High-Fidelity Prototype README

CS 147 Fall 2025

Jasmine Tostado, Jasleen Sihota, Mara Bravo-Santos, Jasmine Aikey

Introduction

Simmr reimagines everyday cooking as a playful, social, and emotionally rich experience, bringing story-driven voice AI guidance, shared cooking sessions, and meal discovery into one ecosystem.

Design tools

We built our high-fidelity prototype using React Native, Expo, Supabase, and OpenAI APIs. We developed in VS Code and Cursor, and tested the app through the Xcode Simulator and on our own devices with the Expo Go app. We use Supabase to store and organize user data in a PostgreSQL database using a system that manages access permissions through RLS (Row Level Security).

Operating Instructions:

1. **Download Expo Go** on an iOS or Android device from the App Store / Google Play.
2. **Open this Expo link or scan the QR code** provided with this report in your browser or in the Expo Go camera.
3. Expo Go will automatically launch the Simmr app on your device. No Xcode/Android Studio setup is required.

If the QR code does not work, follow the prototype link on our team website

Getting started in the app

1. **Sign up or log in**
   Create an account using your email and password, or log in with an existing one. Follow the short onboarding flow to set your name, preferences, and import contacts.

2. **Invite friends**
   In the **Friends** tab, you can create group cooking sessions and view upcoming or past group cooking sessions. For testing, a small set of contacts and invitations may already be pre-populated so you can immediately try the flow.

3. **Explore recipes**
   Go to the **Explore** tab (first icon in the bottom navigation). Browse through multiple sections, including: Recommended recipes, Cooking with kids, Cooking with friends, Recipes trending on TikTok, and Recipes by cuisine type. **Update your pantry** directly from Explore by adding ingredients you have on hand. **Bookmark recipes** using the bookmark icon on recipe cards and detail pages. Saved recipes appear in a dedicated **Saved** tab within Explore.

4. **Chat with Simmr (AI chat)**
   Tap the **Chat** icon in the bottom navigation. This screen uses the **ChatGPT API** to automatically generate responses and suggest recipes. Example: if a user writes, *"I have chicken on hand, what can I cook?"*, the chat will respond with a short message and recommend three recipes (e.g., chicken pasta, chicken wings, chicken pot pie) based on keywords in the user's prompt.

5. **Story Log**
   The **Story Log** tab shows dishes you have cooked using Simmr. After you complete a recipe, the app saves a **story log entry**, including the recipe title and a photo of the final dish, into Supabase. You can revisit previous recipes and choose to cook them again.

6. **Cooking with friends**
   In the **Friends** tab, you can: Plan a **group cooking session (1.** Set a date/time and invite friends and 2. Pick a recipe and story tone), skip the planning and **immediately start cooking with friends,** and **view invites** to group cooking sessions your friends have created

7. **Profile & settings**
   The **Profile** tab lets users adjust preferences: 1. Change narration speed and pacing for voice guidance and 2. Update basic profile details and dietary restrictions.

Context for the prototype

Simmr is a cooking companion that blends recipe guidance with storytelling. It is designed for home cooks who want cooking to feel more engaging, social, and emotionally rewarding, whether they are cooking alone, with friends, or with kids..

The app is meant to be used **before and during cooking**:

- **Before cooking:** Decide what to make based on your pantry items, your mood, and your schedule. Plan group cooking sessions with friends.
- **During cooking:** Follow step-by-step instructions while listening to a themed story. Interact with the voice AI. Adjust pacing and narration speed so cooking feels calm rather than stressful.

Core tasks represented in the hi-fi prototype:

1. **Listening to a story while completing a recipe (Simple Task)**
   A user browses Explore, chooses a recipe, and cooks with story-driven guidance, using next / back/repeat/pause controls to move through each step.
2. **Cooking with friends as an activity (Moderate Task)**
   A user creates a group cooking session in the Friends tab, selects a recipe and story tone, invites friends, and views upcoming events, turning a normal dinner into a structured shared experience.
3. **Entertaining kids while cooking (Complex Task)**
   A user starts a recipe from the *Cooking with kids* section, launching a more visual, kid-friendly flow with warnings and playful narration that keeps kids engaged while an adult leads the cooking.

Summary of limitations

Because of time and technical constraints, the current Simmr build includes both **Wizard-of-Oz** techniques and **hard-coded elements.**

Wizard-of-Oz techniques

1. **Voice interaction simulation**
   - Voice interactions use OpenAI APIs but do not support continuous, real-time conversational flow (no WebSocket implementation).
   - The experience is not fully hands-free; users must press "Speak," record audio, and press send to trigger a response.
   - The system does not detect when users stop talking, requiring explicit submission rather than natural handoff.
   - Users cannot progress through steps with voice alone; navigating the cooking flow still requires button interactions.
   - Voice-driven screens do not advance automatically, resulting in the AI appearing responsive but not actively guiding the cooking flow.

2. **Simulated story personalization**
    - The AI generates stories dynamically, but it does not receive a structured, recipe-specific story specification.
    - User customization is limited; the only adjustable parameter is *story tone*, with additional preferences captured verbally but not fully integrated.
    - Story adaptation to recipe data is handled through simplified logic rather than a complete personalization engine.
    - **Why:** True dynamic story generation would require integrating a large language model with real-time state tracking (current step, user pace, group vs solo cooking). This was beyond the scope of a single-quarter project.

Hard-coded techniques

To keep the demo stable and focused on UX, several elements are intentionally hard-coded:

1. **Explore page recommendations:** *"Browse recipes for you"* and other "Recommended" sections do not use a real recommendation algorithm. Results do not meaningfully adapt to history or preferences; pantry updates only change labels, not the underlying recommendation logic.
2. **Recipes, stories, and visuals:** Recipe titles, steps, images, hazard badges, and story summaries come from a fixed dataset or were generated once and stored. They do not vary by user cooking style, dietary profile, or story tone.
3. **Kid-friendly recipes:** Safety icons, warnings (e.g., "requires parent supervision"), and progress bars are manually configured per recipe instead of dynamically inferred from steps. The four kid-friendly recipes are hardcoded with the images and icons included as we found that was too much to implement alongside our voice ai. After completely a kid-friendly recipe you can still take a photo and save your final dish to the story log.
4. **Profile settings:** Profile toggles and narration-speed settings are represented in the UI but do not yet fully drive recipe surfacing or voice speed under the hood.
5. **Story Log entries:** Story entries saved to Supabase reflect a simple "save this recipe" flow. They do not support editing, versioning, or deeper personalization.

Accessibility considerations & limitations

Accessibility considerations

Throughout Simmr's design and implementation, we considered accessibility in both **visual design** and **interaction patterns:**

1. **Clear status and progress**
    - We added a visible progress bar at the top of cooking screens so users always know where they are in the recipe.
    - Confirmation screens clearly indicate when a story has been saved to the Story Log.
2. **Reducing cognitive load**

- Voice narration can be paused, repeated, or stepped backward/forward, allowing users to control pace instead of being locked to a fixed script.
- Key actions (Start Cooking, Next, Back, Pause, Done) are consistently placed and visually prominent to reduce hunting and mis-taps.

3. **Kid-friendly and safety cues**
   - Kid mode includes explicit warnings for recipes that involve ovens or other hazards, reinforcing that adult supervision is required.
   - Icons and labels were updated to be more inclusive and descriptive (e.g., "Hello there" instead of "Hey family").
4. **Visual legibility**
   - We removed small, hard-to-read text from inside recipe images and moved important information into standard UI text.
   - Primary buttons and tap targets follow mobile guidelines for minimum touch size.

Accessibility limitations

Some important accessibility features are not yet fully implemented:

1. **Screen reader support**
   Not all icons and buttons have tested VoiceOver/TalkBack labels, and we have not run a full audit with assistive technologies.
2. **Motor and hands-free interaction**
   Navigation still depends primarily on touch. Voice commands (such as pausing narration) are not implemented as continuous, hands-free speech controls. There are no alternate layouts for users who have difficulty tapping or swiping.
3. **Cognitive accessibility**
   Storytelling can add cognitive load for users who prefer minimal instructions. Future versions should support a simpler "just the steps" mode and more granular control over narration length and complexity.

These limitations point to next steps: adding robust screen reader support, expanding voice-only control for key tasks, and offering multiple pacing "modes" so Simmr can adapt to a wider range of access needs.