The Bike Kollective Final Report

Lennard Gorter, Cedric Eicher, Connor Pollock
Oregon State University CS 467 - Online Capstone Project
November 29th, 2021

I. INTRODUCTION

The Bike Kollective, the next application to get users from A to B with ease, is a responsive mobile application that enables users to find bicycles nearby that they may ride free of charge. Once a user creates an account they may add bikes to the Kollective, check out a bike for a ride, complete their ride, and provide feedback on their experience. The application offers filtering, sorting, and other functionality to help users find exactly what they're looking for. A free alternative to competitor applications such as Uber, Lyft, or motorized scooters, the Bike Kollective offers a much more efficient means of transportation for short distance rides. The bikes and service are completely crowd-sourced, meaning that everyone is responsible for taking care of the existing bikes and preserving them for the next user. The app fosters a sense of community and mutual resources to help everyone have transportation when they need it.

II. DESCRIPTION

A user must create an account, which may include Google Authentication, and may verify their email address for further authentication. This authentication ensures that users are valid and reduces the likelihood of bike abuse or theft. It is not required, but encouraged. Once logged in, the user will not need to sign in on successive rides. The app will remember if the user is signed in, even if the app is closed and reopened.

Once an account is created the user may view bicycles near them on the map view. Each bicycle is noted with a marker that is color-coded based on check-out eligibility. Eligibility is determined using geofencing, meaning the user must be within a distance of 0.1 mi before they are provided with the bicycle's combination and able to start a ride. This inhibits abuse of users trying to check-out and reserve bikes from across town. From there, the user is taken to the ride screen where a timer tracks how long the bike has been checked out. We've implemented a warning at a ride time of eight hours to prevent bicycle hogging or abuse, but we lock users out after twenty-four hours at which point we suspect theft or negligence. At this point, the bike is listed as stolen and removed from the bike kollective. If the user attempts to log back in, they are informed of their ban from the kollective and returned to the login screen.

In addition to the elegant and visually pleasing map view - we offer a list view. The list view provides more data and details for those who like to know a little bit more about their bike. In the list view, users may see bikes in order of those closest to them, while also being able to view a bike's rating, address, and condition. If desired, users may also filter the list by condition, rating, or tags on the bike (such as a Blue Mountain bike). Checking out a bike and starting a ride from the list view functions just like the map view for an intuitive, seamless, and convenient riding experience.

Once the ride is complete, the user is directed to a form that collects information about the bike ride before completion. The form updates the condition of the bike, records the user-provided rating, and automatically updates the bike location. Once this process is complete the bike will become available for other users to see because it's no longer checked out. The ratings help us provide users a way to check-out the bikes that perform best and leave the

junkers alone. The application is a true community experience from start to finish.

Our community relies on the benevolence of its members. We are able to scale and expand to new locations because our users understand the importance of sharing. If you are able to ride bicycles that used to belong to others, why not contribute your bicycle in a similar fashion? This is how a network of available bicycles is created for convenient use. Adding your bicycle to the Kollective is as easy as taking a picture using your mobile device's camera (or selecting one from your gallery), providing a bit of information in an intuitive form, and locking up your bicycle. We'll take it from there!

III. PROJECT PLAN AND DEVIATIONS

At the outset, we constructed a comprehensive Project Plan to get our development efforts started. We adopted an agile approach that allowed us flexibility and ensured a functioning product at all times. This means that there might have been some deviation from the Project Plan, which we used as a guide and attempted to adhere to, but the main direction remained consistent. We also implemented a weekly scrum meeting to ensure all deviations were captured and calculated with the intent to meet requirements and keep each other in the loop.

Deviations on the bicycle and bicycle maintenance were design decisions intended to protect the integrity of the bicycles. For example, in our project plan we were prepared to allow users/riders to add tags to a bicycle for filtering. In the end, we determined that only the user contributing the bicycle should have that ability. Pre-set tags allowed us to prevent faulty information or inappropriate user behavior. Additionally, we focused on tags that were more bike-focused, than ride-focused. This would allow the tags to be more static at the time of assignment (such as Mountain bike, or Red bike) than introducing subjectivity to the tags. Users still had the ability to note ride opinions via the bike's end condition and a bike rating at the termination of their ride.

In the main User Interface we did deviate from our plan of having a search bar on the map view by only having filters/tags on the list view. The map view seemed much more intuitive without search functionality and a more location-centric approach. Likewise, we talked with our sponsor, Professor William Pfiel, about the search bar to make sure we were not misinterpreting the requirements. He agreed a search bar on the map screen was not necessary and redundant.

Another deviation is the lack of user profile editing. We decided the user accounts didn't have much of a social aspect, so we kept that to a bare minimum. We want users to be up and ready to go as quickly as possible with minimal fuss. The social aspect could be fleshed out further in later versions, but per our existing requirements, social profiles are not value-added and are thus lower priority.

Some other (more minor deviations) are that once a bike is reported missing it is removed from the list and does not show up as opposed to a change in the color code. We opted for this decision based on long-term scalability and impact to the Bike Kollective community. Inevitably, bikes will be stolen or go missing - having them remain on the map is not informative to users and generates visual clutter.

Our last minor deviation was that we do not display a user's previous rides as we felt it didn't add much value and would only hamper a streamlined minimalistic user experience.

Most if not all of our deviations resulted from discussions of value versus requirements, in which requirements had priority.

IV. LIBRARIES, SDKs, APIs, ETC.

The big star of our application is the Flutter mobile development SDK. Flutter is a cross-platform SDK that provides a framework for natively compiled mobile applications on Android or iOS. We chose Flutter because of its ease of use, "deploy anywhere" ability, and familiarity within the team.

For the mapping and location services, we chose to use OpenStreetMap. As an open source solution it fits in perfectly with our mission of a free application powered by its users. After much research into map capabilities, polling limits, and pricing, we decided OpenStreetMap was the better long term choice over the traditional Google Maps.

Lastly, on the backend, our application is powered by Google's database application, Firebase. We used their storage (for ease of image upload and use) and database to power our application with specific tables

for bikes, users, and rides. Firebase is non-relational and best fits our requirements for quick, seamless queries regarding multiple data types and images.

We also used the services of Gmail and EmailJS to implement email verification for users. Gmail hosts our mail server (thebikekollective@gmail.com) and EmailJS automatically sends templated email verification messages when prompted by the app. EmailJS and Gmail were both chosen for their ease of use and high polling rates before payment plans needed to be considered.

For specific functionality we used a plethora of packages from https://pub.dev, the official package repository for Dart and Flutter apps. The following packages were used, with their version noted in parenthesis:

- shared_preferences (2.0.0)
- firebase_core (1.8.0)
- firebase_auth (3.1.4)
- google_sign_in (5.1.0)
- cloud_firestore (2.5.4)
- firebase_storage (10.0.6)
- image_picker (6.0.0)
- latlon (0.6.1)
- location (4.3.0)
- geocoding (2.0.1)
- flutter_map (0.14.0)
- flutter_map_marker_popup (2.0.1)
- flutter_launcher_icons (0.9.2)
- http (0.13.4)

These packages were primarily focused on firebase connectivity (firebase_core, firebase_auth, cloud_firestore, firebase_storeage), Google authentication sign-on (google_sign_in), mapping tiles/visualization (flutter_map, flutter_map_marker_popup), location services and geocoding (latlon, location, geocoding), and local phone access (image_picker, http, flutter_launcher_icons, shared_preferences).

V. CONCLUSION

The Bike Kollective is a powerful application that aims to provide free, easy short-transport rides to those in dense urban areas. Utilizing industry standard APIs, a robust cloud database, flutter and dart packages, and thoughtful implementation the application offers an intuitive, quick, easy, and reliable service to users attempting to travel (ideally short distances). By offering both map view and list view the application provides users with options whether they digest data better visually or with a more detailed text view. Adding a bike makes the community stronger and provides more options for users worldwide.

Future directions for the app would probably focus on the social side of the kollective. This could involve showing user's their previous rides, ride statistics, and maybe even a means to interact with other users. This would all bolster the community aspect of the Bike Kollective and encourage more users to join and take part. Additional directions could involve more focus on the bikes, such as allowing users to favorite a bike, note more unique or personal ride details, and developing means to encourage bike additions or collective maintenance.

In its current state, the Bike Kollective app provides robust support for crowd-sourced bike travel from anywhere, anytime a bike is present. The user interface is seamless and simple to encourage all to join today and have reliable short-distance rides. Travel conveniently with The Bike Kollective today!