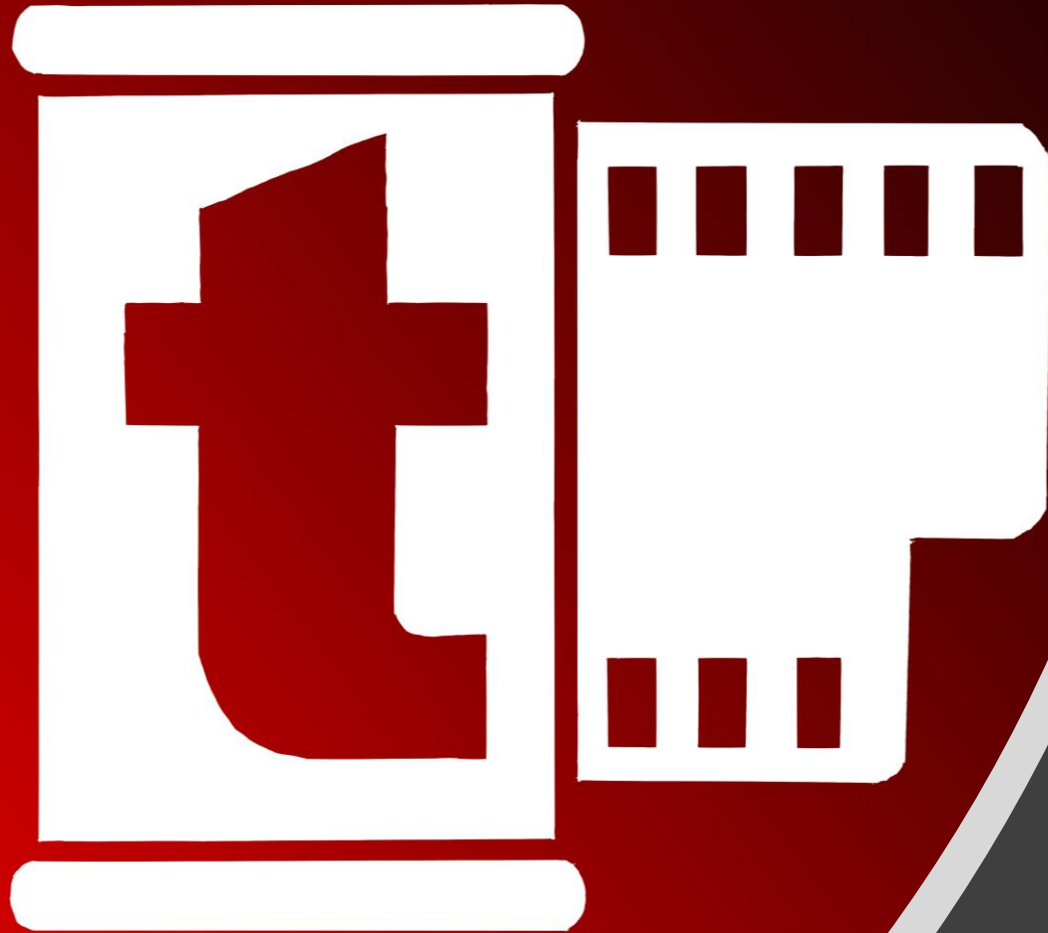


Presentazione progetto  
Programmazione Dispositivi Mobili



# TripPic

Share your best trip memories  
with your friends!

# Our Team



**Chiara**  
ANDREATA



**Giacomo**  
GRANDI



**Federico**  
PEIRETTI

# TripPic Purpose



## **Visualizzazione contenuti su Google Maps**

Facile interazione dell'utente con la mappa Google per visualizzare i contenuti

## **Salvare i contenuti pubblicati dagli amici**

L'utente può visualizzare i contenuti pubblicati dagli amici e salvarli in un proprio elenco di preferiti

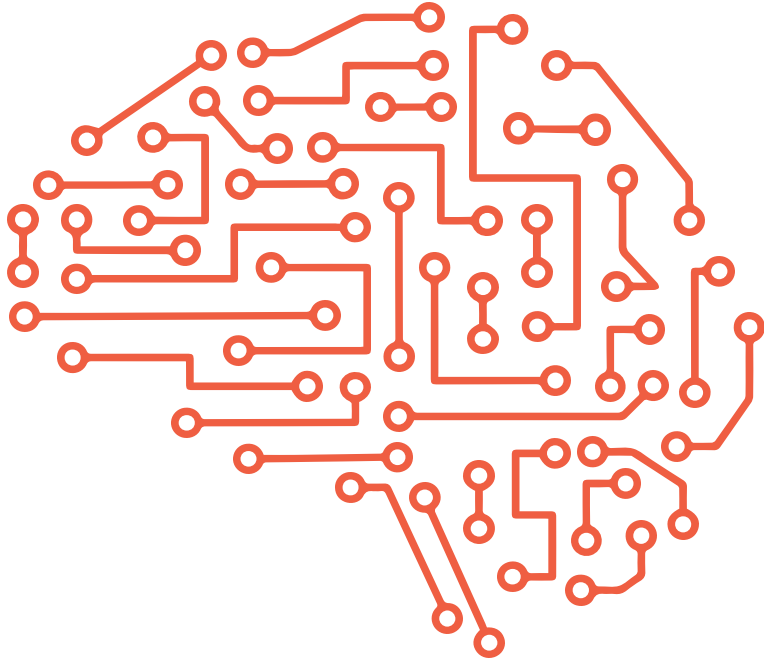
## **Caricare le proprie foto di viaggio**

L'utente può caricare immagini direttamente dal proprio smartphone tramite la galleria o la fotocamera

## **Scoprire nuove mete**

Visualizzando i contenuti degli amici l'utente può scoprire nuovi interessanti posti da visitare

# Project Idea



**TripPic è un'applicazione per dispositivi Android. Si tratta di un social network dedicato a chi ama viaggiare e condividere le proprie esperienze.**

**A differenza dei social più noti, non si tratta solo di una raccolta multimediale, vengono infatti messe in primo piano le località visitate.**

**Ogni utente potrà caricare i propri scatti direttamente sulla mappa di Google.**

**In questo modo condividere la propria esperienza di viaggio è più semplice e divertente.**

# Project Goals

## Diario

- TripPic permette ai suoi utenti di creare un diario di viaggio personale con foto e descrizione dei luoghi visitati
- Ogni utente potrà avere accesso ai suoi contenuti, e crearne di nuovi, ovunque si trovi: basta una connessione internet

## Mappa

- TripPic si basa sull'integrazione delle mappe di Google per caricare i contenuti dell'utente sulla cartina
- Le potenzialità delle note Google Maps vengono messe a disposizione dell'utente

## Social

- Condividere le proprie esperienze e fotografie di viaggio è il fulcro di TripPic.
- Questa applicazione permette una condivisione di contenuti molto facilitata

## Ispirazione

- TripPic permette ai suoi utenti di scoprire nuove località da visitare grazie ai contenuti caricati dai loro amici.

# Project No-Goals

## Notorietà e competizione

- TripPic non è un social network in cui gli utenti possono essere in competizione per la notorietà: non è progettato per mettere in risalto alcuni utenti rispetto ad altri
- Ogni utente per visualizzare i contenuti di altri utenti deve prima aggiungerli come "amici".  
In questo modo la privacy dei contenuti caricati è tutelata
- TripPic non tiene conto del numero di amici degli utenti

## Like

- TripPic non tiene conto e non mostra il numero di contenuti salvati da altri utenti
- Gli utenti possono salvare i contenuti dei propri amici per visualizzarli in una lista separata, ma non possono sapere quante persone hanno salvato un loro contenuto

## Commenti

- TripPic non permette di commentare gli altrui contenuti.
- I contenuti caricati su TripPic non devono essere ragione di liti e "flame" come spesso accade nei social più noti.

# Competitors



## INSTAGRAM



Nota applicazione per la condivisione di immagini con gli amici



## PLACES BEEN



Applicazione che permette di contrassegnare su una mappa personale le città visitate

## JOURNI BLOG



Applicazione che permette di creare un blog pubblico o privato per caricare le proprie foto di viaggio



# Timeline

## scheda progetto

Definizione di obiettivi e caratteristiche dell'applicazione da sviluppare

**Maggio**

### Main Activity

Implementazione della main activity e fragment  
integrazione delle Google API per la mappa

### Jetpack Compose

Utilizzo di Jetpack Compose per la creazione delle interfacce  
Creazione barra di navigazione

**Ottobre**

**Settembre**

### Revisione e test finali

Revisione del progetto, test su differenti dispositivi, preparazione di slides e relazione

### Backend

Sviluppo del backend con integrazione di AWS

### Sviluppo UI

Sviluppo delle interfacce e delle funzionalità principali.  
Permessi di accesso

**Esame**

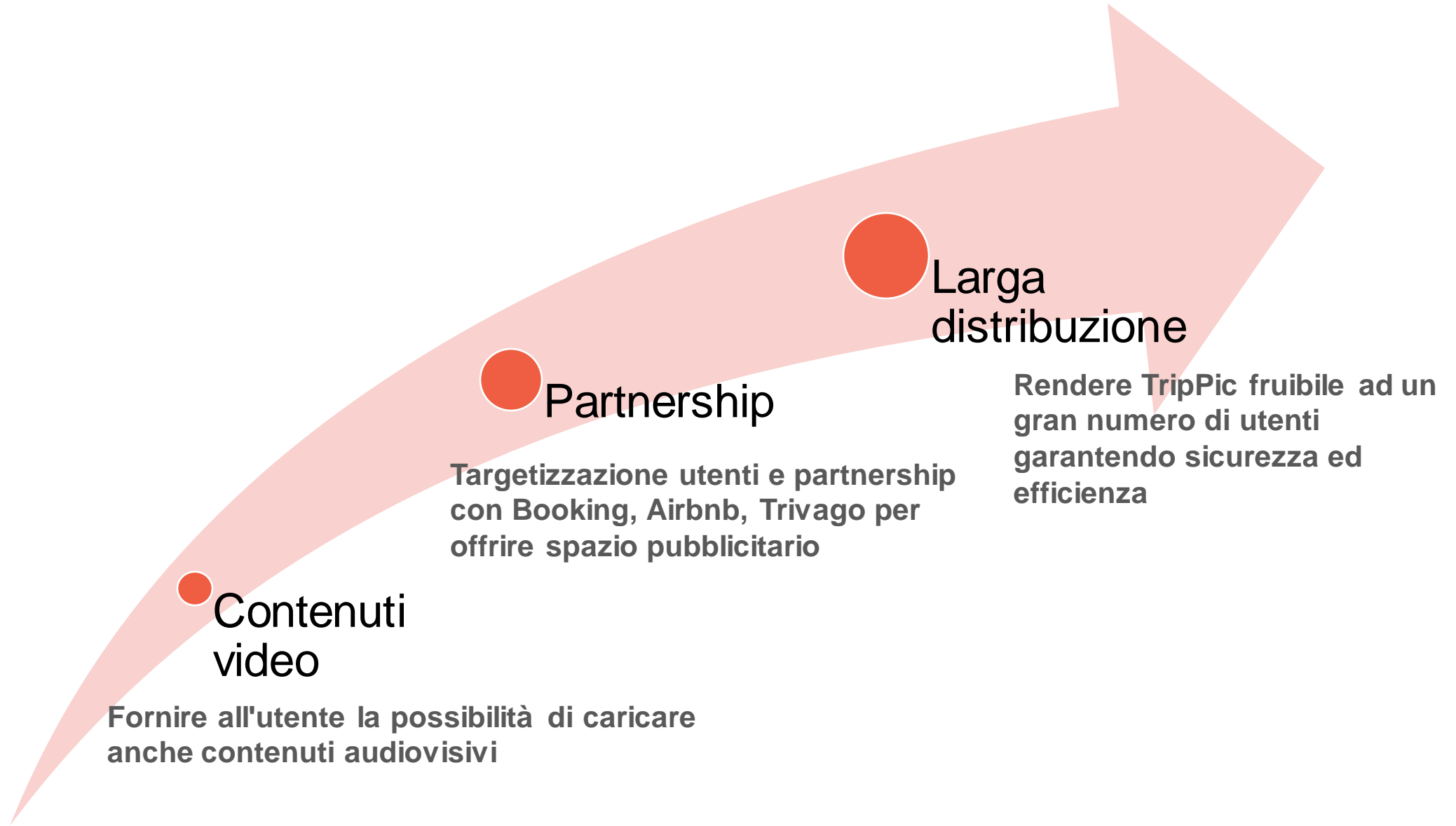
**Gennaio**

**Dicembre**

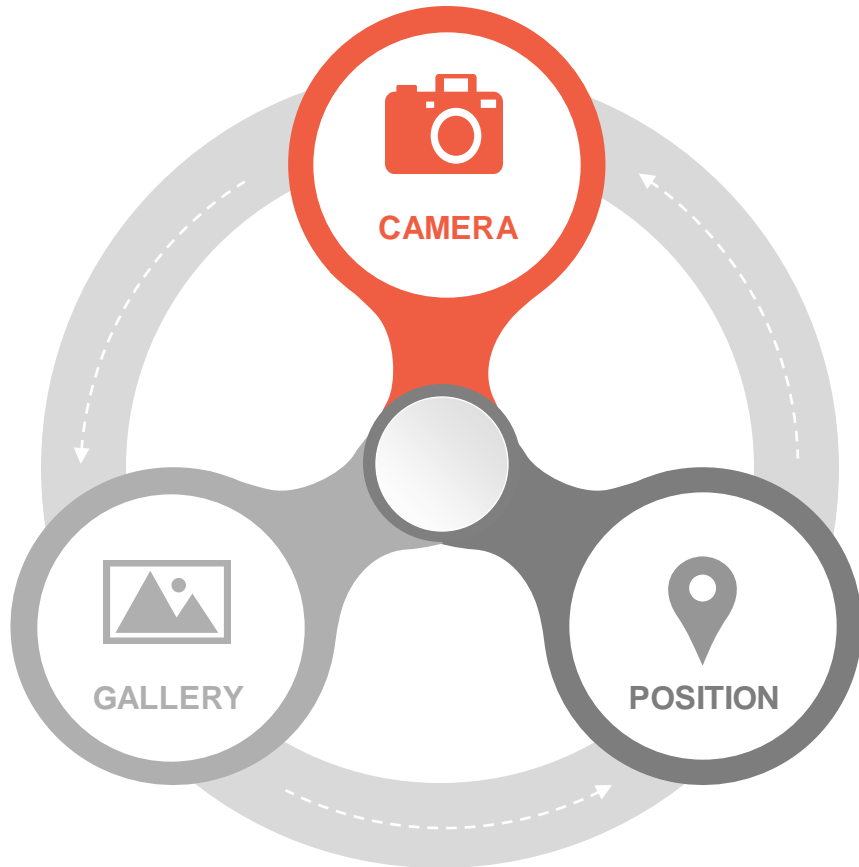
**Novembre**



# Future Developments



# Required Permissions



- 01** GALLERY PERMISSION  
Per creare contenuti l'utente può caricare le proprie fotografie dalla sua galleria. Viene all'utente richiesto il permesso di accedere alla galleria del proprio cellulare.
- 02** CAMERA PERMISSION  
L'utente può inoltre creare nuovi contenuti direttamente scattando immagini con la fotocamera del proprio smartphone. Per questa operazione è necessario che l'utente conceda il permesso.
- 03** POSITION PERMISSION  
L'utente può caricare contenuti associando direttamente la propria posizione. Per fare ciò deve consentire l'accesso di TripPic alla propria posizione.

# Content Sharing

Condivisione dei contenuti mediante i più noti social



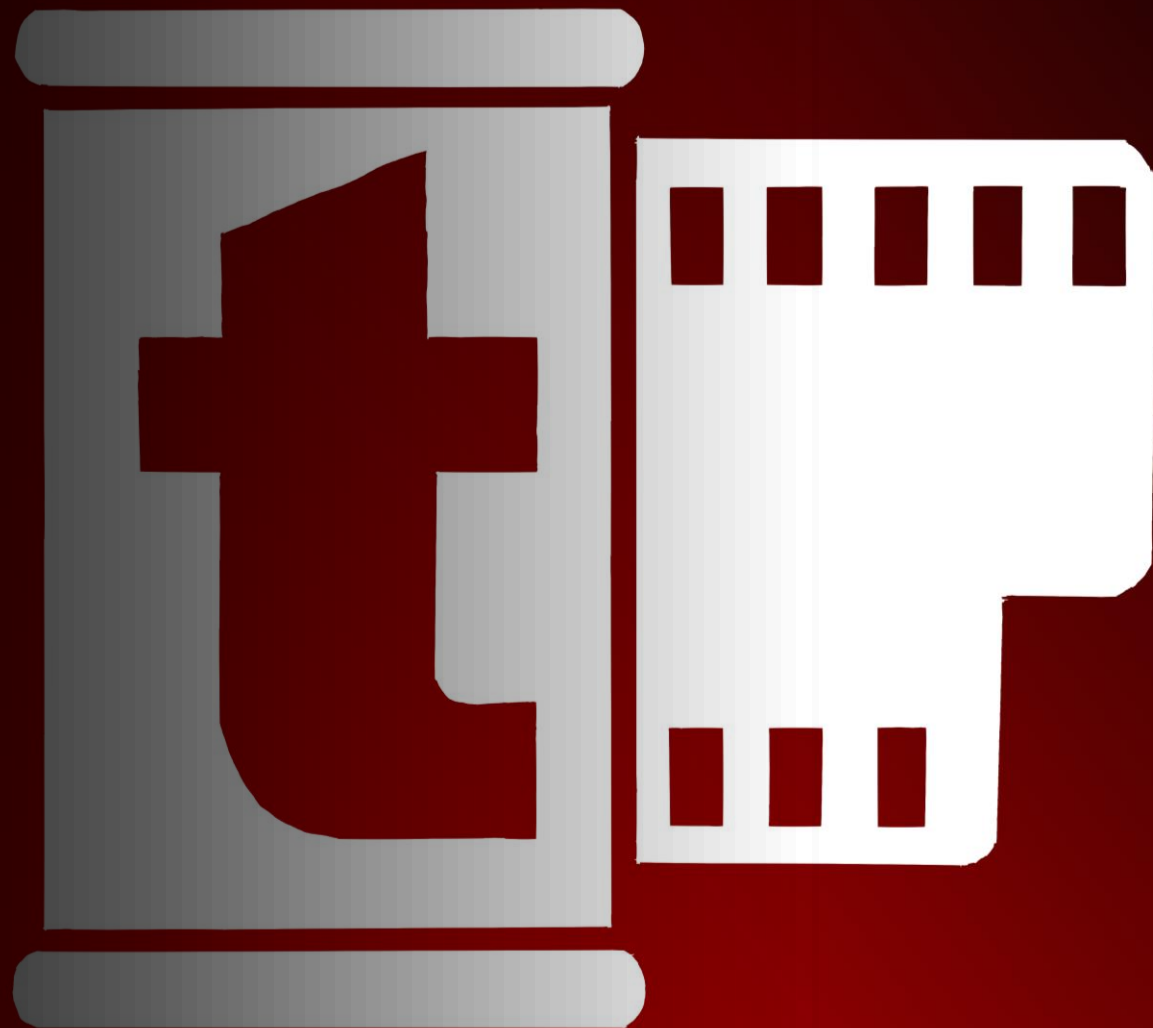
L'utente potrà condividere le proprie immagini, o quelle degli amici, anche su altre piattaforme.

Tra le più diffuse e utilizzate:  
Instagram  
Telegram  
Whatsapp

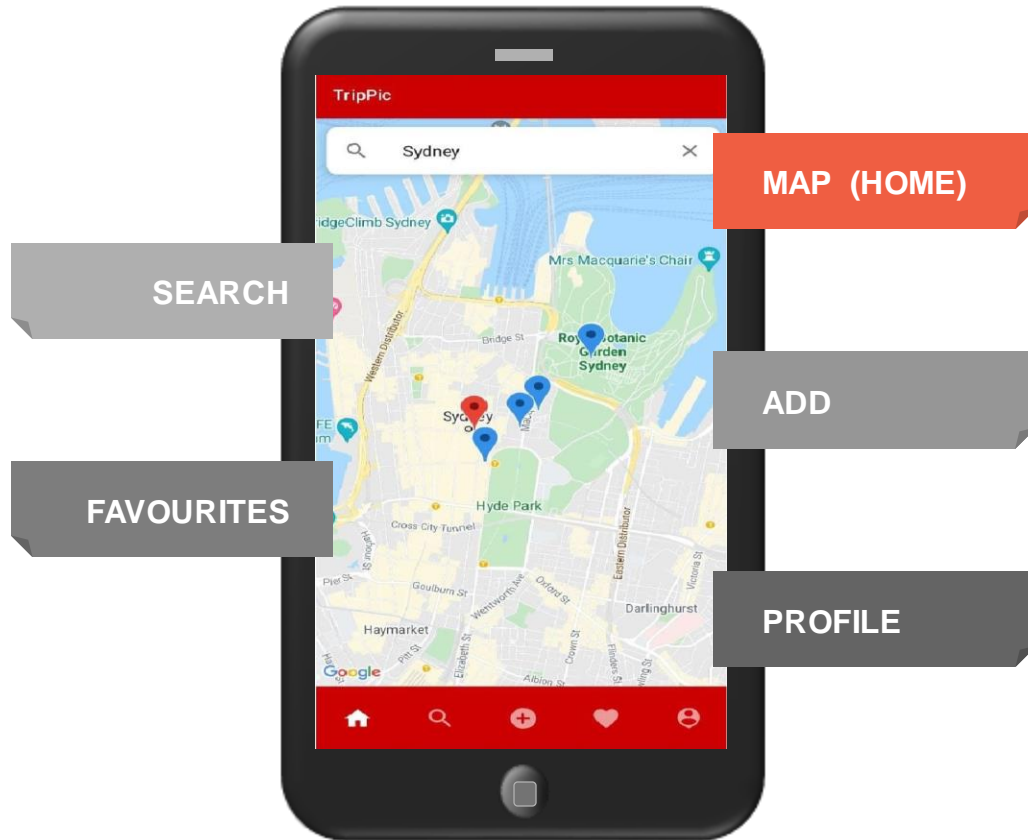


LET'S GO  
INSIDE

---



# Main Functionalities



## MAP (HOME)

La schermata principale di TripPic mostra la mappa di Google con i pin associati ai contenuti degli utenti



## SEARCH

Schermata che permette all'utente di cercare gli altri utenti digitando lo username e di visualizzare i contenuti digitandone il titolo



## ADD

Schermata che permette all'utente di creare un nuovo contenuto caricando un'immagine dalla Gallery o scattando una foto



## FAVORITES

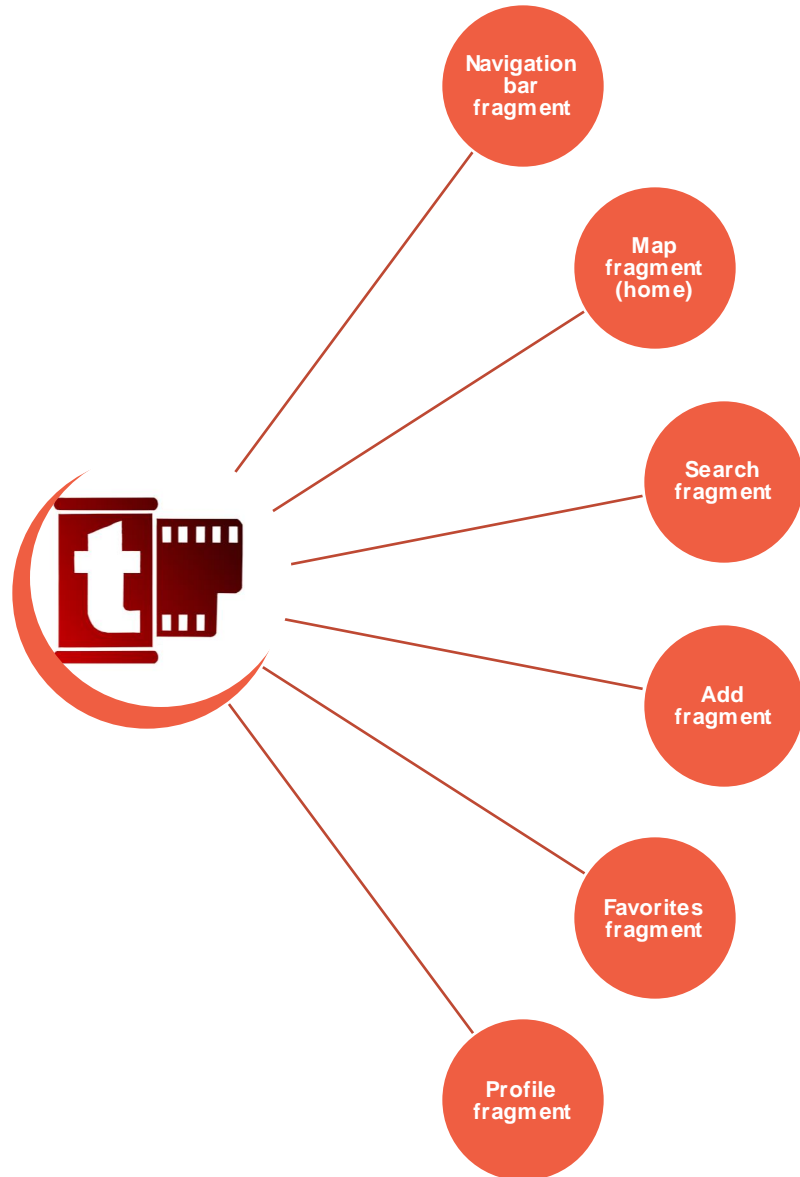
Schermata che permette all'utente di visualizzare l'elenco dei contenuti salvati



## PROFILE

Schermata che permette all'utente di visualizzare le proprie informazioni, i propri contenuti e i propri amici

# Frontend Architecture



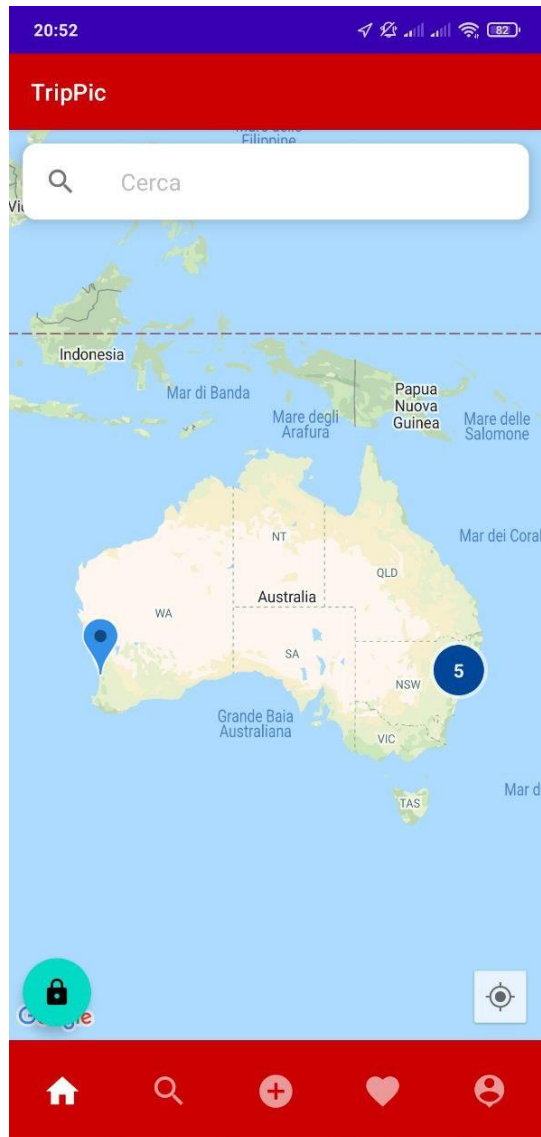
Il frontend di TripPic è strutturato a partire da una activity principale e sottostanti fragment. La scelta di questa struttura si basa sulla necessità di avere una barra di navigazione in basso che permetta all'utente di spostarsi attraverso le varie parti dell'applicazione in modo semplice, intuitivo e rapido.

Tutte le viste dell'applicazione sfruttano gli stessi oggetti kotlin utilizzati per rappresentare utenti e contenuti.

## INTENTS

Per poter caricare contenuti dalla galleria e dalla fotocamera abbiamo utilizzato gli Intents. Allo stesso modo è stato necessario utilizzarli per la condivisione delle immagini su altre piattaforme (come Telegram, Whatsapp...)

# Home Fragment



Il Fragment principale dell'app mostra la mappa fornita da Google.

Da qui è possibile:

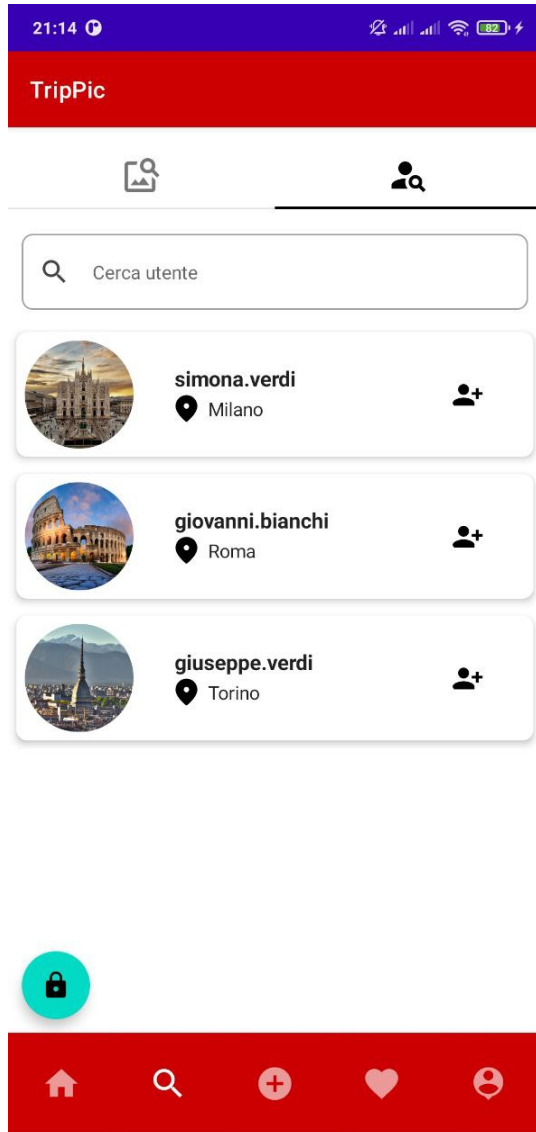


Cercare luoghi sulla mappa tramite il motore di ricerca di Google Maps



Cliccare sui pin presenti nella mappa per poter visualizzare il contenuto relativo

# Search Fragment



Questo fragment permette di cercare:



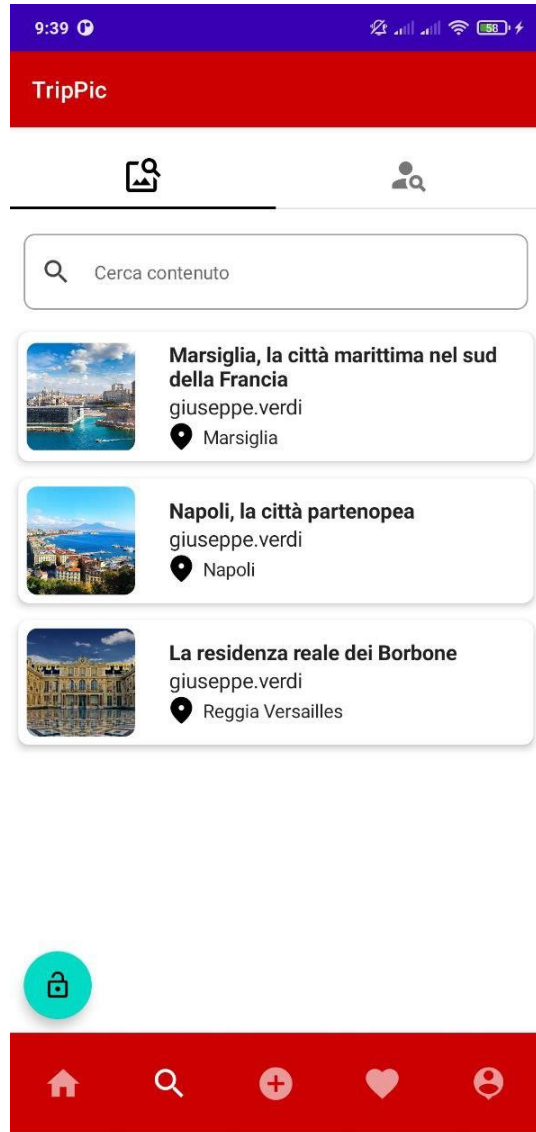
Post specifici tramite il titolo



Utenti tramite lo username, in modo da poter visualizzare il loro profilo e i loro post e aggiungerli ai propri amici



# Search Fragment



Questo fragment permette di cercare:

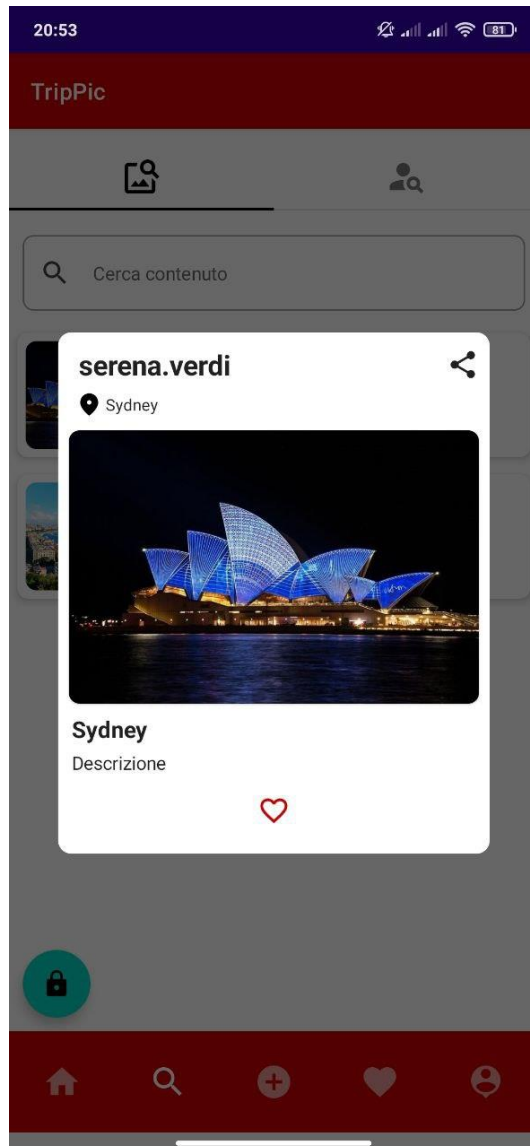


Post specifici tramite il titolo



Utenti tramite lo username, in modo da poter visualizzare il loro profilo e i loro post e aggiungerli ai propri amici

# Show Post



Quando viene aperto un contenuto da una lista di post di un utente si espande una finestra che permette di visualizzare le informazioni del contenuto caricato, condividerlo o salvarlo tra i propri post preferiti.

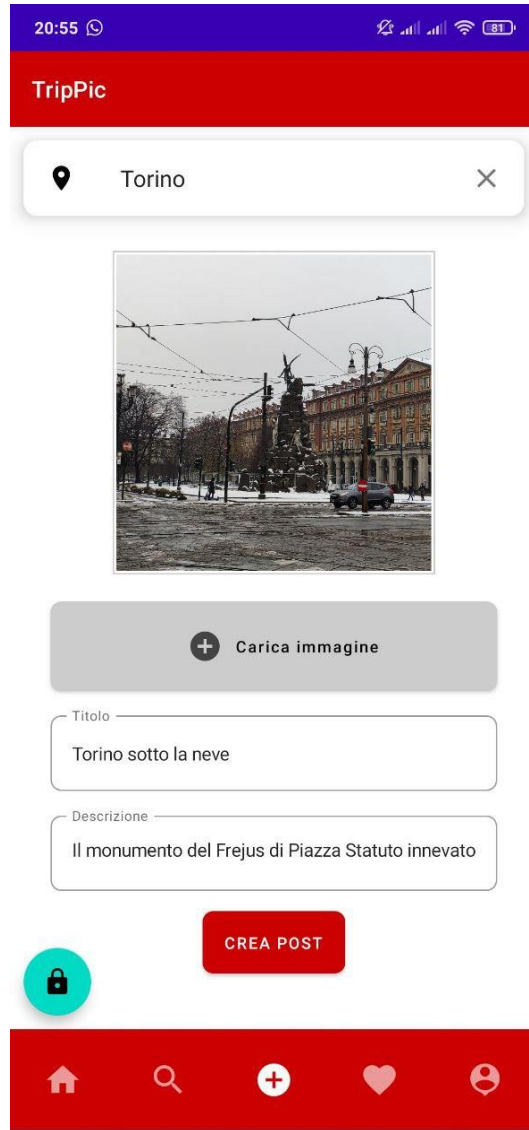


Condivisione dell'immagine del contenuto visualizzato mediante altre applicazioni, ad esempio Telegram, Whatsapp..



Salvataggio del contenuto per poterlo visualizzare in un elenco di preferiti.

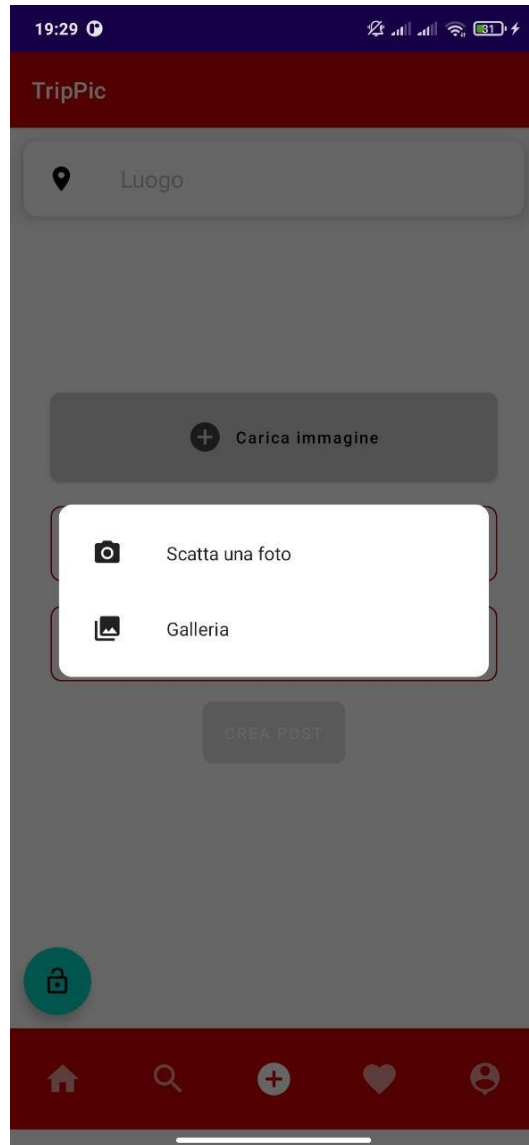
# Add Fragment



In questo fragment l'utente può caricare un nuovo contenuto sul proprio profilo, dovrà occuparsi di:

- Selezionare una **foto** dalla galleria immagini del proprio dispositivo
- Scegliere un **titolo** per il nuovo post
- Scrivere una **descrizione** relativa alla foto
- Scegliere una **posizione** per permettere di visualizzare il pin relativo al post sulla mappa

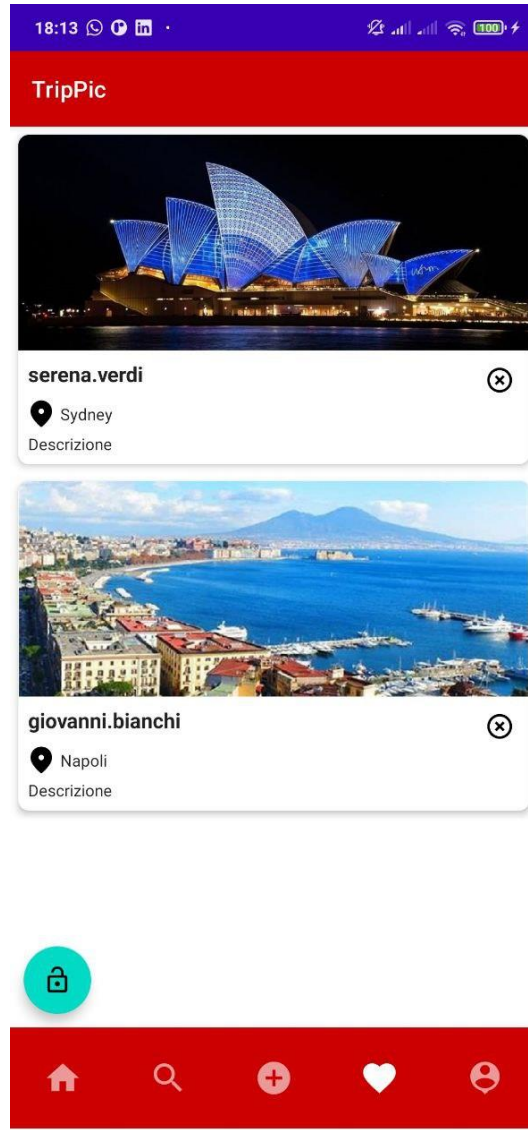
# Add Fragment



In questo fragment l'utente può caricare un nuovo contenuto sul proprio profilo, dovrà occuparsi di:

- Selezionare una **foto** dalla galleria immagini del proprio dispositivo
- Scegliere un **titolo** per il nuovo post
- Scrivere una **descrizione** relativa alla foto
- Scegliere una **posizione** per permettere di visualizzare il pin relativo al post sulla mappa

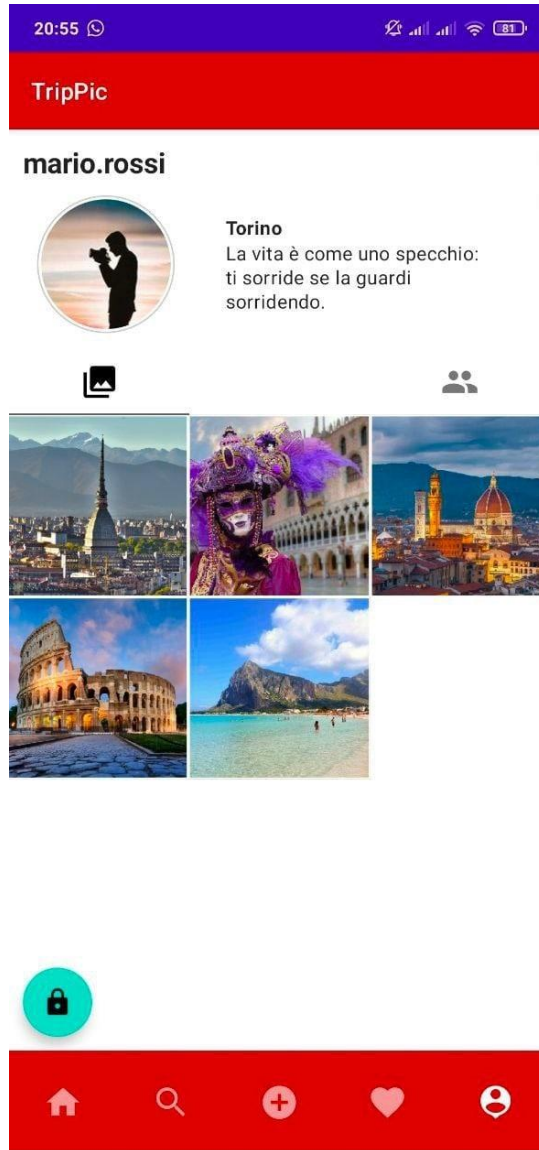
# Favorite Fragment



Questo fragment mostra un **elenco** di tutti i contenuti che l'utente ha inserito tra i **preferiti**.

Cliccando su uno qualsiasi dei post sull'elenco viene mostrato un popup per visualizzare la foto, il titolo, la descrizione e lo username dell'autore del contenuto.

# Profile Fragment



In questo fragment viene visualizzato il proprio profilo: la **foto** profilo, lo **username**, la **città** di provenienza e una breve **biografia**.

Si può inoltre scegliere se visualizzare:

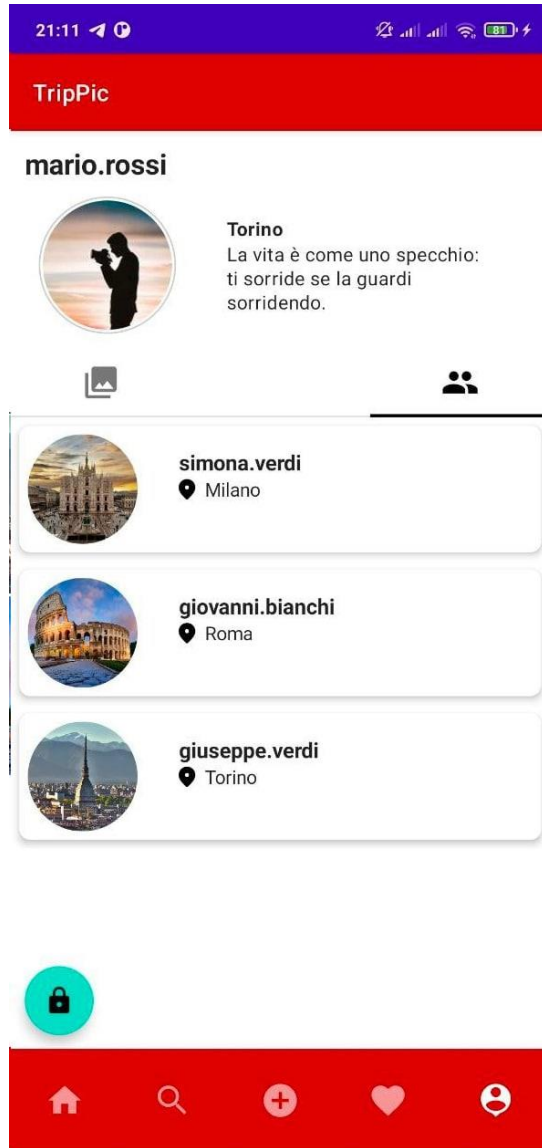


Una griglia dei propri post



L'elenco degli amici

# Profile Fragment



In questo fragment viene visualizzato il proprio profilo: la **foto** profilo, lo **username**, la **città** di provenienza e una breve **biografia**.

Si può inoltre scegliere se visualizzare:



Una griglia dei propri post



L'elenco degli amici

# User Authentication

20:58

3d6f3-dev.auth.eu-central-1.amazonaws.com

Sign In with your social account

Continue with Facebook

We won't post to any of your accounts without asking first

OR

Sign in with your username and password

Username

Password

[Forgot your password?](#)

[Sign in](#)

[Need an account? Sign up](#)

inglese italiano

Mediante questa schermata l'utente ha la possibilità di registrarsi e di accedere al mondo di TripPic.

Deve inserire una foto profilo, fornire una descrizione personale o un pensiero e indicare la città di provenienza

L'utente deve inserire uno username che sarà univoco e identificativo all'interno dell'applicazione

**Se l'utente era già presente nella piattaforma TripPic anziché compiere una nuova iscrizione può effettuare semplicemente il login**

La registrazione e il login sono stati da noi implementati sfruttando il servizio Amazon Cognito



---

# Android Jetpack Compose

---

Frontend  
Development



# Why Jetpack Compose

## ONLY KOTLIN

Jetpack Compose permette di non scrivere codice in XML: si utilizza solo Kotlin. Permette di creare interfacce componibili che sfruttano tutte le potenzialità del linguaggio Kotlin

## LESS CODE

Jetpack Compose permette di creare Ui in modo rapido e senza scrivere moltissimo codice rispetto ai modi tradizionali di sviluppo in Android. Permette inoltre di ridurre notevolmente le occasioni di bug e codice da testare



## DECLARATIVE API

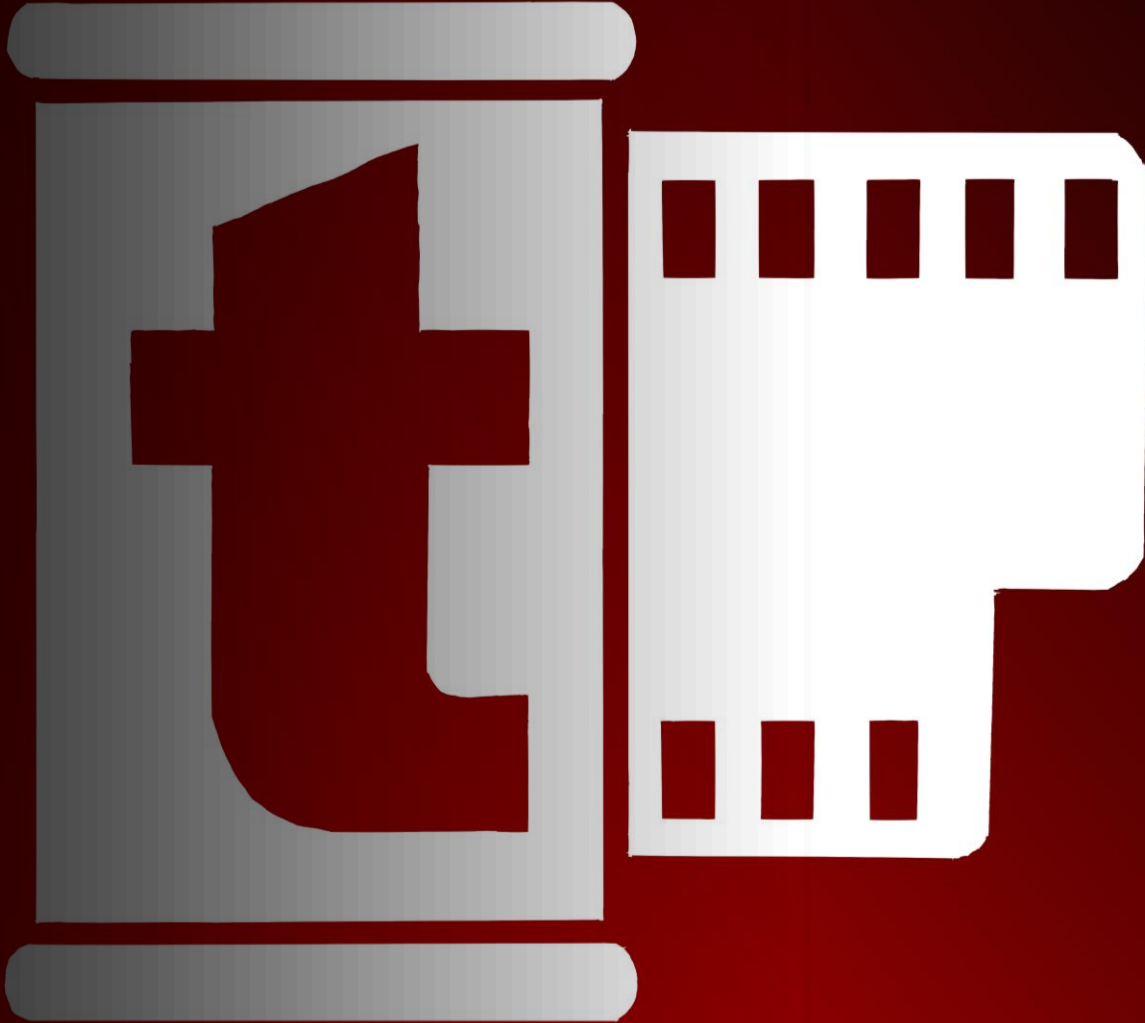
Jetpack Compose utilizza Declarative API, che permettono di descrivere la funzione dei vari elementi dell'interfaccia. Le componenti non sono legate ad uno specifico fragment, ciò permette di riutilizzarle facilmente

## COMPATIBLE WITH EXISTING CODE

Non è necessario iniziare a sviluppare l'applicazione in Compose, o nel caso si abbiano già delle funzionalità dell'app non è obbligatorio riscriverle. Si può migrare a Jetpack Compose in modo incrementale. Noi lo abbiamo fatto dopo più di un mese di sviluppo.



# ARCHITECTURE



# Model View ViewModel

Il pattern architetturale scelto per la progettazione di TripPic è il **Model View ViewModel (MVVM)**.

Si basa sul principio **Separation of concerns**: separare la logica dell'interfaccia utente dal back-end.

*"Mantenere il codice dell'interfaccia utente semplice e privo di logica dell'app per semplificarne la gestione".*

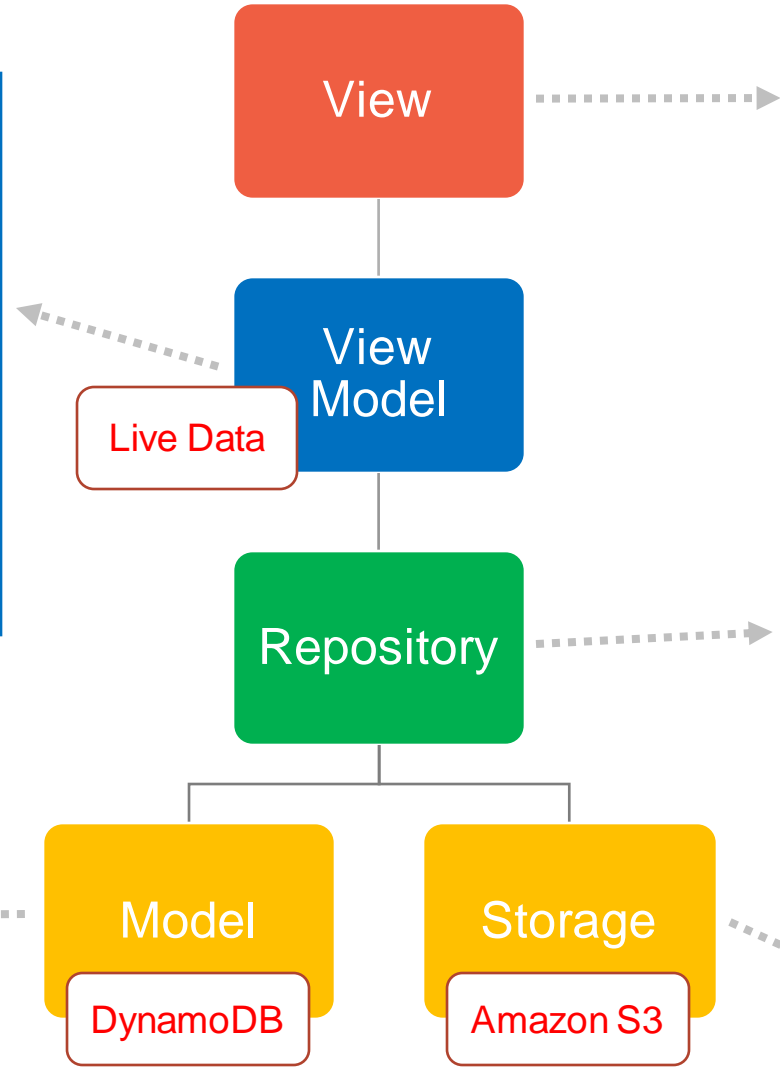
Il ViewModel **fornisce i dati alla View** e riceve da questa direttive sulle operazioni da eseguire.

**Il ViewModel non è a conoscenza della View** con cui sta interagendo.

Mantiene e aggiorna i **LiveData**, oggetti observable che sopravvivono all'andamento del ciclo di vita. Al loro interno sono memorizzati i dati ottenuti dal Model e servono per popolare la UI.

Database NoSQL **DynamoDB**  
**GraphQL API**

Sulla base della definizione del modello di dati GraphQL, Amplify genera un codice lato client per rappresentare i dati (classi Java che implementano Model).



La UI contiene Activity e Fragments.

La View ha un **riferimento al ViewModel** e **osserva gli oggetti LiveData** per ottenere i dati e aggiornare la UI.

Riceve le **interazioni con l'utente** e le notifica al ViewModel per la gestione.

Si tratta di una classe **singleton** che gestisce le query e consente di utilizzare molteplici backend. **Implementa la logica** per decidere se recuperare i dati da db, storage, file o altro.

Da qui il **ViewModel** attingerà i dati per la UI ma **non è a conoscenza da dove sono stati prelevati**.

Inoltre inietta le librerie **Amplify** (AWS Mobile SDK for Android) per interagire con i servizi AWS.

Nello storage sono memorizzati i **contenuti multimediali** degli utenti

---

# Amazon Web Services

---

Amazon Web Services (AWS) è la piattaforma cloud più completa e utilizzata del mondo.

Offre oltre 150 prodotti, tra cui Simple Storage Service (S3) e Elastic Compute Cloud (EC2), fornendo soluzioni on-demand con caratteristiche di high-availability, ridondanza e sicurezza.

Si tratta di un **Backend-as-a-Service**.



# AWS Amplify

## Amplify

AWS Amplify consiste in un set di strumenti e caratteristiche appositamente progettati per consentire agli sviluppatori di costruire applicazioni full-stack in AWS, con la flessibilità di sfruttare i vari servizi AWS man mano che i casi d'uso si evolvono.

Con Amplify, è possibile:

- configurare un backend per la tua app
- creare un potente modello di dati di backend,
- costruire visivamente un'interfaccia utente con Amplify Studio,
- e molto altro ancora...

## CLI Amplify

L'interfaccia a riga di comando (CLI) di Amplify consente di creare, gestire e rimuovere i servizi AWS direttamente da terminale, ad esempio l'autenticazione utente o l'API REST o GraphQL per le tue applicazioni.

Dipende da Node.js.

## Librerie Amplify

Le librerie Amplify consentono di interagire con i servizi AWS da un'applicazione Web o mobile.

# Amazon Cognito

**Amazon Cognito è un provider di identità utente che permette di aggiungere strumenti di registrazione degli utenti, accesso e controllo degli accessi alle app Web e per dispositivi mobili.**

## **Integrazione semplificata con l'app**

Grazie all'interfaccia utente integrata e alla configurazione per i provider di identità federata, è possibile integrare Amazon Cognito in un'app per ospitare funzioni di accesso, registrazione e controllo degli accessi.

## **Federazione delle identità social e aziendali**

Supporta l'accesso degli utenti tramite l'uso di provider di identità social quali Facebook, Google e Amazon e provider di identità aziendali quali SAML e OpenID Connect.

## **Controllo degli accessi alle risorse AWS**

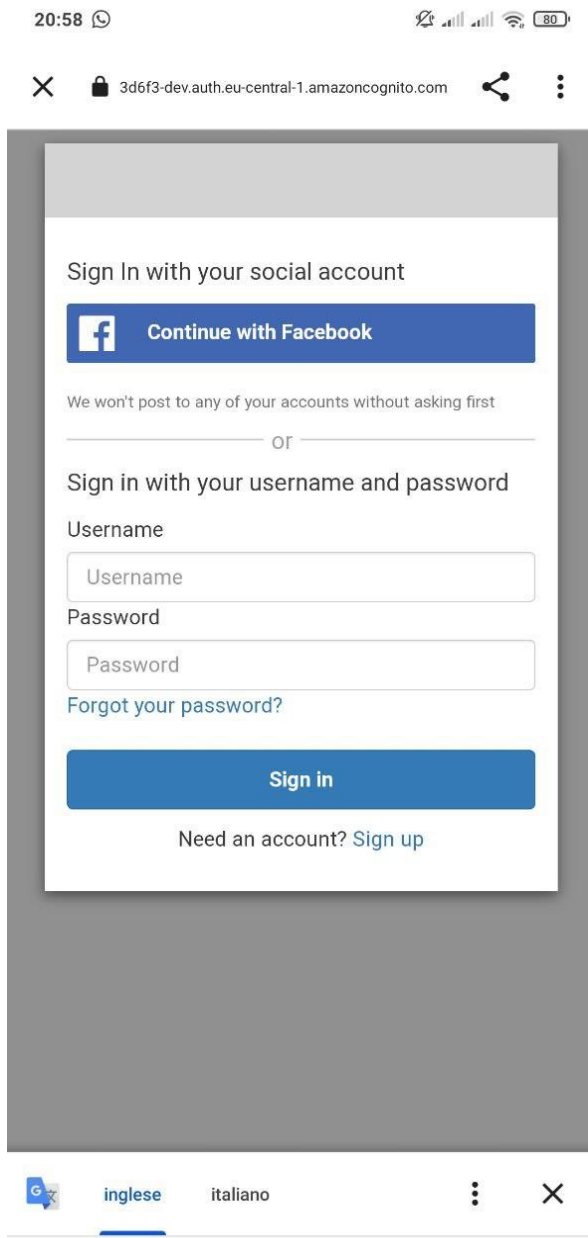
Fornisce soluzioni di controllo degli accessi alle risorse di backend in AWS dalle app.

Si possono definire ruoli e mappare gli utenti a ruoli differenti per consentire all'app di accedere solo alle risorse autorizzate per ciascun utente.

## **Massima sicurezza per app e utenti**

Supporta l'autenticazione a più fattori e la crittografia dei dati in transito e inattivi.





## REGISTRAZIONE/LOGIN TRADIZIONALE

L'utente può accedere a TipPic solo previa registrazione e creazione di un profilo. Lo username con cui l'utente si iscrive alla piattaforma deve essere univoco.



## FACEBOOK

L'utente può effettuare l'iscrizione più rapidamente mediante il suo account di Facebook.



**Interfaccia utente in hosting:** l'applicazione sfrutta le pagine Web Cognito per i flussi dell'interfaccia utente di accesso e registrazione.

L'utente viene reindirizzato a una pagina Web ospitata da Cognito e reindirizzato nuovamente all'applicazione dopo l'accesso.

# Amazon DynamoDB

## Amazon DynamoDB

Amazon DynamoDB è un database NoSQL completamente gestito, serverless, chiave-valore, progettato per eseguire applicazioni ad alte prestazioni su qualsiasi scala.

DynamoDB offre sicurezza integrata, backup continui, repliche multiregione automatizzate, caching interno alla memoria e strumenti di esportazione dei dati.



**amazon**  
DynamoDB

## GraphQL API

GraphQL è un linguaggio di interrogazione lato server per interfacce di programmazione delle applicazioni (API), in grado di fornire ai client unicamente i dati di cui hanno bisogno.

**Nato come alternativa a REST, GraphQL consente agli sviluppatori di ottenere richieste contenenti dati provenienti da più sorgenti, in una singola chiamata API.**

GraphQL è usato per definire uno schema.

Uno **schema GraphQL** è composto da tipi di oggetto che definiscono la tipologia di oggetti che è possibile richiedere e i tipi di campi disponibili al suo interno.

# GraphQL Schema

```
type User
  @model
  {
    id: ID!
    user: String!
    location: Location @hasOne
    description: String
    image: String
  }
```

```
type Post
  @model
  {
    id: ID!
    owner: User! @hasOne
    location: Location! @hasOne
    title: String
    description: String
    image: String!
    date: AWSDateTime!
  }
```

```
type Favorites
  @model
  {
    id: ID!
    user: User! @hasOne
    post: [Post] @hasMany
  }
```

```
type Location
  @model
  {
    id: ID!
    lat: Float!
    lng: Float!
    name: String!
  }
```

```
type Friend
  @model
  {
    id: ID!
    user: String!
    friends: [User]
  }
```

# Amazon S3



## Amazon S3 (Simple Storage Service)

L'archiviazione e le query per i file come le immagini e i video sono un requisito comune per la maggior parte delle applicazioni.

Un'opzione per farlo è effettuare la codifica Base64 del file e inviarlo come stringa da salvare nel database. Ma questo procedimento ha anche degli svantaggi, ad esempio il fatto che il file codificato è più grande del binario originale, le operazioni sono costose a livello di codice e i procedimenti di codifica e decodifica adeguati sono ulteriormente complessi.

Un'altra opzione è avere un servizio di storage specifico creato e ottimizzato per lo storage del file.

I servizi di storage come Amazon S3 esistono per rendere il procedimento più facile, performante ed economico possibile.

# Google Maps SDK for Android

- Con Maps SDK per Android, puoi aggiungere la mappa alla tua app Android utilizzando i dati di Google Maps, le visualizzazioni delle mappe e le risposte ai gesti delle mappe.
- Puoi anche fornire informazioni aggiuntive per le posizioni della mappa e supportare l'interazione dell'utente aggiungendo indicatori, poligoni e sovrapposizioni alla mappa.
- L'SDK supporta entrambi i linguaggi di programmazione Kotlin e Java e fornisce librerie ed estensioni aggiuntive per funzionalità e tecniche di programmazione avanzate.



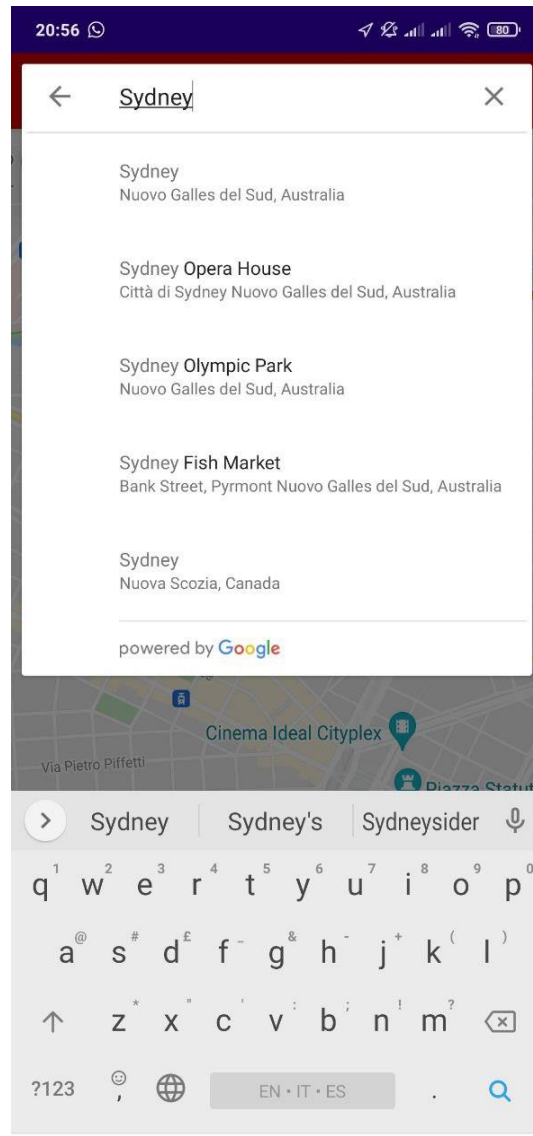


**Utilizzando le Google Maps SDK per Android abbiamo potuto integrare Google Maps nella nostra applicazione.**

**Per fare ciò, abbiamo dovuto creare ed abilitare una API Key in Google Cloud Platform, per permetterne l'utilizzo all'interno dell'applicazione.**

**Il Manifest della nostra applicazione (AndroidManifest.xml) contiene i permessi necessari per accedere alla posizione GPS.**

# Places API



**Places API** è un servizio che restituisce informazioni sui luoghi utilizzando le richieste HTTP. I luoghi sono definiti come posizioni geografiche o punti di interesse importanti.

Abbiamo utilizzato queste API per integrare la barra di ricerca.

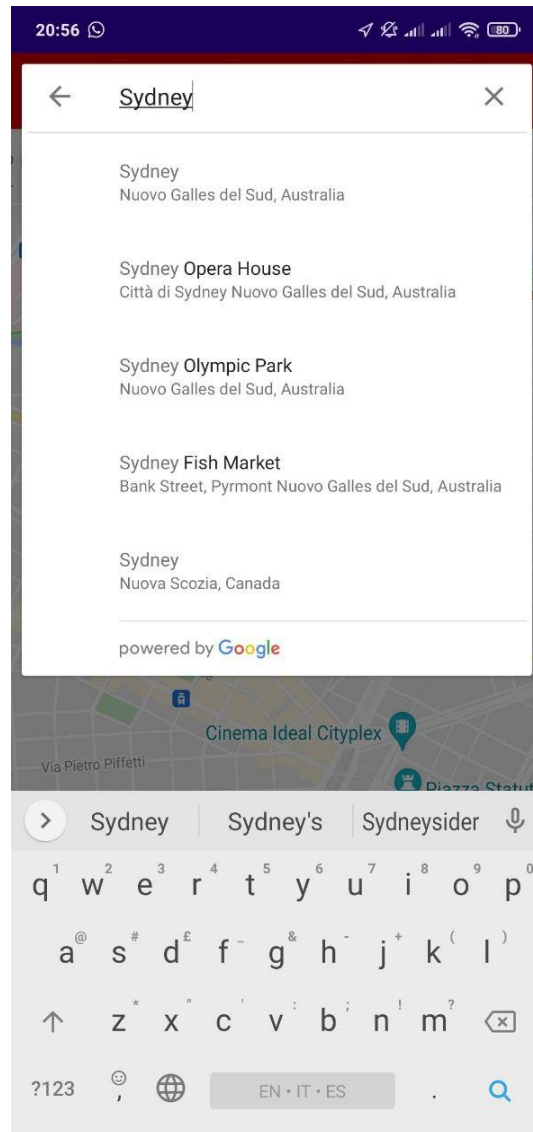
Lo scopo è dare all'utente la possibilità di cercare più facilmente un luogo, digitando il nome di una città o un indirizzo, e poterne ricavare automaticamente la localizzazione sulla mappa di Google e derivarne latitudine e longitudine.

**Place Autocomplete** (vedi slide successiva)

## ViewBinding

Per poter integrare la barra di ricerca abbiamo fatto uso del View Binding, la barra infatti è gestita lato frontend mediante un apposito file XML. Invece il layout del fragment è implementato in Jetpack Compose

# Place Autocomplete



**Place Autocomplete** è un servizio Web che restituisce previsioni di posizione in risposta a una richiesta HTTP. La richiesta specifica una stringa di ricerca testuale e limiti geografici facoltativi.

Il servizio può essere utilizzato per fornire funzionalità di completamento automatico per ricerche geografiche basate su testo, restituendo luoghi come attività commerciali, indirizzi e punti di interesse come tipi di utente.

## ViewBinding

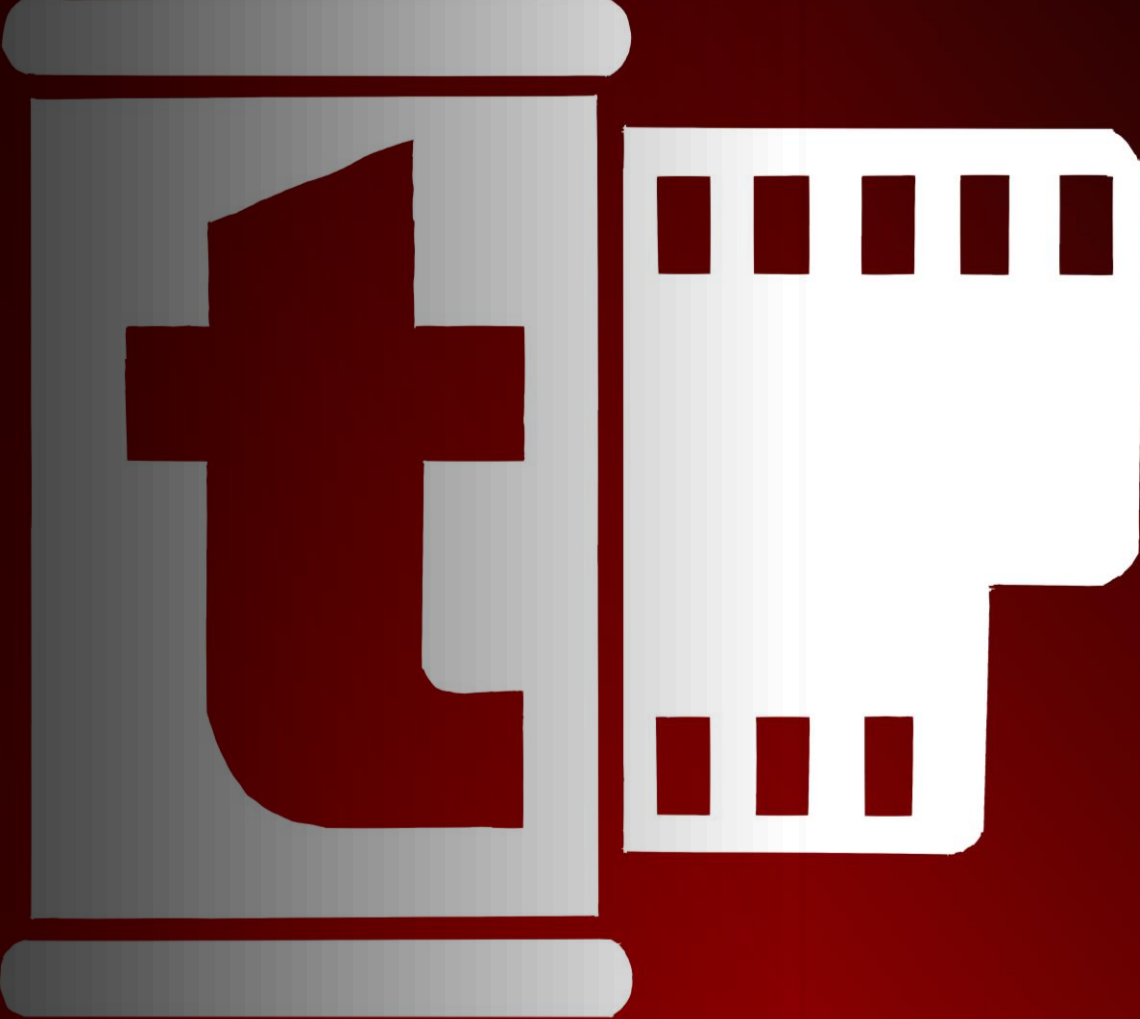
Per poter integrare la barra di ricerca abbiamo fatto uso del View Binding, la barra infatti è gestita lato frontend mediante un apposito file XML. Invece il layout del fragment è implementato in Jetpack Compose





# COMPATIBILITY TEST

---

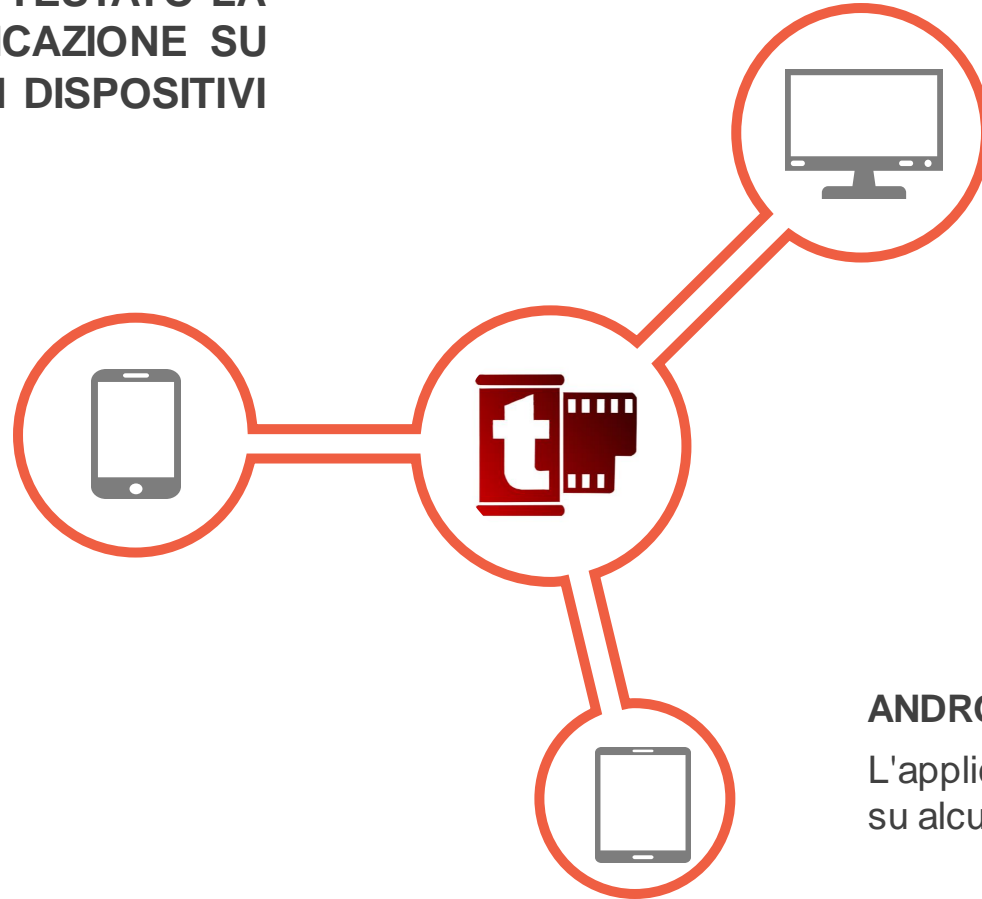


# Trip Pic Compatibility

**ABBIAMO TESTATO LA  
NOSTRA APPLICAZIONE SU  
QUESTI DISPOSITIVI**

## **ANDROID SMARTPHONE**

La nostra applicazione, sia in fase di sviluppo che di test, è stata installata su smartphones fisici



## **VIRTUAL DEVICE**

In fase di sviluppo dell'interfaccia abbiamo prevalentemente testato la nostra applicazione sull'emulatore di Android Studio

## **ANDROID TABLET**

L'applicazione è stata testata su alcuni tablet fisici.

# Technological Constraints

L'app è compatibile con tutti i device Android dalla versione 6.0 Marshmallow (2015) in poi.

Come mostrato in figura, supportando da questa versione in poi, vengono coperti l'84.9% dei dispositivi.

ANDROID PLATFORM VERSION	API LEVEL	CUMULATIVE DISTRIBUTION
4.0 Ice Cream Sandwich	15	
4.1 Jelly Bean	16	99,8%
4.2 Jelly Bean	17	99,2%
4.3 Jelly Bean	18	98,4%
4.4 KitKat	19	98,1%
5.0 Lollipop	21	94,1%
5.1 Lollipop	22	92,3%
6.0 Marshmallow	23	84,9%
7.0 Nougat	24	73,7%
7.1 Nougat	25	66,2%
8.0 Oreo	26	60,8%
8.1 Oreo	27	53,5%
9.0 Pie	28	39,5%
10. Android 10	29	8,2%

## Marshmallow

### Security

- Fingerprint Authentication
- Confirm Credential

### System

- App Linking
- Adoptable Storage Devices

### Multimedia

- 4K Display Mode
- Support for MIDI
- Create digital audio capture and playback objects
- APIs to associate audio and input devices
- List of all audio devices
- Updated video processing APIs
- Flashlight API
- Reprocessing Camera2 API
- Updated ImageWriter objects and ImageReader class

### User Input

- Voice Interactions
- Assist API
- Bluetooth Stylus Support

### User Interface

- Themeable ColorStateLists

### Wireless & Connectivity

- Hotspot 2.0
- Improved Bluetooth Low Energy Scanning

### Android for Work

- Controls for Corporate-Owned, Single-Use devices
- Silent install and uninstall of apps by Device Owner
- Silent enterprise certificate access
- Auto-acceptance of system updates
- Delegated certificate installation
- Data usage tracking
- Runtime permission management
- Work status notification

<https://developer.android.com/about/versions/marshmallow/android-6.0.html>