

TERM1

WEEK

12



TERM1 WEEK 12

MON DEC 23



FINAL EXAM

WHERE ?

WHEN ?

IDT

8.30 – 10.30 AM

WED DEC 25 - *Midnight*



PROJECT SUBMISSION

- ✓ Submit Source code
- ✓ Submit Video Presentation

(GitHub repository link)

(YouTube link)

THU DEC 26



BEST PROJECT SELECTION

- ✓ 12 projects selected for the Flutter Awards

FRI DEC 27



FUTTER AWARDS

COSMO HALL

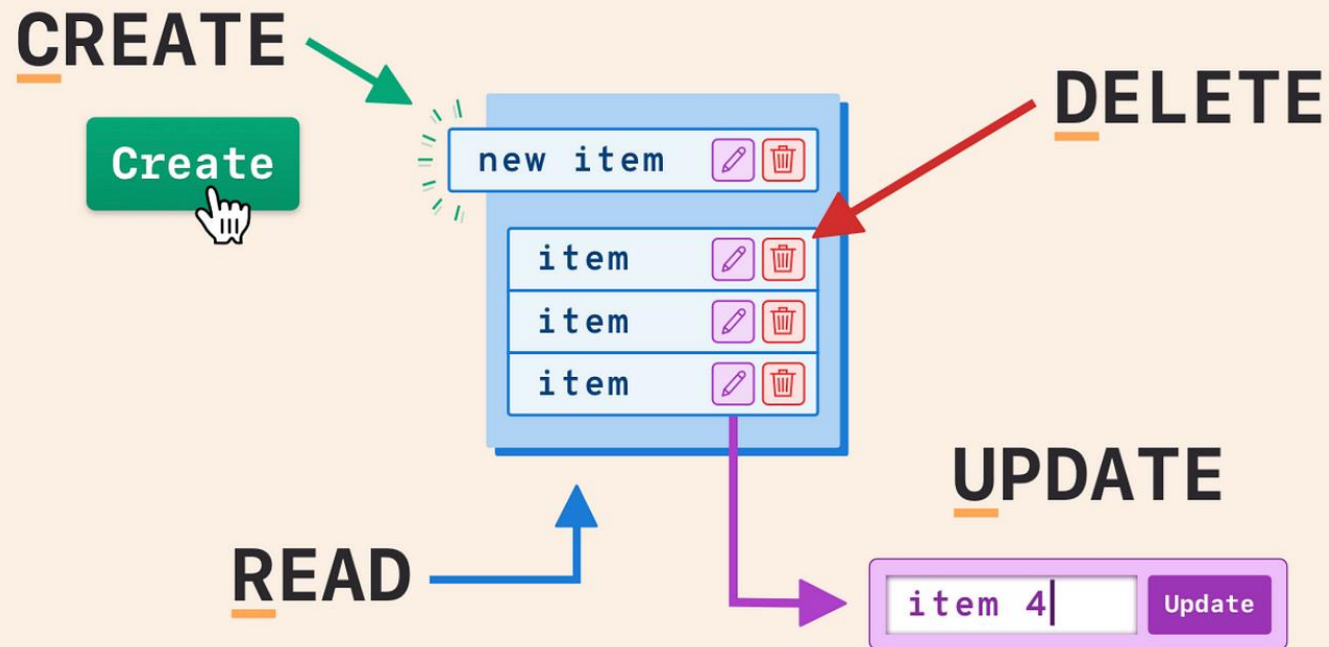
1.00 – 3.00 PM

- ✓ Selected students present their project to jury
- ✓ Certifications

- ✓ *Cheer Time “Share what you brought”*

MOBILE DEVELOPMENT

W10-S1 – CRUD operations



What will you be able to do?

Validate data with Forms

A form for adding a new item. It has two input fields: 'Name' and 'Quantity'. The 'Name' field has a validation message 'Must be between 1 and 50 characters.' and a character count '0/50'. The 'Quantity' field has a validation message 'Must be a valid, positive number.' and a category selector 'Fruit'. There are 'Reset' and 'Add Item' buttons at the bottom.

Pass data back to previous screen

A screen titled 'Your Groceries' showing a list of items. The first item is 'Shampoo' with a purple icon. There is a '+' button at the bottom right and a '2' at the bottom left, indicating the number of items.

Multiple selections using a mode

A screen titled '1 selected Item(s)' showing a list of items with checkboxes. The items are 'a', 'b', and 'c'. Item 'b' is selected, indicated by a checked checkbox. There is a trash icon at the top right and a '1' at the bottom right.

Edit an existing item

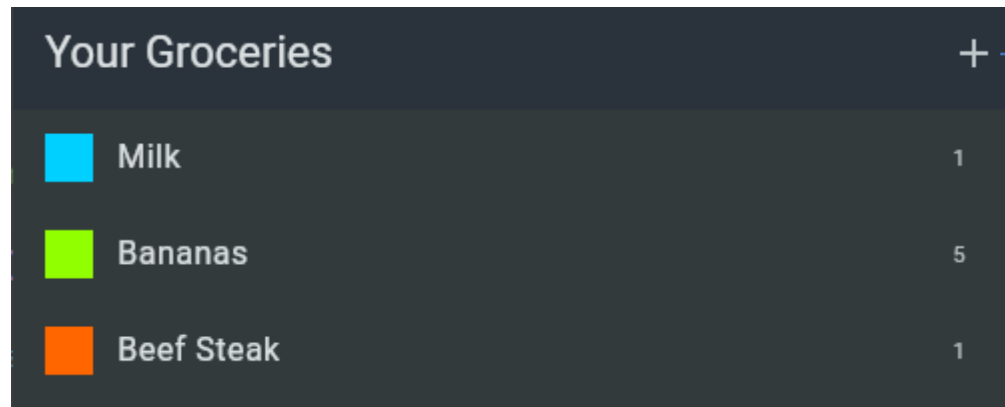
A screen titled 'Edit item' showing a form for editing an existing item. It has input fields for 'Name' and 'Quantity'. The 'Name' field has a value 'aa'. The 'Quantity' field has a value '3/5'. There is a category selector 'Sweets'. There are 'Reset' and 'Edit' buttons at the bottom.

Drag and drop operations

A screen titled 'Your Groceries' showing a list of items. The items are 'a' (green icon), 'c' (orange icon), and 'b' (blue icon). Each item has a quantity '1' and an equals sign '=' at the bottom right. There is a '+' button at the top right.

Open ItemForm

- ✓ When click on the plus button, **switch to the Form**
- ✓ **Observe** the new widgets related to **Form**



Form

TextFormField

The form alternative of TextField



DropdownButtonFormField

The form alternative of DropdownButton

Manages multiple fields **with forms**

<https://docs.flutter.dev/cookbook/forms/validation>

validator callback

Name

Must be between 1 and 50 characters. 0/50

Quantity

Must be a valid, positive number.

Fruit

Reset Add Item

Persist and reset
inputs

Form Usage – *Step by step*

1. Define a GlobalKey<FormState>

```
final _formKey = GlobalKey<FormState>();
```

← to access and control
the state of the Form.

2. Wrap input fields in a Form widget and link its **key** to the GlobalKey

```
Form( key: _formKey, ...
```

3. Use the **validator** property in fields like TextFormField to enforce rules.

```
TextFormField( validator: _yourValidator
```

4. Use the **validate()** method to check all fields. If valid, proceed with the logic

```
if (_formKey.currentState?.validate()) { do something }
```

5. Use the **save()** method to persist form data

```
formKey.currentState?.save())
```

Manage the form **name** field

- ✓ Create a **global key** and link it to the form
- ✓ Define a **validator** for the **name** TextField
- ✓ Define a **callback onSave** for the **name** TextField
- ✓ *On Add Clicked:*

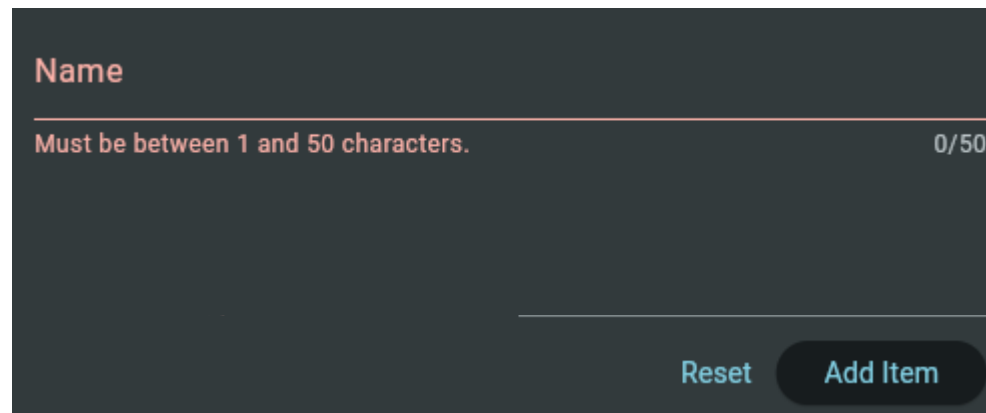
- ✓ **validate** the form

```
_formKey.currentState!.validate();
```

- ✓ **save** the form

```
_formKey.currentState!.save();
```

- ✓ Print the **name** on console



The screenshot shows a dark-themed UI element. At the top, the word "Name" is written in a light color. Below it is a horizontal line representing a text input field. Underneath the line, the text "Must be between 1 and 50 characters." is displayed in a light color, indicating a validation error. To the right of this text, "0/50" is shown. At the bottom of the form, there are two buttons: "Reset" and "Add Item".

Manage the form **quantity** & **category** fields

- ✓ Define a **validator** for the **quantity**
- ✓ Define a **callback onSave** for the **quantity**
- ✓ For the **category**, manage the field value with a state as usual
- ✓ *On Add Clicked:*
 - ✓ Print the **name** + **quantity** + **category** on console

Quantity

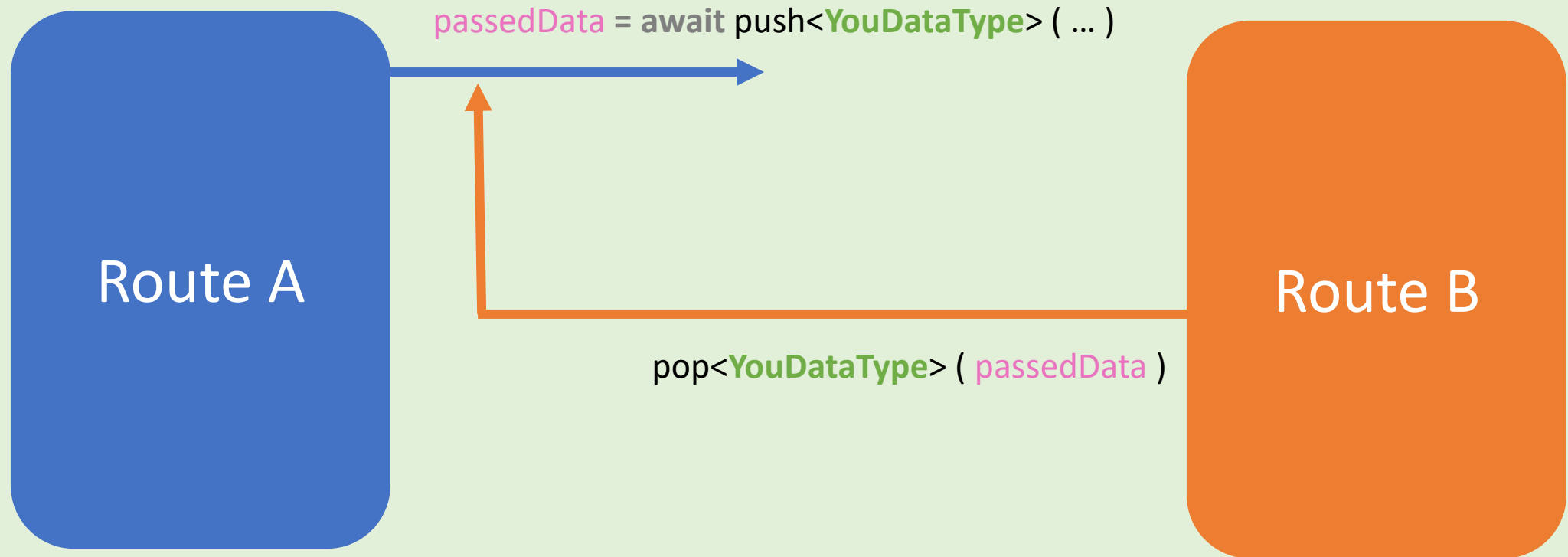
Must be a valid, positive number.

Fruit

Reset Add Item

Return data from a screen

<https://docs.flutter.dev/cookbook/navigation/returning-data>



- ✓ The previous screen wait asynchronously for the pop.
- ✓ After the pop, the push can return the *passed Data*

Pass the **new grocery** to previous screen

- ✓ *On Add Clicked:*
 - ✓ Create a new **Grocery Item**
 - ✓ Pass the created item to the previous screen, using, using the **pop()** return value
 - ✓ On previous screen, get the created item, using an **await push**
 - ✓ **Update the list of groceries**

The screenshot displays two parts of a mobile application interface. The top part is a form titled 'Add a new item' with a back arrow. It contains input fields for 'Name' (Shampoo), 'Quantity' (2), and a category dropdown (Hygiene). There are 'Reset' and 'Add Item' buttons at the bottom right. The bottom part is a list titled 'Your Groceries' with a plus icon. It contains one item: 'Shampoo' with a quantity of 2. A red arrow points from the 'Add Item' button in the top form to the 'Shampoo' item in the bottom list.

Implement the **edit** item action



You should the **SAME** screen
For both EDITING and CREATING

RECOMMENDATIONS

- ✓ Create an enum Mode : *editing, creating*
- ✓ Depending on enum change the text of the button and header
- ✓ The edited Grocery item can be passed to the form as optional (null if the mode is creating)
- ✓ In edition model, fields are initialized with the passed grocery item
- ✓ In edition model, update the grocery item in the list after clicking on edit

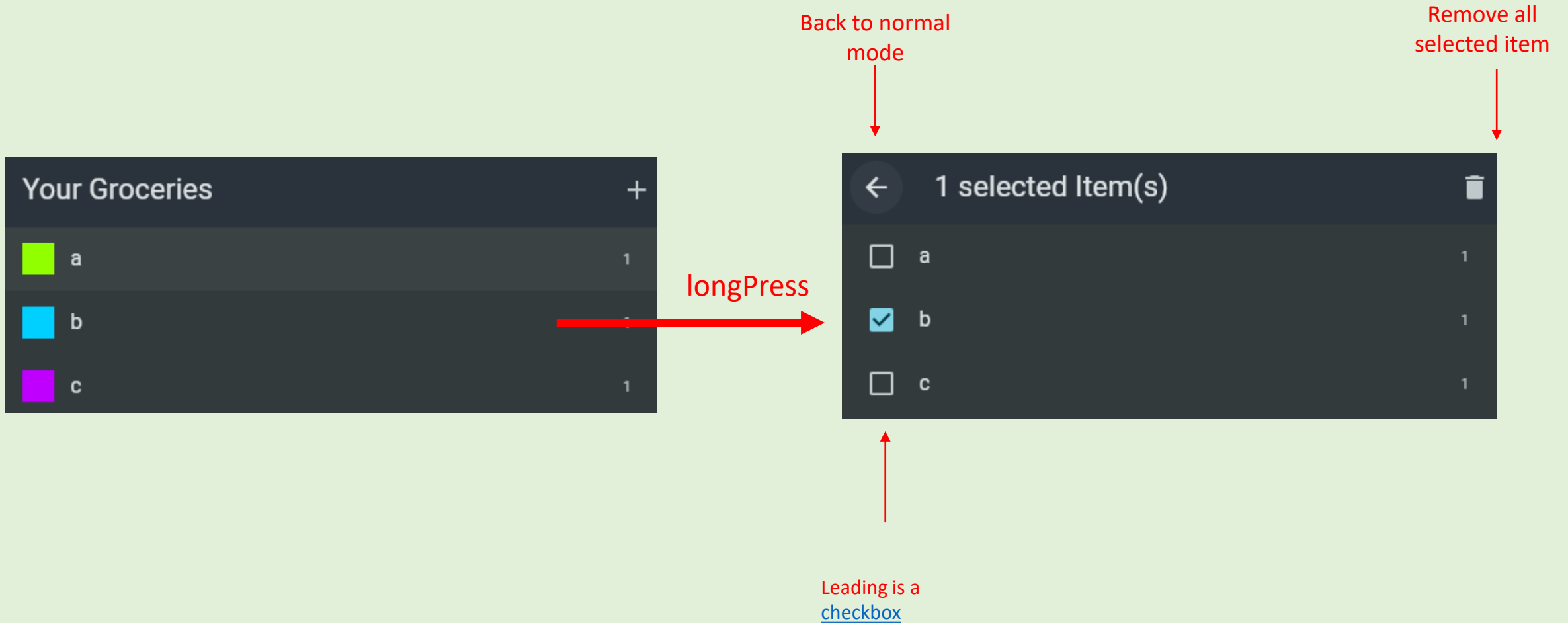
The image displays three screenshots of a mobile application interface for managing groceries, connected by red arrows indicating the flow of the 'edit' action.

Top Screenshot: 'Your Groceries'
This screen shows a list of grocery items. The first item is 'aaa' with a quantity of 1. The second item is 'bbccc' with a quantity of 5. A red arrow labeled 'onClick' points from the 'aaa' item to the 'Add a new item' screen.

Middle Screenshot: 'Add a new item'
This screen is used for adding new items. It has a header with a back arrow and the text 'Add a new item'. The form includes a 'Name' field, a 'Quantity' field (set to 1), and a category dropdown menu (set to 'Fruit'). At the bottom right, there are 'Reset' and 'Add' buttons. A red arrow labeled 'tap' points from the 'Add' button to the 'Edit item' screen.

Bottom Screenshot: 'Edit item'
This screen is used for editing existing items. It has a header with a back arrow and the text 'Edit item'. The form is pre-filled with the item 'aaa' and a quantity of 1. The category dropdown menu is set to 'Sweets'. At the bottom right, there are 'Reset' and 'Edit' buttons.

Selection Mode



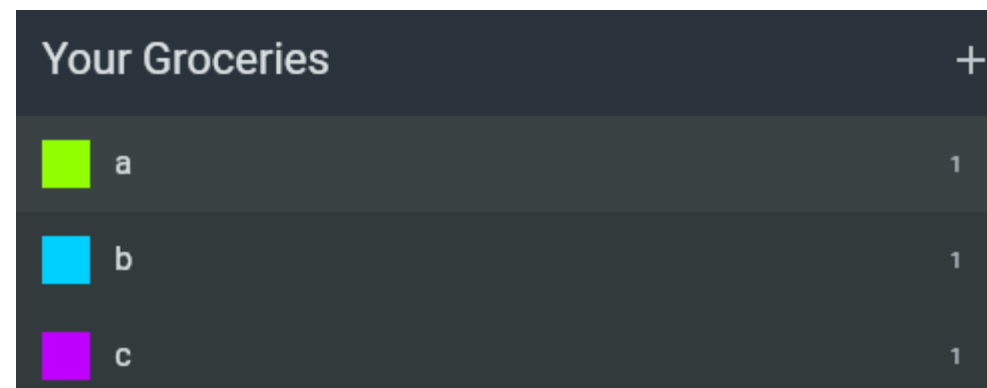
Implement the selection mode



You should the **SAME** screen
For both NORMAL and SELECTION mode

RECOMMENDATIONS

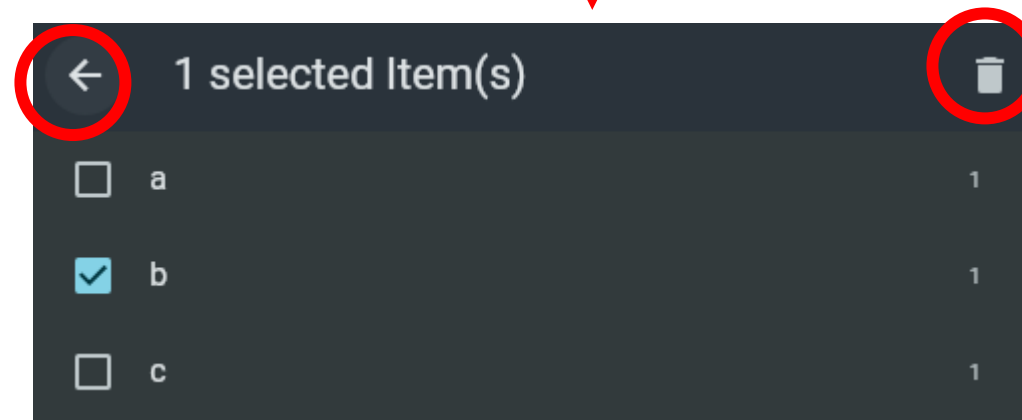
- ✓ Create an enum Mode : *normal*, *selection mode*
- ✓ Depending on the **mode** update the App bar actions and title
 - *The bar displays the number of selected item on selection mode...*
- ✓ On long press, switch to **selection mode**
- ✓ If selection model, display a **checkbox** on each tile leading
- ✓ When check box clicked, **update the selected item** (model)
- ✓ When clicking on remove button, remove the selected items + switch to **normal mode**
- ✓ When clicking on back button, switch to **normal mode**



Back button

longPress

Remove button



Implement the items re-order



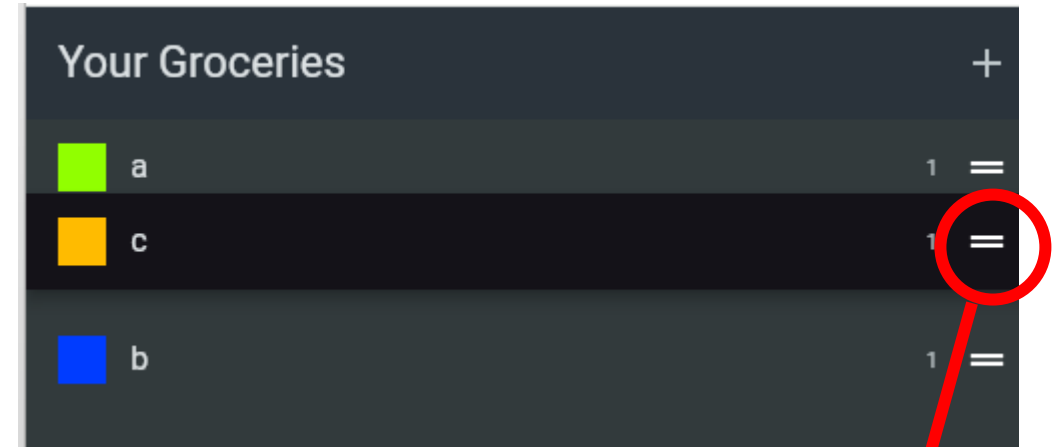
You should the **SAME** screen
For both NORMAL and SELECTION mode

RECOMMENDATIONS

- ✓ Use the ReorderableListView

<https://api.flutter.dev/flutter/material/ReorderableListView-class.html>

<https://www.youtube.com/watch?v=3fB1mxOsqJE>



Drag and drop

