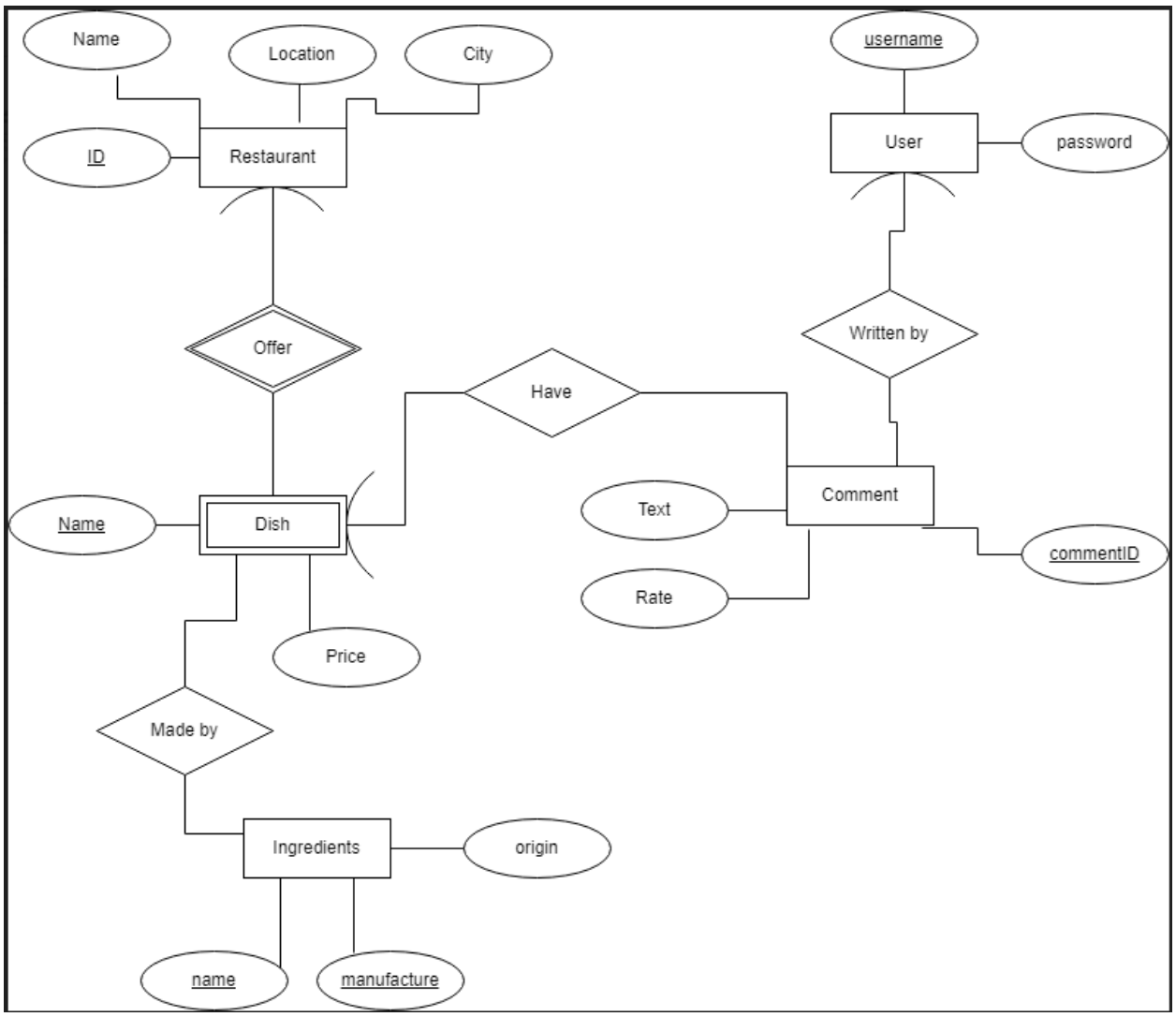
ER DIAGRAM:



RELATION SCHEMA:

Table-Restaurant(ID:INT [PK], Name:VARCHAR(255), Location:VARCHAR(255),City: VARCHAR(100))

Table-Dish(Name:VARCHAR(255), Restaurant:INT [FK to Restaurant.INT], Price: INT, (Name, Restaurant)[PK])

Table-Ingredients(name: VARCHAR(255), manufacture: VARCHAR(255), origin:VARCHAR(255), (name, manufacture)[PK])

Table-User(username: VARCHAR(255)[PK], password: VARCHAR(255))

Table-Comment(CommentID: INT [PK], Text:VARCHAR(1000), Rate: INT)

Table-Have(DishId: INT, CommentId:INT)

Table-Made by (name: VARCHAR(255), manufacture: VARCHAR(255), DishName:VARCHAR(255))

Table-Written by(username: VARCHAR(255), CommentID: INT)

ASSUMPTION:

Table-Dish:

- Every Dish have a name that is less than 255 characters
- Every restaurant do not have two dish with the same name
- All dishes have integer prices

Table-Restaurant:

- Every restaurant have a name that is less than 255 characters
- Every restaurant locates in a address that can be represented in 255 characters

Table-Ingredients

- No two ingredients can have the same name and are produced by the same manufacturer.
- No manufacturer has the same name.

Table-User:

- All users can create a unique username

Table-Comment

- Rate would be an integer from 0 to 5

DESCRIPTION OF RELATIONSHIP:

Relation - Offer:
A restaurant can offer many dishes, a dish can only be offered by a specific restaurant.

Relation - Have:
A comment can be added on exactly one dish. A dish can have multiple comments.

Relation - Written by:
A comment can be written by exactly one user. A user can write many comments.

Relation - Made by:
An ingredient can be used in multiple dishes and a dish can have multiple ingredients.