
Toodler
Make bed
Brush teeth
Shower
Practice

Toodler

You have been contacted by a world recognized agency called **Umbrella** and they specialize in making project management tools and are particularly interested in the **Kanban** technique. They are currently working on a product called **Toodler** and it hasn't been successful lately because of their inability to develop apps. Because they don't know how to make apps they need your help to create this fantastic product using **React Native**. As always failure is not an option!

Database

In this assignment you will not be working with any external database or web service but instead all data will reside in memory. This creates a small problem: every time you shut down the application, all the data will be wiped out clean as well. In **Canvas** a file is provided called **data.json** which contains pre-populated data which can be loaded into memory every time the application is started. Therefore you (*and your group*) won't have to populate everything manually each time the application is restarted.

Structure

Although structure is not explicitly part of the grading it will be taken into account. If the code does not follow the principles laid down here below, each group can receive up to -2 in penalties for their assignment. Here are the rules of structure:

- Code should be broken up into components which follow the **S**ingle **R**esponsibility **P**rinciple
- 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 product should implement:

- **(25%)** Boards
    - **(5%)** A user should see a list of all his boards within the **Board** view
    - **(7.5%)** A user should be able to create a new board. (*see **Models** section below*)
    - **(5%)** A user should be able to delete a board
    - **(7.5%)** A user should be able to modify an already existing board

- **(25%)** Board
    - **(5%)** A user should see all lists which are associated with a certain board
    - **(7.5%)** A user should be able to create a new list. (*see **Models** section below*)
    - **(5%)** A user should be able to delete a list
    - **(7.5%)** A user should be able to modify an already existing list
- **(30%)** Task
    - **(5%)** A user should see all tasks which are associated with a certain list
    - **(5%)** A user should be able to create a new task. (*see **Models** section below*)
    - **(5%)** A user should be able to delete a task
    - **(5%)** A user should be able to modify an already existing task
    - **(10%)** A user should be able to move tasks from one list to another
- **(20%)** Extras - *Students are allowed to do something extra which is not stated in the list above for a better grade. Do note that extras should only be implemented if everything above has been implemented.*

Models

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

BOARD
- Name
- Description (*optional*)
- Thumbnail photo

LIST
- Name
- Color (*a code representing an optional color of the list*)
- BoardId (*the link to a certain board*)

TASK
- Name
- Description
- Is finished? (*a boolean determining whether the task has been completed or not*)
- ListId (*the link to a certain list*)

*(If you feel the need to add models to support the **Extras** section, feel free.)*