**CMPS 312 – Mobile Application Development**

**Midterm Practical Exam – 20 October 2021 (duration 2.5h)**

|  |  |
| --- | --- |
| Student Name (Student ID) |  |

**Objective:** Build a Shopping App on Android to demonstrate your Mobile development skills.

Sync cmps312-Lab repo then copy to your GitHub repo the midterm folder which contains an empty solution that has all the needed dependencies and an *assets* folder that has *shopping-list.json* &***fruits.json***Json files.

# Implementation Tasks

Open the ShoppingApp under midterm subfolder inside your GitHub repository.

## Shopping List Screen

|  |  |
| --- | --- |
|  | 1.1. Add a ShoppingList screen to display the shopping list items. When the screen loads get the shopping items and display them in a LazyColumn**.**  **Tip**: Use a shoppingList state variable in the ShoppingViewModel to hold the shopping list items read from the provided ShoppingRepository.getShoppingList  1.2. Provide a delete icon next to each item in the list to allow the user to click it to delete the item.  **Tips**: - Use Icons.Outlined.*Delete* as the icon for delete  - Add deleteItem method to the ShoppingViewModel to delete the item from the shoppingList |

1.3. Add a Floating Action Button (FAB) to the ShoppingList screen to allow adding an item to the shopping list. When the user clicks on the add button, the app should navigate to **ShoppingItem** screen.

Tip: you may use a simple button instead of FAB if it is easier for you.

## Shopping Item Screen

|  |  |
| --- | --- |
|  | 2.1. Add a ShoppingItem screen to allow the user to select an item from a fruits dropdown and specify the desired quantity. When the user submits the form, the app should add the new item to the shopping list then navigate back to the ShoppingList screen. If implemented correctly the new item should auto-appear in the shopping list.  2.2. Add a title and a back arrow to the screen, when the back arrow is clicked then the app should navigate up to the Shopping List screen.  Tips:  - Add getFruits method to ShoppingViewModel to get the list of fruits from ShoppingRepository.getFruits. Use the provided Dropdown component to create and fill the fruits dropdown.  - Add addItem method to the ShoppingViewModel to add the item from the shoppingList.  - Use ShoppingViewModel as a shared ViewModel to allow communication between the app screens. |

## Grading Rubrics

|  |  |  |  |
| --- | --- | --- | --- |
| **Criteria** | **%** | **Functionality**\* | **Quality of the implementation** |
| 1. Shopping List screen | 60 |  |  |
| 1. Shopping Item Screen | 40 |  |  |
| Provide screenshots in *testing.docx* | - |  | *[-10pts if missing]* |

**\* Possible grading for functionality**: ***Complete and*** ***Working*** (get 70% of the assigned grade), ***Complete and*** ***Not*** ***working*** (lose 60% of assigned grade) and ***Not done*** get 0. The remaining grade is assigned to the quality of the implementation. Must submit screenshots in the **Testing-GradingSheet.docx** (otherwise -10pts).