

# Community Event Organizer App: Requirements Document

## 1. Core Functionalities

- **Event Creation:**
  - Title, description, location, date, and time.
  - Category (e.g., social, cultural, sports, etc.).
  - Image or banner upload.
  - Capacity limit.
  - Ticket pricing and payment options.
- **Event Listing:**
  - Filter and search options (category, location, date).
  - Sort options (date, popularity, etc.).
  - Detailed event information.
- **Event Registration:**
  - User registration/login.
  - Ticket purchase and payment.
  - Waiting list option.
- **Event Calendar:**
  - Visual calendar to display upcoming events.
  - Reminders and notifications.
- **Community Forum:**
  - Discussion boards for event-related topics.
  - Commenting and liking features.
- **Push Notifications:**
  - Real-time updates on event changes, reminders, and important announcements.

## 2. Non-Functional Requirements

- **Performance:**
  - Fast load times and responsive UI.
  - Efficient search and filtering.
- **Security:**
  - Secure user authentication and data encryption.
  - Protection against cyber threats.
- **Usability:**
  - User-friendly interface.
  - Intuitive navigation.
  - Clear and concise information.
- **Scalability:**

- Ability to handle a growing number of users and events.
  - Scalable infrastructure.
- **Accessibility:**
  - Accessible to users with disabilities.
  - Compliance with accessibility standards.

### **3. Requirements Documentation and Tracing**

#### **a. Use Case Diagrams:**

- Visual representation of user interactions with the system.
- Define actors (users) and their goals.
- Identify system responses and alternative flows.

#### **b. User Stories:**

- Short, simple descriptions of a user's desired outcome.
- Format: "As a [user role], I want to [goal] so that [reason]."
  - Example: "As a user, I want to create an event so that I can organize a community gathering."

#### **c. Requirements Traceability Matrix (RTM):**

- Link requirements to design artifacts, test cases, and code.
- Track the implementation and verification of each requirement.

#### **d. Requirements Management Tool:**

- Use a tool like Jira or Azure DevOps to manage requirements.
- Create requirements, link them to tasks and test cases.
- Track changes and version control.

By following this approach, we can ensure that the development team builds the right product and meets the needs of the users.