



## System Architecture Overview

The system is designed following a layered architecture to ensure scalability, maintainability, and clear separation of concerns.

### Presentation Layer

This layer includes the Customer UI and Admin/Cashier UI, both implemented using React. Users interact with the system through these interfaces. All requests from the UI are sent to the backend via HTTPS REST APIs.

### Application Layer

The Application Layer is the core of the system and is implemented using a Node.js backend API.

Authentication and authorization are handled using JWT to ensure secure access.

The backend contains several internal services, including Menu Service, Order Service, and Status Service, which encapsulate the main business logic of the system.

### Data Layer

The system uses PostgreSQL as the primary database. The backend performs SQL read/write operations to persist and retrieve data related to menus, orders, and order statuses.

### External APIs

The backend integrates with external services through HTTPS:

- Map API for location and navigation support
- Notification API (Email/SMS/Push) to notify users about order status changes, triggered via events
- Payment Gateway (planned for future integration)

This architecture allows the system to be modular, extensible, and easy to evolve as new features or integrations are introduced.