



Collaboration Application Requirement Specification (1.0)

1. Overview

The **purpose** of developing this application is: -

- A. demonstrate our capability to develop high-quality, secure, scalable applications
- B. serves as an object to demonstrate observability, data analytics, and data science capabilities, which could be integrated through separate initiatives to follow

The **goal** is to develop a modern collaboration application that allows users to sign up using existing social media logins, organize user groups, manage meetings, enable private messaging, and conduct surveys/polls. The system architecture must follow current industry standards and deliver scalable, secure, and maintainable applications. The implementation must demonstrate software engineering best practices such as modeling, design patterns, simplicity, reusability, code quality, documentation, instrumentation, automation, and verification.

As a stretch goal, we will target integrating AI/ML capabilities.

2. Key Functional Requirements

2.1 User Management and Authentication

- I. User Registration/Login:
 - User must first sign up or log in to applications using their existing social media accounts, such as Facebook, Google, Twitter, LinkedIn, Apple, Hotmail, Outlook, etc.
 - If the social media account has MFA enabled, the user will go through MFA during the social login process
 - For Users who prefer to create a new/separate login, a standard login/email/password registration option with Multi-Factor Authentication (MFA) must be used
 - User must complete any required profile settings, such as name, notification settings
- II. Profile Management:
 - System must allow the User to manage the following: -
 - Preference to reuse social media profile data (check box)
 - User name and notification settings
- III. View Users:
 - The system shows a list of all users. (admin only)
 - Allow the list to be filtered by keyword, group name, etc.

2.2 Group and Event Management

- IV. Request Group Creation:
 - Users must provide a group name (requesting approval from the admin)
- V. Request Group Removal:
 - Users must select an existing group name (requesting approval from the admin)
- VI. Approve Create or Remove a Group (by admin):
 - Admin must see a set of requests for group creation on their dashboard
 - Admin must be able to select each of the group requests to approve or remove
- VII. View Groups:
 - The system shows a list of all groups. (Show only groups owned by)
 - Allow the list to be filtered by keyword
- VIII. Invite Group Member:
 - Group owner must be able to request adding a member by picking a social media identity, assigning roles (e.g., admin, member), and permissions
 - The system must send a notification to the new member, with a link for user registration (if not already registered)
- IX. View Group Members:
 - The system allows users to arrive here from VII. (View Groups)
- X. Edit Group Member Role and Permissions:



The system allows the user to arrive here from IX. (View Group Members) or III. (View Users)

- XI. Remove Group Member:
 - Group owner must be able to remove an existing member
 - The system must send a notification to the removed member
- XII. User Completes Group Membership Invite:
 - User clicks on the link sent by the system
 - User completes workflow I. (User Registration/Login)

2.3 Messaging & Event Management

- Group owners must be allowed to create and schedule meetings or events, send invitations, and track RSVPs
- Integration with calendar tools (e.g., Google Calendar, Outlook)
- XIII. Private and Group Messaging:
 - Real-time private messaging between users, as well as group chat features.
 - Features include typing indicators, reading receipts, and sharing media (images, documents).
- XIV. Schedule Events:
 - Group owner schedules events for the members
 - The system sends notifications and reminders to members with event details
- XV. Notification System:
 - Push notifications for messages, event reminders, and group activities across mobile and web platforms.

2.4 Surveys – Polls

- Users can create surveys or polls within groups to gather opinions or make decisions
- Technically, a survey can have one or many polls
- For simplicity, the system can implement one poll per survey
- XVI. Survey Creation:
 - Group Members must be allowed to create surveys or polls with a question, an answer, and a close date
 - The system must allow specifying questions in rich text
 - System must support answer formats like multiple-choice, rating, and open-ended
 - The system must send notifications and reminders before the close date to group members
- XVII. Answer Survey:
 - Group members must be allowed to submit answers to survey questions
 - Group member must be allowed to review their answers until the survey closes date
- XVIII. View Surveys:
 - The system must show a list of all surveys created by each owner
 - System must allow drilling down into each survey to look at details, including results statistics
 - Real-time visualization of survey results.

2.5 Data Analytics

- XIX. Data Collection:
 - Collect and securely store data on user activities, survey responses, meeting participation, etc.
- XX. Analytics Dashboard:
 - Visualize insights like user engagement, participation metrics, and survey results through a customizable dashboard.



3. Technical Requirements

3.1 Frontend

- Technology: React (or a single codebase solution such as React Native or Flutter for cross-platform compatibility).
- Features:
 - Responsive design is adaptable for web and mobile.
 - Integration of social media sign-in (OAuth).
 - Real-time data and notifications using PWA design

3.2 Backend Services

- Technology Stack: Java (Spring Boot) or Python (FastAPI/Flask/django).
- Architecture:
 - Microservices-based architecture to ensure scalability and independent deployment of various components (e.g., user management, messaging, analytics).
 - API Services: RESTful and/or GraphQL for handling data requests efficiently.

3.3 Database

- Primary Database: PostgreSQL or MySQL for storing user, group, and messaging data.
- NoSQL Database: MongoDB is used to handle non-relational data like survey responses.

3.4 Security

- Authentication: OAuth2 for secure social media login.
- Data Security: End-to-end encryption of data in transit (TLS) and sensitive data at rest (AES).
- Privacy Compliance: Ensure GDPR compliance and adherence to data protection standards.

Deliverables

Sr No	Requirement Name	Type	Complexity	Remarks
I.	User Registration/Login:	screen	simple	
II.	Profile Management:	screen	simple	
III.	View Users:	screen	Medium	
IV.	Request Group Creation:	screen		
V.	Request Group Removal:	screen		
VI.	Approve Create or Remove a Group (by admin)	screen		
VII.	View Groups:	screen		
VIII.	Invite Group Member:	screen		
IX.	View Group Members:	screen		
X.	Edit Group Member Role and Permissions:	screen		
XI.	Remove Group Member:	screen		
XII.	User Completes Group Membership Invite:	screen		
XIII.	Private and Group Messaging:	screen		Give API functionality
XIV.	Schedule Events:	screen		
XV.	Notification System:	screen		
XVI.	Survey Creation:	screen		
XVII.	Answer Survey:	screen		
XVIII.	View Surveys:	screen		
XIX.	Data Collection:	screen		
XX.	Analytics Dashboard:	screen		

Questions



Question	Response
General Question:	
What are the specific fields required for each page?	There are no special field requirements. Minimum fields are needed for the effective and sufficient functioning of use cases to modern current standards.
Would we get the design or should be scope a fixed time for that ?	You need industry standards for the functional requirement/use case design, so that is the design you will follow and document. This document already suggests expectations regarding technologies and best practices, such as testing and deployment automation, for technical design.
User Registration/Login:	
Are there any specific social media APIs preferred for integration?	Login is the only integration. The integration must fully comply with the respective social media IDM requirements. If there is any other low-handling value-added integration that is useful, I am open to that.
For multi-factor authentication (MFA), should we support multiple methods (e.g., SMS, email, authenticator apps)?	Do all media identities support a SAML / redirect type facility for authentication/MFA, forgotten username/passcode, and other functions? This way, integrating each identity we can support requires minimal work on the app side.
User Management and Authorization:	
How should user permissions and roles be defined? Any specific types of roles, beyond admin and member, that should be included?	The admin and members are the only ones at this time.
What should the process for group approval look like, and is there any SLA for the admin to approve/deny?	No SLA.
Group and Event Management:	
Is there a maximum limit for the number of members per group?	200 must be the maximum
Should groups have specific privacy settings (public vs. private)?	Private only
Should calendar integration (Google Calendar, Outlook) include two-way sync?	The admin should be able to schedule meetings for that group on a calendar, and members should be able to see that calendar. I don't see a need for two-way / integration.
Messaging:	
Should real-time messaging be handled via a third-party service (e.g., Firebase) or built in-house using something like WebSockets?	This can be a simple message board, one for each group, where everyone can post messages and content.
Would real-time messaging include multimedia content or just text or Rich Text?	Yes. If there are limitations, the design doc must list either what multimedia content is included or excluded.
Any requirement for archiving or encryption of chat messages?	Chat messages must be encrypted.
What is the expected volume of real-time users/messages per day?	For a technology demonstration, high-scaling technology is needed.
Notifications:	



Should notifications be implemented via push notifications only, or should email and SMS also be supported?	Email and SMS must also be supported.
Should notifications be configurable (allow users to choose types of notifications)?	Yes
Data Analytics:	
What level of granularity is required for analytics (real-time, daily summary)?	Daily summary will suffice.
Any specific KPIs or metrics for user engagement?	# messages, % polls / surveys answered
Survey/Polling:	
Should surveys/polls have permissions based on roles? For example, can only specific members create or respond?	Admins can create polls. Members can answer
What types of visualization are expected for survey results?	1. Summary of polls bar chart: Poll name Green Red Bar showing completion 2. Poll details in table format: Poll question: <XXX> Member name Answer 3. Member dashboard Pie chart poll completion:
Technical Requirements:	
Should there be additional compliance requirements (e.g., HIPAA, ISO) apart from GDPR?	No
Deployment and Scalability:	
Are there specific cloud service providers preferred for hosting (e.g., AWS, Azure)?	At this point, I am open to any of the 3. I am open to using all 3, this way, hybrid cloud technology can be demoed
Should the application be containerized (e.g., Docker) for easier deployment and scaling?	Yes