Technical Documentation: Bulk Notification System (BMP)

1. System Overview

The Bulk Notification System is designed to allow administrators to broadcast messages across multiple channels (Email, Push, WhatsApp) to a large user base. The architecture is built for high availability and reliability, ensuring that sending 10,000+ messages does not cause system downtime or trigger SPAM filters.

2. Operational Workflow (5-Step Wizard)

The Admin follows a linear 5-step process to initiate a broadcast:

1. **Message Composition:** Admin inputs the notification Title and Body. No pre-saved templates are used; content is created fresh for each campaign.
2. **Audience Targeting:** Admin selects the recipient group (e.g., "All Users" or specific segments). The system performs a count check to display the total number of potential recipients.
3. **Channel Selection:** Choice of delivery method—Email (SMTP), Push Notification (FCM), or WhatsApp.
4. **Final Review:** A summary screen displaying the message, target count, and selected channels for final verification.
5. **Execution & Monitoring:** Upon clicking **"Send Now"**, the system triggers the background process and displays a live progress bar with Success/Failure statistics.

3. High-Scale Architecture (Handling 10,000+ Users)

To handle large volumes without crashing the server or being blocked by SMTP providers, the system implements the following technical strategies:

3.1 Asynchronous Queue Processing

The system uses a **Producer-Consumer pattern** via a Redis-backed queue (e.g., BullMQ, Sidekiq, or RabbitMQ).

- **The Producer (API):** When the Admin clicks "Send," the API immediately creates a "Notification Job" entry in the database and pushes the task to the queue. The API then returns a 202 Accepted status to the UI.

- **The Consumer (Worker):** Background workers pick up the job and process it independently of the web server.

3.2 Database Chunking

To prevent high memory (RAM) consumption, the system does not load 10,000 users into memory at once. It uses **Chunking Logic** to fetch and process users in batches of 100.

3.3 Throttling & Rate Limiting

To protect domain reputation and comply with SMTP/Service provider limits:

- **Batching:** Messages are sent in small bursts.
- **Delay (Throttling):** A configurable delay (e.g., 1 second) is introduced between batches to prevent the IP address from being flagged for suspicious activity.

3.4 Real-time Progress Tracking

The Frontend (Screen 7) maintains a live connection with the Backend to show progress:

- **Success Count:** Number of messages successfully handed off to the provider.
- **Failure Count:** Number of messages that failed due to invalid addresses, network errors, or provider rejection.

4. Key Performance Metrics

- **Fault Tolerance:** If a worker fails, the job can be resumed from the last processed ID stored in the Database.
- **Non-Blocking:** The Admin Panel remains fully functional while the notification sends in the background.
- **Scalability:** Multiple workers can be spun up to increase the sending speed if the user base grows beyond 100,000.

6. Sequence Diagram: Bulk Sending Logic