

Contactactor

Code red! I repeat code red! The native Contacts application in iOS and Android has been wiped out clean and nobody can find the source code. The IMDC (International **M**obile **D**evice **C**ommittee) has contacted us in order to release a new software which can replace the native Contact application (*at least for the time being*). This is where you come in, to take over this operation and deliver a new Contact application. As always, failure is not an option!

Structure

If the code does not follow the principles laid down here below, each group can receive up to -2 in deduction for their assignment. Here are the rules of structure:

- Code should be broken up into components which follow the **Single Responsibility Principle**
- Common logic should reside in a separate module and imported into components which make use of this logic
- The folder structure should be in accordance to the course lectures
- Consistency in code, meaning all group members should follow the same set of rules, e.g. 4 spaces as indent, Egyptian style curly braces, etc... hint: *eslint can help*

Assignment description

Here below is an enlisting of all the functionality this application should implement:

- **(50%)** Contacts screen
 - **(15%)** A user should see a list of all his saved contacts
 - **(5%)** Alphabetically ordered (*ascending order*)
 - **(10%)** Only displaying name and thumbnail photo
 - **(15%)** A user should be able to search for contacts by their name
 - **(10%)** Contact list filters dynamically based on search filter
 - **(5%)** Search should be a case insensitive substring search
 - **(20%)** A user should be able to create a new contact (see **Model** section below)
 - **(10%)** A form should be displayed which can be used to add a new contact
 - **(10%)** The user should be stored within the file system using **FileSystem** component provided by Expo. The user should be stored in the following format:
 - `<name-of-contact>-<uuid>.json` - *The uuid portion of the filename should be a generated uuid to distinct two persons who share the same name. The file content should contain a JSON object which contains the properties associated with each user: **name**, **phoneNumber** and **photo***

- **(30%)** Contact detail screen
 - **(10%)** A user should see detailed information on a selected contact
 - **(3.33%)** Name
 - **(3.33%)** Image
 - **(3.33%)** Phone number
 - **(20%)** A user should be able to modify the information on each contact
 - **(10%)** All properties should be editable
 - **(10%)** As a result of modifying the contact the JSON file associated with this contact should be recreated with the new information
- **(30%)** Extras
 - **(10%)** Import contacts from OS
<https://docs.expo.io/versions/latest/sdk/contacts/>
 - **(10%)** Add image with camera or photo import
<https://docs.expo.dev/versions/latest/sdk/imagepicker>
 - **(10%)** A user can make a call to a contact from the application

Model

Here all the models are enlisted and each property they should contain:

CONTACT

- Name
- Phone number
- Image

Submission

A single compressed file (*.zip, *.rar) should be submitted. Don't forget to delete **node_modules/** from the folder before compressing it and eventually submitting.