



RUBRICA CONTATTI

Applicazione per la gestione dei contatti di una rubrica contenente numeri di telefono e indirizzi e-mail realizzata in Java attraverso una GUI in JavaFX. Gruppo 23.

Nazzaro Luca, Radesca Antonio, Savino Michele, Scandone Alessandro






PROJECT SETUP

Nella prima fase è stato creato il repository su GitHub.

In seguito ogni partecipante del gruppo ha collegato il repository di GitHub ad una cartella locale. Ognuno ha creato il proprio **branch**: così facendo, ogni singolo componente ha potuto dare il proprio contributo alla realizzazione del progetto, per poi effettuare dei **merge** che sincronizzassero il lavoro eseguito man mano.

github.com/Rad03anto/RubricaContattiSE

	michelesavino	Test prima parte e modifiche di codice	f89e849 · 3 hours ago	🕒 91 Commits
📁	Documentazione progetto	Aggiornamento doxygen e codice	yesterday	
📁	RubricaContattiProject	Test prima parte e modifiche di codice	3 hours ago	
📁	html	Test prima parte e modifiche di codice	3 hours ago	
📁	latex	Test prima parte e modifiche di codice	3 hours ago	
📄	.gitignore	Test prima parte e modifiche di codice	3 hours ago	
📄	Doxyfile	Test prima parte e modifiche di codice	3 hours ago	
📄	README.md	Update README.md	3 weeks ago	

📖 README

📖 Rubrica Telefonica - Gruppo 23

- 🔧 Progetto di Ingegneria del Software, Anno Accademico 2024-25
- 🎓 Corso di Ingegneria Informatica, coorte 2022-23

📘 Il progetto

Applicazione per la gestione dei contatti di una rubrica contenente numeri di telefono e indirizzi e-mail realizzata in Java attraverso una GUI in JavaFX. Gruppo 23.

💻 Tecnologie adoperate

JAVA JAVAFX APACHE MAVEN DOXYGEN JUNITS GITHUB

👥 Membri

- [Luca Nazzaro](#)
- [Antonio Radesca](#)
- [Michele Savino](#)
- [Alessandro Scandone](#)

📖 Readme
📈 Activity
⭐ 0 stars
👁 2 watching
🍴 0 forks
Report repository





Releases

No releases published
[Create a new release](#)

Packages

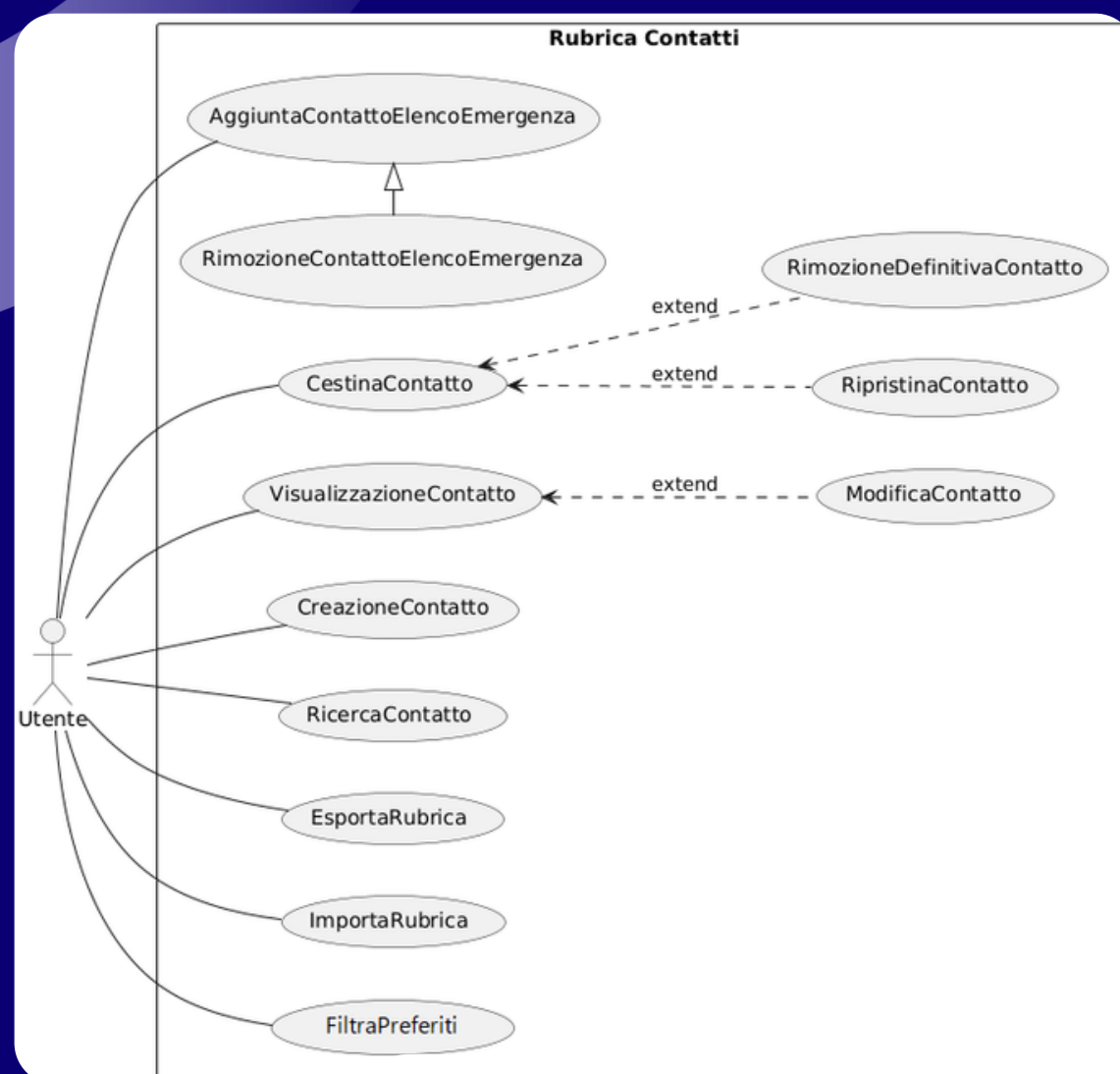
No packages published
[Publish your first package](#)

Contributors 4

-  michelesavino Michele Savino
-  Rad03anto
-  alescand1 Alessandro Scandone
-  LucaNazz

Languages

HTML 44.4% TeX 32.2%
Java 6.3% JavaScript 5.9%
CSS 5.3% PostScript 3.6%
Other 0.3%



REQUIREMENTS ENGINEERING

In questa seconda fase l'obiettivo è stato realizzare un documento in formato libero che includesse la **descrizione dei casi d'uso**, il **diagramma dei casi d'uso** e informazioni che dessero una chiara idea dei requisiti necessari per creare un'applicazione robusta, user-friendly e che rispettassero verificabilità, comprensibilità, tracciabilità, coerenza e completezza. Il tutto, ovviamente, sempre caricato su GitHub.

2. Tabella dei requisiti

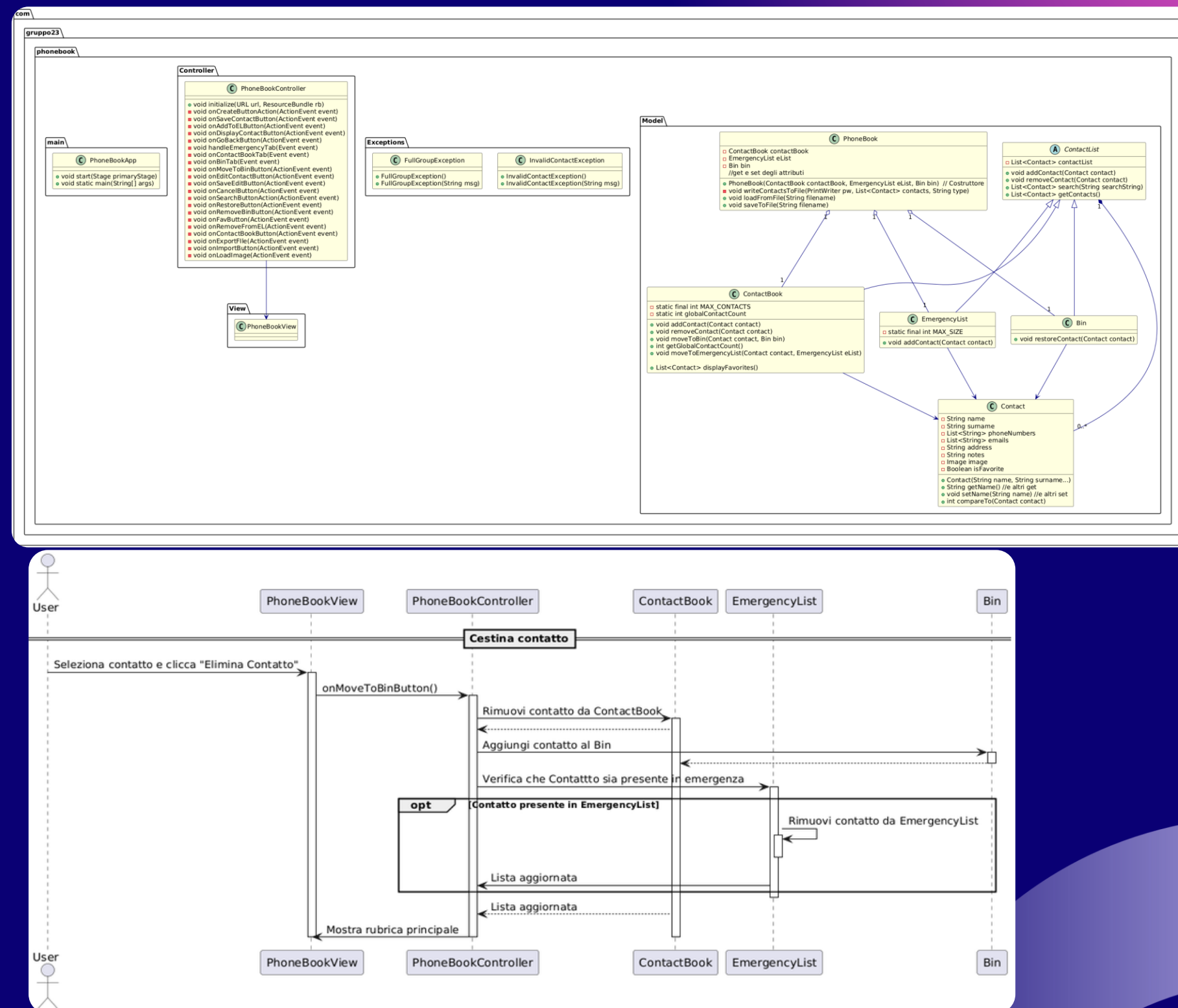
2.1 La tabella dei requisiti è stata strutturata nel seguente modo:

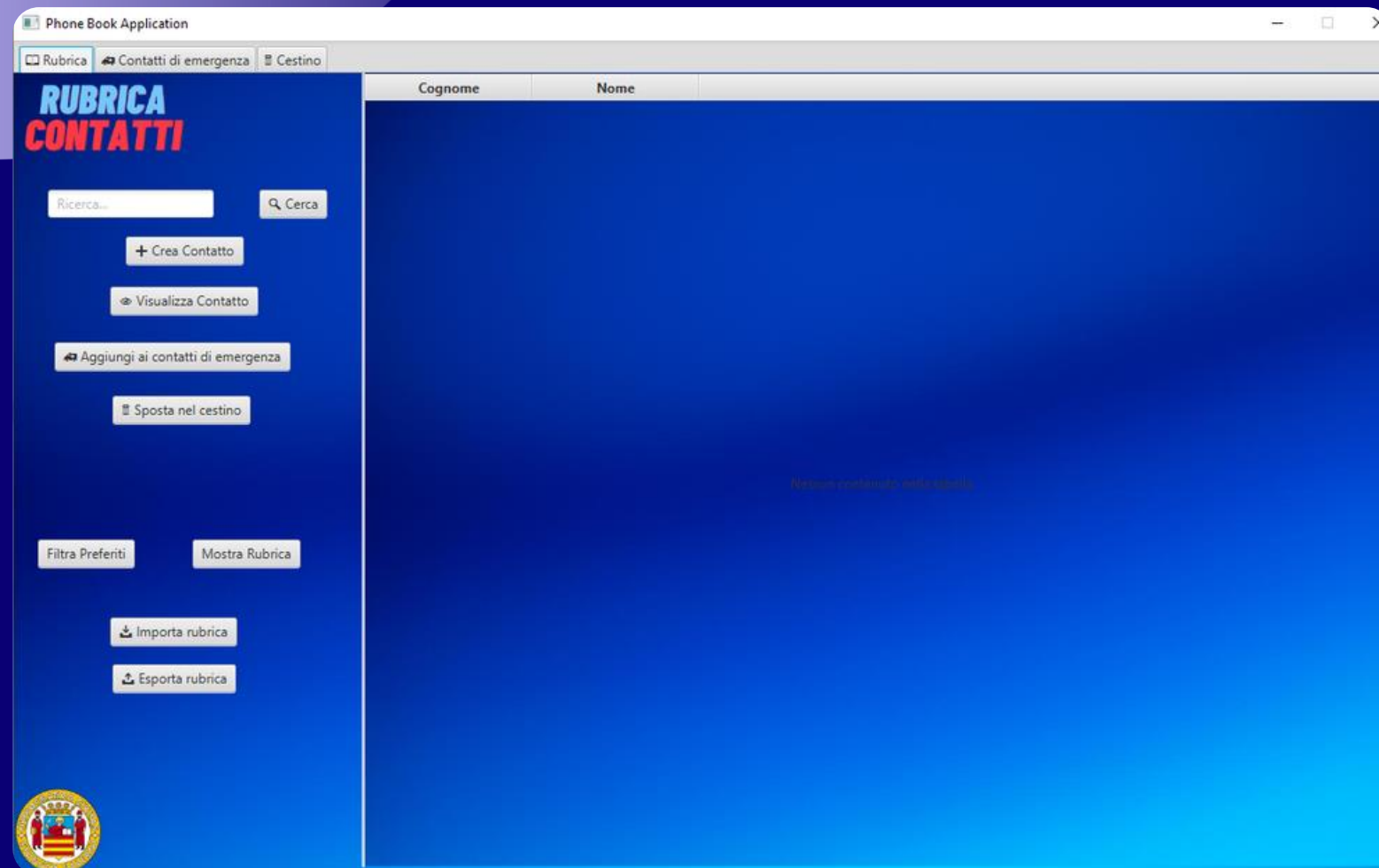
- Categoria per migliorare organizzazione e leggibilità
- Prefisso per una categorizzazione ulteriore dei requisiti. Seguono questa legenda:
 1. Funzionalità individuale IF **IF**;
 2. Business flow BF **BF**;
 3. Dati e formato dei dati DF **DF**;
 4. Interfaccia utente UI **UI**;
 5. Requisiti non funzionali RNF **RNF**.
- ID come identificativo del requisito;
- Nome del requisito;
- Descrizione in Linguaggio Naturale;
- Business Value, Rischio Tecnico e Priorità: quest'ultima è stata calcolata in base ai valori dei primi due.



DESIGN DI DETTAGLIO

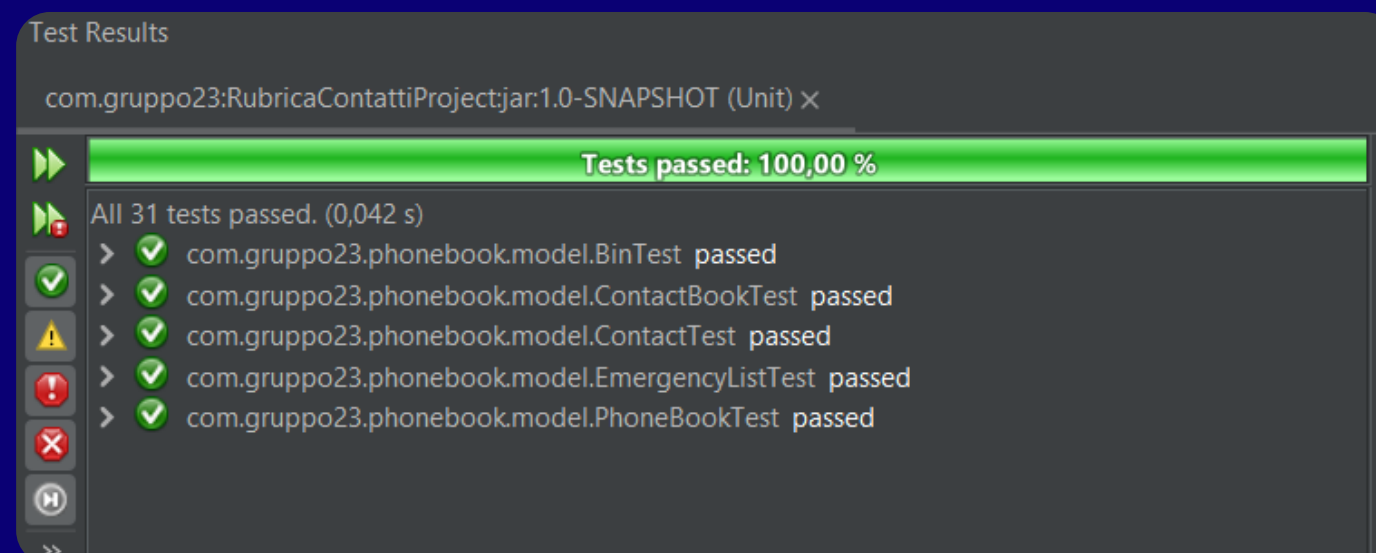
Nella fase di **Design** è stata realizzata la progettazione del sistema. In seguito è stato redatto un documento in formato libero che includesse: **diagrammi delle classi**, **diagrammi di sequenza** per le interazioni più significative, **commenti** ai diagrammi, discutendo le scelte effettuate in termini di **coesione**, **accoppiamento** e **principi di buona progettazione**. In seguito sono stati creati gli scheletri delle classi e documentate le interfacce pubbliche delle classi usando **doxygen**.





IMPLEMENTAZIONE E TESTING

In quest'ultima fase sono state implementate le classi progettate, realizzati **Unit Test** automatizzati tramite **JUnit** per tutte le classi e verificato che tutti i test fossero superati. Per quanto riguarda il **PhoneBookController** è stato realizzato un documento, reperibile nel repository su GitHub, all'interno del quale vengono mostrati, proprio sull'applicazione, tutte le funzionalità a disposizione dell'utente e i risultati che producono.





TIME TO LAUNCH PHONEBOOK!