Nalongsone Danddank Student ID : 14958950 StarID: jf3893pd

Email: [nalongsone.danddank@my.metrostate.edu](mailto:nalongsone.danddank@my.metrostate.edu)\

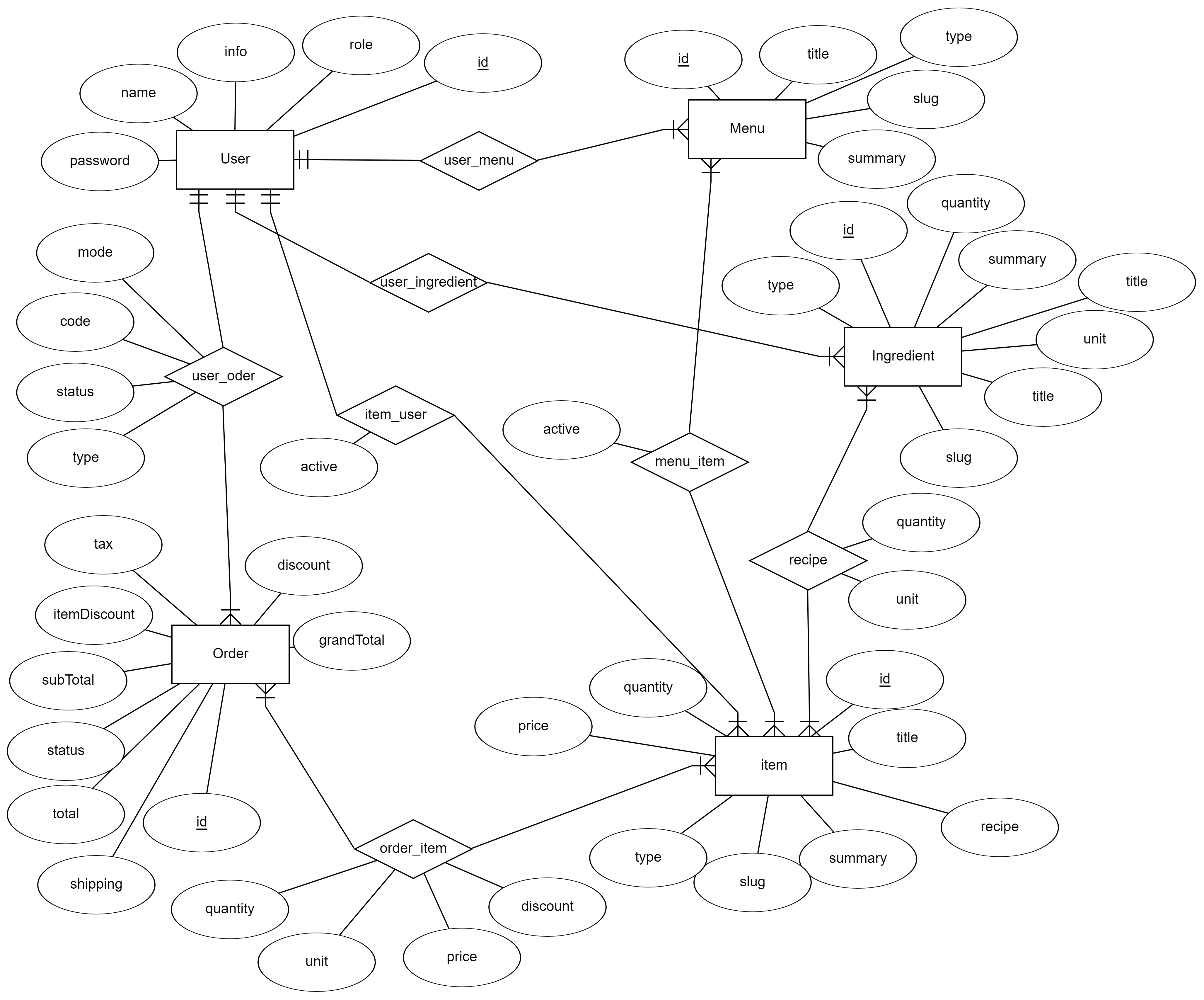
**Metropolitan State University**

**ICS 311 —Database Management Systems**

**Term Project Step 2 : Conceptual Database Design**

**Total: 24 Points**

**Part 1: Entity-Relationship Diagram (ERD) (12 Points):**



1. **User Entity Set** to store user information and used to manage different types of users including admins, chefs, agents, and customers. It can be used to relate the users with Menus, Items, and Orders. Users can track their own orders.
2. **Ingredient Entity Set** is mapped to identify the supplier who can supply the ingredient to re-fill the inventory. There can be a separate table to store the ingredient and supplier relationship to support multiple suppliers for the same ingredient. It uses the columns quantity and unit to track the stock available in the ingredient inventory.
3. **Item Entity Set** is mapped to identify the supplier who can supply the non-cooking item to re-fill the inventory. In a more advanced scenario, there can be a separate table to store the item and supplier relationship to support multiple suppliers for the same item. Similar to Ingredient Table, it uses the columns quantity and unit to track the stock available in the item inventory.
4. **Menu Entity Set** can be used to store the multiple menus of the same restaurant.
5. **Order Entity Set** can be used to store the completed bookings and vendor orders. A logged-in user can also be associated with the order. The vendor orders status can be set to new while placing the order.
6. **Recipe(item\_ingredient) Relationship Set** (item - ingredient: many to many) can be used to track the quantity of the **ingredients** required for an **item** for a single serving.
7. **Menu\_Item** **Relationship Set** (menu - item: many to many) can be used to track the **items** available in the **Menu** Card.
8. **Item\_user Relationship Set** (item -> user: many to one) can be used to identify the **Chef (user’s role)** assigned to cook the **item**.
9. **Transaction (user\_order)** **Relationship Set** (user <- order: one to many) to track the order payments made by the guests to the restaurant and restaurant to the vendors for bookkeeping. We can also use the same table to record the credit(guests) and debit(vendors) transactions.
10. **Order\_item Relationship Set** (Order - Item: many to many) can be set to complete after receiving the items from the vendor. Also, the item price has to be filled manually after receiving the items from the vendor.
11. **User\_ingredient** **Relationship Set** (User <- Ingredient: one to many).
12. **User\_menu Relationship Set** (User <- Menu: one to many).

**Part 2: Relational Schema (12 Points):**

Schema: **Self\_Order\_Foods\_Restaurant**

1. Table: **User**( id <integer, Primary key>, name <varchar>, phoneNum<varchar>, email <varchar, UNIQUE>, password <varchar>, role <varchar>, createAt <varchar>, address <varchar>);
2. Table: **Ingredient**( id <integer, Primary key>, userId<integer, foreign key:User>, type <varchar>, quantity<integer>, summary <varchar>, title <varchar>, unit<integer>);
3. Table: **Item**( id<integer, primary key>, userId<integer, foreign key:User>, title<varchar>, recipe<varchar>, summary<varchar>, slug<varchar>, type<varchar>, price<float>, quantity<integer> );
4. Table: **Order**( id<integer, primary key>, userId<integer, foreign key:User>, shippingFee<float>, total<float>, status<varchar>, subTotal<float>, itemDiscount<float>, tax<float>, discount<float>, grandTotal<float> );
5. Table: **Menu**( id<integer, primary key>, userId<integer, foreign key:User>, title<varchar>, type<varchar>, slug<varchar>, summary<varchar>);
6. Table: **Recipe**(id<integer, primary key>, itemId<integer, foreign key:Item>, Ingr\_id<integer, foreing key: Ingredient>, quantity<integer>, unit<integer>, instructions<varchar>);
7. Table: **Menu\_Item**(id<integer, primary key>, itemId<integer, foreign key:Item>, menuid<integer, foreing key: Menu>, active<integer>);
8. Table: **Item\_user**(id<integer, primary key>, itemId<integer, foreign key:Item>, userid<integer, foreing key: User>, active<integer>);
9. Table: **Transaction “user\_order”** (id<integer, primary key>, orderId<integer, foreign key:Order>, userid<integer, foreing key: User>, code<varchar>, type<varchar>, mode<varchar>, status<varchar>, createAt<datetime>, content<varchar>);
10. Table: **Order\_item** ((id<integer, primary key>, itemId<integer, foreign key:Item>, orderid<integer, foreing key: Order>, price<float>, discount<float>, quantity<integer>, unit<varchar>, createAt<datetime>, content<varchar>);