

L'applicazione deve avere un'interfaccia grafica conforme a quanto rappresentato in Figura 1, con finestra principale di dimensioni 800px X 600px.

[illegible]

Figura 1

OVERVIEW

L'applicazione gestisce i dati in maniera persistente, in modo da consentirne il recupero ad ogni avvio.

Per avviare l'applicazione è necessario inserire un codice di sblocco OTP (One Time Password) numerico, che viene generato dall'applicazione stessa ogni 10 secondi durante la fase di login.

DETTAGLI SUL COMPORTAMENTO

All'avvio l'applicazione presenta una schermata di inserimento password (Figura 1). Il bottone "Sblocca" si abilita contestualmente all'inserimento di un codice.

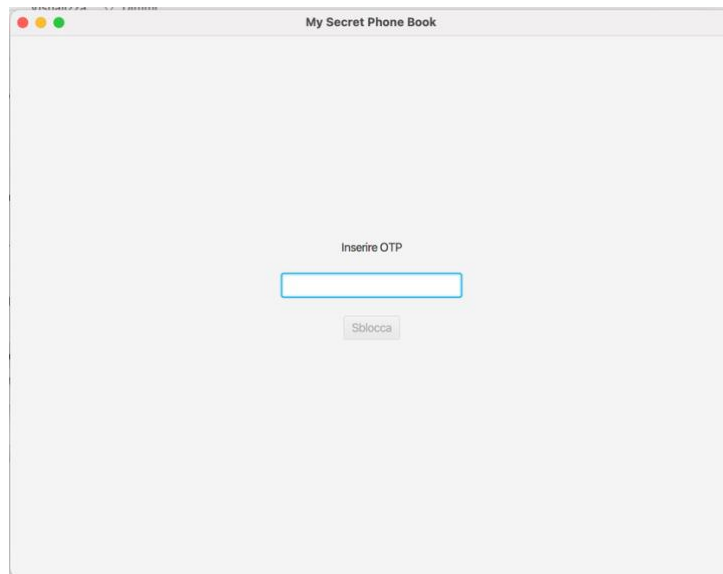


Figura 2

L'applicazione genera un nuovo OTP numerico ogni 10 sec e lo notifica all'utente scrivendolo su un file di testo `otp.txt`

Il file `otp.txt` contiene un'unica riga nel seguente formato:

Il codice di sblocco OTP è: 46

Il valore numerico corrispondente al codice OTP è generato in maniera random¹ come un numero intero tra 0 a 500.

All'utente è richiesto di inserire il codice OTP correntemente valido per sbloccare l'applicazione. In caso di codice errato, appare un messaggio di errore (Figura 3) e l'applicazione resta in attesa di un nuovo inserimento.

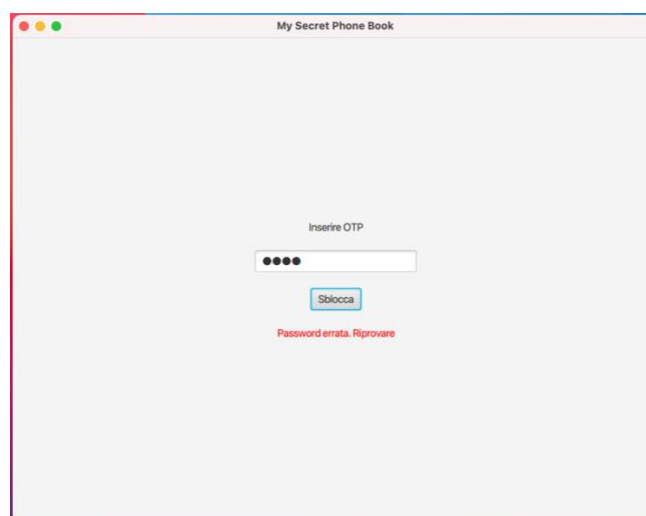


Figura 3

¹ <https://docs.oracle.com/javase/8/docs/api/java/util/Random.html>

Dopo lo sblocco l'applicazione presenta la schermata² riportata in Figura 4. Il pannello laterale sinistro consente l'inserimento di un nuovo contatto mediante un form che contiene i campi da compilare. Si noti che il pulsante "Aggiungi contatto" risulta disabilitato finché il form non è completamente compilato. Dopo l'aggiunta di un nuovo contatto, l'applicazione ripulisce automaticamente i campi di input.

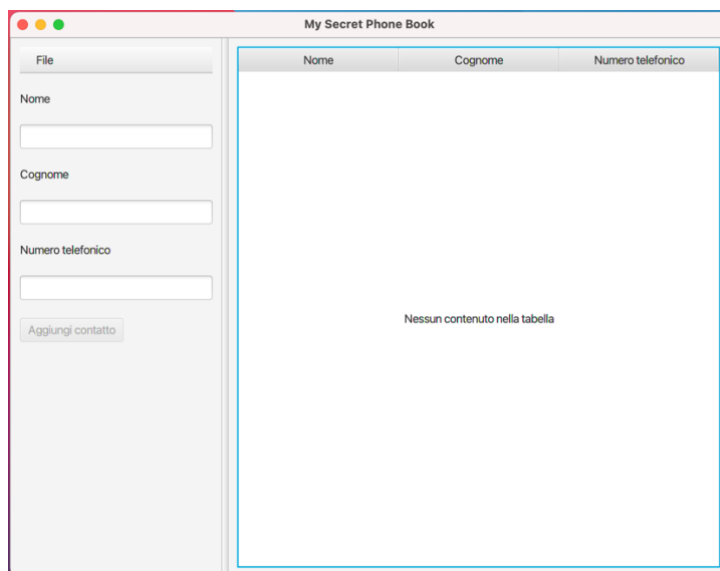


Figura 4

L'applicazione **non** consente l'inserimento di contatti che hanno contemporaneamente lo stesso nome, cognome e numero telefonico e apre una **finestra di errore**³ in caso di tentato inserimento (Figura 5).

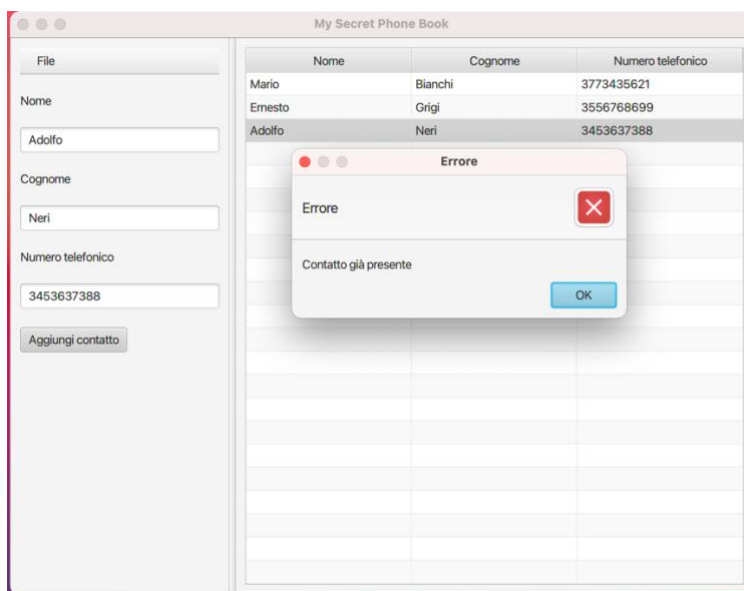


Figura 5

E' possibile cancellare un contatto inserito mediante il **menu contestuale** associato alla tabella⁴. Al menu contestuale si accede mediante click sul tasto destro del mouse (Figura 6). Oltre alla cancellazione dell'elemento selezionato, il menu

² Suggerimento: per la gestione di schermate alternative si valuti l'impiego di [StackPane](https://docs.oracle.com/javase/8/javafx/api/javafx/scene/control/Alert.html)

³ <https://docs.oracle.com/javase/8/javafx/api/javafx/scene/control/Alert.html>

⁴ Componente JavaFX [ContextMenu](#)

consente di copiare sulla Clipboard⁵ di sistema il contatto selezionato per consentire all'utente di esportare i campi in formato testo su altre finestre/applicazioni mediante la funzionalità *Incolla*. In particolare, il contatto verrà copiato come una riga testuale in formato CSV, con i tre campi separati dal carattere “;”.

Esempio: la funzionalità *Incolla* produrrà il seguente output su un generico editor di testo se si copia il contatto selezionato in Figura 6:

Cesare;Neri;3453637388;

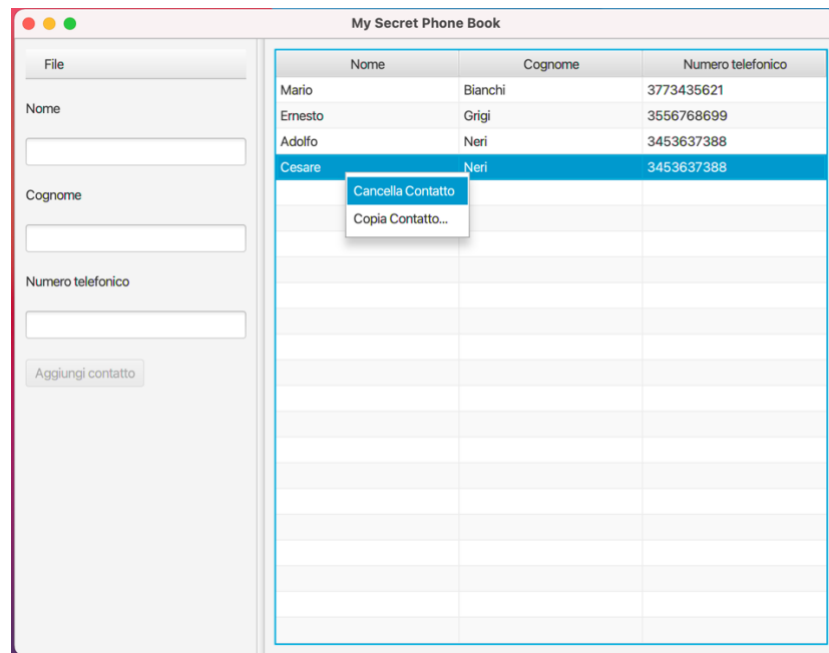


Figura 6

E' possibile modificare i campi per un contatto inserito direttamente dalla tabella, cliccando due volte sul campo da modificare (Figura 7). Si noti che il controllo duplicati deve essere garantito anche in fase di editing.

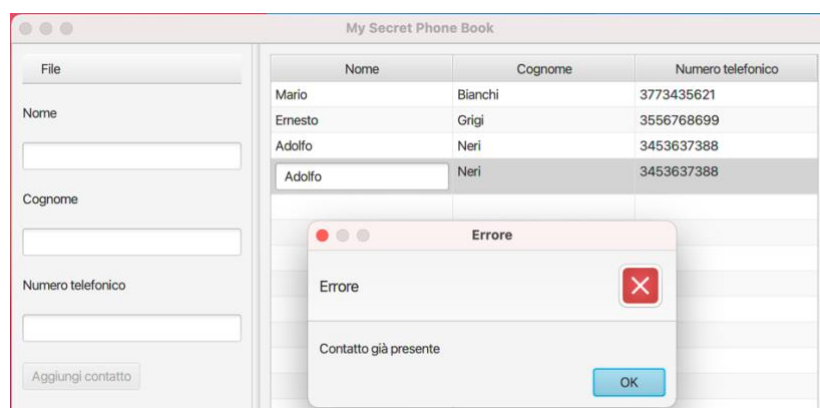


Figura 7

⁵ [Clipboard JavaFX](#)



Figura 8

L'applicazione consente di salvare l'elenco di contatti inseriti, in modo da riproporli ad un successivo avvio, cliccando sul bottone *Salva* nel menu *File*. Il salvataggio avviene su un file binario **saved.bin** in maniera serializzata. Al primo avvio il file di stato non esiste ed andrà automaticamente creato.

Si noti che il bottone *Salva* risulta disabilitato finché non si apportano cambiamenti all'elenco di contatti precedentemente salvato (Figura 8).

Il menu *File* presenta inoltre il bottone *Esci* che consente di terminare l'applicazione (senza salvare automaticamente i cambiamenti).

NOTE SULL'IMPLEMENTAZIONE

- Un contatto è rappresentato dalla classe *Contact* che deve presentare almeno gli attributi *nome*, *cognome*, *numero*. Scegliere per essi un tipo di dato che si ritiene opportuno. Due contatti si assumono uguali se hanno lo stesso valore per gli attributi corrispondenti. Prevedere i getter e i setter per tutti gli attributi.
- La funzionalità di generazione automatica dell'OTP ogni 10 secondi andrà realizzata a partire dall'implementazione di una classe (da denominare *GenerateOTP*) che implementa l'interfaccia *Runnable*. Si noti che il meccanismo di generazione OTP andrà arrestato a seguito dello sblocco dell'applicazione.
- L'applicazione va progettata seguendo il pattern MVC. La logica di funzionamento dell'applicazione è gestita da una classe *Controller* che implementa l'interfaccia *Initializable*. Il metodo astratto *initialize* dovrà essere implementato allo scopo di inizializzare tutti gli elementi utili alla creazione dell'interfaccia e per gestire eventuali operazioni preliminari necessarie per il corretto funzionamento. Il controller conterrà inoltre i metodi di gestione degli eventi associati ai componenti grafici per realizzare i comportamenti richiesti. Dovranno inoltre essere prodotti: il file FXML che contiene le specifiche della *View* e la classe principale dell'applicazione JavaFX da denominare *MySecretPhoneBook*.

PRIMA DI CREARE IL PROGETTO LEGGERE LE ISTRUZIONI PER LA CONSEGNA!!!

ISTRUZIONI PER LA CONSEGNA

Per almeno una delle classi prodotte, deve essere presente un commento in formato Javadoc della classe e di ciascuno dei suoi metodi pubblici (si veda il seguente link per una descrizione del formato Javadoc).

<https://www.oracle.com/technetwork/articles/java/index-137868.html>

Il progetto dovrà essere impostato in modo che tutte le classi prodotte siano in un package denominato "gruppoXX"

Per la consegna, la cartella di base contenente i file sorgenti (.java e .fxml) di tutto il progetto (quindi, la cartella "gruppoxx"), deve essere compressa in formato .ZIP (attenzione, non in formato .RAR o altri formati di compressione), e deve essere rinominata "gruppoXX.zip", dove XX è il numero del gruppo (esempio: gruppo09.zip).

Solo uno dei membri di ciascun gruppo deve effettuare la consegna sulla piattaforma.