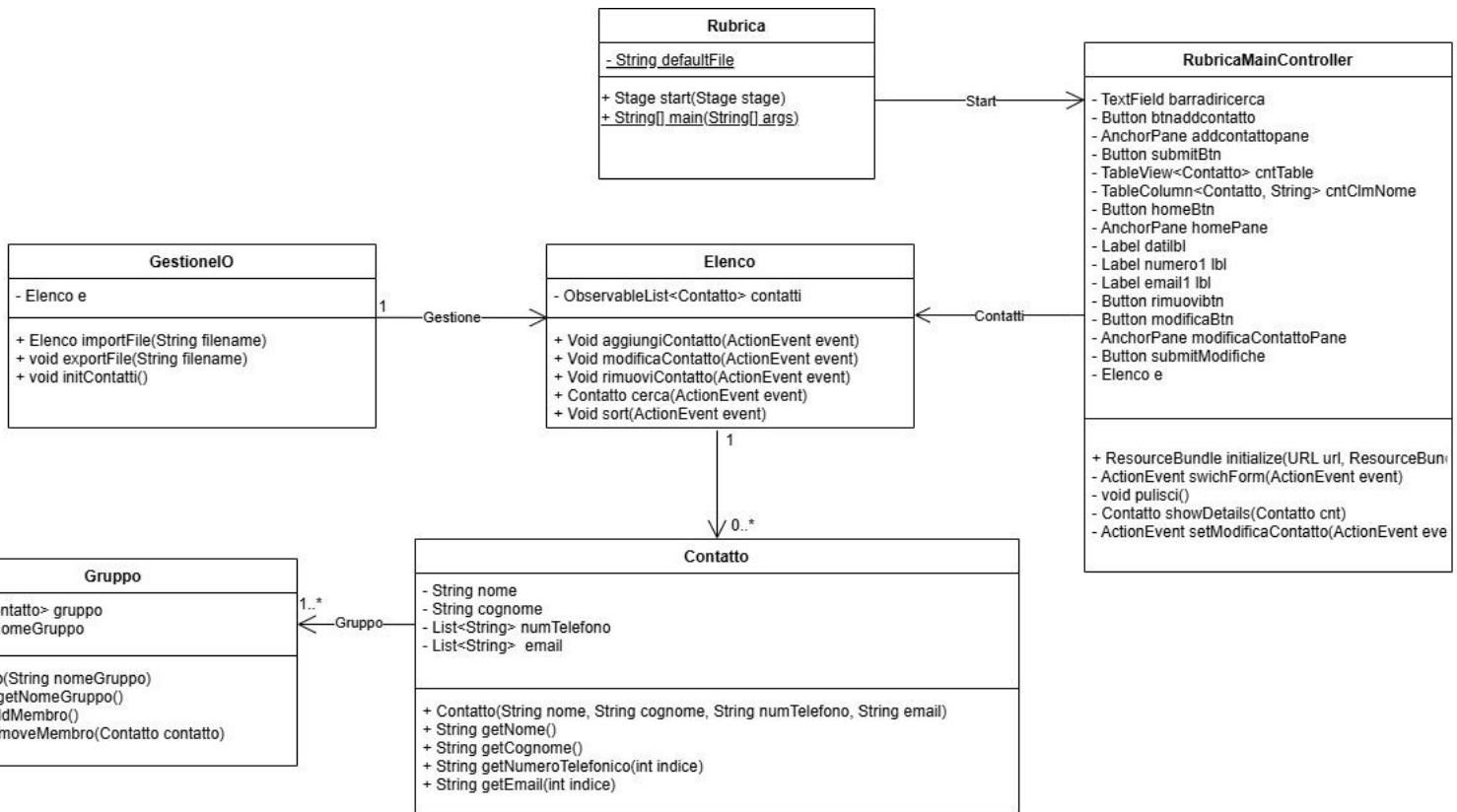


Diagramma delle classi:



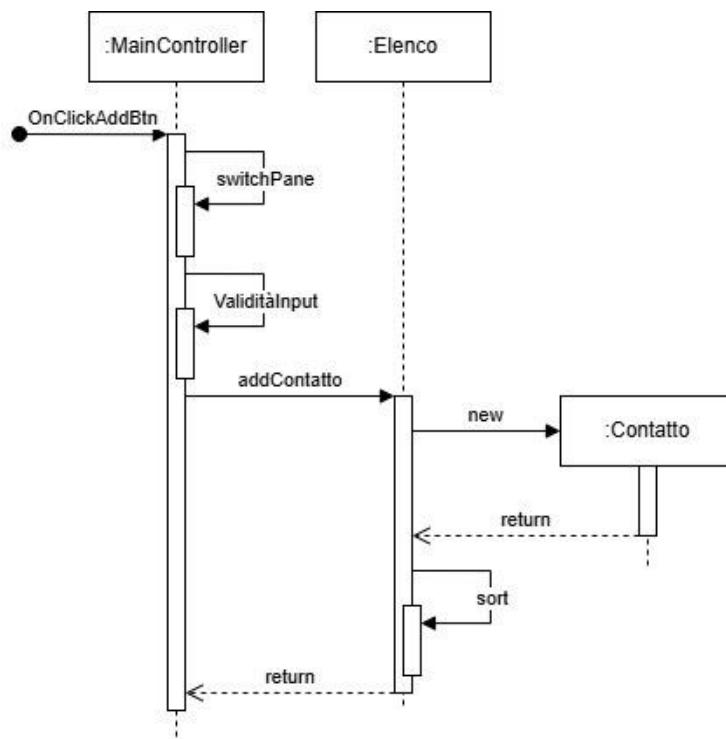
Coesione

- **Rubrica:**
Si occupa di avviare l'applicazione e contiene due metodi principali: start e main. Presenta un livello di **coesione Funzionale**, perché realizza un singolo compito.
- **GestioneIO :**
Gestisce esclusivamente la lettura, scrittura e inizializzazione dei contatti da file, rispetta il principio della singola responsabilità e ha un livello di **coesione Comunicazionale**, più funzionalità che lavorano sugli stessi dati.
- **Elenco:**
Si occupa della gestione dell'elenco dei contatti (aggiunta, modifica, ricerca, ecc.), ha un livello di **coesione Comunicazionale** perché esegue azioni che si riferiscono principalmente alla lista dei contatti presente nella classe (elenco).
- **RubricaMainController :**
Gestisce l'interfaccia grafica e collega gli eventi con i relativi metodi di gestione. Il livello di coesione presente è minore rispetto alle altre classi essendo la classe centrale del programma.
- **Gruppo:**
Si occupa della gestione di gruppi di contatti, con metodi dedicati per aggiungere e rimuovere membri, presenta un livello di **coesione Funzionale**.
- **Contatto:**
Rappresenta la classe Contatto e si occupa dei suoi attributi (nome, cognome, telefono, email), livello di **coesione Funzionale**.

Accoppiamento

- **GestioneIO e Elenco:**
Queste due classi sono accoppiate attraverso un'associazione (GestioneIO gestisce un'istanza di Elenco). **accoppiamento per Dati**, poiché GestioneIO usa la struttura Elenco per la persistenza dei dati.
- **Elenco e RubricaMainController :**
accoppiamento per Dati , Il controller interagisce con la lista dei contatti per aggiornare l'interfaccia grafica.
- **Gruppo e Contatto:**
accoppiamento per timbro , la classe Gruppo contiene una lista di contatti, però della struttura non usa tutti i suoi dati.

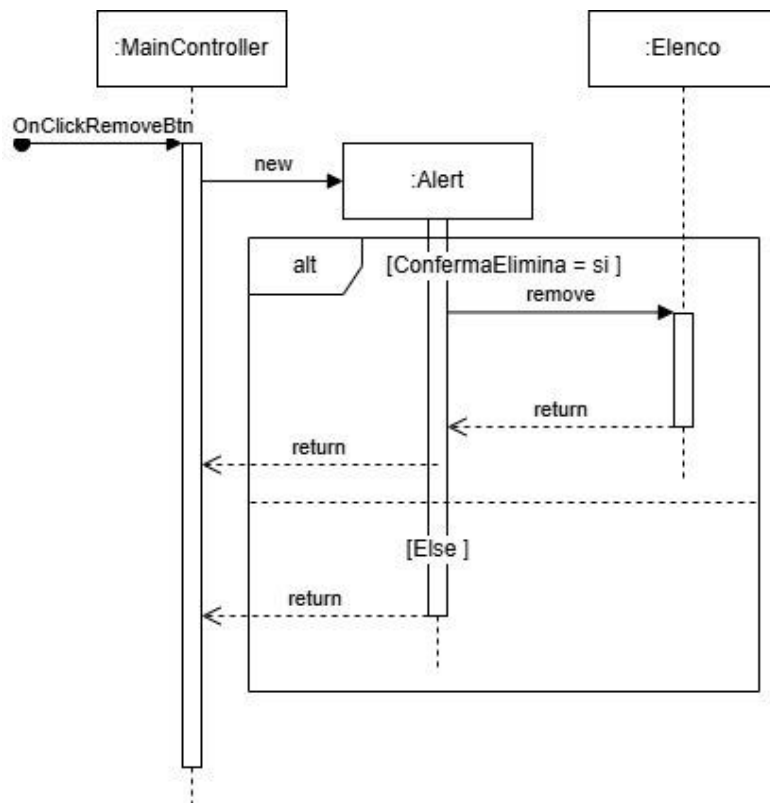
Diagrammi di sequenza:



Il diagramma rappresenta il flusso per l'**aggiunta di un nuovo contatto**, attivato tramite l'evento OnClickAddBtn.

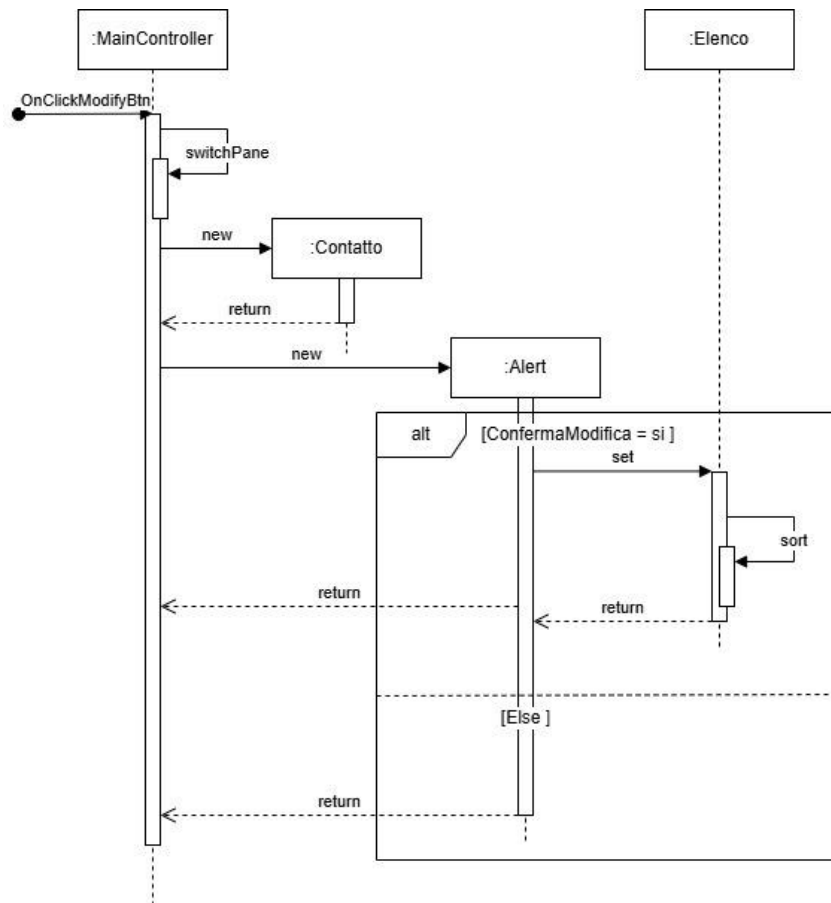
In seguito all'evento, il controller cambia interfaccia, controlla la validità dei dati inseriti e va ad aggiungere il contatto nell'elenco.

Infine ordina l'elenco aggiornato.



Il diagramma rappresenta il flusso per la **rimozione di un contatto esistente**, attivato tramite l'evento **OnClickRemoveBtn**.

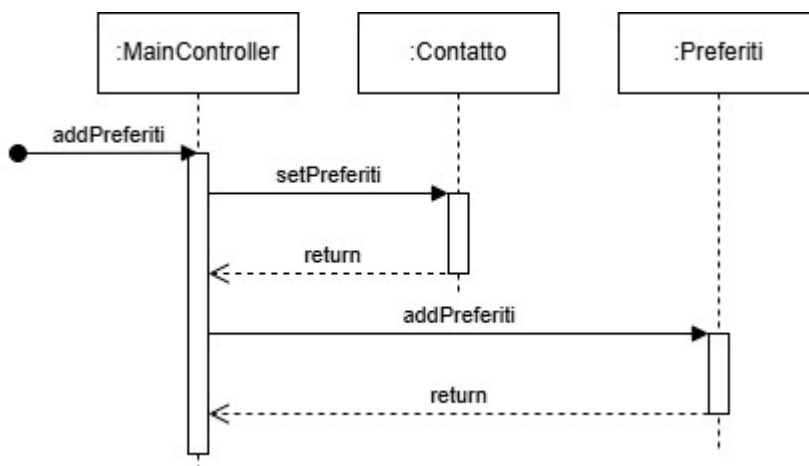
In seguito all'evento, crea un **Alert** per far confermare all'utente la rimozione del contatto, in caso di conferma viene rimosso il contatto dall'elenco



Il diagramma rappresenta il flusso per la **modifica di un contatto presente nell'elenco**, attivato tramite l'evento **OnClickModifyBtn**.

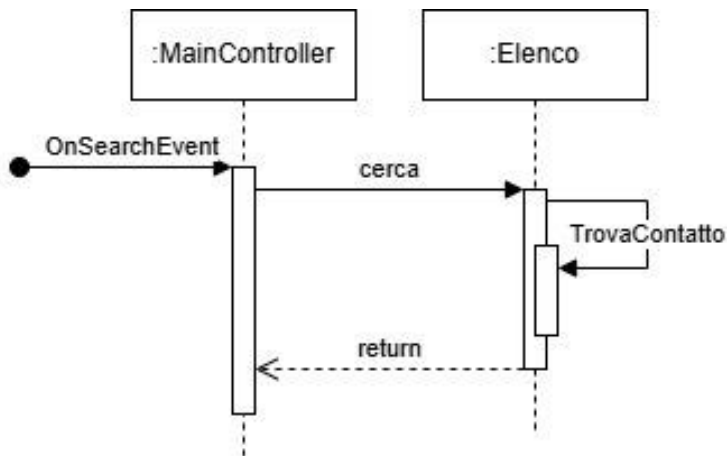
In seguito all'evento, viene cambiata l'interfaccia, vengono aggiornati i dati del contatto in seguito alla conferma di modifica da parte dell'utente.

Infine ordina l'elenco con i dati aggiornati.



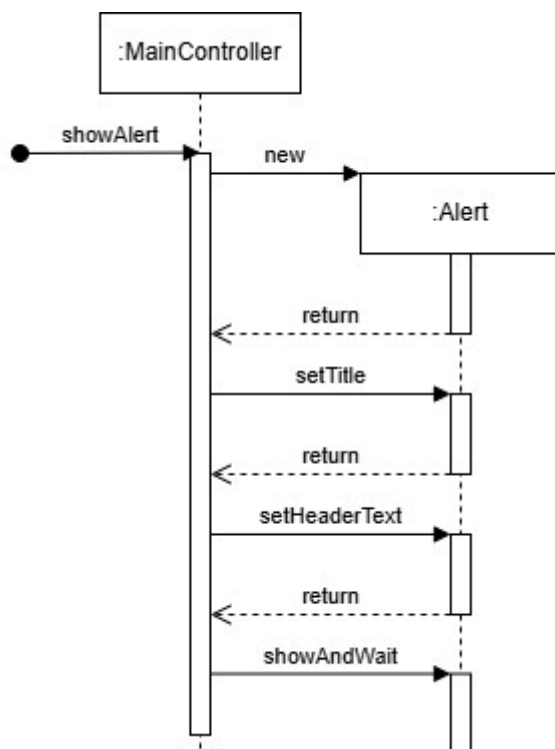
Il diagramma rappresenta il flusso per l'aggiunta **di un contatto al gruppo preferiti**, attivato tramite l'evento **AddPreferiti**.

In seguito all'evento, viene aggiunto un nuovo contatto al gruppo preferiti.



Il diagramma rappresenta il flusso per la **ricerca dei contatti**, attivato tramite l'evento OnSearchEvent.

In seguito all'evento, viene cercato il contatto all'interno dell'elenco



Il diagramma rappresenta il flusso per la **Gestione degli Alert**, attivato tramite l'evento showAlert.

In seguito all'evento, viene mostrata una finestra pop-up di Alert.