

# CMPS 312 Mobile Application Development – Assignment 2

## QuickMart Shopping App

**Deadline:** Sunday 15<sup>th</sup> October at 11:59 PM

### Instructions:

In this assignment, you will design and implement QuickMart, a Grocery Shopping App with 3 screens:

- Product screen for browsing and searching products and adding desired ones to the cart.
- Cart screen for updating the quantity of cart items or deleting a cart item.
- Home screen to switch between the two screens.

Required implementation tasks:

#### 1. Project Setup

- Sync the Lab GitHub repository and copying the **QuickMart** folder into your own repository under /assignments/assignment2 subfolder.
- Open the **QuickMart** project in Android Studio. The project contains the following components:
  - **drawable**: contains images used in the app.
  - **assets**: includes **products.json** and **product-categories.json** files.

#### 2. Data Model: Create two serializable data classes: Product and CartItem.

<pre>+-----+            Product            +-----+   - title: String               - category: String            - description: String         - price: double               - rating: int                 - imageName: String         +-----+</pre>	<pre>+-----+            CartItem           +-----+   - productId: String           - productName: String        - quantity: int              - unitPrice: double         +-----+   + calculateTotalPrice(): double   +-----+</pre>
--	--

#### 3. Data Repository

- Create a Kotlin object named **ProductRepository** having a **productList** property.
- Implement the **initProducts(context: Context)** function to read and parse data from **products.json** into the **productList**.
- Implement **getProducts(name: String, category: String = "All")** to search products by name and category (i.e., return the products having a name that contains the name parameter and the category matches the category parameter).
- Implement **getProductCategories(): List<String>** to return the product categories from **product-categories.json**

- Create a Kotlin object named **CartRepository** having a **cartItems** property.
- Implement **addItem(cartItem: CartItem)** to add an item to the cart.
- Implement **updateItem(productId: Int, quantity: Int)** to update the quantity of an item in the cart.
- Implement **getCartTotal()** to return the total amount of items in the cart.

#### 4. Product Screen

- Implement and test **ProductCard**, **ProductList** and **ProductScreen** composables as shown in Figure 1. The card should define the layout for a single product, and **ProductList** should load and display the list of products using a **LazyVerticalGrid**.
- Allow users to add products to their cart from the Product screen.
- Add the ability to search products by name and category (tip: add a **TopAppBar** with search text box and product categories dropdown).

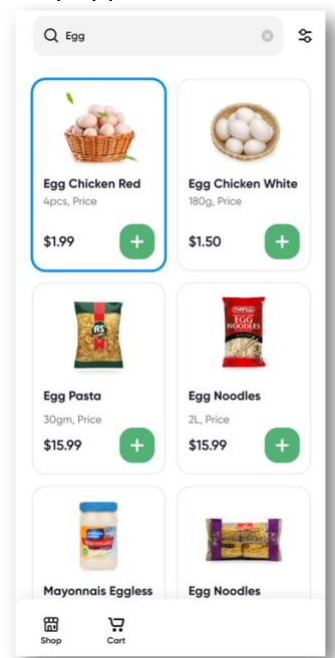


Figure 1. Product screen

#### 5. Cart screen

- Implement and test **CartItemCard**, **CartItems** and **CartScreen** composables as shown in Figure 2. To display the cart items including their name, quantity, and total price.
- Implement the functionality to allow the user to update the item quantity and to remove items from the cart.

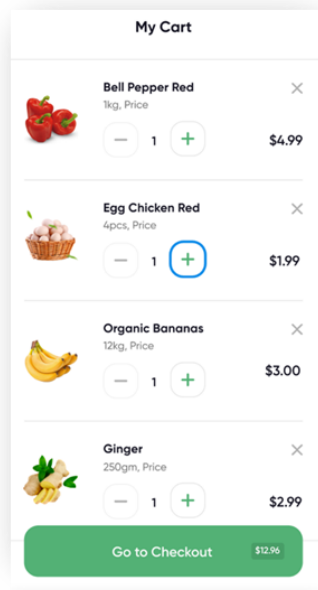


Figure 2. Cart screen

## 6. Home screen

- Implement and test **HomeScreen** composable as shown in Figure 3 having a Scaffold and a BottomAppBar for switching between the Product Screen and Cart Screen.

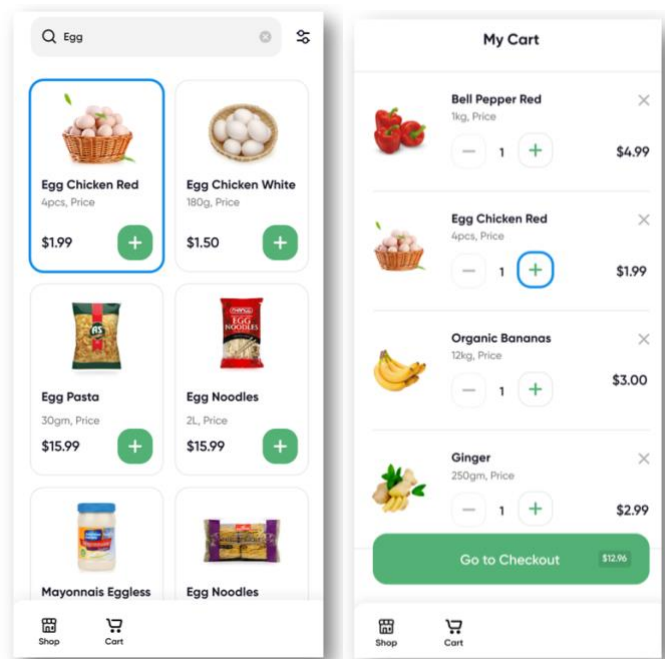


Figure 3. Home screen

### Submission:

Besides your implementation, make sure you submit the testing document using the provided Word template and including screenshots as evidence of your working implementation.