Università di Federico II

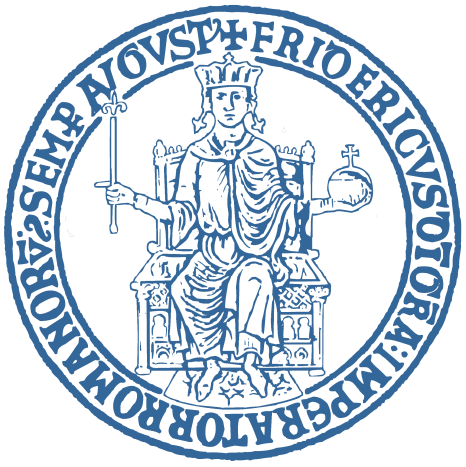
Object Orientation

Progettazione e Sviluppo di una Rubrica Telefonica Avanzata

Authors
Oleksandr Sosovskyy
Francesco Magri

Professor
Prof. Porfirio
Tramontana

Settembre 2022



Indice

Capitolo 1

Introduzione

1.1 Descrizione

Si progetterà un'applicativo per la gestione di una Rubrica Telefonica Avanzata. A seguito di alcune problematicità riscontrate nell'analisi della traccia col prof. Silvio Barra, si è deciso consapevolmente di optare per un'implementazione "multi-rubrica": è ammessa la possibilità per più persone di gestire una rubrica personale, i cui contatti mantengono la completa indipendenza informativa rispetto a eventuali omonimi presenti in altre rubriche. Il Sistema garantisce tutte le funzionalità principali indicate nella traccia (come la memorizzazione e gestione delle informazioni dei contatti, la ricerca dei contatti per nome, email, account di messaging e per numero di telefono oppure il reindirizzamento della chiamata al secondo numero indicato) e altre funzionalità secondarie non esplicitate ma aggiunte per una migliore usabilità, quindi senza la necessità di intervenire direttamente sulla Base di Dati per specifiche operazioni perché gestibili dall'applicativo. Infine, un indirizzo elettronico associa un contatto ai suoi account presso i sistemi di messaging. Il Sistema ignora il modo in cui le suddette informazioni esterne vengano reperite: si ammette che siano recuperate dal dispositivo in uso magari attraverso delle API esterne, inserite nella Base di dati e, alla dichiarazione della corretta email, associate al contatto corrispondente. L'applicativo si occupa solo del terzo e ultimo passaggio indicato.

1.2 Traccia

Si sviluppi un sistema informativo, composto da una base di dati relazionale e da un applicativo Java dotato di GUI (Swing o JavaFX), per la gestione di una rubrica telefonica avanzata. La rubrica deve essere in grado di consentire la memorizzazione e visualizzazione di dati riguardanti contatti. Per ogni contatto la rubrica deve ricordare il nome e cognome della persona cui si riferisce. Inoltre, può essere eventualmente memorizzata anche una foto (della quale bisogna ricordare il percorso sul disco dove può essere reperita). I contatti possono essere organizzati in gruppi (dotati di un nome) che comprendono uno o più contatti. gruppi possono anche intersecarsi tra loro, cosicché un contatto può anche appartenere a più di un gruppo. Per ogni contatto, bisogna mantenere un insieme (eventualmente vuoto) dei suoi indirizzi di posta elettronica. Inoltre, bisogna anche ricordare tutti gli account a sistemi di messaging. Per ognuno di questi account, bisogna ricordare il fornitore (ad esempio Whatsapp, Telegram, Teams, etc.), il nickname dichiarato dal contatto, la frase di benvenuto e l'indirizzo e-mail collegato a tale account. L'indirizzo e-mail deve essere necessariamente tra gli account di posta già salvati per il contatto. Per ogni contatto possono poi essere mantenuti suoi indirizzi fisici (per ognuno di essi, bisogna conoscere Via, Città, CAP, Nazione). In particolare, deve essere obbligatoriamente definito un indirizzo principale, mentre possono esserci uno o più eventuali altri indirizzi secondari. Infine, per ogni contatto devono essere mantenuti i suoi eventuali numeri telefonici, distinguendo tra telefoni fissi e mobili. Per ogni telefono mobile, può essere indicato un telefono fisso cui verranno reindirizzate eventuali chiamate senza risposta e, analogamente per ogni telefono fisso deve essere indicato un telefono mobile per il reindirizzamento. Ogni contatto deve avere almeno un telefono fisso e un telefono mobile. Si tenga conto che il sistema accetta contatti che dichiarino stessi indirizzi fisici o numeri telefonici ma non accetta contatti che dichiarano stesse e-mail. Il sistema deve fornire funzionalità per aggiungere nuovi contatti e creare, modificare o eliminare ognuna delle informazioni in esso contenute. Inoltre, devono essere realizzate funzionalità per la ricerca dei contatti per nome, per email, per account di messaging e per numero di telefono.

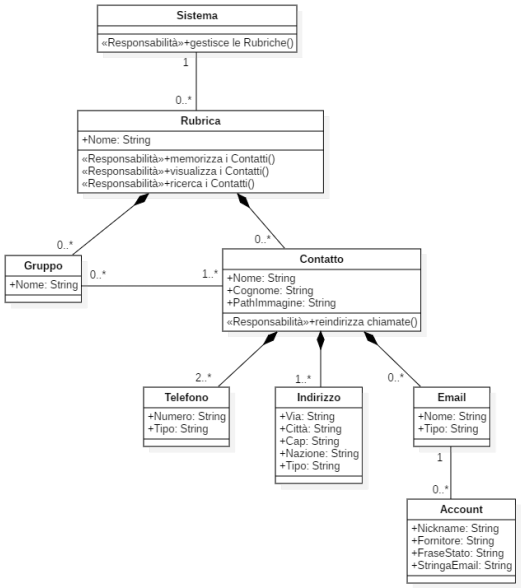
Capitolo 2

Diagrammi

2.1 Diagramma delle classi

La modellazione porta a un diagramma gerarchico in cui si è scelto di rappresentare delle relazioni tra diverse classi attraverso composizioni per esprimere che un aggregato non può esistere senza il corrispettivo aggregante. L'aggiunta di una classe *Sistema* è necessaria per gestire una molteplicità di rubriche non comunicanti tra loro.

2.1.1 Diagramma

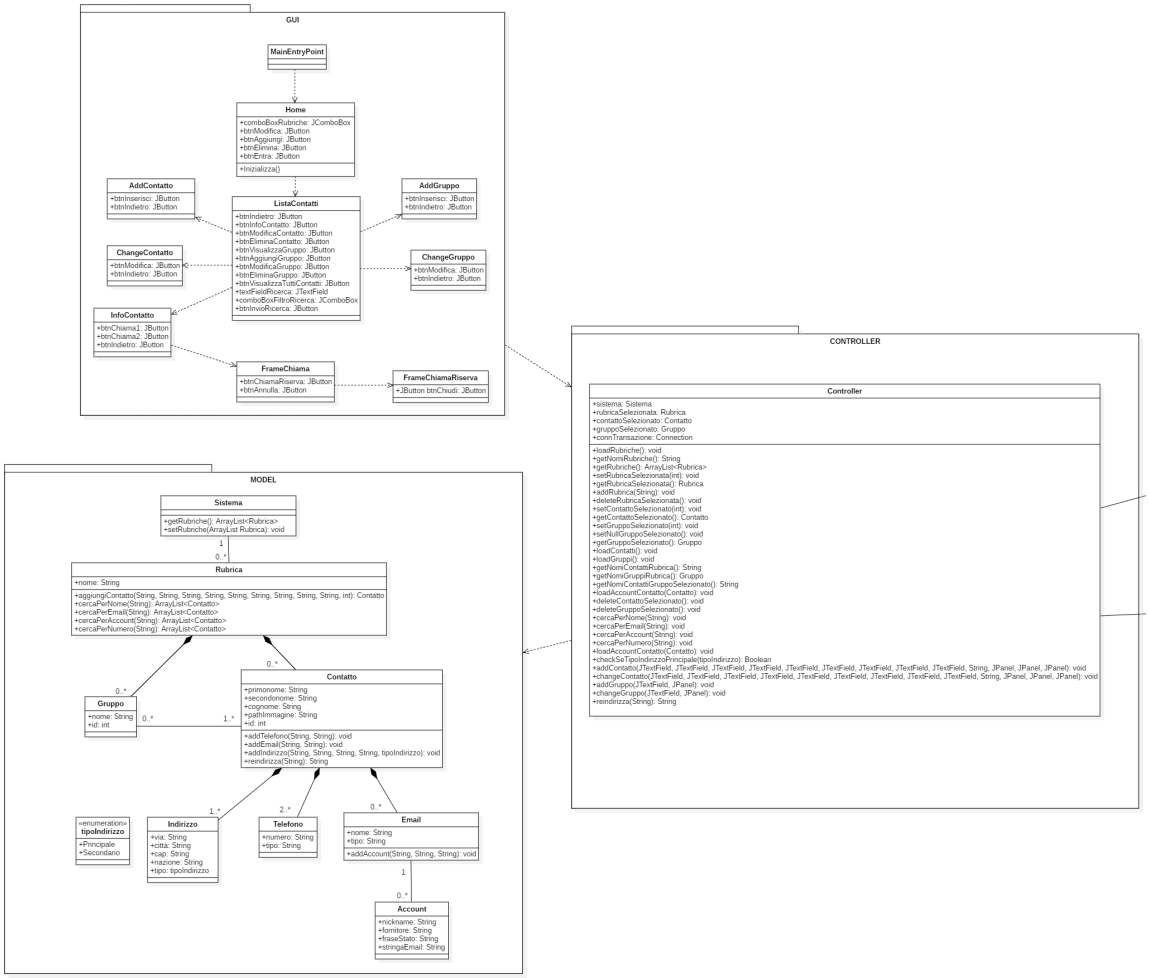


2.2 Diagramma della soluzione

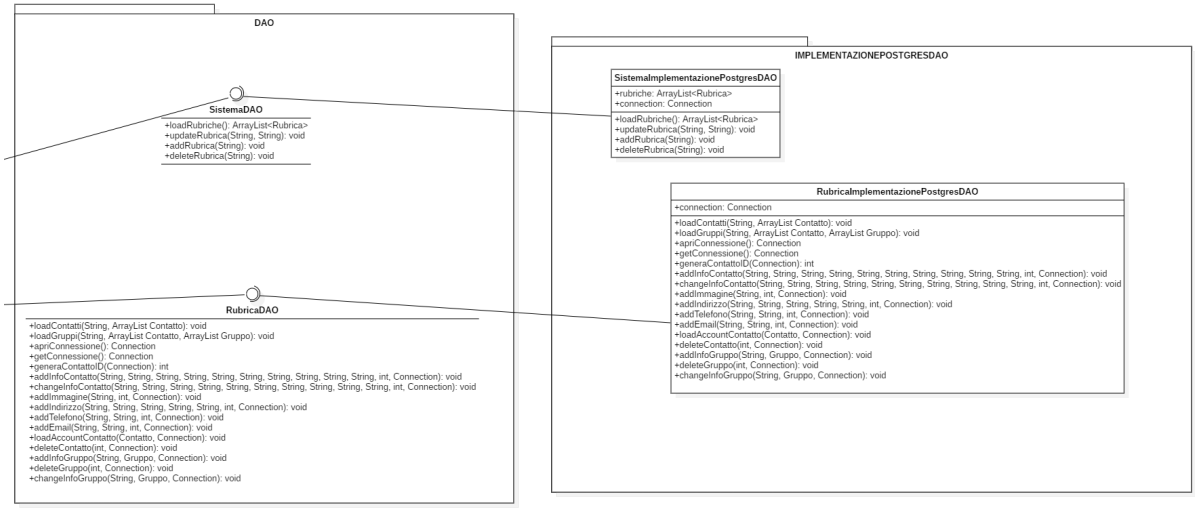
Si è seguito attentamente (a costo di frequenti refactoring) il pattern BCE(D) per mantenere la completa separazione tra GUI e Model. Nel dettaglio, i package individuati sono i seguenti:

- **GUI**: include l'insieme dei frame comunicanti tra loro che rappresentano l'intera interfaccia grafica dell'applicativo. A causa della grande quantità di componenti grafici, nel Diagramma sono indicati solo quei pulsanti e altri componenti che corrispondono a precise e intuitive funzionalità dell'interfaccia.
- **Controller**: include solo una classe che media la comunicazione tra Model, Gui e Database.
- **Model**: include le classi già presentate nel Diagramma del Problema, a meno di alcune modifiche che rappresentano un'implementazione più di basso livello.
- **DAO**: include le interfacce delle classi e dei metodi responsabili della comunicazione diretta col Database
- **Implementazione DAO**: include le implementazioni delle interfacce del DAO per un Database Postgres.

2.2.1 Diagramma GUI - Controller - Model



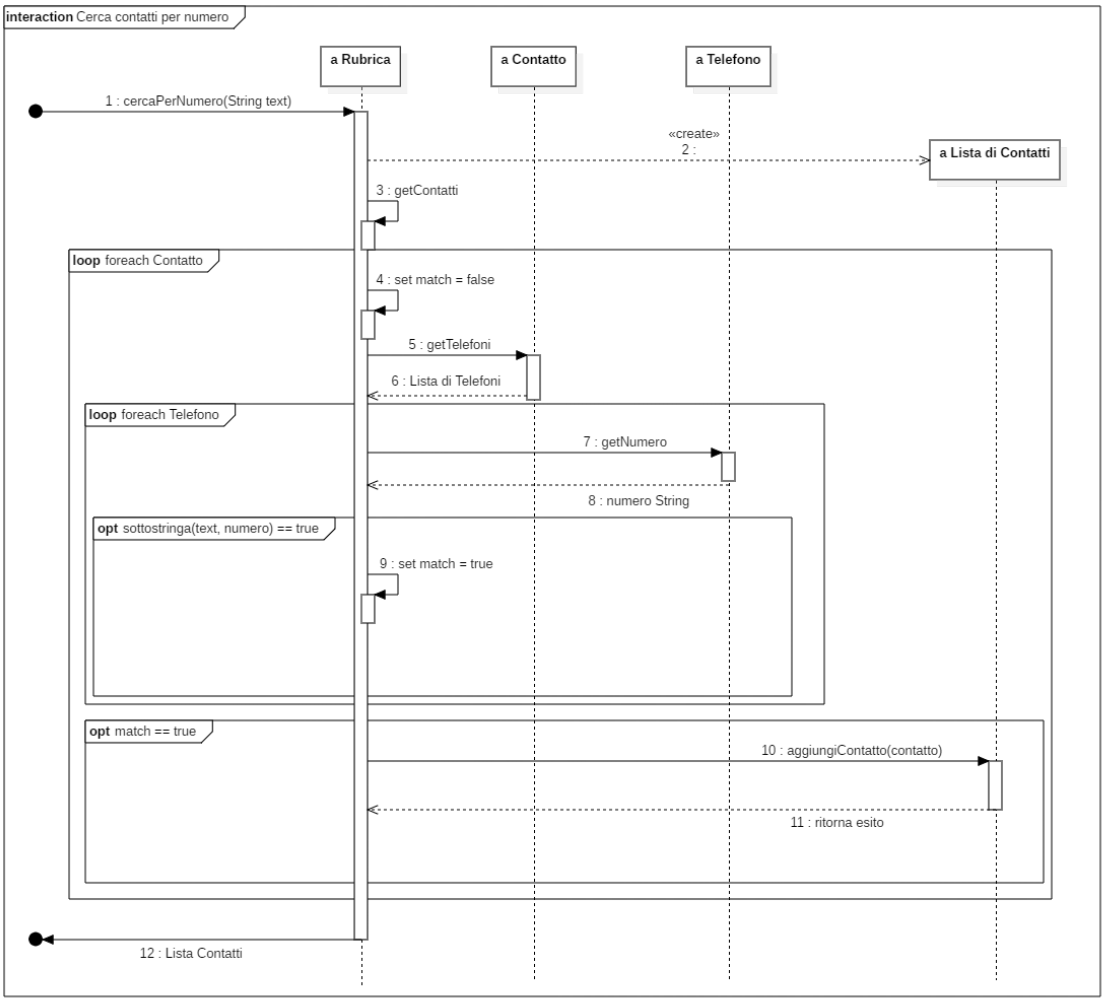
2.2.2 Diagramma DAO - Implementazione DAO



2.3 Sequence Diagram della ricerca per numero

Il seguente Sequence Diagram rappresenta l'algoritmo di ricerca dei contatti per numero. La ricerca avviene a livello di una rubrica. Per ogni contatto della rubrica si confronta ciascun suo numero di telefono con la parola chiave digitata dall'utente nella barra di ricerca (*text*). Nel caso in cui text è sottostringa di un numero, allora il confronto ha successo ($match == true$) e il contatto corrispondente è aggiunto a una lista. Al termine di tutti i confronti, l'algoritmo restituisce la lista dei contatti trovati.

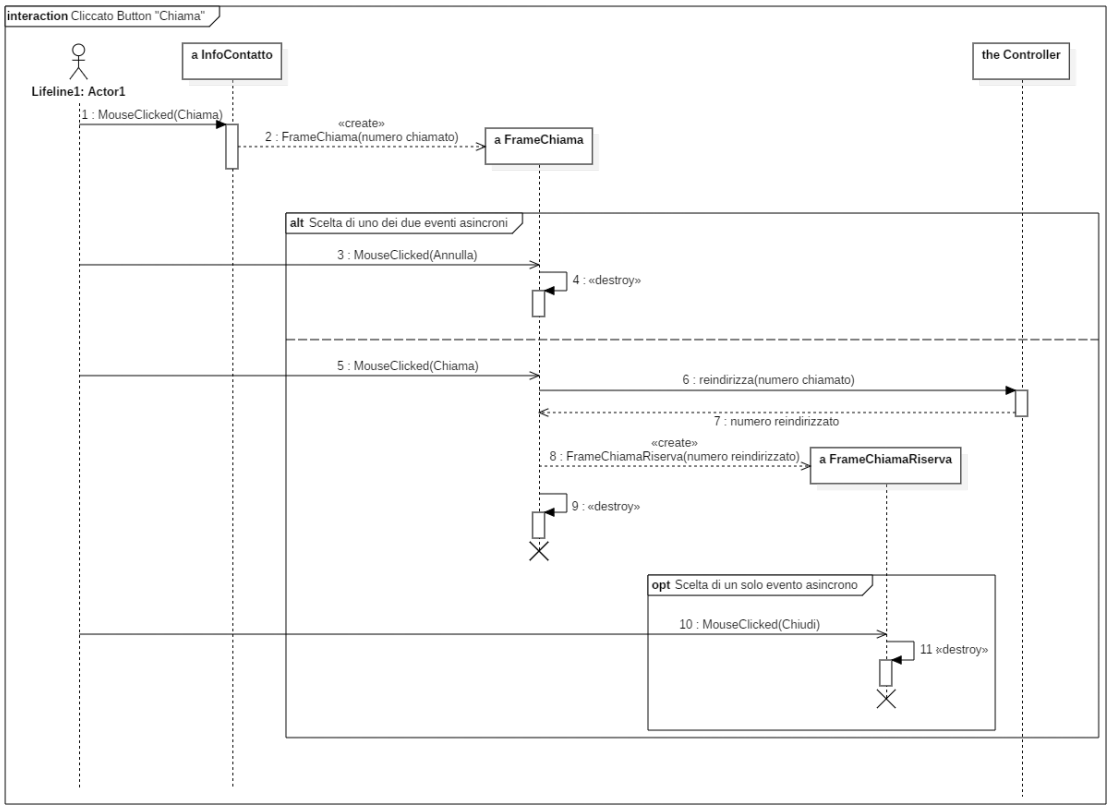
2.3.1 Diagramma



2.4 Sequence Diagram del reindirizzamento di una chiamata

Il seguente Sequence Diagram rappresenta una successione di interazioni asincrone dell'utente con la GUI quando dal frame *InfoContatto* l'utente clicca il pulsante per chiamare un numero mobile o fisso. Con l'obiettivo della sinteticità, generalità e completezza, si rappresenta una versione semplificata rispetto all'implementazione reale dell'algoritmo, dove sono indicate solo le variabili e metodi ritenuti indispensabili alla spiegazione del funzionamento della simulazione di una chiamata. Quando l'utente decide di chiamare un numero (fisso o mobile) viene costruito *FrameChiama* che rappresenta una prima chiamata destinata a fallire. L'utente può quindi cliccare sul pulsante "Annulla" per tornare indietro, oppure su "Chiama" per chiamare un secondo numero. Nel secondo caso, viene invocato il controller che riceve il numero chiamato e restituisce un secondo numero dello stesso contatto a cui reindirizzare la chiamata. Un nuovo frame, *FrameChiamaRiserva*, viene dunque aperto alla distruzione conseguente del primo e simula una seconda chiamata. Al fallimento anche di quest'ultima, l'utente può solo premere il pulsante "Chiudi" per distruggere l'ultimo frame e tornare a InfoContatto.

2.4.1 Diagramma



Capitolo 3

Manuale d'uso

3.1 Installazione

Per usare l'applicativo, si suggerisce di seguire i seguenti passaggi:

1. importare il progetto da GitHub seguendo questo link;
2. se non presente, scaricare PostgreSQL;
3. creare un Database di nome **Rubrica** con owner **postgres** e password **1234** , oppure modificare gli attributi *nomeutente*, *password* e *url* della classe ConnessioneDatabase con i propri valori;
4. eseguire questa query per costuire lo schema del Database;
5. eseguire questa query per popolare il Database.

3.2 Esempi di funzionalità notevoli

Si consiglia di provare le seguenti funzionalità:

- per il reindirizzamento della chiamata si prema il pulsante "Visualizza contatto" presso la lista di contatti per aprire la scheda contatto da cui poi si può chiamare il numero fisso o il numero mobile (l'uno reindirizza sempre l'altro);
- per verificare l'associazione degli account a un contatto tramite email si provi ad aggiungere a un contatto uno dei seguenti indirizzi di posta elettronica: **altra@gmail.com**, **a@gmail.com** o **test@libero.com**;
- per la ricerca di contatti secondo un criterio, accedere alla lista di contatti di una rubrica, selezionare il criterio di ricerca presso la combobox collocata in alto e nell'adiacente barra di ricerca digitare la parola chiave (può anche non essere esatta perché il match avviene per sottostringa).

3.3 Documentazione Javadoc

Per la documentazione esatta delle classi, dei metodi e degli attributi è stato generato un Javadoc del codice sorgente. Per consultare il Javadoc accedere a questo link.