

Software Requirements Specification Documentation Sample

Note: This writing sample documents internal developer documentation, in particular a software requirements specification, for a new feature for a web-based event hosting app ("Reactivities"). With "Reactivities," users create social events, join others' events, and chat with event hosts and attendees. I coded the app myself!

The software requirements specification below outlines the requirements for a new feature, a group chat area where users who are signed up for an event can socialize with each other and the host leading up to the event.

Software Requirements Specification (SRS)

Project: Group Chats for Pre-Event Socializing

Document Version: 2.0

Prepared By: Craig Knepley

Last Updated: 05/07/25

1. Introduction

1.1 Purpose

This document provides a comprehensive overview of the *Group Chats for Pre-Event Socializing* feature, specifying functionality, user expectations, constraints, and criteria required for implementation. This feature will allow attendees of upcoming events on *Reactivities* to join themed group chats before the event, fostering engagement and community interaction ahead of in-person gatherings.

1.2 Scope

The Group Chats feature will allow users to:

- Create or join group chats tied to specific upcoming events.

- Participate in text-based or voice-enabled pre-event discussions.
- Receive notifications when new messages, participants, or sub-events (e.g., after-parties) are created.
- View and manage participation directly within each event's detail page.

The feature will be integrated into both *Reactivities*' web and mobile platforms, enhancing the overall social experience surrounding scheduled events.

1.3 Definitions, Acronyms, Abbreviations

- **Event Room:** A discussion space linked to a specific event, allowing attendees to communicate prior to the event date.
- **Host Room:** A chat channel accessible only to event hosts and co-hosts for coordination and planning.

1.4 References

- *Reactivities* [System Architecture Document](#), v2
 - *Reactivities* [API Documentation](#), v2
-

2. Overall Description

2.1 Product Perspective

Group Chats for Pre-Event Socializing enhances the *Reactivities* social experience by enabling real-time communication among attendees. Each chat room will be linked to an existing event or set of events and integrated into the event's/events' detail page.

2.2 Product Functions

Primary functions include:

- **Chat Creation:** Allow hosts to enable or disable pre-event group chats.
- **Join Chat:** Allow attendees to join a chat for any event to which they are registered.
- **Messaging:** Support real-time text and (optional) voice messaging.
- **Notifications:** Alert users when new messages or participants join.
- **Moderation:** Allow hosts to mute or remove users from the chat.

2.3 User Classes and Characteristics

- **Event Attendees:** Can join and participate in group chats, send messages, and receive notifications.
- **Event Hosts:** Can create, moderate, and manage event chats.

2.4 Operating Environment

- **Web:** Latest versions of Chrome, Firefox, Safari, and Edge.
- **Mobile:** iOS (14+) and Android (8.0+).
- **Backend:** ASP.NET Core API, hosted on Azure, using SQL and Redis for data persistence and caching.

2.5 Constraints

- Chat data must be retained until 24 hours after the event ends.
- The system must support up to 500 concurrent participants per event chat.
- Message delivery latency must not exceed 200 ms.

2.6 Assumptions and Dependencies

- The *Reactivities* Notification Service can handle increased real-time message traffic.

- Event data is consistently synchronized between the Chat Service and Event Service.
-

3. Functional Requirements

3.1 Chat Creation and Management

- **FR1:** Hosts must be able to enable or disable pre-event group chats during event creation or editing.
- **FR2:** Each event chat must be uniquely identified by chat_id and linked to an existing event_id.
- **FR3:** Chats must support up to 500 concurrent participants.

3.2 Join Chat

- **FR4:** Users must be able to join a chat only for events they are registered to attend.
- **FR5:** Hosts may invite additional users (e.g., guest speakers).
- **FR6:** Users receive a notification upon successfully joining a chat.

3.3 Messaging

- **FR7:** Users must be able to send and receive real-time messages with latency under 200 ms.
- **FR8:** Message history should persist for the event duration plus 24 hours.
- **FR9:** Voice messages (if enabled) must be limited to 60 seconds each.

3.4 Notifications

- **FR10:** Users must receive in-app notifications for new messages and participants.

- **FR11:** Notifications must respect user notification preferences.

3.5 Moderation

- **FR12:** Hosts must be able to remove participants or mute message privileges.
 - **FR13:** Moderation actions must be logged for audit purposes.
-

4. Non-Functional Requirements

4.1 Performance

- **NFR1:** Message delivery latency \leq 200 ms.
- **NFR2:** System must handle 5,000 concurrent active chats globally.

4.2 Scalability

- **NFR3:** System must scale horizontally during peak traffic (e.g., major event weekends).

4.3 Security

- **NFR4:** All chat data must be encrypted in transit (TLS 1.3) and at rest.
- **NFR5:** Only authenticated *Reactivities* users may access event chats.

4.4 Usability

- **NFR6:** Chat UI must display active participants, message timestamps, and notification indicators.
-

5. System Features

5.1 Event Chat UI

- **SF1:** Users can view a list of available chats for upcoming events.
- **SF2:** Users can send and receive messages within a responsive, minimal chat interface.

5.2 Host Tools

- **SF3:** Hosts can enable/disable chat functionality for their events.
- **SF4:** Hosts can view message logs and participant lists.

5.3 Notifications UI

- **SF5:** New messages or participant join events trigger a notification badge and optional push alert.
-

(end of writing sample)