The Final Project

Due Friday, December 7th, at 11:59 PM

Overview

This quarter we have worked together to explore a lot of different React Native (RN) topics. We were able to go from knowing very little about RN, to understanding a lot of its core concepts and intricacies. We put this into practice with assignments. Now we will get to put this into practice with a slightly bigger project.

In short, when you're working on a new project in the real world, it's unlikely that anyone will hand you a starter file, and sometimes your tasks will be ambiguous. This final project is aimed at giving you an opportunity to become comfortable starting a project from scratch while using the guidance from CS47 and the internet (via snippets online) as a resource.

We hope that you take advantage of this project to explore creating an app that you've always wanted to make. Be flexible. If you don't have an idea yet, don't worry — we will also be suggesting some ideas for projects that you can put together. Overall, we hope you enjoy building your own project in a real-life scenario :)

Design Details

The design for your project is up to you. You have a lot of leeway to try different styles and components. At a high-level, however, we will grade you on design.

You don't have to sketch design prototypes, or come up with the most aesthetically pleasing app (though that would be appreciated), but **your app's design should not prevent us from using it.**

Briefly, here are a few guidelines to consider:

1. Components should have flexible sizing. They should show up clearly regardless of the screen size (tablet or mobile phone).
    a. Even though components have flexible sizing, they should still be aligned and properly sized when relevant.
    b. For Text components, font sizes should be readable cross-platform.
2. Your app should look good on both Android and iOS.
3. Use colors wisely. Consider creating a color palette and using it consistently.

a. In general, you will want to create a contrast between backgrounds and the components on top of them (this includes Text components).
b. Read Color from *Universal Principles of Design* for an overview of how colors are typically used. Consider standardizing analogous, triadic, complementary, or quadratic color palettes.
c. Colors can be useful to cue change of state as we did in the NYT app.

Implementation Details

Show off your ability to build cross-platform mobile apps using React-Native! To achieve this, we ask that you meet the following requirements with your implementation:

1. [Tier 1 requirements] Create an app that:
    a. Uses navigation and has multiple screens.
    b. Has appropriate UI styling and use of flexboxes.
    c. Implements some kind of networking solution (download + display content from the internet).
        i. AND/OR a local data storage solution (store + display content from memory). See API list below.
    d. Implements more than four components / APIs and satisfies basic organization requirements. See details below.
2. [Tier 2 requirements] Implement one or more of the following:
    a. State management using Redux.
    b. Firebase.
    c. A Native Module for any purpose that is NOT supported by RN currently (please double check with us if you decide to go through this route).
    d. Local storage via Realm (requires detaching from Expo).
    e. Any advanced 3rd party library. You can use any 3rd party library that you want in your final project. However, in order to get credit for this point, you must use an *advanced* library that forms a core feature of your application. If you go this route, we recommend that you check with us to confirm that your implementation will fulfil this requirement. See details below for examples of acceptable libraries.

Contact us ASAP if you feel that your project is significant but does not exactly meet all points above.

API Use / List

Note that while fulfilling the requirements above, you will end up implementing some public APIs by default (for instance, we count the proper use of a StackNavigator as an implementation of the react navigation library). In addition to that, we hope that this list of APIs serves as a source of inspiration.

Storage Solutions:
AsyncStorage
Realm

Miscellaneous:
KeyboardAvoidingView
Modal
Switch
Alert

Third-party APIs:
Api-sauce: https://github.com/infinitered/apisauce

Advanced APIs:
WebRTC: https://github.com/oney/react-native-webrtc
Maps: https://github.com/react-community/react-native-maps (+ this is built into Expo)

Keep checking periodically. We will be adding more APIs here that you can use for inspiration.

Code Organization

This requirement is very minimal. You can choose to implement your code however you want. At a high level, we will be awarding points to solutions that are well organized and structured. We want to encourage that you consider the following:

- All of your app's code should NOT be in a single `App.js` file.
- If your project is really big, divide the code in your workspace into meaningful components. For instance, if a certain code snippet is responsible for rendering a search box, create a `SearchBox.js` component to hold all of its functionality.
- Consider using a Metrics and Colors file. The Metrics file is a good way to keep all proportions in your app consistent. The Colors file is a clean way to define a color scheme that you will use consistently.

Final Presentation

The app by itself tells half the story or less, the rest of the story is told by the creator. Prepare a **6-9 mins** presentation for week 10 Tuesday. **Your presentation should include a live demo (or a video of the entire app running) AND a discussion of the work that went into it**. Here are few questions to help prepare for the presentation:

1. What is the problem that you are attempting to solve?
2. How does your app solve the problem?
3. How did you employ an advanced feature in the app?
4. What was challenging about building the app? How did you overcome it?
5. What are things you learnt by building your app?
6. What, if any, are your future plans for the app?

Live demos are highly encouraged.

Grading

Plan ahead for this project. Factor in how long assignment 3 took you and multiply that by 2 or 3. Here is our distribution of grades for this project:

Design	5 points
Appropriate UI styling and use of flexboxes.	10 points
Using Navigation	5 points
Networking or AsyncStorage	5 points
Advanced Feature	10 points
Presentation	5 points
Total	**40 points**

If using code from StackOverflow or other online sources, please add a comment above the relevant code blocks defining where the code is from.

Deliverables

First, **delete the `node_modules` folder** from your project folder, then **zip the project + the presentation**, and **submit on Canvas**.