

UNIVERSITÀ DEGLI STUDI DI NAPOLI FEDERICO II  
SCUOLA POLITECNICA E DELLE SCIENZE DI BASE  
DIPARTIMENTO DI INGEGNERIA ELETTRICA E TECNOLOGIE DELL'INFORMAZIONE



CORSO DI LAUREA IN INFORMATICA  
INSEGNAMENTO DI BASI DI DATI E OBJECT ORIENTATION  
ANNO ACCADEMICO 2021/2022

Progettazione e sviluppo di una Base di dati relazionale  
ai fini della creazione di una rubrica telefonica avanzata

Giuseppe Longobardo  
NN86003725

Andrea Pinto  
N86003824

Sebastiano Sannino  
N86003772

Docenti: Silvio Barra & Porfirio Tramontana

12 Giugno 2022

# Indice

|          |  |           |
|----------|--|-----------|
| <b>1</b> | <b>Analisi generale del progetto</b>                   | <b>4</b>  |
| 1.1      | Descrizione sintetica e analisi dei problemi . . . . . | 4         |
| 1.2      | Scelte implementative . . . . .                        | 4         |
| 1.3      | Database . . . . .                                     | 5         |
| <b>2</b> | <b>Progettazione concettuale</b>                       | <b>6</b>  |
| 2.1      | Class Diagram . . . . .                                | 6         |
| 2.2      | EnServizio . . . . .                                   | 7         |
| 2.3      | Partecipazione 1..* - 0..* . . . . .                   | 7         |
| 2.4      | Attributo composto : PostaElettronica . . . . .        | 7         |
| 2.5      | Analisi delle ridondanze . . . . .                     | 8         |
| 2.6      | Attributi mutipi . . . . .                             | 8         |
| 2.7      | Attributi composti . . . . .                           | 8         |
| 2.8      | Generalizzazioni . . . . .                             | 8         |
| 2.9      | Analisi degli identificativi . . . . .                 | 9         |
| 2.10     | Class Diagram Ristrutturato . . . . .                  | 10        |
| 2.11     | Dizionari . . . . .                                    | 11        |
| 2.11.1   | Dizionario delle classi . . . . .                      | 11        |
| 2.11.2   | Dizionario delle associazioni . . . . .                | 13        |
| 2.11.3   | Dizionario dei vincoli . . . . .                       | 15        |
| <b>3</b> | <b>Progettazione logica</b>                            | <b>19</b> |
| 3.1      | Schema logico . . . . .                                | 20        |
| <b>4</b> | <b>Progettazione fisica</b>                            | <b>21</b> |
| 4.1      | Definizioni tabelle . . . . .                          | 21        |
| 4.1.1    | Tabella contatto . . . . .                             | 21        |
| 4.1.2    | Tabella gruppo . . . . .                               | 22        |
| 4.1.3    | Tabella partecipazione . . . . .                       | 22        |
| 4.1.4    | Tabella sistema di messaging . . . . .                 | 22        |
| 4.1.5    | Tabella e-mail . . . . .                               | 23        |
| 4.1.6    | Tabella indirizzo principale . . . . .                 | 23        |
| 4.1.7    | Tabella indirizzo secondario . . . . .                 | 24        |
| 4.1.8    | Tabella numero fisso . . . . .                         | 24        |
| 4.1.9    | Tabella numero mobile . . . . .                        | 25        |
| 4.1.10   | Tabella reindirizzamento . . . . .                     | 25        |
| 4.2      | Definizioni procedure . . . . .                        | 26        |
| 4.2.1    | Procedura check equal (Numeri di telefono) . . . . .   | 26        |

|          |  |           |
|----------|--|-----------|
| 4.2.2    | Check unico (Indirizzo principale)         | 26        |
| 4.2.3    | Check indirizzo (Indirizzi secondari)      | 27        |
| 4.2.4    | Procedura check equal (Numeri di telefono) | 27        |
| 4.2.5    | Cancella gruppo                            | 28        |
| 4.2.6    | Cancella contatto cascade                  | 29        |
| 4.2.7    | Cancella contatto cascade (2)              | 30        |
| 4.2.8    | Cancella contatto cascade (3)              | 31        |
| 4.3      | Definizioni trigger                        | 32        |
| 4.3.1    | Check equal MOBILE                         | 32        |
| 4.3.2    | Check equal FISSO                          | 32        |
| 4.3.3    | Check indirizzo unico                      | 32        |
| 4.3.4    | Check indirizzo principale                 | 32        |
| 4.3.5    | Check gruppo                               | 33        |
| 4.3.6    | Cancella contatto                          | 33        |
| <b>5</b> | <b>Applicativo Java</b>                    | <b>34</b> |
| 5.1      | HomePage                                   | 34        |
| 5.2      | Visualizzazione contatto                   | 35        |
| 5.3      | Modifica contatto                          | 36        |
| 5.4      | Crea contatto                              | 37        |
| 5.5      | Crea gruppo                                | 38        |
| 5.6      | Area privata                               | 39        |
| <b>6</b> | <b>Riflessioni finali</b>                  | <b>40</b> |
| 6.1      | Le nostre idee                             | 40        |

# Capitolo 1

## Analisi generale del progetto

### 1.1 Descrizione sintetica e analisi dei problemi

Si progetterà una base di dati relazionale per la gestione di una *rubrica telefonica avanzata*.

La rubrica sarà in grado di consentire la memorizzazione e la visualizzazione dei dati riguardanti i *contatti*.

Quest'ultimi potranno essere organizzati in *gruppi* da uno o più partecipanti.

Per ciascun contatto bisognerà obbligatoriamente definire un *indirizzo principale* e almeno un *numero fisso* e un *numero mobile*.

Non sarà possibile inoltre, creare un contatto con una e-mail già inserita nella base di dati.

Infine sarà presente anche un area protetta, dove bisognerà inserire una password per ottenere le informazioni dei contatti privati.

La GUI dell'applicativo Java sarà svolto con *Java Swing*.

### 1.2 Scelte implementative

L'IDEE utilizzato per programmare in Java è stato *IntelliJ*.

Il layout principali che sono stati utilizzati nella creazione della GUI sono : *Border Layout* e *GridLayoutManager (IntelliJ)*.

L'applicativo ha una finestra ridimensionabile con un settaggio minimo a *1280x720*.

Tutte le immagini in formato *png* che sono state utilizzate nell'applicazione, saranno salvate nella cartella *.images*.

### **1.3 Database**

Il database è *Hostato online* e tutte le query di creazione e gestione delle tabelle, tramite *trigger* e *procedure*, sono state eseguite su *PGAdmin 4*.

## Capitolo 2

# Progettazione concettuale

### 2.1 Class Diagram

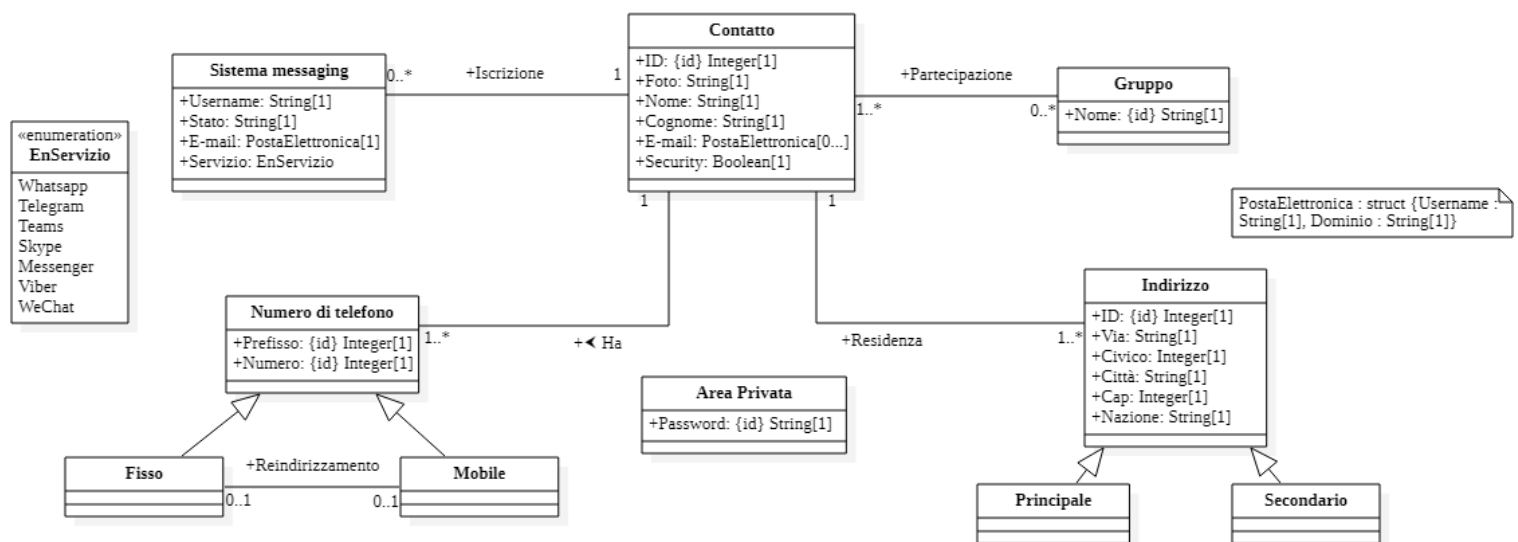


Figura 2.1: Class diagram

# Precisazioni sul Class Diagram

## 2.2 EnServizio

Si è scelto di prendere in considerazione solo una ristretta cerchia di sistemi di messaging, in modo da escludere i sistemi più "obsoleti" e/o meno conosciuti ed utilizzati.

## 2.3 Partecipazione 1..\* - 0..\*

Il motivo per il quale la classe *Gruppo* ha l'obbligo di partecipazione con contatto è perchè se nel gruppo non vi sono contatti, esso verrà automaticamente eliminato.

## 2.4 Attributo composto : PostaElettronica

Si è deciso di creare l'attributo PostaElettronica e di inserirlo in contatto, siccome sarà una delle informazioni principali di quest'ultimo. Sarà poi illustrato, nella **sezione 2.2**, in che modo verrà ristrutturato, creando la classe con le relative associazioni

# Ristrutturazione

## 2.5 Analisi delle ridondanze

Nel Class Diagram non vi è la presenza di ridondanze.

## 2.6 Attributi mutipli

E' necessario gestire l'attributo multiplo, dell'attributo composto *PostaElettronica* presente nell classe Contatto. Si provvederà con la creazione di una classe *E-mail* che avrà un associazione con *Contatto* del tipo [E-mail : 1 - Contatto : 0..\*], in modo che un contatto possa contenere più mail o possa anche non contenerne alcuna.

## 2.7 Attributi composti

Verrà gestito l'attributo composto *PostaElettronica* presente nella classe contatto. Si provvederà con la creazione di un classe con gli attributi :

- *Username* di tipo String che conterrà l'username della posta elettronica del contatto.
- *Dominio* di tipo String che conterrà il dominio della posta elettronica del contatto.

## 2.8 Generalizzazioni

Sia per quanto riguarda la generalizzazione *Numero di telefono* e *Indirizzo* si è scelto di accorpare la classe generale nelle classi figlio,



siccome :

- Distinzione netta, e recupero facilitato di informazioni sia per i numeri di telefono (Mobili e Fissi) sia per gli indirizzi (Primario e Secondari).
- Non vi è la necessità di creare un attributo discriminante nella classe generale.
- Non vi è la necessità di trovare una soluzione per creare l'associazione *Reindirizzamento* nella classe generale.
- Vi è un'unica associazione sia per *Numero di telefono* e *Indirizzo*, dunque il numero di associazioni rimarrà sempre ristretto.

## 2.9 Analisi degli identificativi

Per alcune classi, ove non vi è la possibilità di creare una chiave primaria composta e/o una chiave primaria con un singolo attributo, si è scelto di inserire chiavi primarie surrogate, quest'ultime sono identificate dal loro nome: *ID*.

D'altronde per le classi dove è possibile identificare una chiave primaria in maniera netta, o tramite l'ausilio della composizione di più attributi si è optato di crearle.

## 2.10 Class Diagram Ristrutturato

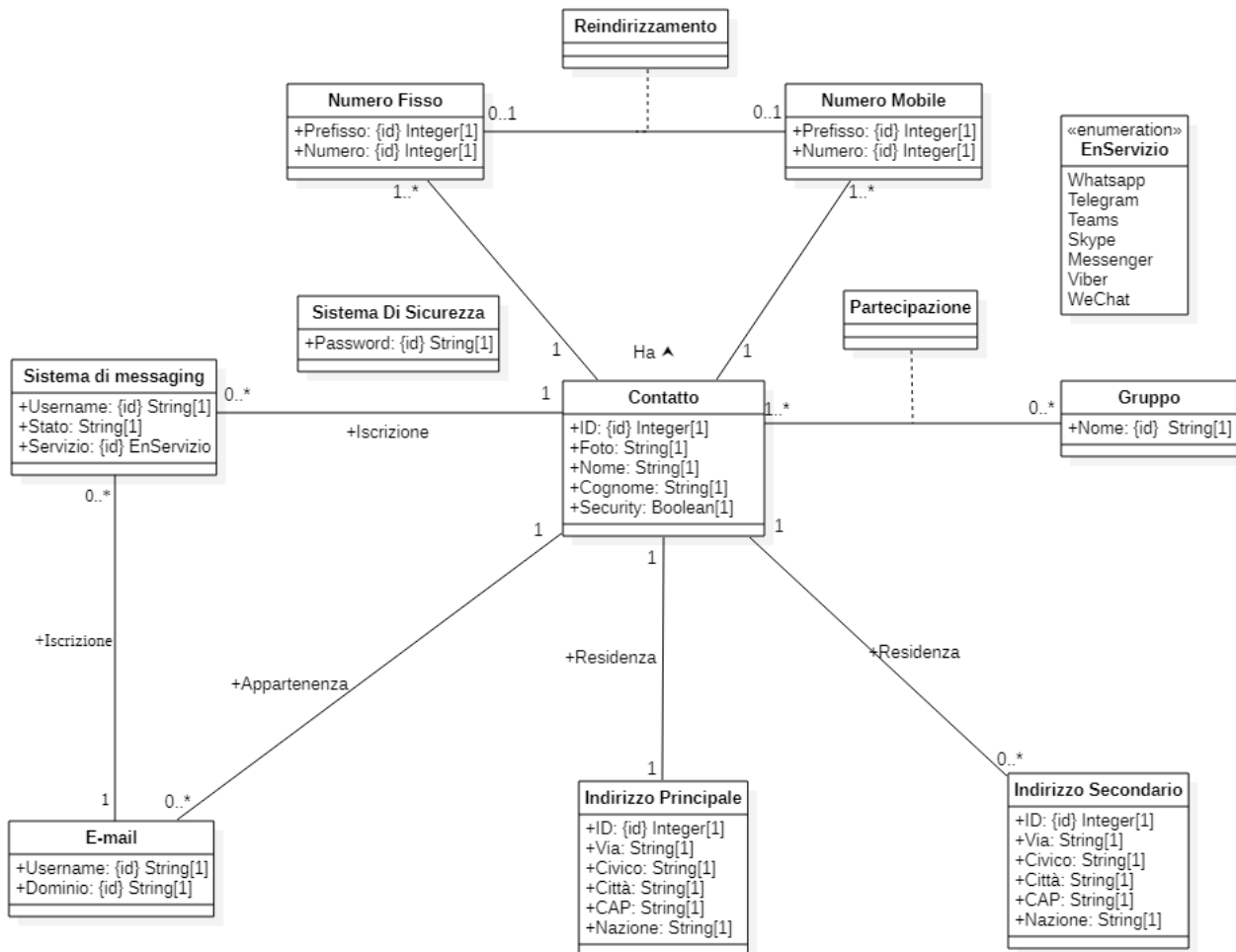


Figura 2.2: Class Diagram ristrutturato

## 2.11 Dizionari

### 2.11.1 Dizionario delle classi

| CLASSE               | DESCRIZIONE  | ATTRIBUTI  |
|----------------------|--|--|
| Contatto             | Descrittore di un contatto                                     | <b>ID</b> ( <i>Integer</i> ): Chiave primaria.<br>Identifica univocamente ciascun contatto.<br><b>Foto</b> ( <i>String</i> ): Individua il path della foto del contatto<br><b>Nome</b> ( <i>String</i> ): Il nome del contatto<br><b>Cognome</b> ( <i>String</i> ): Il cognome del contatto<br><b>Security</b> ( <i>Boolean</i> ): Segnala con true se il contatto è privato, false altrimenti |
| Gruppo               | Descrittore dei gruppi   | <b>Nome</b> ( <i>String</i> ): Chiave primaria.<br>Indica il nome del gruppo.<br>(Non vi possono essere due gruppi con nomi uguali.)   |
| Sistema di sicurezza | Gestisce l'area privata tramite il salvataggio di una password | <b>Password</b> ( <i>String</i> ): Chiave primaria.<br>Indica la password (Che sarà sempre unica) dell'area privata.   |
| Sistema di Messaging | Descrittore di un sistema di messaging                         | <b>Username</b> ( <i>String</i> ): Chiave primaria composta.<br>Indica l'username con il quale l'utente si è iscritto al sistema.<br>Identifica univocamente ciascun contatto.<br><b>Stato</b> ( <i>String</i> ): Indica lo stato del profilo su quel sistema di messaging<br><b>Servizio</b> ( <i>EnServizio</i> ): Chiave primaria composta.<br>Indica il nome del servizio di messaging     |
| Numero Fisso         | Descrittore di un numero fisso                                 | <b>Prefisso</b> ( <i>Integer</i> ): Chiave primaria composta.<br>Indica il prefisso del numero.<br><b>Numero</b> ( <i>Integer</i> ): Chiave primaria composta.<br>Indica il numero.  |

|                      |  |  |
|----------------------|--|--|
| Numero Mobile        | Descrittore di un numero mobile                      | <p><b>Prefisso</b>(<i>Integer</i>) : Chiave primaria composta. Indica il prefisso del numero.</p> <p><b>Numero</b> (<i>Integer</i>) : Chiave primaria composta. Indica il numero.</p>  |
| E-mail               | Descrittore della posta elettronica di un contatto   | <p><b>Username</b>(<i>String</i>) : Chiave primaria composta. Indica l'username della mail.</p> <p><b>Dominio</b>(<i>String</i>) : Chiave primaria composta. Indica il dominio della mail.</p>   |
| Indirizzo principale | Descrittore dell'indirizzo principale di un contatto | <p><b>ID</b>(<i>Integer</i>) : Chiave primaria. Identifica univocamente ciascun indirizzo principale.</p> <p><b>Via</b>(<i>String</i>) : Indica la via</p> <p><b>Civico</b>(<i>String</i>) : Indica il civico.</p> <p><b>Città</b>(<i>String</i>) : Indica la città.</p> <p><b>CAP</b>(<i>String</i>) : Indica il CAP.</p> <p><b>Nazione</b>(<i>String</i>) : Indica la nazione.</p> |
| Indirizzo secondario | Descrittore dell'indirizzo secondario di un contatto | <p><b>ID</b>(<i>Integer</i>) : Chiave primaria. Identifica univocamente ciascun indirizzo principale.</p> <p><b>Via</b>(<i>String</i>) : Indica la via</p> <p><b>Civico</b>(<i>String</i>) : Indica il civico.</p> <p><b>Città</b>(<i>String</i>) : Indica la città.</p> <p><b>CAP</b>(<i>String</i>) : Indica il CAP.</p> <p><b>Nazione</b>(<i>String</i>) : Indica la nazione.</p> |

Figura 2.3: Dizionario delle classi

## 2.11.2 Dizionario delle associazioni

| ASSOCIAZIONE  | DESCRIZIONE  | CLASSI COINVOLTE  |
|---------------|--|---|
| Iscrizione    | Stabilisce in che modo un contatto può interagire con i sistemi di messaging | <b>Contatto (1)</b> : Ogni sistema di messaging si riferisce solo ad un contatto.<br><b>Sistema di messaging(0..*)</b> : Un contatto può non essere iscritto a nessun sistema di messaging oppure può essere iscritto a più di un sistema di messaging. |
| Iscrizione(2) | Stabilisce in che modo un sistema di messaging interagire con le e-mail      | <b>E-mail(1)</b> : Ogni sistema di messaging ha associata una sola e-mail.<br><b>Sistema di messaging(0..*)</b> : Un email può non essere utilizzata per le iscrizioni oppure può esserlo per più di un iscrizione.                                     |
| Appartenenza  | Stabilisce in che modo un contatto può associare le sue mail.                | <b>Contatto(1)</b> : Ogni E-mail si riferisce solo ad un contatto.<br><b>E-mail(0..*)</b> : Un contatto può non possedere alcuna mail oppure può possederne al più di una.  |
| Residenza     | Stabilisce in che modo un contatto può associare il suo indirizzo principale | <b>Contatto(1)</b> : Ogni indirizzo principale si riferisce solo ad un contatto.<br><b>Indirizzo principale(1)</b> : Ogni contatto deve avere solo un indirizzo principale.   |
| Residenza (2) | Stabilisce in che modo un contatto può associare il suo indirizzo secondario | <b>Contatto(1)</b> : Ogni indirizzo secondario si riferisce solo ad un contatto.<br><b>Indirizzo secondario(0..*)</b> : Ogni contatto può non avere indirizzi secondari o può averne più di uno.  |

|                  |  |   |
|------------------|--|---|
| Partecipazione   | Stabilisce in che modo un contatto è inserito in un gruppo | <b>Contatto</b> (1..*): Ogni gruppo è formato da minimo 1 a più contatti.<br><b>Gruppo</b> (0..*): Un contatto può non far parte di gruppi, oppure può essere presente in più di uno.   |
| Ha               | Indica in che modo un contatto può avere numeri telefonici | <b>Contatto</b> (1): Ogni numero si riferisce ad un solo contatto.<br><b>Numero Fisso</b> (1..*) - <b>Numero Mobile</b> (1..*): Un contatto deve obbligatoriamente avere un numero mobile e un numero fisso e può averne anche più di uno |
| Reindirizzamento | Indica in che modo sono gestiti i reindirizzamenti         | <b>Numero Fisso</b> (0..1): Un numero mobile può avere un numero fisso a cui reindirizzare la chiamata oppure no.<br><b>Numero Mobile</b> (0..1): Un numero fisso può avere un numero mobile a cui reindirizzare la chiamata oppure no    |

Figura 2.4: Dizionario delle associazioni

### 2.11.3 Dizionario dei vincoli

| NOME VINCOLO              | DESCRIZIONE   |
|---------------------------|---|
| area_privata_pkey         | L'attributo password è chiave primaria della tabella area privata.                              |
| contatto_pkey             | L'attributo ID è chiave primaria per la tabella contatto.                                       |
| email_pkey                | La coppia di attributi (username,dominio) è chiave primaria per la tabella email.               |
| email_id_contatto_fkey    | L'attributo id_contatto è chiave esterna e si riferisce all'attributo ID della tabella contatto |
| gruppo_pkey               | L'attributo nome è chiave primaria per la tabella gruppo.                                       |
| indirizzo_principale_pkey | L'attributo ID è chiave primaria per la tabella indirizzo_principale.                           |

|   |  |
|---|--|
| indirizzo_principale_id<br>_contatto_fkey | L'attributo id_contatto è chiave esterna e si riferisce all'attributo ID della tabella contatto    |
| indirizzo_secondario_pkey                 | L'attributo ID è chiave primaria per la tabella indirizzo_secondario.                              |
| indirizzo_secondario_id<br>_contatto_fkey | L'attributo id_contatto è chiave esterna e si riferisce all'attributo ID della tabella contatto    |
| numero_fisso_pkey                         | La coppia di attributi (prefisso,numero,id_contatto) è chiave primaria per la tabella numero_fisso |
| numero_fisso_id<br>_contatto_fkey         | L'attributo id_contatto è chiave esterna e si riferisce all'attributo ID della tabella contatto    |
| numero_mobile_pkey                        | La coppia di attributi (prefisso,numero) è chiave primaria per la tabella numero_fisso             |
| numero_mobile_id<br>_contatto_fkey        | L'attributo id_contatto è chiave esterna e si riferisce all'attributo ID della tabella contatto    |



|                                 |  |
|---------------------------------|--|
| partecipazione_pkey             | La coppia di attributi (id_contatto,nome_gruppo) è chiave primaria per la tabella partecipazione   |
| partecipazione_id_contatto_fkey | L'attributo id_contatto è chiave esterna e si riferisce all'attributo ID della tabella contatto  |
| partecipazione_nome_gruppo_fkey | L'attributo nome_gruppo è chiave esterna e si riferisce all'attributo nome della tabella contatto  |
| reindirizzamento_pkey           | La coppia di attributi (prefisso_fisso, numero_fisso, id_contatto_fisso, prefisso_mobile, numero_mobile) è chiave primaria per la tabella reindirizzamento |
| sistema_di_messaging_pkey       | La coppia di attributi (servizio,email_username, email_dominio) è chiave primaria per la tabella sistema_di_messaging                                      |
| check_servizi                   | Controllo che il nome del servizio inserito sia un sistema di messaging reale.   |

|                           |   |
|---------------------------|---|
| uniqueusername            | La coppia di attributi (servizio,username) è unica.   |
| cancella_contatto_cascade | Trigger che si attiva quando viene eliminato un contatto.<br>Cancella automaticamente tutte le relazioni con l'ID contatto cancellato   |
| cancella_gruppo           | Trigger che si attiva quando in un gruppo ci sono 0 contatti partecipanti.<br>Cancella automaticamente quel gruppo  |
| check_equal               | Trigger che si attiva quando viene inserito un numero mobile o fisso.<br>Controlla che il numero inserito non sia già presente nel database. In caso affermativo impedisce l'inserimento                          |
| check_indirizzo           | Trigger che si attiva quando viene inserito un indirizzo secondario.<br>Controlla che l'indirizzo appena inserito non sia già presente tra gli indirizzi principali. In caso affermativo impedisce l'inserimento. |
| check_unico               | Trigger che si attiva quando viene inserito un indirizzo principale.<br>Controlla che sia stato inserito un indirizzo principale per quel contatto.<br>In caso affermativo impedisce l'inserimento                |

Figura 2.5: Dizionario dei vincoli

## Capitolo 3

# Progettazione logica

In questo capitolo verrà trattata la fase successiva, la quale prevede un astrazione a livello più basso della precedente.

Verrà dunque tradotto lo schema concettuale visto in precedenza, in schema logico.

In questa fase saranno scritte in grassetto le chiavi primarie e sottolineate le chiavi esterne.

### 3.1 Schema logico

Contatto (ID, FOTO, NOME, COGNOME, SECURITY)

Email ( (USERNAME, DOMINIO), IDCONTATTO\*)

Gruppo (NOME)

Partecipazione ( (IDCONTATTO\*, NOMEGRUPPO\*) )

Numero Fisso ( (PREFISSO, NUMERO), IDCONTATTO\*)

Numero Mobile ( (PREFISSO, NUMERO), IDCONTATTO\*)

Reindirizzamento ( ( (PREFISSO FISSO\*, NUMERO FISSO\*), (PREFISSO MOBILE\*, NUMERO MOBILE\*) ) )

Sistema di Messaging ( (USERNAME, SERVIZIO), STATO, IDCONTATTO\*)

Indirizzo principale (ID, VIA, CIVICO, CITTA', CAP, NAZIONE, IDCONTATTO\*)

Indirizzo secondario (ID, VIA, CIVICO, CITTA', CAP, NAZIONE, IDCONTATTO\*)

Figura 3.1: Schema logico

## Capitolo 4

# Progettazione fisica

In questo capitolo verrà trattata la progettazione fisica, quindi verranno mostrate le query di creazione tabelle e di creazione trigger. Questa è la fase finale della creazione del database, tramite l'ausilio dello schema logico si potranno finalmente implementare le query di creazione.

### 4.1 Definizioni tabelle

#### 4.1.1 Tabella contatto

```
CREATE TABLE CONTATTO (  
    ID SERIAL,  
    FOTO VARCHAR(200) DEFAULT NULL,  
    NOME VARCHAR(20) NOT NULL,  
    COGNOME VARCHAR(20) NOT NULL,  
    SECURITY BOOLEAN DEFAULT FALSE,  
  
    PRIMARY KEY(ID)  
);
```

Figura 4.1: Contatto

#### 4.1.2 Tabella gruppo

```
CREATE TABLE GRUPPO (  
    NOME VARCHAR(50),  
  
    PRIMARY KEY(NOME)  
);
```

Figura 4.2: Gruppo

#### 4.1.3 Tabella partecipazione

```
CREATE TABLE PARTECIPAZIONE (  
    ID_CONTATTO INTEGER NOT NULL,  
    NOME_GRUPPO VARCHAR(50) NOT NULL,  
  
    PRIMARY KEY(ID_CONTATTO, NOME_GRUPPO),  
    FOREIGN KEY(ID_CONTATTO) REFERENCES CONTATTO(ID),  
    FOREIGN KEY(NOME_GRUPPO) REFERENCES GRUPPO(NOME)  
);
```

Figura 4.3: Partecipazione

#### 4.1.4 Tabella sistema di messaging

```
CREATE TABLE SISTEMA_DI_MESSAGING (  
    USERNAME VARCHAR(50),  
    SERVIZIO VARCHAR(10),  
    STATO VARCHAR(100),  
    EMAIL_USERNAME VARCHAR(50) NOT NULL,  
    EMAIL_DOMINIO VARCHAR(30) NOT NULL,  
    ID_CONTATTO INTEGER NOT NULL,  
  
    PRIMARY KEY(USERNAME, SERVIZIO),  
    FOREIGN KEY(ID_CONTATTO) REFERENCES CONTATTO(ID),  
    FOREIGN KEY(EMAIL_USERNAME, EMAIL_DOMINIO) REFERENCES EMAIL(USERNAME, DOMINIO),  
  
    CONSTRAINT check_servizi  
    CHECK (SERVIZIO IN ('Messenger', 'Signal', 'Skype', 'Teams', 'Telegram', 'Viber', 'WeChat', 'Whatsapp'))  
);
```

Figura 4.4: Sistema di messaging

#### 4.1.5 Tabella e-mail

```
CREATE TABLE EMAIL (  
    USERNAME VARCHAR(50),  
    DOMINIO VARCHAR(30),  
    ID_CONTATTO INTEGER NOT NULL,  
  
    PRIMARY KEY(USERNAME, DOMINIO),  
    FOREIGN KEY(ID_CONTATTO) REFERENCES CONTATTO(ID)  
);
```

Figura 4.5: E-mail

#### 4.1.6 Tabella indirizzo principale

```
CREATE TABLE INDIRIZZO_PRINCIPALE (  
    ID SERIAL,  
    VIA VARCHAR(50) NOT NULL,  
    CIVICO INTEGER NOT NULL,  
    CITTA VARCHAR(50) NOT NULL,  
    CAP VARCHAR(12) NOT NULL,  
    NAZIONE VARCHAR(60) NOT NULL,  
    ID_CONTATTO INTEGER NOT NULL,  
  
    PRIMARY KEY(ID),  
    FOREIGN KEY(ID_CONTATTO) REFERENCES CONTATTO(ID)  
);
```

Figura 4.6: Indirizzo principale

#### 4.1.7 Tabella indirizzo secondario

```
CREATE TABLE INDIRIZZO_SECONARIO (  
    ID SERIAL,  
    VIA VARCHAR(50) NOT NULL,  
    CIVICO INTEGER NOT NULL,  
    CITTA VARCHAR(50) NOT NULL,  
    CAP VARCHAR(12) NOT NULL,  
    NAZIONE VARCHAR(60) NOT NULL,  
    ID_CONTATTO INTEGER NOT NULL,  
  
    PRIMARY KEY(ID),  
    FOREIGN KEY(ID_CONTATTO) REFERENCES CONTATTO(ID)  
);
```

Figura 4.7: Indirizzo secondario

#### 4.1.8 Tabella numero fisso

```
CREATE TABLE NUMERO_FISSO (  
    PREFISSO VARCHAR(5),  
    NUMERO VARCHAR(10),  
    ID_CONTATTO INTEGER NOT NULL,  
  
    PRIMARY KEY(PREFISSO, NUMERO, ID_CONTATTO),  
    FOREIGN KEY(ID_CONTATTO) REFERENCES CONTATTO(ID)  
);
```

Figura 4.8: Numero fisso



#### 4.1.9 Tabella numero mobile

```
CREATE TABLE NUMERO_MOBILE (  
    PREFISSO VARCHAR(5),  
    NUMERO VARCHAR(10),  
    ID_CONTATTO INTEGER NOT NULL,  
  
    PRIMARY KEY(PREFISSO, NUMERO),  
    FOREIGN KEY(ID_CONTATTO) REFERENCES CONTATTO(ID)  
);
```

Figura 4.9: Numero mobile

#### 4.1.10 Tabella reindirizzamento

```
CREATE TABLE REINDIRIZZAMENTO (  
    PREFISSO_FISSO VARCHAR(5) NOT NULL,  
    NUMERO_FISSO VARCHAR(10) NOT NULL,  
    ID_CONTATTO_FISSO INTEGER NOT NULL,  
    PREFISSO_MOBILE VARCHAR(5) NOT NULL,  
    NUMERO_MOBILE VARCHAR(10) NOT NULL,  
  
    PRIMARY KEY(PREFISSO_FISSO, NUMERO_FISSO, ID_CONTATTO_FISSO, PREFISSO_MOBILE, NUMERO_MOBILE),  
    FOREIGN KEY(PREFISSO_FISSO, NUMERO_FISSO, ID_CONTATTO_FISSO) REFERENCES NUMERO_FISSO(PREFISSO, NUMERO, ID_CONTATTO),  
    FOREIGN KEY(PREFISSO_MOBILE, NUMERO_MOBILE) REFERENCES NUMERO_MOBILE(PREFISSO, NUMERO)  
);
```

Figura 4.10: Reindirizzamento

## 4.2 Definizioni procedure

### 4.2.1 Procedura check equal (Numeri di telefono)

```
-- Funzione e trigger per individuare e bloccare un numero mobile già presente nei numeri fissi e viceversa

CREATE OR REPLACE FUNCTION check_equal() RETURNS TRIGGER AS $$
BEGIN
    IF tg_name = 'check_equal_fisso' THEN
        IF EXISTS (
            SELECT *
            FROM numero_mobile
            WHERE numero_mobile.prefisso = new.prefisso
            AND numero_mobile.numero = new.numero
        ) THEN
            RAISE EXCEPTION 'Il numero è già presente nella tabella dei numeri mobili';
        END IF;
    ELSIF tg_name = 'check_equal_mobile' THEN
        IF EXISTS (
            SELECT *
            FROM numero_fisso
            WHERE numero_fisso.prefisso = new.prefisso
            AND numero_fisso.numero = new.numero
        ) THEN
            RAISE EXCEPTION 'Il numero è già presente nella tabella dei numeri fissi';
        END IF;
    END IF;
    RETURN NEW;
END
$$ LANGUAGE plpgsql;
```

Figura 4.11: Check equal

### 4.2.2 Check unico (Indirizzo principale)

```
-- Funzione e trigger per far sì che ogni contatto abbia solo un indirizzo principale

CREATE OR REPLACE FUNCTION check_unico() RETURNS TRIGGER AS $$
BEGIN
    IF EXISTS (
        SELECT *
        FROM indirizzo_principale
        WHERE id_contatto = new.id_contatto
    ) THEN
        RAISE EXCEPTION 'È già presente un indirizzo principale per la persona specificata';
    END IF;
    RETURN new;
END;
$$ LANGUAGE plpgsql;
```

Figura 4.12: Check unico

### 4.2.3 Check indirizzo (Indirizzi secondari)

```
-- Funzione e trigger per indirizzo secondario è già presente come indirizzo principale della stessa persona

CREATE OR REPLACE FUNCTION check_indirizzo() RETURNS TRIGGER AS $$
BEGIN
    IF EXISTS (
        SELECT *
        FROM indirizzo_principale
        WHERE id_contatto = new.id_contatto
              AND via = new.via
              AND civico = new.civico
              AND citta = new.citta
              AND cap = new.cap
              AND nazione = new.nazione
    ) THEN
        RAISE EXCEPTION 'Questo indirizzo è già presente come indirizzo principale per la stessa persona';
    END IF;
    RETURN NEW;
END
$$ LANGUAGE plpgsql;
```

Figura 4.13: Check indirizzo

### 4.2.4 Procedura check equal (Numeri di telefono)

```
-- Funzione e trigger per individuare e bloccare un numero mobile già presente nei numeri fissi e viceversa

CREATE OR REPLACE FUNCTION check_equal() RETURNS TRIGGER AS $$
BEGIN
    IF tg_name = 'check_equal_fisso' THEN
        IF EXISTS (
            SELECT *
            FROM numero_mobile
            WHERE numero_mobile.prefisso = new.prefisso
                  AND numero_mobile.numero = new.numero
        ) THEN
            RAISE EXCEPTION 'Il numero è già presente nella tabella dei numeri mobili';
        END IF;
    ELSIF tg_name = 'check_equal_mobile' THEN
        IF EXISTS (
            SELECT *
            FROM numero_fisso
            WHERE numero_fisso.prefisso = new.prefisso
                  AND numero_fisso.numero = new.numero
        ) THEN
            RAISE EXCEPTION 'Il numero è già presente nella tabella dei numeri fissi';
        END IF;
    END IF;
    RETURN NEW;
END
$$ LANGUAGE plpgsql;
```

Figura 4.14: Check equal

### 4.2.5 Cancella gruppo

```
-- Funzione e trigger per la cancellazione automatica di un gruppo quando il numero di componenti è zero

CREATE OR REPLACE FUNCTION cancella_gruppo() RETURNS TRIGGER AS $$
    DECLARE
        partecipanti    INTEGER;
    BEGIN
        SELECT count(*) INTO partecipanti
        FROM partecipazione
        WHERE nome_gruppo = old.nome_gruppo;

        IF partecipanti = 0 THEN
            DELETE FROM gruppo
            WHERE nome = old.nome_gruppo;
        END IF;

        RETURN NULL;
    END
$$ LANGUAGE plpgsql;
```

Figura 4.15: Cancella gruppo

#### 4.2.6 Cancella contatto cascade

```
-- Eliminazione di tutte le informazioni di un contatto

CREATE OR REPLACE FUNCTION cancella_contatto_cascade() RETURNS TRIGGER AS $$
BEGIN
    IF EXISTS (
        SELECT *
        FROM email
        WHERE id_contatto = old.id
    ) THEN
        DELETE FROM email
        WHERE id_contatto = old.id;
    END IF;

    IF EXISTS(
        SELECT *
        FROM indirizzo_principale
        WHERE id_contatto = old.id
    )THEN
        DELETE FROM indirizzo_principale
        WHERE id_contatto = old.id;
    END IF;

    IF EXISTS (
        SELECT *
        FROM indirizzo_secondario
        WHERE id_contatto = old.id
    ) THEN
        DELETE FROM indirizzo_secondario
        WHERE id_contatto = old.id;
    END IF;

    IF EXISTS (
        SELECT *
        FROM reindirizzamento
        WHERE id_contatto_fisso = old.id
    ) THEN
```

Figura 4.16: Cancella contatto cascade

#### 4.2.7 Cancella contatto cascade (2)

```
DELETE FROM reindirizzamento
WHERE id_contatto_fisso = old.id;
END IF;

IF EXISTS (
  SELECT *
  FROM numero_fisso
  WHERE id_contatto = old.id
) THEN
  DELETE FROM numero_fisso
  WHERE id_contatto = old.id;
END IF;

IF EXISTS (
  SELECT *
  FROM numero_mobile
  WHERE id_contatto = old.id
) THEN
  DELETE FROM numero_mobile
  WHERE id_contatto = old.id;
END IF;

IF EXISTS (
  SELECT *
  FROM partecipazione
  WHERE id_contatto = old.id
) THEN
  DELETE FROM partecipazione
  WHERE id_contatto = old.id;
END IF;

IF EXISTS (
  SELECT *
  FROM sistema_di_messaging
  WHERE id_contatto = old.id
) THEN
```

Figura 4.17: Cancella contatto cascade 2

#### 4.2.8 Cancella contatto cascade (3)

```
        DELETE FROM sistema_di_messaging
        WHERE id_contatto = old.id;
    END IF;

    RETURN OLD;
END
$$ LANGUAGE plpgsql;
```

Figura 4.18: Cancella contatto cascade 3

## 4.3 Definizioni trigger

### 4.3.1 Check equal MOBILE

```
CREATE TRIGGER check_equal_mobile  
BEFORE INSERT ON numero_mobile  
FOR EACH ROW  
EXECUTE FUNCTION check_equal();
```

Figura 4.19: Check equal MOBILE

### 4.3.2 Check equal FISSO

```
CREATE TRIGGER check_equal_fisso  
BEFORE INSERT ON numero_fisso  
FOR EACH ROW  
EXECUTE FUNCTION check_equal();
```

Figura 4.20: Check equal FISSO

### 4.3.3 Check indirizzo unico

```
CREATE TRIGGER check_indirizzo_unico  
BEFORE INSERT ON indirizzo_principale  
FOR EACH ROW  
EXECUTE FUNCTION check_unico();
```

Figura 4.21: Check indirizzo unico

### 4.3.4 Check indirizzo principale

```
CREATE TRIGGER check_indirizzo_principale  
BEFORE INSERT ON indirizzo_principale  
FOR EACH ROW  
EXECUTE FUNCTION check_indirizzo();
```

Figura 4.22: Check indirizzo principale



### 4.3.5 Check gruppo

```
CREATE TRIGGER check_gruppo
  AFTER DELETE ON partecipazione
  FOR EACH ROW
  EXECUTE FUNCTION cancella_gruppo()
```

Figura 4.23: Check gruppo

### 4.3.6 Cancella contatto

```
CREATE TRIGGER cancella_contatto
  BEFORE DELETE ON contatto
  FOR EACH ROW
  EXECUTE FUNCTION cancella_contatto_cascade()
```

Figura 4.24: Cancella contatto

## Capitolo 5

# Applicativo Java

### 5.1 HomePage

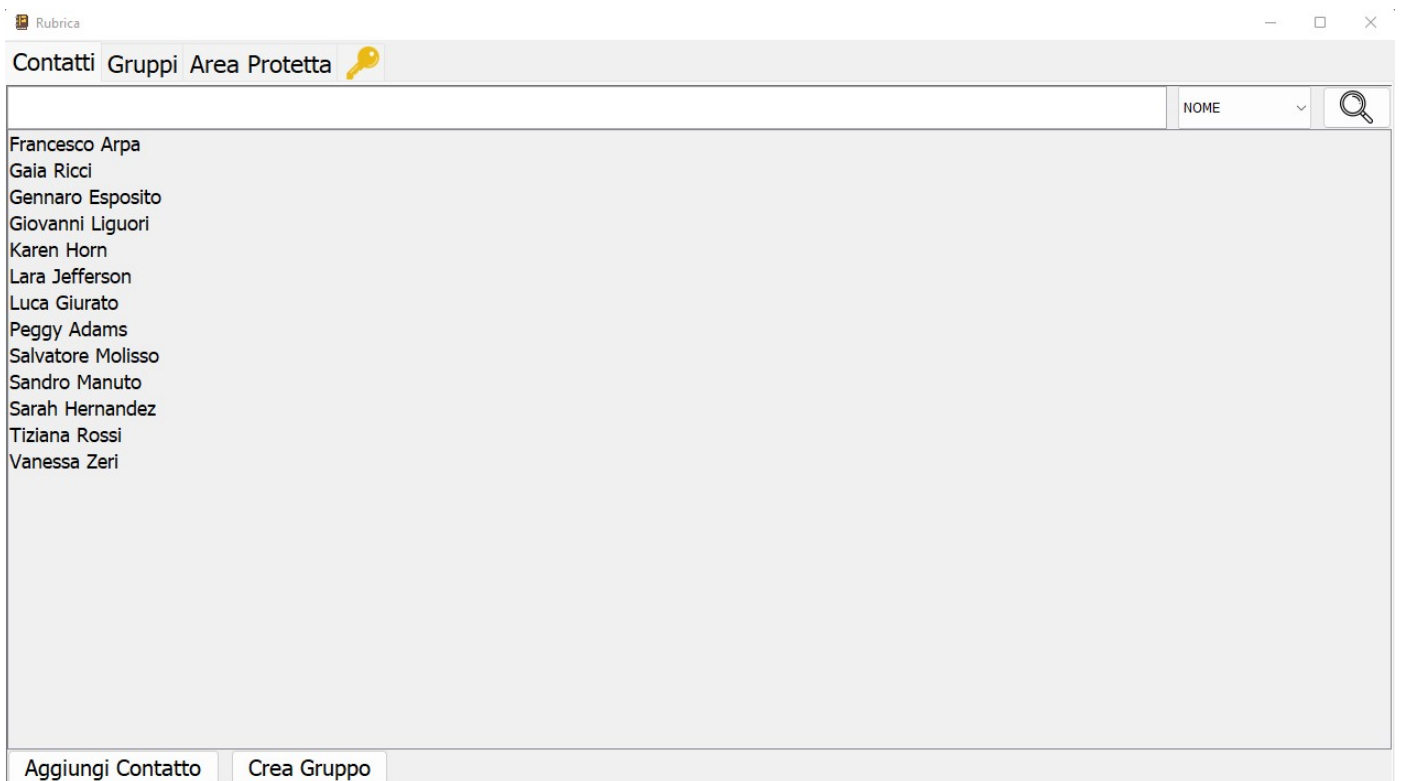


Figura 5.1: Homepage

La homepage è strutturata con uno SplitPane per la lista contatti(***BorderLayout***) e le informazioni del contatto selezionato(***GridLayoutManager***). La lista dei

contatto utilizza una **JList**, i cui dati sono prelevati dal database.  
In alto a sinistra è possibile effettuare una ricerca mirata per ciascun contatto nell'area pubblica.

## 5.2 Visualizzazione contatto

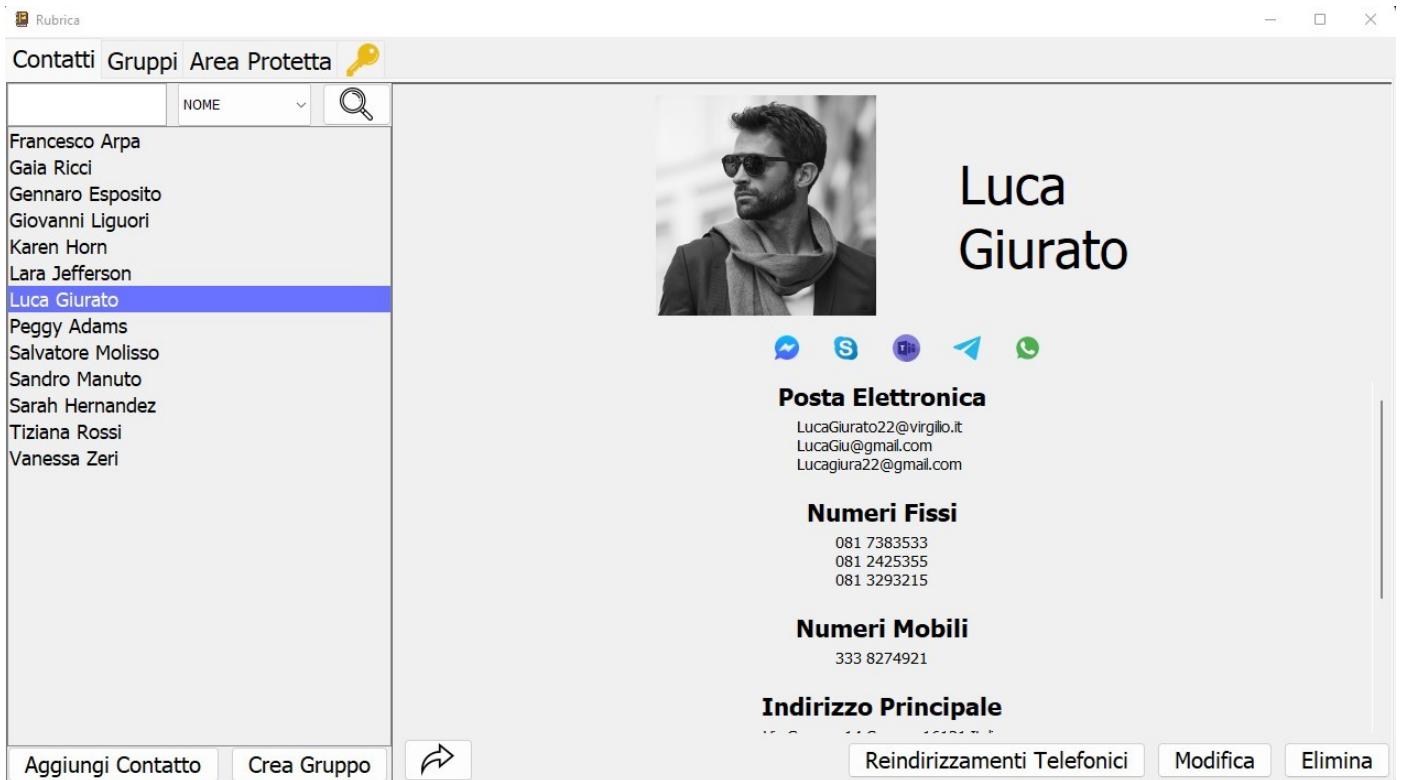


Figura 5.2: Visualizzazione contatto

Tutte le informazioni del contatto saranno prelevate dal database a run-time quando il contatto sarà selezionato.

L'immagine del profilo (nel caso in cui non ne fosse stata selezionata alcuna) sarà generata automaticamente e randomicamente dall'applicativo, selezionando alcune immagini di default presenti nella cartella **.images**.

Compariranno selezionabili soltanto i sistemi di messaging per cui il contatto è effettivamente iscritto.

## 5.3 Modifica contatto

The screenshot shows a web application window titled 'Rubrica'. It has three tabs: 'Contatti', 'Gruppi', and 'Area Protetta' (with a key icon). The 'Contatti' tab is active. On the left, there is a list of contacts: Francesco Arpa, Gaia Ricci, Gennaro Esposito, Giovanni Liguori, Karen Horn, Lara Jefferson, Luca Giurato (highlighted in blue), Peggy Adams, Salvatore Molisso, Sandro Manuto, Sarah Hernandez, Tiziana Rossi, and Vanessa Zeri. Below the list are two buttons: 'Aggiungi Contatto' and 'Crea Gruppo'. The main area on the right is for editing the selected contact, Luca Giurato. It features a profile picture of a man with sunglasses. To the right of the picture are input fields for 'Nome' (Luca) and 'Cognome' (Giurato), and a checkbox for 'Contatto privato'. Below these are social media icons for WhatsApp, Telegram, Facebook, Messenger, and Email. There are three sections for contact information, each with a '+' and '-' button: 'Posta Elettronica' (LucaGiurato22@virgilio.it, LucaGiu@gmail.com, Lucagiura22@gmail.com), 'Numeri Fissi' (0817383533, 0812425355, 0813293215), and 'Numeri Mobili' (3338274921). At the bottom right are three buttons: 'Reindirizzamenti Telefonici', 'Aggiorna', and 'Annulla'.

| Nome | Cognome |
|------|---------|
| Luca | Giurato |

☐ Contatto privato

**Posta Elettronica** + -

- LucaGiurato22@virgilio.it
- LucaGiu@gmail.com
- Lucagiura22@gmail.com

**Numeri Fissi** + -

- 0817383533
- 0812425355
- 0813293215

**Numeri Mobili** + -

- 3338274921

Reindirizzamenti Telefonici   Aggiorna   Annulla

Figura 5.3: Modifica contatto

Quando viene effettuata un operazione di modifica il sistema provvede ad "eliminare" il contatto dal database e generarne un altro con le modifiche apportate. Tutti i *textField* saranno pre-impostati con le informazioni già presenti del contatto.

## 5.4 Crea contatto

The screenshot shows a window titled "Rubrica" (Address Book) with three tabs: "Contatti" (Contacts), "Gruppi" (Groups), and "Area Protetta" (Protected Area). The "Contatti" tab is active. On the left, a list of contacts is displayed, with "Luca Giurato" selected. The main area on the right is for creating a new contact. It features a green placeholder icon for a profile picture. Below the icon are social media sharing icons for WhatsApp, Telegram, Messenger, and others. The form includes fields for "Nome" (Name) and "Cognome" (Surname), a checkbox for "Contatto privato" (Private Contact), and sections for "Posta Elettronica" (Email), "Numeri Fissi" (Fixed Numbers), "Numeri Mobili" (Mobile Numbers), "Indirizzo Principale" (Main Address), and "Indirizzi Secondari" (Secondary Addresses). Each of these sections has a text input field and "+" and "-" buttons. At the bottom, there are buttons for "Aggiungi Contatto" (Add Contact), "Crea Gruppo" (Create Group), "Reindirizzamenti Telefonici" (Phone Redirects), "Aggiungi" (Add), and "Annulla" (Cancel).

Figura 5.4: Crea contatto

Per scelta implementativa si è impostato un limite di textField massimi per ciascun campo di 10.

Non sarà possibile aggiungere il contatto senza il riempimento obbligatorio dei seguenti campi :

- Nome
- Cognome
- Indirizzo Principale

Per l'aggiunta di sistemi di messaging è necessario riempire almeno un campo e-mail.

L'aggiunta di un reindirizzamento e/o di un sistema di messaging aprirà nuove window per creare meno confusione nella visualizzazione dei contenuti.

## 5.5 Crea gruppo

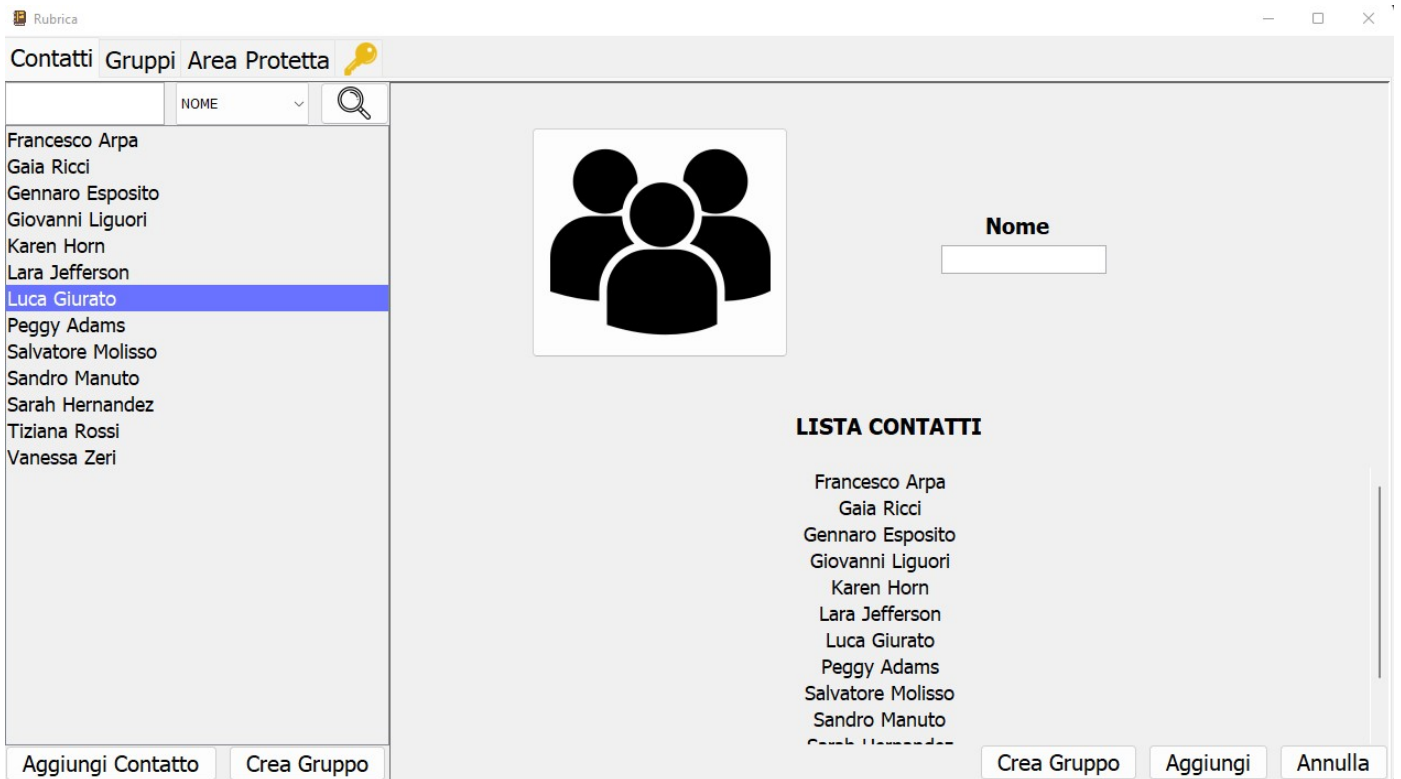


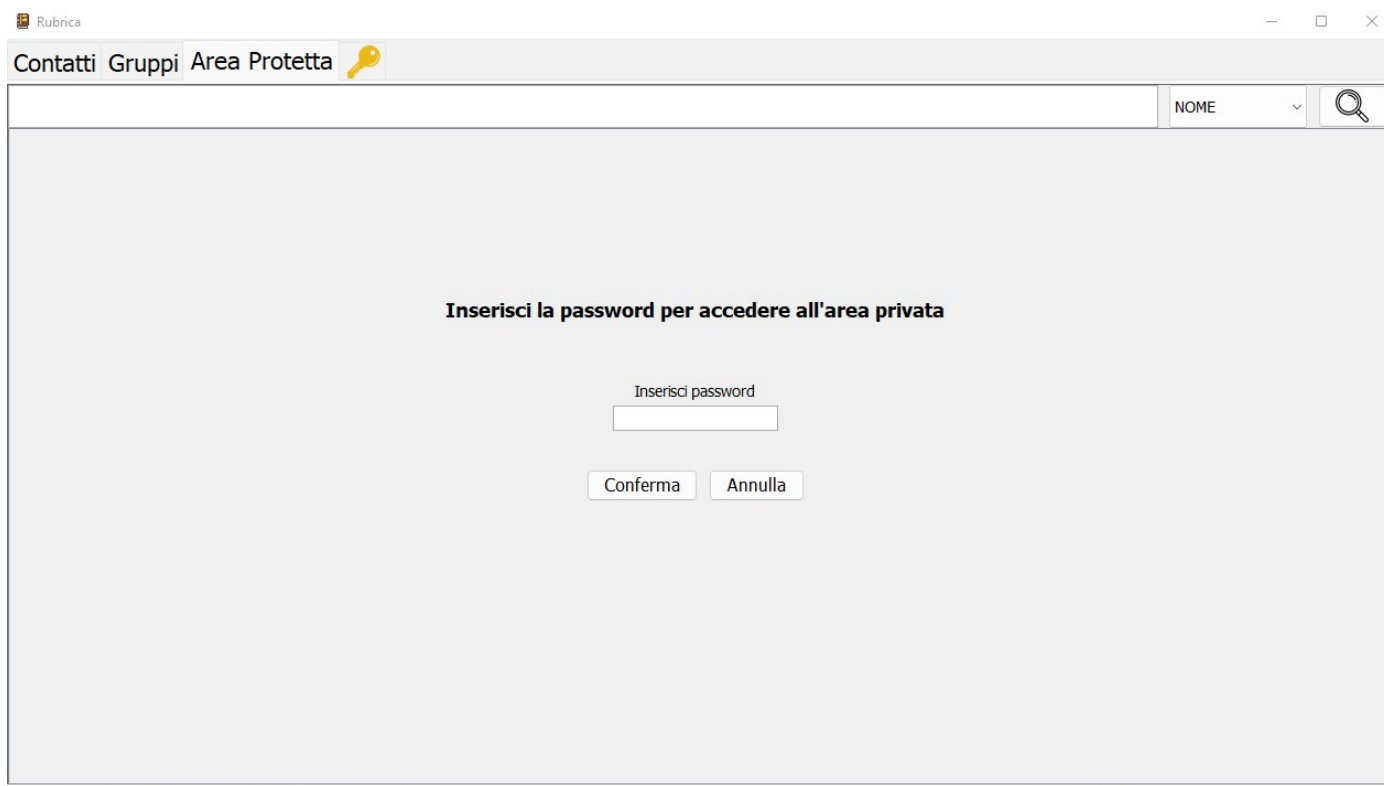
Figura 5.5: Crea Gruppo

Viene generata una lista con tutti i contatti pubblici, ed è possibile aggiungerli al gruppo.

La generazione del gruppo viene effettuata SOLO quando verrà cliccato "Crea Gruppo", con la selezione del contatto e il pulsante "aggiungi" verrà solamente aggiunto ad una lista temporanea il contatto.

Nel caso fosse stato aggiunto qualche contatto e poi selezionato il pulsante "annulla", la lista temporanea verrà svuotata e il gruppo non sarà creato.

## 5.6 Area privata



The screenshot shows a web application window titled 'Rubrica' (Address Book). The navigation bar includes 'Contatti' (Contacts), 'Gruppi' (Groups), and 'Area Protetta' (Protected Area), which is currently selected and accompanied by a yellow key icon. In the top right corner of the main area, there is a search bar with the placeholder text 'NOME' and a magnifying glass icon. The central part of the screen displays a login prompt: 'Inserisci la password per accedere all'area privata' (Enter the password to access the private area). Below this text is a single-line password input field. At the bottom of the form, there are two buttons: 'Conferma' (Confirm) and 'Annulla' (Cancel).

Figura 5.6: Area privata

Per l'accesso all'area privata bisognerà inserire una password (Al primo accesso sarà fatta generare dall'utente e salvata nel database).

E' possibile inoltre, modificare la password cliccando sull'icona della chiave.

I contatti, e tantomeno la preview di essi, non saranno visibili all'utente fino a quando non digiterà la password corretta.

## Capitolo 6

# Riflessioni finali

### 6.1 Le nostre idee

Lavorare in gruppo per la conclusione del progetto è stata una grande crescita personale per ognuno di noi.

Nella creazione del class diagram, per la creazione della GUI dell'applicativo e l'implementazione di qualche algoritmo, ci sono state opinioni contrastanti che hanno sempre trovato un accordo generale.

Il nostro scopo era quello di creare un applicativo che fosse il più "intuitivo" e semplice possibile : L'utente si sarebbe dovuto sentire subito a suo agio esplorando l'applicativo.