

BULMAP



1. PRODUCT

PRODUCT VISION

Bulmap's vision is to **redefine local social interactions** by creating a **real-time, geolocated platform** where users can **capture and explore life happening around them**. Our goal is to **bring people closer** by connecting those in the same area and **promoting local events, artisans, and small businesses**.

We aim to **foster community discovery and interaction**, turning the app into a hub for **real-time experiences**. Users can immerse themselves in what's happening nearby, **support local initiatives**, and **build stronger connections** with their surroundings, while **brands and creators engage authentically with their audience**.

Ultimately, we're building a social experience where spontaneity and presence take center stage, offering businesses an organic, highly targeted way to connect with the right audience.

KEY KPIS TO TRACK

- Growth in monthly active users (**MAU**)
- **Time** spent per user on the **map**
- **Interaction rate** on posts (likes, comments)
- **Advertising revenue** per certified user
- Commercial **campaigns** number and **revenue**
- **Conversion rate** from standard users to certified/premium users

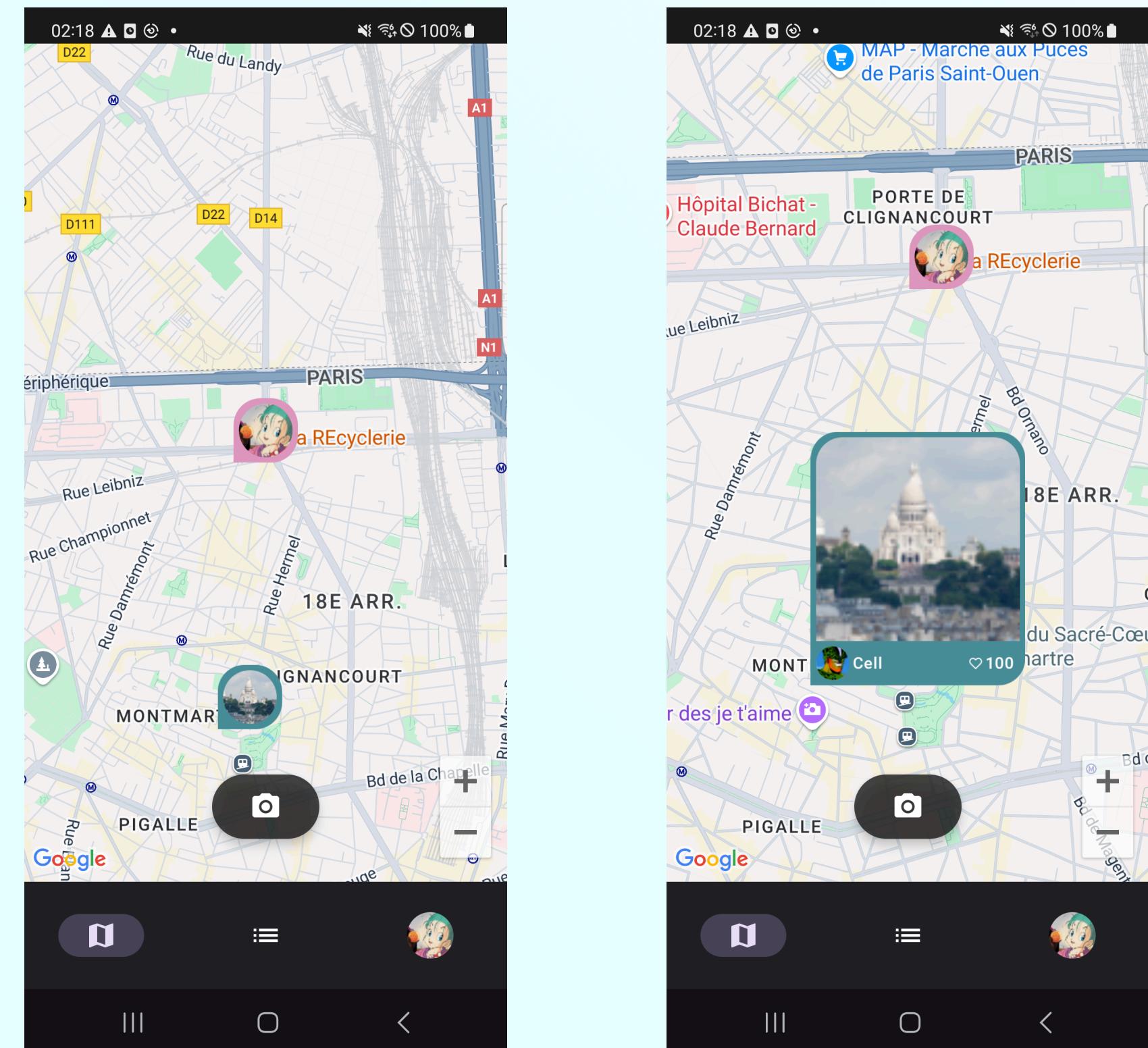
COST STRUCTURE

- App development and maintenance
- Hosting and servers for photo storage and traffic surge management
- Marketing and user acquisition: Digital advertising, influencer collaborations, and launch campaigns
- Content moderation and customer support
- Technical infrastructure: Security, map API services (Google Maps, OpenStreetMap)

2. FEATURES

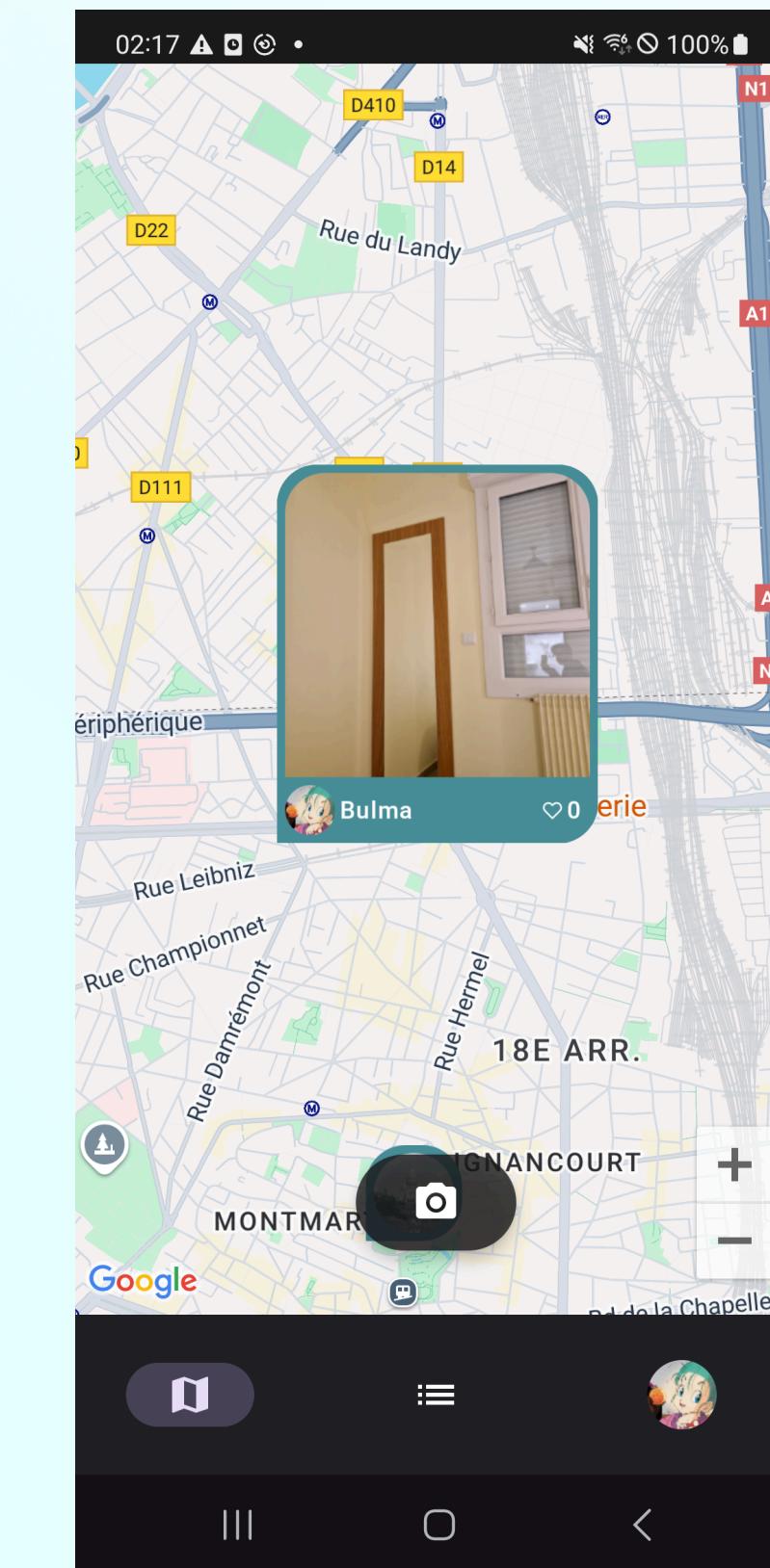
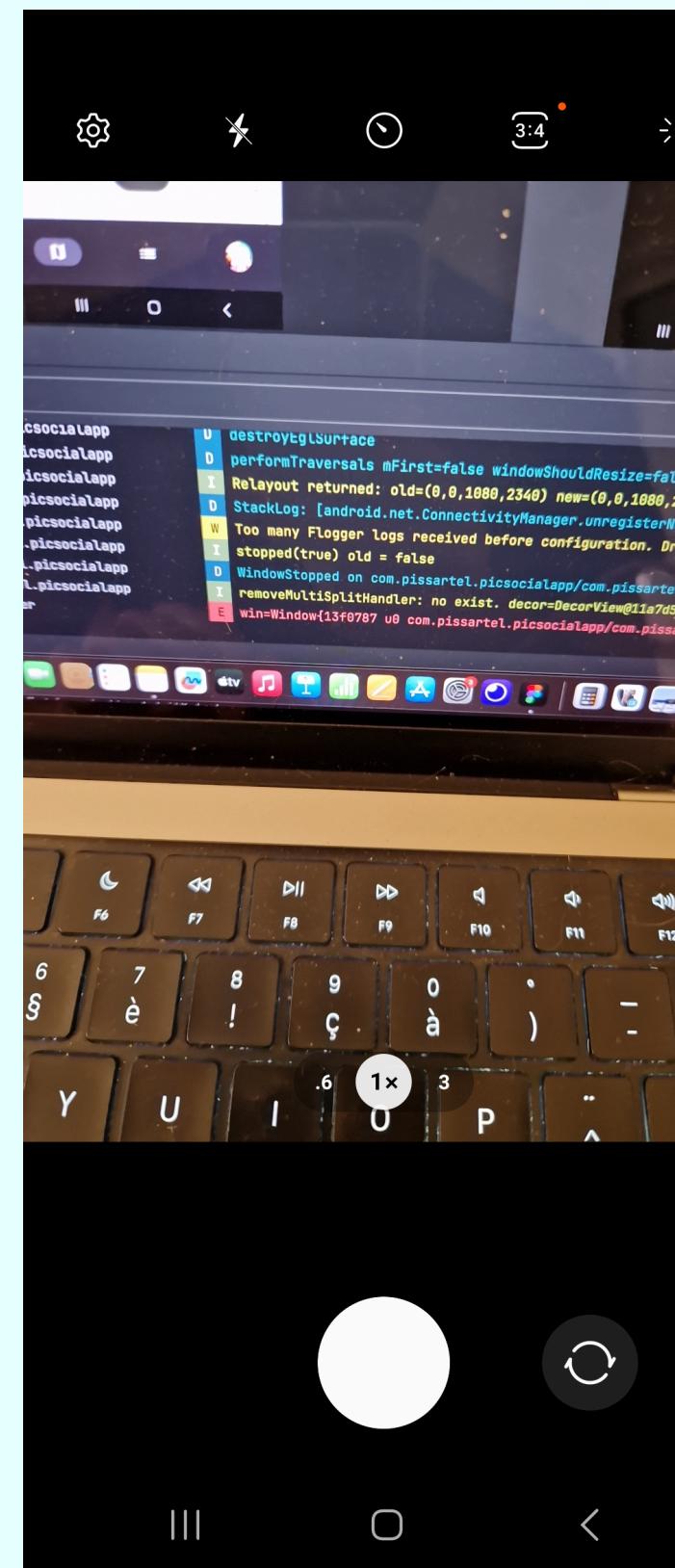
1. INTERACTIVE MAP WITH REAL-TIME PHOTOS

Users can view an **interactive map** that displays photos shared by others within a **5 km radius**, visible for **10 minutes** (standard for regular users).



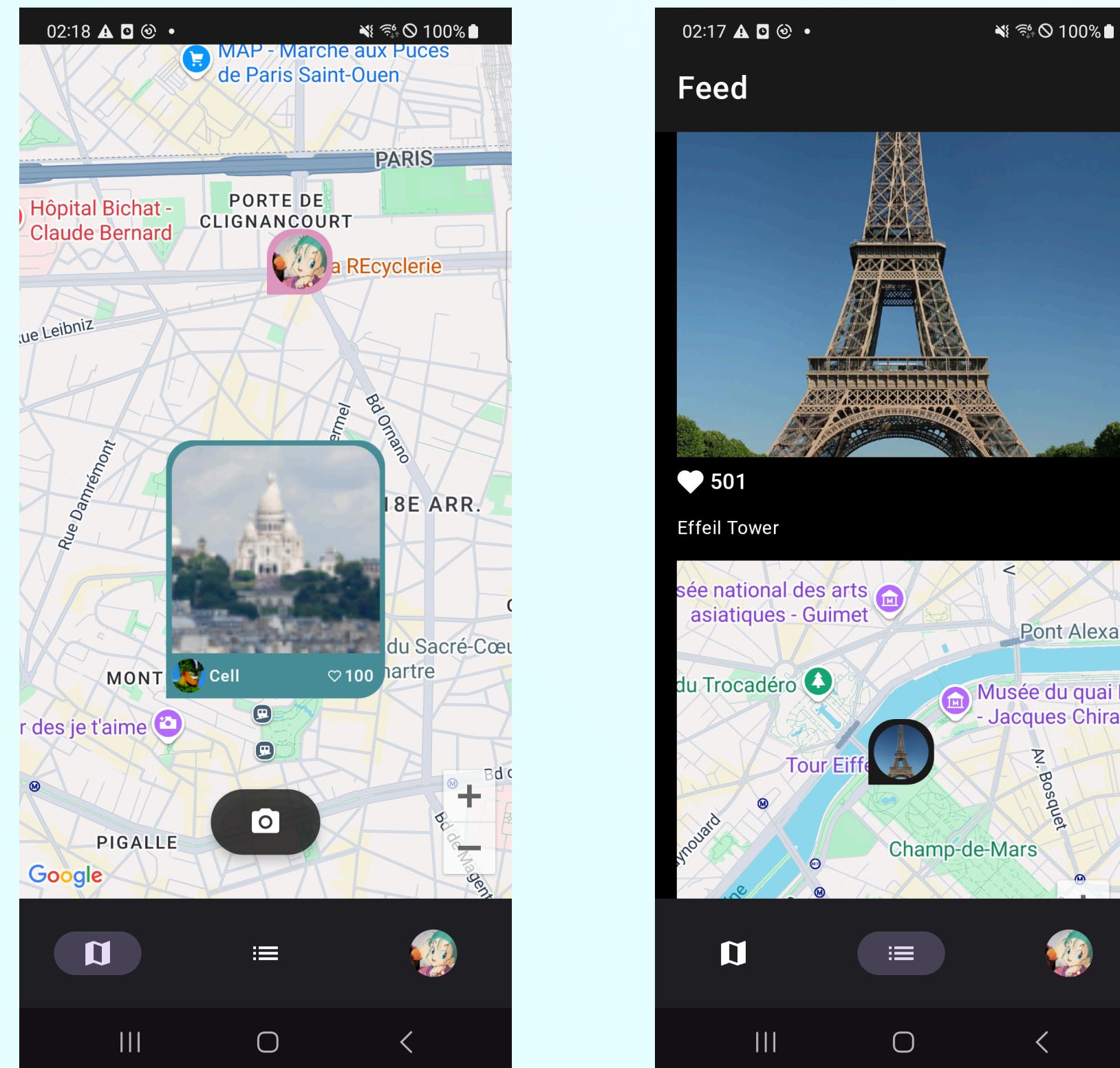
2. PHOTO POSTING ON THE MAP

Users can **take and upload a photo** directly on the map. The photo will be visible for **10 minutes** and in a **5 km radius** around the location where it was posted.



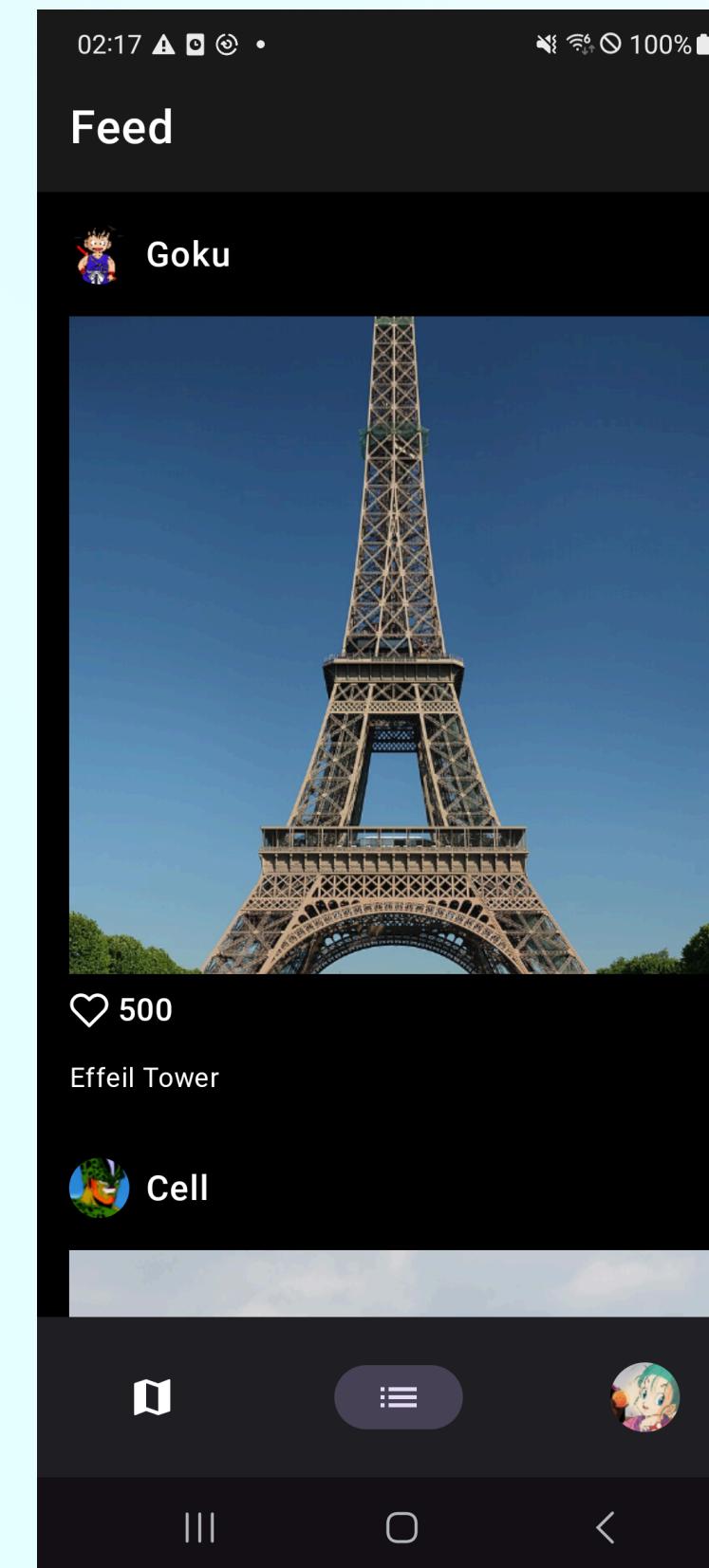
3. LIKE AND COMMENT ON MAP PHOTOS

Users can like photos directly on the map or open them in full-screen mode, where they can **like, comment, or share** the content.



4. FEED WITH LATEST PHOTOS

In addition to the map, there's a **personal feed** where users can see the latest photos shared on the platform, without any radius limitation (duration for visibility in the feed can be determined later).

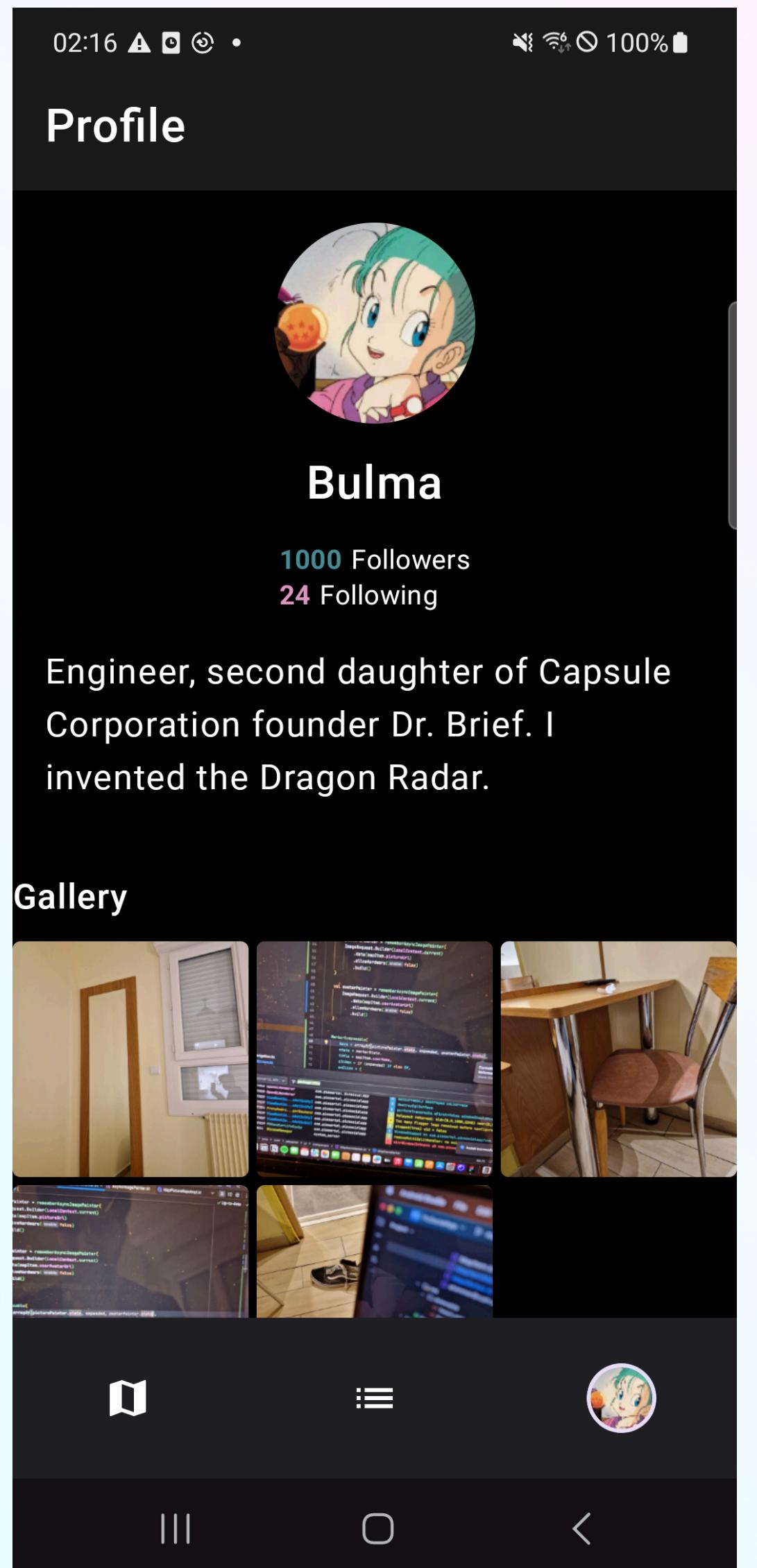


5. USER PROFILES

Each user has a profile displaying:

- Avatar and username
- Number of followers and following
- Photos they've shared, without radius or time limitations

The last tab provides access to the personal profile, where users can view and manage their account details.



6. CERTIFICATION FOR ADVANCED FEATURES

The **certification** feature is designed for **influencers** and **content creators** looking to boost their visibility and monetize their content. Certified users can access to exclusive advanced posting options, such as visibility or extended radius options.

This feature let influencers **grow their community** and collaborate with **advertisers** for targeted, highly visible **campaigns**.

7. PAID ADVERTISING CAMPAIGNS FOR BUSINESSES

Advertisers can run **targeted campaigns** on the platform, creating posts with adjustable time and radius settings depending on their budget.

3. WORK DONE AND TO BE DONE

WORK DONE SO FAR

- Interactive map displaying photos within a 5 km radius
- Photo capture functionality
- Adding the captured photo to the user's profile gallery and the map
- Detailed view of a post
- Likes
- Feed with recent posts
- User profile with their data and photo gallery

WORK TO BE DONE

- Real data integration (everything except photo capture is currently mocked)
- Implementing time-based visibility rules for posts
- Comments
- Accessing another user's profile
- Following other users
- Certification system
- Advertising campaigns for sponsors

4. TECHNICAL CHOICES

TECHNICAL STACK

- **Android Jetpack Compose** : modern, industry-standard toolkit for building native Android UIs with a more intuitive, efficient, and declarative approach.
- **Kotlin Coroutines/flows** : provide structured concurrency and ensure a smooth, asynchronous, and responsive app experience, especially for handling complex tasks like network requests and real-time updates.
- **MVVM** : Google-recommended, industry-standard architecture, ensuring scalability, maintainability, and best practices for modern app development. Addition of **effects** (similar to **MVI**) to display toasts/ notifications or perform navigation after an asynchronous event.
- **Coil, Google Map API, hilt** : industry-standard.
- **Voyager and Tab Navigator** : usually, I use a custom in-house system, but due to time constraints, I used this library
- **Modules** : enable greater scalability.

TECHNICAL STACK

- **Uncle Bob Clean Architecture :**
 - Promoted by Google, clean architecture with a domain layer and use cases may seem overkill at first, but it provides modularity and a clear separation of concerns.
 - I simplified it a bit by using fun interfaces to define use cases, which allows me to have simple use cases (that only call a single method from a repository) directly linked to their implementation through Dependency Injection (DI).
 - At the same time, it still offers the flexibility to have implementation classes for more complex use cases (involving multiple repository calls, additional operations, or testing).