



clarus

Seamless, flexible communication between founders and designers

Final Report

CS 147 Fall 2025

Siya Goel | Avey Etaghene | Tyler Abernethy | Candace Cang

December 7, 2025



Table of Contents

Table of Contents	2
Project Name and Value Proposition	4
Team Member Names and Roles	4
Problem/Solution Overview	4
Needfinding	5
Interviews	5
Synthesis	6
POVs and Experience Prototypes	8
Final POV Statements	8
Alex's POV	8
Dave's POV	8
Samantha's POV	9
Top Solutions	10
Experience Prototypes	10
Solution 1	10
Solution 2	13
Solution 3	14
Design Evolution	16
Tasks that Matter	16
Simple Task	16
Moderate Task	17
Complex Task	18
Major UI Changes	20
Initial Sketches	20
Low-Fi to Med-Fi	21
Med-Fi to High-Fi	23
Did Not Implement	26
Values in Design	26
How Do These Values Show Up	27
Value Tensions and How We Manage Them	28
Final Prototype Implementation	29
Tools Used	29
Figma	29



Github	29
React Native + Expo	30
AI Tools Used	30
Wizard of Oz Techniques	31
Hard-Coded Techniques	32
Reflection and Next Steps	32
Key Learnings	32
Design Thinking Process	32
Studio Theme	33
Product	34
Future Work	35
Final Remarks	36



PROJECT NAME AND VALUE PROPOSITION

Project Name: clarus

Value Proposition: Seamless, flexible communication between founders and designers.

TEAM MEMBER NAMES AND ROLES



Siya G
Developer



Avey Etaghene
Researcher



Tyler A
Developer



Candace C
Designer

PROBLEM & SOLUTION OVERVIEW

Designers and founders often rely on text comments that fail to capture nuance, emotion, and context.



clarus closes that gap by letting people leave fast voice and video feedback that feels more natural and informative. This creates stronger understanding, better decisions, and faster product improvement.

NEEDFINDING

Interviews

We began by looking at the domain of *creative tools* because we found that the topic was applicable to many groups of users: developers, independent learners, nonprofits all struggle with storytelling, documentation, and design work. As we compared these groups, we realized that **startup founders** were a uniquely high-need population: working in high-pressure, low-resource environments, they switch rapidly between roles, handling design, product, marketing, and communication tasks simultaneously. This pushed us to narrow our scope to **early stage startup founders**, especially those navigating workflows that mix design tools, GenAI, and cross-functional collaboration.

Because founders vary widely in background and behavior, we intentionally interviewed participants across axes such as technical expertise, industry, age, gender, funding stage, and YC status. This diversity helped us understand how different demographic and work backgrounds influence workflows. For example, while technical founders may distrust creative AI, non-technical founders often rely on it heavily. First-time founders face resource constraints, whereas serial founders face challenges stemming from scale and speed. Gathering this variety of perspectives allowed us to narrow our target user population while also identifying commonalities across groups.

Recruitment was done through personal networks, founder hubs like AGI House, public cafés, and online outreach. All interviews were 30–50 minutes, conducted either in person or via video call with one interviewer and 1–2 notetakers, and all participants signed consent forms.

We interviewed six early-stage startup founders representing a wide range of technical backgrounds, industries, and funding stages. This diversity helped us capture both shared pain points and edge-case behaviors across creative workflows.

Name*	Background	Technical	Industry	Stage
-------	------------	-----------	----------	-------



Sam	Recent college grad; First time founder	Yes	Legal Tech + YC	Seed
Alex	20+ years engineering; dev-tools founder	Yes	Developer Tools	Pre-Seed
Mike	10+ years finance non-technical founder	No	Consumer	Series A
Silus	Finance, Consulting, Data science; YC Founder	Yes	Fintech/YC	Series A
Steven	2x ex-founder; non-technical	No	Consumer	Seed
Samantha	Ph D in Genetics, 2x Technical Founder	Yes	Medtech	Seed

*Names have been anonymized for interviewee privacy.

Table 1: Summary of interview participants and key findings from each interview

Across interviews, we focused on one core question: **How do early founders communicate their ideas and collaborate on creative tasks, and where do tools (including AI) help them or break down?** To address this, we asked open-ended questions about their day to day workflows, the tools they use, how they work with designers, what slows down collaboration, and how GenAI fits into (or disrupts) their creative process. Our goal was to uncover the friction points, workarounds, surprises, and unmet needs shaping early founders' communication and design pipelines.

Synthesis

After completing our interviews, we synthesized our findings using empathy maps (as seen in Fig 1). Each team member extracted pain points, surprises, behaviors, and needs from the transcripts. We then grouped these insights into thematic clusters to identify patterns across technical and non-technical founders, early vs. later stage teams, and different industries. This helped us separate individual quirks from widespread, structural issues in founder designer collaboration.

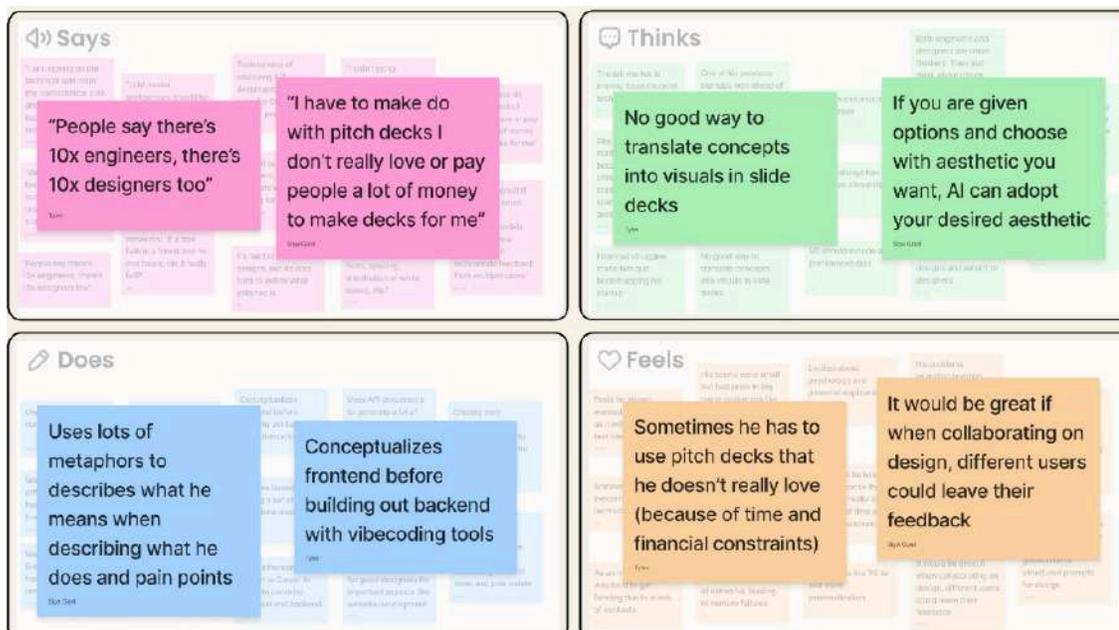


Fig 1: Highlights for Steven, a 2x ex-founder who is also non-technical.

Four themes emerged:

1. Customization

Founders wanted GenAI tools that could adapt to their aesthetic, but current design tools make editing difficult, forcing them to rely on designers even for small changes.

2. Lack of Context

GenAI tools lose context quickly, and there's no integration between chatbots, design platforms, and backend tools. Founders often had to re-explain their preferences and vision.

3. Opinions of AI on Creativity

Founders were split: some avoided AI for creative work, others used it for quick visuals, but most agreed AI can't yet match a designer's creativity and judgment.

4. Communication and Collaboration

Working with designers is an extensive process of Figma comments, long meetings, and lots of waiting.



POVs & EXPERIENCE PROTOTYPES

Here we summarize our POVs based on user insights and the experience prototypes we created to test our ideas. These prototypes let us see what worked, what didn't, and which concepts resonated most with founders.

Final POV Statements



Alex's POV (Communication)

We met Alex, a technical founder with 20+ years of engineering experience now building a dev-tools startup.

We were surprised to learn that Alex feels he wastes time in long design meetings and is stressed by his heavy dependence on designers for every visualization need.

We wonder whether this dread of miscommunication interrupts how he shows up for his broader project responsibilities.

It would be game-changing to make communication with designers seamless, enjoyable, and efficient, creating an experience that feels like co-creation instead of dependency.

From Alex's POV

- HMW reduce the need for frequent, time-draining meetings?
- HMW transform traditional verbal communication into a visual experience?
- HMW reduce miscommunication between engineers and designers?
- HMW create collaborative spaces where design decisions feel like co-creation, not dependency?



Dave's POV (Customization & Creativity)



We met Dave, a 2x founder of research-based startups with a strong technical background.

We were surprised to hear him describe AI design tools as “regimented,” unable to adapt to his preferences, forcing him to rely on designers to refine his ideas.

We wonder if he would be more creatively engaged if he could visually evolve his aesthetic through AI in a more customized, expressive way.

It would be game-changing to create a design tool that helps founders customize and evolve their visual styles and needs without complete reliance on designers, bridging the gap between creative and AI outputs.

From Dave’s POV

- HMW give founders tools to translate creative intent into visuals without design expertise?
- HMW create design tools that adapt to a founder’s aesthetic rather than forcing them to adapt to the tool?
- HMW capture the emotion and feeling behind a founder’s vision?



Samantha’s POV (Tool Fragmentation & Context Loss)

We met Samantha, a 2x founder with a PhD in Genetics who actively combines multiple AI tools (ChatGPT → Gamma) in her creative workflow.

We were surprised at how frequently she must switch between tools, re-enter context, and rebuild her creative momentum.

We wonder whether switching platforms turns creativity into unnecessary overhead rather than inspiration.

It would be game-changing to build a unified creative workspace that maintains persistent context across brainstorming, design, and presentation work.

From Samantha’s POV

- HMW keep founders in creative flow as they switch between tools?
- HMW maintain context across multiple AI models/tools without restarting?
- HMW help founders choose tools without cognitive overload?



Top Solutions

This section outlines the solution concepts that rose to the top after synthesizing our POVs, HMWs, and early design explorations. We selected these ideas because they each addressed a distinct and recurring founder needs improving collaboration, making creative direction easier, and reducing friction across design workflows. These solutions were also chosen for their feasibility, emotional resonance, and strong potential to be tested through low-fidelity prototypes. Together, they represent the most promising directions for supporting early-stage founders in communicating their vision and working more effectively with designers.

1. Real time collaborative visual playground (Alex)
2. AI-curated imagery flow that guides design direction (Dave)
3. Peer-based how-to skill-sharing platform for choosing design tools (Samantha)

Alex's	Dave's	Samantha's
A live visual playground that designers and founders can work on together while communicating in real time	AI can present a bunch of images and inspirational visuals that users can choose between and it can guide the direction of designs - almost like providing a survey to provide product context	A how-to skill sharing platform for founders to share tips and tricks on which design tools to use

Fig 2: Solutions based on each synthesis of Alex, Dave and Samantha

Experience Prototypes

Solution 1 (Collaborative Spaces)

Assumption Being Tested

Founders want to be hands-on in the creative process, and real-time collaboration reduces miscommunication more effectively than text-based feedback.

Prototype Setup



Fig 3: Testing for First Prototype

To test how founders prefer to communicate design ideas, we created a prototype with three distinct modes of interaction. In the **voice conversation** condition, the founder verbally described a birthday invitation while the facilitator sketched edits in real time based on the spoken feedback. In the **text conversation** mode, the founder sent written descriptions and revision requests exclusively through text, and the facilitator updated the drawing accordingly. Finally, in the **collaborative editing** condition, both the founder and facilitator edited the design simultaneously using shared drawing tools, combining live conversation with hands-on co-creation. This setup allowed us to directly compare how different communication channels shape clarity, engagement, and ease of iteration.

What Worked

The **voice conversation** mode successfully captured every element the participant described, and the founder noted that the real-time verbal exchange made them feel genuinely “heard.” The **collaborative editing** mode emerged as the most engaging experience overall; participants described it as “fun,” highlighting how it balanced the founder’s creative control with the designer’s expertise and created a natural sense of co-creation. Even though it was slower, the **text conversation** mode supported clearer organization of thoughts, giving users space to articulate ideas more deliberately.

What Didn’t Work



- Voice mode: Annotations were unclear, making it difficult to interpret the visual changes.
- Text mode: Slow and tedious; participants avoided giving additional feedback because typing took too long.
- Collaborative editing: Needed clearer brainstorming structure

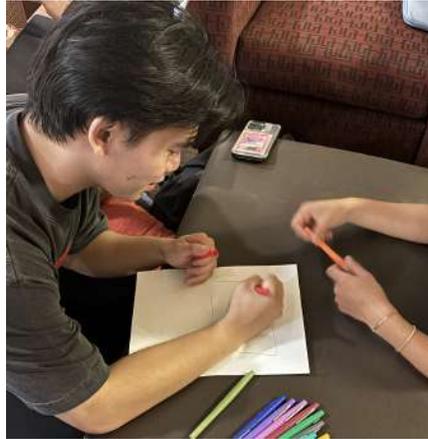


Fig 4: Testing for first prototype

Insights

Overall preference: Collaborative → Voice → Text

Visual quality preference: Voice → Collaborative → Text

Key tradeoffs:

Text = more organized but slower

Voice/Collaborative = better output but require more user involvement

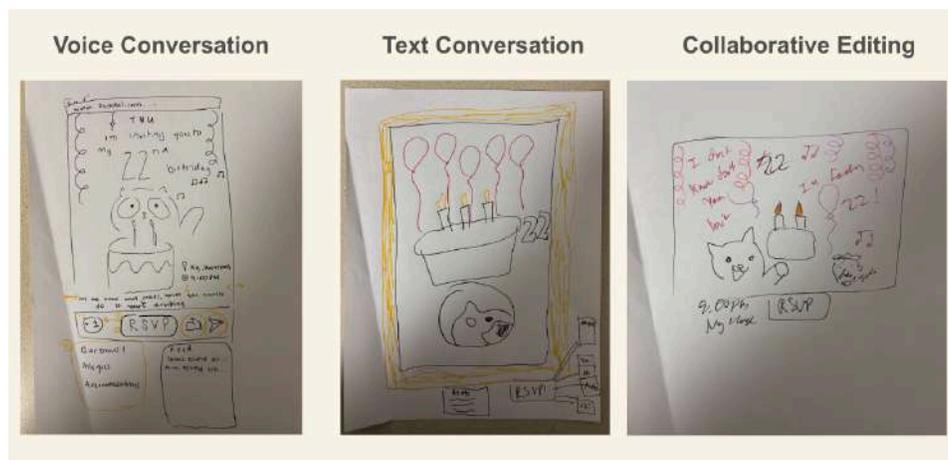


Fig 5: Final artifacts from first prototype



Solution 2 (AI guided visual choices)

Assumption Being Tested

Founders find it easier and more inspiring to make visual design decisions when choosing between curated options rather than starting from a blank canvas.

Prototype Setup

To evaluate how founders prefer to initiate visual design work, we created three contrasting invitation-creation methods. In the **freehand drawing** condition, founders designed the invitation entirely from scratch. In the **premade selection** condition (see Fig. 6), they chose between several fully designed templates with no customization allowed. In the **layered choice flow** condition (see Fig. 7), founders built the invitation step-by-step by selecting elements like color palettes, fonts, layouts, and wording. This setup allowed us to observe how much structure, freedom, and customization founders desire when making creative aesthetic choices.

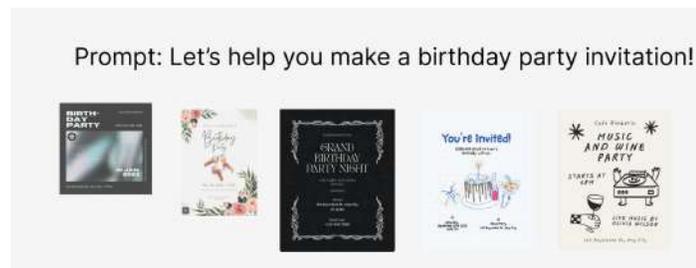


Fig 6: Artifact for second prototype, premade selection condition



Fig 7: Artifact from second prototype, layered choice flow condition



What Worked

Founders appreciated the freedom and personal control of the freehand approach, noting it best reflected their own creative intent. The layered choice flow gave just enough structure to simplify decisions while still feeling personalized. Participants also recognized that premade templates produced the most polished outputs visually, even if the process felt less flexible.

What Didn't Work

- Premade templates: Too restrictive; no way to import images or tweak components.
- Layered flow: Still lacked enough customization for founders who want more control.
- Freehand: Harder to produce visually polished results compared to curated options.

Insights

- **Process preference:** Freehand → Layered Flow → Premade
- **Aesthetic output preference:** Premade → Layered Flow → Freehand
- **Key tradeoffs:**
 - Freehand = maximum control, lowest polish
 - Layered flow = balanced structure and flexibility
 - Premade = fastest, best-looking but least customizable



Fig 8: Final Product from second prototype, left to right: freehand, premade, layered

Solution 3 (Peer Skill-Sharing Platform)



Assumption Being Tested

Founders are willing to participate in a peer driven sharing economy where they give and request advice on using design tools, and feel comfortable engaging socially with other founders.

Prototype Setup

We created **a game based simulation** where participants exchanged advice using a token economy. Each round, a founder could ask for help, offer guidance, or pass, with each action carrying a token cost. In the second round, a coin flip added uncertainty by determining whether requests would be accepted. This setup allowed us to examine social dynamics, willingness to share knowledge, and how founders balance giving versus receiving help.



Fig 9: Testing for third prototype

What Worked

Founders readily participated, demonstrating confidence in sharing their expertise and genuine interest in hearing others' experiences. The game surfaced real design tool questions they were already considering in their startups, and the added randomness introduced emotional stakes that made the interaction feel more realistic and meaningful.

What Didn't Work

- Token system: Cost didn't feel meaningful; limited rounds reduced incentive pressure.
- Expertise confidence: Less experienced founders doubted their ability to give design-specific advice.



Insights

- **Assumption supported:** Founders willingly share advice when given opportunity and structure.
- **Social dynamics:** Experienced founders tended to offer more guidance; newer founders asked more.
- **Key tradeoffs:**
 - High willingness to help, but uneven confidence across participants.
 - Emotional impact of ignored requests suggests careful UX needed for community features.

Solution Selection

Testing three contrasting experience prototypes revealed a consistent insight: founders valued **clear, expressive communication** above automation or social features. Solution 2 (AI-guided choices) streamlined decisions, and Solution 3 (peer skill-sharing) surfaced strong community energy, but neither addressed the core issue of misaligned *feedback between founders and designers*.

Solution 1, built around collaborative spaces and multimodal communication, was the only approach that made participants feel genuinely “heard.” Voice notes and live co-editing felt intuitive and closely mirrored real creative conversations. Because it directly reduced ambiguity and preserved intent, we selected Solution 1 as the foundation for our final direction.

DESIGN EVOLUTION

The final iteration of clarus brings structure to the previously unstructured flow of design feedback. By integrating multimodal commenting — voice, video, and text tied directly to specific mockup elements — and real-time synchronous calling, clarus supports natural expression while eliminating ambiguity about what feedback refers to. Automated task generation reduces the need for time-consuming feedback synthesis, turning feedback into actionable results.

Tasks That Matter

Simple Task: Asynchronous Multimodal Commenting

Scenario:



Maya, a first-time founder, opens her designer's newest mockup for her onboarding flow. She's between meetings and doesn't have time for a call, but she notices immediately that the sign-up button feels too small on mobile. Instead of typing out a long explanation, she taps the area and records a quick 15-second voice note: "I think this button needs to be bigger because I missed it at first glance. Maybe increase contrast, too?"

The voice note is pinned directly to the sign-up element, giving her designer Leo clear context without the need to schedule another meeting.

The simplest task is **leaving a voice, video, or text comment asynchronously** and it mirrors the most frequent real world action in founder designer workflows. Users can tap anywhere on a mockup/file they are editing and instantly express their thoughts in the modality that best fits their intention. This task matters because it keeps natural communication patterns while preventing the ambiguity typically caused by text only feedback/long slack threads.

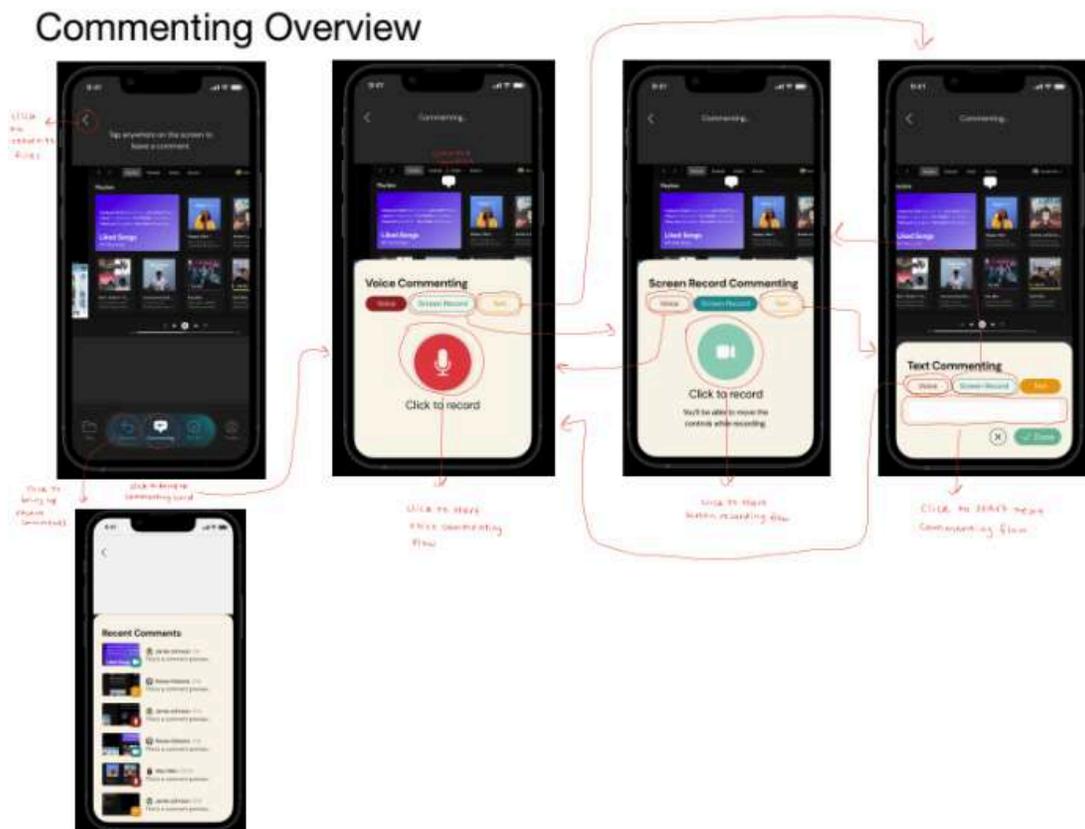


Fig 10: Simple Task Annotated Workflow



Moderate Task: Viewing and Completing AI-Generated To-Dos

Scenario:

Later that afternoon, Maya checks the AI-generated to-do list that clarus created from her earlier voice notes. clarus has automatically grouped her feedback into actionable items such as “Increase button size on mobile screens” and “Adjust color contrast for accessibility.”

Her designer, Leo, opens the same to-do list and immediately understands the priorities without digging through old messages. As Leo completes the changes, Maya sees real-time updates reflected in the checklist, keeping both of them aligned without back-and-forth clarification.

The moderate task is **managing AI-generated action items extracted from multimodal feedback**. This task addresses a crucial pain point we repeatedly observed: founders share ideas, but designers are left to interpret and prioritize them manually. clarus bridges this gap by automatically synthesizing feedback into categorized to-dos.

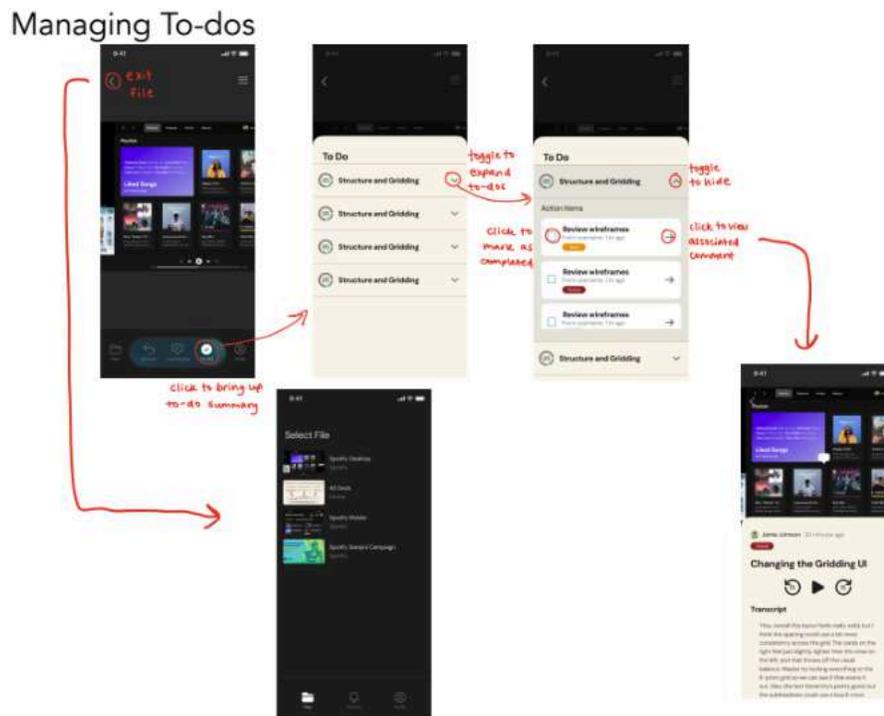


Fig 11: Medium Task Annotated Workflow

Complex Task: Real-Time Collaboration via Calls

Scenario:



The next day, Maya and Leo need to discuss a more nuanced change: refining the hero section of their company’s landing page. They join a real-time call inside clarus, and Maya starts screen sharing. As she scrolls through the page, she says, “I want this header to feel more bold and confident, kind of like Notion’s vibe, but still warm.”

Leo takes live notes directly on the shared mockup while Maya speaks, attaching them to specific elements. Because clarus provides clear confirmation steps, countdown timers, and visible exit paths, Maya feels comfortable navigating the call and sharing her screen. The session ends with aligned understanding and a clear set of shared next steps, an outcome that would have required multiple rounds of email or Slack messages in traditional tools.

The most complex task in clarus is **initiating a real-time call with a designer and sharing the screen to collaborate live**. This workflow supports moments when asynchronous feedback is not enough — when ambiguity, urgency, or detailed visual clarification requires synchronous discussion. This task restores the sense of “being in the room together,” which founders repeatedly described as missing from existing tools. As it involves coordination with other teammates, this is the most complex task in clarus.



Fig 12: Complex Task Workflow



Fig 13 a and b: Complex Task Calling Screen

UI Iterations: From Sketching to High-Fi Prototyping

Initial Sketches

After defining our tasks, we created sketches and storyboards to explore interface ideas across mobile, web, and AR/VR. Our concepts ranged from a mobile app with multimodal commenting and AI-generated to-dos to web-based tools that transformed voice feedback into UI edits. We also imagined spatial AR/VR collaboration where teams could co-edit and annotate in shared 3D environments. Together, these sketches helped us map the design space—from simple asynchronous commenting to rich synchronous and immersive collaboration—and clarified how Clarus could evolve across different levels of fidelity.

Comparing these concepts revealed that some modalities created more friction than value. VR, while compelling, introduced accessibility barriers, adoption challenges, and technical complexity that conflicted with our goal of lightweight collaboration. Mobile emerged as the most practical and inclusive direction—one that aligned with how founders and designers already communicate. Paired with the interview insight that founders wanted clearer ways to express feedback, not AI-generated designs, we focused on a mobile solution centered on multimodal commenting and documentation. This choice supported real communication habits while remaining simple, flexible, and user-aligned.

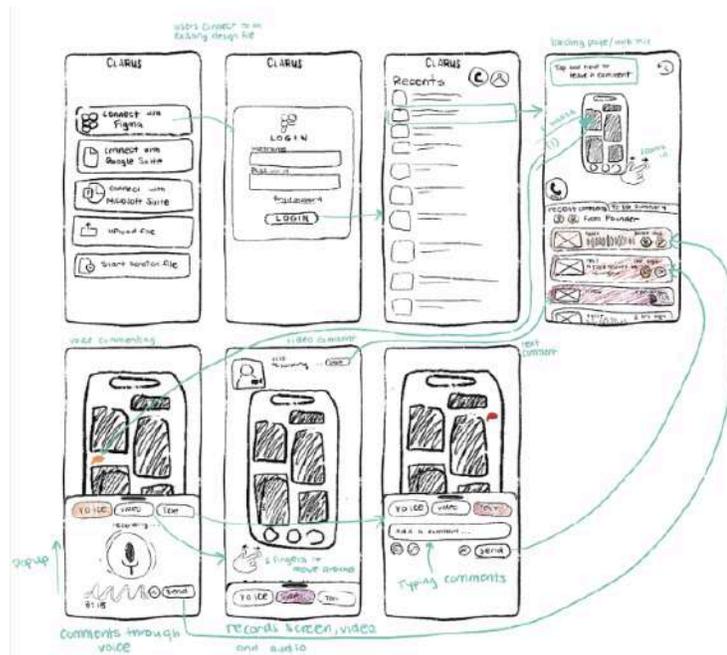


Fig 15: Key Screens From Low fi Prototype (Complex Task)

From usability tests, we found that the moderate task (realtime calling) consistently created the most friction for users, while the simple and complex tasks performed more smoothly. Participants were generally satisfied with the simple and complex tasks, but the moderate task felt the “least intuitive,” leading to lower satisfaction and more navigation errors. Even though users completed all tasks in similar time ranges, the moderate task produced the most misclicks and confusion, showing that fewer screens do not necessarily translate to a better user experience. In contrast, the simple task’s longer flow felt straightforward, and the complex task—though slower at times—felt more relevant because users found the AI-generated summaries genuinely helpful.

These patterns revealed clear pain points and design implications. Users struggled with unclear instructions in the simple task, mixed commenting-and-calling flows in the moderate task, and confusing icons in the complex task. Together, these challenges pointed to three major design needs: clearer terminology to reduce ambiguity, more distinct separation between editing and commenting actions, and improved affordance through intuitive iconography and button placement. Overall, the usability data shows that the core idea resonates with users, but clarity, guidance, and visual cues must be strengthened to support confident and efficient navigation.



These friction points pushed us toward the med-fi version, where we focused on switching tasks, revising the call flow and icons, and enforcing clearer mode separation (commenting vs. calling vs. AI to-dos). We also layered in lightweight guidance, such as visible instructions and clearer anchors. The med-fi prototype marked the moment where clarus gained a skeleton, something users could finally navigate with intention.

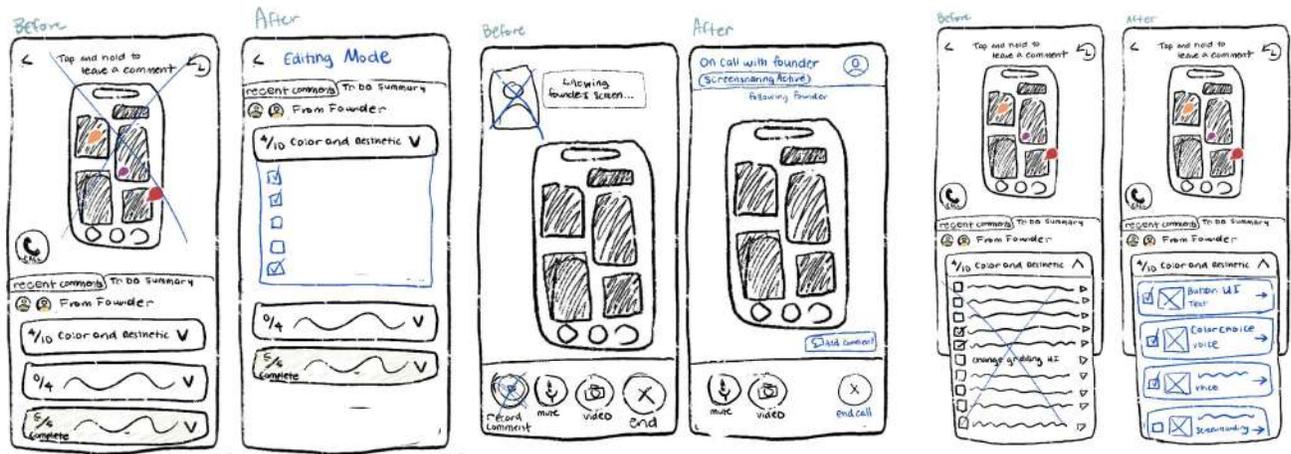


Fig 16: Revised interface sketches after lo-fi prototype usability testing

Med-Fi to High-Fi, Fixing the Friction

Our move to high-fi was shaped by a formal heuristic evaluation that uncovered **55 violations**, including **18 high-severity (3-4)** issues that materially affected clarity, safety, and navigation. These issues revealed the subtle but critical moments where the system failed to support users. The majority of these issues focused on H4 (Consistency & Standards), H7 (Flexibility & Efficiency of Use), and H1 (Visibility of System Status). The major violations and fixes we made are listed below, with several violations combined to summarize the key changes:

Simple Task: Adding Multimodal Comments

Problem: Users got trapped in comment and recording flows (H1, H3).

Fix: Added consistent back buttons, visible exits, and confirmation modals before discarding input.

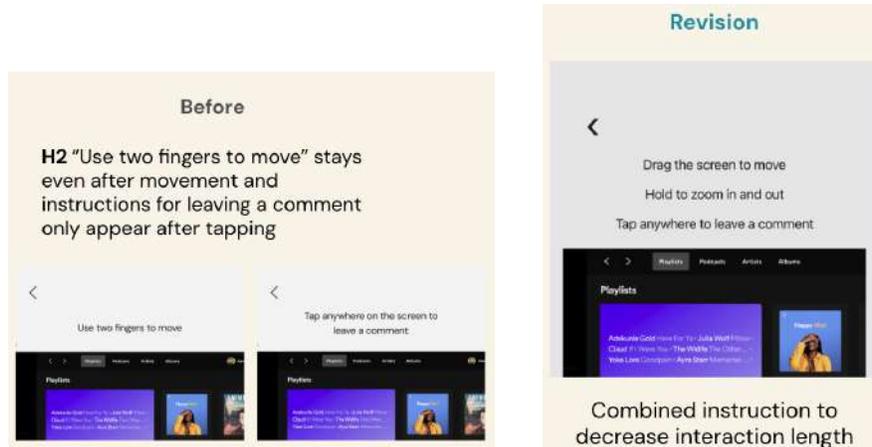


Fig 17: Before and After

Problem: Comment cards lacked control (H7).

Fix: Added delete and replay options, giving users control over recorded feedback.

Problem: Colors and icons lacked hierarchy (H4).

Fix: Standardized saturation levels and icon weights for predictable interaction patterns.

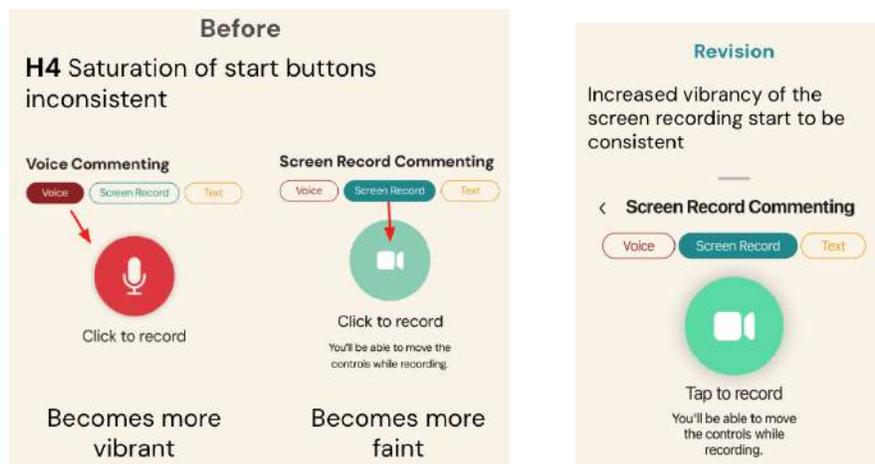


Fig 18: Before and After

Moderate Task: Viewing AI-Generated To-Dos

Problem: Back button placement was unclear (H3).

Fix: Moved the back button onto the panel itself for predictable navigation.

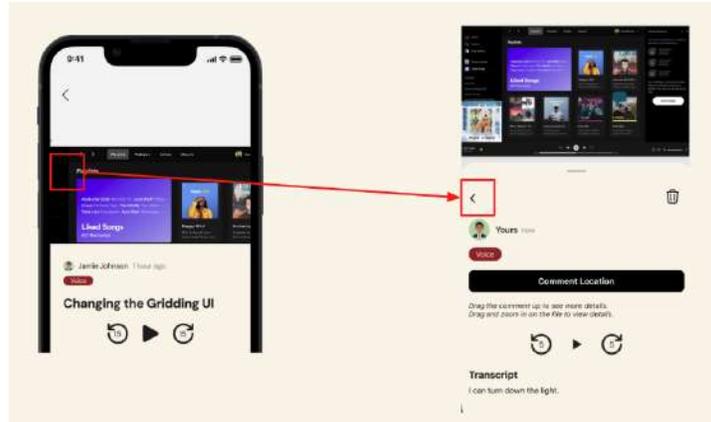


Fig 19: Before and After

Complex Task : Calling & Screen Sharing

Problem: Risky actions (calls + screen sharing) launched with no warning (H5).

Fix: Added countdown timers, confirmation prompts, and clearer system status before starting.

Problem: Entire profile card was clickable to initiate calls, which made it look like a link to a contact card (H2).

Fix: Restricted call initiation to a dedicated call button only.



Fig 20: Before and After



Problem: No way to leave an initiated call (H3).

Fix: Added a visible end-call button on the ringing call screen.



Fig 21: Before and After

Did Not Implement

H11 Accessibility: No live transcription and captions during live calls

- Adds to clutter during calls which many users did not like during interviews in A5.

H7 Flexibility and Efficiency of Use: No ability to call someone while commenting, must navigate to “files”

- From interviews in A5, we discovered that users prefer having the call button here.
- Gives users the ability to start a call and then navigate to a file. More control and usability.

Values in Design

Across our needfinding, experience prototyping, and iterative design cycles, three core values surfaced repeatedly as the foundation of effective founder–designer collaboration. These values didn’t come from abstract design theory; they emerged directly from the frustrations, desires, and behaviors we observed in testing.

1. Clarity : Reducing Ambiguity at Every Touchpoint



Founders consistently emphasized how easily feedback gets lost, misinterpreted, or buried in existing tools. They struggled to express ideas precisely, and designers struggled to understand "what the founder really meant." We identified **clarity** as a guiding value (and our namesake) because communication in early-stage product development is incredibly fragile.

Clarity shaped decisions around navigation, comment anchoring, confirmation states, and visual hierarchy. We aimed to ensure that every action in clarus from tapping to comment to initiating a screen share is predictable, transparent, and unambiguous.

2. Empathy : Supporting Human Ways of Communicating

Interviews repeatedly showed that tone, emotion, and nuance are lost in text-only tools. Founders often said things like "*I don't know how to explain it, but I can show you*" or "*It's easier to just talk it out.*" These insights pointed us toward **empathy** as a core value.

Empathy meant designing for *how people naturally express themselves* through speech, intonation, gestures, hesitation, and emphasis not forcing communication through rigid text boxes. This value drove our decision to support voice and video comments, replayability, and interface simplicity that reduces cognitive load.

3. Co-Creation: Treating Founders and Designers as Partners

Our experience prototypes showed that founders loved being "in the room" creatively, but not doing the designer's job. What they wanted was **shared space**, not full control. Collaborative editing, real-time calls, and simultaneous sketching consistently created the strongest sense of alignment.

This led us to embrace **co-creation** as a value not just "sending designs back and forth," but building a tool that supports collaborative dialogue, shared context, and joint decision-making. Co-creation became a lens for everything from live call features to how AI synthesizes feedback for *both* roles.

How Do These Values Show Up

Clarity → clear pathways, explicit controls, visible status

- Pinned multimodal comments reduce ambiguity by tying feedback directly to UI elements.



- Confirmation modals, countdown timers, and clear exit buttons prevent accidental actions.
- Standardized iconography and visual hierarchy make the system predictable and easy to learn.

Empathy → support for voice, video, and expressive feedback

- Voice and video commenting preserve tone, emotion, hesitation, and emphasis things text alone loses.
- Replayable recordings let designers hear the “why,” not just the “what.”
- The interface avoids overwhelming users with clutter, reducing cognitive load during complex tasks.

Co-Creation → real-time collaboration and shared understanding

- Live calls + screen sharing recreate the energy of in-person work sessions.
- Collaborative notes during calls ensure both sides walk away with the same interpretation.
- AI-generated to-dos translate shared conversations into mutual next steps.

Value Tensions and How We Managed Them

Clarity vs. Co-Creation

Real-time collaboration is inherently messy, and too much structure can interrupt the natural flow of creative work. We addressed this tension by providing light, intentional scaffolding, clear visual controls, safe exits, and a consistent layout so teams can stay oriented without feeling constrained. This approach keeps collaboration fluid while still preventing clutter and confusion.

Clarity vs. Empathy

Highly structured systems risk flattening tone, emotion, and nuance, which can make feedback feel impersonal. To balance clarity with human expression, we support multimodal communication—voice, video, and text—so users can choose the medium that best conveys emotion and intent. Comment replay preserves original tone, allowing teams to stay organized without losing the empathy embedded in each message.

Co-creation vs. Empathy

Co-creation often requires fast iteration and honest critique, but that speed and directness can unintentionally diminish empathy between collaborators. We



addressed this by designing interactions that encourage thoughtful pacing and emotional awareness while still supporting momentum. Features like contextualized comments, optional tone-preserving voice notes, and clear attribution help collaborators understand not just *what* was said but *why*, ensuring feedback remains both productive and considerate.

Empathy vs Efficiency

Supporting many modes of expression—voice, video, and text—can slow down tasks and introduce friction. To avoid this, we designed every modality to be equally lightweight, enabling quick capture and seamless playback. Users can express themselves in whatever format feels natural without sacrificing speed, allowing the system to remain efficient while preserving emotional depth.

FINAL PROTOTYPE IMPLEMENTATION

Tools Used

Figma (High-Fi Redesign + Interactive Prototyping)

Pros:

- Let us design, iterate, and test UI flows before implementing them in code.
- Components, variants, and interactive flows clarified how screens should behave.
- Essential for heuristic evaluation and for documenting fixes during high-fi revisions.

Cons:

- Could not simulate actual voice/video commenting, live calls, or countdown timers.
- Required manual duplication of screens for every state change, making large flows harder to maintain.

GitHub (Version Control + Collaboration)

Pros:

- Allowed branching, merging, and parallel development among teammates.
- Pull requests helped maintain consistent coding standards.
- History tracking made it easy to revert design or logic experiments.

**Cons:**

- Merge conflicts occasionally slowed development.
- Requires strict coordination to avoid overwriting UI changes.

React Native + Expo + Supabase (Core App Development & Testing Environment)

Pros:

- Enabled us to build a realistic mobile environment and test flows directly on physical devices.
- Cross-platform setup reduced overhead and made it easy for teammates to contribute without heavy native tooling.
- Expo provided fast live reloading, which accelerated iteration and debugging during development.
- Supabase helped us create login and account features and save user comments.

Cons:

- Limited in simulating complex native behaviors such as real-time calling or social networks.
- Multimedia actions (audio/video recording) required external libraries or permissions outside the scope of the class.
- Some advanced Expo modules (camera, microphone, background permissions) were restricted or required additional configuration.

AI Tools

We used a combination of AI-assisted development tools—primarily Cursor and Builder.io—to accelerate implementation and support complex features that would have been challenging to build from scratch within our timeline.

Builder.io was used to translate portions of our Figma design into HTML and structural code that we could reference while building React Native components. Although the output was not directly plug-and-play, it provided useful starting points and clarified layout hierarchies that reduced design-to-code translation time.

Cursor served as our main AI coding assistant. It helped us write React Native logic, debug unexpected edge cases, and scaffold complex features such as the screen-recording workflow, multimodal comment system, and transcription



pipelines. Cursor was especially valuable for quickly iterating on gesture handling, optimizing state management, and resolving TypeScript errors. In many cases, it accelerated problem-solving by offering alternative implementations or pointing out issues that would have taken significantly longer to identify manually.

Using AI tools meaningfully shaped our development workflow. They allowed us to prototype rapidly, explore multiple solutions, and implement technically demanding features—such as synchronized audio/snapshot recording and dynamic comment rendering—that might not have been feasible within a short course timeline. At the same time, using these tools required careful oversight: AI-generated code often needed refinement, restructuring, or debugging to meet performance and correctness standards. Ultimately, AI served as a powerful collaborator, helping us focus our time on higher-level design decisions and user-experience improvements while still enabling robust technical implementation.

Wizard of Oz Techniques Used

Med-Fi Wizard of Oz Features

In the med-fi prototype, Wizard-of-Oz techniques were central to simulating nearly all system behavior. Since the prototype lived entirely in Figma, user interactions could not trigger any real functionality, so the team relied on curated flows to imitate real system responses. Commenting, audio/video feedback, and transcripts were all pre-scripted, and comment bubbles appeared in predetermined locations regardless of where the user clicked. The Activity feed showed preset “changes” that did not reflect any actual user actions. The calling flow was also fully simulated—users tapped through a staged sequence of screens without any connection to real collaborators. These Wizard-of-Oz components allowed realistic user testing despite the absence of backend logic or multimodal recording features.

High-Fi Wizard of Oz Features

In the high-fi version, Wizard-of-Oz elements were significantly reduced. The most prominent remaining example is the simulated real-time call: users experience a ringing screen, an in-call interface, mic/video toggles, and a note-taking flow, but no actual network-based call occurs. This allowed the team to evaluate the interaction design while avoiding the privacy and technical complexity of real audio/video streaming. Because real-time syncing, commenting, file uploads, and



to-dos were now functional in code, the high-fi relied far less on simulated behavior compared to the med-fi.

Hard-Coded Techniques Used

Med-Fi Hard-Coded Features

The med-fi prototype contained extensive hard-coded elements due to Figma constraints. All interface elements were static, including preset design files, fixed comment locations, and default text/audio/video comment placeholders. The Activity feed and file views did not reflect true underlying data—they displayed predetermined examples regardless of user behavior. The user profile, designer list, and comment modalities were also fixed. All comments, to-dos, and transcripts were mock examples, as no real recording or backend was connected.

High-Fi Hard-Coded Features

The high-fi significantly reduced hard-coding because a real backend and multimodal functionality were introduced. Some hard-coded elements still exist: designer options in the calling screen are fixed, several sample designer comments are prewritten to simulate collaboration, the user profile is static with no customization, and users must follow the same commenting flow each time with no adaptive shortcuts. However, most core features—comment creation, transcripts, file uploads, activity feed updates, and to-dos—now operate dynamically through Supabase and real code, replacing the static flows used in the med-fi.

REFLECTION AND NEXT STEPS

Key Learnings

Key Learnings on Design Thinking Process

Empathize with users every step of the way

Working on clarus taught us that founder designer collaboration isn't just a workflow problem — it's an emotional one. Every prototype revealed anxieties about being misunderstood, hesitations around giving feedback, and frustration when tone or intent was lost. Empathizing early, often, and in multiple modalities helped us design features like voice/video commenting and pinned annotations that honor



how people naturally communicate. We learned that great collaborative tools don't replace human expression, they amplify it.

Prototypes reveal the truth users won't say out loud

Our Interviews told us founders “liked giving feedback,” but only when we put voice, text, and collaborative editing prototypes in front of them did we see what actually worked. Collaborative spaces sparked energy; text alone slowed everything down; voice only lacked structure. Seeing these reactions even the awkward ones pushed clarus toward a multimodal, communication-first direction. We learned that prototypes surface honesty in a way questions never can.

Iteration is not about refinement—it's about revelation

Each version of clarus changed our understanding of the problem itself. The med-fi made us realize our navigation was confusing; the high-fi revealed technical friction points in multimodal commenting; and our Wizard-of-Oz flows exposed unmet expectations around call behavior. Iteration didn't just polish our ideas—it revealed blind spots and reframed the problem. Users consistently surprised us, reminding us that iteration is the engine of discovery, not just improvement.

Key Learnings on Studio Theme: Intelligent Creative Tools

AI should enhance human agency, not automate it away

Across our prototypes, we found that users wanted control over how intelligence shaped their workflow. They didn't want AI to rewrite their feedback or make creative decisions for them—but they valued AI that organized, summarized, and clarified their intent. This taught us that intelligent creative tools work best when they empower rather than replace; when the user feels amplified, not overridden.

The value of intelligence emerges through context, not complexity

We learned that sophisticated AI does not automatically create a better experience. Users cared less about the “intelligence” of our models and more about whether the system responded appropriately to their context—where the comment was placed, the tone of voice, the modality used, or the urgency of the feedback. Intelligence that is context-aware, even if simple, consistently outperformed intelligence that was technically impressive but detached from user needs.



Collaboration is the core of intelligent creative work—not an afterthought

Throughout the studio, we noticed that most teams gravitated toward tools that enhanced collaboration, not just creation. This reinforced a key insight: intelligent creative tools are most impactful when they strengthen how people work together. Even the most advanced AI becomes irrelevant if collaborators can't share context, understand each other's intent, or build on one another's ideas. Seeing other teams tackle similar challenges—shared canvases, co-editing, feedback loops—highlighted that intelligence in creative tools must support the *collective* process, not just individual productivity. For clarus, this validated our focus on multimodal communication, shared recents, structured to-dos, and simulated live reviews. Intelligence becomes meaningful when it elevates collaboration, not replaces it.

Key Learnings on Product

Communication is the product, not a feature

Across needfinding and testing, we realized clarus wasn't really about commenting tools or design screens; it was about reducing miscommunication. The more we learned, the clearer it became that our product's value came from closing the gap between what founders *say* and what designers *interpret*. This shifted our mindset from building "features" to building communication clarity through shared context, visible intent, safe real-time flows, and structured synthesis.

Multimodality is essential—not optional—for real collaboration

Text comments alone could not capture the richness of founder–designer communication. Voice conveyed tone and personality; screen recordings conveyed flow and behavior; text conveyed precision. Our product only became truly usable when we treated multimodality as foundational. This shifted our perspective from "offering multiple input types" to "supporting the full spectrum of human expression."

The smallest interaction details determine whether collaboration feels 'safe'

Through testing, we learned that people hesitate to give feedback when they fear being unclear or sounding harsh. Seemingly small details—like confirmation messages, safe exits, visible comment anchors, and predictable modal behavior—had an outsized impact on user confidence. We realized that product



quality isn't defined only by powerful features, but by micro-interactions that reassure users they won't lose data, embarrass themselves, or break the workflow.

Future Work

Enhanced Accessibility & Expressiveness

With additional time and resources, we would expand clarus to support more inclusive and expressive communication for all users. This includes adding live captions during calls, on-device or cloud-based automatic transcript generation, and support for screen readers and keyboard navigation to ensure the interface is fully navigable for users with visual or motor impairments. We also see significant value in multilingual voice-note transcription and translation, enabling global teams to collaborate without language barriers. These improvements would preserve the emotional richness of multimodal feedback while ensuring that clarus remains usable, legible, and equitable for diverse teams and contexts.

True Real-Time Collaboration (Beyond Simulation)

Our high-fidelity prototype includes simulated calling and screen-sharing flows that focus on interaction design rather than technical implementation. In future iterations, we aim to replace these Wizard-of-Oz elements with fully functional real-time collaboration tools. This includes actual voice/video calling, live screen sharing, and shared cursors so collaborators can point, annotate, and review designs together. We envision real-time synchronized notes, co-viewing of mockups, and the ability to watch feedback unfold dynamically rather than asynchronously. These capabilities would transform clarus from a guided prototype into a fully interactive collaboration platform—one where founders and designers can co-create, discuss, revise, and align instantly, all within a single unified workspace.

Smarter AI Support for Creative Interpretation

As clarus evolves, we also envision a more intelligent layer of AI that understands not just the content of feedback but the intent behind it. Future versions could analyze tone, urgency, and design semantics to generate more context-aware to-dos, prioritize tasks, or detect conflicting feedback. AI could also summarize large threads of feedback, highlight patterns across multiple collaborators, or recommend next steps based on previous design iterations. This



deeper intelligence would move clarus from simple organization toward true creative augmentation—helping teams interpret, align, and act on feedback with far greater clarity.

Expanded File Support & Cross-Tool Integration

Another future direction is broadening file compatibility beyond static mockups. Integrations with tools like Figma, Framer, Adobe XD, or Canva would allow users to comment directly on live design files instead of exported screens. clarus could also support version history, allowing users to compare iterations and trace how feedback influenced design changes over time. This would position clarus as a bridge between design platforms and collaboration systems, reducing friction and keeping feedback tightly connected to the evolving work.

Final Remarks

Thank you for embarking on this clarus journey with us. This project became more than a prototype, it became a reminder that collaboration is emotional, human, and often fragile. We learned that founders and designers don't just want features, they want to feel understood.

While what we built is only a glimpse of what clarus could become and with more time, we hope to grow clarus into a fully live, accessible, real-time environment that strengthens not just workflows, but relationships. This project taught us that great tools don't replace the human in the loop, they elevate them. And that is the heart of clarus: turning messy feedback into meaningful connection, one conversation at a time.

Our team could not be possible without the support from the CS 147 staff, especially our TA Diya Sabharwal, Head TA Alan Chang, and Professor James Landay. We would also like to thank Haven, our studio's other team, for their feedback as well as all of our interviewees and participants for making clarus what it is today.