

# Code Review Scribe

**Date:** Oct 13, 2025

**Project:** Restaurant Management System

**Committer:** Phyo

**Reviewer:** Ryan and Grace

**Scribe:** Ko Sett

**Scope:** `src/components/pages/server/index.jsx` (Component: `ServerView`)

**Related Modules:** Cook UI, Server UI

## Summary

This review focused on the **Cook UI and Server UI integration**, especially how order synchronization behaves between both interfaces. The main focus was the **Server UI**, but functional issues found in the **Cook UI** were also discussed as they directly affect workflow and data consistency.

## Context & Functional Findings

### Cook UI issue:

- If a server orders *three identical menu items* (e.g., *Pizza*), the Cook UI shows “**Pizza ×3**” instead of **three separate cards** (*Pizza, Pizza, Pizza*).
- Expected behavior: Cook UI should display *one card per item* for better workflow tracking.

## Restaurant POS

### Kitchen Queue

Table 4

Order #T4-1760369707308

Active

Margherita Pizza x3

Pending

Mark Ready

Cancel

### Server–Cook interaction issue:

- When the server updates an order (e.g., reduces from 3 → 2 items), the system **creates a new order** instead of updating the existing one.
- Example:
  - Original: Order #1 → 3 items
  - Update: 2 items → creates *Order #2*, leaving Order #1 active.
- Expected: Updates should modify the existing order, not create a new one.

#### Order for Table 4



Menu	Current Order
Margherita Pizza \$12.99	Margherita Pizza \$12.99 each 3 \$38.97 Remove
Caesar Salad \$8.50	
Spaghetti Carbonara \$15.00	Margherita Pizza \$12.99 each 2 \$25.98 Remove
Bruschetta \$6.75	
Tiramisu \$7.00	

Total: \$64.95

Cancel Submit Order


### Ingredient constraint scenario:

- If the cook only has ingredients for 2 of 3 pizzas, the workflow should gracefully reflect this (e.g., partial acceptance).

### New Cook UI feature:

- Add “Accept” button:
  - Once the cook clicks *Accept*, the server **cannot cancel** the order.
  - The server can cancel **only until the cook** accepts.
  - If the server cancels before acceptance, the order disappears from the Cook UI.

Issue	Details	Recommendation
File/Component name mismatch	File: <i>index.jsx</i> , Component: <i>ServerView</i>	Rename file to <i>ServerView.jsx</i> or component to <i>Index</i> . One component per file per standard.
Missing doc block	No JSDoc describing component purpose, props, return	Add component documentation comment above <i>ServerView</i> .
Import grouping	React and internal imports not visually separated	Add a blank line after React imports.
Large file (288 lines)	Near limit of 300 lines	Split into smaller components (TablesGrid, ActiveBills, ReadyForPickup).

```
src > components > pages > server >  index.jsx > ...
1  /**
2   * ServerView.jsx
3   *
4   * Server interface for managing tables and orders
5   *
6   * Created by Phyto, 10 October 2025
7   */
8
9  import { useState, useMemo } from 'react'
10 import { useContext } from '../contexts/AppContext'
11 import Button from '../ui/Button'
12 import Card from '../ui/Card'
13 import ModalServer from '../ui/ModalServer'
```

## Code Quality & Structure

- Good naming conventions (*handle\**, *canDeliverBill* clear).
- Move *TOTAL\_TABLES* constant to **module scope** (avoid re-creation on render).
- Extract helper functions (e.g., *calculateOrderTotal*, *canDeliverBill*) into **utils/selectors** module.
- Continue using *useMemo/useCallback* for derived data and handlers.

## UX & Accessibility

Aspect	Issue	Recommendation
Table Buttons	Missing accessibility attributes	Add <i>aria-label</i> (Table #, status), and <i>aria-pressed</i> for selection.
Action Buttons	Unclear labels	Add descriptive labels for <i>Serve</i> and <i>Deliver Bill</i> .
Confirmations	No confirmation on Deliver Bill	Add confirmation dialog + toast feedback.
Disabled State	Not clear why button disabled	Add tooltips or text hints.
ModalServer	Missing Escape and focus handling	Ensure focus trap and return focus to table after close.

## Data & Formatting

- Replace manual formatting `${orderTotal.toFixed(2)}` with shared *formatCurrency* utility.
- Centralize status literals (*Active*, *Ready*, *Served*) in shared constants.

```
const orderTotal = calculateOrderTotal(order.items)
const itemCount = order.items?.length || 0

return (
  <Card key={order.id} className="p-4">
    <div className="flex items-center justify-between">
      <div>
        <div className="font-semibold text-lg">Table {order.tableNumber}</div>
        <div className="text-sm text-gray-600">
          {itemCount} item{itemCount !== 1 ? 's' : ''} | Total: ${orderTotal.toFixed(2)}
        </div>
      </div>
    </div>
  </Card>
)
```

## Standards & Enhancements

- Add blank lines between import groups (React, hooks, context, UI, modals).
- Add header note: “Deliver Bill” is only enabled when all items are marked *Served*.
- Optional:
  - Show per-table item counts (ready/in-progress).
  - Group ready items by table and item type.
  - Prepare for i18n (centralize user-facing strings).
  - Memoize extracted subcomponents for performance.