

PROG2007 - Assignment 2

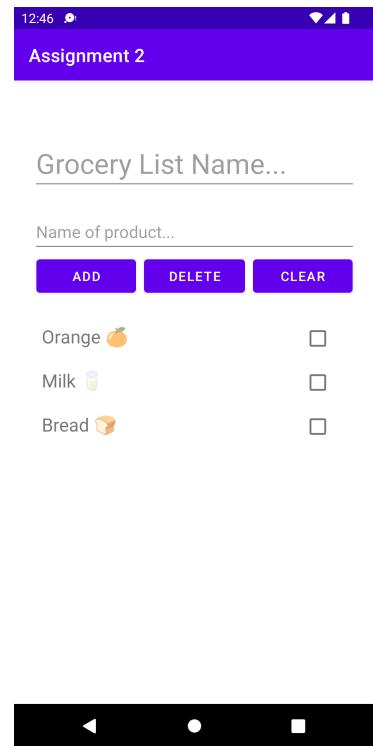
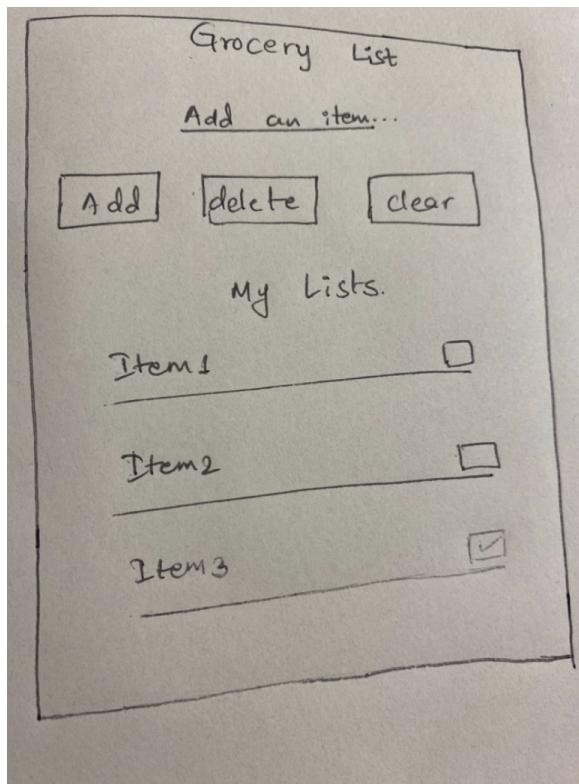
Student: Jonas Johan Solsvik

Deadline: Sunday, 2021-10-24 @23:59

Task 1 - Kotlin/Android Studio

1.1 Overall design

Let's consider you are making the Grocery shopping list application. Create an app that allows you to add items in your shopping list, delete items or clear all items from the list once you have finished shopping. The design of the app should look as shown in the figure.



The assignment sketch on the **left**. The final result on the **right**.

1.2 Feature checklist

Your application should have following features:

1. A text space to add the new items ✓

The screenshot shows a simple user interface. At the top is a text input field with the placeholder "Name of product...". Below the input field are three purple rectangular buttons with white text: "ADD", "DELETE", and "CLEAR". An orange oval highlights the "ADD" button.

2. An “Add” button which allows to add new entries on the list ✓

The screenshot shows a list item "Banana" followed by three purple buttons: "ADD", "DELETE", and "CLEAR". The "ADD" button is circled in orange.

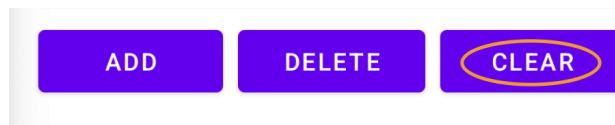
3. A “Delete” button to delete the entries from the list ✓

The screenshot shows a list of items: "Orange" with an orange icon, "Milk" with a glass icon, and "Bread" with a bread icon. To the right of each item is a small square checkbox. The checkboxes for "Orange" and "Milk" are checked (containing a blue checkmark), while the checkbox for "Bread" is empty.

4. Checkboxes next to the added entries. The checkbox should allow to select an item such that you can remove a specific item from the list once you bought it. ✓

See screenshot above.

5. A “Clear” button which clears the entire list at once ✓



6. A scrollable list to maintain a long list of generated items ✓

The image shows a mobile application interface for a grocery list. At the top, there is a title "My long grocery list" with a teal underline. Below the title is a search bar containing the text "Fish". Underneath the search bar are three purple buttons: "ADD", "DELETE", and "CLEAR".

The main area displays a list of items with checkboxes to their right. An orange circle highlights the vertical scroll bar in the center of the list. The items listed are:

- Bread 🍞
- Popcorn
- Chocolate
- Beer
- Paper
- Books
- Toothpaste

For each item, there is a small icon to its left and a checkbox to its right. The "Books" item has a checked checkbox.

Item	Icon	Checkmark
Bread	🍞	□
Popcorn		□
Chocolate		□
Beer		□
Paper		□
Books		✓
Toothpaste		□

Task 2 - Gordon Rasmey cooking app

You are hired by Gordon Ramsay to develop a cooking app. He provided some of his main concerns about the app as below:

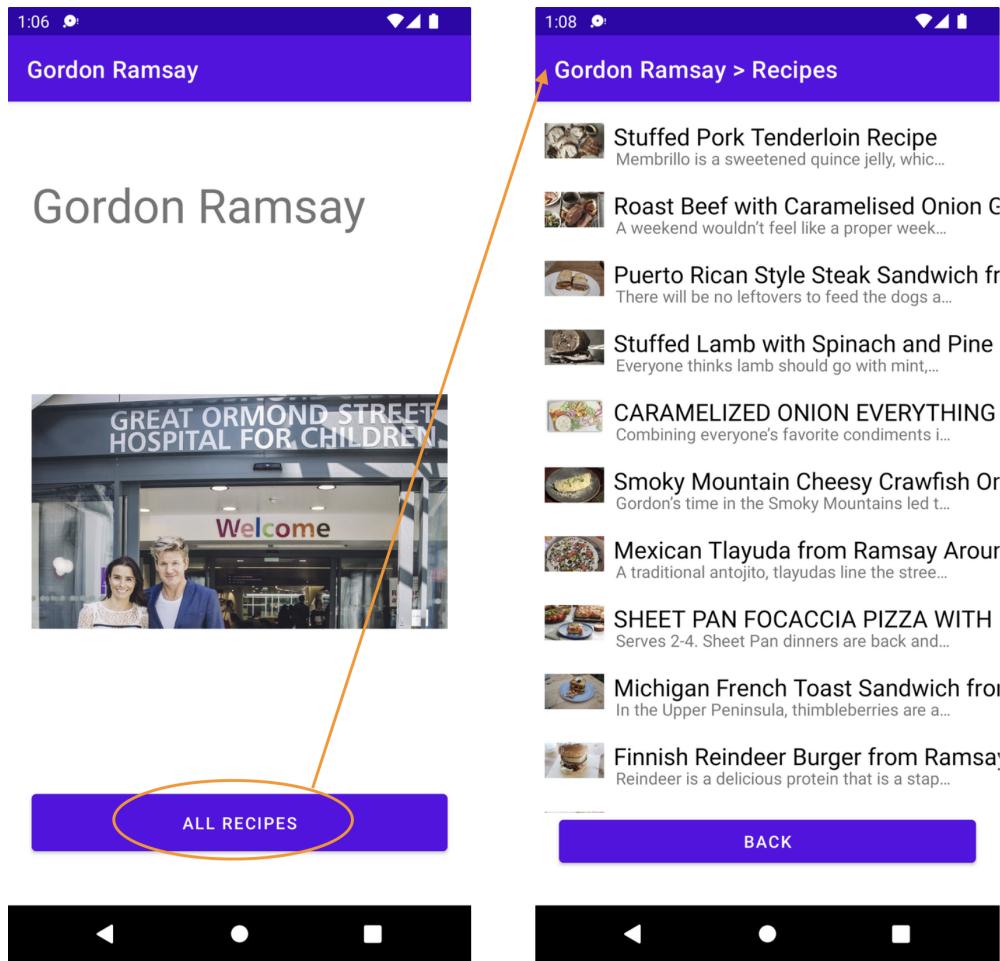
2.1. In the first page of the app, he wants his picture and his name.



Gordon Ramsay



2.2 After clicking on his name another page will open where you can see a list of food with their name and pictures.



Bonus feature: The list of recipes is dynamically fetched over the internet by making a HTTP request to <https://www.gordonramsay.com/gr/recipes/> and parse its contents. This makes sure that the list is always up to date.

2.3 By clicking on each food's name another page will open

1:08 Gordon Ramsay > Recipes

- Stuffed Pork Tenderloin Recipe
Membrillo is a sweetened quince jelly, whic...
- Roast Beef with Caramelised Onion G
A weekend wouldn't feel like a proper week...
- Puerto Rican Style Steak Sandwich fr
There will be no leftovers to feed the dogs a...
- Stuffed Lamb with Spinach and Pine
Everyone thinks lamb should go with mint,...
- CARAMELIZED ONION EVERYTHING
Combining everyone's favorite condiments i...
- Smoky Mountain Cheesy Crawfish O
Gordon's time in the Smoky Mountains led t...
- Mexican Tlayuda from Ramsay Arour
A traditional antojito, tlayudas line the stree...
- SHEET PAN FOCACCIA PIZZA WITH
Serves 2-4. Sheet Pan dinners are back and...
- Michigan French Toast Sandwich fro
In the Upper Peninsula, thimbleberries are a...
- Finnish Reindeer Burger from Ramsa
Reindeer is a delicious protein that is a stap...

BACK

◀ ⏴ ▶

1:14 Gordon Ramsay > Recipe

Smoky Mountain Cheesy Crawfish Omelette from Ramsay Around The World

Gordon's time in the Smoky Mountains led to the discovery of an amazing farmstead cheese made from raw cow's milk, flavored with basil and dried tomatoes. Local ingredients like crawfish and mushrooms take this omelette out of the mountains and into your

BACK

◀ ⏴ ▶

2.4. In this new page you will see the picture of the food, its name, and a short description about the food's recipes.

(SEE SCREENSHOT in section 2.3)

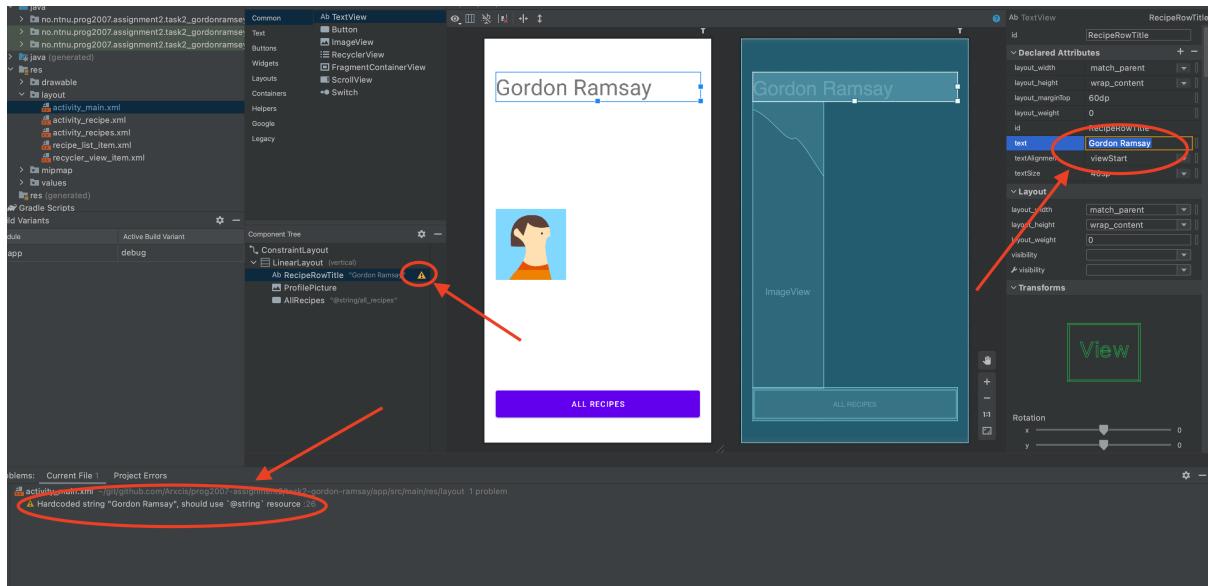
2.5. bonus: Add the search option to search between foods.

(NOT IMPLEMENTED, but added another bonus thing which is mentioned in section 2.2)

Task 3: Android Studio Quiz

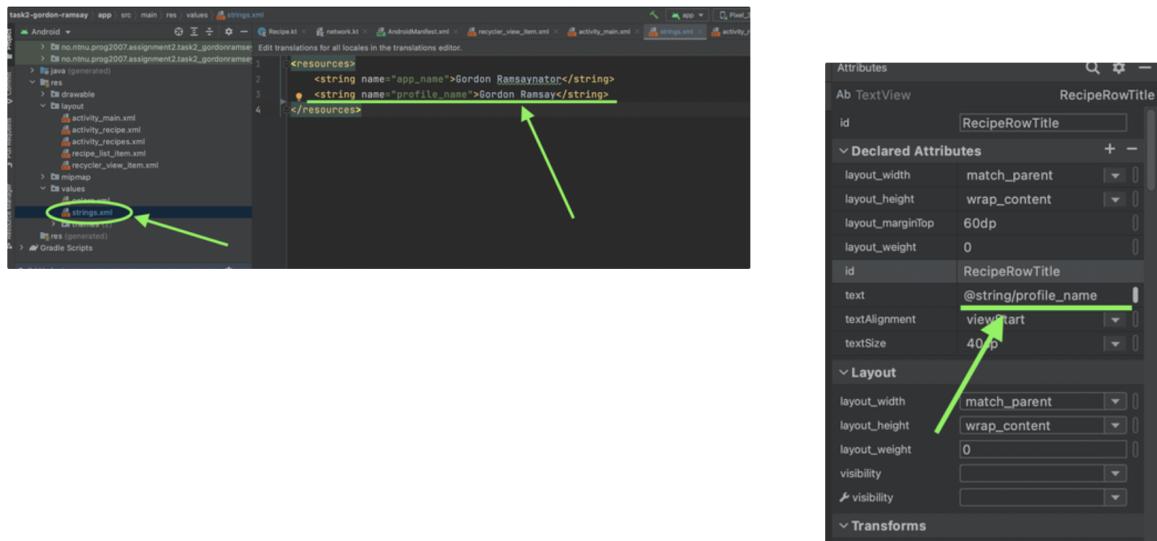
3.1 and 3.2 Create a textView by using the Design menu and change the text. Did you notice the "hardcoded text" warning? How could you resolve it? And why is it important to resolve it?

When creating a TextView, the default text is always textView. Maybe you want to change the text to "Name" instead. If you just type it directly into the Design Menu, you will get the warning mentioned above.



The solution to this problem is to move the hardcoded text-string into a resource file. The default resource-file for strings can be found at `res/values/strings.xml`. Putting our string in this file and referring to it with the id

`@string/profile_name` resolves the issue.



Why is this important?

There are two main reasons why we want to do this:

1. **Reusability** - The same string may be used in many places. Having strings in a central location makes it easy to change text across the codebase from a single location, which is linked to the second reason.
2. **Localisation** - In the beginning we may only write english text in our app, but if we want to add support for a second language, we could now do this easily by adding a new file with the name `res/values-no/strings.xml` for people with Norwegian-locale configured on their phone. For every new language, we just add a new file:

```
res/  
  values/strings.xml      // default (english)  
  values-no/strings.xml   // norwegian  
  values-fr/strings.xml   // french  
  values-se/strings.xml   // swedish
```

For more see: <https://developer.android.com/guide/topics/resources/localization>

3.3 In the case that your application contains several textviews and you want to keep the same design for all of them, what is the best solution?

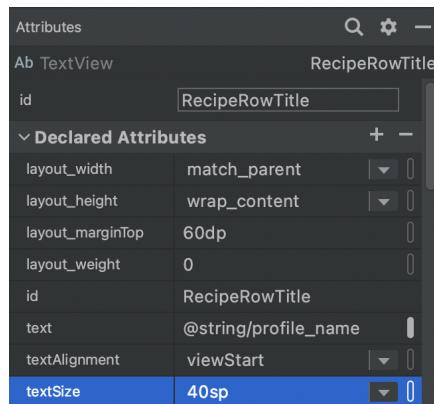
To keep the same design for all textviews, we could create a `style.xml`-file together with the existing `color.xml` and `strings.xml` files.

```
res/  
  values/  
    strings.xml  
    color.xml  
    styles.xml // <-- Create this file
```

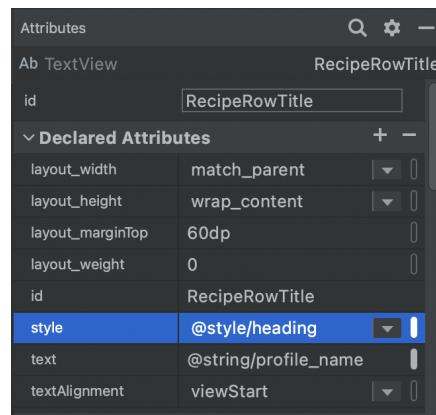
The contents of `styles.xml`:

```
<?xml version="1.0" encoding="utf-8"?>  
<resources>  
  <style name="heading">  
    <item name="android:textSize">40sp</item>  
  </style>  
</resources>
```

Example: Hardcoded textSize = 40sp



Example: Softcoded textSize imported from `styles.xml`:



For more info, see: <https://developer.android.com/guide/topics/resources/style-resource>

3.4 So far, we have used TextView, ImageView, and buttons. Can you name two more views? Could you name some properties that all of these views have in common? What is the unit for expression location and dimension of the views?

Naming two more views:

- CheckBox
- RecyclerView

Common view properties:

- layout_margin
- layout_height, layout_width
- layout_weight.
- id

Android Dimension units:

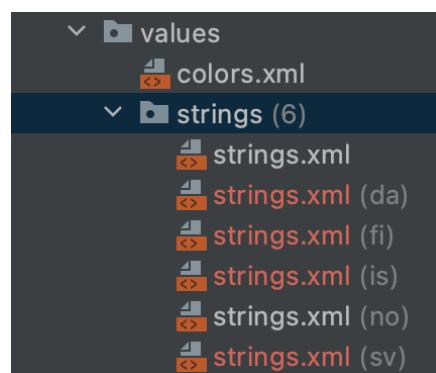
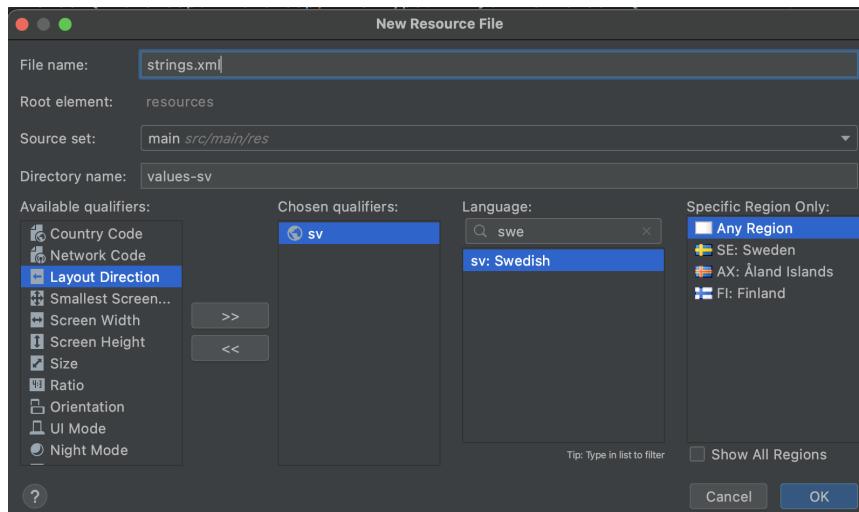
- **dp** - Density-independent Pixels - An abstract unit that is based on the physical density of the screen.
- **sp** - Scale-independent Pixels - This is like the dp unit, but it is also scaled by the user's font size preference.
- **pt** - Points - 1/72 of an inch based on the physical size of the screen, assuming a 72dpi density screen.
- **px** - Pixels - Corresponds to actual pixels on the screen.
- **mm** - Millimeters - Based on the physical size of the screen.
- **in** - Inches - Based on the physical size of the screen.

For more info, see: <https://developer.android.com/guide/topics/resources/more-resources.html#Dimension>

Task 4 - Localization

Localization is an integral design part of any application due globalization these days. Localization refers to adapting application content for a specific region. Read and learn more about localizing your app at <https://developer.android.com/guide/topics/resources/localization>. Your task is to localize your application you develop in Task 1 to Nordic countries. List and explain what changes you've made to localize your app.

4.1 Adding nordic locales files



4.2 Viewing locales in Designer

