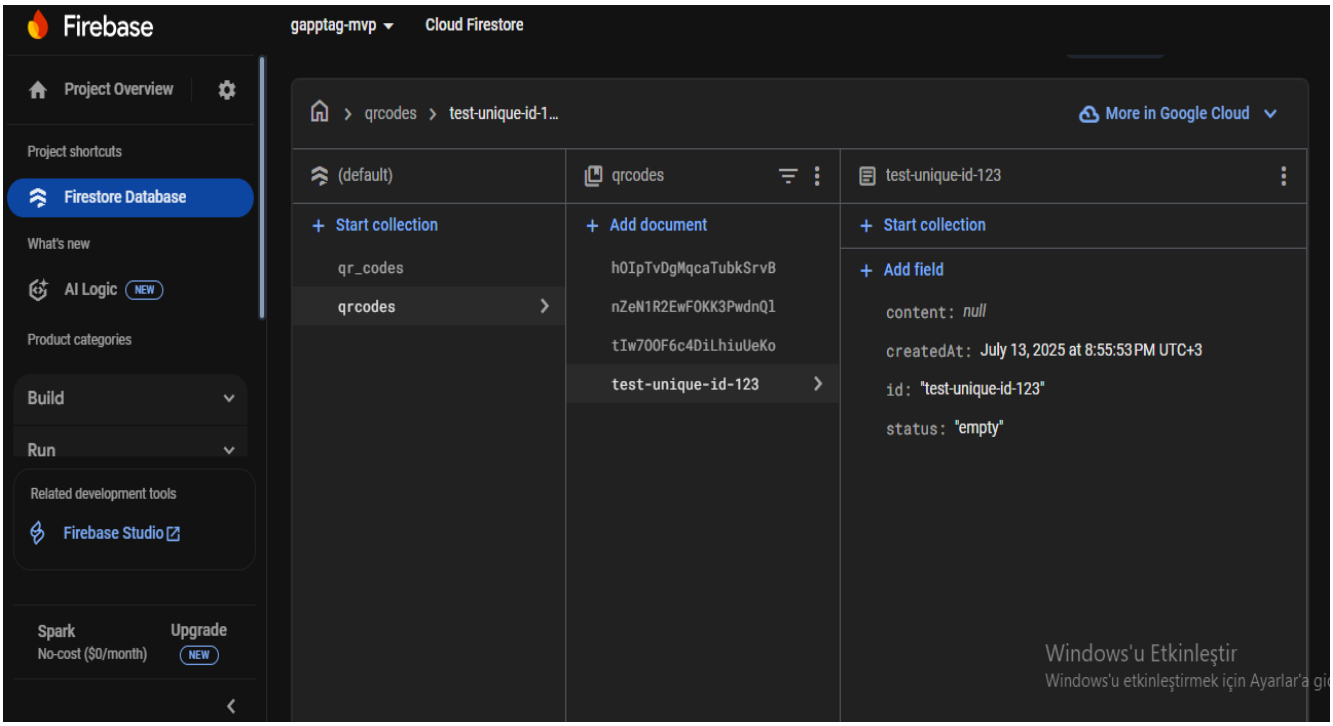
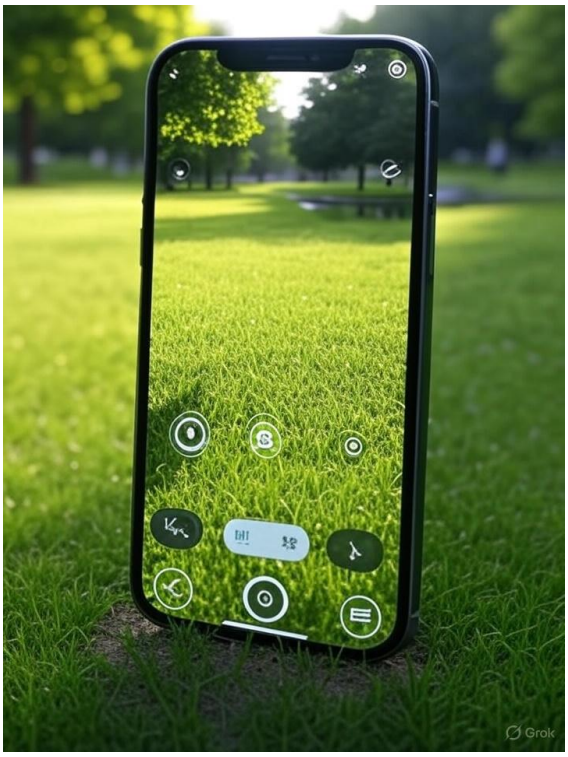
Gapptag Application - MVP Functional Overview



1. In the application's database, there are pre-created, expandable empty fields, each containing a unique ID, as shown in the example above.
2. A QR code has been generated for each empty field. In other words, each QR code corresponds to one of these empty fields.
3. Custom-designed QR visuals specific to the application are also stored in a separate database, such as on a server, ready for use.
4. As a result, a passive QR-ID pool has been created in the background.

The user downloads the application and either creates a profile or logs in via a social media API.

Immediately afterward, a camera screen like the one shown below appears on the mobile device, and the application becomes ready for use.



1. The application launches with a camera that includes an AR lens interface like the one above and has QR code scanning capability.
2. The user creates a video using the available tools. After previewing the content, they decide to share it and apply the desired privacy settings.
3. The user presses a button with a “Save” function, and the following actions take place:

"Save" button functions:

1. Match an empty field from the QR-ID database.
2. Save the video content to this field and move it to a user-specific location in the database.
3. Attach the custom-designed QR visual, which points to this field, onto the AR object.
4. Place the AR object, now containing the QR visual, at the user's current GPS location.

The user then sees the AR object representing their created content on their screen, similar to the example visuals shown below. This object is now the Gapptag associated with the content

the user has left at that location. The user has now shared an experience they had at that location or a message intended for other users at that spot.

Another user arrives at that location and receives a notification on their device, such as: “Would you like to open the app to view Gapptags left at your current location?” This feature can be enabled via NFC.

When the other user opens the app, it launches with a screen similar to the one above. They see the Gapptag left by the previous user in the same way.

The screen scans the QR code on the AR object, and the video content previously added by the other user begins to play. At this point, the user sees a visual like the one shown below.

A woman had recorded herself walking some time ago and left that moment at this location using a Gapptag.

Thanks to Gapptags, other users will now be able to watch and experience what previous people went through in that park before them.

Moreover, if the interaction settings allow, the user can directly contribute to the content. Alternatively, they can engage with it using typical social media actions such as "like," "save," or "share."

While the video content or content stream is active, the user can add to it by creating a new video, personalizing it, applying privacy settings, and pressing the “Save” button, which triggers the following actions:

"Save" button functions:

1. Add this new content to the previously assigned QR-ID field designated for content additions.

In this way, the application fulfills its intended function. However, this version is suitable only as a basic MVP.

After the MVP stage, the application will be further developed with features such as:

1. A **"Magnet Algorithm"** to merge Gapptags with similar content,
2. A **"Lifespan Algorithm"** to prevent Gapptag clutter and create a monetization method,
3. **Customized AR lenses** to provide location-specific user experiences,

ensuring the creation of unique and enhanced user experiences.