# Product Requirements Document (PRD) - Notification Service for Ticket Management System

## 1. Title

Notification Service - Ticket Management System

## 2. Purpose

The Notification Service is designed to handle all user-facing communications related to Ticket events in a Ticket management system. These include Ticket assignments, due date reminders, status updates, and comments. The service aims to enhance user engagement and Ticket compliance.

## 3. Scope

This service will support:

* Real-time and scheduled notifications
* Delivery via Email, SMS, and In-App
* Personalization of message content
* User preference management
* Notification status tracking and retries

## 4. Target Audience

* End Users (Ticket assignees, creators)
* Administrators and Project Managers
* Developers and QA teams

## 5. Assumptions

* Users have verified contact information
* External messaging services (SMTP, SMS gateway) are available
* The system is event-driven with a reliable messaging queue

## 6. Functional Requirements

#### 6.1 Event Consumption

* The system shall subscribe to Ticket-related events via Kafka/RabbitMQ.

#### 6.2 Notification Dispatch

* The system shall support sending notifications through Email, SMS, and In-App channels.

#### 6.3 User Preferences

* The system shall store and retrieve user notification preferences (opt-in/out for channels).

#### 6.4 Template Management

* Templates shall support dynamic placeholders (e.g., Ticket name, due date).

#### 6.5 Logging and Auditing

* Each notification event shall be logged with status and timestamps.

#### 6.6 Retry Mechanism

* Failed messages shall be retried up to 3 times with exponential backoff.

## 7. Non-Functional Requirements

* **Performance:** Notification should be sent within 5 seconds of receiving the event.
* **Scalability:** Handle 10,000+ events per minute.
* **Availability:** 99.9% uptime with failover support.
* **Security:** Secure APIs and encrypted credentials for external services.

## 8. Constraints

* SMS and Email services may incur third-party costs.
* Template changes must go through version control.
* Notification delivery is subject to external system availability.

## 9. Success Metrics

* Notification delivery success rate > 98%
* User opt-out rate < 5%
* Mean notification delivery time < 2 seconds

## 10. Future Enhancements

* Push notifications (mobile)
* Multi-language template support
* Analytics dashboard for notification KPIs

## 11. Sequence Diagram (Plain UML)

A diagram of a company

AI-generated content may be incorrect.

Fig 1: Notification Sequence Diagram

## 12. Class Diagram (Plain UML)

A diagram of a computer

AI-generated content may be incorrect.

Fig 2: Notification class Diagram

## 13. Data-flow Diagram

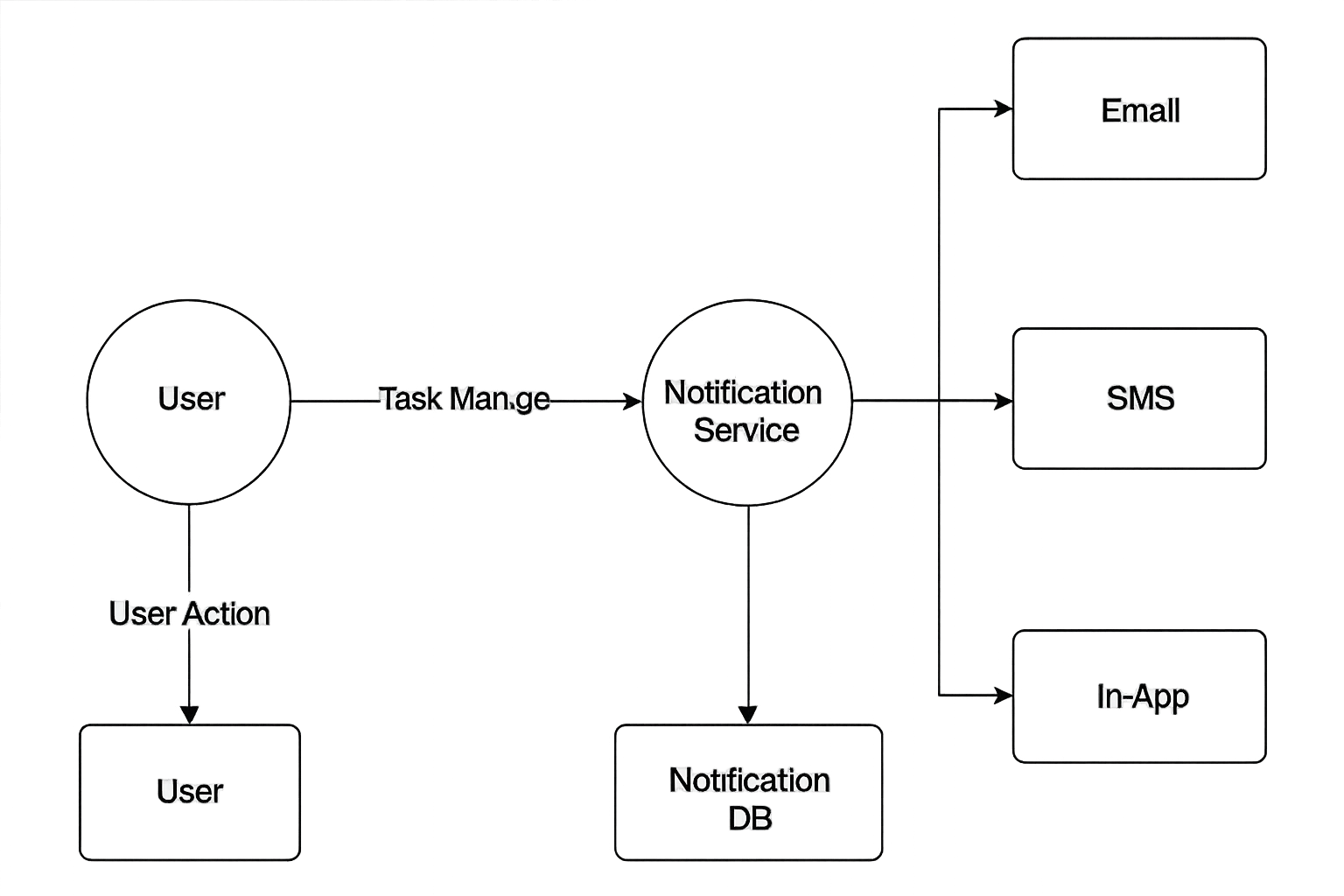


Fig 3: DFD- Diagram

## 14. Database Schema Design

## Entities & Relationships

### 1. users

| **Field** | **Type** | **Description** |
| --- | --- | --- |
| user\_id | UUID / INT | Primary Key |
| First\_name | Varchar | Not null |
| Last\_name | varchar | Not null |
| email | VARCHAR | Email address |
| phone\_number | VARCHAR | For SMS notifications |
| created\_at | TIMESTAMP | Account creation timestamp |

### 2. user\_preferences

| **Field** | **Type** | **Description** |
| --- | --- | --- |
| preference\_id | UUID / INT | Primary Key |
| user\_id | UUID / INT | Foreign key → users(user\_id) |
| email\_enabled | BOOLEAN | Opt-in for Email |
| sms\_enabled | BOOLEAN | Opt-in for SMS |
| in\_app\_enabled | BOOLEAN | Opt-in for In-App |
| updated\_at | TIMESTAMP | Last modified timestamp |

### 3. notification\_templates

| **Field** | **Type** | **Description** |
| --- | --- | --- |
| template\_id | UUID / INT | Primary Key |
| channel | ENUM | 'EMAIL', 'SMS', 'IN\_APP' |
| template\_name | VARCHAR | Logical name of the template |
| content | TEXT | Template body with placeholders |
| created\_at | TIMESTAMP | Creation time |
| updated\_at | TIMESTAMP | Last update time |

### 4. notifications

| **Field** | **Type** | **Description** |
| --- | --- | --- |
| notification\_id | UUID / INT | Primary Key |
| user\_id | UUID / INT | Foreign key → users(user\_id) |
| template\_id | UUID / INT | Foreign key → notification\_templates |
| channel | ENUM | Notification type (EMAIL/SMS/IN\_APP) |
| status | ENUM | SENT / FAILED / PENDING / RETRY |
| message\_payload | JSON | Rendered message |
| retries | INT | Number of retries |
| sent\_at | TIMESTAMP | Time it was sent |
| created\_at | TIMESTAMP | Time of creation |

### 5. events\_log

| **Field** | **Type** | **Description** |
| --- | --- | --- |
| event\_id | UUID / INT | Primary Key |
| event\_type | VARCHAR | e.g., "TICKET\_ASSIGNED", "TICKET\_COMMENTED" |
| payload | JSON | Raw Ticket event data |
| received\_at | TIMESTAMP | Time event was consumed |

### 6. Database Scheme

A screenshot of a computer

AI-generated content may be incorrect.

|  |  |  |
| --- | --- | --- |
| Kaushik Nandy | Created | V 1.0 |
|  |  |  |
|  |  |  |