**Business Requirements Document (BRD)**

**Project Title:**

Community Management

**1. Problem Statement**

In many residential communities, communication remains dependent on outdated and inefficient methods such as paper notices, word of mouth, or emails. These methods often result in missed updates, misinformation, or delayed communication, causing confusion and disengagement among residents. A centralized, real-time platform is needed to enhance how residents stay informed about events, notices, and other critical updates within the community.

**2. Objectives**

* Create a user-friendly platform to centralize communication in residential communities.
* Deliver real-time notifications and reminders for community events.
* Allow users to search, RSVP, and provide feedback on events.
* Enable residents to personalize their experience via notification settings and tags.
* Provide seamless access through email sign-in and social authentication.

| **Feature** | **Description** |
| --- | --- |
| **Event Board** | Central hub to view all upcoming and past community events. |
| **Notifications** | Alerts about new events, updates, cancellations, and reminders. |
|  |  |
| **Calendar Integration** | Visual monthly/weekly/daily view of scheduled events. |

**3. Critical Features**

**4. Important Features**

| **Feature** | **Description** |
| --- | --- |
| **Search** | Allows users to find events using keywords, tags, or filters. |
| **Upcoming Events** | Highlights near-future events on the homepage/dashboard. |
| **Feedback** | Users can rate and review events to help organizers improve. |
| **Tags** | Events can be categorized using tags for better discoverability. |
| **Notification Settings** | Users can customize how and when they receive updates. |

**5. User Roles**

| **Role** | **Permissions** |
| --- | --- |
| **ROLE\_ADMIN** | Create, edit, delete events; manage users, feedback, and settings. |
| **ROLE\_USER** | View events, RSVP, receive notifications, give feedback. |

**6. Authentication Methods**

* **Sign in with Email**
* **OAuth Integration**: Google, Facebook

**7. Core Entities & Relationships**

| **Entity** | **Description** |
| --- | --- |
| **User** | Resident or admin with access to the platform. |
| **Event** | Community event such as meetings, clean-up drives, or announcements. |
| **Notification** | Message or alert tied to an event or system activity. |
| **RSVP** | User response/confirmation for event attendance. |
| **Feedback** | Comments or reviews from users about an event. |
| **Event Category** | Classification tags (e.g., Safety, Social, Maintenance). |
|  |  |

**Entity Relationship Mapping**

| **Entity 1** | **↔** | **Entity 2** | **Type** |
| --- | --- | --- | --- |
| User | ↔ | Event | One-to-Many |
| Event | ↔ | Notification | One-to-Many |
| Event | ↔ | RSVP | One-to-Many |
| User | ↔ | Feedback | One-to-Many |
| Event | ↔ | Event Category | Many-to-Many |
| User | ↔ | Reminder | One-to-Many |

**8. Microservices Architecture**

| **Microservice** | **Responsibilities** |
| --- | --- |
| **User Management** | Handle authentication, profile, and roles. |
| **Event Management** | CRUD operations for community events. |
| **Notification Service** | Generate and deliver system/user notifications. |
| **Feedback Service** | Manage user reviews and ratings. |
| **Reminder Service** | Create, store, and trigger personalized reminders. |

**9. Non-Functional Requirements**

* **Responsiveness:** Mobile-friendly UI
* **Scalability:** Microservices to support multiple communities
* **Security:** JWT-based authentication, encrypted data
* **Usability:** Simple and intuitive design for all age groups
* **Reliability:** High uptime and fault-tolerant services