**App Features**

1. **Club Initialization**:
   * **Functionality**: Allow club administrators to set up their clubs by defining club information, appointing leads and sub-leads, and adding members.
   * **Implementation**:
     + Create forms for entering club details (name, description, etc.).
     + Options to add leads, sub-leads, and members by selecting from registered users.
     + Store club information in MongoDB.
2. **Recruitment**:
   * **Functionality**: Enable clubs to recruit new members through application forms.
   * **Implementation**:
     + Create customizable application forms for each club.
     + Allow users to apply to clubs.
     + Store applications in MongoDB and notify club leads/sub-leads.
3. **Calendar**:
   * **Functionality**: Provide a calendar to schedule and view events.
   * **Implementation**:
     + Integrate a calendar widget (e.g., flutter\_calendar\_carousel).
     + Allow club leads to create, update, and delete events.
     + Display events in a calendar view.
     + Store events in MongoDB with proper indexing for efficient retrieval.
4. **Announcements**:
   * **Functionality**: Allow clubs to make announcements that can be public or private.
   * **Implementation**:
     + Create an announcement section in each club’s portal.
     + Allow club leads to post announcements.
     + Provide options to mark announcements as public or private.
     + Notify members of new announcements through in-app notifications.
5. **Member and Leader Profiles**:
   * **Functionality**: Maintain profiles for all members and leaders with relevant information.
   * **Implementation**:
     + Create profile pages for each user.
     + Include information such as name, role, contact details, and club memberships.
     + Allow users to update their profiles.
     + Store profile information in MongoDB.

**Technical Stack and Tools**

1. **Frontend**:
   * **Framework**: Flutter
   * **State Management**: Provider or Bloc
   * **Networking**: HTTP or Dio for API calls
   * **UI Components**: Material Design components, Custom Widgets
2. **Backend**:
   * **Server**: Node.js with Express.js
   * **Database**: MongoDB
   * **Authentication**: JWT (jsonwebtoken)
3. **Testing**:
   * **Frontend**: flutter\_test, mockito
   * **Backend**: jest, supertest

**User Roles**

1. **Administrator**:
   * Full access to all features.
   * Can manage all clubs and users.
2. **Club Lead/Sub-Lead**:
   * Manage their specific club.
   * Create events, announcements, and handle recruitment.
3. **Member**:
   * View their club’s events, announcements, and profiles.
   * Participate in club activities.
4. **Public User**:
   * View public announcements and events.
   * Apply for club membership.

**User Flows**

1. **Club Initialization**:
   * Admin/Lead initializes club → Sets leads and members → Club portal created.
2. **Recruitment**:
   * User views clubs → Applies for membership → Club lead reviews application → Accept/Reject decision.
3. **Event Scheduling**:
   * Lead creates event → Event appears in the club’s calendar → Members get notified.
4. **Announcements**:
   * Lead posts announcement → Notification sent to members → Members view announcement.
5. **Profile Management**:
   * User creates/updates profile → Information stored in MongoDB → Profile visible in club portal.

**Security Considerations**

1. **Authentication & Authorization**:
   * Implement JWT for secure user authentication.
   * Role-based access control to ensure users only access authorized features.
2. **Data Validation**:
   * Validate all inputs on both frontend and backend.
   * Sanitize data to prevent SQL/NoSQL injection attacks.
3. **Data Privacy**:
   * Ensure private announcements and user data are securely stored and accessed.

**Next Steps**

1. **Finalize Requirements and Design**:
   * Confirm feature set and user roles.
   * Create detailed wireframes and UI mockups.
2. **Set Up Development Environment**:
   * Initialize the Flutter project and backend server.
   * Set up MongoDB and configure necessary libraries.
3. **Begin Development**:
   * Start with core functionalities (authentication, club initialization).
   * Implement additional features iteratively, ensuring regular testing and feedback.