Social and Business Calendar and Event Management System Documentation

Project Overview

The system is a full-stack web and mobile application designed to help users create groups (e.g., friends, family, colleagues), organize events (e.g., birthday parties, trips), and manage their calendars (e.g., doctor appointments, business meetings). The application will feature calendar sharing, event invitations, and group management, similar to Google Calendar but with enhanced social and business features.

Architecture Diagram

Description:
- Frontend: Built with React.js for the web application.
- Backend: Node.js/Express.js for handling API requests.
- Database: MongoDB or PostgreSQL for data storage.
- Mobile App: Flutter for cross-platform mobile application.
- Data Visualization: D3.js or Chart.js for interactive charts and graphs.

Example User Stories

- As a user, I want to register and log in so that I can access the application.
- As a user, I want to create and manage groups so that I can organize my contacts.
- As a user, I want to create and manage events so that I can invite others and keep track of important dates.
- As a user, I want to create and manage my calendar so that I can organize my schedule.
- As a user, I want to share my calendar with selected users so that they can see my availability.
- As a business user, I want to schedule and manage jobs and meetings so that I can coordinate with my team.

User Roles and Permissions

1. User:
   - Register and log in.
   - Create and manage groups.
   - Create and manage events.
   - Create and manage calendars.
   - Share calendars with chosen users and groups.

2. Administrator:
   - Oversee all activities within the system.
   - Manage users and roles.
   - Monitor system performance and security.

Database Schema

User Table:
- user_id (Primary Key)
- name

- email
- password
- role (user, administrator)

Group Table:
- group_id (Primary Key)
- name
- description
- user_id (Foreign Key)

Event Table:
- event_id (Primary Key)
- title
- description
- date
- time
- location
- group_id (Foreign Key)
- creator_id (Foreign Key)

Calendar Table:
- calendar_id (Primary Key)
- user_id (Foreign Key)
- event_id (Foreign Key)
- date
- time
- description

Sequence Diagram for User Registration and Event Creation

User Registration:

1. User submits registration form.
2. System validates the input.
3. System creates a new user in the database.
4. System sends confirmation email to user.

Event Creation:

1. User submits event creation form.
2. System validates the input.
3. System creates a new event in the database.
4. System notifies the user of successful event creation.

API Documentation

Authentication:

- POST /api/auth/register: Register a new user.
- POST /api/auth/login: Log in a user.
- GET /api/auth/logout: Log out a user.

User Management:

- GET /api/users: Retrieve a list of users.
- GET /api/users/:id: Retrieve a specific user.
- PUT /api/users/:id: Update a user's information.

- DELETE /api/users/:id: Delete a user.

Group Management:

- POST /api/groups: Create a new group.
- GET /api/groups: Retrieve a list of groups.
- GET /api/groups/:id: Retrieve a specific group.
- PUT /api/groups/:id: Update a group's information.
- DELETE /api/groups/:id: Delete a group.

Event Management:

- POST /api/events: Create a new event.
- GET /api/events: Retrieve a list of events.
- GET /api/events/:id: Retrieve a specific event.
- PUT /api/events/:id: Update an event's information.
- DELETE /api/events/:id: Delete an event.

Calendar Management:

- POST /api/calendars: Create a new calendar entry.
- GET /api/calendars: Retrieve a list of calendar entries.