

Documento di Design

Nazzaro Luca, Radesca Antonio, Savino Michele, Scandone Alessandro

Indice

1. Introduzione
2. Principi di buona progettazione
3. Livelli di coesione
4. Livelli di accoppiamento

1. Introduzione

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: utilizzata per l'import/export dei dati. Quando vado a importare/esportare la rubrica, importo/esporto la lista principale, il gruppo di emergenza e il cestino.

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.

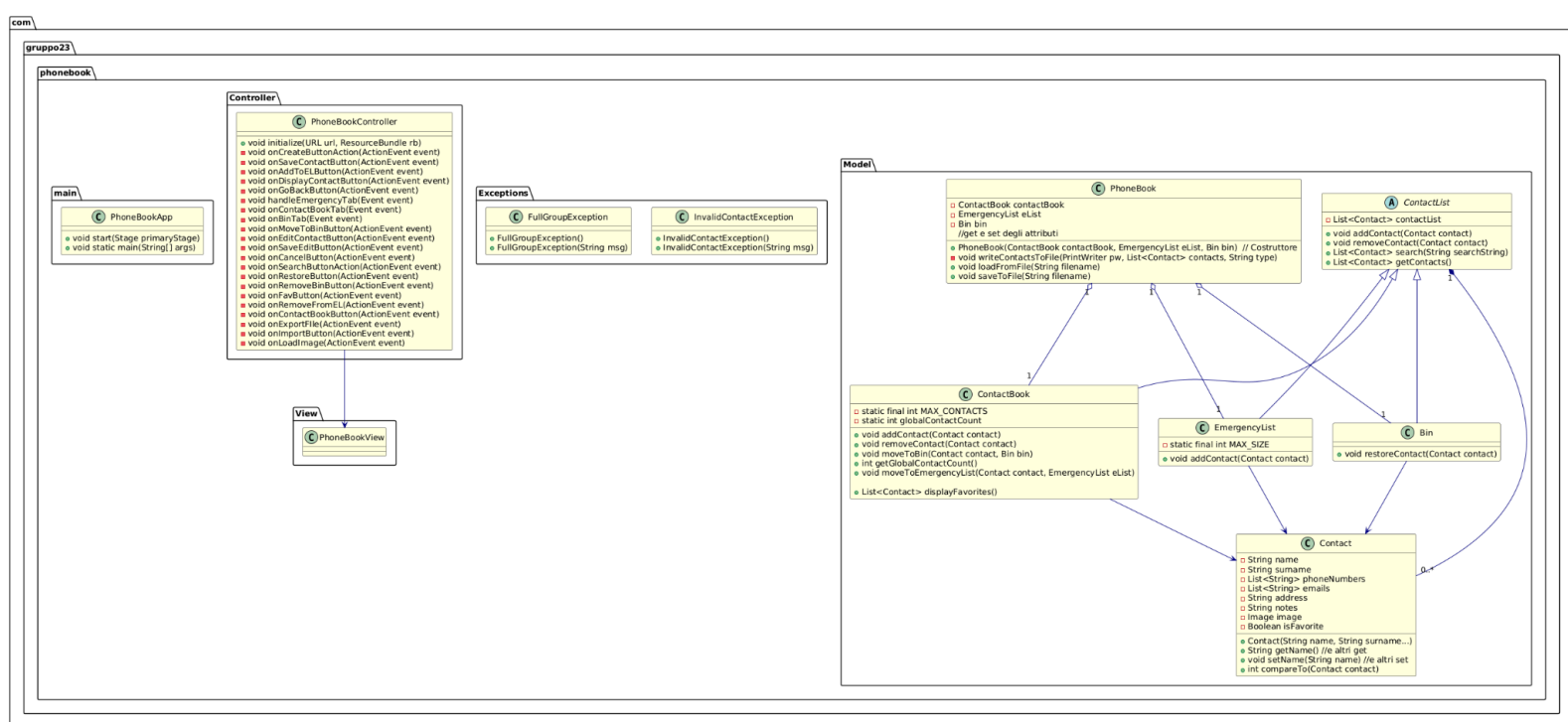
La funzionalità di ogni componente viene esplicitata nei diagrammi di sequenza.

OBIETTIVI:

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

DIAGRAMMA DELLE CLASSI:

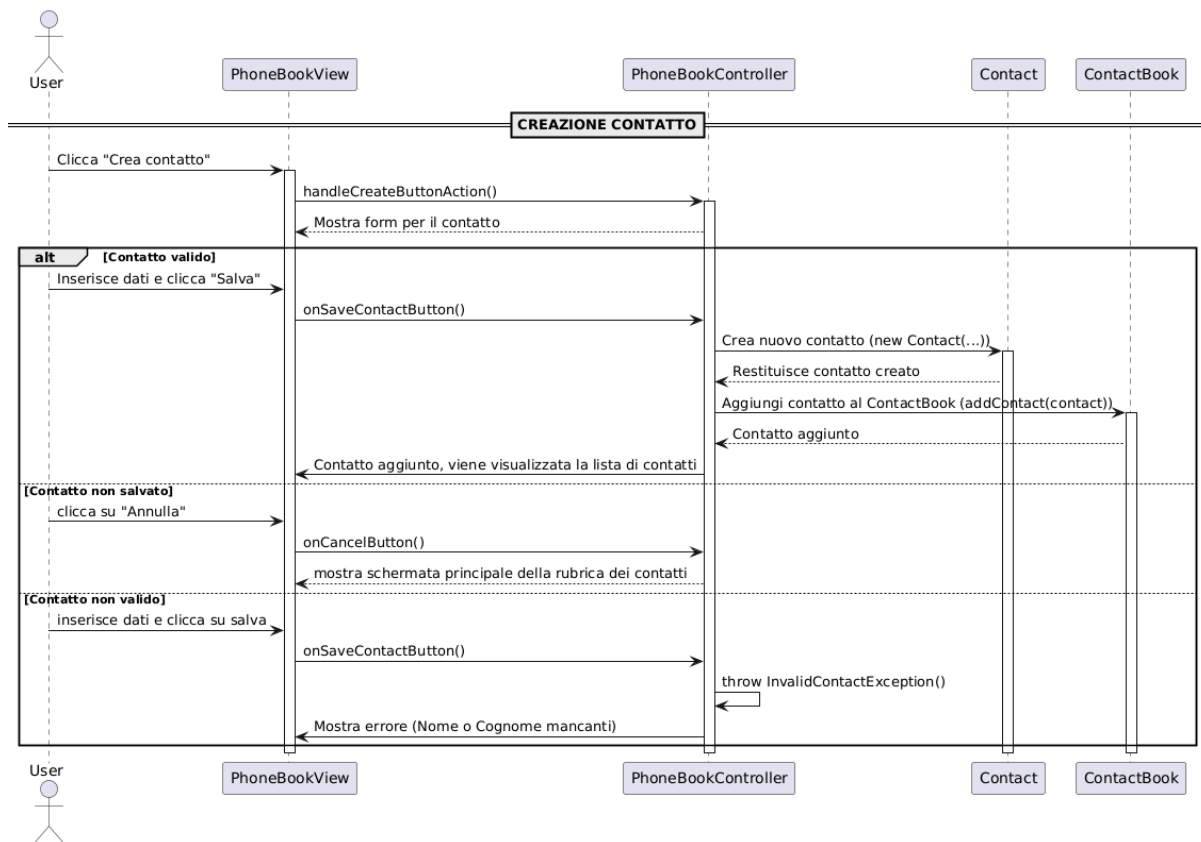
Il diagramma delle classi è stato ideato per essere unito con il diagramma dei package. Le classi inserite hanno attributi e metodi e specificano la loro visibilità e i parametri. I metodi sovrascritti con override sono riportati anche nella sottoclasse, mentre quelli ereditati e non sovrascritti non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



2. 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.

Creazione Contatto



Il diagramma illustra il processo di **creazione di un contatto** nella rubrica. L'Utente seleziona l'opzione "*Crea Contatto*", dopodiché viene visualizzato un form con i campi necessari per l'inserimento dei dati del contatto.

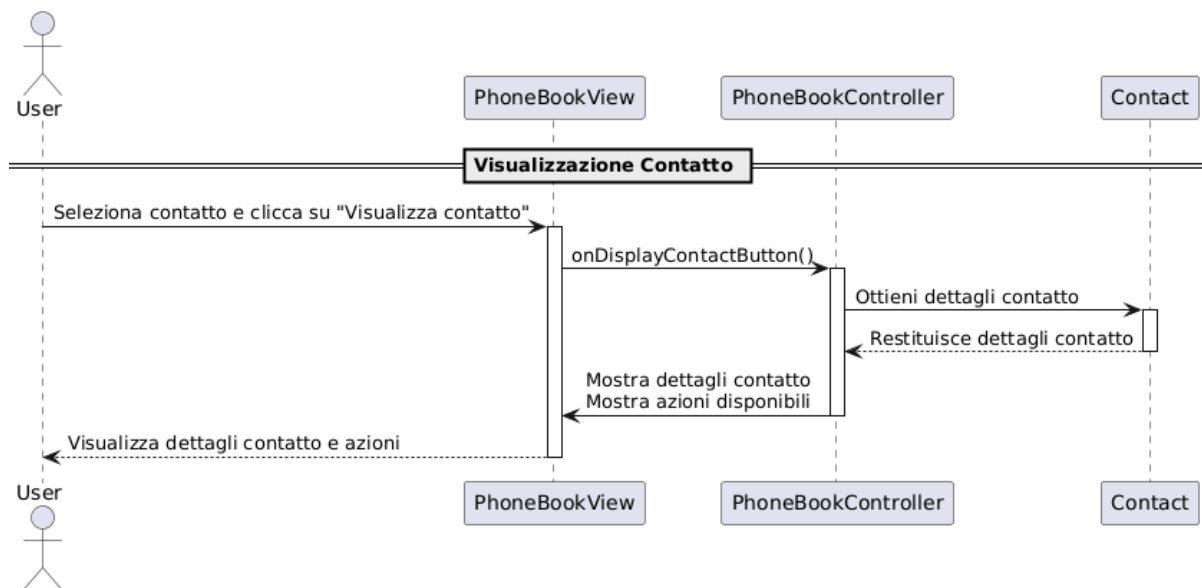
1. Salvataggio del contatto valido:

- L'utente compila almeno uno dei campi obbligatori (*Nome* o *Cognome*) e preme il pulsante "*Salva Contatto*".
- Il sistema crea un nuovo contatto e lo aggiunge alla RubricaContatti.
- Una volta completata l'operazione, viene visualizzata la lista aggiornata dei contatti.

2. Annullamento dell'operazione:

- L'utente può annullare la creazione del contatto premendo il pulsante "*Annulla*".
 - Il sistema ritorna alla schermata principale della rubrica dei contatti senza salvare alcuna modifica.
3. **Errore per contatto non valido:**
- Se l'utente tenta di salvare il contatto senza compilare almeno uno dei campi obbligatori (*Nome* o *Cognome*), il sistema genera un errore.
 - Un messaggio di errore viene mostrato all'utente per segnalare i campi mancanti.
 - La creazione del contatto viene interrotta.

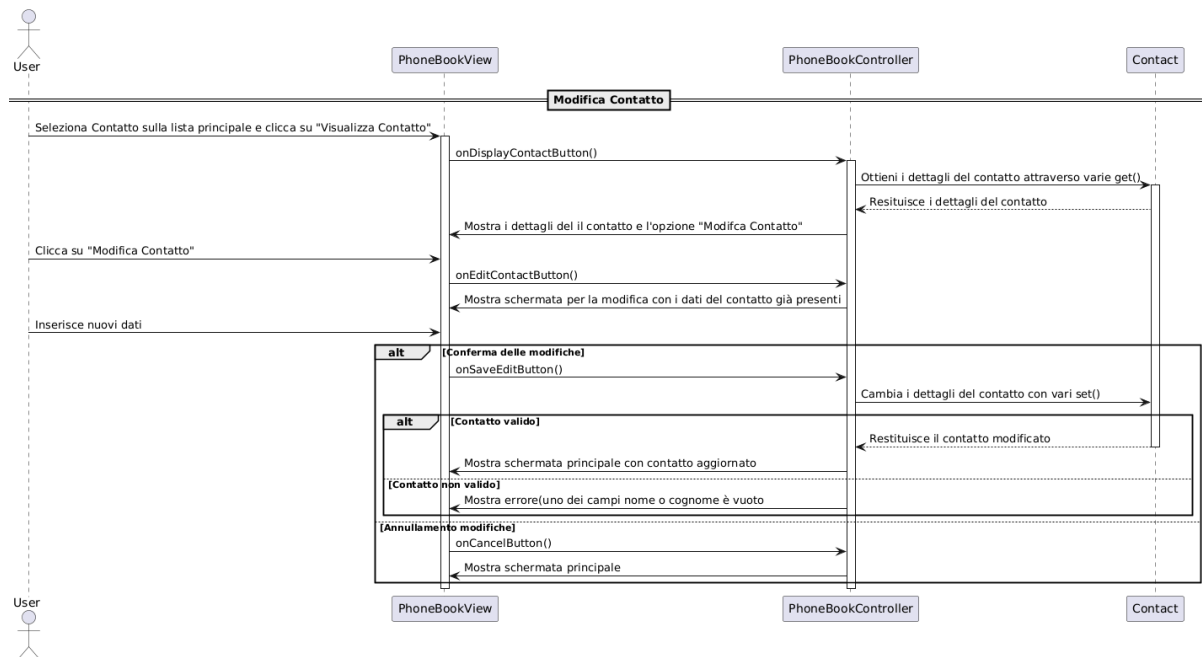
Visualizzazione Contatto



Il processo di **Visualizzazione Contatto** consente all'utente di accedere ai dettagli di un contatto presente nella rubrica. L'utente seleziona un contatto dalla lista visualizzata nella rubrica dei contatti e clicca l'opzione "Visualizza contatto"

1. Il sistema attiva l'operazione "**Visualizza Contatto**" e recupera i dati completi del contatto selezionato.
2. Viene mostrata una schermata dedicata con i seguenti dettagli del contatto:
 - **Nome**
 - **Cognome**
 - **Numeri di telefono**
 - **Email** (se presenti)
 - Altre **informazioni aggiuntive** salvate durante la creazione del contatto.
3. L'utente a questo punto ha accesso alle azioni disponibili sul contatto.

Modifica Contatto



Il diagramma illustra il processo di **modifica di un contatto** nella rubrica. L'utente seleziona un contatto dalla lista principale, visualizza i dettagli del contatto e poi può modificarlo.

1. Visualizzazione del contatto:

- L'utente seleziona un contatto dalla lista principale e clicca su “Visualizza Contatto”.
- Il sistema recupera i dettagli del contatto selezionato e li mostra all'utente.
- Viene visualizzata una schermata con le informazioni dettagliate del contatto, inclusi nome, cognome, numeri di telefono e altre informazioni memorizzate.
- Il sistema mostra anche l’opzione “Modifica Contatto”.

2. Modifica del contatto:

- L'utente clicca su “Modifica Contatto” per avviare la modifica del contatto.
- Il sistema mostra una schermata con i campi del contatto precedentemente compilati con i dati esistenti.
- L'utente inserisce i nuovi dati nei campi di modifica.

3. Conferma delle modifiche:

- L'utente preme il pulsante “Salva Modifiche” per confermare le modifiche.
- Il sistema valida i dati e, se i campi obbligatori sono completi, aggiorna il contatto con le nuove informazioni.
- Se il contatto è valido, il sistema restituisce il contatto modificato.
- Il sistema mostra la lista aggiornata dei contatti nella rubrica principale, con il contatto appena modificato.

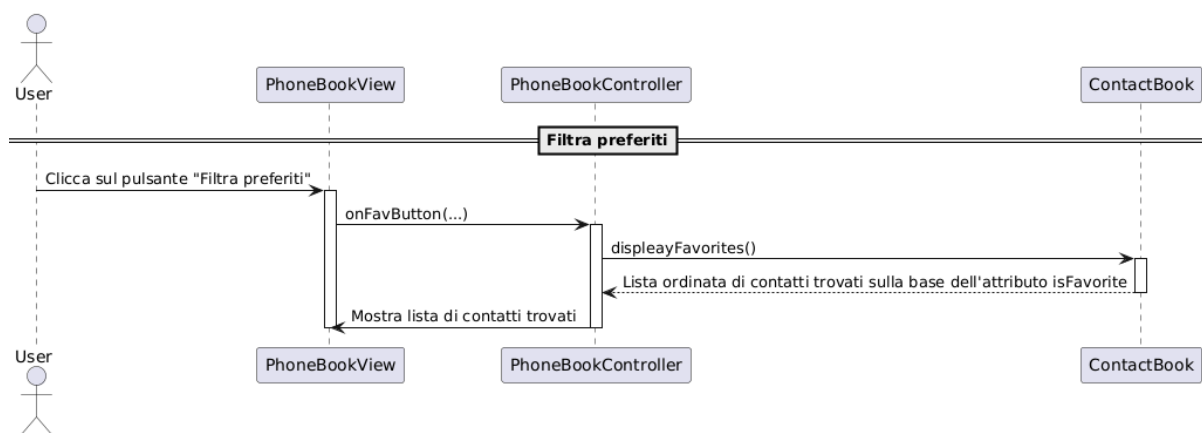
4. Errore nella modifica del contatto:

- Se l'utente lascia vuoti i campi obbligatori (ad esempio, nome o cognome), il sistema genera un errore.
- Il sistema mostra un messaggio di errore, indicando che i campi obbligatori non sono stati completati.
- La modifica del contatto non viene applicata e la schermata di modifica rimane aperta per consentire all'utente di correggere i dati.

5. Annullamento delle modifiche:

- L'utente può annullare la modifica del contatto premendo il pulsante “**Annulla**”.
- Il sistema ritorna alla schermata principale della rubrica, senza salvare alcuna modifica al contatto.
- La lista dei contatti rimane invariata.

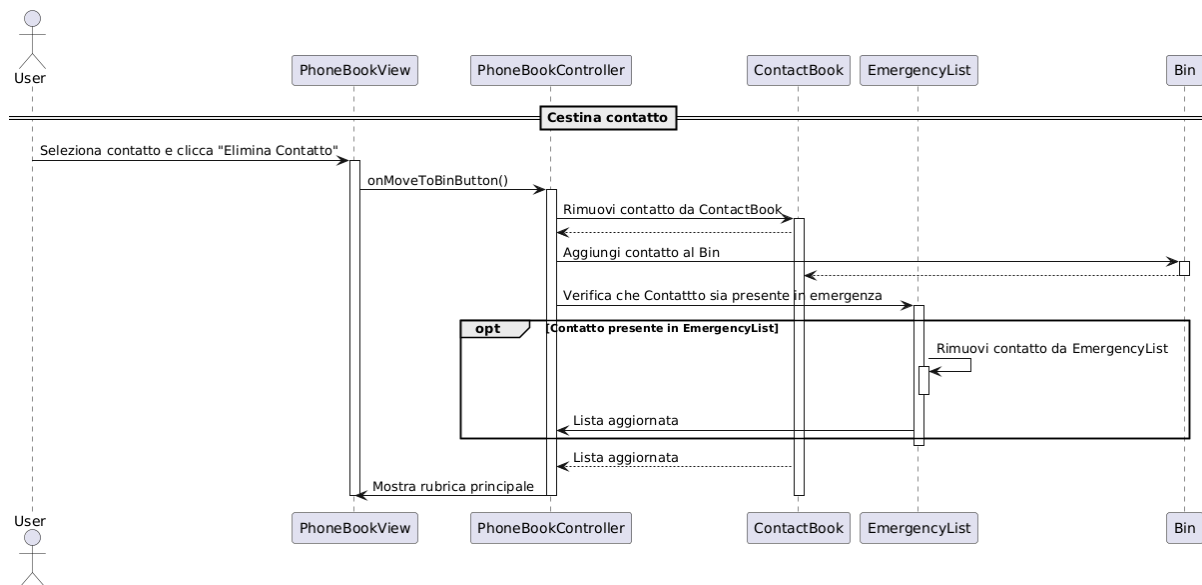
Filtra Preferiti



Il diagramma illustra il processo di **Filtra Preferiti** nella Rubrica. L'utente, dalla schermata principale della rubrica, seleziona l'opzione “*Filtra Preferiti*”.

1. Il sistema attiva l'operazione “*Filtra Preferiti*” e recupera dalla rubrica tutti i contatti contrassegnati come **preferiti**.
2. Viene mostrata una schermata dedicata contenente la lista dei contatti preferiti con i seguenti
3. L'utente a questo punto può:
 - Visualizzare i dettagli di un contatto preferito.
 - Tornare alla lista completa dei contatti.

Cestina Contatto

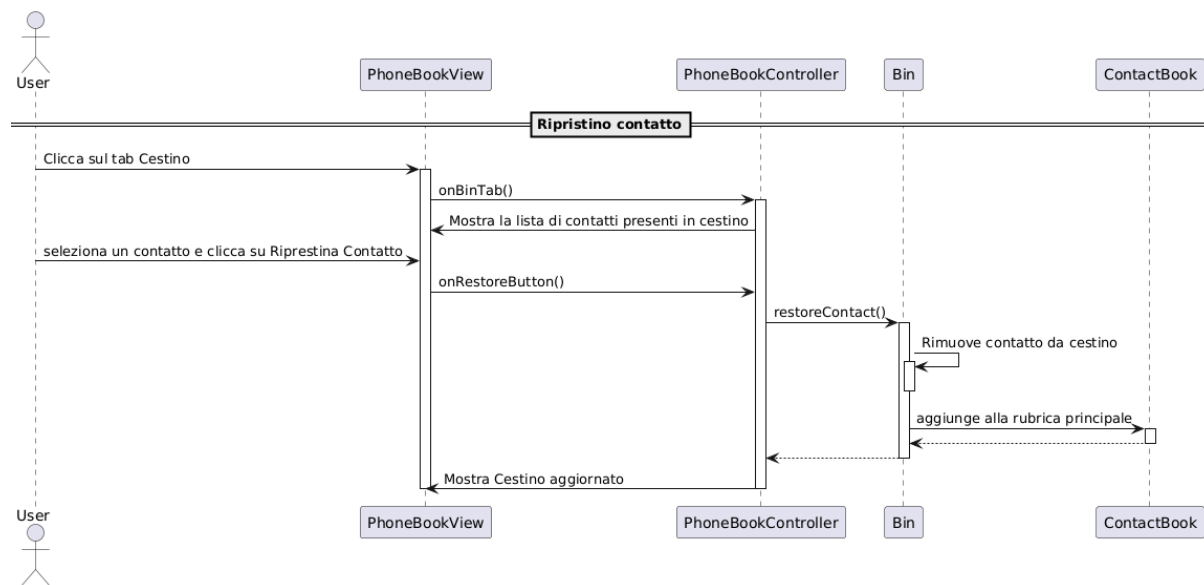


Il diagramma illustra il processo di **spostamento di un contatto nel cestino** nella rubrica. L'Utente seleziona un contatto dalla lista dei contatti visualizzata nella schermata principale della rubrica.

1. Spostamento del contatto nel cestino:

- L'utente seleziona un contatto dalla rubrica principale.
- L'utente preme l'opzione "Elimina Contatto".
- Il contatto viene rimosso dalla rubrica principale e spostato in Cestino
- Il sistema verifica se il contatto è presente nella lista dei contatti di emergenza.
- Se è presente in lista di emergenza, il contatto viene rimosso dalla lista di emergenza
- Una volta completata l'operazione, viene visualizzata la lista aggiornata dei contatti senza il contatto appena spostato.

Ripristino Contatto

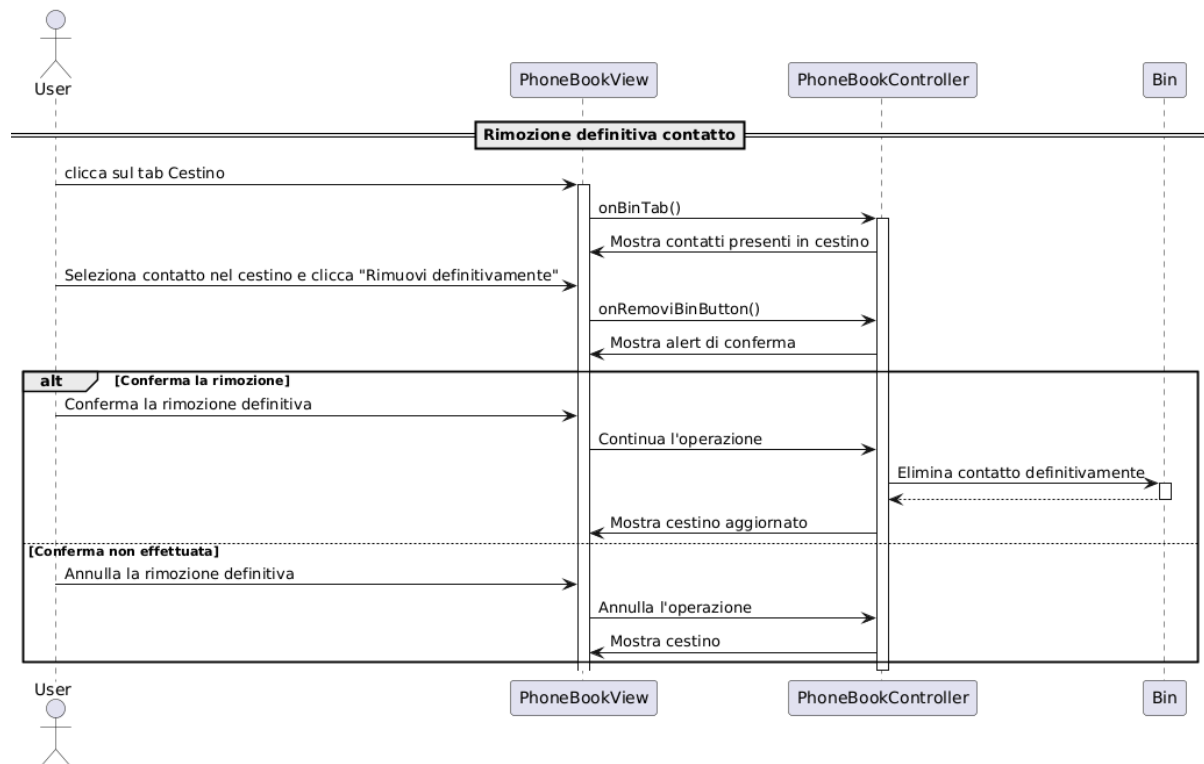


Il diagramma illustra il processo di **ripristino di un contatto dal cestino** nella rubrica. L'utente seleziona un contatto dalla lista dei contatti presenti nel cestino e lo ripristina nella rubrica principale.

1. Ripristino del contatto dal cestino:

- L'utente apre la rubrica e clicca la sezione "Cestino".
- L'app mostra i contatti presenti nel Cestino.
- L'utente seleziona un contatto che desidera ripristinare.
- L'utente preme l'opzione "Ripristina contatto".
- Il sistema rimuove il contatto dal Cestino e lo aggiunge alla Rubrica Principale.
- Una volta completata l'operazione, viene visualizzata la lista aggiornata dei contatti all'interno del cestino.

Rimozione definitiva di un Contatto



Il diagramma illustra il processo di rimozione definitiva di un contatto dalla rubrica. L'utente seleziona un contatto nel cestino e lo elimina definitivamente.

1. Apertura del Cestino:

- L'utente apre la sezione “Cestino” nella rubrica.
- L'app mostra i contatti attualmente presenti nel Cestino.

2. Selezione del contatto da eliminare:

- L'utente seleziona un contatto dalla lista dei contatti presenti nel Cestino che desidera eliminare definitivamente.

3. Scelta dell'opzione di eliminazione definitiva:

- L'utente preme l'opzione “Rimuovi definitivamente”.
- Il sistema mostra un messaggio di conferma: “Sei sicuro di voler eliminare definitivamente il contatto?”.
- Il messaggio presenta due opzioni selezionabili: “OK” e “Annulla”.

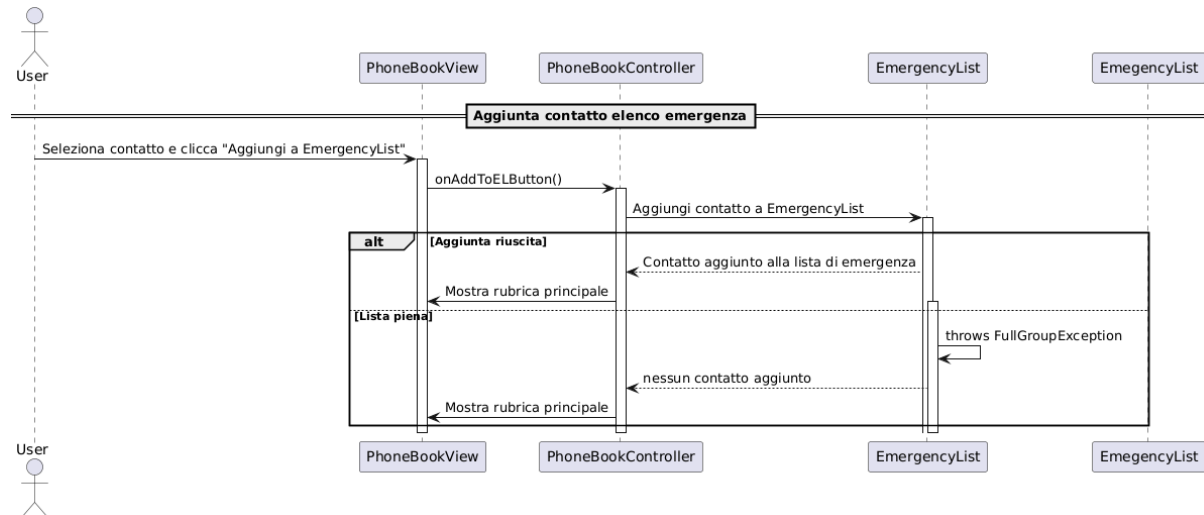
4. Conferma dell'eliminazione definitiva:

- L'utente clicca su “OK” per confermare l'eliminazione definitiva del contatto.
- Il sistema elimina definitivamente il contatto dal Cestino e dalla rubrica.
- Una volta completata l'operazione, il contatto non sarà più presente nel Cestino né nella rubrica e non sarà recuperabile.

5. Annullamento dell'operazione:

- Se l'utente clicca su “Annulla”, l'operazione di eliminazione viene annullata.
- L'utente torna alla schermata del Cestino e la lista dei contatti rimane invariata.

Aggiunta contatto elenco Emergenza



Il diagramma illustra il processo di **aggiunta di un contatto all'elenco di emergenza** nella rubrica.

1. Selezione del contatto dalla rubrica principale:

- L'utente seleziona un contatto dalla lista principale dei contatti presenti nella rubrica.

2. Aggiunta del contatto all'elenco di emergenza:

- L'utente preme l'opzione “Aggiungi a EmergencyList”.
- Il sistema effettua un controllo sul numero di contatti già presenti nell'elenco "Contatti di Emergenza" per verificare che non venga superato il limite massimo stabilito.

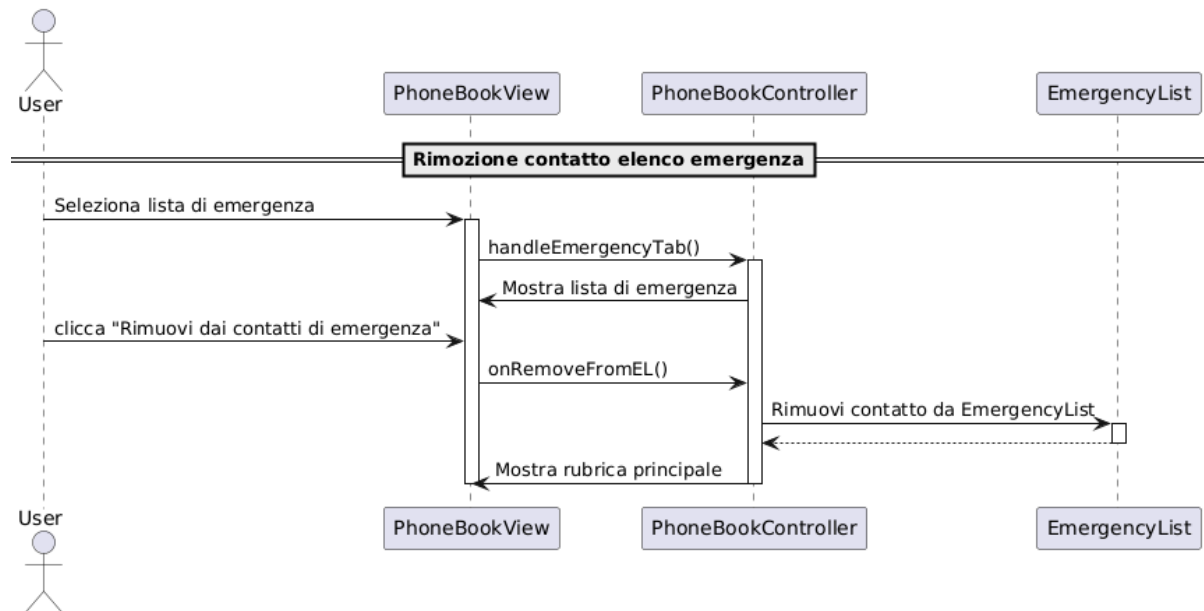
3. Verifica del limite massimo:

- Se il controllo sul numero di contatti restituisce un esito positivo (il limite non è stato superato), il sistema aggiunge il contatto selezionato all'elenco "Contatti di Emergenza" senza rimuoverlo dalla rubrica principale.
- Una volta completata l'operazione, il contatto è visibile sia nella rubrica principale che nell'elenco "EmergencyList".

4. Errore nel caso in cui il limite massimo sia raggiunto:

- Se il sistema effettua il controllo e rileva che il limite massimo di contatti nell'elenco di emergenza è già stato raggiunto, il sistema visualizza un messaggio di errore: “**Limite massimo raggiunto**”.
- L'utente viene informato dell'impossibilità di aggiungere il contatto all'elenco di emergenza.
- Viene mostrata all'utente la rubrica principale.

Rimozione di un contatto dall'elenco di Emergenza



Il diagramma illustra il processo di **rimozione di un contatto dall'elenco di emergenza**. L'utente seleziona un contatto dall'elenco e lo rimuove, riportandolo alla rubrica principale.

1. Selezione dell'elenco "Contatti di Emergenza"

- L'utente seleziona la lista dei contatti di emergenza nella rubrica. L'app mostra tutti i contatti attualmente presenti nell'elenco di emergenza.

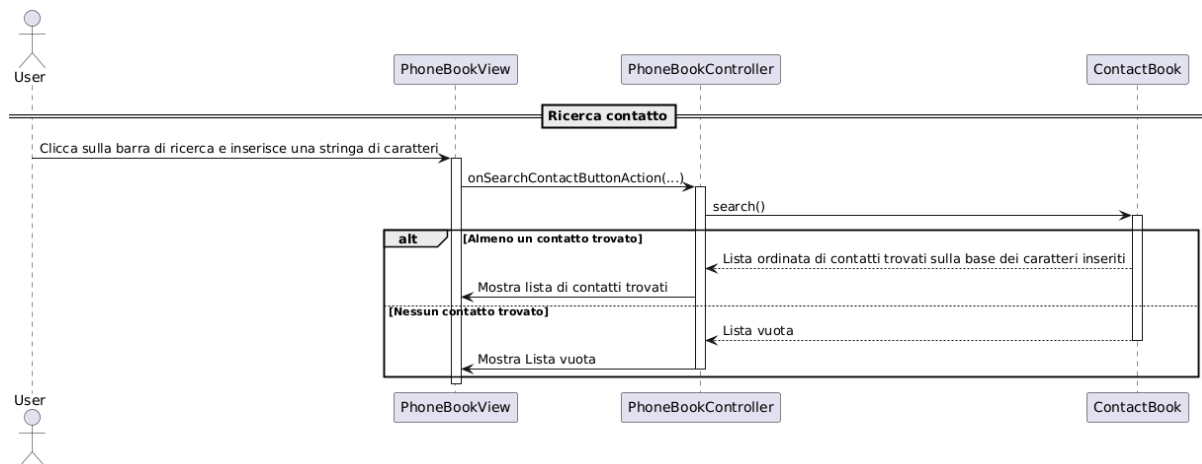
2. Selezione del contatto da rimuovere

- L'utente seleziona un contatto dalla lista dei **"Contatti di Emergenza"**. Il sistema mostra i dettagli del contatto selezionato, inclusi nome, cognome e altre informazioni rilevanti.

3. Conferma per la rimozione

- L'utente preme l'opzione **"Rimuovi dai contatti di emergenza"**, quindi viene mostrata la lista dei **"Contatti di Emergenza"** aggiornata.

Ricerca di un Contatto



Il diagramma illustra il processo di **ricerca di un contatto** nella rubrica. L'utente clicca sulla barra di ricerca e inserisce una stringa di caratteri per visualizzare i contatti che corrispondono al criterio di ricerca.

1. Inserimento della lettera o sottostringa nella barra di ricerca:

- L'utente inserisce una lettera o una sottostringa nella barra di ricerca, situata nella **rubrica principale** o in un **elenco** specifico. Il sistema avvia la ricerca mentre l'utente digita.

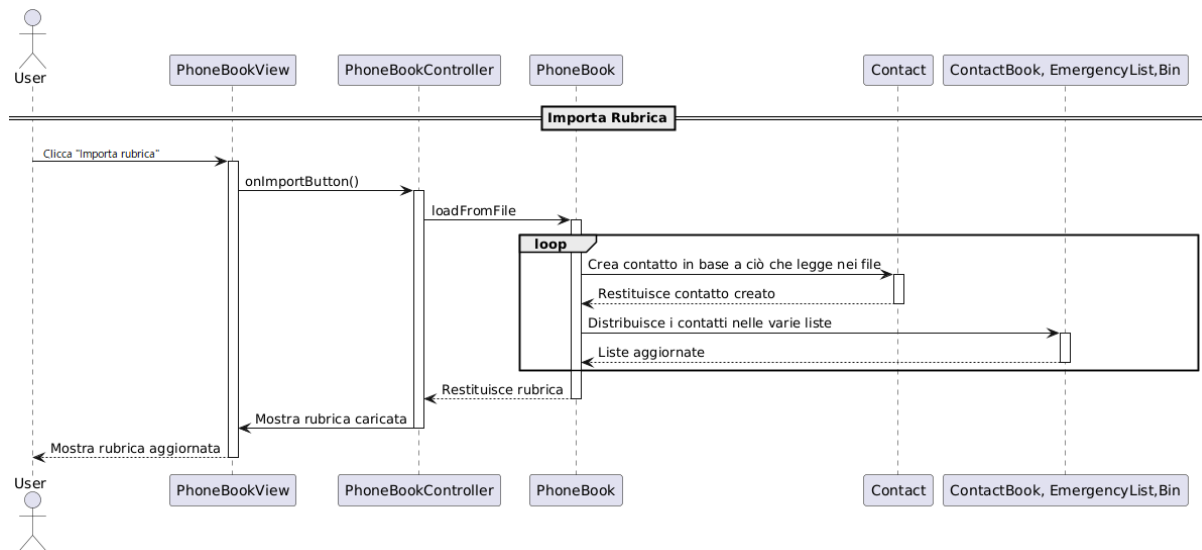
2. Visualizzazione dei contatti che corrispondono alla ricerca:

- Il sistema visualizza i contatti che contengono la sottostringa inserita nel **nome** e/o nel **cognome**. I contatti vengono mostrati in **ordine alfabetico**, prima per **cognome** e, nel caso in cui non ci fosse un cognome, viene trattato il nome come se fosse cognome per il corretto ordinamento alfabetico.

3. Nessun contatto trovato

- La lista risulta vuota.

Importa Rubrica



Il diagramma illustra il processo di **importazione di una rubrica**. L'utente può importare una rubrica da una fonte esterna, integrando i contatti già esistenti nell'applicazione.

1. Selezione dell'opzione "Importa rubrica"

- L'utente seleziona l'opzione **"Importa rubrica"** nel menu principale dell'applicazione. Il sistema avvia la procedura di importazione.

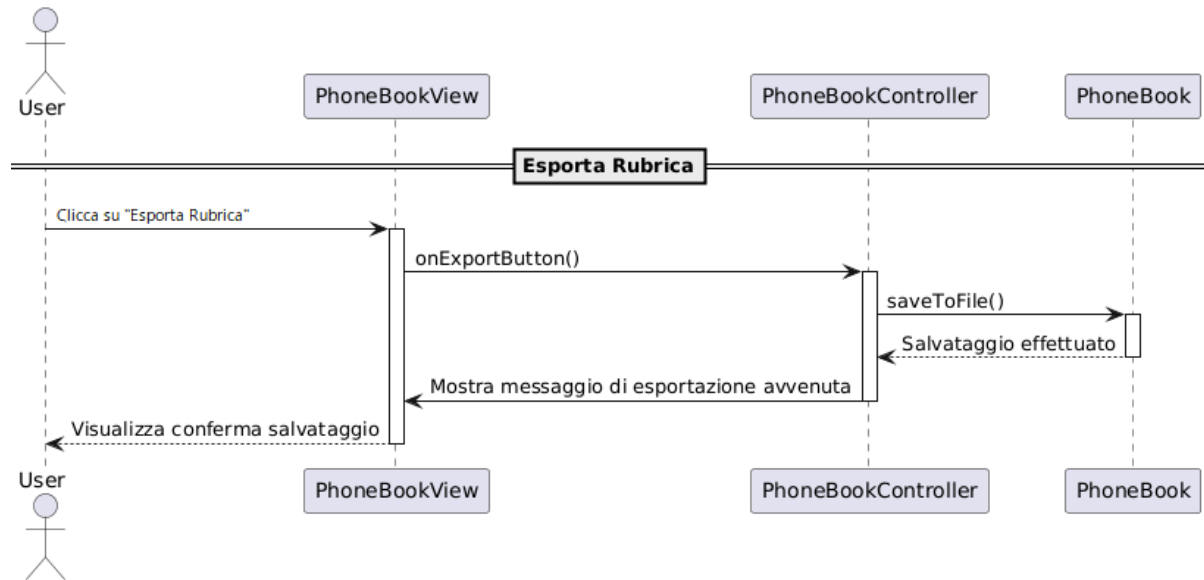
2. Importazione dei contatti

- Il sistema importa i contatti selezionati dalla fonte esterna nella rubrica principale dell'app. I nuovi contatti vengono aggiunti alla **RubricaContatti**, senza rimuovere quelli già esistenti. Il sistema distribuisce i contatti e aggiorna le varie liste.

3. Completamento dell'importazione

- La rubrica principale viene aggiornata con i nuovi contatti.

Esporta Rubrica



Il diagramma illustra il processo di **esportazione della rubrica**. L'utente può esportare i contatti presenti nella rubrica in un file CSV locale.

1. Selezione dell'opzione "Esporta rubrica"

- L'utente seleziona l'opzione **“Esporta rubrica”** nel menu principale dell'applicazione per avviare il processo di esportazione della rubrica contatti.

2. Salvataggio della rubrica in un file CSV

- Il sistema salva i dati della rubrica in un file locale. Ogni contatto presente nella rubrica viene esportato nel file, pronto per essere trasferito o utilizzato in altre applicazioni.

3. Principi di buona progettazione

Principio di Singola Responsabilità (SRP): Ogni classe ha una singola responsabilità precisa e rappresenta una singola entità del progetto.

Principio aperto/chiuso: Rispettato attraverso incapsulamento, ereditarietà ed Override

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.

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.

4. Livelli di 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. Dunque le classi presentano metodi che operano su un'unica "area" dell'intera applicazione.

Classe	Livello di Coesione	Descrizione
ContactList	Comunicazionale	Questa classe astratta contiene metodi e attributi ereditabili da altre classi.
ContactBook	Comunicazionale	Questa classe rappresenta una rubrica per la gestione dei contatti, estendendo la funzionalità di ContactList . Fornisce operazioni aggiuntive come l'aggiunta, la rimozione, lo spostamento dei contatti in un cestino o in una lista di emergenza, nonché il conteggio globale dei contatti.
Contact	Funzionale	Questa classe rappresenta un contatto

		<p>individuale. Essa include attributi per informazioni personali e funzionalità di accesso e modifica, oltre a metodi di utilità come la comparazione tra contatti.</p>
PhoneBook	Sequenziale	<p>È una classe che lavora sull'Import/Export dei file, possedendo due metodi diversi ed essendo dotata di un metodo ausiliario.</p>
Bin	Comunicazionale	<p>I metodi della classe Bin operano sui dati dei contatti rimossi dalla rubrica e li trattano come una lista separata (il cestino). Tuttavia, questi contatti sono ancora correlati alla rubrica principale e alla lista di emergenza, poiché provengono dalla stessa rubrica. Il metodo restoreContact ripristina i contatti dal cestino e li rimette nella rubrica principale, mantenendo i dati coerenti tra le diverse strutture.</p>
EmergencyList	Comunicazionale	<p>I metodi della classe EmergencyList agiscono sui dati dei contatti di emergenza che appartengono alla stessa rubrica, ma con una restrizione specifica: la lista può contenere un numero massimo di 15 contatti. I dati dei contatti sono correlati perché fanno parte della stessa rubrica, ma la classe aggiunge una logica per gestire la quantità di contatti in relazione agli altri gruppi (rubrica principale, cestino).</p>
PhoneBookApp	Funzionale	<p>I metodi della classe PhoneBookApp sono legati alla funzionalità di avvio e gestione dell'interfaccia grafica. Il metodo start gestisce il caricamento della vista e la configurazione della finestra principale, mentre il metodo main avvia l'applicazione. Entrambi i metodi sono focalizzati sulla gestione</p>

		dell'applicazione e non interagiscono con altri componenti, quindi la coesione è funzionale.
FullGroupException	Funzionale	I metodi della classe FullGroupException sono progettati per gestire un'unica responsabilità: rappresentare un'eccezione che segnala quando una lista ha raggiunto il suo limite massimo e non può contenere altri elementi.
InvalidContactException	Funzionale	I metodi della classe InvalidContactException sono progettati per gestire un'eccezione specifica che si verifica quando un contatto non è valido, ad esempio quando manca il nome o il cognome durante la creazione o modifica del contatto.

5. Livelli di accoppiamento

Nome Classe 1	Nome Classe 2	Livello Accoppiamento	Descrizione
ContactBook	Contact	Dati	La classe ContactBook contiene riferimenti agli oggetti Contact senza memorizzare direttamente i loro dettagli (nome, numero, ecc.). Si occupa di gestire i contatti tramite questi riferimenti, ma non dipende dai dati interni di Contact .
PhoneBookController	ContactBook, EmergencyList, Bin	Dati	La classe PhoneBookController interagisce con ContactBook, EmergencyList e Bin passando dati relativi ai contatti (ad esempio, aggiungendo o rimuovendo contatti).
PhoneBookController	PhoneBookView	Controllo	La classe PhoneBookController interagisce con la PhoneBookView per gestire gli eventi UI (ad esempio, quando un utente preme un bottone per aggiungere un contatto). PhoneBookView invoca metodi in PhoneBookController per eseguire azioni.
PhoneBook	ContactBook, Bin, EmergencyList	Dati	La classe PhoneBook interagisce con ContactBook, Bin e EmergencyList . PhoneBook non dipende dai dettagli interni dei contatti, ma usa ContactBook come un'interfaccia per eseguire alcune operazioni.
Bin	ContactBook	Controllo	La classe Bin gestisce una lista di contatti rimossi da ContactBook e

			interagisce con ContactBook per ripristinare i contatti. Tuttavia, Bin non dipende dai dettagli interni di ContactBook , ma piuttosto utilizza metodi pubblici di ContactBook (ad esempio, per rimuovere o aggiungere contatti).
ContactBook	EmergencyList, Bin	Dati	La classe ContactBook può spostare contatti dalla rubrica principale (contactbook) al cestino o all'elenco di emergenza, senza dipendere dai dettagli interni delle due classi citate (Bin e EmergencyList).