

List of updates

High-Priority Updates

These features will enhance core functionality and user experience immediately.

1. Include OrderItems in Get Table/Get All Tables

- **Goal:** Ensure the response for table-related endpoints includes associated `OrderItems` for better visibility.
- **Implementation:**
 - Use **eager loading** (`Include` in Entity Framework) to fetch related `OrderItems`.
 - Ensure `DTOs` are structured to include this data without overloading the response.

2. Table Update Response with Order

- **Goal:** When a table is updated (e.g., adding orders), return the table along with the associated `Order` data.
- **Implementation:**
 - Update API responses to include nested `Order` data.

3. Authentication

- **Goal:** Secure the application by implementing authentication for waiters, chefs, and admins.
- **Implementation:**
 - Use **JWT** for token-based authentication.
 - Assign roles (e.g., `Waiter`, `Chef`, `Admin`) to control access.

4. Error Handling Update

- **Goal:** Improve error messages and ensure consistent handling across the app.
- **Implementation:**
 - Use middleware for centralized exception handling.
 - Standardize error response structure with meaningful messages.

5. Upload Images for Menu Items

- **Goal:** Allow users to upload and view images for menu items.
 - **Implementation:**
 - Store images in the file system or cloud (e.g., AWS S3, Azure Blob Storage).
 - Add a column in the `MenuItem` entity for image URLs.
-

Mid-Priority Updates

These features add significant functionality but are not immediately critical.

6. Ingredients with Price / Additional Ingredients for OrderItems

- **Goal:** Allow items to have customizable ingredients with extra charges.
- **Implementation:**
 - Create an `Ingredient` entity with `Name`, `Price`, and relationships to `OrderItem`.

7. Table with Areas

- **Goal:** Categorize tables into areas (e.g., "Indoor", "Outdoor").
- **Implementation:**
 - Add an `Area` entity and associate it with `Table`.

8. Table with Reservation for Date and Time

- **Goal:** Enable customers to book tables in advance.
- **Implementation:**
 - Add `Reservation` entity with `TableId`, `DateTime`, and `CustomerInfo`.

9. OrderItem Size for Pizza

- **Goal:** Allow pizzas to have sizes with different prices.
- **Implementation:**
 - Add a `Size` property or a related entity to `OrderItem`.

10. Enable Menu Updates History

- **Goal:** Track changes made to menu items.
- **Implementation:**
 - Create a history table or use versioning to track updates.

Low-Priority / Future Features

These features enhance scalability and advanced use cases.

11. Ordering for Takeaway/Delivery (with Map)

- **Goal:** Support takeaway and delivery orders with map integration for delivery.
- **Implementation:**
 - Add `DeliveryAddress` and `Takeaway` fields to `Order`.
 - Use APIs like Google Maps for delivery location tracking.

12. Create Reports for Daily Sales

- **Goal:** Provide insights into daily revenue and order statistics.

- **Implementation:**

- Aggregate sales data by date and generate summaries.

13. Offline Sync

- **Goal:** Allow offline order-taking with synchronization upon reconnection.

- **Implementation:**

- Use local storage on the frontend and a sync mechanism on the backend.

14. Multi-Branch Support

- **Goal:** Expand app functionality to support restaurant chains.

- **Implementation:**

- Add a `Branch` entity and associate it with tables, orders, and users.