Business Requirements Document (BRD)

Project Info

Project Title: Smart Community Communication System

Prepared By:

Amith Sharma R

Varun Kumar M

Date: April 16, 2025

Executive Summary

In residential communities, communication often relies on outdated tools like physical notices, email threads, or chat groups. These methods are prone to delays, missed messages, and lack formal accountability. The Smart Community Communication System is designed to centralize, secure, and accelerate communication between residents and administrators. It will deliver structured event updates, emergency notifications, and feedback collection in a mobile-friendly, secure platform with real-time capabilities.

Problem Statement

Residents frequently miss important updates such as emergency water shutdowns, community meetings, or safety alerts due to:

- Outdated, fragmented communication tools

- Lack of real-time alerts

- No central repository for events or notices

- No structured emergency alert workflow

These issues create confusion, low engagement, and compromise community safety.

Business Objectives

- Improve communication transparency across all community members.

- Enable admins to manage users, emergency alerts, and events easily.

- Ensure residents receive real-time notifications even when they are offline.

- Provide role-based access and secure identity validation.

Project Scope

The scope includes designing, developing, and deploying a web-based platform accessible to admins and residents. It will feature:

- Registration with community access code

- Admin approval of new users

- Event creation, RSVP system

- emergency alerts

- Multi-channel notification system (In-App, SMS)

- Feedback system

- Admin dashboard for monitoring and management

Critical Features (To be delivered in 3 weeks)

Week 1:

- Secure registration and login with JWT authentication

- Admin approval flow for new users

- One user = one verified community

Week 2:

- Event board (Create, View)

- RSVP functionality

- Emergency alert (admin receives notification in real time)

- Admin dashboard for managing events and users

Week 3:

- Notification system (In-App delivery + fallback design)

- Feedback form + admin view

- Notification preferences (basic toggle)

- Final UI polishing + testing + deployment

Functional Requirements

- Register with access code & phone number

- Role-based login (ADMIN/USER)

- Admin approval for access

- Create/view/edit/delete events

- RSVP to events

- Send emergency alerts

- Submit and view feedback

- Send notifications

- View events by category and status

Non-Functional Requirements

- 99.9% uptime for cloud hosting

- Notification delivery < 2 seconds for high-priority messages

- Scalable to multiple communities and 10,000+ users

- Secure (JWT, password encryption, HTTPS)

- Responsive UI/UX for desktop and mobile

- API integration with Firebase, Twilio.

Timeline – MVP Delivery (3 Weeks)

Week 1: Auth module, registration with access code, admin approval flow

Week 2: Event board, RSVP, emergency alert system, admin dashboard

Week 3: Notification preferences, feedback module, final testing & deployment

Post-MVP (Future Enhancements):

- WhatsApp integration

- OAuth (Google/Facebook login)

- Chatbot for resident queries

- Analytics for admin panel

Project Cost Estimate

Component	Estimated Cost (INR)
UI/UX Design (Figma - Free Tier)	₹0
Front end (React + Tailwind)	₹0 (Student built)
Back end (Spring Boot + MySQL)	₹0 (Local or free DB)

Component	Estimated Cost (INR)
Twilio API (SMS demo)	₹500 – ₹1,000
Hosting (Render/Firebase - Free)	₹0
Testing + Deployment	₹0
Total Estimate	₹500 – ₹1,000 INR

Assumptions

- Communities will provide their own access codes and initial admin contact

- Admins will verify users manually using provided phone numbers

- SMS costs are billed externally and not part of hosting

- System will be hosted on a shared cloud environment

Risks & Mitigation

Risk: Delay in admin approval → Mitigation: Add mobile contact and reminder system

Risk: User doesn't receive in-app messages → Mitigation: Implement fallback SMS channel

Risk: Emergency alert overload or spam → Mitigation: Limit Emergency to verified users with cool down time.