COBI.Bike iOS Coding Challenge

The goal is to develop a simple community-driven station-less bike sharing system. Every user can contribute his bike to the system and the bikes can be parked at any arbitrary place. The respective iOS app app gives users an overview where bikes are available, allows them to rent them and to return them as well as adding their own bikes to the system. Each bike is to be equipped with a simple padlock to restrict access to the users of the app. A simple backend synchronises the rental status between apps.

User Stories

1. Add bikes to the system

The user must be able to add his own bike to the system. As soon as the bike profile is saved, the bike is available for rent.

- Have a form to enter the following information
    - Name: validate that the name is not empty
    - Frame color: offer a color chooser with a small predefined set of colors
    - Pin: only accept a pin with exactly 4 digits (0000 to 9999)
    - Location: the current device location is used as the initial location
- Add the bike to the backend via the `POST /bikes` endpoint

2. Find available bikes on a map

All bikes around the users location that are not in use should be shown on a map. When selected, the user should get the option to rent it.

- Receive bikes around the current location via the `GET /bikes` endpoint (the variant with location parameters)
- Show a map with bikes available nearby (i.e. the ones that are not rented already)
- Tapping on a bike shows the bike profile created in user story 1. and and allows to rent a bike (3.)

3. Rent a bike

When the user decides to ride a bike, he can rent it.

- Set the rental status on the backend via `PUT /bikes/:id/rented`
- Update the state of the app accordingly

4. Return a bike

When the user is done riding, he can return the bike. In this case, the new status is sent to the server along with the current location.

- Update the location of the currently rented bike profile to the current device location: `PUT /bikes/:id`
- Return the bike: `DELETE /bikes/:id/rented`
- Update the state of the app accordingly

Requirements

- Use Swift
- Use the attached node.js server as the backend for your app. Please read the README.md carefully — it contains additional important information (like a documentation of the API endpoints.).
- The app state doesn't necessarily needs to be persisted between app launches
- Make sure the app has a decent user experience
- Use libraries and architectures/patterns at your discretion
- Write tests to verify the apps behavior
- Handle server errors gracefully (i.e. a bike that was available when showing the map might not be available when the user tries to rent it)

Upload your final code to Github and make it available to COBI. Make sure to also submit the node.js backend, especially when it is modified (doesn't need to be part of the repository though).