

Notifications	2
Capabilities	2
Positives	2
Negatives	2
App in Background	2
Gestures	2
Capabilities	2
Positives	3
Negatives	3
Sensors	3
Capabilities	3
Positives	3
Negatives	3
Haptics	3

Notifications

The likely uses of this feature for mobile devices would be to provide a mode of interaction with the device even while it is locked. Notifications, if allowed to be viewed on the lock screen, can be interacted with; so long as this app is given priority in some way, they will remain at the top for easy access. This may not be as useful as intended, after some research was completed.

Capabilities

Notifications can be wielded on both iOS and Android when done through Expo, although neither platform can do push notifications on the emulator. Notifications can be created, updated, and imperatively deleted on both iOS and Android; this capability can allow the notification used to stay on top. However, the symmetry between iOS and Android through Expo comes with some downsides: namely, the notification system is limited in structure and flexibility.

Positives

Notifications for web apps are baked into Expo's React-based features. As such, getting it up and running for testing is easy and fast.

Negatives

To do any sound customization or activity with notifications while the app is in the background (which may be needed if Komoot is still the primary navigation and routing app), we need to implement EAS builds (more complicated and time-consuming to test compared to Expo Go; in addition, this will need XCode for iOS development). While complicated, EAS build will likely be needed anyways for a release build; taking the time to figure out how to get this working may be worthwhile.

App in Background

Research on the viability of running web app features in the background generally refer us to using the OS-specific systems that reference background tasks, or APIs in those OSs that keep the app active instead of being cleaned out of memory. While I did not go very far in detail on these topics, it seems that for running the app in the background, we will need to step outside Expo into system-specific code.

Related future task: EAS build setup -> research, risk evaluation, implementation.

Gestures

If the app is running but the user is doing something else (biking), what methods should be used to interact with it? If you have to go through a menu while steering, that's not intuitive. A swipe may be best, but the gesture has to provide enough information to perform an action.

Capabilities

With gestures, you can make simple or complicated interactions with the mobile device's screen call other effects. Sometimes, these gestures are far easier to do on the screen itself instead of trying to read something on the screen to touch a specific button, or act within a specific area on the screen.

Positives

This API is fully integrated into React Native (deterministic and local) and even maintained in part by developers at Expo. As such, it should be fully functional (and bugs should at least be identifiable if we have issues).

Negatives

Usability and effectiveness become limited in cases where gestures are already taken by different frames/mounted views. For example, a map view takes up many of the existing movement gestures; if we still want a gesture control for more functions while keeping the map on-screen, the complexity of any gestures must increase to not overlap with map control movements.

As such, adding this feature would initially require a data page with information with a list of gestures that are not yet used, and which are programmed as special gestures. The data page should separate different primary frames that are mounted; usually, a mounted frame is needed for the app to work.

These generally do not work unless the app is open. As such, the device screen should not sleep on its own (see `KeepAwake`).

Sensors

Mobile devices are capable of more than just gathering location information. The sensors that are most likely to be useful are the Accelerometer, Barometer, and DeviceMotion.

Capabilities

The Accelerometer can measure changes in 3D acceleration, including vibrations. Listeners can be programmed to respond to changes in the movement state of the device. The Accelerator is not available on the iOS Simulator.

The Barometer measures air pressure in kPA and can be used as a relative measure of altitude changes while weather conditions are consistent. It is not available for the iOS Simulator.

Device motion allows for changes in the orientation of the device's primary axes to be measured and acted upon. As such, it can provide additional information on the user's steering and direction of motion depending on the scenario (and depending on the device's position when the app is in use on a moving vehicle)

Positives

We can use this information to increase the accuracy of our location data, especially in cases where accurate location services are limited (some routes may have tunnels).

Negatives

We would have to become much more knowledgeable on the technical aspects of using sensor data to build motion and location profiles. In addition, lots of factors can impact the data from these sensors; as such, they should not be absolutely relied on.

Haptics

There used to be a section on Haptics alongside sensors, but the topic fits better in a User Story for customizations and polish (under responsiveness).