Below is a sample Software Requirements Specification (SRS) for a new feature at SoundDate, a fictional dating app where users match with each other for blind dates based on the sound of each other’s voice. The new “Group Chats for Pre-Match Socializing” feature would enable group chats via voice message for speed dating events. This SRS defines the feature's objectives, requirements, constraints, and success criteria.

Software Requirements Specification (SRS)

Project: Voice-Only Group Chats for Pre-Match Socializing

Document Version: 1.0
Prepared By: Craig Knepley
Last Updated: 09/10/23

1. **Introduction**

1.1 Purpose

This document provides a comprehensive overview of the Group Chats feature, specifying functionality, user expectations, constraints, and criteria required for implementation. The Group Chats feature will enable users to join or create voice-only group chats during speed dating events, facilitating pre-match socializing and enhancing engagement.

1.2 Scope

The Group Chats feature will allow users to:

- Create themed group chats for voice-only interactions.

- Join existing public or private group chats.

- Broadcast live audio in real-time within a group.

- Receive notifications when new messages or group events occur.

The feature will be integrated into SoundDate's mobile and web applications, targeting enhanced user engagement before one-on-one matches.

1.3 Definitions, Acronyms, Abbreviations

- TTL: Time-to-Live – the duration for which a voice message remains accessible.

- Voice Stream: Real-time audio broadcast within the group.

1.4 References

- SoundDate System Architecture Document, v2.3

- SoundDate API Documentation, v4.1

2. **Overall Description**

2.1 Product Perspective

Group Chats will add a social layer to SoundDate by allowing users to meet and chat based on shared interests, creating a unique environment for pre-match interactions. This feature will integrate with the existing SoundDate API and leverage the Notification Service for in-app alerts.

2.2 Product Functions

The primary functions of the Group Chats feature are:

- Group Creation: Allow users to create themed group chats.

- Join Group Chat: Allow users to search for, request access to, or join public/private groups.

- Real-Time Voice Streaming: Enable real-time voice streaming within the group.

- Audio-Based Notifications: Notify users of group activity, such as new messages or member arrivals.

2.3 User Classes and Characteristics

- General Users: Can join and participate in voice-only group chats, initiate voice streaming, and receive notifications.

- (*future iteration*) Moderators: Can monitor group activity and remove users if necessary.

2.4 Operating Environment

- Mobile App: iOS (14+) and Android (8.0+).

- Web App: Latest versions of Chrome, Firefox, and Safari.

- Backend: Hosted on AWS with server-side Node.js, using NoSQL and relational databases for data persistence.

2.5 Constraints

- Voice message data must be encrypted and stored temporarily, with a 24-hour TTL.

- The system should support at least 1,000 concurrent users across all group chats.

- Real-time voice streaming latency must not exceed 200 ms.

2.6 Assumptions and Dependencies

- The Notification Service is capable of handling additional notification loads.

- The backend infrastructure can scale to support increased audio streaming demands during peak hours.

3. **Functional Requirements**

3.1 Group Chat Creation and Management

- FR1: Users must be able to create a new group chat by specifying a theme and choosing between public or private access.

- FR2: Each group chat must be assigned a unique identifier (group_id) for backend reference.

- FR3: A group must support a maximum of 20 simultaneous participants.

3.2 Join Group Chat

- FR4: Users must be able to search for public groups based on theme keywords.

- FR5: Users should be able to request access to private groups; private group access must be limited to invited or approved users.

- FR6: Users must receive a notification upon successful entry into a group.

3.3 Real-Time Voice Streaming

- FR7: Users within a group must be able to broadcast a live audio stream with a duration limit of 1 minutes per session.

- FR8: Audio streams should have target a latency under 200 ms.

- FR9: Only one user can broadcast at a time within a group, with queuing for other users.

3.4 Voice Message Storage

- FR10: All voice messages must be temporarily stored with a 24-hour TTL, and automatically deleted afterward.

- FR11: Messages must be encrypted at rest and in transit.

3.5 Audio-Based Notifications

- FR12: Users must receive an in-app notification when a new voice message is posted in their group(s), and when a new user joins their group(s).

- FR13: Notifications should respect users' notification settings.

4. **Non-Functional Requirements**

4.1 Performance

- NFR1: Group voice chat latency must not exceed 200 ms.

- NFR2: The system must support at least 1,000 simultaneous group chat sessions across the platform.

4.2 Scalability

- NFR3: The system should handle traffic spikes during peak usage (6-9 PM) without degradation of service.

- NFR4: Group voice chat capacity should be horizontally scalable.

4.3 Security

- NFR5: All audio data must be encrypted in transit and at rest.

- NFR6: Only authenticated users can access group chats; private groups require explicit access approval.

4.4 Usability

- NFR7: UI for group chats must be intuitive, allowing users to see active participants, start streaming, and listen with minimal interaction.

5. **System Features**

5.1 Group Management UI

- SF1: Users can see a list of available public groups and filter by theme.

- SF2: Users can create and name a new group, and select its visibility (public/private).

5.2 Audio Streaming UI

- SF3: A countdown timer should be displayed while a user is streaming, indicating time left.

5.3 Notifications UI

- SF4: Notifications should be presented in real-time, appearing in the app's Notification Center according to the user’s notification settings.