

# Documento dei Requisiti

Sviluppo di un programma che implementa una rubrica telefonica

Partecipanti al gruppo: Beatrice Rebecca, Caliendo Antonio, Cambria Jacopo, Gaeta Michela

## Indice

<b>1. Introduzione .....</b>	<b>2</b>
<b>a. Obiettivi del progetto .....</b>	<b>2</b>
<b>2. Requisiti funzionali.....</b>	<b>2</b>
<b>a. Funzionalità individuali.....</b>	<b>2</b>
<b>b. Esigenze dati e formato dei dati.....</b>	<b>4</b>
<b>c. Funzionalità aggiuntive .....</b>	<b>4</b>
<b>3. Requisiti non funzionali .....</b>	<b>4</b>
<b>a. Interfaccia utente .....</b>	<b>4</b>
<b>b. Interfaccia utente per funzionalità aggiuntive .....</b>	<b>5</b>
<b>4. Tabella dei requisiti .....</b>	<b>5</b>
<b>5. Modello del sistema .....</b>	<b>9</b>
<b>a. Descrizione dei casi d'uso .....</b>	<b>9</b>
<b>b. Diagrammi dei casi d'uso .....</b>	<b>12</b>
<b>c. Diagramma dei casi d'uso - Codice sorgente UML .....</b>	<b>12</b>

# 1.Introduzione

## a. Obiettivi del progetto

Il seguente progetto ha come obiettivo la realizzazione di un software per la gestione di una rubrica di contatti personali.

Questa sarà dotata di un'interfaccia grafica (GUI) che consentirà agli utenti di aggiungere, modificare, eliminare, cercare o salvare i propri contatti in modo efficiente ed intuitivo.

Inoltre, il programma permetterà di salvare e caricare l'intera rubrica da file, offrendo agli utenti la possibilità di gestire facilmente i propri dati.

La realizzazione del progetto segue un approccio, suddiviso in diverse fasi:

1. Analisi dei requisiti, per identificare le funzionalità principali e le caratteristiche funzionali e non funzionali;
2. Progettazione di alto e basso livello, con la definizione delle strutture di dati e dei flussi operativi;
3. Implementazione, tramite strumenti di sviluppo avanzati e di linguaggi di programmazione moderni;
4. Testing, per garantire l'affidabilità, l'usabilità e la qualità complessiva del programma.

L'intero processo sarà documentato in dettaglio e reso disponibile attraverso un repository su **GitHub**, configurato con **Maven**.

Gli attori principali sono i generici utenti, che interagiranno con il programma per gestire i propri contatti in modo semplice e veloce. L'applicazione è pensata per essere utilizzata da persone senza competenze tecniche specifiche, rendendo il processo di gestione dei contatti accessibile a tutti.

L'obiettivo finale è creare un software robusto, scalabile e manutenibile, che soddisfi le esigenze degli utenti nella gestione dei contatti.

## 2. Requisiti funzionali

### a. Funzionalità individuali

**FI0.0 Aggiunta contatto da tastiera:** Il programma deve essere in grado di permettere l'aggiunta di un nuovo contatto mediante informazioni valide. L'utente deve essere in grado di aggiungere un contatto inserendo nome, cognome, numero/i telefonico/i e contatto/i di posta elettronica in apposite caselle testuali ad essi dedicate.

L'azione di aggiunta avrà come conseguenza la visualizzazione del contatto all'interno di una tabella in cui verranno esplicitati i vari campi.

**FI0.1 Modifica di un contatto esistente:** Il programma deve permettere la modifica di un contatto già inserito in precedenza, l'operazione sarà accessibile a prescindere dall'origine del contatto (*se inserito da tastiera o caricato mediante apposito file*).

Dopo aver selezionato il contatto da modificare, l'azione porta alla possibilità di alterare uno qualsiasi dei campi di cui un contatto è composto. I dati vecchi verranno sovrascritti con quelli appena modificati solo se questi ultimi risulteranno essere validi.

L'azione di modifica avrà come conseguenza la visualizzazione del contatto all'interno della rubrica in cui verranno esplicitati i vari campi contenenti i nuovi valori modificati.

**FI0.2 Rimozione di un contatto:** Il programma deve permettere la rimozione di un contatto esistente, a prescindere dalla sua provenienza (*se inserito da tastiera o caricato mediante apposito file*). Dopo aver selezionato l'utente da rimuovere sarà possibile procedere con l'operazione mediante un apposito pulsante.

La rimozione di un contatto avrà come conseguenza la cancellazione permanente e definitiva del contatto dalla lista, questo, infatti, al termine dell'operazione non sarà più visibile nell'elenco e non sarà più accessibile.

**FI0.3 Ricerca di un contatto tramite sottostringa:** Il programma rende possibile la ricerca di un contatto tramite sottostringa. Mediante un apposito campo di testo sarà possibile inserire la sottostringa da ricercare all'interno del campo nome o del campo cognome del contatto. Dopo aver confermato la sottostringa da ricercare, sarà possibile osservare una lista contenente tutti i contatti che la includono.

**FI0.3.0 Contatto non trovato:** In caso di contatto non trovato, verrà visualizzata la lista di partenza. Inoltre, apparirà una finestra con il messaggio "*Contatto non trovato*" per notificare l'utente.

**FI0.4 Caricare contatti da file:** Il programma rende possibile l'inizializzazione della lista contatti mediante file esterno. L'utente sarà in grado di selezionare, tramite un apposito pulsante, un file con estensione .csv (comma separated value) presente sul proprio dispositivo, da cui importare i contatti.

La conseguenza del caricamento del file sarà la visualizzazione dei nuovi contatti nella tabella principale, i contatti precedentemente inseriti verranno sovrascritti e non saranno più accessibili e/o modificabili.

**FI0.5 Esportare elenco contatti su file:** Il programma permette di copiare l'elenco contatti attuale all'interno di un apposito file. Scelta l'operazione, l'intera lista contatti verrà esportata all'interno di un file. Non sarà possibile copiare solo contatti selezionati ma l'operazione riguarderà sempre l'intera rubrica.

La conseguenza dell'azione comporta la creazione di un file con estensione .csv (comma separated value), nel quale verrà rappresentato lo stato della lista contatti nel momento in cui si effettua l'operazione.

La lista contatti presente nella rubrica rimarrà invariata.

## b. Esigenze dati e formato dei dati

**DF0.0 Formato contatto:** Il generico contatto presenta le seguenti informazioni:

- Nome
- Cognome
- Numero/i (*numero1, numero2, numero3*)
- E-mail (*e-mail1, e-mail2, e-mail3*)

Un contatto per il quale anche solo uno dei campi descritti non risulta essere valido non potrà essere inserito.

**DF0.0.1 Limite indirizzi e numeri di telefono:** Per ogni generico contatto sarà possibile definire da 0 a 3 numeri di telefono e da 0 a 3 indirizzi di posta elettronica.

**DF0.0.2 Riempimento campi:** I campi *e-mail* e *numero* possono essere vuoti, tuttavia, almeno uno tra i campi nome e cognome deve essere riempito al fine di consentire la creazione e/o modifica di un contatto

**DF0.0.2.1 Validazione campi numero e e-mail:** Il campo numero sarà valido solo se l'utente inserirà delle cifre numeriche mentre il campo e-mail sarà valido solo se l'utente inserirà un testo contenente il simbolo '@'.

## c. Funzionalità aggiuntive

**FIA0.6 Reset contatti:** Il programma permette la cancellazione permanente e definitiva dell'intera rubrica. Una volta selezionata l'azione, l'intera lista dei contatti posizionata all'interno della tabella verrà eliminata. I contatti presenti in precedenza non saranno più accessibili e/o visualizzabili.

## 3. Requisiti non funzionali

### a. Interfaccia utente

**UI0.0 Layout principale:** Aperta l'applicazione, sarà possibile osservare un banner superiore contenente pulsanti interattivi per accedere alle varie operazioni riguardanti l'intera rubrica.

All'interno del banner sarà presente un campo di testo che permetterà la ricerca di un contatto tramite sottostringa iniziale.

Al di sotto del banner sarà possibile osservare la rappresentazione della lista contatti.

**UI0.1 Rappresentazione lista contatti:** La lista contatti verrà rappresentata mediante una tabella composta da tante colonne quanti sono i campi caratterizzanti di un contatto (*Cognome, Nome, Numero1, Numero2, Numero3, E-mail1, E-mail2, E-mail3*), ciascuna colonna sarà dotata di etichetta che specializzerà il campo. I contatti dovranno essere visualizzati in ordine alfabetico (per cognome e nome) inoltre ognuno di

essi verrà disposto lungo una riga; eventuali campi vuoti indicheranno che per un determinato contatto l'informazione associata alla colonna non è stata definita.

**UI0.1.1 Eliminazione contatti:** Quando un contatto viene eliminato, la sua riga corrispondente nella tabella viene cancellata

**UI0.5 Editor contatti:** L'editor dei contatti sarà sempre disponibile al lato della schermata; tuttavia, i suoi pulsanti saranno abilitati solo quando si deciderà di aggiungere, rimuovere o modificare un contatto.

**UI0.5.0 Disattivazione pulsante di aggiunta/modifica/rimuovi:** I pulsanti di modifica e rimozione saranno disattivati se non è selezionato alcun contatto, mentre il pulsante di aggiunta sarà disattivato se i campi del nome e del cognome sono entrambi vuoti.

La disattivazione del pulsante comporta l'impossibilità di aggiungere, modificare o rimuovere il contatto dalla rubrica.

## b. Interfaccia utente per funzionalità aggiuntive

**UIA0.1.2 Reset rubrica:** Quando l'utente effettua il reset della lista contatti nella sua interezza, la tabella torna nel suo stato iniziale, senza alcuna riga piena.

## 4. Tabella dei requisiti

Poiché non è stato possibile interagire con un cliente reale, si è simulato il processo di raccolta dei requisiti assumendo il punto di vista di potenziali utenti finali.

Quest'approccio ha permesso di definire in modo strutturato sia i **requisiti funzionali**, che descrivono le principali funzionalità del sistema, sia quelli **non funzionali**, che specificano sia i vincoli che le caratteristiche operative.

La seguente tabella raccoglie l'elenco dei requisiti e delle funzionalità identificati.

ID	Tipologia requisito	Descrizione	Priorità	Rischio tecnico di realizzazione	Legame con altri requisiti
FI0.0	Funzionalità individuali  (Requisito Funzionale)	Aggiunta contatto da tastiera	ALTA	BASSO	DF0.0, DF0.0.1, DF0.0.2, DF0.0.2.1, UI0.5, UI0.5.0

ID	Tipologia requisito	Descrizione	Priorità	Rischio tecnico di realizzazione	Legame con altri requisiti
FI0.1	Funzionalità individuali  <b>(Requisito Funzionale)</b>	Modifica di un contatto esistente	ALTA	MEDIO	FI0.0, DF0.0, DF0.0.1, DF0.0.2, DF0.0.2.1, UI0.5, UI0.5.0
FI0.2	Funzionalità individuali  <b>(Requisito Funzionale)</b>	Rimozione di un contatto	ALTA	BASSO	FI0.0, UI0.1.1
FI0.3	Funzionalità individuali  <b>(Requisito Funzionale)</b>	Ricerca di un contatto tramite sottostringa	ALTA	BASSO	FI0.3.0, UI0.0
FI0.3.0	Funzionalità individuali  <b>(Requisito Funzionale)</b>	Contatto non trovato	ALTA	BASSO	FI0.3
FI0.4	Funzionalità individuali  <b>(Requisito Funzionale)</b>	Caricare contatti da file	ALTA	BASSO	UI0.0
FI0.5	Funzionalità individuali  <b>(Requisito Funzionale)</b>	Esportare elenco contatti su file	ALTA	BASSO	UI0.0

ID	Tipologia requisito	Descrizione	Priorità	Rischio tecnico di realizzazione	Legame con altri requisiti
DF0.0	Esigenze dati e formato dati  (Requisito Funzionale)	Formato contatto	ALTA	BASSO	DF0.0.1, DF0.0.2, DF0.0.2.1
DF0.0.1	Esigenze dati e formato dati  (Requisito Funzionale)	Limite indirizzi e numeri di telefono	ALTA	MEDIO	DF0.0, DF0.0.2, DF0.0.4
DF0.0.2	Esigenze dati e formato dati  (Requisito Funzionale)	Riempimento campi	ALTA	BASSO	DF0.0, DF0.0.1, DF0.0.2.1
DF0.0.2.1	Esigenze dati e formato dati  (Requisito Funzionale)	Validazione campi numero e email	ALTA	MEDIO	DF0.0, DF0.0.1, DF0.0.2
FIA0.6	Funzionalità aggiuntive  (Requisito Funzionale)	Reset contatti	BASSA	MEDIO	UI0.0, UI0.1
UI0.0	Interfaccia utente  (Requisito non Funzionale)	Layout principale	MEDIA	BASSO	UI0.1

<b>ID</b>	<b>Tipologia requisito</b>	<b>Descrizione</b>	<b>Priorità</b>	<b>Rischio tecnico di realizzazione</b>	<b>Legame con altri requisiti</b>
<b>UI0.1</b>	Interfaccia utente <b>(Requisito non Funzionale)</b>	Rappresentazione lista contatti	<b>MEDIA</b>	<b>BASSO</b>	<b>DF0.0</b>
<b>UI0.1.1</b>	Interfaccia utente <b>(Requisito non Funzionale)</b>	Eliminazione contatti	<b>MEDIA</b>	<b>BASSO</b>	<b>FI0.2, UI0.1</b>
<b>UI0.5</b>	Interfaccia utente <b>(Requisito non Funzionale)</b>	Editor contatti	<b>ALTA</b>	<b>BASSO</b>	<b>UI0.0</b>
<b>UI0.5.0</b>	Interfaccia utente <b>(Requisito non Funzionale)</b>	Disattivazione pulsante di aggiunta	<b>ALTA</b>	<b>BASSO</b>	<b>DF0.0, DF0.0.1, DF0.0.2, DF0.0.2.1, FI0.0, FI0.1, UI0.5</b>
<b>UIA0.1.2</b>	Interfaccia utente per funzionalità aggiuntive <b>(Requisito non Funzionale)</b>	Reset rubrica	<b>BASSA</b>	<b>MEDIO</b>	<b>FIA0.6, UI0.0, UI0.1</b>



## 5. Modello del sistema

### a. Descrizione dei casi d'uso

#### 1. **Nome:** Aggiungi contatto

**Attore partecipante:** Utente

**Precondizioni:**

- Utente accede alla rubrica attraverso un dispositivo

**Post condizioni:**

- Utente aggiunge un nuovo contatto alla sua rubrica

**Flussi di eventi normali:**

1. Utente accede alla rubrica avviando il programma
2. Utente compila i campi richiesti
3. Utente clicca sull'opzione "Aggiungi contatto" presente nel menù
4. Il contatto viene aggiunto alla rubrica

**Flussi di eventi alternativi:**

- 3a. Utente non inserisce nome e cognome del contatto
  - 3.a.1. L'esecuzione riprende dal passo 2
- 3b. Nel campo "numero di telefono" viene inserito un carattere alfabetico
  - 3.b.1. L'esecuzione riprende dal passo 2
- 3c. Nel campo "e-mail" non viene inserito il carattere speciale "@"
  - 3.c.1. L'esecuzione riprende dal passo 2

#### 2. **Nome:** Modifica contatto

**Attore partecipante:** Utente

**Precondizioni:**

- Utente accede alla rubrica attraverso un dispositivo
- Utente selezione contatto da modificare

**Post condizioni:**

- Utente modifica contatto selezionato

**Flussi di eventi normali:**

1. Utente accede alla rubrica avviando il programma
2. Utente seleziona o cerca un contatto già esistente
3. Utente procede a modificare uno o più campi del contatto
4. Utente seleziona l'opzione "Modifica contatto"
5. Utente salva le modifiche effettuate
6. I valori del contatto verranno sovrascritti

**Flussi di eventi alternativi:**

- 4a. I campi del nome e cognome del contatto sono vuoti
  - 4.a.1. L'esecuzione riprende dal passo 3
- 4b. Nel campo "numero di telefono" viene inserito un carattere alfabetico
  - 4.b.1. L'esecuzione riprende dal passo 3
- 4c. Nel campo "e-mail" non viene inserito il carattere speciale "@"
  - 4.c.1. L'esecuzione riprende dal passo 3

### **3. Nome:** Rimuovi contatto

**Attore partecipante:** Utente

**Precondizioni:**

- Utente accede alla rubrica attraverso un dispositivo
- Utente seleziona contatto da rimuovere

**Post condizioni:**

- Utente rimuove un contatto dalla sua rubrica

**Flussi di eventi normali:**

1. Utente accede alla rubrica avviando il programma
2. Utente seleziona o cerca un contatto già esistente
3. Utente seleziona l'opzione "Rimuovi contatto"
4. Il contatto non sarà più visibile in rubrica

### **4. Nome:** Importa da file

**Attore partecipante:** Utente

**Precondizioni:**

- Utente accede alla rubrica attraverso un dispositivo
- Utente seleziona il file da cui importare i contatti

**Post condizioni:**

- Nella rubrica saranno presenti tutti i contatti importati dal file

**Flussi di eventi normali:**

1. Utente accede alla rubrica avviando il programma
2. Utente clicca sull'opzione "Importa da ..." presente nella parte superiore del layout
3. Utente seleziona il file da cui importare i contatti
4. I contatti vengono importati

**Flussi di eventi alternativi:**

**3a.** Il File da importare non viene trovato

**3.a.1.** Il sistema mostra un messaggio di errore "File non trovato"

**3.a.2.** L'esecuzione riprende dal passo 2

### **5. Nome:** Esporta su file

**Attore partecipante:** Utente

**Precondizioni:**

- Utente accede alla rubrica attraverso un dispositivo

**Post condizioni:**

- Utente esporta i contatti su un file

**Flussi di eventi normali:**

1. Utente accede alla rubrica avviando il programma

2. Utente clicca sull'opzione "Esporta su file" presente nella parte superiore del layout
3. Utente crea un file ".csv" su cui scrivere

**Flussi di eventi alternativi:**

- 3.b. Il file su cui esportare risulta essere pieno
  - 3.b.1. Il file su cui esportare verrà sovrascritto
  - 3.b.2. L'operazione termina correttamente

**6. Nome:** Cerca contatto

**Attore partecipante:** Utente

**Precondizioni:**

- Utente accede alla rubrica attraverso un dispositivo

**Post condizioni:**

- Utente trova e visualizza il contatto/la lista di contatti nella sua rubrica

**Flussi di eventi normali:**

1. Utente accede alla rubrica avviando il programma
2. Utente clicca sulla barra di ricerca presente nella parte superiore
3. Utente inserisce una sottostringa
4. Utente visualizza il contatto/la lista di contatti aventi nome o cognome che iniziano con la sottostringa inserita

**Flussi di eventi alternativi:**

- 3a. I parametri inseriti per la ricerca non corrispondono a nessun contatto in rubrica
  - 3.a.1. Il sistema visualizza una finestra con la scritta "Contatto non trovato"
  - 3.a.2. L'esecuzione riprende dal passo 3

**7.Nome:** Reset rubrica

**Attore partecipante:** Utente

**Precondizioni:**

- Utente accede alla rubrica attraverso un dispositivo

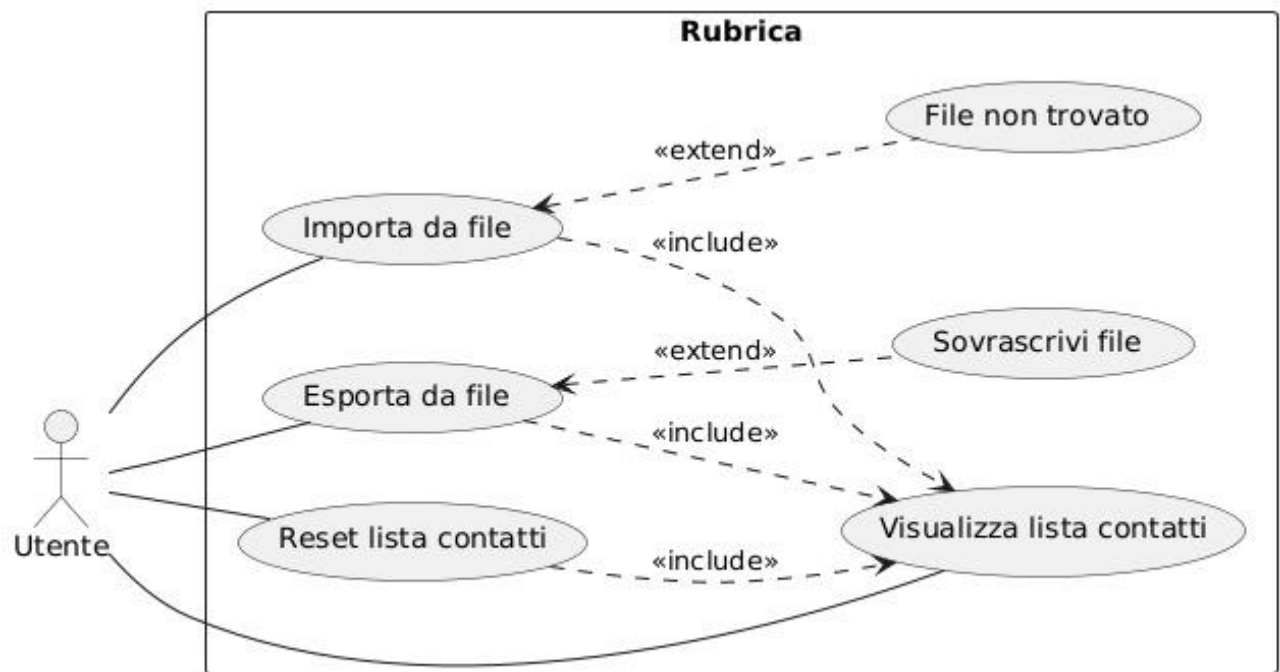
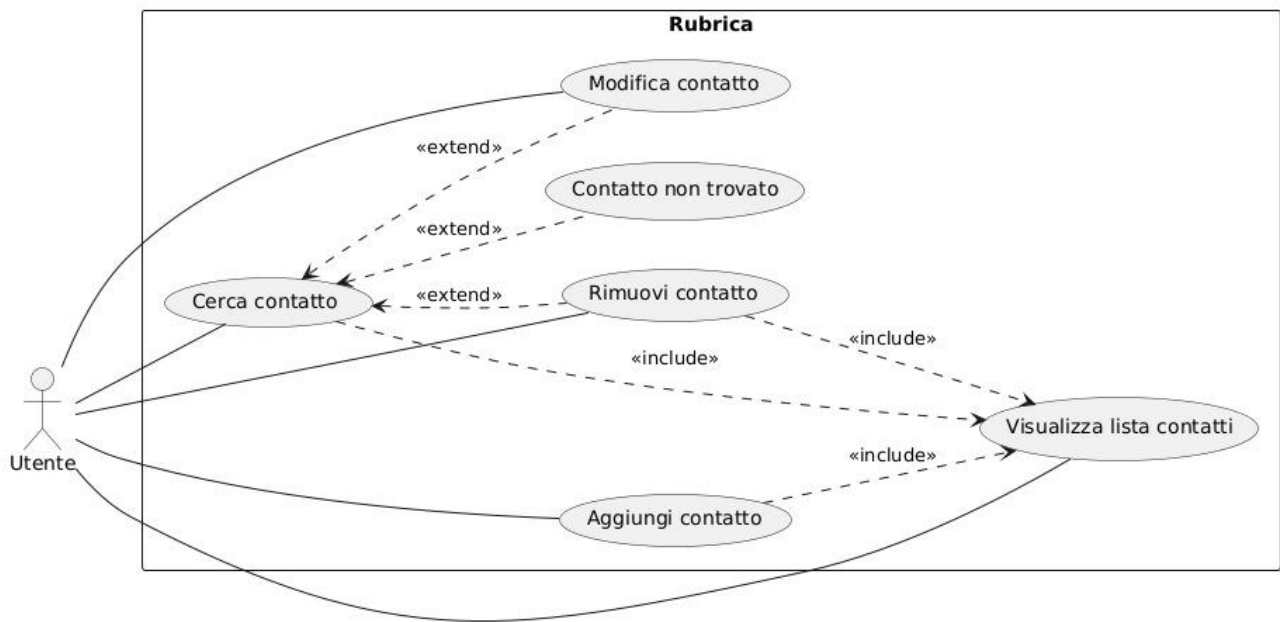
**Post condizioni:**

- Utente rimuove tutti i contatti presenti in rubrica
- La rubrica è vuota

**Flussi di eventi normali:**

1. Utente accede alla rubrica avviando il programma
2. Utente clicca sull'opzione "Rimuovi tutti i contatti" presente nella parte superiore del layout

## b. Diagrammi dei casi d'uso



## c. Diagramma dei casi d'uso - Codice sorgente UML

Il seguente codice rappresenta il diagramma UML dei casi d'uso per la gestione della rubrica. È stato generato utilizzando PlantUML e descrive le interazioni tra l'attore "Utente" e le funzionalità offerte dal sistema. Ogni caso d'uso è modellato con relazioni di inclusione ed estensione, dove necessario.

**Codice:**

```
@startuml
left to right direction
actor Utente as u
rectangle Rubrica{
usecase "Aggiungi contatto" as uc1
usecase "Cerca contatto" as uc2
usecase "Visualizza lista contatti" as uc3
usecase "Contatto non trovato" as uc4
usecase "Rimuovi contatto" as uc5
usecase "Modifica contatto" as uc6
}
u -- uc1
u -- uc2
u -- uc3
u -- uc5
u -- uc6
uc1 ..> uc3:<<include>>
uc2 ..> uc3:<<include>>
uc5 ..> uc3:<<include>>
uc2 <.. uc4:<<extend>>
uc2 <.. uc5:<<extend>>
uc2 <.. uc6:<<extend>>
@endum
```

```
@startuml
left to right direction
actor Utente as u
rectangle Rubrica{
usecase "Reset lista contatti" as uc1
usecase "Importa da file" as uc2
usecase "Esporta da file" as uc3
usecase "File non trovato" as uc4
usecase "Visualizza lista contatti" as uc5
usecase "Sovrascrivi file" as uc6
}
u -- uc1
u -- uc2
u -- uc3
u -- uc5
uc1 ..> uc5:<<include>>
uc2 ..> uc5:<<include>>
```

```
uc2 <.. uc4:<<extend>>  
uc3 ..> uc5:<<include>>  
uc3 <.. uc6:<<extend>>  
@endum
```