

Documentazione progettazione di dettaglio

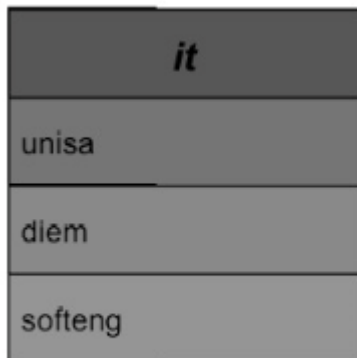
Indice:

Indice:	1
1. Diagramma dei package.....	3
2. Diagramma delle classi.....	4
3. Diagrammi di sequenza.....	7
a. AGGIUNGI	7
a.1 COMUNICAZIONE TRA IL GESTORE E LA SUA IMPLEMENTAZIONE	7
b. MODIFICA.....	8
b.1. COMUNICAZIONE TRA IL GESTORE E LA SUA IMPLEMENTAZIONE	8
c. RIMUOVI.....	9
d. IMPORTA.....	10
d.1 COMUNICAZIONE TRA IL GESTORE E LA SUA IMPLEMENTAZIONE	10
e. ESPORTA	11
f. RESET RUBRICA.....	12
g. RICERCA	13
g.1 RICERCA (CONTATTO TROVATO)	13
g.2 RICERCA (CONTATTO NON TROVATO).....	13
g.1.1 COMUNICAZIONE TRA IL GESTORE E LA SUA IMPLEMENTAZIONE (STRINGA NON VUOTA).....	14
g.2.1 COMUNICAZIONE TRA IL GESTORE E LA SUA IMPLEMENTAZIONE (STRINGA VUOTA) ..	14
4. Codici UML Diagrammi di Sequenza.....	15
a. AGGIUNGI	15
a.1. Comunicazione Gestore-Implementazione (Aggiungi)	16
b. MODIFICA.....	16
b.1. Comunicazione Gestore-Implementazione (Modifica)	17
c. RIMUOVI.....	18
d. IMPORTA.....	18
d.1. Comunicazione Gestore-Implementazione (Importa)	19
e. ESPORTA	20
e.1. Comunicazione Gestore-Implementazione (Esporta)	20
f. RESET	21
g. RICERCA.....	22

g.1. CERCA (CONTATTO TROVATO)	22
g.2. CERCA (CONTATTO NON TROVATO)	23
g.1.1 Comunicazione Gestore-Implementazione (Stringa non Vuota).....	23
g.2.1 Comunicazione Gestore-Implementazione (Stringa Vuota)	24

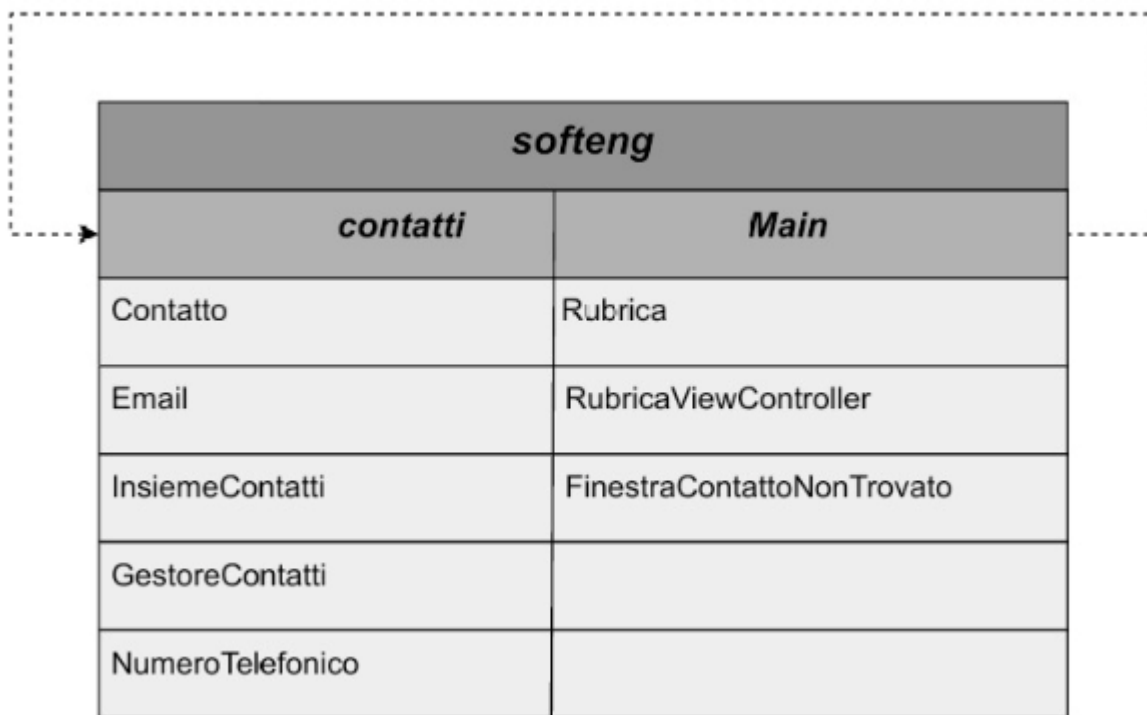
1. Diagramma dei package

VISTA GENERALE

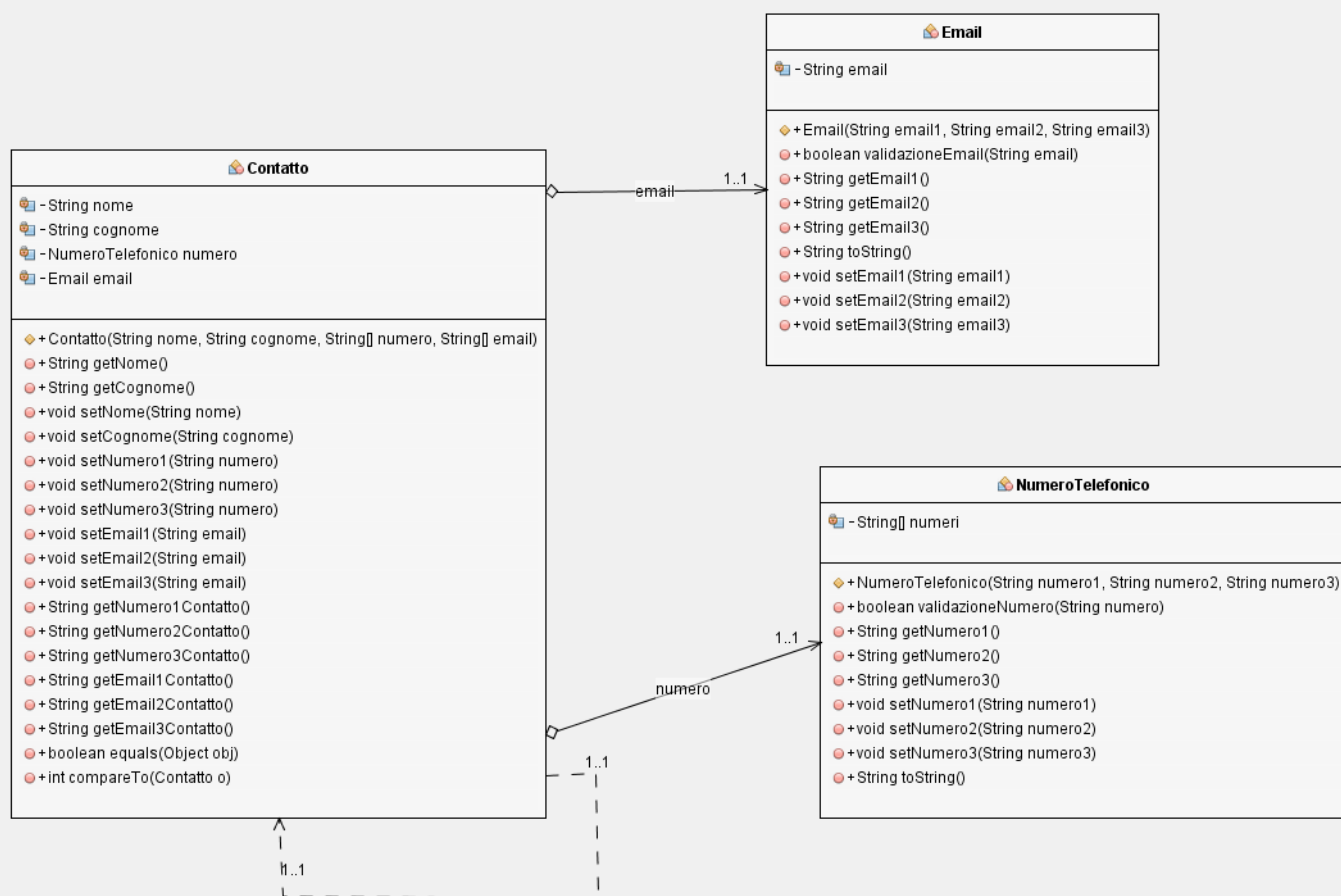


VISTA IN DETTAGLIO

con dipendenze



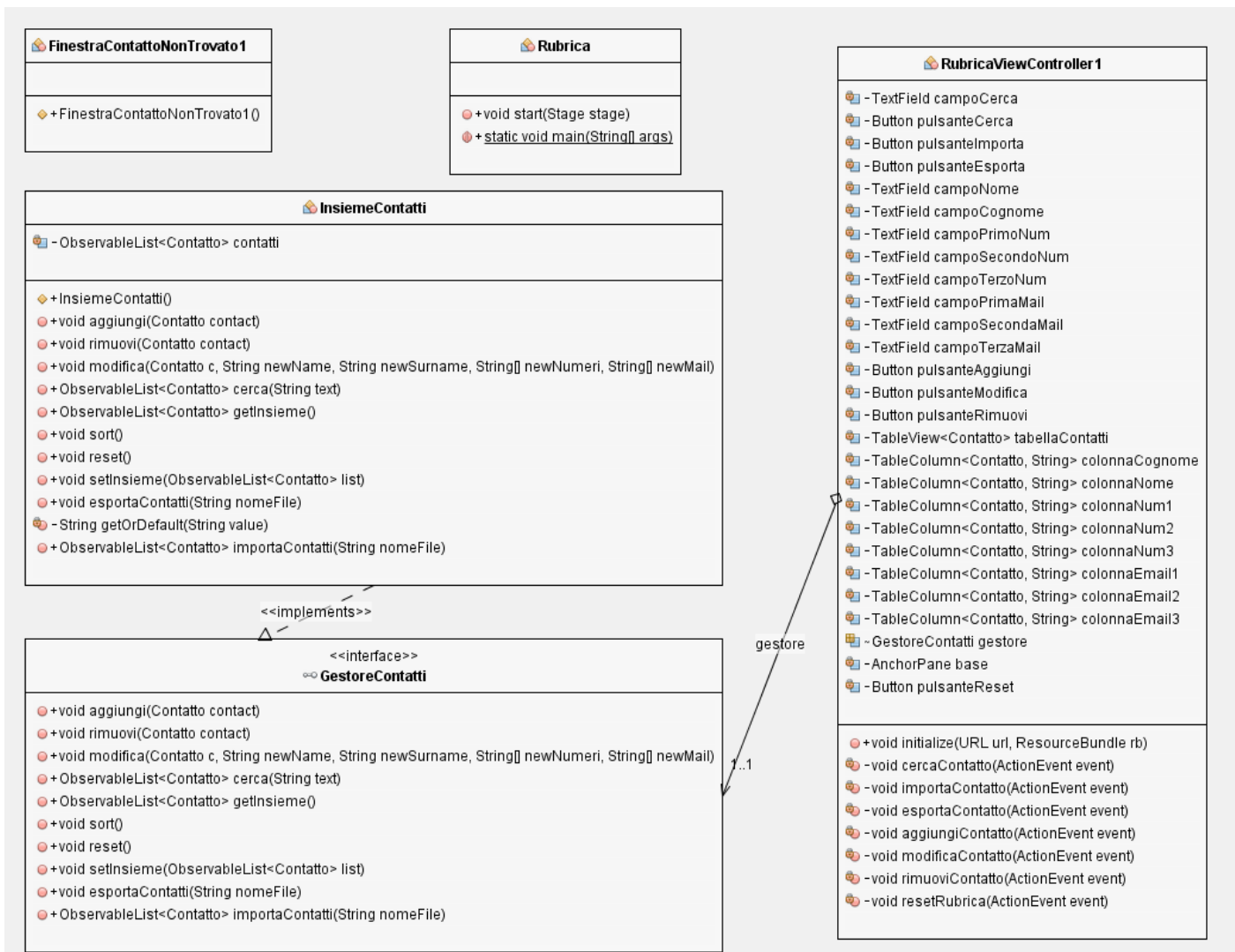
2. Diagramma delle classi



Al fine di sfruttare i principi della programmazione orientata agli oggetti, ponendo particolare attenzione a coesione ed accoppiamento si è deciso di dividere il problema in diverse classi, ciascuna con una responsabilità ben definita:

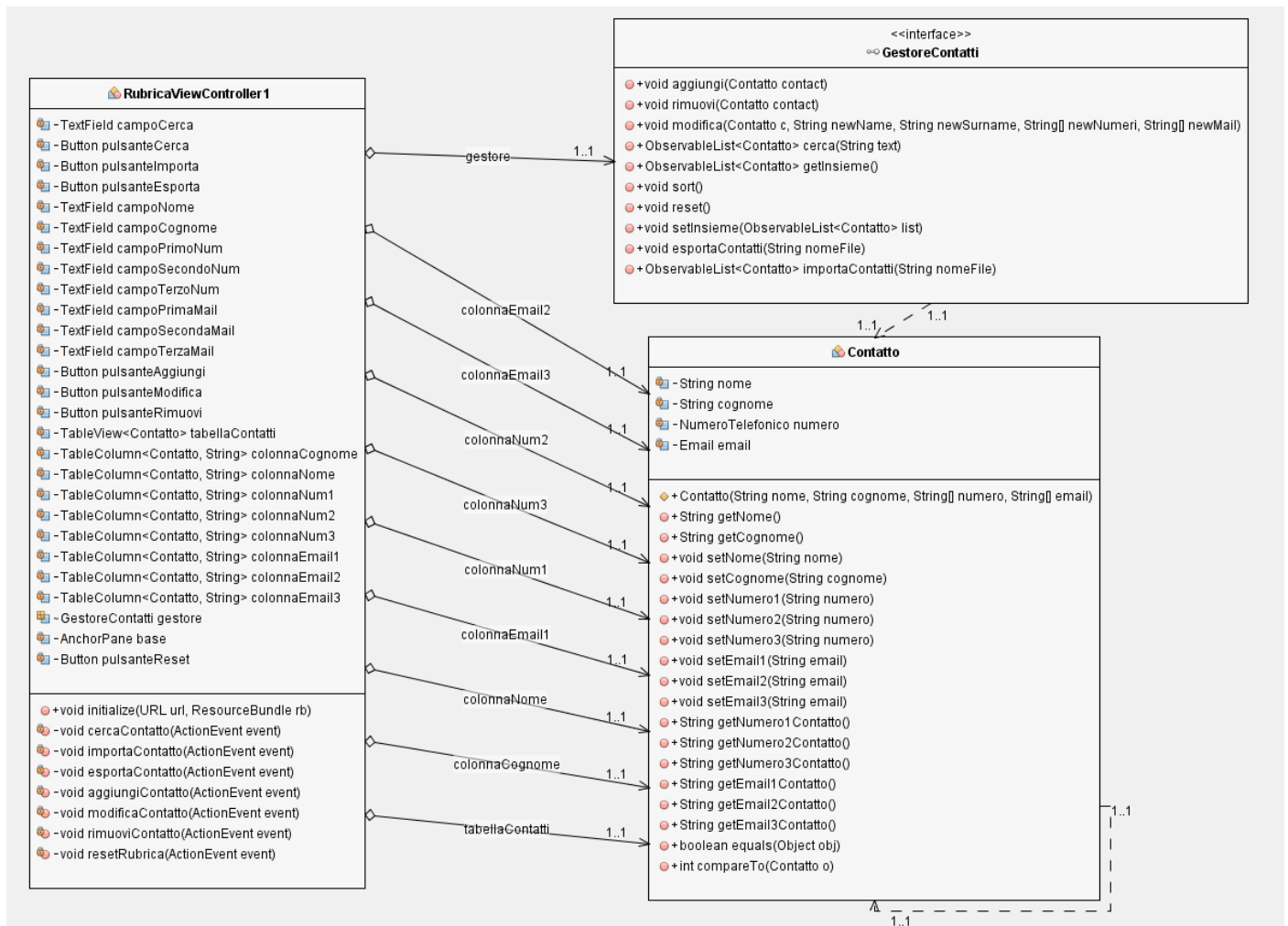
- Classe contatto:** La classe rappresenta l'entità principale del sistema: il contatto, i metodi implementati e gli attributi sono legati alla sua gestione diretta. L'utilizzo di altre entità come **Email** e **NumeroTelefonico** permette di delegare la gestione dettagliata di alcuni attributi (*email e numeri di telefono*) ad altre classi, aumentando la coesione.
 La classe contatto comunica con le classi NumeroTelefonico ed Email solo mediante i loro metodi pubblici (*accoppiamento per dati*).
 La classe si occupa di aggregare gli elementi contenuti in NumeroTelefonico e Email ma lascia la gestione degli elementi alle classi ad essi dedicate.
- Classe NumeroTelefonico:** La classe è responsabile della gestione di un insieme di numeri telefonici (*massimo tre numeri*), il fatto di essere una classe separata da contatto permette di tenere la coesione alta e consente, in futuro, di poter anche implementare metodi per logiche di validazione. La classe non dipende da alcuna altra classe (*nessun accoppiamento*).
- Classe E-mail:** Analogamente a come avviene per i numeri, questa classe si occupa della gestione di un insieme di indirizzi E-mail (*massimo tre indirizzi e-mail*). Le stesse considerazioni sulla coesione e sulle future logiche di validazione implementabili valgono anche per questa classe. Analogamente a

come avviene per la classe NumeroTelefonico, anche questa classe è indipendente dalle altre(*accoppiamento per dati*).



- **Interfaccia GestoreContatti:** L'interfaccia permette di eliminare la rigida dipendenza che normalmente esisterebbe tra la classe **InsiemeContatti** e **RubricaViewController**, utilizzando la tecnica dell'inversione della dipendenza; in questo modo sarà possibile sostituire in futuro la collezione di contatti con, ad esempio, un database o un'altra struttura che implementi questa interfaccia. I metodi presenti hanno lo scopo di definire funzionalità di base che un qualsiasi "contenitore" di Contatti deve avere.
- **Classe InsiemeContatti:** La classe rappresenta un'implementazione dell'interfaccia **GestoreContatti**, funge da "contenitore" per il generico oggetto Contatto. Implementa i metodi dell'interfaccia permettendo alla struttura di comunicare con il controller. La classe conterrà diversi Contatti e richiamerà i soli metodi pubblici per interagire con essi (*accoppiamento per dati*).
- **Classe Rubrica:** La classe rappresenta il Main, permette di caricare la View inserita in un file fxml e di avviare la GUI. Risulta totalmente indipendente da tutte le classi elencate in precedenza seguendo il principio di responsabilità unica.
- **Classe FinestraContattoNonTrovato:** La classe permette di gestire un messaggio pop-up che viene mostrato solo a seguito di una ricerca che non trova alcuna corrispondenza. Non utilizza un file .fxml

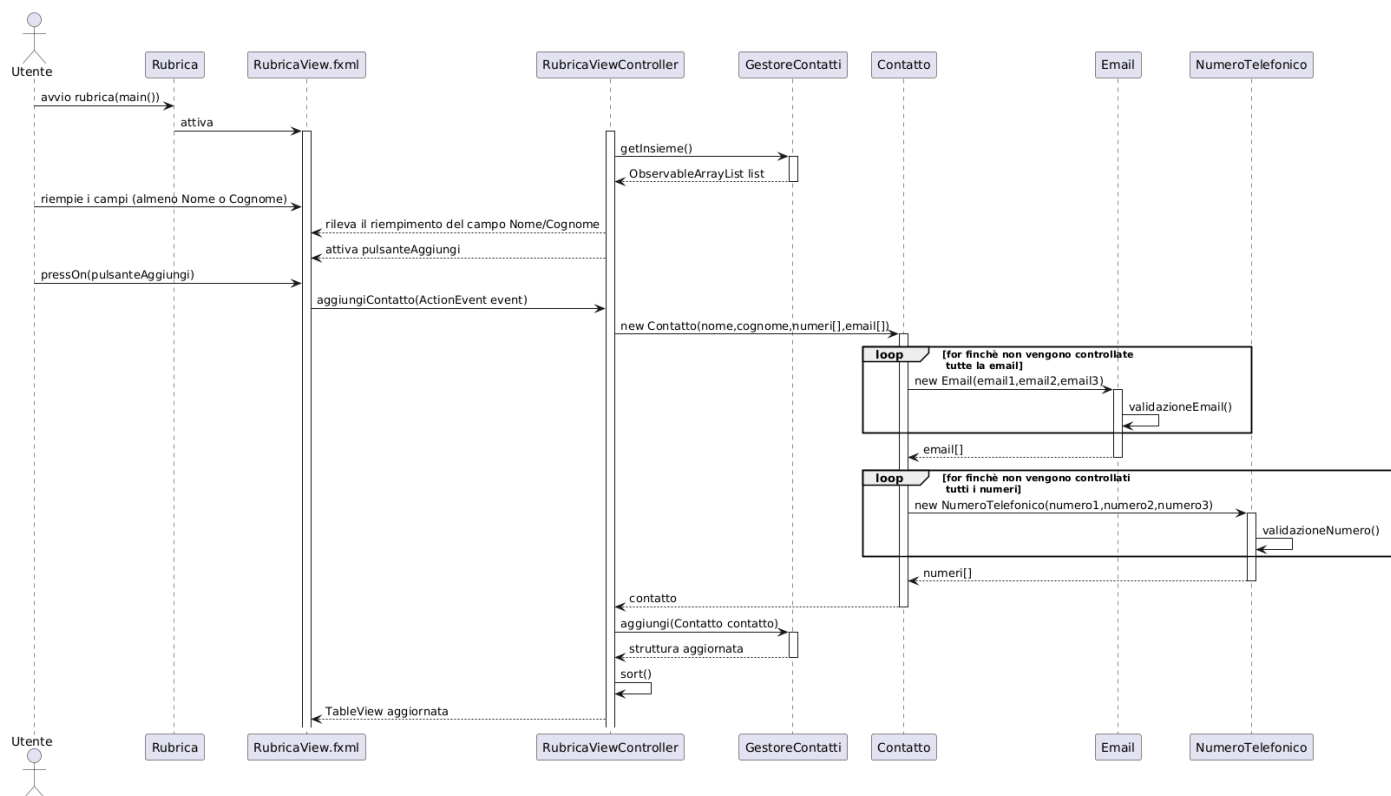
per la sua interfaccia, in quanto essendo quest'ultima molto semplice, risulta essere facilmente implementabile nella classe stessa.



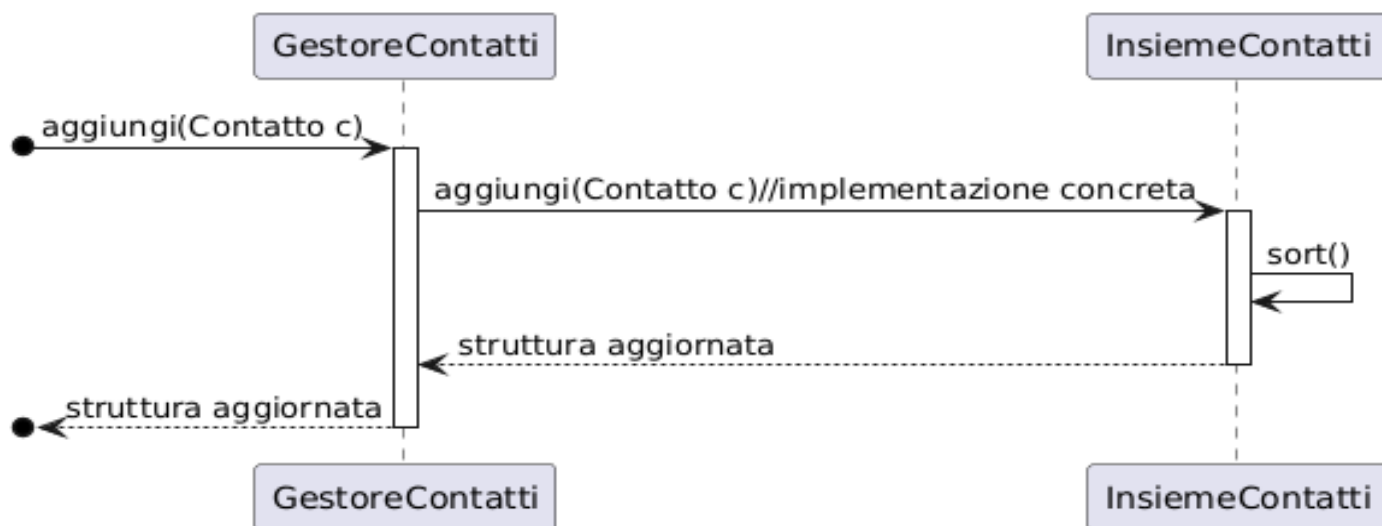
- Classe RubricaViewController:** La classe rappresenta il mezzo attraverso il quale le implementazioni della struttura sono in grado di comunicare con gli elementi grafici, gestendo anche le azioni degli utenti (come il clic sui pulsanti). Questa classe comunicherà con l'interfaccia **GestoreContatti** (la classe Controller non gestirà direttamente i contatti, ma delega questa responsabilità), richiamando i metodi associati a ciascun pulsante inserito nel file. fxml. Le interazioni con la classe Contatti saranno dettate dai soli metodi pubblici (accoppiamento per dati), anche l'interazione con il GestoreContatti sarà guidata dai metodi messi a disposizione.

3. Diagrammi di sequenza

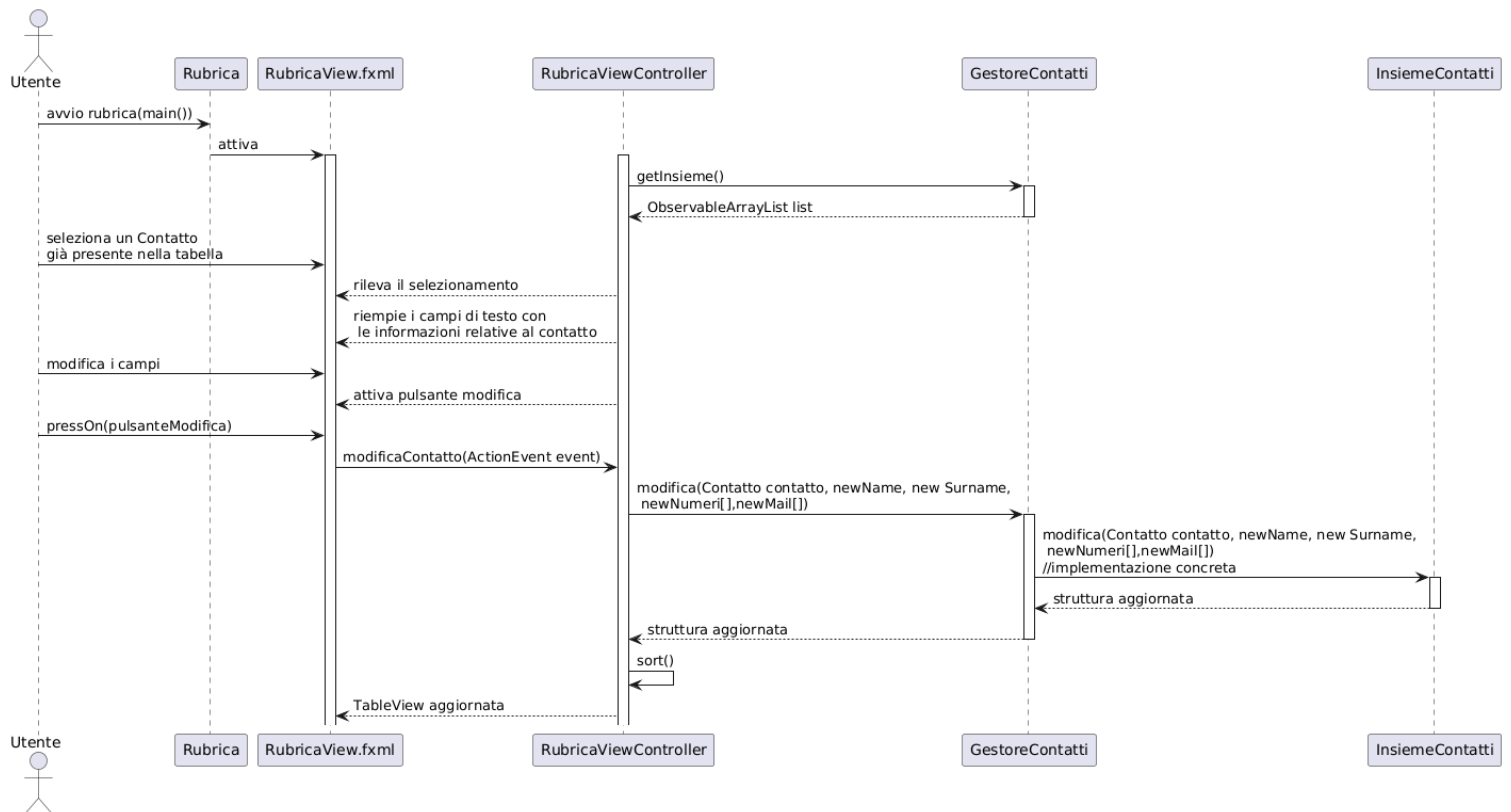
a. AGGIUNGI



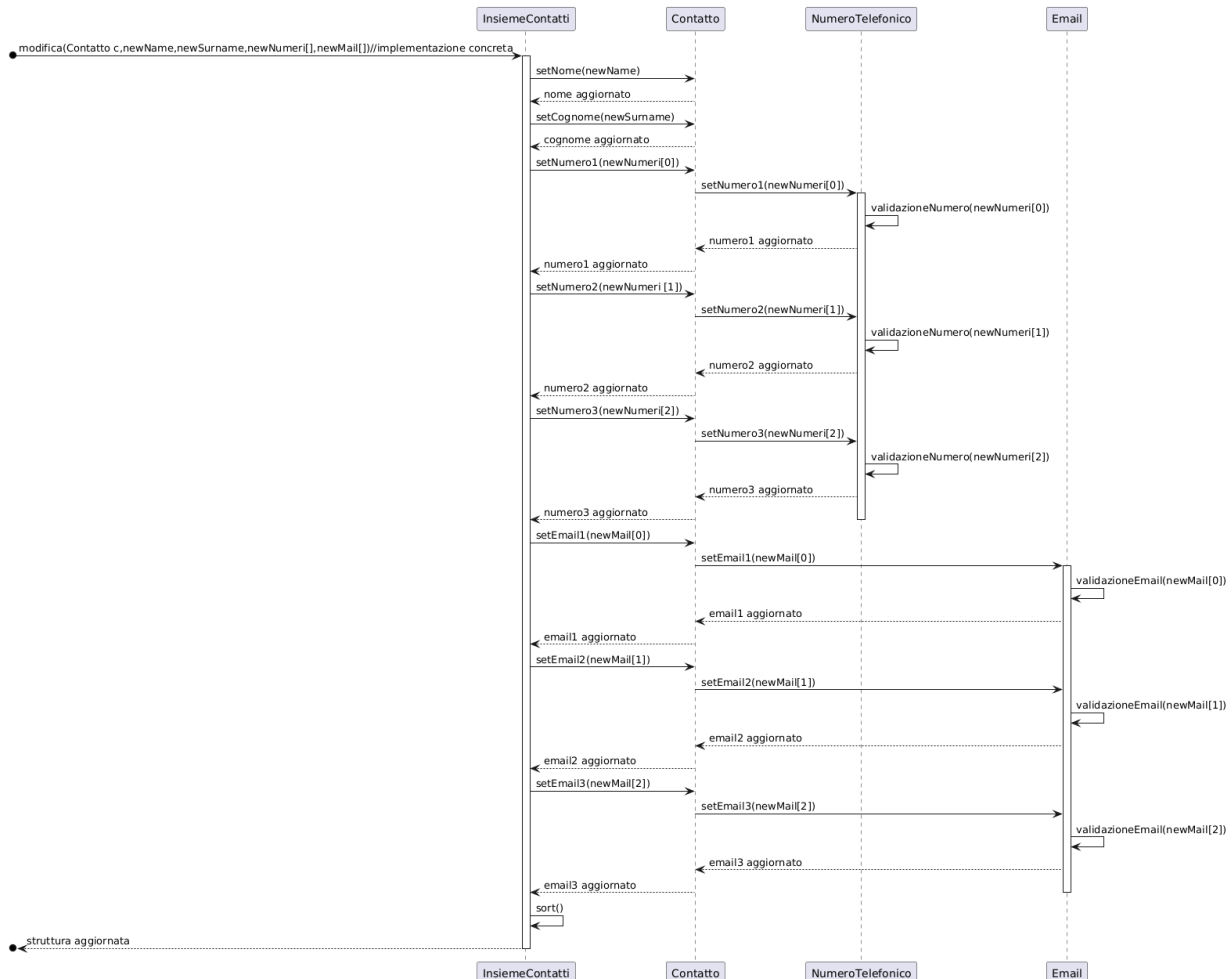
a.1 COMUNICAZIONE TRA IL GESTORE E LA SUA IMPLEMENTAZIONE



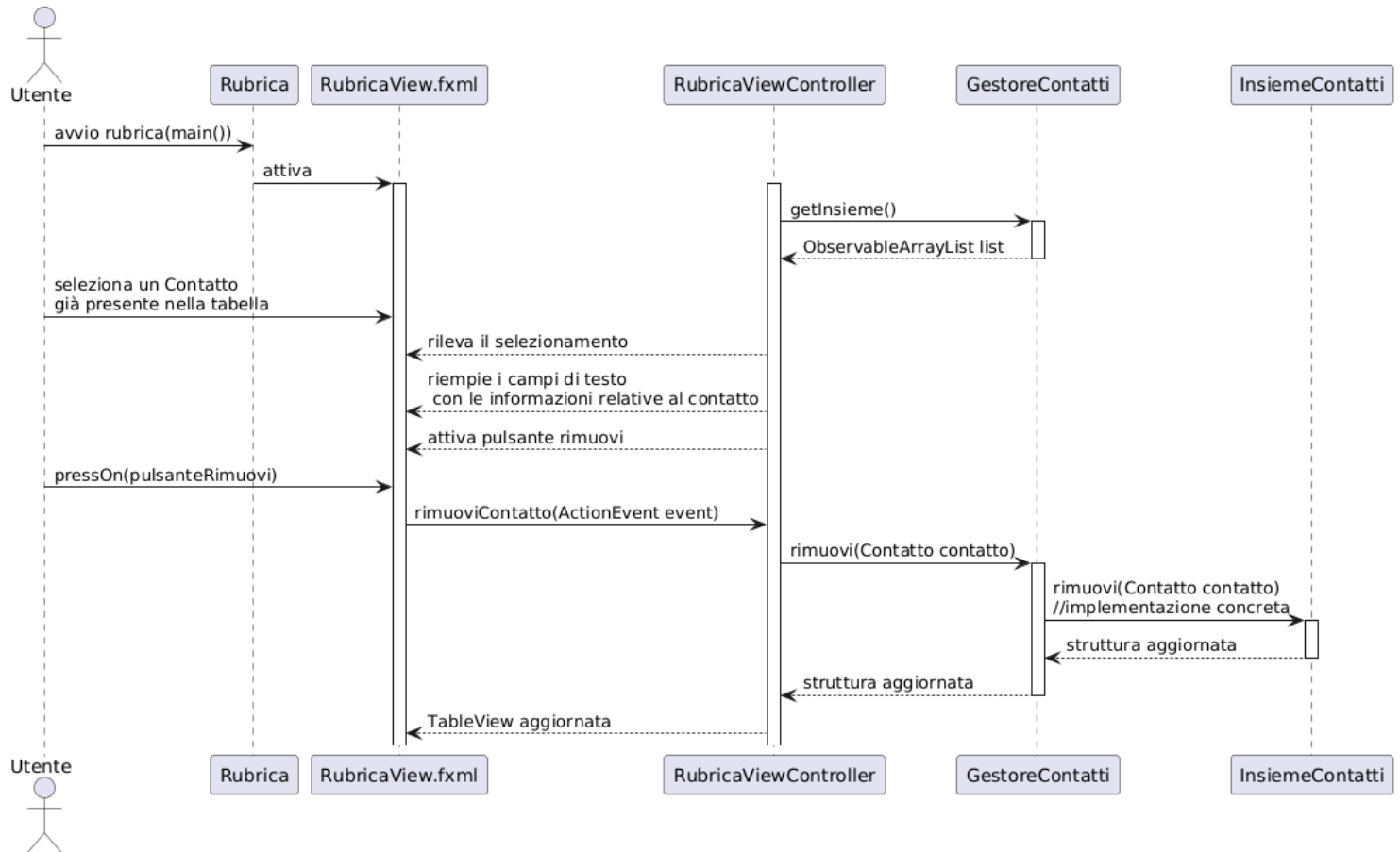
b. MODIFICA



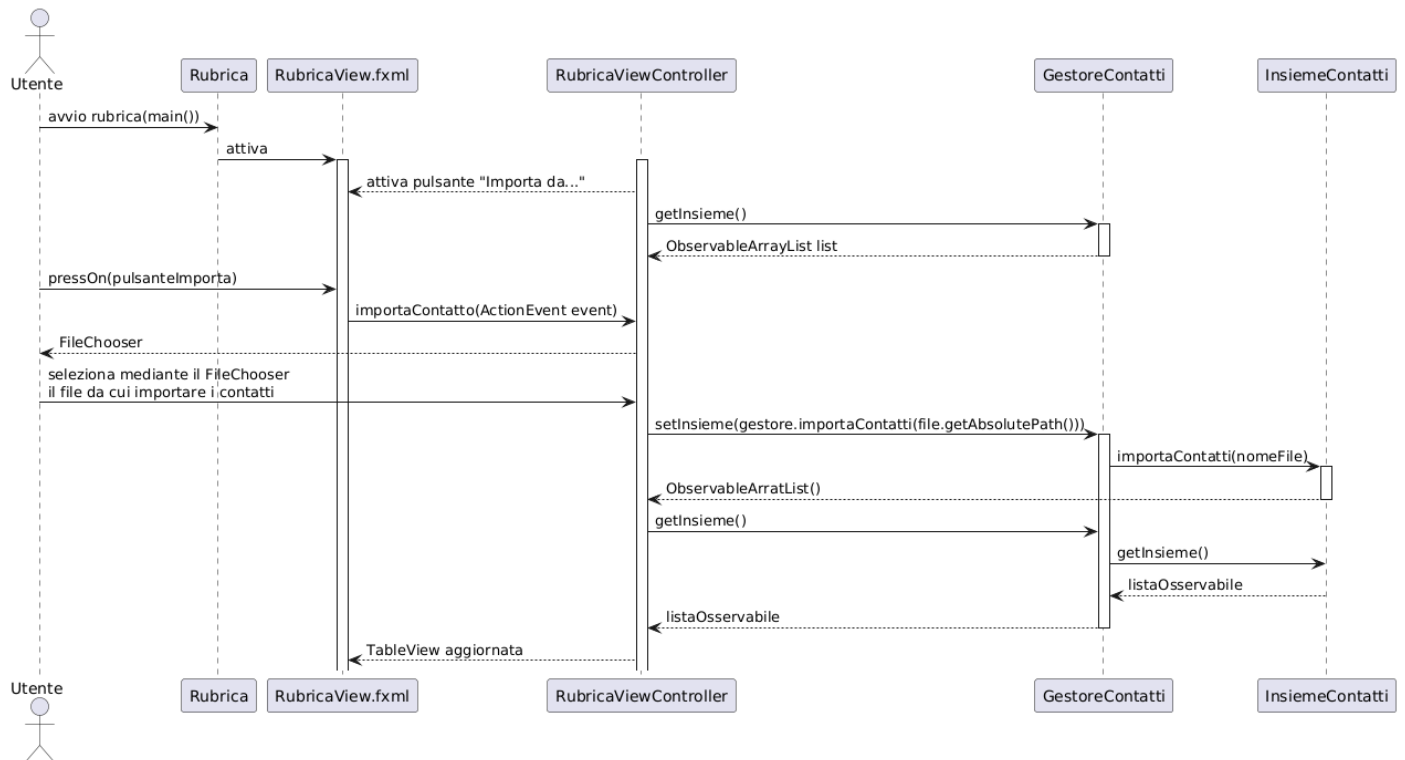
b.1. COMUNICAZIONE TRA IL GESTORE E LA SUA IMPLEMENTAZIONE



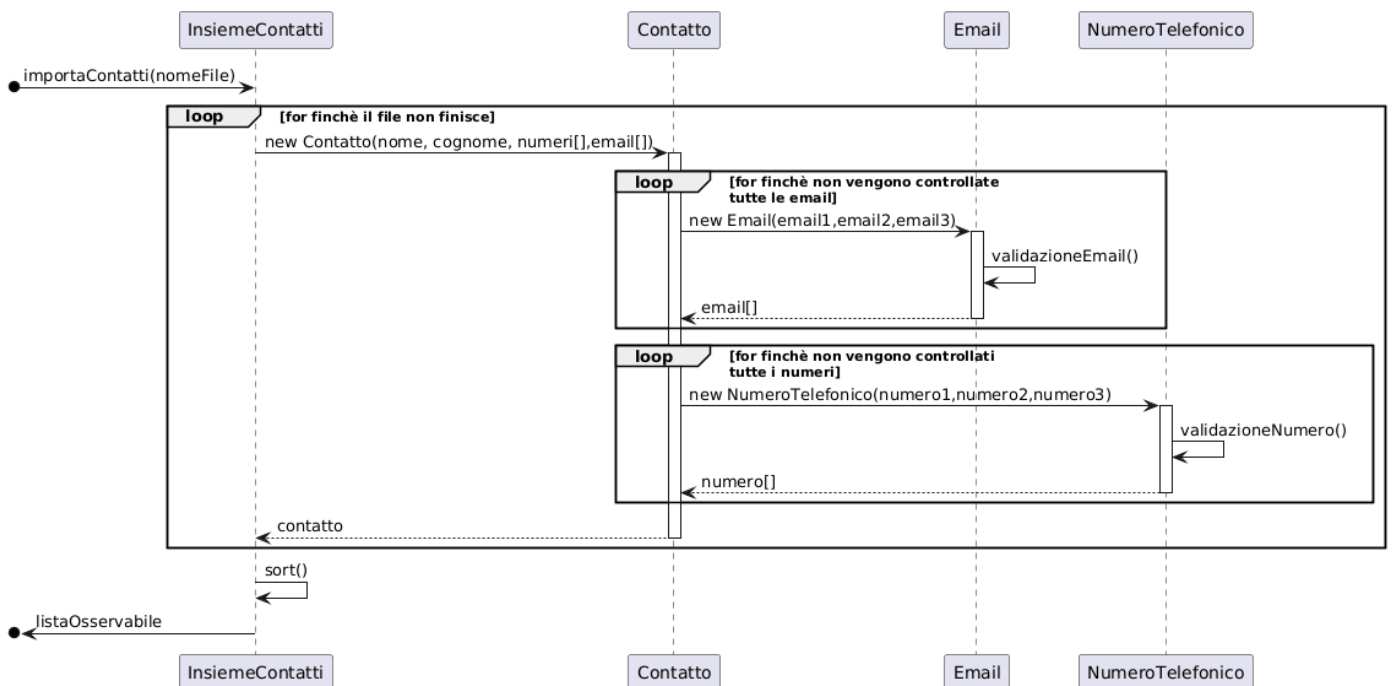
c. RIMUOVI



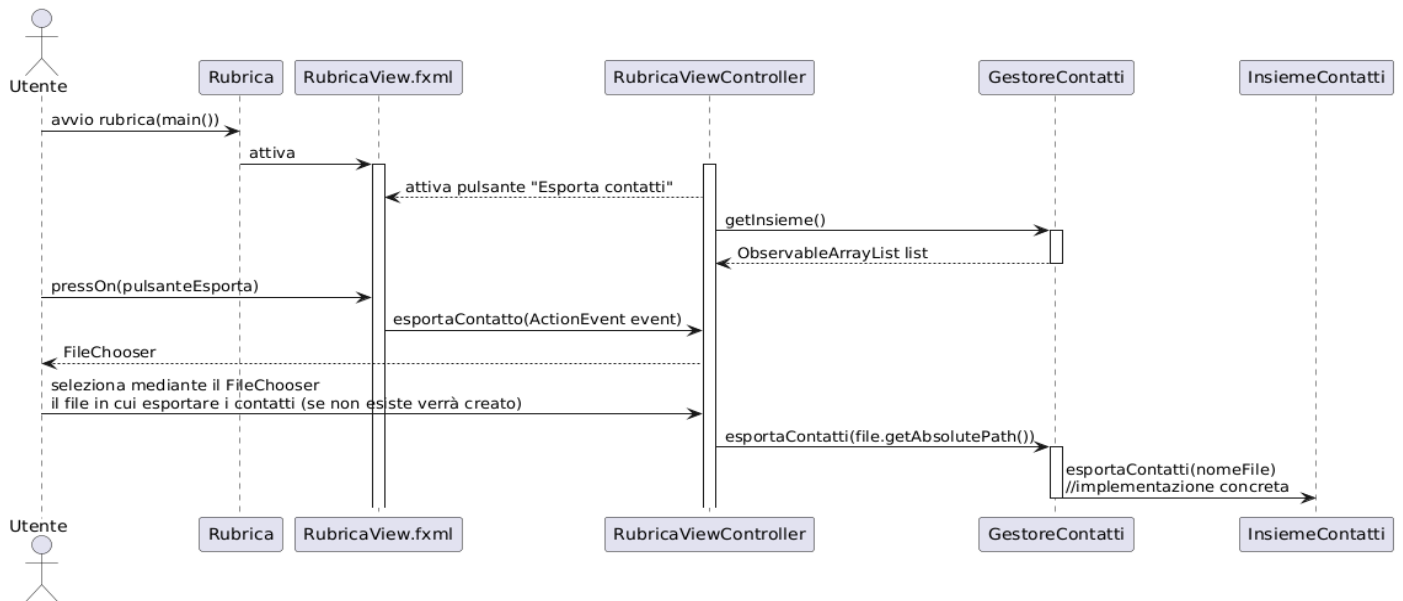
d. IMPORTA



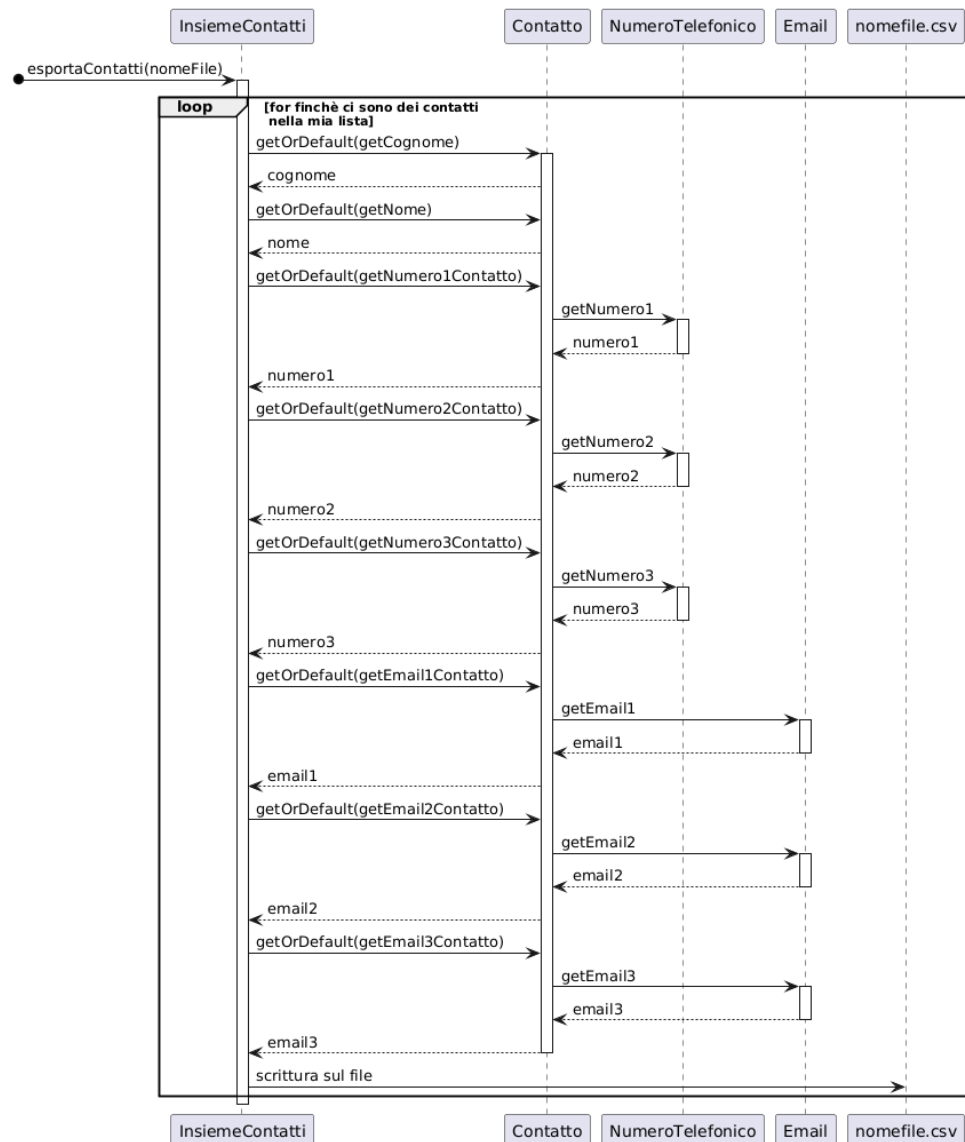
d.1 COMUNICAZIONE TRA IL GESTORE E LA SUA IMPLEMENTAZIONE



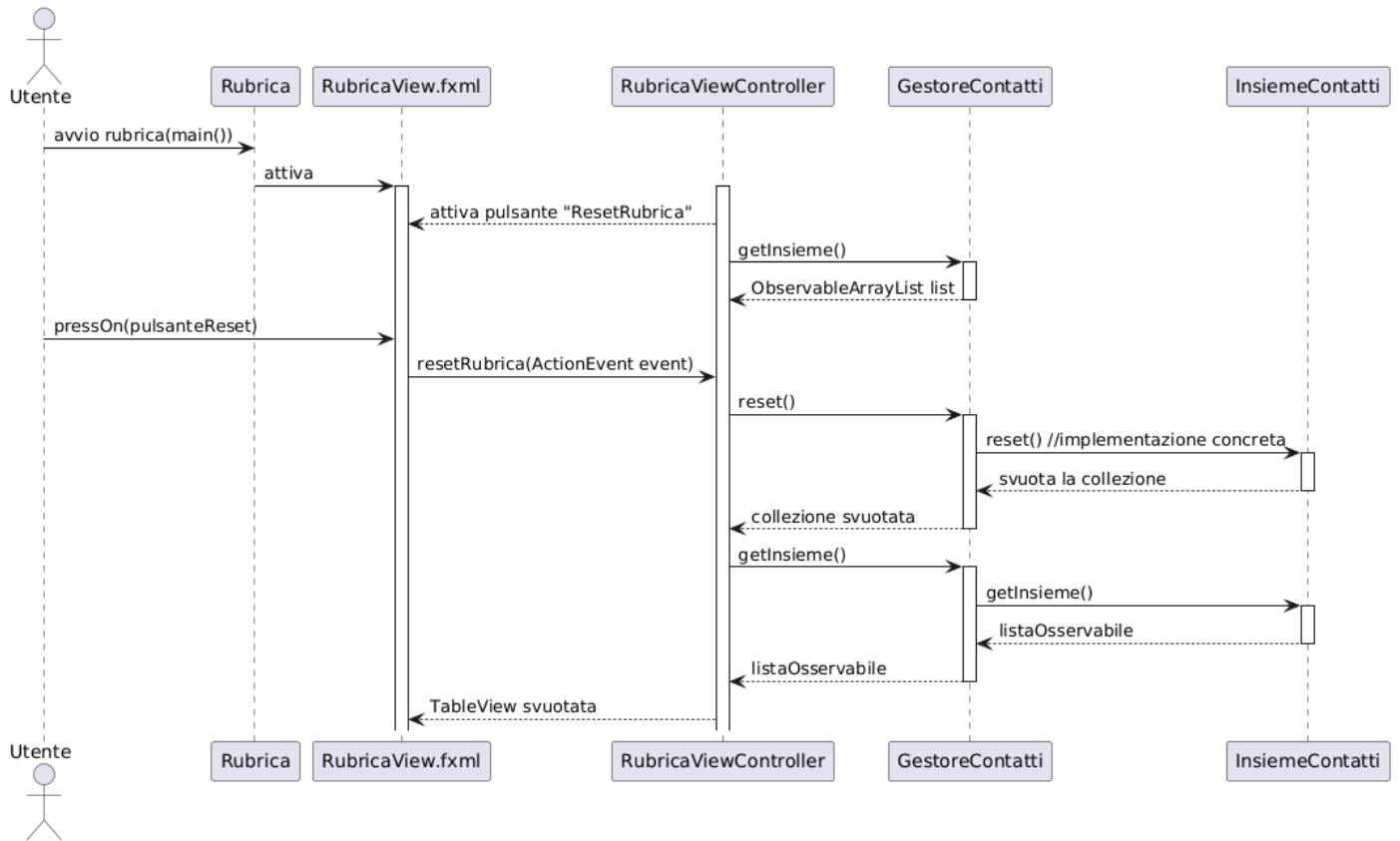
e. ESPORTA



e.1 COMUNICAZIONE TRA INSIEME CONTATTI E CONTATTO

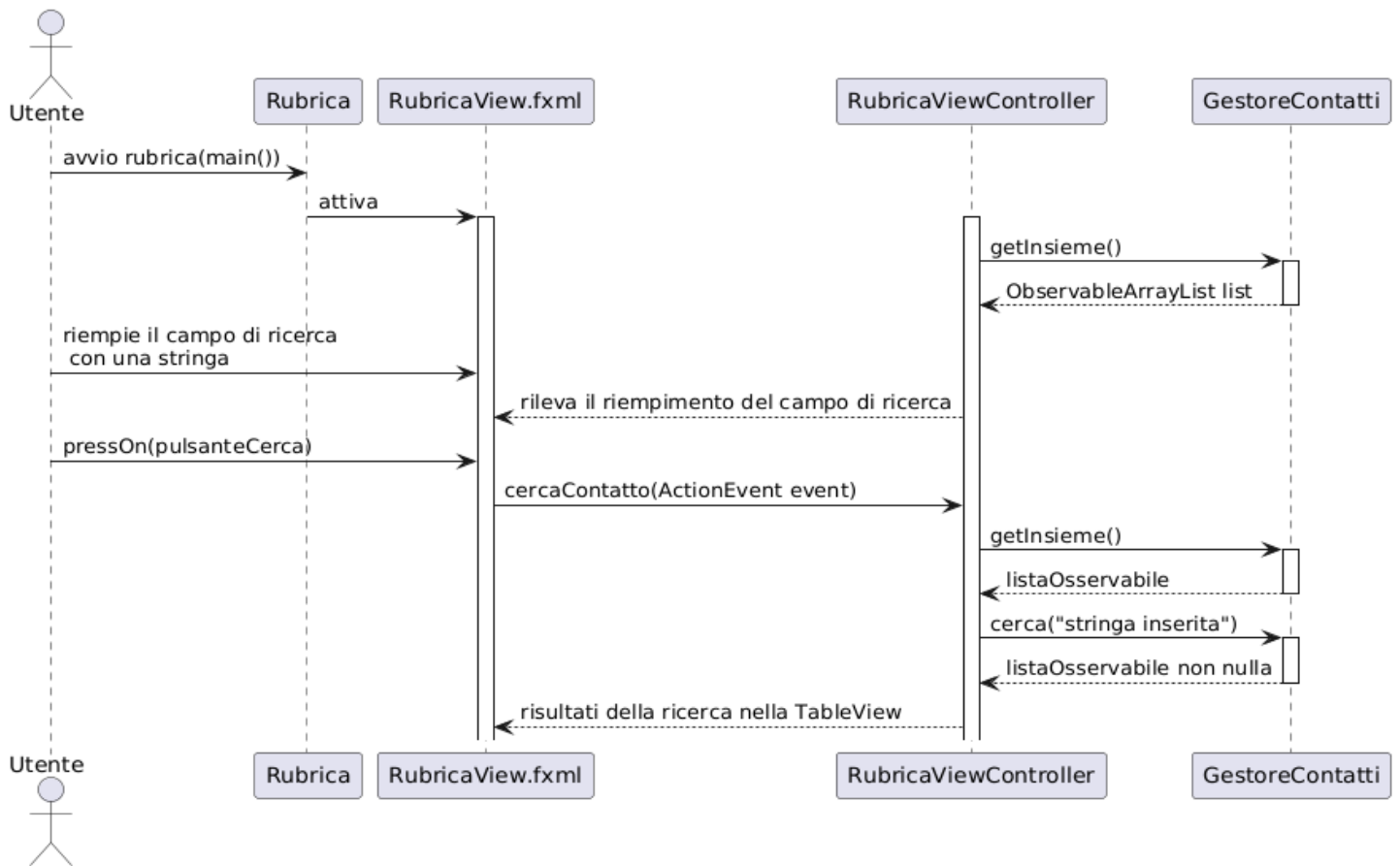


f. RESET RUBRICA

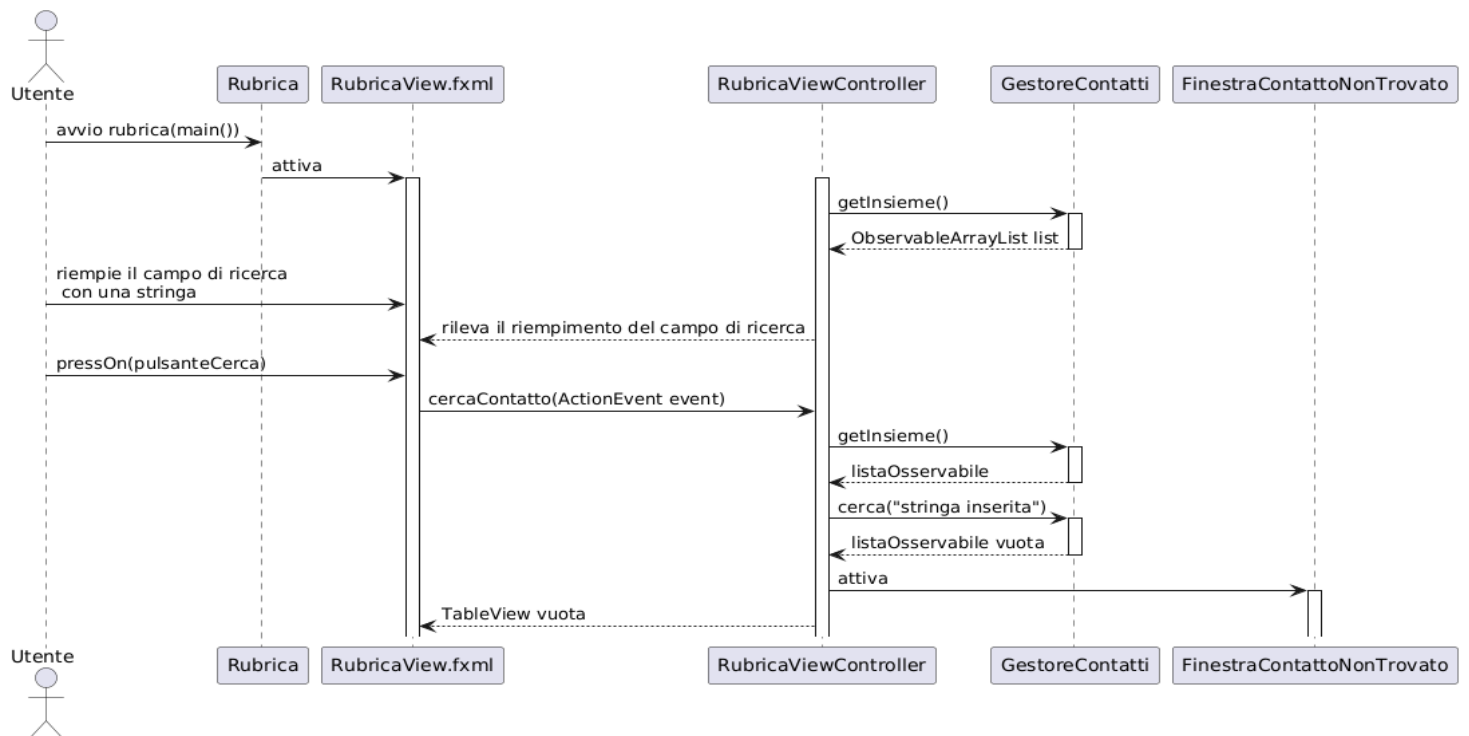


g. RICERCA

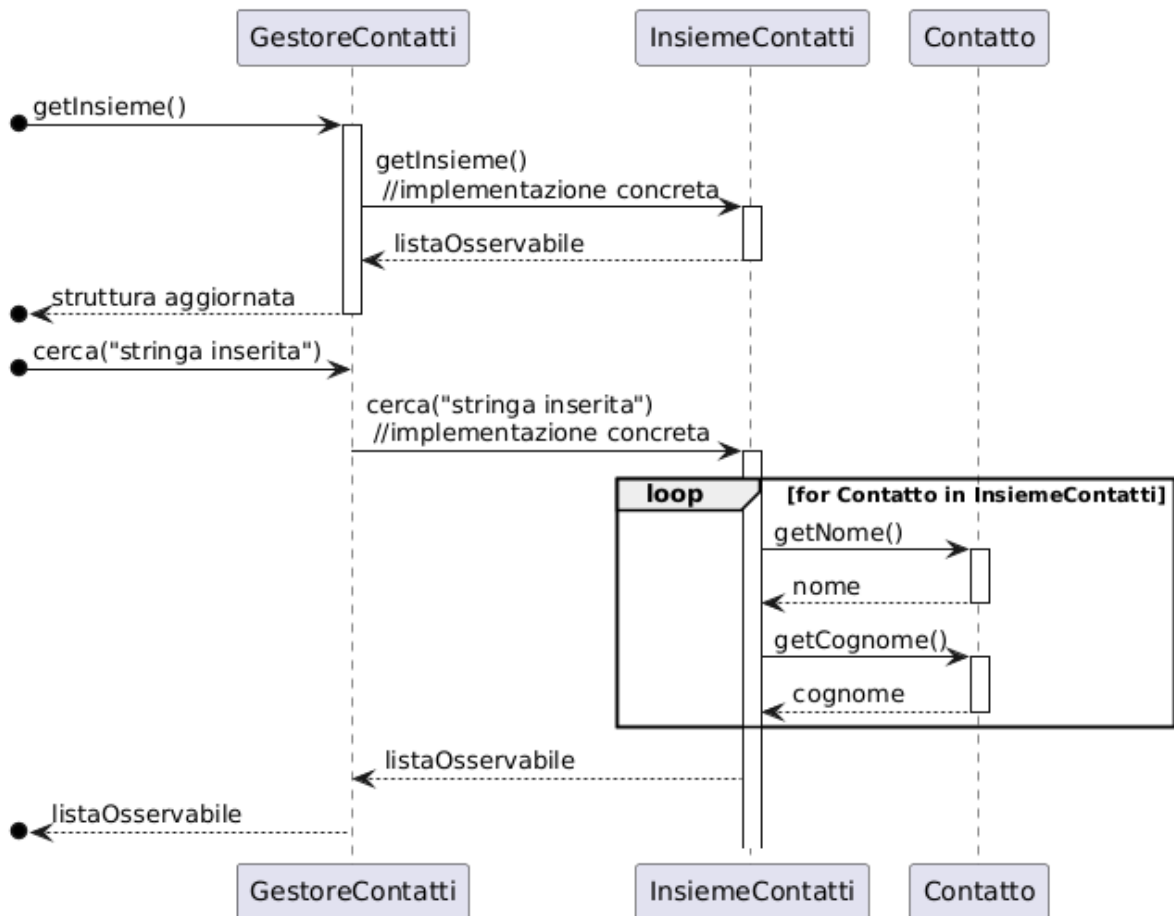
g.1 RICERCA (CONTATTO TROVATO)



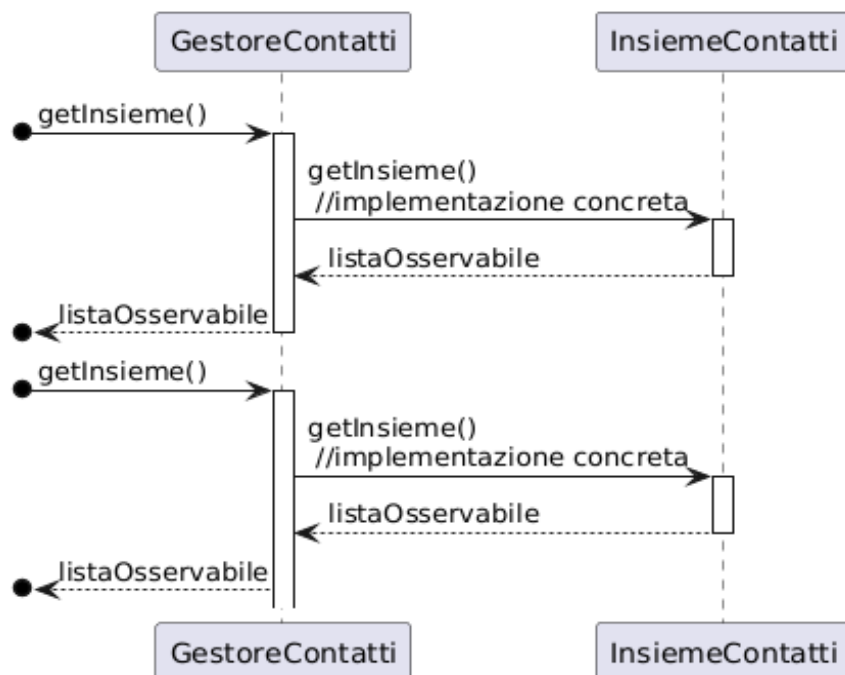
g.2 RICERCA (CONTATTO NON TROVATO)



g.1.1 COMUNICAZIONE TRA IL GESTORE E LA SUA IMPLEMENTAZIONE (STRINGA NON VUOTA)



g.2.1 COMUNICAZIONE TRA IL GESTORE E LA SUA IMPLEMENTAZIONE (STRINGA VUOTA)



4. Codici UML Diagrammi di Sequenza

a. AGGIUNGI

```
@startuml
actor Utente
Utente->>Rubrica: avvio rubrica(main())
Rubrica->>RubricaView.fxml: attiva
activate RubricaView.fxml
activate RubricaViewController
RubricaViewController->>GestoreContatti: getInsieme()
activate GestoreContatti
GestoreContatti-->>RubricaViewController: ObservableArrayList list
deactivate GestoreContatti
Utente->>RubricaView.fxml: riempie i campi (almeno Nome o Cognome)
RubricaViewController-->>RubricaView.fxml: rileva il riempimento del campo Nome/Cognome
RubricaViewController-->>RubricaView.fxml: attiva pulsanteAggiungi
Utente->>RubricaView.fxml: pressOn(pulsanteAggiungi)
RubricaView.fxml->>RubricaViewController: aggiungiContatto(ActionEvent event)
RubricaViewController->>Contatto: new Contatto(nome,cognome,numeri[],email[])
activate Contatto
loop for finchè non vengono controllate\n tutte la email
Contatto->>Email: new Email(email1,email2,email3)
activate Email
Email->>Email: validazioneEmail()
end
Email-->>Contatto: email[]
deactivate Email
loop for finchè non vengono controllati\n tutti i numeri
Contatto->>NumeroTelefonico: new NumeroTelefonico(numero1,numero2,numero3)
activate NumeroTelefonico
NumeroTelefonico->>NumeroTelefonico: validazioneNumero()
end
NumeroTelefonico-->>Contatto: numeri[]
deactivate NumeroTelefonico
Contatto-->>RubricaViewController: contatto
deactivate Contatto
RubricaViewController->>GestoreContatti: aggiungi(Contatto contatto)
activate GestoreContatti
GestoreContatti-->>RubricaViewController: struttura aggiornata
deactivate GestoreContatti
RubricaViewController->>RubricaViewController: sort()
RubricaViewController-->>RubricaView.fxml: TableView aggiornata
@enduml
```

a.1. Comunicazione Gestore-Implementazione (Aggiungi)

```
@startuml
[o->GestoreContatti: aggiungi(Contacto c)
activate GestoreContatti
GestoreContatti->InsiemeContatti: aggiungi(Contacto c)\n//implementazione concreta
activate InsiemeContatti
InsiemeContatti->InsiemeContatti: sort()
InsiemeContatti-->GestoreContatti:struttura aggiornata
deactivate InsiemeContatti
[o<--GestoreContatti:struttura aggiornata
deactivate GestoreContatti
@enduml
```

b. MODIFICA

```
@startuml
actor Utente
Utente->Rubrica: avvio rubrica(main())
Rubrica->RubricaView.fxml: attiva
activate RubricaView.fxml
activate RubricaViewController
RubricaViewController->GestoreContatti: getInsieme()
activate GestoreContatti
GestoreContatti-->RubricaViewController: ObservableArrayList list
deactivate GestoreContatti
Utente->RubricaView.fxml:seleziona un Contacto\ngià presente nella tabella
RubricaViewController-->RubricaView.fxml: rileva il selezionamento
RubricaViewController-->RubricaView.fxml: riempie i campi di testo con \n le informazioni relative al
contatto
Utente->RubricaView.fxml:modifica i campi
RubricaViewController-->RubricaView.fxml: attiva pulsante modifica
Utente->RubricaView.fxml: pressOn(pulsanteModifica)
RubricaView.fxml->RubricaViewController: modificaContatto(ActionEvent event)
RubricaViewController->GestoreContatti: modifica(Contacto contatto, newName, new Surname,\n
newNumeri[],newMail[])
activate GestoreContatti
GestoreContatti->InsiemeContatti: modifica(Contacto contatto, newName, new Surname,\n
newNumeri[],newMail[])\n//implementazione concreta
activate InsiemeContatti
InsiemeContatti-->GestoreContatti:struttura aggiornata
deactivate InsiemeContatti
GestoreContatti-->RubricaViewController: struttura aggiornata
deactivate GestoreContatti
RubricaViewController->RubricaViewController:sort()
RubricaViewController-->RubricaView.fxml: TableView aggiornata
```


@enduml

b.1. Comunicazione Gestore-Implementazione (Modifica)

@startuml

```
[o->InsiemeContatti:modifica(Contatto
c,newName,newSurname,newNumeri[],newMail[])//implementazione concreta
activate InsiemeContatti
InsiemeContatti->>Contatto:setNome(newName)
Contatto-->>InsiemeContatti: nome aggiornato
InsiemeContatti->>Contatto:setCognome(newSurname)
Contatto-->>InsiemeContatti: cognome aggiornato
InsiemeContatti->>Contatto:setNumero1(newNumeri[0])
Contatto->>NumeroTelefonico:setNumero1(newNumeri[0])
activate NumeroTelefonico
NumeroTelefonico->>NumeroTelefonico:validazioneNumero(newNumeri[0])
NumeroTelefonico-->>Contatto:numero1 aggiornato
Contatto-->>InsiemeContatti:numero1 aggiornato
InsiemeContatti->>Contatto:setNumero2(newNumeri [1])
Contatto->>NumeroTelefonico:setNumero2(newNumeri[1])
NumeroTelefonico->>NumeroTelefonico:validazioneNumero(newNumeri[1])
NumeroTelefonico-->>Contatto:numero2 aggiornato
Contatto-->>InsiemeContatti:numero2 aggiornato
InsiemeContatti->>Contatto:setNumero3(newNumeri[2])
Contatto->>NumeroTelefonico:setNumero3(newNumeri[2])
NumeroTelefonico->>NumeroTelefonico:validazioneNumero(newNumeri[2])
NumeroTelefonico-->>Contatto:numero3 aggiornato
Contatto-->>InsiemeContatti:numero3 aggiornato
deactivate NumeroTelefonico
InsiemeContatti->>Contatto:setEmail1(newMail[0])
Contatto->>Email:setEmail1(newMail[0])
activate Email
Email->>Email:validazioneEmail(newMail[0])
Email-->>Contatto:email1 aggiornato
Contatto-->>InsiemeContatti:email1 aggiornato
InsiemeContatti->>Contatto:setEmail2(newMail[1])
Contatto->>Email:setEmail2(newMail[1])
Email->>Email:validazioneEmail(newMail[1])
Email-->>Contatto:email2 aggiornato
Contatto-->>InsiemeContatti:email2 aggiornato
InsiemeContatti->>Contatto:setEmail3(newMail[2])
Contatto->>Email:setEmail3(newMail[2])
Email->>Email:validazioneEmail(newMail[2])
Email-->>Contatto:email3 aggiornato
Contatto-->>InsiemeContatti:email3 aggiornato
deactivate Email
```

```

InsiemeContatti->InsiemeContatti:sort()
[o<--InsiemeContatti:struttura aggiornata
deactivate InsiemeContatti
@enduml

```

c. RIMUOVI

```

@startuml
actor Utente
Utente->Rubrica: avvio rubrica(main())
Rubrica->RubricaView.fxml: attiva
activate RubricaView.fxml
activate RubricaViewController
RubricaViewController->GestoreContatti: getInsieme()
activate GestoreContatti
GestoreContatti-->RubricaViewController: ObservableArrayList list
deactivate GestoreContatti
Utente->RubricaView.fxml:seleziona un Contatto\ngià presente nella tabella
RubricaViewController-->RubricaView.fxml: rileva il selezionamento
RubricaViewController-->RubricaView.fxml: riempie i campi di testo\n con le informazioni relative al
contatto
RubricaViewController-->RubricaView.fxml: attiva pulsante rimuovi
Utente->RubricaView.fxml: pressOn(pulsanteRimuovi)
RubricaView.fxml->RubricaViewController: rimuoviContatto(ActionEvent event)
RubricaViewController->GestoreContatti: rimuovi(Contatto contatto)
activate GestoreContatti
GestoreContatti->InsiemeContatti: rimuovi(Contatto contatto)\n//implementazione concreta
activate InsiemeContatti
InsiemeContatti-->GestoreContatti: struttura aggiornata
deactivate InsiemeContatti
GestoreContatti-->RubricaViewController: struttura aggiornata
deactivate GestoreContatti
RubricaViewController-->RubricaView.fxml: TableView aggiornata
@enduml

```

d. IMPORTA

```

@startuml
actor Utente
Utente->Rubrica: avvio rubrica(main())
Rubrica->RubricaView.fxml: attiva
activate RubricaView.fxml
activate RubricaViewController
RubricaViewController-->RubricaView.fxml: attiva pulsante "Importa da..."

```

```

RubricaViewController->GestoreContatti: getInsieme()
activate GestoreContatti
GestoreContatti-->RubricaViewController: ObservableArrayList list
deactivate GestoreContatti
Utente->RubricaView.fxml: pressOn(pulsanteImporta)
RubricaView.fxml->RubricaViewController: importaContatto(ActionEvent event)
RubricaViewController-->Utente: FileChooser
Utente->RubricaViewController: seleziona mediante il FileChooser \nil file da cui importare i contatti
RubricaViewController->GestoreContatti:setInsieme(gestore.importaContatti(file.getAbsolutePath()))
activate GestoreContatti
GestoreContatti->InsiemeContatti:importaContatti(nomeFile)
activate InsiemeContatti
InsiemeContatti-->RubricaViewController:ObservableArratList()
deactivate InsiemeContatti
RubricaViewController->GestoreContatti:getInsieme()
GestoreContatti->InsiemeContatti:getInsieme()
InsiemeContatti-->GestoreContatti:listaOsservabile
GestoreContatti-->RubricaViewController.listaOsservabile
deactivate GestoreContatti
RubricaViewController-->RubricaView.fxml: TableView aggiornata
@enduml

```

d.1. Comunicazione Gestore-Implementazione (Importa)

```

@startuml
[o->InsiemeContatti:importaContatti(nomeFile)
loop for finchè il file non finisce
InsiemeContatti->Contatto: new Contatto(nome, cognome, numeri[],email[])
activate Contatto
loop for finchè non vengono controllate\ntutte le email
Contatto->Email: new Email(email1,email2,email3)
activate Email
Email->Email:validazioneEmail()
Email-->Contatto: email[]
deactivate Email
end
loop for finchè non vengono controllati\ntutte i numeri
Contatto->NumeroTelefonico: new NumeroTelefonico(numero1,numero2,numero3)
activate NumeroTelefonico
NumeroTelefonico->NumeroTelefonico:validazioneNumero()
NumeroTelefonico-->Contatto: numero[]
deactivate NumeroTelefonico
end
Contatto-->InsiemeContatti:contatto
deactivate Contatto
end

```

```
InsiemeContatti->InsiemeContatti:sort()
[o<-InsiemeContatti:listaOsservabile
@enduml
```

e. ESPORTA

```
@startuml
actor Utente
Utente->Rubrica: avvio rubrica(main())
Rubrica->RubricaView.fxml: attiva
activate RubricaView.fxml
activate RubricaViewController
RubricaViewController-->RubricaView.fxml: attiva pulsante "Esporta contatti"
RubricaViewController->GestoreContatti: getInsieme()
activate GestoreContatti
GestoreContatti-->RubricaViewController: ObservableArrayList list
deactivate GestoreContatti
Utente->RubricaView.fxml: pressOn(pulsanteEsporta)
RubricaView.fxml->RubricaViewController: esportaContatto(ActionEvent event)
RubricaViewController-->Utente: FileChooser
Utente->RubricaViewController: seleziona mediante il FileChooser \nil file in cui esportare i contatti (se
non esiste verrà creato)
RubricaViewController->GestoreContatti: esportaContatti(file.getAbsolutePath())
activate GestoreContatti
GestoreContatti->InsiemeContatti: esportaContatti(nomeFile) \n//implementazione concreta
deactivate GestoreContatti
@enduml
```

e.1. Comunicazione Gestore-Implementazione (Esporta)

```
@startuml
[o->InsiemeContatti:esportaContatti(nomeFile)
activate InsiemeContatti
loop for finchè ci sono dei contatti\n nella mia lista
InsiemeContatti->Contatto:getOrDefault(getCognome)
activate Contatto
Contatto-->InsiemeContatti:cognome
InsiemeContatti->Contatto:getOrDefault(getNome)
Contatto-->InsiemeContatti:nome
InsiemeContatti->Contatto:getOrDefault(getNumero1Contatto)
Contatto->NumeroTelefonico:getNumero1
activate NumeroTelefonico
NumeroTelefonico-->Contatto:numero1
deactivate NumeroTelefonico
Contatto-->InsiemeContatti:numero1
InsiemeContatti->Contatto:getOrDefault(getNumero2Contatto)
```

```

Contatto->NumeroTelefonico:getNumero2
activate NumeroTelefonico
NumeroTelefonico-->Contatto:numero2
deactivate NumeroTelefonico
Contatto-->InsiemeContatti:numero2
InsiemeContatti->Contatto:getOrDefault(getNumero3Contatto)
Contatto->NumeroTelefonico:getNumero3
activate NumeroTelefonico
NumeroTelefonico-->Contatto:numero3
deactivate NumeroTelefonico
Contatto-->InsiemeContatti:numero3
InsiemeContatti->Contatto:getOrDefault(getEmail1Contatto)
Contatto->Email:getEmail1
activate Email
Email-->Contatto:email1
deactivate Email
Contatto-->InsiemeContatti:email1
InsiemeContatti->Contatto:getOrDefault(getEmail2Contatto)
Contatto->Email:getEmail2
activate Email
Email-->Contatto:email2
deactivate Email
Contatto-->InsiemeContatti:email2
InsiemeContatti->Contatto:getOrDefault(getEmail3Contatto)
Contatto->Email:getEmail3
activate Email
Email-->Contatto:email3
deactivate Email
Contatto-->InsiemeContatti:email3
deactivate Contatto
InsiemeContatti->nomefile.csv: scrittura sul file
end
deactivate InsiemeContatti
@enduml

```

f. RESET

```

@startuml
actor Utente
Utente->Rubrica: avvio rubrica(main())
Rubrica->RubricaView.fxml: attiva
activate RubricaView.fxml
activate RubricaViewController
RubricaViewController-->RubricaView.fxml: attiva pulsante "ResetRubrica"
RubricaViewController->GestoreContatti: getInsieme()
activate GestoreContatti

```

```

GestoreContatti-->RubricaViewController: ObservableArrayList list
deactivate GestoreContatti
Utente-->RubricaView.fxml: pressOn(pulsanteReset)
RubricaView.fxml-->RubricaViewController: resetRubrica(ActionEvent event)
RubricaViewController-->GestoreContatti: reset()
activate GestoreContatti
GestoreContatti-->InsiemeContatti: reset() //implementazione concreta
activate InsiemeContatti
InsiemeContatti-->GestoreContatti:svuota la collezione
deactivate InsiemeContatti
GestoreContatti-->RubricaViewController:collezione svuotata
deactivate GestoreContatti
RubricaViewController-->GestoreContatti: getInsieme()
activate GestoreContatti
GestoreContatti-->InsiemeContatti: getInsieme()
activate InsiemeContatti
InsiemeContatti-->GestoreContatti: listaOsservabile
deactivate InsiemeContatti
GestoreContatti-->RubricaViewController: listaOsservabile
deactivate GestoreContatti
RubricaViewController-->RubricaView.fxml: TableView svuotata
@enduml

```

g. RICERCA

g.1. CERCA (CONTATTO TROVATO)

```

actor Utente
Utente-->Rubrica: avvio rubrica(main())
Rubrica-->RubricaView.fxml: attiva
activate RubricaView.fxml
activate RubricaViewController
RubricaViewController-->GestoreContatti: getInsieme()
activate GestoreContatti
GestoreContatti-->RubricaViewController: ObservableArrayList list
deactivate GestoreContatti
Utente-->RubricaView.fxml:riempie il campo di ricerca\n con una stringa
RubricaViewController-->RubricaView.fxml: rileva il riempimento del campo di ricerca
Utente-->RubricaView.fxml: pressOn(pulsanteCerca)
RubricaView.fxml-->RubricaViewController: cercaContatto(ActionEvent event)
RubricaViewController-->GestoreContatti: getInsieme()
activate GestoreContatti
GestoreContatti-->RubricaViewController: listaOsservabile
deactivate GestoreContatti
RubricaViewController-->GestoreContatti: cerca("stringa inserita")
activate GestoreContatti

```

GestoreContatti-->RubricaViewController: listaOsservabile non nulla
deactivate GestoreContatti
RubricaViewController-->RubricaView.fxml: risultati della ricerca nella TableView
@enduml

g.2. CERCA (CONTATTO NON TROVATO)

@Startuml
actor Utente
Utente->Rubrica: avvio rubrica(main())
Rubrica->RubricaView.fxml: attiva
activate RubricaView.fxml
activate RubricaViewController
RubricaViewController->GestoreContatti: getInsieme()
activate GestoreContatti
GestoreContatti-->RubricaViewController: ObservableArrayList list
deactivate GestoreContatti
Utente->RubricaView.fxml:riempie il campo di ricerca\n con una stringa
RubricaViewController-->RubricaView.fxml: rileva il riempimento del campo di ricerca
Utente->RubricaView.fxml: pressOn(pulsanteCerca)
RubricaView.fxml->RubricaViewController: cercaContatto(ActionEvent event)
RubricaViewController->GestoreContatti: getInsieme()
activate GestoreContatti
GestoreContatti-->RubricaViewController: listaOsservabile
deactivate GestoreContatti
RubricaViewController->GestoreContatti: cerca("stringa inserita")
activate GestoreContatti
GestoreContatti-->RubricaViewController: listaOsservabile vuota
deactivate GestoreContatti
RubricaViewController->FinestraContattoNonTrovato: attiva
activate FinestraContattoNonTrovato
RubricaViewController-->RubricaView.fxml:TableView vuota
@enduml

g.1.1 Comunicazione Gestore-Implementazione (Stringa non Vuota)

@startuml
[o->GestoreContatti: getInsieme()
activate GestoreContatti
GestoreContatti->InsiemeContatti: getInsieme()\n //implementazione concreta
activate InsiemeContatti
InsiemeContatti-->GestoreContatti:listaOsservabile
deactivate InsiemeContatti
[o<--GestoreContatti:struttura aggiornata
deactivate GestoreContatti
deactivate GestoreContatti

```

[o->GestoreContatti: cerca("stringa inserita")
GestoreContatti->InsiemeContatti: cerca("stringa inserita") \n //implementazione concreta
activate InsiemeContatti
loop for Contatto in InsiemeContatti
InsiemeContatti->Contatto: getNome()
activate Contatto
Contatto-->InsiemeContatti: nome
deactivate Contatto
InsiemeContatti->Contatto: getCognome()
activate Contatto
Contatto-->InsiemeContatti: cognome
deactivate Contatto
end
InsiemeContatti-->GestoreContatti: listaOsservabile
[o<--GestoreContatti: listaOsservabile
@enduml

```

g.2.1 Comunicazione Gestore-Implementazione (Stringa Vuota)

```

@startuml
[o->GestoreContatti: getInsieme()
activate GestoreContatti
GestoreContatti->InsiemeContatti: getInsieme()\n //implementazione concreta
activate InsiemeContatti
InsiemeContatti-->GestoreContatti: listaOsservabile
deactivate InsiemeContatti
[o<--GestoreContatti: listaOsservabile
deactivate GestoreContatti
[o->GestoreContatti: getInsieme()
activate GestoreContatti
GestoreContatti->InsiemeContatti: getInsieme()\n //implementazione concreta
activate InsiemeContatti
InsiemeContatti-->GestoreContatti: listaOsservabile
deactivate InsiemeContatti
[o<--GestoreContatti: listaOsservabile
@enduml

```