Project 3:
Tasks

1. (10 pts) Provide the problem requirements and the conceptual model in UML for your project. You can reuse the one made on previous projects, but describe the functionalities that you selected to be used as an in-memory key-value storage, (e.g. most viewed products, a shopping cart, current logged-in users, etc).

Verbs & nouns
Problem:
Within the local community, there is a challenge in identifying and promoting the best drinks and desserts available across various cafes. Visitors often find it difficult to decide where to order, and once they make a choice, there is a risk of the selected item being misrepresented on the menu. This discrepancy between expectation and reality can result in unsatisfied cravings and a less-than-optimal dining experience. The lack of accurate and detailed information about the offerings at different cafes makes it challenging for people to discover the hidden gems within the community's beverage and dessert scene.

Objective:
Our objective is to create a database that catalogs cafes within the local community, listing the various foods and drinks available at each cafe. This database allows users to explore and compare different cafes based on the specific types of drinks and foods they offer. By compiling this information the goal is to recommend cafes to the community and allow them to see which cafe is best for a type of craving, encouraging them to discover different locations in the area.

Rules:

- Catalog the different drinks and desserts in each cafe
- Identify and compare cafes to find which ones are most to least recommended
- Identify and compare drinks and desserts of each cafe to find which ones are most to least recommended
- Promote recommended cafes based on different communities and locations

List of Verbs

- Identify
- Promote
- Challenging
- Discover
- Catalog
- Explore
- Compare
- compile
- Recommend
- Encourage

List of Nouns

- Cafe
- Desserts
- Drinks

- Menu
- Database
- Community
- Location

Cafe

- cafe_id: INT
- cafeName: TEXT
- cafeType: TEXT
- cafe_description: TEXT

Location

- location_id: INT
- country : TEXT
- city : TEXT
- neighborhood : TEXT

Menu

- menu_id: INT
- menu_name: TEXT
- seasonal: TEXT

Noncaf Drinks

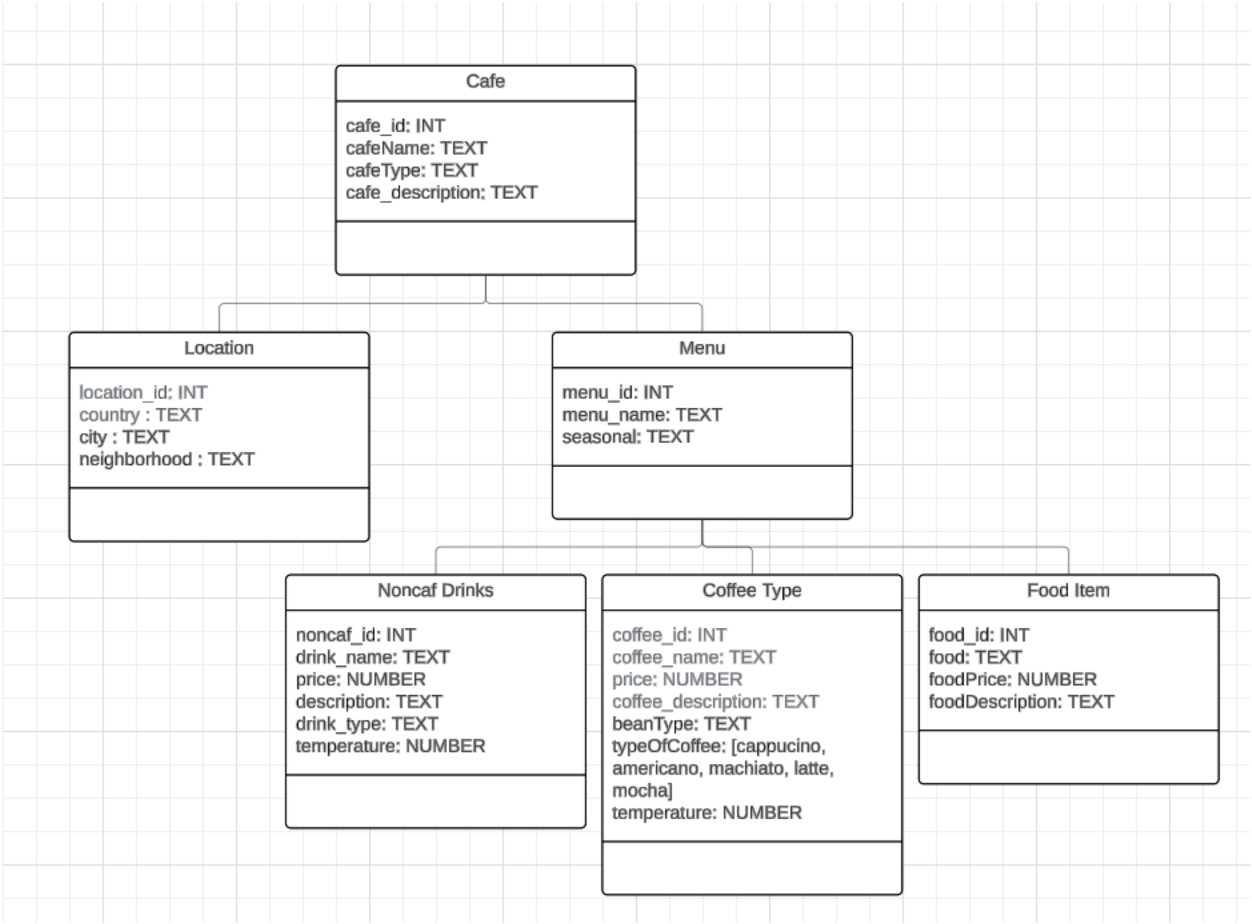
- noncaf_id: INT
- drink_name: TEXT
- price: NUMBER
- description: TEXT
- drink_type: TEXT
- temperature: NUMBER

Coffee Type

- coffee_id: INT
- coffee_name: TEXT
- price: NUMBER
- coffee_description: TEXT
- beanType: TEXT
- typeOfCoffee: [cappucino, americano, machiato, latte, mocha]
- temperature: NUMBER

Food Item

- food_id: INT
- food: TEXT
- foodPrice: NUMBER
- foodDescription: TEXT



2. (30 pts) Describe the Redis data structures that you are going to use to implement the functionalities you described in the previous point. (example To implement the most viewed products I will use a Redis sorted set with key "mostViewed:userId", product ids as the values and a score of the number of views of the product.). You can use/describe more than one data structure, you will need to implement at least one.

I’m sorting the coffee by price.
Sorted set (coffee.price, coffee_id)
Sorted set with key “coffeePrice:cafe_id”

3. (30 pts) The redis commands that you would use to interact with your specific Redis structures.

Example: I will keep a sorted set with the most viewed products in my application. Therefore I need:

Describe the commands for all your use cases. Make sure to include all CRUD operations

- **Initialize:**
  FLUSHALL
- **Make a sorted set for the cafe_id of 1 that adds coffee_id of 7 with its price of $5.**
  ZADD coffeePrice:1 5 7
- **I am reading the top 5 expensive coffees from cafe_id of 1**
  ZRANGE coffeePrice:1 0 5 REV
- **I am changing the price of coffee_id of 7 to $8**
  ZADD coffeePrice:1 7 8
- **I am deleting coffee_id of 7**
  ZREM coffeePrice:1 7