

CS 3200 - Spring 2020 - Homework 5

Introduction

For this assignment you will be building a simple grocery list application. The purpose of this assignment is to introduce:

1. Redux
2. Alerts
3. Persistent Storage
4. React Native Vector Icons

Grocery List Usage

Your program will allow a user to create lists. They may have any number of lists. Each list can contain any number of grocery items. You can mark items as 'done'. You can also delete items, or delete entire lists.

Requirements

Create program that does the following

1. (15pts) Uses styling throughout. I don't care how things are styled as long as it doesn't hinder usability. I recommend using nativebase as much as possible.
2. (25pts) The main screen of the app shows a list of grocery lists.
 - a. Each list displays a name, and icon
 - b. Touching a list opens the list for use
 - c. A list can be delete by swiping to the left
 - d. A list can be edited by swiping to the right
 - e. There should be an option to create a new list
3. (20pts) Create / Edit list screen
 - a. User enters list name
 - b. User selects an icon. Can just be selected from a hard coded list of available icons
 - c. A save button that saves the list
 - i. If the user hasn't entered a name, or selected an icon an alert should be show to notify the user of the mistake and the list should not be saved
 - ii. Saving should redirect the user back to the main screen
 - d. A cancel button that that returns to the main screen without saving the list (you can just rely on the back button for this)

4. (30pts) The list screen. This is the screen that is visible when you select a shopping list from the main screen
 - a. Show a list of items needed for shopping
 - b. Touching an item should mark it as 'done'
 - i. There should be some visible indication that it has been marked as done (like maybe graying it out or adding a green check mark)
 - c. Touching an item again should mark it as not done
 - d. Items can be deleted by swiping to the left.
 - e. Adding items
 - i. There should be some obvious way to add items to the list
5. (25pts) Global State
 - a. You must use Redux to maintain app state
 - b. You should not use Redux to hold state that is transient (for example the state of a TextInput)
6. (25pts) Persistent Data
 - a. Your app should maintain all data between executions of your app
 - i. If you uninstall your app it will delete all your data. This can be useful if you screw up some saved data and want to delete it for a fresh start.
 - ii. I highly recommend doing this step last
7. (10pts) Code Review
 - a. App code should be DRY
 - b. App code should be as declarative as possible
 - c. App code should be consistently formatted.

Bonus points!

You can earn some extra credit by doing the following extra requirements. You can only earn extra credit if you have made a reasonable effort to do all the homework requirements (I don't expect anybody to have built the app perfectly but if there are major features just missing then don't expect to get extra credit). If at the end of the semester you have more than 100% on assignments then that will be capped at 100%. But this does mean, for example, that if you get 30 extra points on this assignment but then miss 30 on the next assignment that they will even out to 100%.

1. REQUIRED: Your app should start with an alert that tells the TA which bonus point items you completed, and how many points you think you deserve for it!
2. Adding Items
 - a. (10pts) When adding an item to a list the user should be able to search from a list of known items and select one.
 - b. (10pts) If the user wants to add an item that is not in the list of known items they should still be allowed to do that.

- c. (10pts) The user should be prompted to add items that aren't in the list of known items to the list of known items, this data should be saved between app executions.
- 3. Pictures with items
 - a. (15pts) the user should be able to select a picture from their device for each item and the app should remember which picture was selected on subsequent app executions