



Low Level Design Document

Introduction

This Low Level Design (LLD) document outlines the core backend architecture for **QuickServe - Restaurant Order API**. The system manages restaurant orders, menu items, and user authentication using Node.js, Express, and MongoDB.

1. System Components

Component	Description	Key Responsibilities
API Server	Express-based HTTP server	Route requests, handle middleware
Auth Module	JWT-based authentication	User login, registration, auth check
Menu Module	Menu CRUD operations	Manage menu items
Order Module	Order CRUD and tracking	Place, update, track orders
Database Layer	MongoDB via Mongoose	Data persistence, schema validation

2. Class/Interface Overview

Class/Interface	Description	Key Methods/Attributes
UserController	Handles user endpoints	register() , login() , authCheck()
MenuController	Menu item endpoints	createItem() , getItems() , updateItem() , deleteItem()
OrderController	Order endpoints	placeOrder() , getOrders() , updateOrderStatus()
User (Model)	User schema/model	username , passwordHash , role
MenuItem (Model)	Menu item schema/model	name , price , description , available
Order (Model)	Order schema/model	userId , items , status , createdAt

Relationships:

- Order references User and MenuItem
- Controllers interact with models via Mongoose

3. Data Structure Overview

Model	Fields
-------	--------



User	<code>_id</code> , <code>username</code> , <code>passwordHash</code> , <code>role</code> (enum: 'admin', 'user')
MenuItem	<code>_id</code> , <code>name</code> , <code>price</code> , <code>description</code> , <code>available</code> (bool)
Order	<code>_id</code> , <code>userId</code> , <code>items</code> (array of {menuItemId, quantity}), <code>status</code> (enum: 'pending', 'preparing', 'served'), <code>createdAt</code>

4. Algorithms / Logic

Order Placement Flow (Pseudocode):

```
function placeOrder(userId, items) {  
  if (!userAuthenticated(userId)) return error('Unauthorized');  
  if (!validateItems(items)) return error('Invalid menu items');  
  createOrder({ userId, items, status: 'pending', createdAt: now() });  
  return success('Order placed');  
}
```

Authentication Flow:

- On login, verify credentials, issue JWT token.
- Protect endpoints with JWT middleware.

5. Error Handling

Scenario	Handling Approach
Invalid credentials	Return 401 Unauthorized
Unauthorized access	Return 403 Forbidden
Resource not found (menu/order)	Return 404 Not Found
Validation errors (input/schema)	Return 400 Bad Request
Database errors	Return 500 Internal Server Error

End of Document