

Documentazione di progetto
Programmazione di Dispositivi Mobili

Lost&Found

900008 Youssef Benbouzid

912941 Matias Maciej Bonoli

904767 Anas Dahiri

Università degli Studi di Milano Bicocca

A.A. 2024-2025

Indice

- 1. Introduzione2
- 2. Architettura3
- 3. Tecnologie utilizzate.....5
 - 3.1. Backend e persistenza dei dati.....5
 - 3.2. Mappe e servizi di geolocalizzazione.....6
- 4. Funzionalità7
 - 4.1. Accesso e registrazione.....7
 - 4.2. Schermata principale: Lost e Found.....8
 - 4.3. Schermata Oggetto.....9
 - 4.4. Schermata di inserimento10
- 5. Conclusioni e sviluppi futuri11

Introduzione

Lost&Found è un'applicazione mobile per dispositivi Android progettata per semplificare la segnalazione di oggetti smarriti e ritrovati, particolarmente utile in aree geografiche ristrette come i campus universitari.

Grazie a un'interfaccia intuitiva, gli utenti possono registrare rapidamente e in tempo reale gli oggetti smarriti o ritrovati, facilitando la comunicazione diretta tra i due interessati.

Ogni segnalazione include il nome, una descrizione dettagliata, un'immagine e la località esatta in cui è stato perso o ritrovato l'oggetto; gli utenti possono poi selezionare categorie specifiche da una lista predefinita, rendendo più facile l'identificazione degli oggetti, e fornire un recapito telefonico per essere contattati.

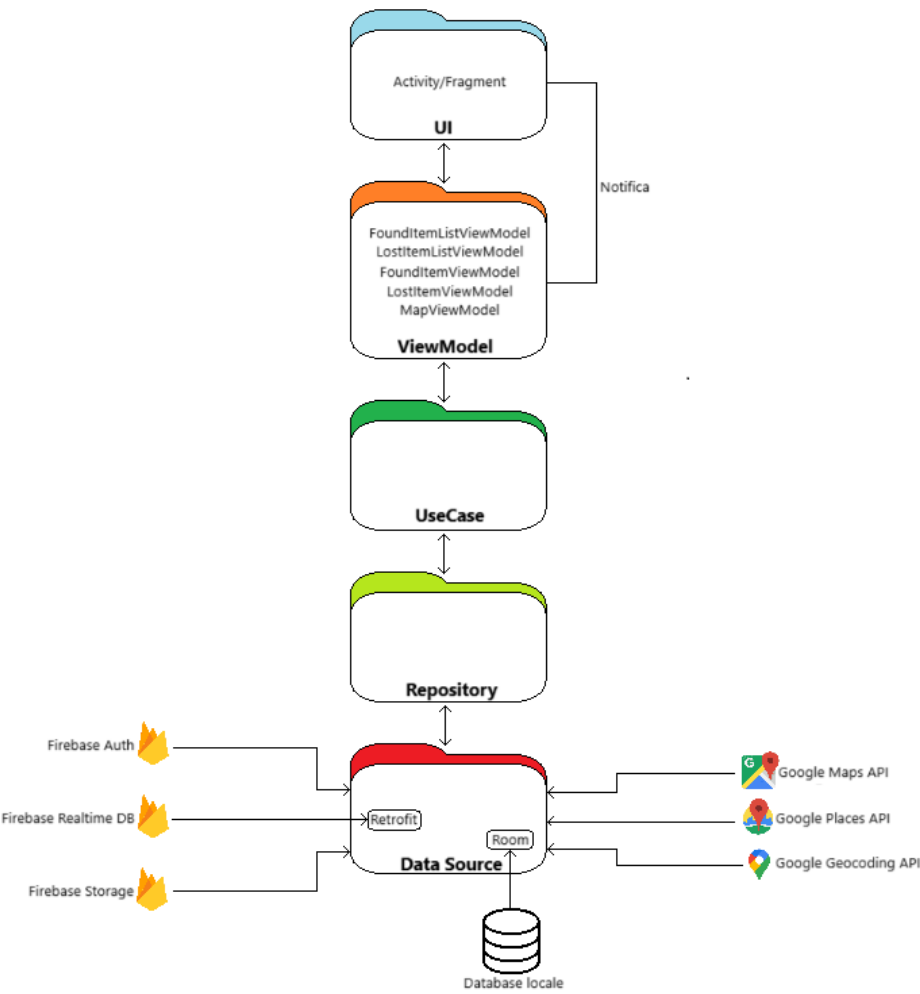
Architettura

L'applicazione è stata progettata secondo i principi della **Modern App Architecture**, che garantisce una solida separazione delle responsabilità e una maggiore affidabilità.

L'architettura si compone dei seguenti tre livelli:

- **Presentation Layer:** gestisce l'interazione con l'utente, inclusi input e output, e si occupa della visualizzazione e dell'aggiornamento dei dati utilizzando **Jetpack Compose**. Comprende:
 - **UI Elements:** Composable che visualizzano i dati in modo dichiarativo.
 - **State Holders:** principalmente **ViewModel**, che gestiscono lo stato e la logica dell'interfaccia utente in modo reattivo utilizzando **StateFlow** o **LiveData**.
- **Domain Layer:** rappresenta il livello di collegamento tra il UI Layer e il Data Layer.
- **Data Layer:** implementa la logica di business e si occupa del salvataggio e della modifica dei dati. Comprende gli elementi **Data Source** (fonti di dati, come il database locale o servizi remoti) e **Repository** (fornisce un'interfaccia coerente per l'accesso ai dati, combinando e gestendo le diverse fonti di dati).

Inoltre, è stato adottato il design pattern **Dependency Injection** utilizzando **Hilt**, che permette alle classi di dichiarare le proprie dipendenze senza doverle creare direttamente. **Hilt** fornisce automaticamente le dipendenze a runtime, migliorando la modularità del codice e facilitando il testing.



Tecnologie Utilizzate

L'applicazione è stata sviluppata utilizzando **Kotlin**, uno dei linguaggi di programmazione più utilizzati per lo sviluppo di applicazioni mobile su Android, insieme a Java.

L'interfaccia utente è stata realizzata con **Jetpack Compose**, un moderno framework dichiarativo per la creazione di interfacce utente su Android, considerato generalmente più intuitivo e reattivo rispetto al tradizionale approccio basato su XML.

Backend e persistenza dei dati

Per la gestione del backend e della persistenza dei dati sono state utilizzate le seguenti tecnologie:

- **Firebase**: piattaforma di sviluppo di Google utilizzata per fornire funzionalità in tempo reale e sincronizzazione dei dati:
 - **Firebase Authentication**: gestisce l'accesso degli utenti tramite e-mail e password.
 - **Firebase Realtime Database**: salva i dati relativi agli oggetti smarriti o ritrovati, con supporto per aggiornamenti in tempo reale. L'utilizzo di Firebase Realtime Database garantisce **Live Update** e **Cross Device Synchronization**, che consentono di sincronizzare automaticamente i dati tra tutti i dispositivi connessi; ogni modifica ai dati nel database viene immediatamente inviata a tutti i client in ascolto, senza la necessità di aggiornamenti manuali o polling periodico.
 - **Firebase Storage**: gestisce le immagini degli oggetti smarriti o ritrovati.

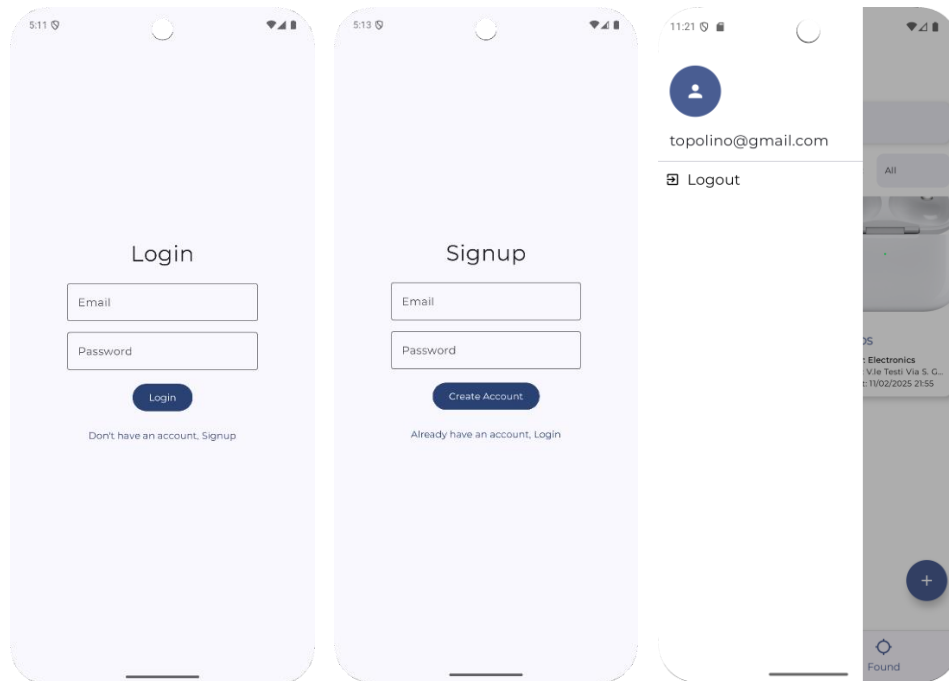
- **Room:** libreria di persistenza che fornisce un'astrazione sopra **SQLite**, facilitando l'interazione con il database locale tramite **DAO** (Data Access Objects).
- **Retrofit:** utilizzato per gestire le richieste HTTP verso servizi web e per l'interazione con le **API RESTful**, compresa la comunicazione con Firebase.

Mappe e servizi di geolocalizzazione

- **Google Maps API:** integra mappe interattive all'interno dell'applicazione, migliorando l'esperienza utente nella ricerca e localizzazione degli oggetti.
- **Google Places API:** fornisce informazioni dettagliate sui punti geografici e consente di cercare luoghi specifici.
- **Google Geocoding API:** utilizzata per convertire indirizzi fisici in coordinate geografiche (geocoding) e viceversa (reverse geocoding).

Funzionalità

Accesso e registrazione

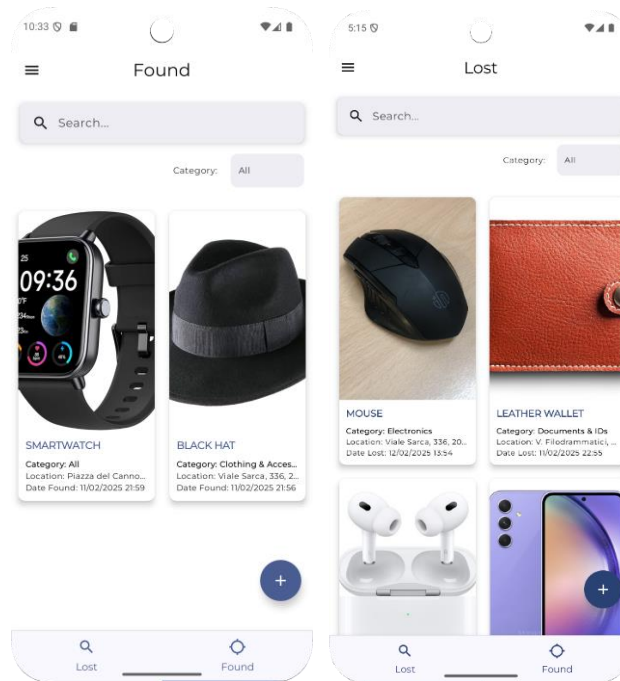


All'apertura dell'applicazione, viene visualizzata la **pagina di Login**, che permette all'utente di accedere inserendo le proprie credenziali (e-mail e password).

- Se l'utente possiede già un account, può autenticarsi rapidamente utilizzando Firebase Authentication, che garantisce un accesso sicuro e affidabile.
- In caso contrario, è possibile creare un nuovo account tramite la **pagina di Signup**, dove l'utente può registrarsi fornendo le proprie informazioni.
- Una volta completata la registrazione o l'autenticazione, l'utente viene reindirizzato alla schermata principale dell'applicazione.

L'utente può fare uso della funzionalità di Logout, che permette di uscire dal proprio account e tornare alla pagina di Login in modo sicuro

Schermata principale: Lost e Found

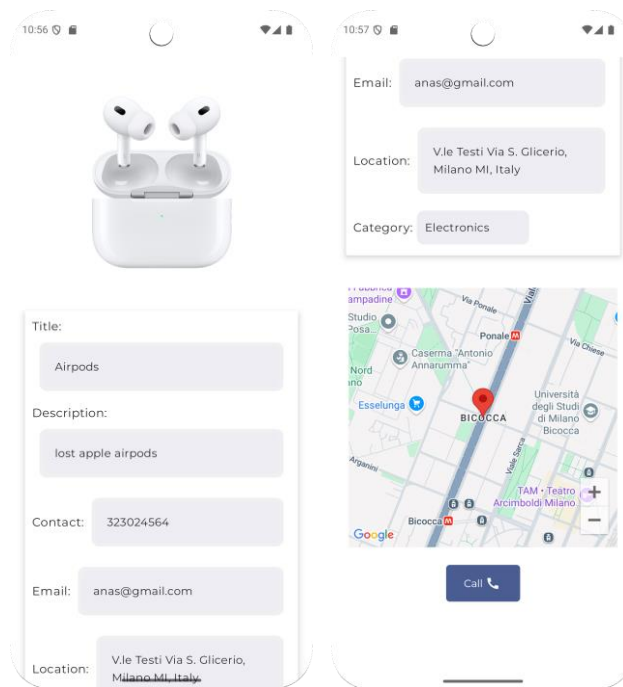


Questa è la schermata principale dell'applicazione, organizzata in due sezioni: **Lost**, per visualizzare gli oggetti smarriti, e **Found**, per consultare gli oggetti ritrovati.

Componenti Principali

- **Barra di Ricerca:** per cercare oggetti tramite parole chiave.
- **Filtro per Categoria:** consente di filtrare gli oggetti in base alla categoria, con
- **Lista degli oggetti:** visualizza gli oggetti in **card** contenenti alcune informazioni. È possibile cliccare sulla card per avere maggiori informazioni sull'oggetto
- **Action Button:** consente di aggiungere un nuovo oggetto ritrovato.
- **Navigation Bar:** permette di navigare tra le sezioni "Lost" (oggetti smarriti) e "Found" (oggetti ritrovati).

Schermata Oggetto



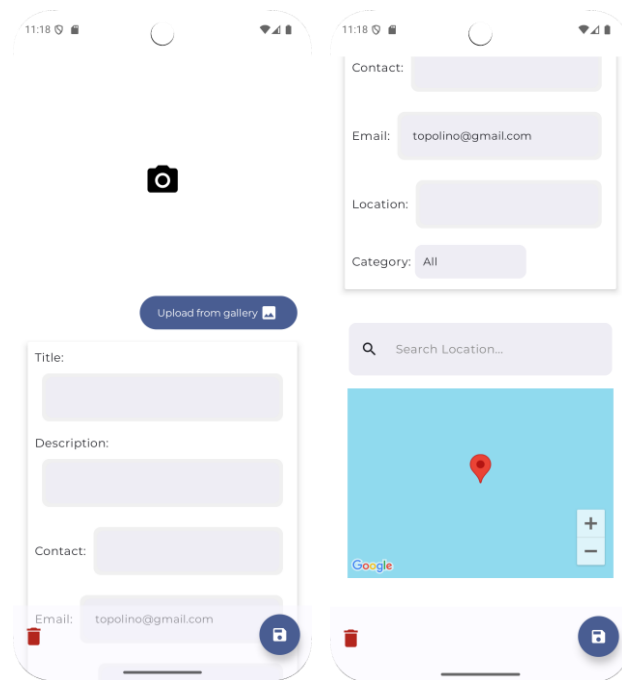
In questa schermata è possibile visualizzare tutti i dettagli relativi ad un oggetto perso/ritrovato.

Componenti:

- I **campi** contengono le informazioni tra cui: nome, descrizione, contatto, e-mail, indirizzo e la categoria.
- Un'**immagine** dell'oggetto.
- Una **mappa** che mostra la posizione esatta dell'oggetto, su cui è possibile cliccare per essere rediretti su Google Maps per poter raggiungere il luogo.
- Un **Action Button** per poter chiamare l'inserzionista.

Se l'utente è il proprietario dell'inserzione, potrà anche **modificare** i campi oppure **cancellare** l'inserzione con il relativo bottone.

Schermata di inserimento



Questa è la schermata per **inserire gli oggetti**.

Funzionalità:

- **Inserimento immagine** dalla galleria.
- **Compilazione testuale** dei campi, il campo e-mail viene auto compilato con l'email dell'utente.
- **Ricerca di un indirizzo**, su una search bar apposita, che verrà poi mostrato sulla mappa.

Dopo aver compilato tutto il form è possibile salvare, e pubblicare sulla bacheca.

Conclusioni e sviluppi futuri

Nonostante l'applicazione abbia raggiunto il suo obiettivo principale, ci sono numerosi miglioramenti che potrebbero essere implementati per migliorarne ulteriormente l'usabilità e la sicurezza:

- **Mappa unica per oggetti nelle vicinanze:** un possibile sviluppo riguarderebbe l'integrazione di una mappa unica che mostri, in tempo reale, tutti gli oggetti smarriti e ritrovati nelle vicinanze. Questo miglioramento faciliterebbe l'interazione tra gli utenti, permettendo loro di visualizzare facilmente gli oggetti disponibili nella zona.
- **Gestione del profilo utente:** gli utenti potrebbero, ad esempio, aggiornare le proprie informazioni personali, come numero di telefono e indirizzo e-mail, e visualizzare la cronologia degli oggetti smarriti e ritrovati segnalati, e modificare le impostazioni di notifica.
- **Notifiche:** altro sviluppo futuro potrebbe essere l'implementazione di notifiche push per avvisare gli utenti in tempo reale quando un oggetto smarrito che hanno segnalato è stato ritrovato, o quando un oggetto ritrovato corrisponde a una segnalazione precedente.
- **Funzionalità di ricerca avanzata:** potrebbe essere utile aggiungere una funzione di ricerca avanzata, che permetta agli utenti di cercare oggetti smarriti o ritrovati in base a vari criteri oltre alla categoria, come luogo, data e descrizione.
- **Live Chat:** un'aggiunta fondamentale per migliorare l'interazione tra gli utenti sarebbe l'introduzione di una funzionalità di live chat

per consentire la comunicazione diretta in tempo reale, scambiandosi informazioni.

- **Recupero delle credenziali:** per consentire a un utente di recuperare la propria password in caso di smarrimento.