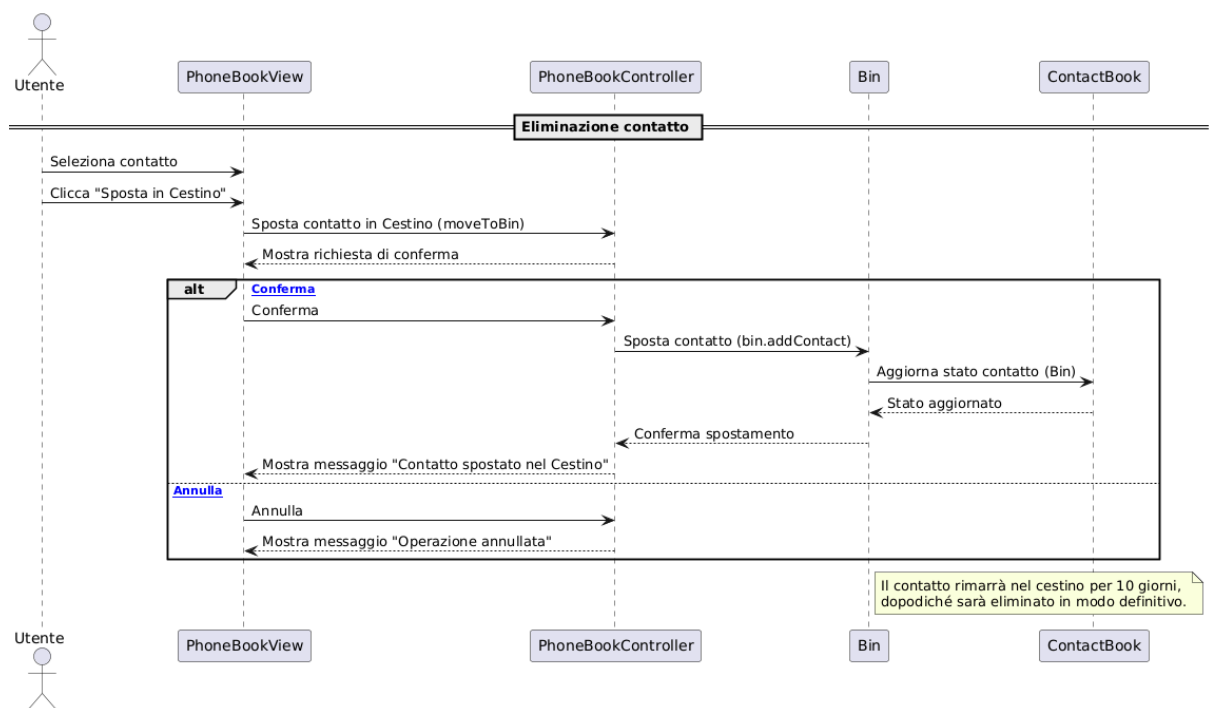
Modifica contatto



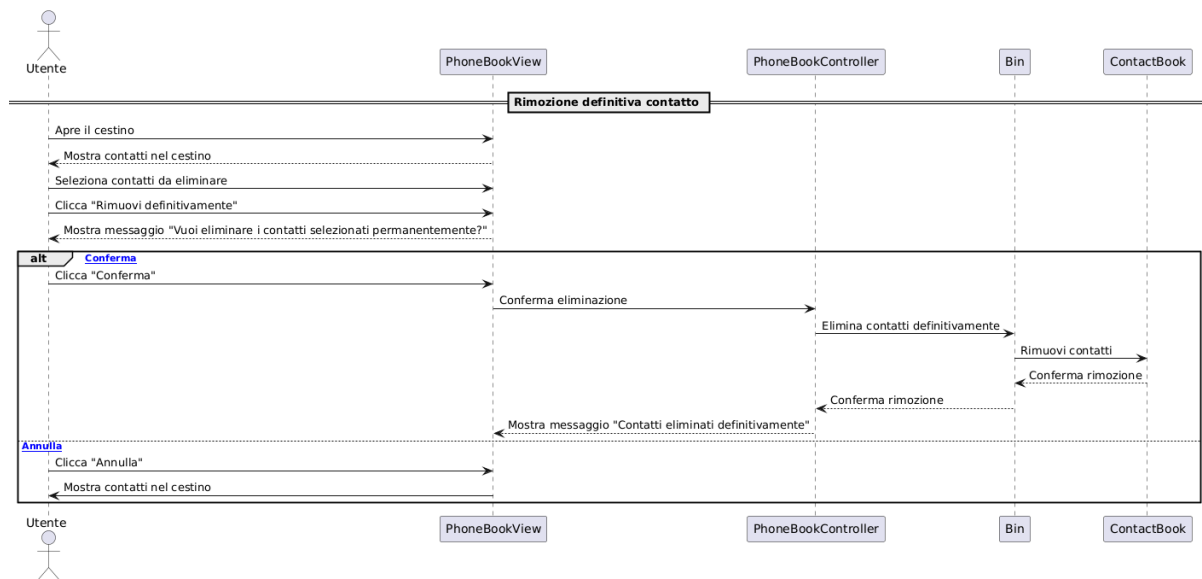
Visualizza contatto



Eliminazione contatto



Rimozione definitiva contatto



Ricerca contatto

