

Il ruolo degli LLM nello sviluppo del software

**Un'analisi sul supporto di ChatGPT nel processo di realizzazione
dell'app Communimib**

**Relatore: Prof. Micucci, Correlatore: Dott. Maria Teresa Rossi
Luca Pinciroli 885969, A.A. 2023-2024**

Un minimo di contesto

- Ampia diffusione dei **Large Language Models** in molti contesti
- Una data storica: 30 novembre 2022, nasce **ChatGPT**
- La figura dello sviluppatore si adatta a nuovi strumenti
- **OBIETTIVO:** analizzare il ruolo di ChatGPT in materia di sviluppo software

Conduzione dell'esperimento

- **Sviluppo completo** di un'applicazione per dispositivi mobili utile all'ateneo
 - Analisi dei requisiti
 - Implementazione
 - Testing
- Redazione di un **registro** atto a catalogare le interazioni con ChatGPT

Communimib

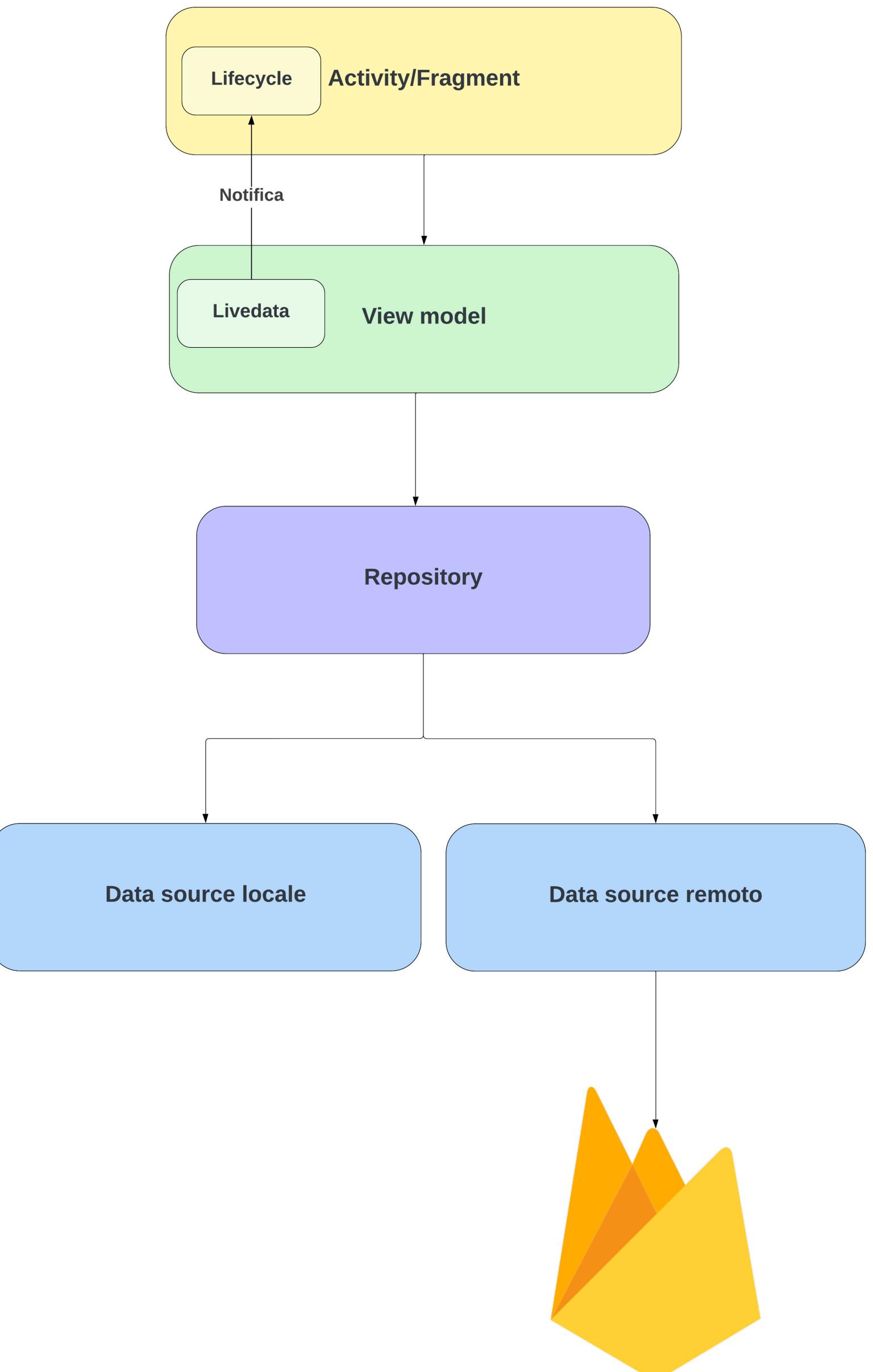
Il banco di prova per l'analisi sperimentale

- Strumento di comunicazione nato per interconnettere l'università
- Funzionalità chiave:
 - **Bacheca**
 - **Sistema di segnalazione**



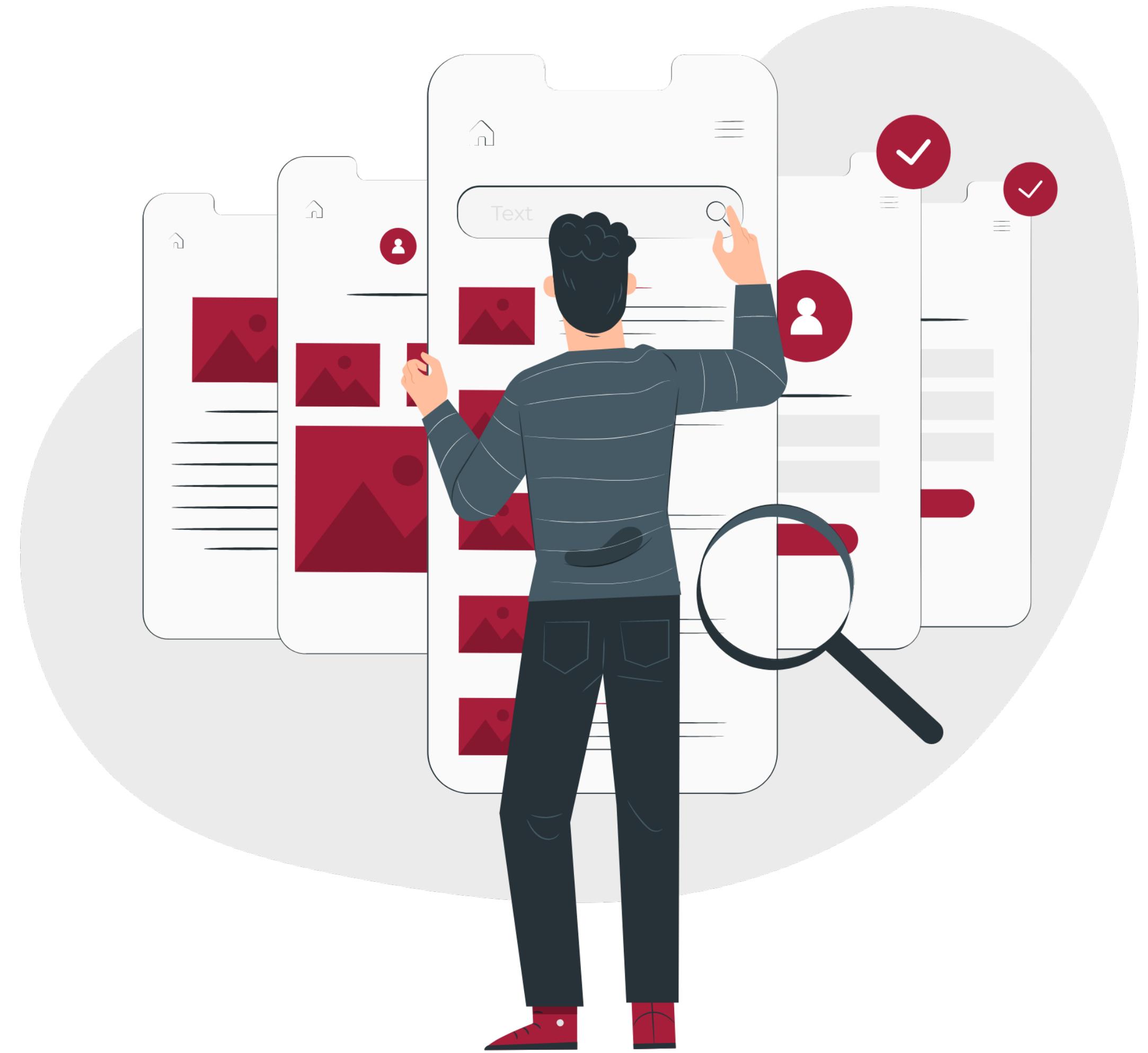
Uno sguardo sotto il cofano

- Applicazione nativa **Android** scritta in **Java**
- Modern App Architecture
- Pattern architetturale Model-View-ViewModel
- **Firebase** come sistema backend



Analisi dei requisiti

- Fase fondamentale per la buona riuscita del progetto
- Diverse competenze richieste
- Importante conoscere il dominio
- ChatGPT **non pervenuto**: nessuna interazione in questa fase



Implementazione

- Intense sessioni di scrittura di codice e bug-fixing
- **Uso corposo** di ChatGPT durante le diverse attività
- Efficacia altalenante, ma nel complesso ottima





Sto scrivendo un'app android in Java.
Come faccio, all'interno di un metodo di una classe, a creare un thread che esegua periodicamente del codice (a intervalli di 5 secondi) e che sia interrompibile manualmente all'occorrenza?



Per creare un thread che esegua periodicamente del codice a intervalli di 5 secondi e che sia interrompibile manualmente, puoi utilizzare un'implementazione con **'ScheduledExecutorService'**.

Questa classe fornisce un'interfaccia per eseguire task in modo pianificato, consentendo anche l'interruzione del task. Ecco un esempio di come potresti farlo all'interno del metodo di una classe in un'app Android in Java:

java

```
import java.util.concurrent.Executors;
import java.util.concurrent.ScheduledExecutorService;
import java.util.concurrent.TimeUnit;

public class MyClass {
    private ScheduledExecutorService executorService;

    public void startPeriodicTask() {
        executorService = Executors.newSingleThreadScheduledExecutor();
        executorService.scheduleAtFixedRate(() -> {
            // Codice da eseguire periodicamente
            // Ad esempio, puoi inserire qui la logica che desideri eseguire ogni 5 secondi
            // Assicurati di non bloccare il thread principale con operazioni lunghe
            // Se devi aggiornare l'interfaccia utente, utilizza Handler.post o
runOnUiThread
            }, 0, 5, TimeUnit.SECONDS);
    }

    public void stopPeriodicTask() {
        if (executorService != null && !executorService.isShutdown()) {
            executorService.shutdownNow();
        }
    }
}
```

Testing

- Validazione del funzionamento dell'app
- ChatGPT non sempre eccelle, ma è **fondamentale** in questa fase
- Codice da sistemare, ma **buoni suggerimenti**





...

Implementa il metodo `getUserByEmail`.



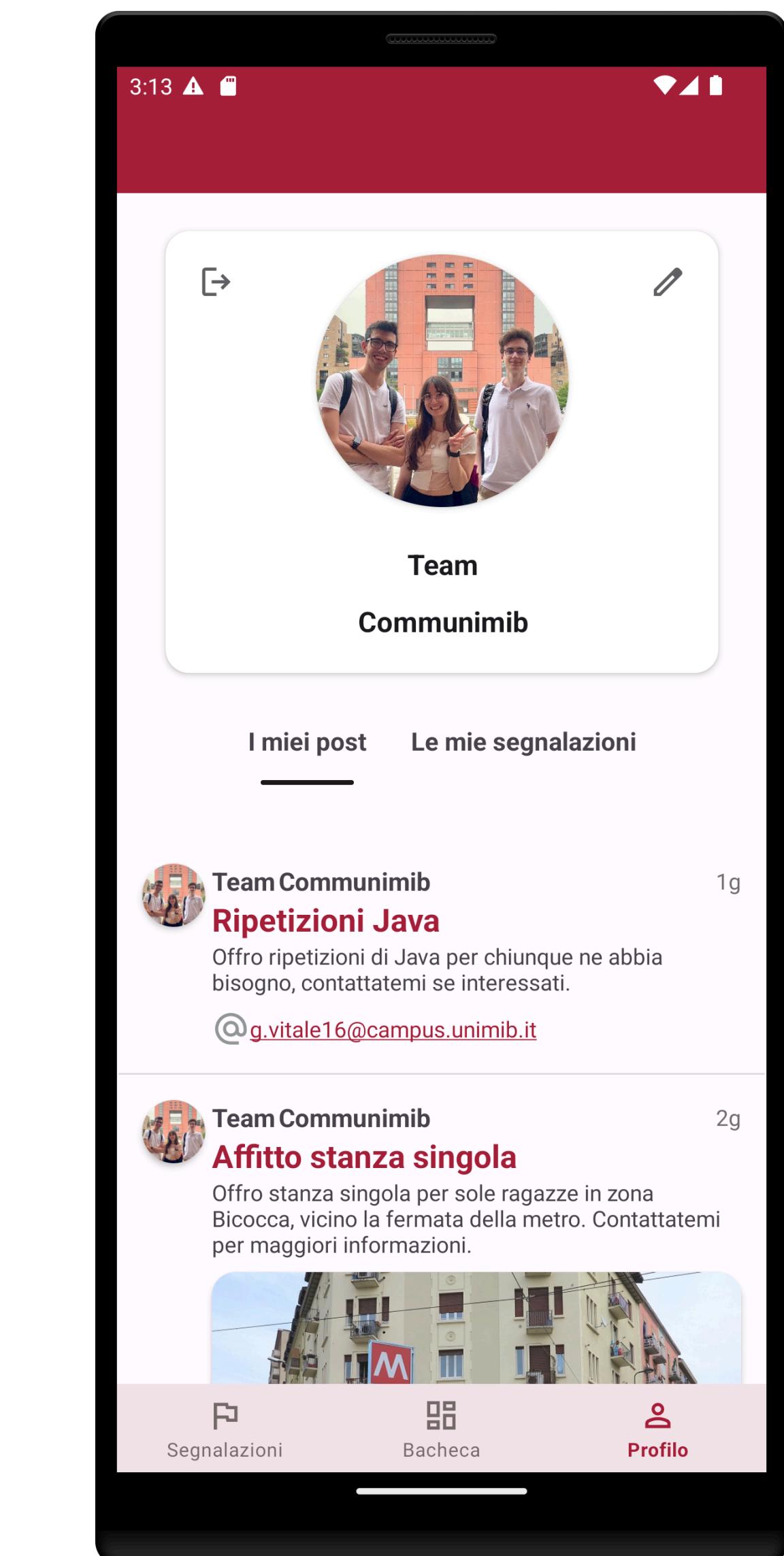
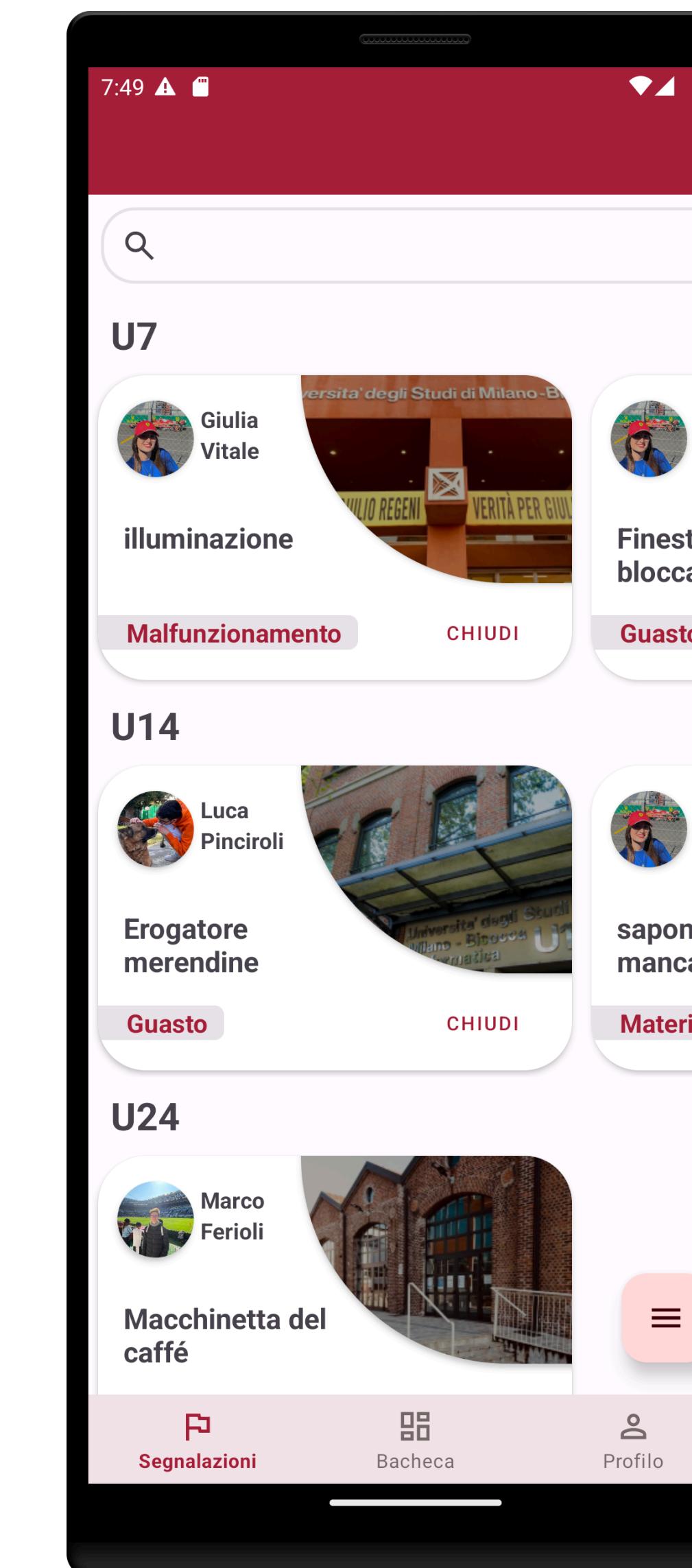
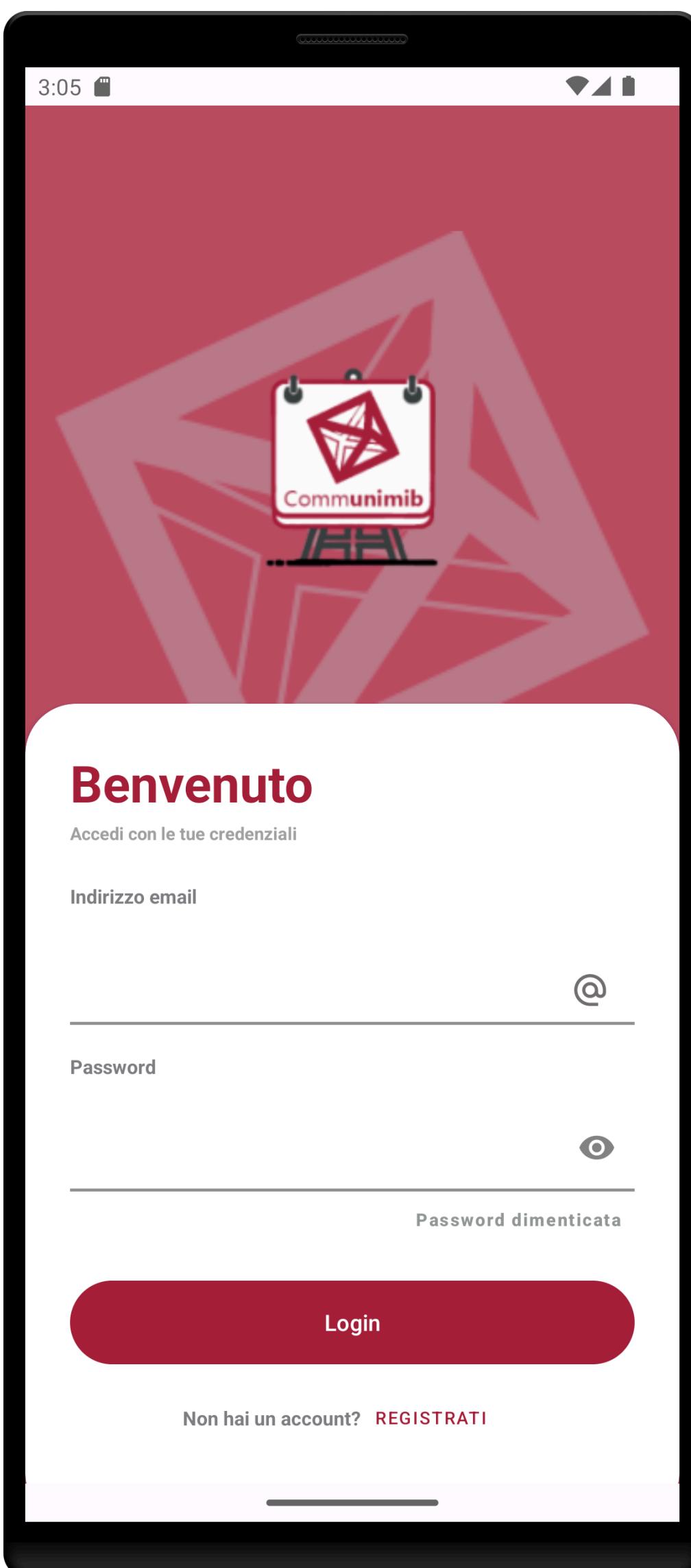
Per implementare il metodo `'getUserByEmail'` nella classe `'FakeUserRemoteDataSource'`, dovresti iterare attraverso la mappa degli utenti e cercare un utente con l'email specificata.

Ecco un esempio di come potresti farlo:

java

```
@Override  
public void getUserByEmail(String email, Callback callback) {  
    User foundUser = null;  
    for (User user : users.values()) {  
        if (user.getEmail().equals(email)) {  
            foundUser = user;  
            break;  
        }  
    }  
    if (foundUser != null) {  
        callback.onComplete(new Result.Success(foundUser));  
    } else {  
        callback.onComplete(new Result.Error("User not found with email: " + email));  
    }  
}
```

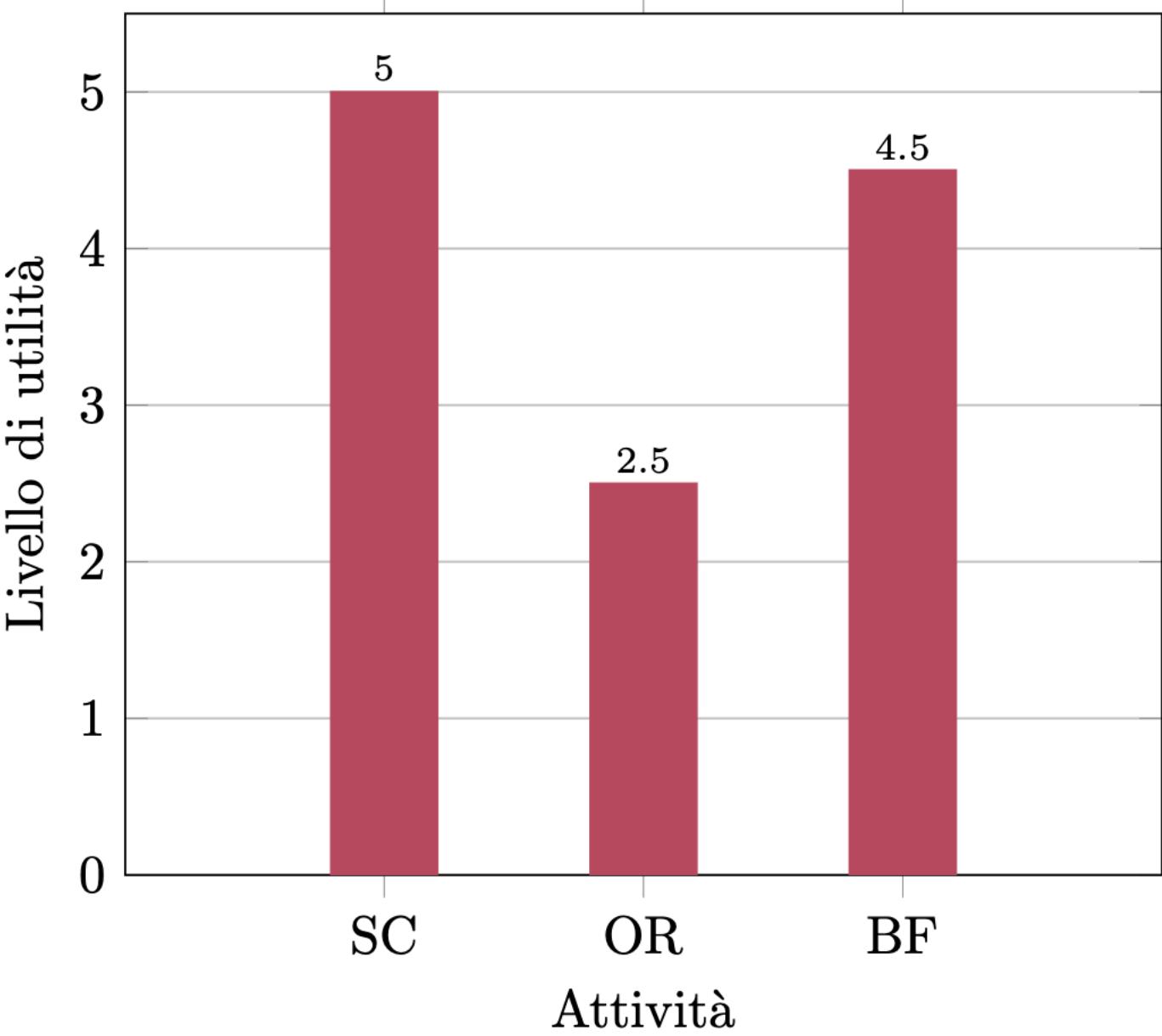
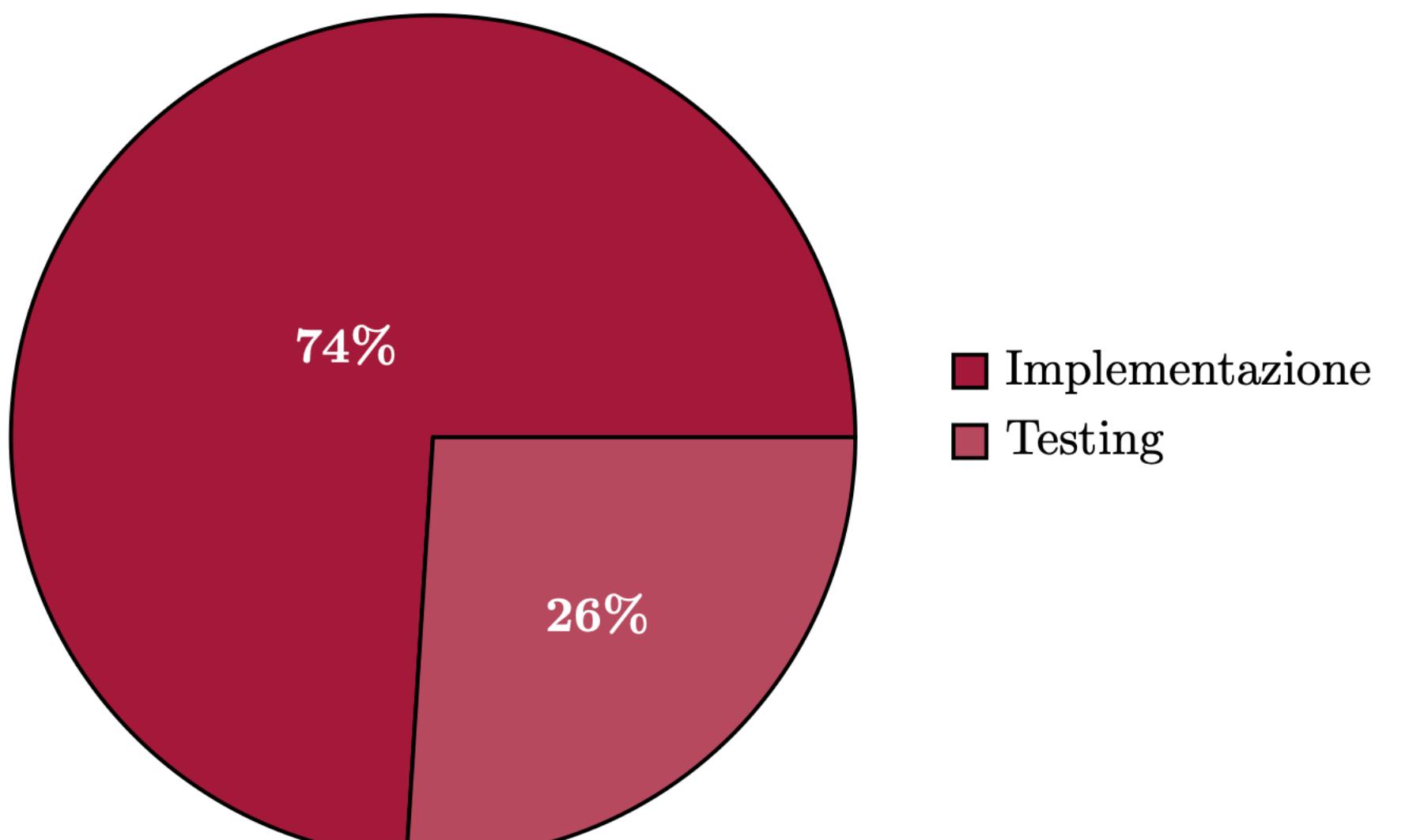
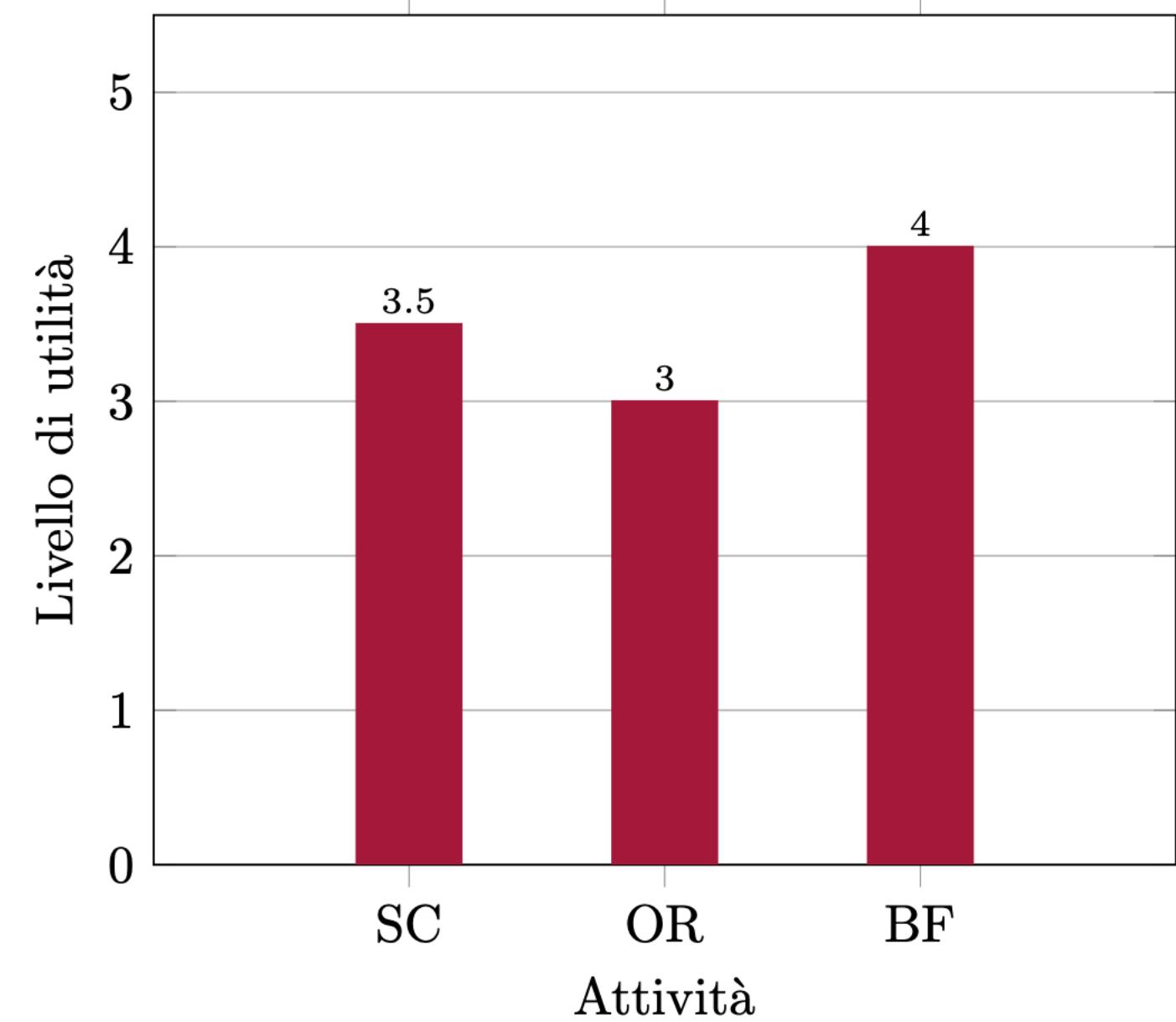
In questo metodo, scorriamo attraverso tutti gli utenti nella mappa `'users'` e confrontiamo l'email di ciascun utente con l'email fornita come parametro. Se troviamo un utente con l'email corrispondente, chiamiamo il metodo `'onComplete'` del callback con un oggetto `'Result.Success'` contenente l'utente trovato. Se non troviamo alcun utente con l'email specificata, chiamiamo `'onComplete'` con un oggetto `'Result.Error'` contenente un messaggio appropriato.



ChatGPT

strumento inutile, fedele alleato o minaccia da temere?

Dati alla mano...



Grazie per l'attenzione

Luca Pinciroli 885969, A.A. 2023-2024