Trip Planning Application - Architecture & Implementation Guide

1. Overview

A trip planning application where administrators create trips, publish to a website, and users join these trips. Notifications are sent automatically before the trip starts, and when the organizer starts the journey, real-time updates are sent to all participants.

2. Architecture

- Frontend: React

- Backend: Spring Boot REST API

- Database: MySQL/PostgreSQL

- Notifications: Email, Push, SMS

- Scheduler: Automatic reminder before trip date

- Optional: WebSocket for live updates

3. Database Design

Tables:

- Trip(id, name, description, startDate, status)

- User(id, name, email, phone)

- TripParticipants(id, tripId, userId)

4. API Endpoints

Trips:

- POST /api/trips

- GET /api/trips

- GET /api/trips/{id}

- PUT /api/trips/{id}/start

Users:

- POST /api/trips/{id}/join

- GET /api/trips/{id}/members

5. Notification System

- Email via Spring Mail

- Push using Firebase FCM

- SMS via Twilio

- Scheduler sends reminders 24 hrs before trip

6. React Frontend Components

- TripList

- TripDetails

- JoinTripButton

- Pages: Home, TripPage

- Fetching via tripApi.js/userApi.js

7. Optional Real-Time Updates

- Spring Boot WebSockets or SSE

- React WebSocket client connects to receive live status

This document summarizes the backend REST implementation, frontend design, and notification approach for the Trip Planning Application.