

Voci Volontari Cittadini

Documentazione Progetto
Programmazione Dispositivi Mobili
a.a. 2024/2025

Università degli Studi di Milano – Bicocca

Gruppo "Pepe Imperium"

Pesce Gabriel – 899638 (referente)

Samuele Poretti - 904280

Matthias Muia - 894374

Nezha Oltion - 902221

Enrico Zhou - 899736

Indice

1. Introduzione

2. Tecnologie

2.1. IDE

2.2. Linguaggi e Librerie

2.3. Firebase

2.4. API

3. Architettura

4. Funzionalità

4.1. Registrazione Utente

4.2. Login

4.3. Home Page

4.4. Profilo Utente

4.5. Modifica Profilo Utente

4.6. Richieste

4.7. Aggiungi Senzatetto

4.8. Aggiungi Richieste

5. Design

5.1. Palette Colori

6. Sviluppi Futuri

7. Ringraziamenti

Introduzione

La decisione di sviluppare l'app **VoCi** è nata dal desiderio di fornire ai volontari uno strumento utile per catalogare e gestire le richieste e gli aggiornamenti relativi alle persone senzatetto di Milano.

L'idea è stata ispirata da un membro del team che collabora con *VoCi ONLUS*, un'organizzazione impegnata nell'aiutare i senzatetto a livello locale.

VoCi si propone di supportare i volontari registrati, consentendo loro di inserire, aggiornare e ottenere informazioni sui senzatetto presenti in una determinata area, facilitando così l'assistenza e la gestione delle loro necessità.

Lo svolgimento delle attività di progetto è stato suddiviso in task e assegnate ai membri del progetto:

1. Sketch delle schermate dell'applicazione
2. Implementazione del Database
3. Sistema di autenticazione e modifica del volontario
4. Creazione dei senzatetto
5. Creazione delle richieste
6. Implementazione dell'API per l'utilizzo della mappa
7. Ridefinizione di casi secondari in base a comportamenti imprevisti dell'utente
8. Test finale dell'applicazione
9. Scrittura della documentazione

Durante le varie fasi si sono effettuati incontri regolari tra i membri del progetto per verificare il punto della situazione e l'assegnazione di nuove task.

Tecnologie

IDE

Abbiamo utilizzato **Android Studio** per scrivere e testare il codice

Linguaggi e Librerie

La nostra applicazione è scritta principalmente in **Kotlin**, un linguaggio moderno, conciso e type-safe, progettato per lo sviluppo Android. Abbiamo utilizzato diverse librerie per migliorare le funzionalità:

- **Jetpack**: Una suite di librerie di Google che semplifica attività comuni nello sviluppo Android, come la gestione della navigazione, dei cicli di vita, dei componenti UI e altro.
- **Material Design 3**: L'ultima iterazione del sistema di design Material di Google per Android, con un linguaggio di design coerente ed elegante e componenti predefiniti come pulsanti, campi di testo e card.
- **Gradle**: Uno strumento di automazione della build che gestisce le dipendenze, compila l'app e la prepara per la distribuzione.

Firebase

La piattaforma di sviluppo mobile di Google. Utilizziamo Firebase per:

- **Archiviazione dei dati utente e dell'app**: Firebase Firestore, un database cloud NoSQL, fornisce un'archiviazione dati flessibile e scalabile.
- **Autenticazione Firebase (Auth)**: Semplifica l'autenticazione e la gestione degli utenti, permettendo agli utenti di accedere tramite diversi metodi, come email/password.
- **Git & GitHub**: Strumenti essenziali per il controllo di versione e la condivisione del codice. Avremmo potuto utilizzare GitLab, ma siamo più familiari con GitHub grazie a corsi precedenti.

API

Abbiamo utilizzato Mapbox Forward e Reverse Geocoding e Suggestions per l'aggiunta della posizione e SDK Mapbox per le Mappe

- **Forward Geocoding**: Consente di convertire un indirizzo o un luogo in coordinate geografiche (latitudine e longitudine).
- **Reverse Geocoding**: Fa il contrario del forward geocoding: converte una coppia di coordinate geografiche in un indirizzo leggibile.
- **Mapbox Maps SDK**: Per aggiungere mappe dinamiche personalizzabili. Offre funzionalità come rendering offline, marker personalizzati, overlay grafici e supporto per stili di mappe personalizzati.

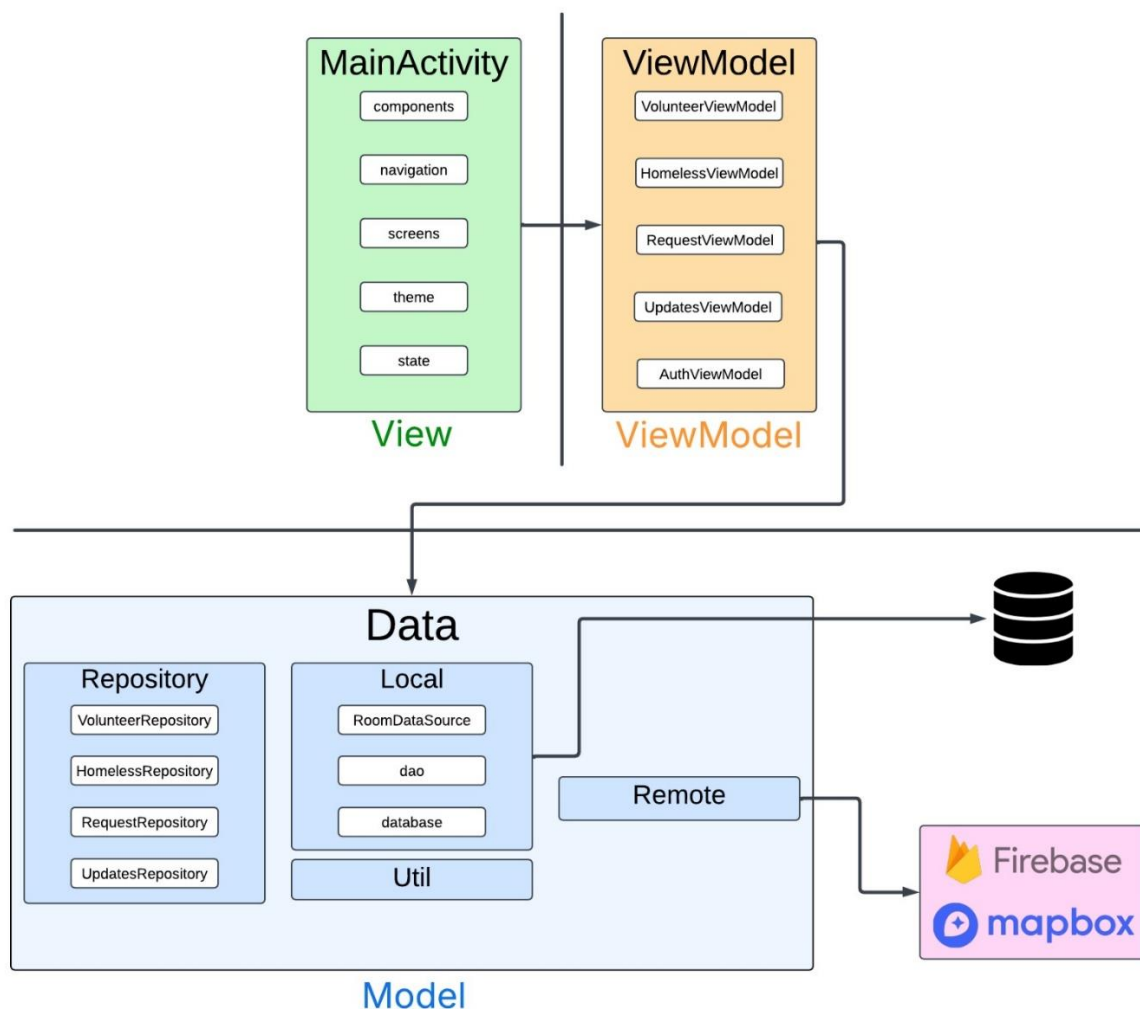
Documentazione

<https://sporetti.github.io/Voci-App/>

Architettura

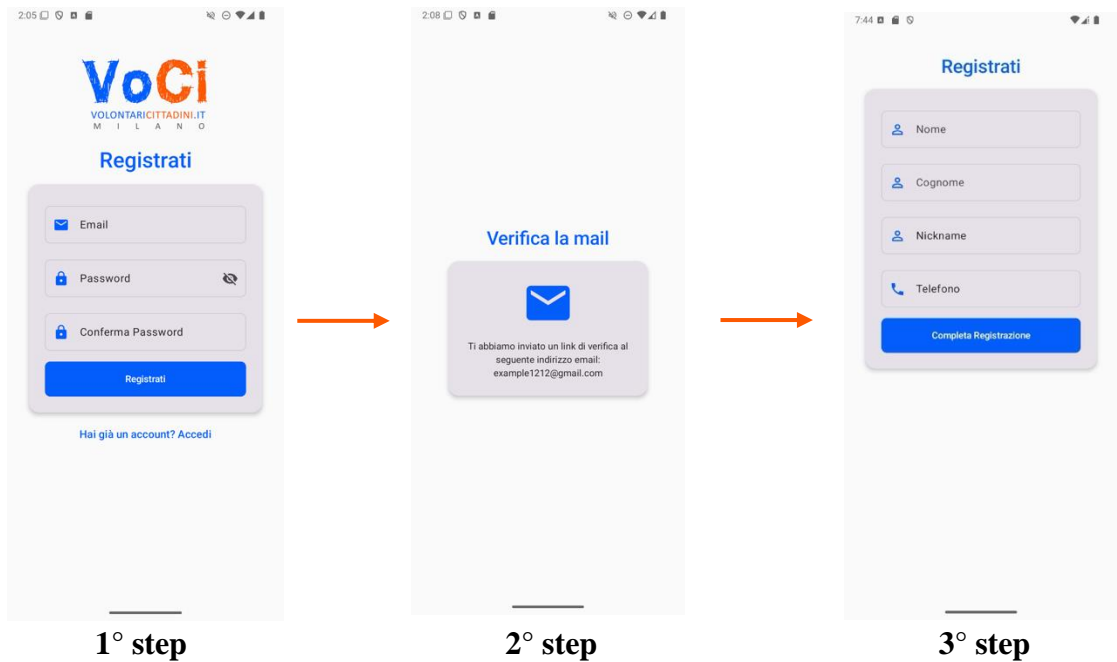
L'applicazione adotta un'architettura **Model-View-ViewModel (MVVM)**, suddividendo le responsabilità tra quattro principali package:

1. **Interfaccia utente (View):** Mostra i dati forniti dal ViewModel e consente all'utente di interagire attraverso operazioni di input e output.
2. **ViewModel:** Media tra la View e il Repository, fornendo i dati necessari alla View e gestendo la logica dell'interfaccia utente. Mantiene le variabili osservabili per aggiornare dinamicamente l'interfaccia utente.
3. **Repository:** Si occupa di comunicare con il Data Source per ottenere i dati richiesti, trasformandoli o filtrandoli prima di inviarli al ViewModel.
4. **Data Source:** Gestisce l'accesso ai dati provenienti da servizi esterni, come API, Firebase o database locali.



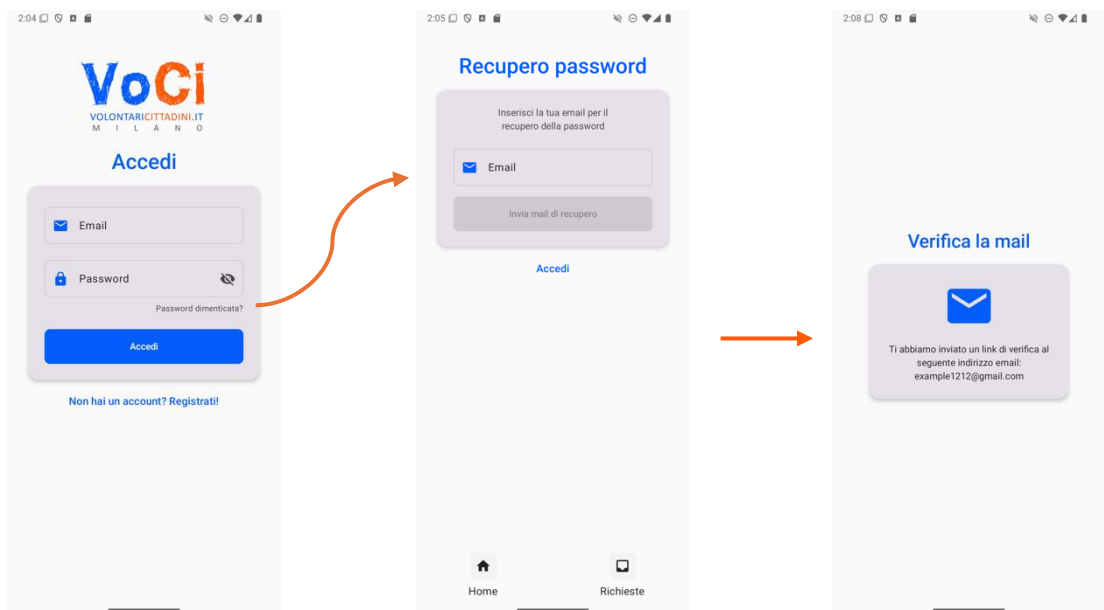
Funzionalità

1. Registrazione Utente



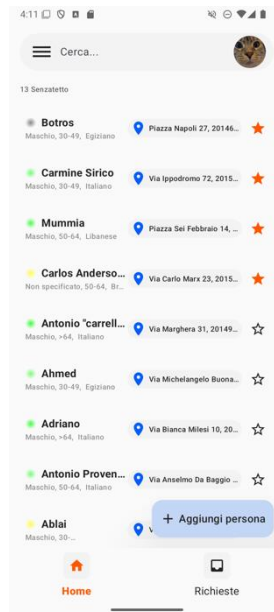
La prima cosa che deve fare l'utente per registrarsi è inserire una email e una password valida (1° step). Successivamente riceverà un'email di verifica alla mail (2° step). Una volta verificata, potrà inserire un nickname, nome, cognome e il proprio numero di telefono (3° step). A registrazione completata, verrà reindirizzato alla Home Page.

2. Login



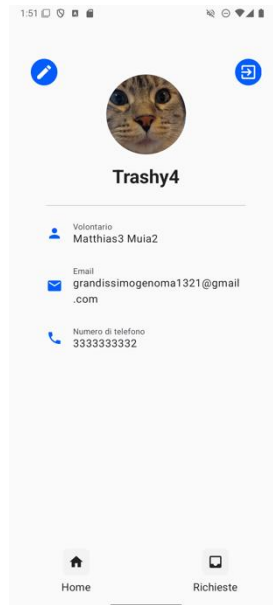
Per accedere all'applicazione l'utente deve inserire email e password. Se dimentica la password, può recuperarla cliccando su "**password dimenticata**", inserendo la propria email, riceverà un link per creare la nuova password.

3. Home Page



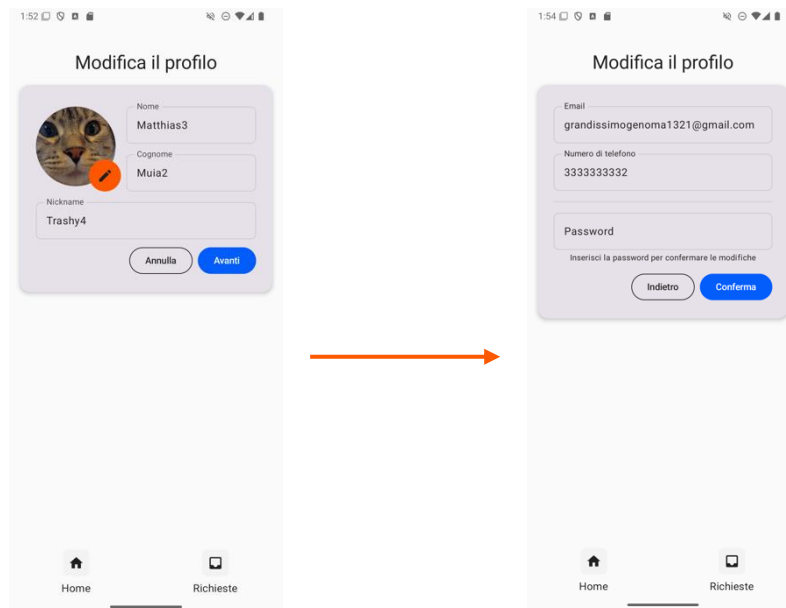
In questa schermata, l'utente può visualizzare la lista dei senzatetto registrati e cercare un senzatetto specifico usando la barra di ricerca, è possibile anche navigare tra le pagine di Home Page e Richieste tramite una barra di navigazione inferiore. (Al prossimo riavvio dell'applicazione se l'utente è già registrato salterà direttamente in questa schermata)

4. Profilo Utente



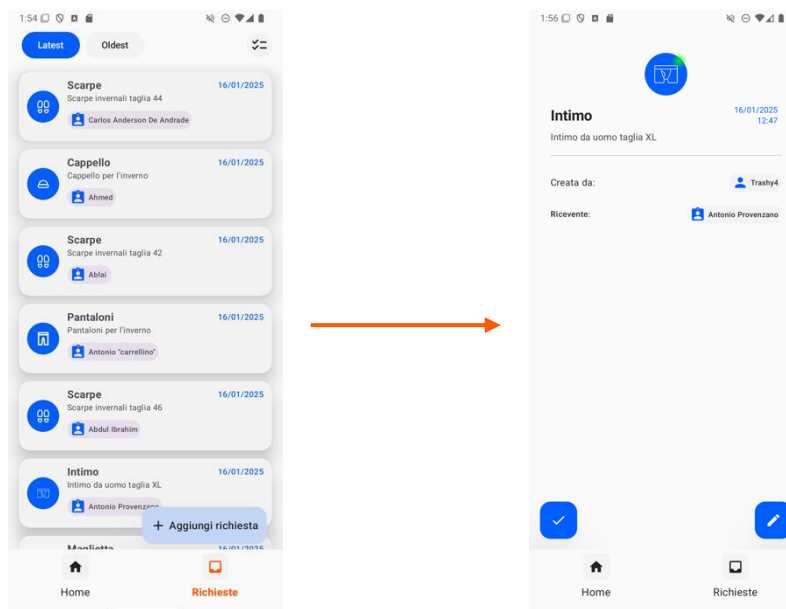
Presenta le informazioni del volontario, offre la possibilità di modificare i dati e la funzionalità di logout.

5. Modifica Profilo Utente



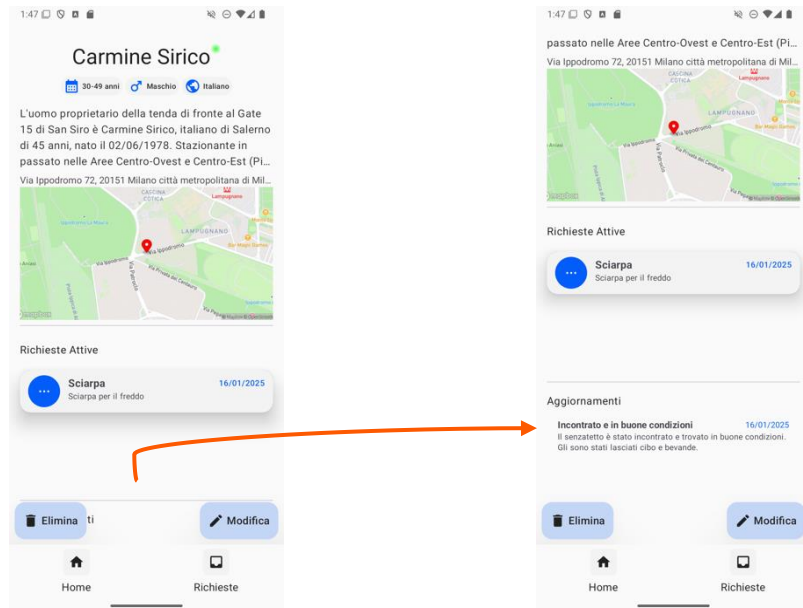
Permette la modifica dei dati dell'utente (compresa l'immagine del profilo tramite URL). Per confermare le modifiche dei dati, l'utente dovrà inserire la password. Per la modifica dell'email, verrà inviata un'email di conferma. Il logout verrà effettuato appena confermata la nuova email.

6. Richieste



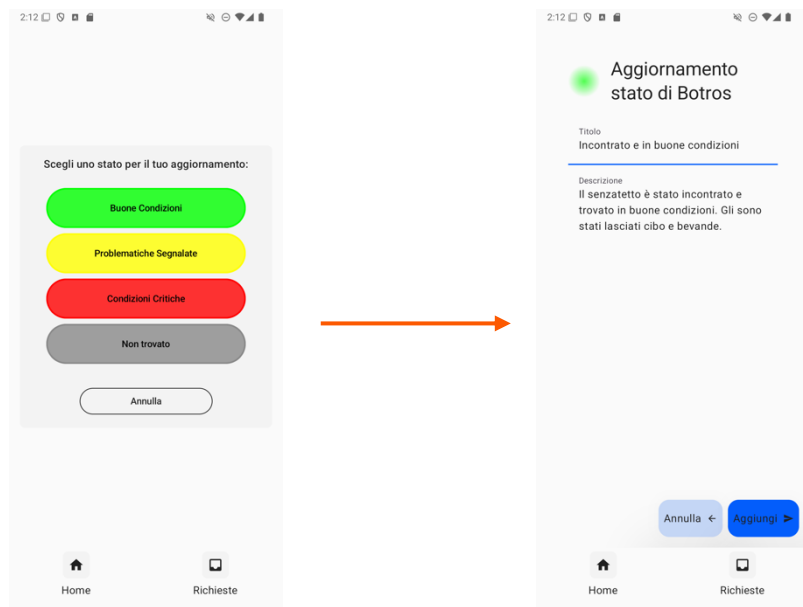
Viene visualizzata la lista delle richieste per ogni senzاتetto, divise in richieste **Recenti** e **Passate**, all'interno di ogni richiesta troviamo i dettagli di chi l'ha creata, a chi è destinata e i dettagli della richiesta.

7. Profilo Senzatetto



Il profilo di un senzatetto include informazioni personali come nome, età, genere, nazionalità e provenienza. Viene visualizzata la posizione sulla mappa per facilitare l'individuazione. Sono elencate le **richieste attive** del senzatetto. È presente una sezione dedicata agli **aggiornamenti** sullo stato del senzatetto, che consente di monitorarne la situazione attuale e passate.

8. Stato Senzatetto



Questa schermata permette di aggiornare lo stato di un senzatetto, offrendo la possibilità di scegliere una condizione adeguata in base alla situazione osservata. Una volta scelta lo stato del senzatetto viene portato in una schermata dove si possono scrivere ulteriori informazioni sullo stato del senzatetto.

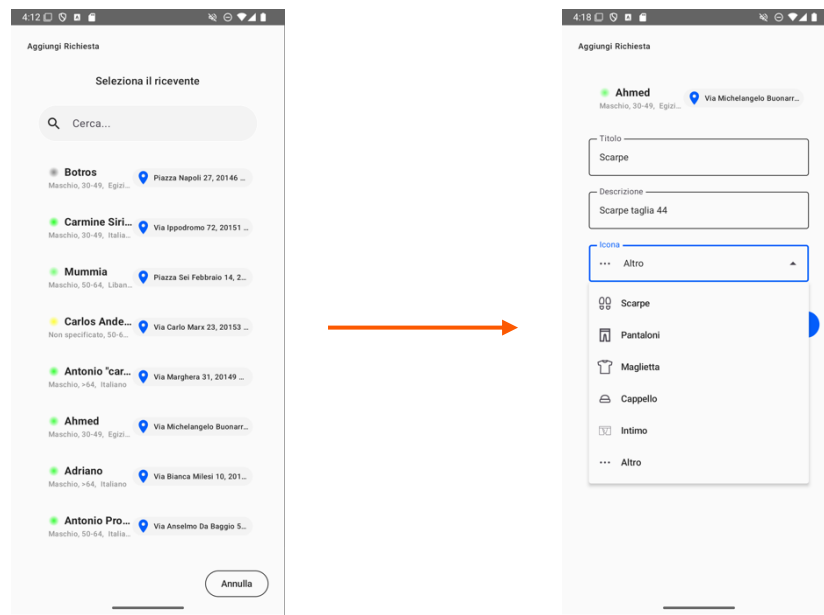
1. Aggiungi Senzatetto

The process consists of the following steps:

- Step 1:** A form titled "Aggiungi persona" with a "Nome" (Name) input field and "Annulla" (Cancel) and "Avanti" (Next) buttons.
- Step 2:** A map view of Milan with a search bar "cerca..." and a "Current Location" pin. A "Confirm" button is at the bottom.
- Step 3:** A zoomed-in map view showing a specific location in Milan, with a "Confirm" button at the bottom.
- Step 4:** A form titled "Aggiungi persona" with "Animale" (Animal) set to "No", "Sesso" (Gender) set to "Non specificato", and "Indietro" (Back) and "Avanti" (Next) buttons.
- Step 5:** A form titled "Aggiungi persona" with "Eta" (Age) set to "40-45, 53, 60-70, ...", "Nazionalità" (Nationality) input field, and "Indietro" (Back) and "Avanti" (Next) buttons.
- Step 6:** A form titled "Aggiungi persona" with a "Descrizione" (Description) input field containing a detailed text about the person's background and current situation, and "Indietro" (Back) and "Aggiungi" (Add) buttons.

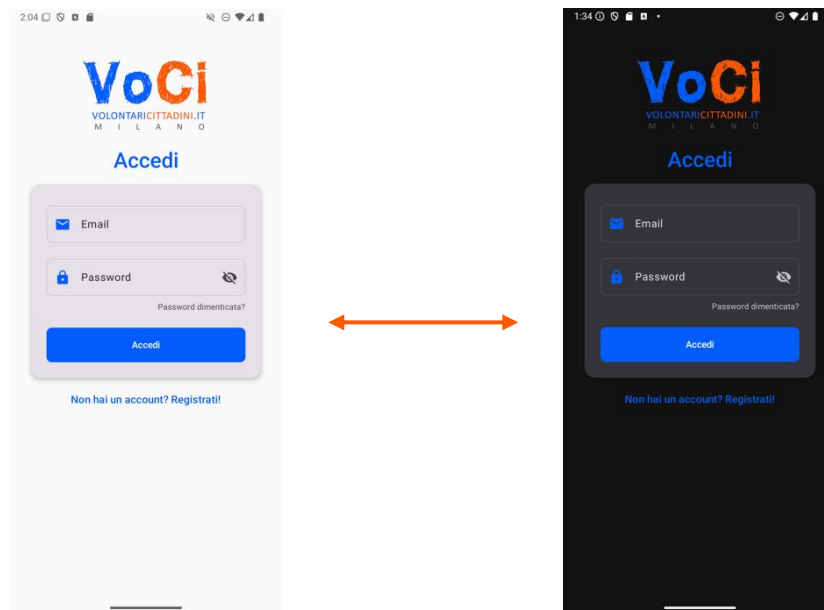
Permette l'inserimento di un nuovo senzatetto nel sistema fornendo le informazioni necessarie (nome, indirizzo, nazionalità...) e un campo **"Descrizione"** per fornire ulteriori dettagli.

2. Aggiungi Richieste



Consente l'aggiunta una nuova richiesta per un senzatetto specificando la tipologia di richiesta (Abbigliamento ecc...)

3. Tema



Light mode

Dark mode

Abbiamo integrato anche i temi **light mode** e **dark mode** per adattarsi alle preferenze degli utenti.

La **light mode** utilizza colori chiari, ideali per ambienti luminosi e lettura diurna.

La **dark mode** sfrutta tonalità scure, perfette per ridurre l'affaticamento visivo di notte.

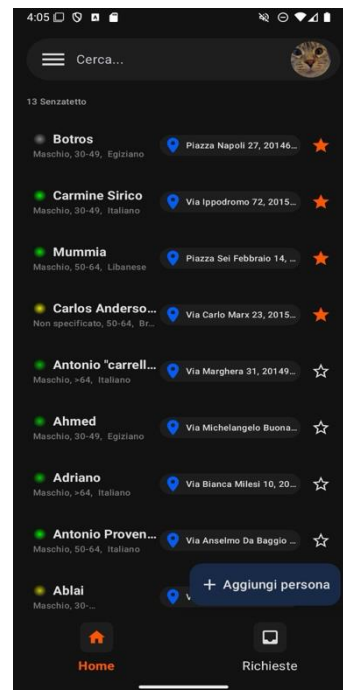
Design

Per la progettazione delle varie interfacce abbiamo utilizzato il programma **Figma** con l'ausilio della libreria **Material Design 3**, utile per utilizzare componenti Android direttamente all'interno delle bozze.

Le seguenti schermate mostrano l'evoluzione della Home Page, dalla semplice bozza all'implementazione vera e propria.



Prima



Dopo

Palette Colori

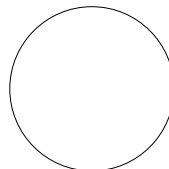
Abbiamo voluto utilizzare una palette minima ma essenziale. Per dare importanza e priorità al logo dell'applicazione, abbiamo utilizzato i seguenti colori:



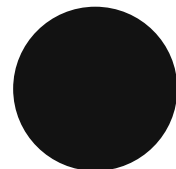
#FF5700



#035DFB



#FAFAFA



#121212

Sviluppi futuri

Personalizzazione

Consentire agli utenti di personalizzare l'interfaccia e le notifiche secondo le proprie preferenze.

Statistiche e report

Generazione automatica di statistiche utili per monitorare il numero di persone assistite, le risorse utilizzate e le aree più critiche.

Pubblicazione su Google play

Condivisione dell'applicazione con altre organizzazioni e volontari che potrebbero trovarla utile per migliorare l'impatto delle loro attività sul territorio.

Supporto multilingue

Si prevede di aggiungere il supporto multilingue per includere volontari e utenti di diverse nazionalità.

Scalabilità

Estendere l'applicazione ad altre città o regioni con configurazioni personalizzate.

Sistema di notifiche

Avvisi per eventi straordinari (es. emergenze climatiche) o richieste urgenti.

Rete di donazioni

Permettere agli utenti di donare direttamente beni o denaro attraverso l'app.

Ringraziamenti

Come referente del gruppo, ci tengo a ringraziare tutti i membri del progetto per il loro impegno costante fin dal primo giorno, attraverso chiamate online e discussioni via messaggi. Il lavoro è stato distribuito in modo equo e uniforme, senza lasciare indietro nessuno. Il progetto è stato portato avanti con grande positività e costanza, mantenendo sempre un atteggiamento positivo. 😊