Documentation

Design Patterns

Singleton Pattern

used for the RestaurantConfig

service to ensure only one instance of the configuration (tax rate, currency) exists throughout the application lifecycle.

File:

Services/Configuration/RestaurantConfig.cs

Factory Method Pattern

Used to dynamically create payment processors (Stripe or PayPal) based on user selection, without exposing the instantiation logic to the client.

File:

Services/Factory/PaymentFactory.cs

Usage:

Used to create instances of

StripeAdapter

or

PayPalAdapter

C# Language Features

Async / Await

Implemented to handle operations without blocking the main execution thread, useful for database access and simulated payment processing.

Example: GenericRepository.GetAllAsync() retrieves data asynchronously.

Example: StripeAdapter.ProcessPaymentAsync() simulates an asynchronous payment call.

LINQ (Language Integrated Query)

Used in the

ReportingService

to filter, sort, and group data efficiently.

File:

Services/Reporting/ReportingService.cs

Features:

GetDailySalesAsync

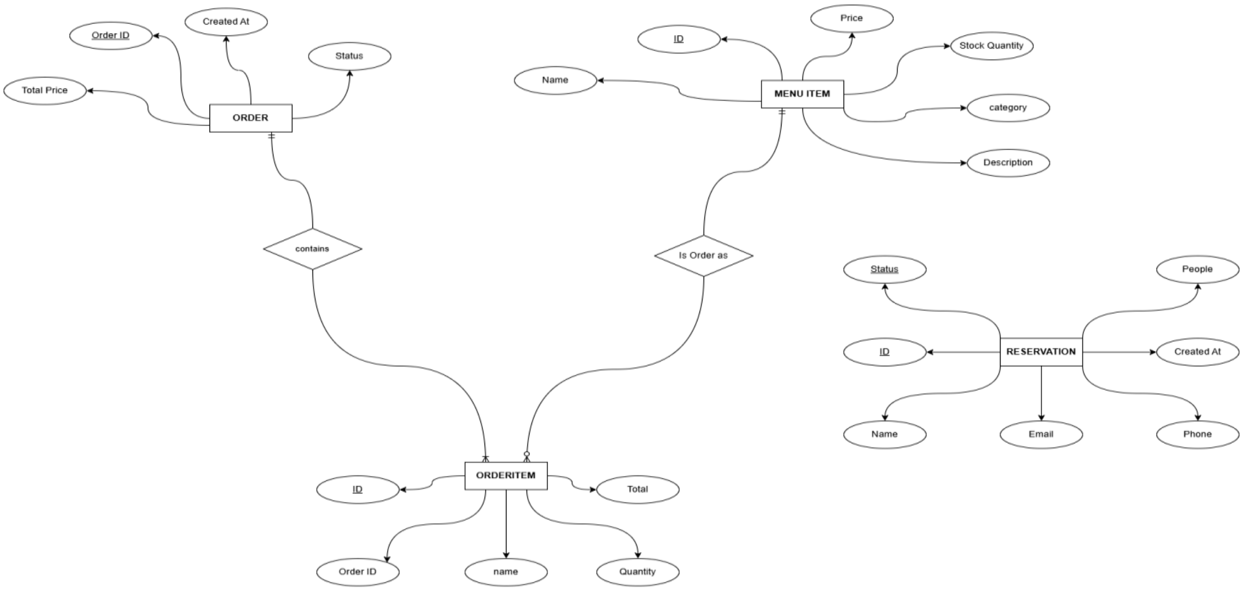
: Uses .Where() and .Sum().

GetTopSellingItemsAsync

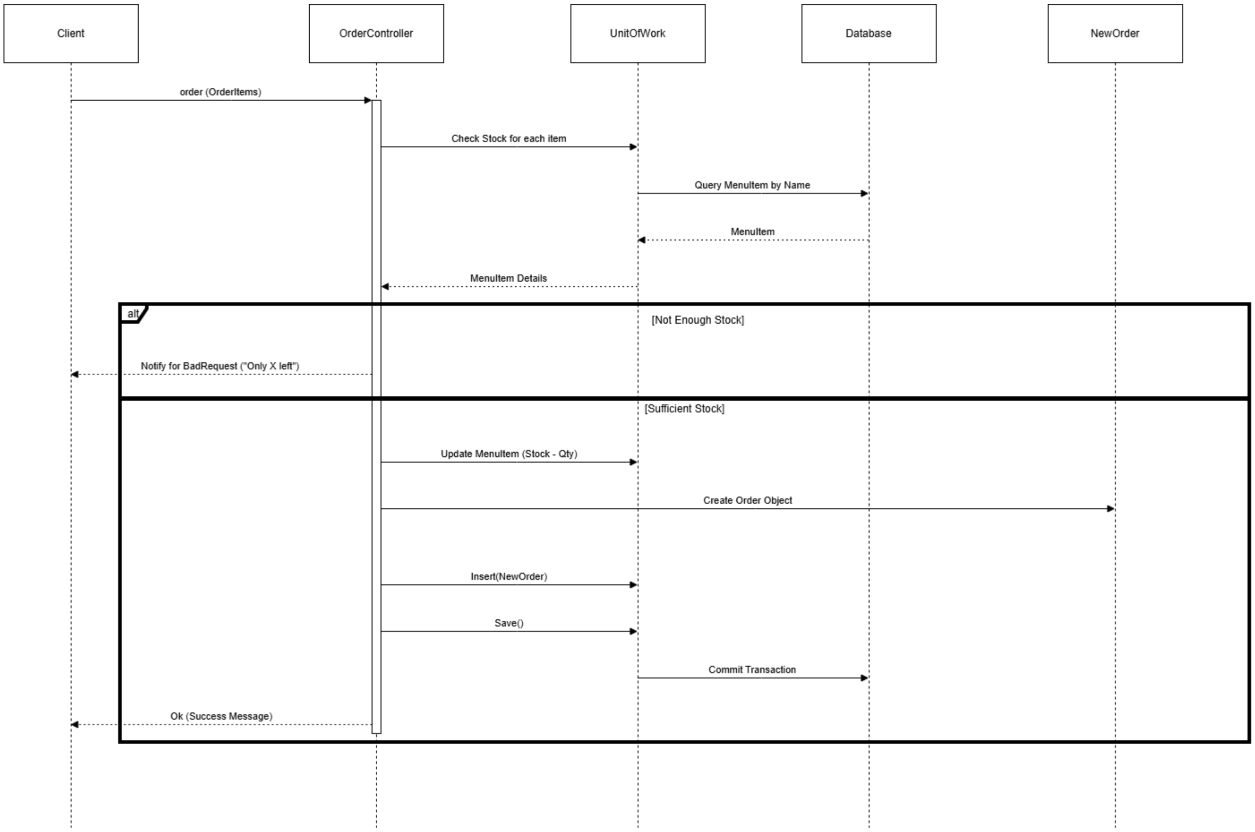
: Uses .GroupBy(), .Select(), and .OrderByDescending().

Diagrams

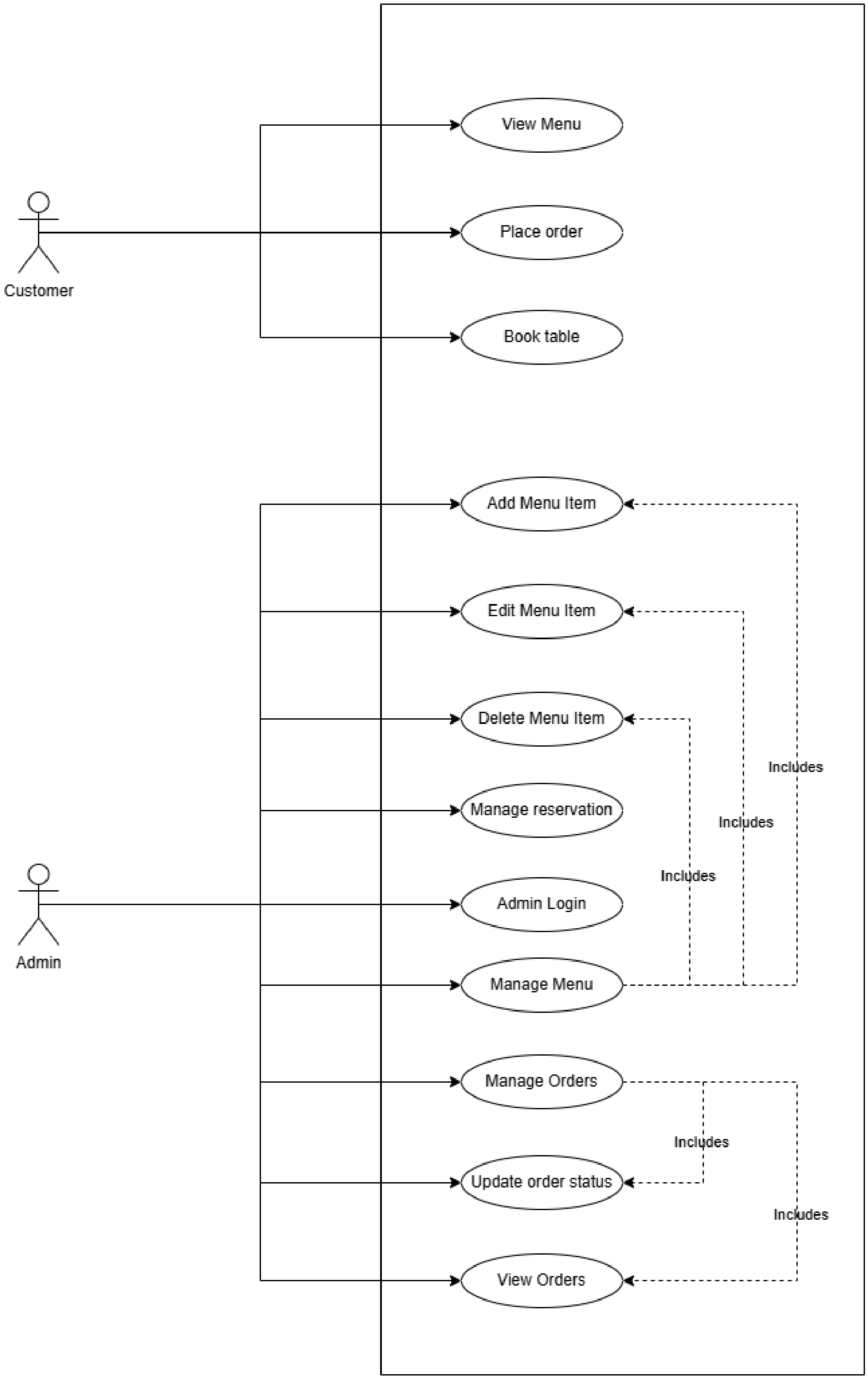
ERD :



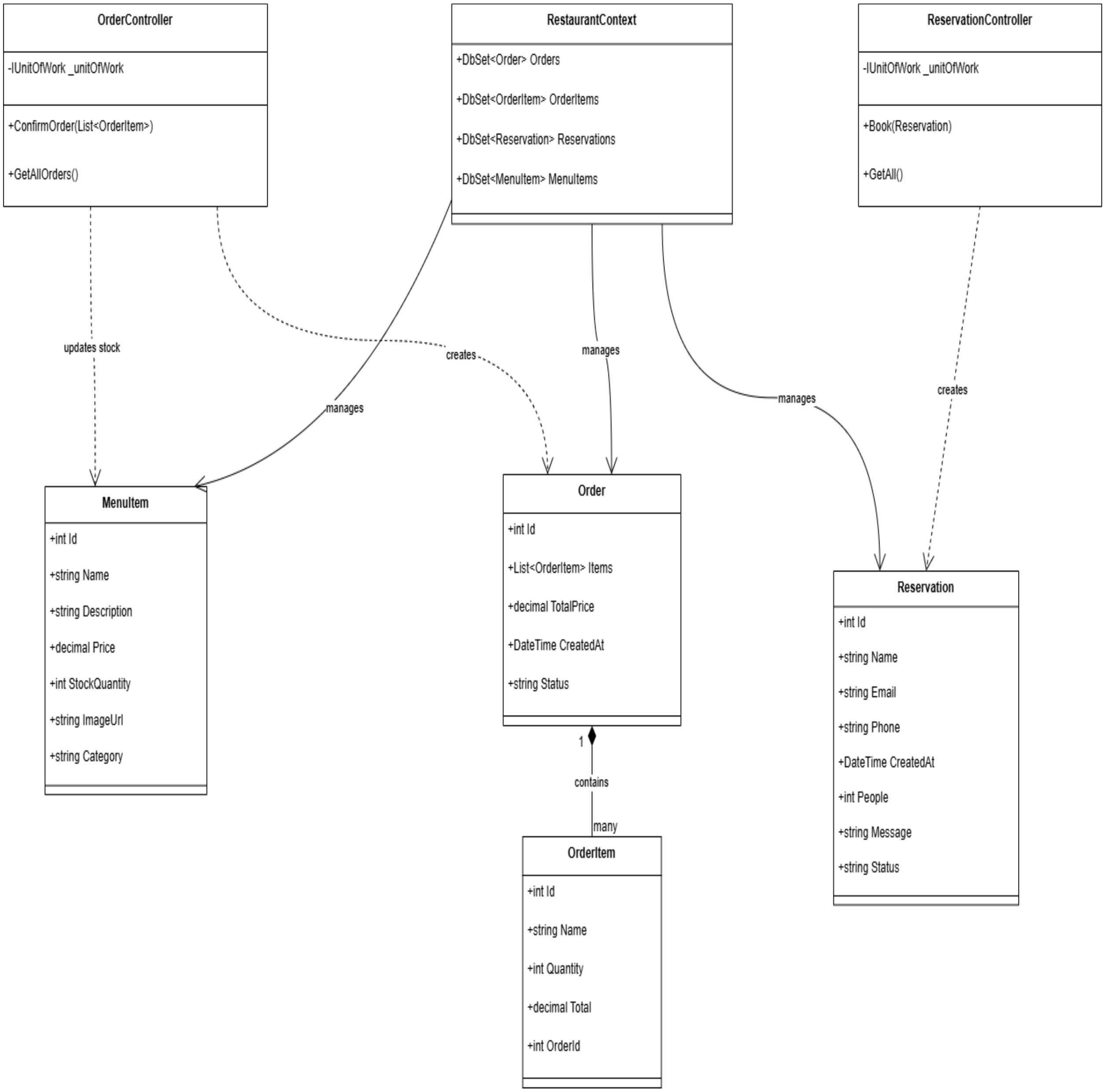
sequence Diagram :



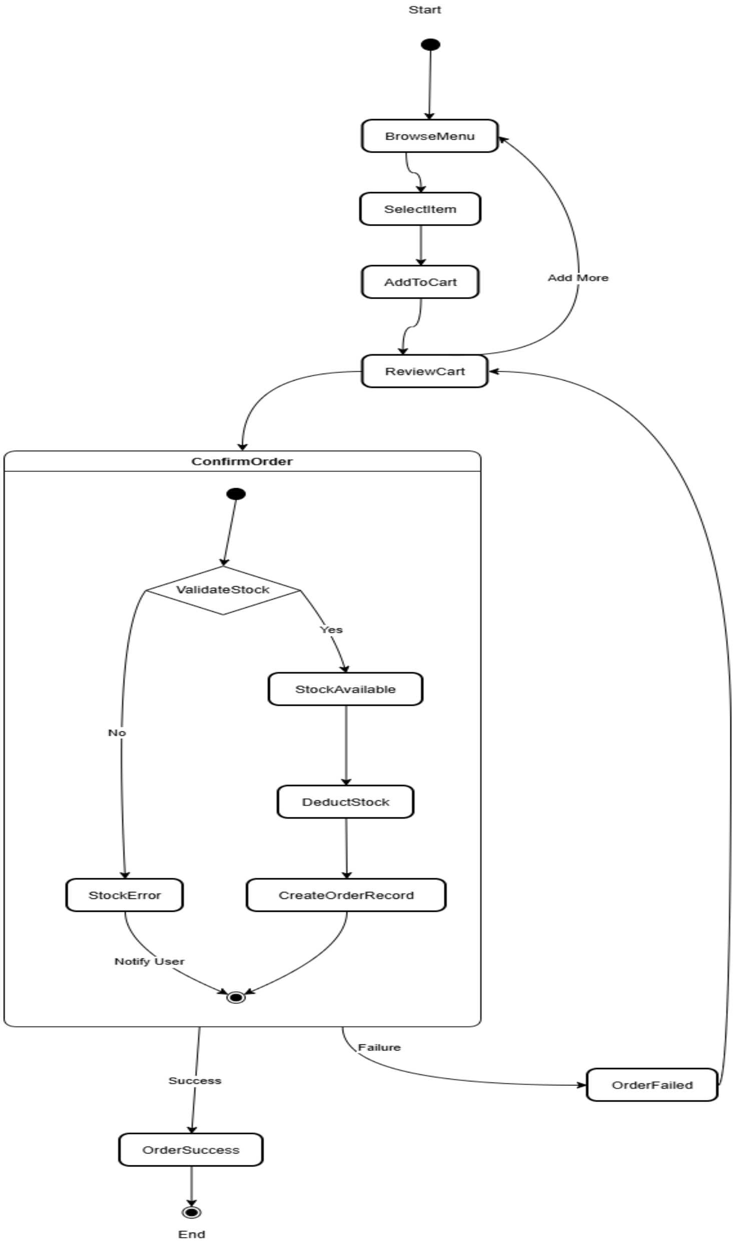
Use Case Diagram :



Class Diagram :



Activity Diagram:



Source Code (GitHub)

Go To GitHub Repo

Team

- **Muhammed Awad farag**
- **Ahmed Ayman tayel**
- **Omar Muhammed Mahmoud**