Thali Book - API Design Document

# 1. Project Overview

Thali Book is a full-stack restaurant reservation platform that allows customers to search for restaurants, book tables, view reviews, and manage their reservations. Restaurant Managers can manage their listings and availability, while Admins can approve or remove listings and view analytics.

# 2. Architecture Overview

The system is built using Java Spring Boot (backend), React (frontend), and PostgreSQL (database). It uses a layered architecture with controllers, services, repositories, and entities. Role-based access is enforced via JWT authentication.

# 3. User Roles

- Customer: Can search, book, cancel reservations, and review restaurants.

- Restaurant Manager: Can manage listings, availability, and view bookings.

- Admin: Can approve/remove restaurants and view analytics.

# 4. API Endpoints

## 4.1 Customer APIs

* POST /api/auth/register – Register as a customer
* POST /api/auth/login – Login and receive JWT token
* GET /api/restaurants/search – Search restaurants by date, time, people, city/state/zip
* GET /api/restaurants/{id} – Get restaurant details
* POST /api/bookings – Book a table
* GET /api/bookings/my – View customer bookings
* DELETE /api/bookings/{bookingId} – Cancel a booking
* POST /api/reviews – Submit a review
* GET /api/reviews/restaurant/{restaurantId} – Get reviews for a restaurant

## 4.2 Restaurant Manager APIs

* POST /api/restaurants – Add a new restaurant
* PUT /api/restaurants/{id} – Update restaurant details
* PUT /api/restaurants/{id}/availability – Update table sizes and times
* GET /api/restaurants/my – View manager's restaurants
* GET /api/restaurants/{id}/bookings – View bookings
* GET /api/restaurants/{id}/reviews – View reviews

## 4.3 Admin APIs

* GET /api/admin/restaurants/pending – List unapproved restaurants
* PUT /api/admin/restaurants/{id}/approve – Approve restaurant listing
* DELETE /api/admin/restaurants/{id} – Remove a restaurant
* GET /api/admin/analytics – View reservation analytics
* GET /api/admin/bookings/stats – View booking statistics

# 5. Security

JWT-based authentication is used to secure APIs. Each JWT contains user ID, email, and role. Access is controlled via role-based guards at the controller level in Spring Boot.

# 6. Error Handling & Validation

Spring Boot's @Valid annotation will be used to validate input DTOs. A global ExceptionHandler will return JSON responses with status codes and error messages.

# 7. JSON Response Format

All endpoints return JSON. Standard success responses contain `status`, `message`, and `data` fields. Errors contain `status`, `error`, and `message`.