

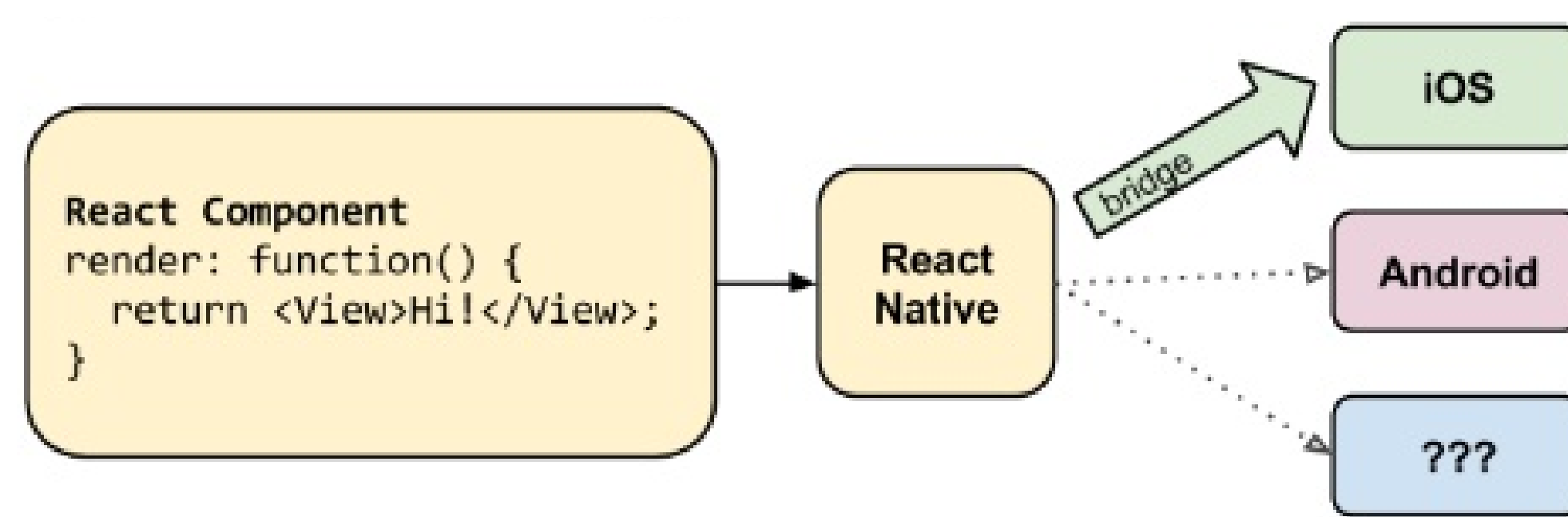
Kevin Benkart, Ares Canetti, Nicholas Paolino and Emma Theberge
CSC/FAA 4815 Directed Study, Special Project Application Design and Development
Christopher Stuetzle & Dan Vlahos

Abstract

We developed an iOS application for sharing geotagged photography and photogenic locations among both amateur and professional photographers as a joint independent study between the Computer Science and Visual and Performing Arts departments at Merrimack College. Rove allows users to place pins on a map for places they've visited. They can also add tags to them to help others find it, and place photos of the location; to help other users find the perfect places for their photography needs. We developed the app in ReactNative using Expo and used Firebase to store generated information. Locations are stored in Firebase as a coordinate pair, along with a Super pin that describes its location type (coastal, rural, city etc.), and sub pin types that describe notable characteristics of the site (graffiti, abandoned, architecture etc.). While building the app, we ran into technological challenges, such as adding assets into the expo build and creating a storage system for organizing data in Firebase such that it could be easily written to and read by our application.

Expo

We used Expo and Snack Expo as our platform and framework to build our app in React Native. React Native allows us to build our project in one language and run it natively on a number of platforms including android, iOS, and web applications. Snack Expo is a tool built to run React Native applications in a browser as well as simulate the app running on android and iOS device. The code code is dynamically bundled and compiled to run it in the Expo Client making the application much quicker to develop.

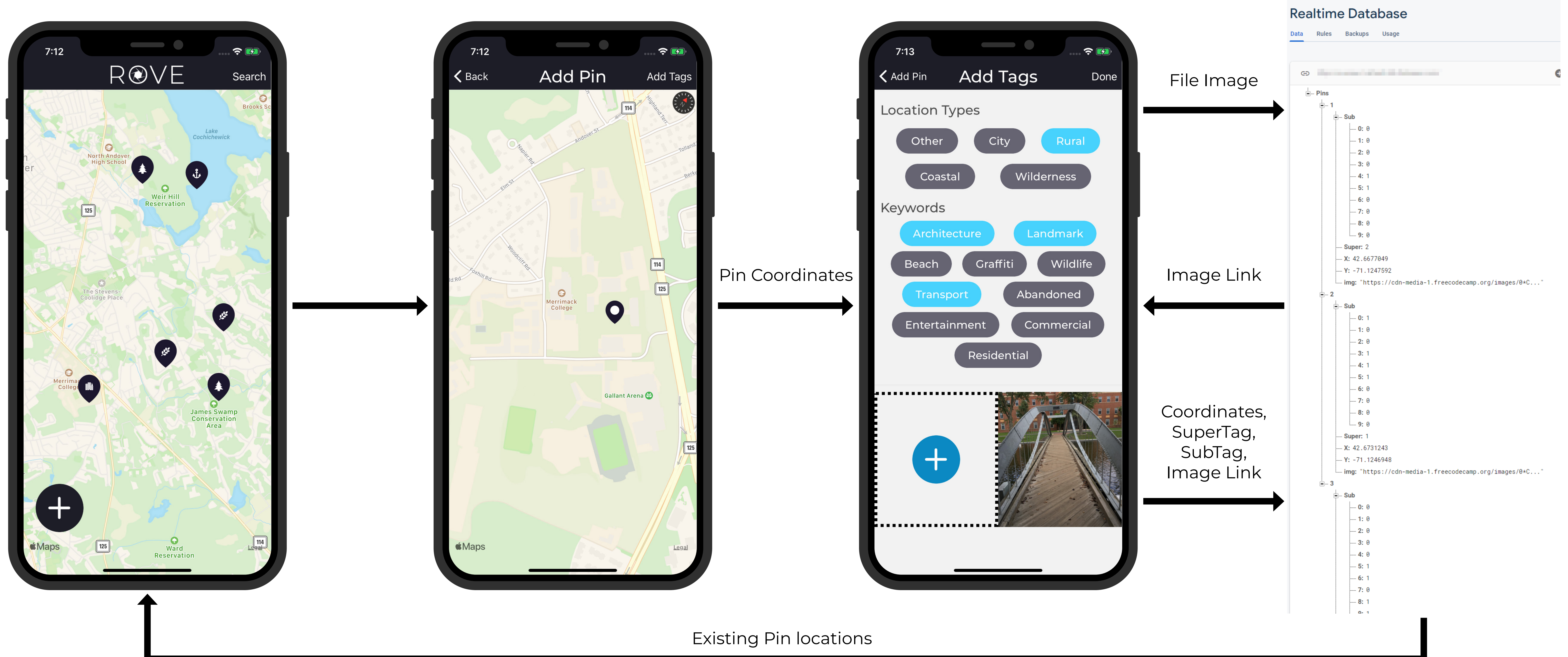


The flow of React Native code to native code

Map and Pins

Rove uses the `react-native-maps` library created by Airbnb that allows React Native Map components to use the native map frameworks iOS (MapKit) + Android. The MapView component allows users to zoom in and out, as well as rotate/pinch the map. The app populates the map on run with all the locations that have been marked using the Marker component, which we refer to as "Pins". These pins are used to label locations but also offer a glance into the type of location it is. The two types of pins a location can have is a super pin type (City, Rural, Coastal, Wilderness and Other) which a location can only have one of, and a sub pin type which a location can have many of. Tapping a pin will take you to that pin's image gallery where you can see photos of that location placed by users and add your own photos taken at the location. Users can also create new pins of locations they feel are great for photo shoots by simply using the add pin button, and then pressing and holding to add a new pin. After placing a pin on the map they can then select one super pin type location type, and as many sub pin types, along with adding a photo taken from that location.

Pin Data Flow



Firebase & Storage Design

After a user generates a new pin and/or adds a photo, that data is sent to storage to be accessed by other instances of the app. We do this by storing the information in Firebase. Firebase is an online storage platform that enables the storage and retrieval of information to and from systems with a focus of web and mobile applications. Once data is generated it is stored and organized. Within firebase there are a variety of storage methods and functionality. For Rove the two main storage builds used are the aptly named "Storage" which is used to store file data such as images, the other is "Realtime Database" which is where we store each pins information in variable form is stored and retrieved from. Images sent to firebase are sent to Storage and then turned into links which point to that image in our file system, that link is then added to a pins information in the Realtime Database as a string variable. Pins data is organized into parts, firstly the coordinates of latitude and longitude, next the pins type and tags are converted into numerical values representing there information, and lastly a list of image links attached to that pin as well as a counter to know how many images are expected to load when users look at that pins screen.

Future Work

- Android compatability.
- Implement and account system to be able attach people to favorited pins and galleries.
- Add integration with other map tools such as "google street view" or "GPS routes".
- Add a notes/chat feature for users to note information about the areas they liked such as best time of day to visit.
- Build a wider selection of Super and Sub tags with the help of user driven data.
- Ability to upload multiple photos when adding a new pin.

Sources & Libraries

React Native Flow Image: <https://atomate.net/2018/11/code-sharing-between-react-web-and-react-native/>
Rove logo and identity created by: Kevin Benkart and Nicholas Paolino
Images from the gallery Screen: <https://merrimack-internal-smugmug.com/Communications-and-Marketing-/Special-Projects/Tool-Box/>
School logo: https://www.merrimack.edu/about/offices_services/communications/branding-toolkit/
expo: <https://docs.expo.io/>
expo-app-loading: <https://docs.expo.io/versions/latest/sdk/app-loading/>
expo-image-picker: <https://docs.expo.io/versions/latest/sdk/imagepicker/>
firebase: <https://firebase.google.com/>
react: <https://reactjs.org/>
react-native: <https://reactnative.dev/>
react-native-maps: <https://github.com/react-native-maps/react-native-maps>
@react-navigation/native: <https://www.npmjs.com/package/@react-navigation/native>
react-native-responsive-dimensions: <https://www.npmjs.com/package/react-native-responsive-dimensions>
@react-navigation/stack: <https://reactnavigation.org/docs/stack-navigator>



Pin Images

Tap a pin on the map on the Home screen to view photos uploaded by other users from that location and add your own.