**CMPS 312 – Mobile Application Development**

**Final Practical Exam – 17 November 2021 (duration 2.5h)**

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

Sync cmps312-lab repo, then copy the FinalExam folder to your GitHub repo. It contains an empty solution with all the needed dependencies.

Implement a Pizza Ordering app to allows users to add pizzas for customers. The user can add a customer and add pizza for that customer. The app data shown in Figure 1 should be managed in an SQLite database using **Room Library.**

Diagram

Description automatically generated *Figure 1. Product Review Entity-Relations*

1. Create two Entity classes **Customer and Pizza** with proper annotations. The properties of each entity are shown in Figure 1. Note that ids should be auto assigned by the database. Make sure you add foreign key annotation with cascade delete and update.
2. Create a PizzaOrderDAO Interface and add the necessary functions and annotations to allow the user to:
   * **Add** a new Customer
   * **Get** all customers
   * **Delete** a customer **and** all associated **Pizzas**
   * **Add** a new Pizza for a customer
   * **Get** all Pizza orders for a specific customer given customerId
   * Update a Pizza
3. Implement the PizzaOrderRepository with all necessary methods listed above.

The app UI and the requirements for each screen are described in the table below.

Customer

List Screen

Create

**Customer**

screen to display the

list of

**customers**

.

When the screen loads, get the

**customers**

and display them in a LazyColumn.

Provide a delete icon next to each

product

in the list to allow the user to delete

it

.

When a product is deleted

,

its associated reviews

should

be deleted as well.

**Add**

**Customer**

**Screen**

The Add

Customer

screen

allows the user to enter

the

customer

details

such as

the

customer

name

and

address

.

When the

submit

button is clicked, the app

should add

the product

to the database

and

navigate

back to

the

Customers

List screen.

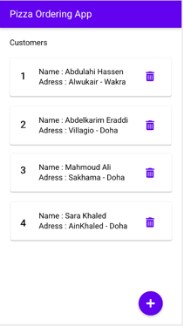
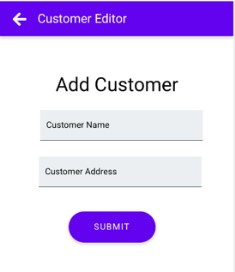
The

customers

list should be auto refreshed to list all the

customers

including the newly added one.



|  |  |
| --- | --- |
| **Pizza List**      **Screen**                              **Add / Update**  **Pizza Order** | When the user clicks on the customer clickable card, the app should navigate to the PizzaList Screen. This Screen should list the pizza ordered by the selected customer and let the user add or update pizza orders.    When the user clicks on the add button, the application should navigate the user to the PizzaEditor Screen that lets the user add a pizza type (Veggie, Hawaiian, Buffalo, Pepperoni and Margherita) . Also the user should have the option to select the pizza size (Small, Medium, or Large). Both the Pizza name and pizza size should be a Dropdown.    When the user clicks the Submit button, the app should add the pizza to the database and navigate back to the PizzasList screen. The pizzas list should be auto refreshed to list all the customer pizzas including the newly added one.    When the user clicks the edit button the app should load the details of the selected pizza in the Pizza screen to allow the user to edit them.  When the user clicks the Submit button, the app should update the pizza in the database and navigate back to the Pizzas List screen. The Pizzas list should be auto refreshed to list all the pizzas including the updated one.    Note that Add and Update pizza should reuse the same Pizza screen. |

**Grading Rubrics**

|  |  |  |  |
| --- | --- | --- | --- |
| **Criteria** | **%** | **Functionality\*** | **Quality of the implementation** |
| **1. Customer List** | 20 |  |  |
| **2. Add Customer** | 20 |  |  |
| **3. Delete Customer** | 8 |  |  |
| **4. Pizza List** | 20 |  |  |
| **5. Add Pizza** | 16 |  |  |
| **6. Update Pizza** | 16 |  |  |
| **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 30% of assigned grade)
* *Not done* get 0
* The remaining grade is assigned to the quality of the implementation.
* Must submit screenshots in this Testing-GradingSheet.docx (otherwise - 10pts).