

Test Task React Native

Kudos

Project Overview

There is a lot of ways to say thanks or share your appreciations to co-members for their contributions and help. One of those practice is Kudo Cards. You could learn more about them on <https://management30.com/product/kudo-cards/>

In short, our app should allow users to send kudos to one another.

Components of the app

You can find design for all main screens here:

<https://marvelapp.com/2fa8gi5/screen/35999510/handoff>

Given mockups show you the basic styling challenges we want to see you solve. But feel free to modify it. The main thing it to make your app look consistent and pleasing to the eye. But also to show that you can not only use UI frameworks but tweak styles on your own quite a bit.

External screens

- **Welcome:** links to Sign up and Log in
- **Sign up:** first name, last name, email, password. Should have validations
- **Log in:** email, password. Should have validations

Internal screens

- **Explore:** List of all kudos
 - Feature: like by tapping on a heart. Increments likes counter.
 - [Allow to like kudo only one time and show heart as red if already liked.](#)
 - [Update list on pull down](#)
 - [Show loader when list is loaded for the first time](#)
 - [Infinite scroll \(load new kudos by pulling up\)](#)
 - [Tap two times on kudo to like \(as in Instagram\)](#)
- **Kudo (show):** photo, message, sender, receiver, likes.
 - [Edit button only if current user is sender.](#)
- **Kudo form:** photo uploader (allows both camera and library), photo preview, message input, receiver information.
 - New form is opened from a users screen.
 - [Edit form is opened from Kudo screen.](#)

- Validate presence of photo and message
- **Users.** List of all users. Show user avatar as initials. Tap on a user to open New Kudo form.
- **Profile.**
 - Sign out in the header.
 - User information.
 - Received kudos below.
 - [Show three categories under tabs: received, sent, liked kudos.](#)
- Bottom menu on Explore, Users and Profile screens.

Bonus challenges

Bonus tasks are blue. We expect you to do at least 2-3 of them.

Simplifications

In case you have some complications with completing our full task here are few possible simplifications:

- Don't make real authentication. As alternative, show a list of users and choose one to sign in with.
- Don't do photo upload. Give a set of images to choose from. Or give an input to insert an image url.

Technical tasks

We would be highly delighted to see you use and show a knowledge of:

- Packages: redux, redux-saga, router, photopicker.
- Technologies and techniques: Local Storage, Flexbox, FlatList, external API, Async Fetching, default data on loading, normalized data in redux store (no or minimal duplication).
- Code quality and code organisation: TDD, linter, proptypes, Higher-Order Components, es6 syntax.

Our main interest is a mobile part. As for a server side you can use services like Firebase. But if you build a server on your own that's also a big plus.

Requirement to code

- Should be available to test on real device (there are services like Expo);
- Provide all support & maintenance code like for real Commercial Production Product
- Deliver source code for review by GitHub