

Università degli Studi di Salerno

Dipartimento di Ingegneria dell'Informazione, Elettrica e Matematica Applicata

Project Design

Foschillo Gennaro

Contents

1	Cla	ss Diagram Generale	3	
2	LowLevelClasses			
	2.1	Classe Contact	4	
	2.2	Classe ContactChecker	4	
	2.3	Classe ContactList	5	
	2.4	Classe ContactListManager	5	
	2.5	Classe File Manager	5	
3	Hig	hLevelClasses	6	
	3.1	Classe DashboardController:	7	
	3.2	Classe ContactApplication:	7	

4	Sequence Diagrams				
	4.1	Add Contact	7		
	4.2	Remove Contact	8		
	4.3	Modify Contact	9		
	4.4	View Contact	10		
	4.5	Search Contact	11		
	4.6	Load File	11		
	4.7	Save File	12		

1 Class Diagram Generale

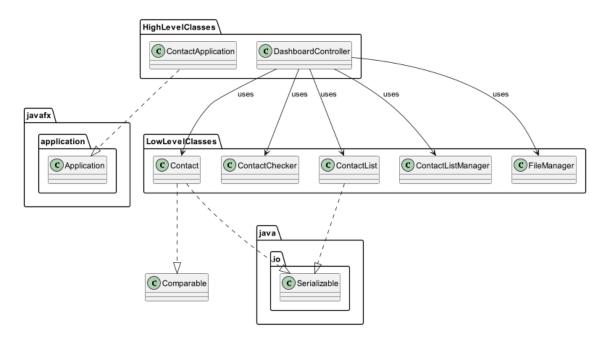


Figure 1: Diagramma delle classi completo

Il progetto è strutturato in due package principali, ciascuno con una responsabilità ben definita, al fine di favorire la separazione delle responsabilità e la manutenibilità del codice:

- *HighLevelClasses:* Questo package contiene le classi responsabili della gestione e visualizzazione della User Interface. In particolare, include:
 - La classe ContactApplication, che estende javafx.application. Application.
 Si occupa di inizializzare la scena a partire da un file FXML, gestendo quindi la presentazione della finestra iniziale.
 - La classe DashboardController, che gestisce l'interazione tra l'utente e la scena, gestendo le operazioni di modifica, aggiunta, visualizzazione, rimozione di contatti, il salvataggio (manuale) dei contatti ed il caricamento (manuale e automatico).
- LowLevelClasses: In questo package sono raggruppate le classi che si occupano della logica applicativa a basso livello e della gestione dei dati. In particolare, troviamo:
 - Il modello dei dati rappresentato dalla classe *Contact*
 - Le classi di utilità quali ContactChecker, per la validazione degli input (email e numeri), e ContactListManager, che fornisce metodi per la ricerca e l'aggiornamento dei contatti.
 - La classe ContactList, che funge da contenitore per i contatti, e la classe FileManager, che gestisce il caricamento e il salvataggio dei dati su file, utilizzando la libreria Gson per la serializzazione.

Questa organizzazione permette di isolare la logica di presentazione (GUI) dalla logica di businesse di gestione dei dati, facilitando così l'estendibilità e la manutenzione dell'applicazione. Nella parte successiva del documento verranno presentati i diagrammi delle classi specifiche per ciascun componente, mentre il diagramma generale fornisce una panoramica complessiva dell'architettura del progetto.

2 LowLevelClasses

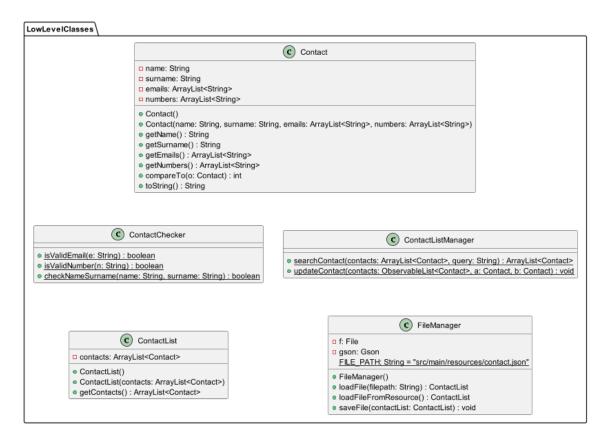


Figure 2: Diagramma delle classi LowLevel

2.1 Classe Contact

Questa classe astrae il concetto di Contatto, avendo le sole proprietà di: name, surname, emails e numbers. I metodi associati sono minimi per delegare a questa classe l'unica responsabilità di rappresentazione dei dati e che quindi altre operazioni vengano gestite in classi Utility separate successivamente descritte. Questo permette di mantenere un livello di coesione alto.

2.2 Classe ContactChecker

Questa classe utility si occupa della validazione dei campi email e numeri di telefono secondo il formato scelto nell'analisi dei requisiti. Ciò viene fatto attraverso il matching di una regular expression. Inoltre, è presente un metodo per verificare, all'interno di un contatto, la presenza di almeno un campo tra nome e cognome. Questa logica verrà utilizzata nel package HighLevelClasses per la validazione degli input di aggiunta e modifica performati dall'utente.

2.3 Classe ContactList

È una classe Wrapper con il solo compito di incapsulare i contatti all'interno di una struttura dati del tipo ArrayList, avendo come unico metodo di utilità il ritorno della lista di contatti. Questo permette di separare la logica di gestione da quella di rappresentazione della struttura dati, favorendo la modularità del codice.

2.4 Classe ContactListManager

Questa classe Utility si occupa delle operazioni principali da performare sulla lista di contatti:

- search: ricerca di un contatto all'interno di una lista generica: ciò viene fatto attraverso la Stream API offerta da Java.
- update: aggiornamento di un contatto all'interno della lista (dovuto ad una modifica di un contatto esistente).

2.5 Classe File Manager

Questa classe gestisce la persistenza dei dati dell'applicazione attraverso 3 metodi:

- loadFileFromResource: Operazione che viene effettuata all'avvio dell'applicazione per caricare il file se presente.
- loadFile & saveFile: Operazioni che possono essere svolte dall'utente all'interno della GUI.

3 HighLevelClasses



Figure 3: Diagramma delle classi HighLevel

3.1 Classe DashboardController:

Questa classe ha il compito di inizializzare la scena FXML con gli elementi principali e di controllare gli eventi performati dall'utente. I metodi descritti nel diagramma delle classi sono necessari per il corretto funzionamento della scena. La classe contiene anche logica di validazione, aggiornamento e interazione con il modello dei dati, garantendo così una buona separazione delle responsabilità tra la presentazione e la gestione dei dati.

3.2 Classe ContactApplication:

Questa classe permette l'avvio dell'applicazione, presentando la scena iniziale tramite FXML e impostando alcune proprietà della finestra (ad es. dimensioni, non ridimensionabile).

4 Sequence Diagrams

4.1 Add Contact

La seguente sequenza mostra il flusso di eventi quando l'utente aggiunge un nuovo contatto. Quando l'utente seleziona l'opzione di aggiunta, verrà aperta una Dashboard di aggiunta con i campi inizialmente vuoti. L'utente dovrà aggiungere almeno un nome o cognome per poter confermare la scelta. Una volta premuto il bottone di conferma, ogni e-mail e numero inserito viene validato e , in caso di errore, l'aggiunta non viene performata e l'utente dovrà modificare il campo errato e cliccare di nuovo il bottone di conferma. Finita la validazione dei campi, viene creata una nuova istanza di Contact con le nuove informazioni inserite che sarà aggiunta alla lista dei contatti e la tabella sarà aggiornata.

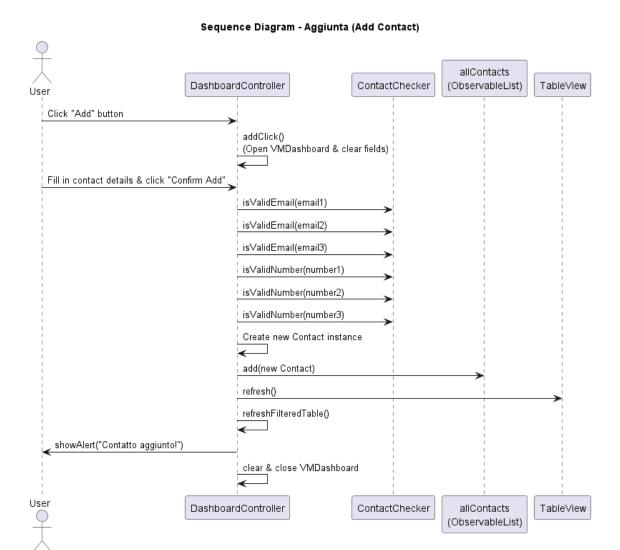


Figure 4: Sequence Diagram Aggiunta

4.2 Remove Contact

Questo diagramma descrive la sequenza di eventi per l'eliminazione di un contatto. Dopo che l'utente ha selezionato l'opzione elimina, verrà visualizzato un Alert per confermare la scelta. Se viene confermata, il contatto viene rimosso dalla lista, viene aggiornata la tabella principale e la tabella di ricerca. In caso contrario, l'utente ritorna alla dashboard principale.

Sequence Diagram - Rimozione del Contatto

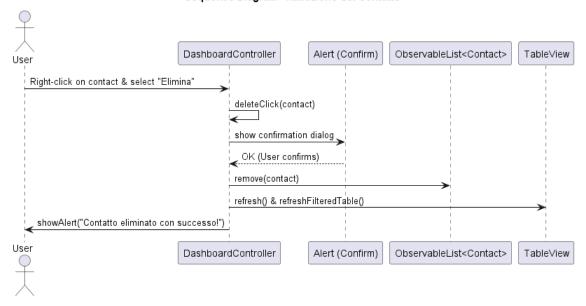


Figure 5: Sequence Diagram Rimozione

4.3 Modify Contact

Il diagramma segue la sequenza per la modifica di un contatto: dal momento in cui l'utente seleziona il contatto da modificare, Viene aperta la Dashboard di modifica, ed i campi vengono caricati con le informazioni del contatto corrente da modificare. Quando l'utente confermerà la modifica, ci sarà una fase di validazione delle email e numeri. Una volta terminata, il contatto modificato sovrascriverà il contatto esistente, verranno aggiornate le liste e la tabella.

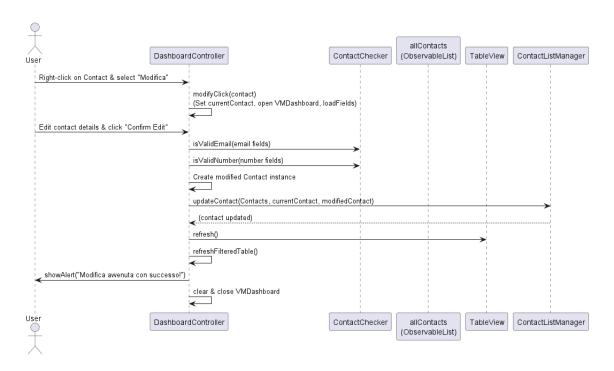


Figure 6: Sequence Diagram Modifica

4.4 View Contact

Questo diagramma illustra come l'utente può visualizzare i dettagli di un contatto selezionato. Quando l'utente clicca sul contatto da visualizzare, i campi vengono caricati con le informazioni del contatto, viene disabilitata la modifica e viene mostrato il bottone per uscire dalla visualizzazione.

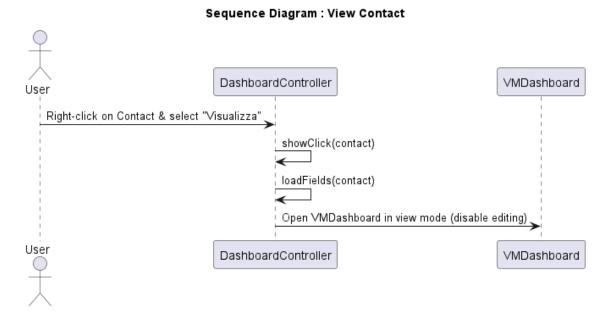


Figure 7: Sequence Diagram Visualizzazione

4.5 Search Contact

Il diagramma descrive la sequenza per la ricerca di un contatto. Viene illustrato come il sistema filtra la lista dei contatti in base alla query immessa dall'utente, aggiornando la visualizzazione della TableView.

Sequence Diagram : Search Contact

Figure 8: Sequence Diagram Ricerca

4.6 Load File

Qui viene mostrata la sequenza di eventi per il caricamento di un file contenente i contatti. Il diagramma evidenzia come il sistema apre un FileChooser, carica il file selezionato, deserializza i dati e aggiorna la visualizzazione.

Sequence Diagram : Load Contact List

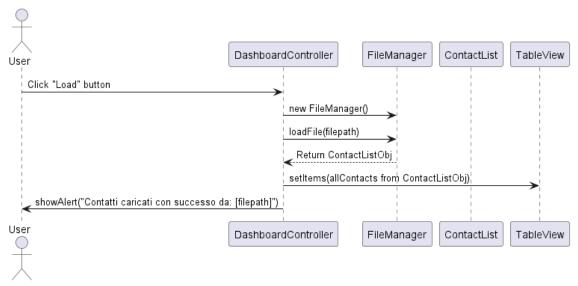


Figure 9: Sequence Diagram Carica File

4.7 Save File

Questo diagramma descrive la sequenza per il salvataggio della lista dei contatti su file. Viene illustrato come il sistema serializza i dati e li scrive nel file, fornendo all'utente un feedback di conferma.

Sequence Diagram : Save Contact List

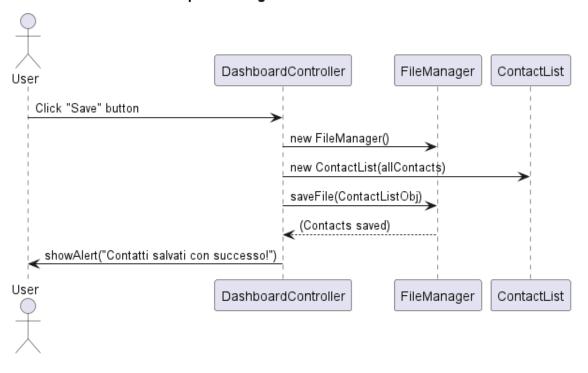


Figure 10: Sequence Diagram Salva File