Scenario:
You need to build a backend system for a restaurant. Building a proper backend system with RESTful APIs is required for this project. UI is not required. The project is explained in detail below. It should cover these two aspects of a restaurant: **Menu** and **Billing**

Food items(Menu)

Add CRUD operations for food menu items. A menu item should have the following properties:

- Id
- Name
- Price
- Image
- Description
- Category
- Type of dish

Example:

id	1
name	Chicken momo
price	190
image	http://www.random-momo-image.url.com/image.png
description	This dish contains chicken wrapped in flour dough which is steamed
category	momo
flavours	Spicy, salty, steamed

Note: A single category can have multiple food items. Example: “momo” category can have chicken momo, buff momo, pork momo
A single dish can have multiple flavours. Also, there will be many dishes under the same flavours as well. Example: chicken momo, buff chowmien can both have salty flavours. Please design the database considering these relationships.

Billing

A bill should have the following properties.

- Bill id
- Bill number
- Date
- Time
- Table no
- Food items with quantity, individual rate, amount and total amount

Bill id	1		**date**	2020-01-01
Bill number	8981		**time**	14:45
Table number	7			
sn	**Food items**	**quantity**	**rate**	**amount**
1	Chicken momo	1	190	190
2	Cheese pizza	2	450	900
3	Cold coffee	2	150	300
4	Chicken thukpa	1	200	200
5	Milk tea	3	60	180
			total	1770
			Service charge	177
			Total after service charge	1947

Note: Please note that the **total**, **service charge** and the **total after service charge** should not be saved in the database and needs to be calculated every time a bill is retrieved based on the amount values which are stored in the table.

Deliverables:

1. A Spring boot application with JPA and MySQL
2. The code should be uploaded to a github repository
3. Provide RESTful APIs for performing CRUD operations for the above mentioned scenario.
4. MySQL database structure with all the required tables. Should also incorporate one-to-many and many-to-many relationships shown above.
5. An api to list all the food items (for menu purposes)
6. An api to get a particular bill with **total**, **service charge** and the **total after service charge.** Example: **localhost:8080/bill/1** should provide a bill with all the **food items, quantity,amount,service charge, total** associated with **bill id 1**