Evive Engineering Test:

Your solution will be evaluated on object oriented design, testability, extensibility, and solution organization. You can use any language and tooling you would like. Your solution should consist of:

- Source Code
- Tests
- Build Scripts that will allow the application to be run and tested locally from source
- Any dependencies should be detailed in a readme

You can return via zip file or send a link to a GitHub repository.

Menu Ordering System:

Create a system that takes orders for breakfast, lunch, and dinner. Consider the following menus:

Breakfast

Main	Side	Drink
1: Eggs	2: Toast	3: Coffee

Lunch

Main	Side	Drink
1: Sandwich	2: Chips	3: Soda

<u>Dinner</u>

Main	Side	Drink	Desert
1: Steak	2: Potatoes	3: Wine	4: Cake

Rules:

- 1. An order consists of a meal and collection of comma separated item lds.
- 2. The system should return the name of the items ordered
- 3. The system should always return items in the following order: meal, side, drink
- 4. If multiple items are ordered, the number of items should be indicated
- 5. Each order must contain a main and a side
- 6. If no drink is ordered, water should be returned
- 7. At breakfast, multiple cups of coffee can be ordered
- 8. At lunch, multiple sides can be ordered
- 9. At dinner, dessert must be ordered
- 10. At dinner, water is always provided

Sample Input/Output

In: Breakfast 1,2,3

Out: Eggs, Toast, Coffee

In: Breakfast 2,3,1

Out: Eggs, Toast, Coffee

In: Breakfast 1,2,3,3,3

Out: Eggs, Toast, Coffee(3)

In: Breakfast 1

Out: Unable to process: Side is missing

In: Lunch 1,2,3

Out: Sandwich, Chips, Soda

In: Lunch 1,2

Out: Sandwich, Chips, Water

In: Lunch 1,1,2, 3

Out: Unable to process: Sandwich cannot be ordered more than once

In: Lunch 1,2,2

Out: Sandwich, Chips(2), Water

In: Lunch

Out: Unable to process: Main is missing, side is missing

In: Dinner 1,2,3,4

Out: Steak, Potatoes, Wine, Water, Cake

In: Dinner 1,2,3

Out: Unable to process: Dessert is missing