Task: Design and Implement a Restaurant Ordering System in Golang

Background

As the technical lead of a restaurant, you need to develop a backend system that supports user ordering, kitchen order management, and automatic order status updates. Please use Golang to design and implement this system and provide appropriate APIs for frontend calls.

Functional Requirements

1. **User Ordering**
   - Users can browse the menu (each dish has a name and price).
   - Users can select multiple dishes and submit an order (an order can contain multiple dishes).
   - After submitting the order, it enters the kitchen waiting for processing with an initial status of Pending.
2. **Kitchen Order Processing**
   - Chefs can view all unfinished orders and process the dishes in each order one by one.
   - Each dish's status can be updated (Preparing -> Ready).
   - When all dishes in an order are marked as Ready, the order status automatically updates to Completed.
3. **Order Query**
   - Users can check the status of their orders (including the status of each dish).
   - The kitchen can view all unfinished orders.

Technical Requirements

- Clear interfaces/object-oriented/functional programming/domain-driven design.
- The storage solution can be a simple local map storage or incorporate cache/RDS, etc.
- Support multiple users ordering simultaneously and multiple chefs processing dishes concurrently. (You can use goroutines and channels to simulate kitchen order processing.)
- The final presentation can be in the form of tests, terminal input/output, or a graphical interface (not limited to Golang).

- Good git commit habits are preferred.
- Not all functions need to be completed, but the final program must be able to run.

Submission:

Please package the source code (including the .git folder) into a zip file and send it to hr@n7interactive.com.

Time Requirement:

Please submit the final zip code file within 72 hours of receiving this task.