High-Fidelity Prototype README

Planning life by how you feel, not just by what you do

Team Members:

Amesha Banjara

Bennie Nkwantabisa

Gabriella Ugwonali

Gil Silva

CS 147 Autumn 2025

KYRO

Kyro helps users balance productivity and emotional well-being by bringing emotional awareness into daily planning. Unlike traditional productivity tools that only manage tasks, Kyro focuses on **energy**, **emotional state**, and **reflection**, encouraging healthier routines and more intentional choices throughout the day. Through gentle mood logging, emotional summaries, and weekly insights, Kyro supports those who want to manage both their time **and** their mental energy.

How to Access

This prototype is **optimized for iOS (iPhone SE)** and runs through Expo Go. Android and web browser support are **planned**, but are not fully functional in the current high-fidelity build.

1. Download **Expo Go** from the App Store.
2. Open the app.
3. Scan the QR code on the right.
4. Wait for the prototype to load automatically. **Enjoy Kyro!**

The purpose of this high-fi prototype is to simulate Kyro's core emotional-reflection experience for usability testing.

Below, you'll find what tasks a user should be able to accomplish with this prototype.

High-Fi Prototype

KYRO - High-Fi Prototype

Task 1:

Users can submit a reflection in two ways:

Typing: Users type what they want to share, and Kyro generates a simulated emotional summary.

Recording: Users press the microphone button to "record," but no real audio is captured. Instead, the app simulates the recording flow and returns a scripted summary.

User Flow of Current Prototype

1. **Open Kyro**. The landing screen welcomes the user back and loads personalized content as the background gradient subtly shifts to indicate loading.
2. **View summary**. The "Hello Gabriella" screen previews recent activities and prompts the user to reflect on their week.
3. **Start a reflection**. The user is asked "Do you want to talk about your week?" and select "Let's talk".
4. **See past events for inspiration**. Kyro displays a "Since we last spoke..." and shows events that the user has completed since their last reflection. Kyro listens while displaying pulsing lights to signal active listening.
5. **Start recording**. The user is asked to "Press the microphone when you're ready" or has the option to select "Can't talk right now" and can provide a text entry. When the user indicates that they are done, Kyro summarizes the input with pre-written text ("Interesting... You said...") and presents insights or follow-up questions.
6. **Save or exit**. The user confirms the reflection summary and returns to the home flow.

This task simulates Kyro's core emotional logging loop.

Task 2:

To be implemented by final

Task 3:

To be implemented by final

Limitations

To focus on interaction design and usability testing, the following features are simplified:

- No real voice transcription
- No real AI-generated insights
- Emotional patterns and insights are static
- No backend, authentication, or database
- Only tap gestures (no drag, swipe, or long-press)
- Mood charts, stats, and scores are pre-populated

Wizard-of-Oz Techniques

This prototype does not use live Wizard-of-Oz techniques. All emotional summaries, transcripts, and insight data are pre-written and hard-coded directly into the interface, rather than provided by a human during testing.

However, in a future testing setup, we could listen to a participant's actual audio, manually transcribe it, and input a corresponding emotional summary as well as ask for a brief summary of their calendar and personalize their experience.

Hard-Coded Items

- All dialogue ("I'm listening...", "You said...")

- Reflection summaries
- Emotional labels & weekly insights
- Mood graphs and stats
- User persona (name displayed in app)
- Background textures, gradient design, and illustrated face assets
- Color Change based on emotion

These ensure a consistent, stable testing experience.

Design Tools

We built our high-fidelity prototype with **React Native** and **Expo**, using a lightweight design system and planned backend integrations.

Core Framework

- React Native
- Expo SDK
- TypeScript

Navigation & UI

- React Navigation (stack + tabs)
- Custom bottom tab bar
- expo-linear-gradient
- DM Sans font
- Paper texture + face illustrations (custom assets)

Backend & APIs

- Firebase Auth (Google Sign-In)

- Google Calendar API
- Firestore (planned)

Audio & AI (Planned)

- expo-av (audio)
- Speech recognition (requires native build)
- Gemini / Claude API (reflection summaries)

Accessibility Considerations and Limitations

Considerations

- High-contrast gradients for readability
- Black text for readability
- Large tap targets and generous spacing
- Simple, linear navigation structure
- Emoji icons paired with text labels
- “Can’t talk right now” prompt for text-based reflections

Current Limitations

- Screen reader support has not been implemented
- No haptic or audio feedback for key actions
- Text resizing and dynamic type are not yet supported