clarus

A6: High-Fi Prototype Readme

Siya Goel, Tyler Abernethy, Candace Cang, Avey Etaghene

Prototype Link

Github Link

Purpose:

clarus exists to fix the chaotic communication loop between founders and designers. Today's design tools rely heavily on text-based comments that quickly become overwhelming and difficult to turn into actionable next steps, with feedback scattered across screenshots, messages, calls, and recordings. clarus solves this by transforming unstructured, multimodal communication into organized, digestible context. Using AI, it filters and synthesizes voice, video, and text into a unified, navigable system that gives founders real visibility while freeing designers from guesswork. In short, clarus turns messy communication into seamless, structured collaboration.

Use Cases and Tasks:

1. **Leaving Multimodal Feedback Directly on Design Mockups:**
    a. Users can tap anywhere on a mockup to leave voice notes, text comments, or screen-recorded feedback.
        i. *Examples: Suggesting layout changes, pointing out spacing issues, or leaving a quick voice note about color choices*.
2. **Reviewing Recent Comments:**
    a. Users can quickly catch up on feedback through the Recents or Activity tab, seeing all comments across collaborators and files.

        i. *Examples: A designer reviewing all comments left overnight, or a founder checking which designs need attention before a meeting.*

3. **Auto-Generating To-Do Items From Comments:**
    a. Comments are automatically turned into categorized action items that teams can track and resolve. Users can also customize and edit the names of AI generated todos.
        i. *Examples: Turning several feedback notes into tasks under "Typography," "Spacing & Layout," or "Color & Contrast."*
4. **Managing Progress Through Categorized To-Dos:**
    a. Teams can view and manage progress with auto-generated categories, each containing action items linked to specific comments.
        i. *Examples: Designers marking tasks as resolved after making updates; founders checking progress across different design categories.*
5. **Calling a Collaborator for Live Review Sessions:**
    a. Users can start a call for real-time feedback and share their screen to walk through design changes.
        i. *Examples: A founder hopping on a call to clarify urgent edits; a designer screensharing to explain a new layout.*
6. **Adding Notes During Screensharing Sessions:**
    a. While screensharing, users can add comments or notes directly to files to capture important decisions made during the call.
        i. *Examples: Adding notes and comments about agreed-upon improvements to a homepage mockup or capturing action items discussed verbally.*

Tools Used:

We built our app using React Native as the core framework and Expo for testing features like voice notes and screen recording on real devices. We developed and debugged the code in VSCode while using GitHub for version control and team collaboration. Supabase powered our backend, handling authentication, storage, and real-time syncing for comments and to-dos. Builder.io helped translate Figma designs into usable code references, speeding up UI development. Cursor supported us with AI-assisted debugging and implementing complex features like transcription and screen-recording workflows.

Operation Instructions:

Install the Expo Go app and sign in or create an account. Scan the QR code on your mobile device to load and run the application in Expo Go.

You can also find the QR code by clicking on the link here.

Limitations:

The app currently restricts deeper interaction between collaborators. There is no real social networking or true calling functionality, and comments cannot be replied to or threaded, making it hard to follow conversations or maintain context. Users' comments also disappear after logout, further limiting long-term collaboration. Without push notifications, users are not alerted when new feedback or updates are added, slowing down communication and requiring manual check-ins.

Additionally, several features limit how users manage and navigate their design workflow. To-dos cannot be moved between categories once sorted, reducing flexibility in reorganizing tasks. Comment navigation is also limited, with no filtering by modality, collaborator, or recency beyond the Activity feed. Currently, the app has no offline support—because it relies on real-time connectivity, users cannot review files, leave comments, or access their project when the network connection is weak or unavailable.

Wizard of Oz Features:

The calling feature is built entirely using a Wizard-of-Oz approach: the app simulates the full experience of starting a call—opening the call screen, toggling mic and video options, and taking notes—without actually connecting to another device. This lets users feel the real workflow of “calling a designer” while avoiding the complexity, privacy concerns, and mobile data requirements of true audio/video integration. It also makes testing effortless, since no second user needs to be online.

Hard-Coded Features:

The app includes several hard-coded elements to simulate collaboration. Designer comments are pre-written to mimic real conversations, and the call feature uses a preset list of designers with no way to add new ones. User identity is also static, with only one profile option and no ability to change usernames or customize profiles. All comments must be left through the same fixed flow, with no shortcuts or modality preferences based on user behavior. These hard-coded components effectively support the prototype but limit personalization and flexibility.

Accessibility:

Ways Addressed

The app supports accessibility through multimodal commenting—text, audio, video, and transcripts—giving users multiple ways to share and understand feedback. A soft white/brown file background reduces eye strain, and the UI is clearly separated into distinct sections for easier navigation. Larger buttons, boxes, and text fields support users with low dexterity, and consistent visual cues (such as messages when exiting a call or visual cues when completing a to-do) help users understand system status.

Ways Not Addressed

Some accessibility needs remain unmet. Comment colors and the use of green/red are not optimized for color-blind users. There are also no live captions during calls, adjustable text or button sizes, or screen reader support. The app also lacks high-contrast options and language or translation support. Additionally, interaction speeds and animations cannot be slowed down, limiting accessibility for users who need additional processing time.