How to Prepare Prototypes for Moderated Usability Testing

CRAFT INNOVATIONS

craftinnovations.global

Contents

How to prepare a prototype for moderated usability testing

Moderated usability testing measures UX and customer experience by evaluating interface convenience, tracking real user behavior, and uncovering friction points.

✦ During usability testing, your prototype serves as the measuring instrument. Any technical flaw in the prototype distorts user behavior, leading to false product findings in your report.

As a research & CX design agency, Craft Innovations has completed over 47,000 hours of usability research across products serving 54 million users. Over years of running research sessions, we have nailed down every technical detail in a prototype that can compromise an entire research project.

This handbook compiles the standard checks we developed to fix those issues. Run through these checks before your first session to eliminate prototype-induced bias and ensure your research measures actual product usability.

While this handbook uses Figma to demonstrate technical interaction checks, 85% of these readiness principles apply to any prototyping tool (Framer, Axure, ProtoPie, Penpot, etc.).

How to use this handbook

Every recommendation carries 1 of 2 markers:

must have — Do this. A prototype missing any of these produces findings we would not put in front of a stakeholder.

optional — Worth doing. These improve the depth of what you learn rather than the validity of it.

There are 27 must-have checks and 8 optional ones. The last chapter collects all of them onto a single page: **The Prototype Readiness Check**.

✦ Look out for **AI Tip** blocks throughout this handbook. They highlight specific steps where AI can automate tedious tasks.

1. Why this is worth a day of your time

✦ Preparing a prototype properly takes ~1 working day for a flow of 20–30 screens, and it is the cheapest insurance on the project. A session run on a broken prototype still costs you the recruiting fee, the incentive, the moderator, an hour of the respondent's time, and unreliable findings.

1.1 Prevent Figma hints from giving away correct answers

By default, Figma flashes every hotspot on the screen when a participant taps an unwired area, so on a sparsely wired screen that flash points straight at the correct button — **chapter 4** covers how to keep it from becoming an answer key.

1.2 Capture misclicks to reveal true mental models

The most valuable research insights come from wrong turns, as they reveal the exact gap between a user's mental model and your interface logic. When secondary buttons do not respond, dead taps leave no trace.

Connecting alternative paths and secondary sections prevents this data loss. Preserving real-time mistakes stops post-hoc rationalization. This keeps your qualitative findings grounded in actual user behavior.

✦ In this handbook, a **misclick** is any tap that was not the intended path — a tap on something unwired, or a tap on a working control that leads somewhere else. Some testing tools count those as two separate events; here the distinction does not change what you build.

1.3 Protect CSAT and effort scores from technical friction

Technical friction directly depresses Customer Satisfaction (CSAT) and Customer Effort Score (CES) ratings. When participants spend minutes recovering from a dead end or prototype reload, their frustration lands directly on your product's usability score.

Eliminating technical prototype bugs before fieldwork guarantees clean metrics. It separates tool friction from product usability. That way, the numbers in your report describe your product rather than your prototype.

2. Before you build the prototype

3 decisions to settle before you make the first connection.

2.1 Reserve the time must have

Book preparation as a task, with a name and a day on it. For a flow of 20–30 screens, plan on most of a working day for the connections and checks in this handbook, + 1 hour for the dry run in **chapter 8**.

2.2 Work on a frozen testing copy must have

Duplicate the file and test from the copy only. Name it so nobody has to ask which one is live. We use *[Product] — Usability test — [date]*.

Design files stay alive: a colleague adjusts a component, a token gets renamed, someone tidies a layer group. If your test link points at the working file, a change made on day 2 means respondents on day 1 and day 2 saw different products, and you will not know which findings apply to which version.

Freeze the copy once the dry run passes. After that, changes go through **section 8.6**.

2.3 Clean the file before you share it must have

Delete every trace of how the file was made. Specifically:

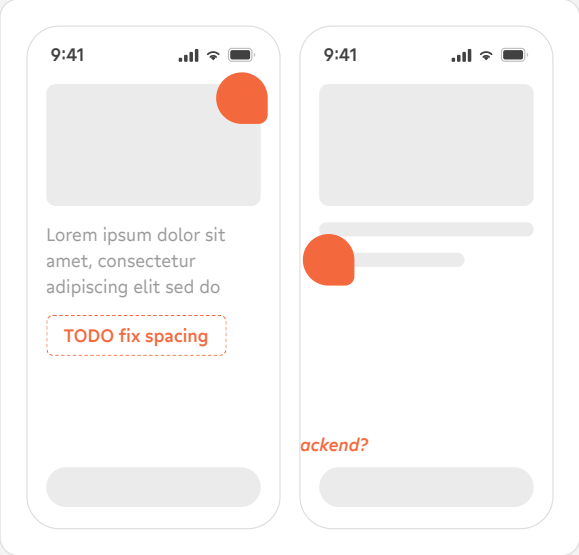
- Figma comments, including resolved ones as they still show as pins.
- Annotation layers, red boxes, arrows, and notes-to-self left on top of screens.
- TODO, FIX, placeholder layer names that surface in the layers panel.
- Lorem ipsum and unreplaced Text defaults.
- Duplicate or half-finished screens.

Notes and comments the team left for each other can surface in front of respondents — Figma comments only if they get switched on, everything else immediately. Either way, the respondent stops testing the product and starts reading the file.

AI Tip: Export your frame/layer names and text strings into AI, then ask it to flag leftover placeholder copy (Lorem ipsum), unresolved TODOs, and internal technical tags. AI catches forgotten strings across dozens of screens in seconds.

Bad

Comments, annotations and placeholder copy. Respondents read all of it.



Good

A file that says nothing about how it was made.

3. The start screen

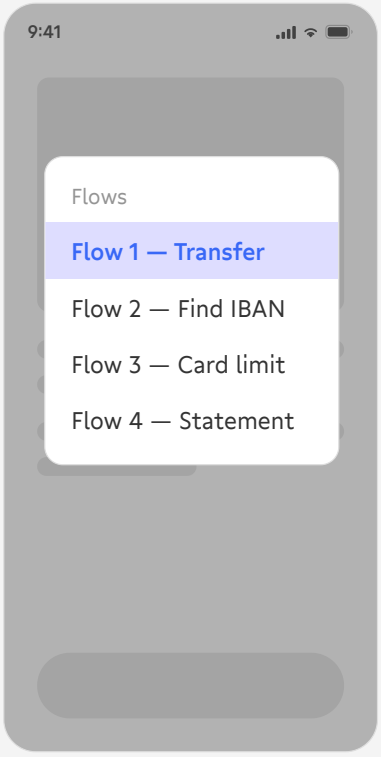
The start screen is a menu of the session's tasks, built as the prototype's first screen.

It does 2 jobs: it lets anyone recover the exact position in the flow in 1 tap after a freeze, a reload, or a wrong turn, and it stops respondents from wandering into task 3 before the moderator has introduced it. Without it, every technical hiccup costs the respondent a full walk through the flow from the beginning.

What about Figma's own flow list? It looks equivalent but it is not. Switching flows is a moderator action, so either the moderator takes over the respondent's screen mid-session or talks them through a Figma UI that has nothing to do with your product. And the moment that panel appears, the respondent is reminded they are inside a design tool.

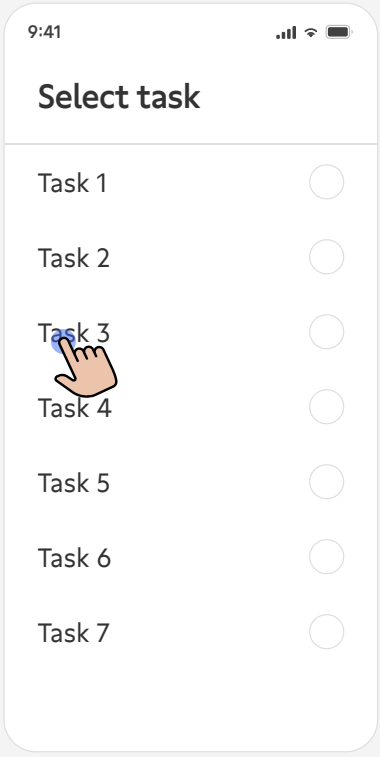
Figma flow list

The moderator drives; the respondent is looking at Figma instead of your app.



Start screen

The respondent drives and never leaves the prototype.



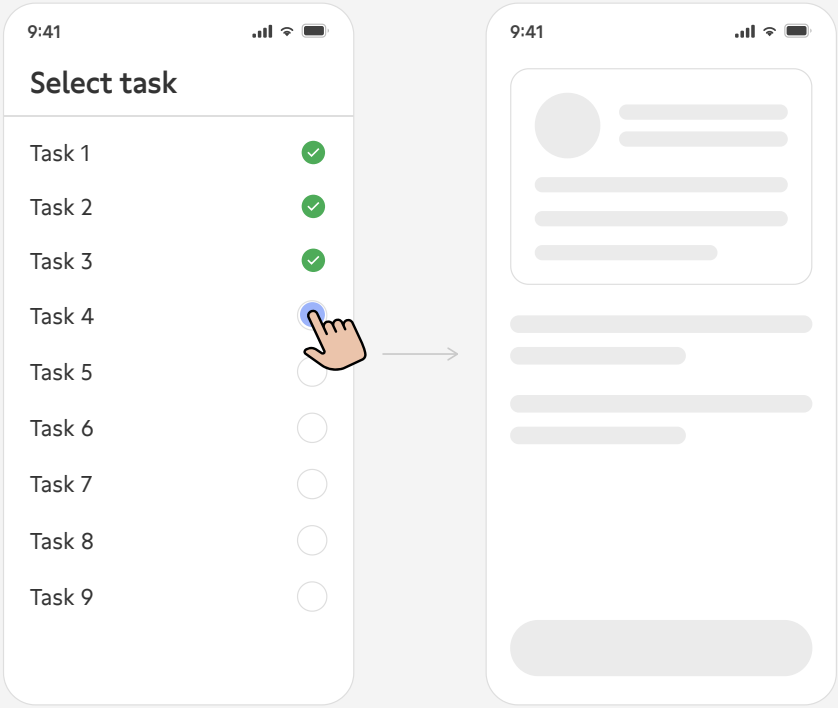
3.1 A start screen listing every task must have

Build a first screen with the tasks numbered 1 to N (numbers only, no titles). Each number links to the first screen of its flow, which you can connect later once the flows exist.

Titles are the thing to resist. A list reading "1. Transfer money · 2. Find your IBAN · 3. Change the card limit" tells the respondent what is coming and lets them plan. Bare numbers give you the navigation without the preview.

Good

Untitled, ordered list of tasks that lead the user to separate flows.



3.2 One-click return to the start screen must have

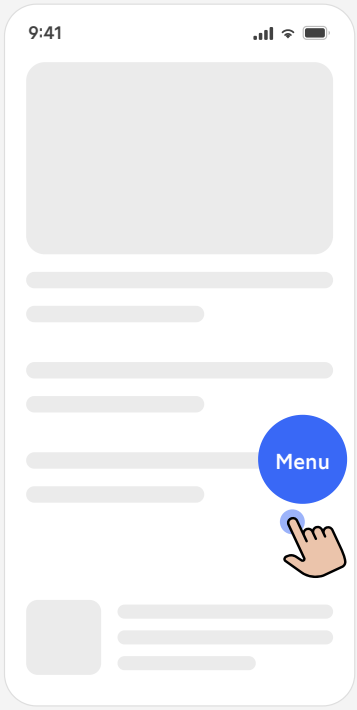
From any screen, 1 tap must return the respondent to the start screen, even from the middle of a flow.

Put the hotspot somewhere that exists in the interface but carries no meaning. On mobile the system clock works well: present on every screen, never tapped on purpose, and takes up no extra screen space.

Do not add a visible "Menu" or "Back to tasks" button. It is a control that does not exist in your product, it sits in view for the entire session, and sooner or later someone taps it mid-task out of curiosity.

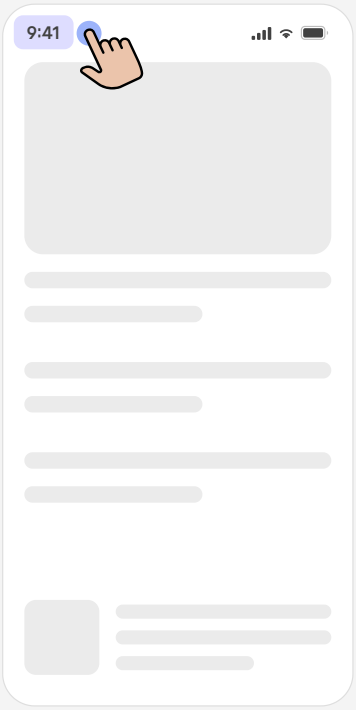
Bad

Distracting element in the interface.



Good

Return hotspot hidden in the system clock.



3.3 A task brief screen before each task optional

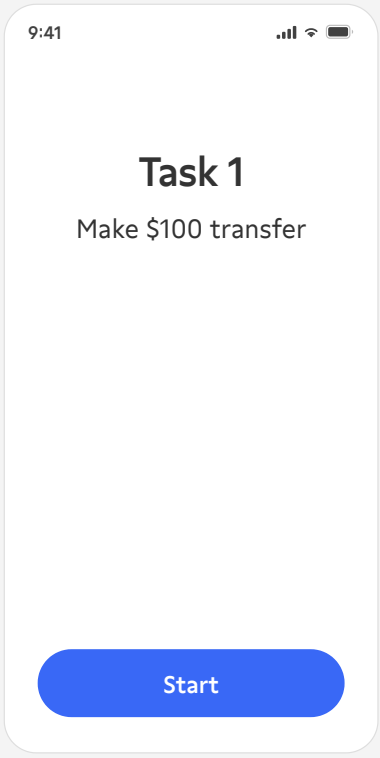
Add a screen stating the task in writing with a "Start" button. It helps participants who prefer reading over listening, lets them re-read at their own pace, and gives you a clean start time (time-on-task begins only when they tap "Start").

Skip it when tasks are short and easy to keep in mind.

Build it when tasks contain exact figures, account numbers, or multiple conditions — details people easily forget midway through.

Good

Start screen with clear task copy and a "Start" button.



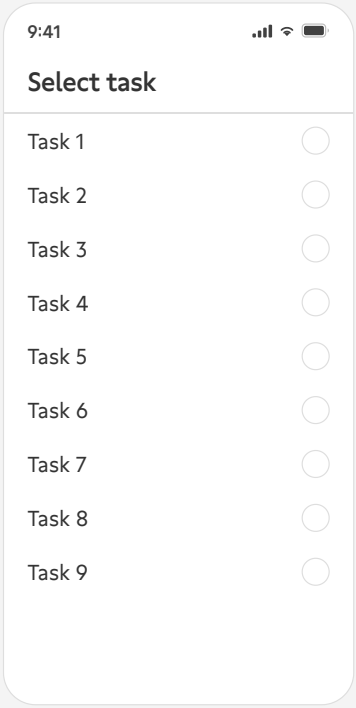
3.4 Mark completed tasks on the start screen optional

Mark finished tasks with a check. Visible progress holds attention and prevents participants from losing track of what they've completed.

Set the state change to happen automatically on the last screen of each flow, or let the moderator advance it. Automating this removes one more distraction for the moderator during the usability session.

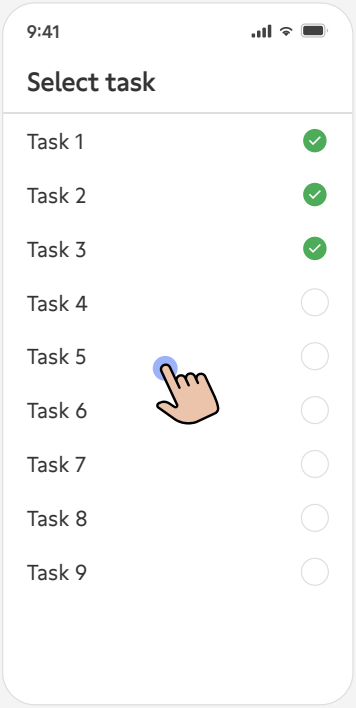
No progress marks

A respondent has to ask which tasks they have already done.



Progress marked

Finished tasks carry a check. Progress is visible and clear.



4. Clickability

Clickability is the ability of interface elements to actually respond to a user's click or tap.

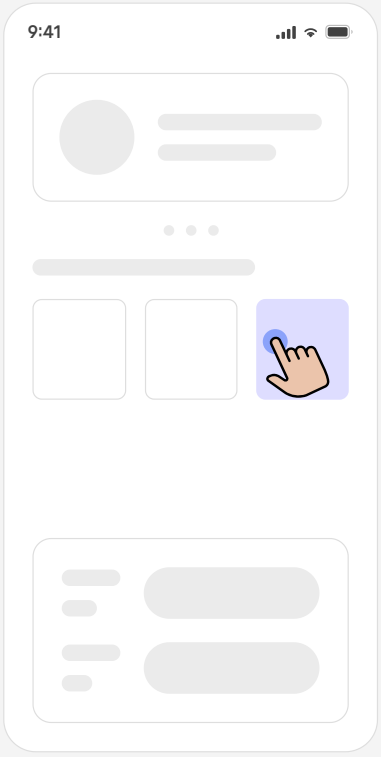
This stems from Figma's default behavior: clicking a non-interactive area highlights all active hotspots. If only one button is interactive, that blue flash becomes a direct cheat code. If 20 elements respond, it turns into background noise that respondents quickly learn to ignore.

4.1 Everything in the research focus is clickable must have

Make every element and entry point related to your test tasks clickable. Teams often connect just 1 happy path, leaving alternative routes broken.

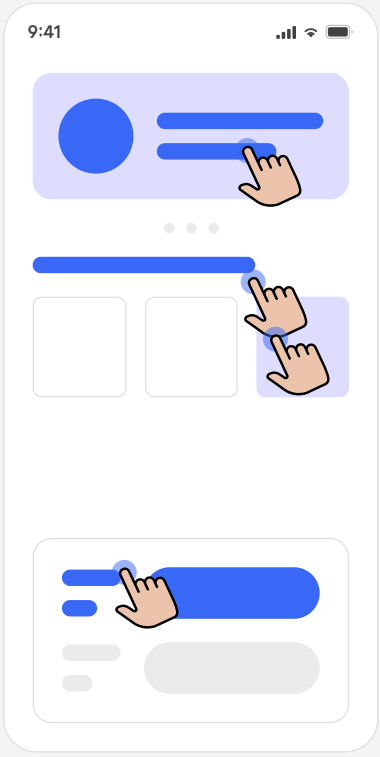
Only one entry point

Only one path works; all alternative routes into the task are unclickable.



Multiple entry points

Multiple entry points into the same feature, matching real app behavior.



4.2 Everything outside the research focus is clickable too must have

Connect every section outside your research focus, giving each one a simple response or placeholder screen. A respondent looking for a feature does not know where your research focus lives (they simply follow their own mental model).

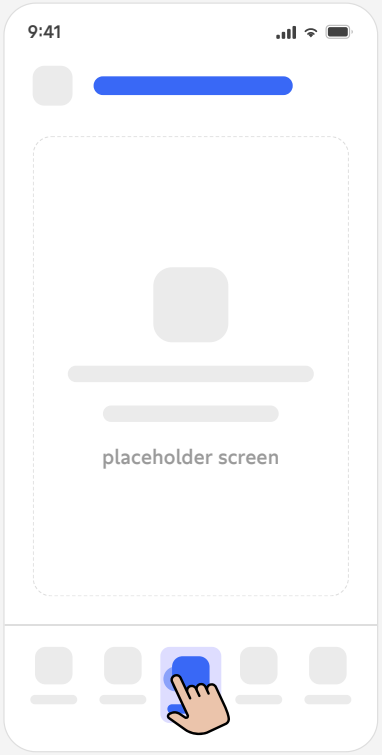
If they open Payments when your task is about Transfers, that detour is a vital finding about your information architecture. Leaving secondary sections inert silently forces them onto the correct path, and you conclude your navigation is clear. Connect those sections so their real first instinct gets recorded instead of corrected.

Secondary sections inert

Only the section under research is wired.

Secondary sections connected

Every section opens something, even a bare placeholder.



4.3 At least 5 clickable zones on every screen optional

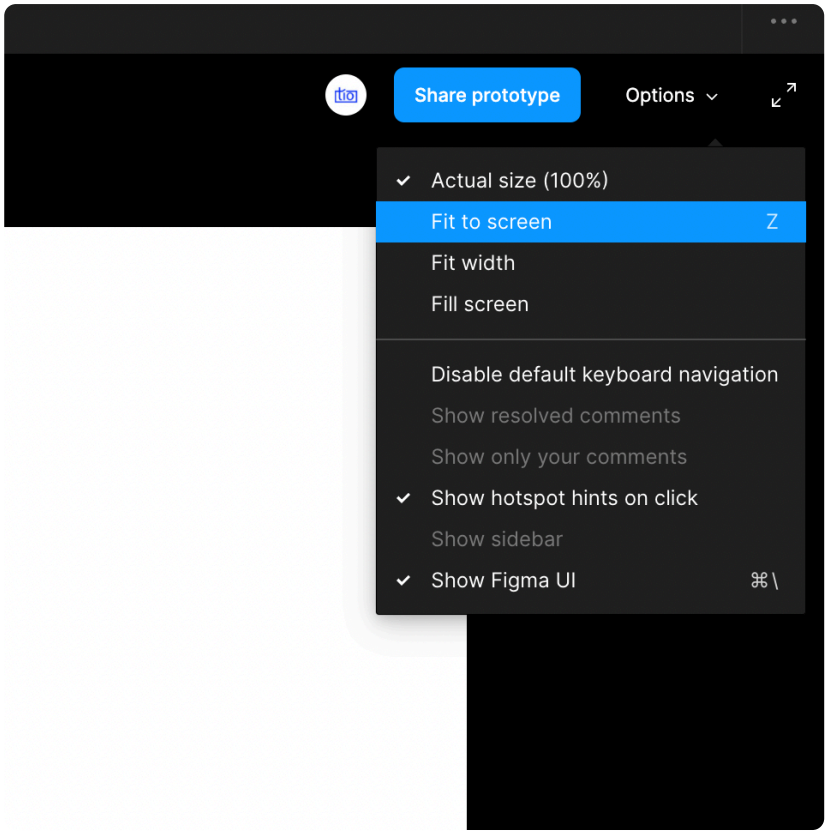
Sparse screens turn Figma's hotspot flashes into unintended answer keys. When a participant misclicks on a frame with only one or two interactive buttons, Figma instantly gives away the correct path.

To prevent this, keep at least 5 distinct clickable zones on every screen, especially simple ones. With 5+ zones, the whole frame lights up at once, turning the flash into background noise that respondents quickly learn to ignore.

Pay special attention to success states, OTP screens, and confirmation dialogs, as teams frequently forget to wire secondary elements there.

4.4 "Show hotspot hints on click" stays on must have

Leave the "Show hotspot hints on click" setting in Figma enabled.



Disabling hints creates a dead end when a participant misclicks. Without visual feedback, they cannot tell if they missed the target, if a feature is missing, or if the screen froze. This confusion frustrates respondents and skews your research data. On web prototypes, where users also expect hover feedback, a non-responsive screen reads as broken.

4.5 Placeholder screens for secondary sections optional

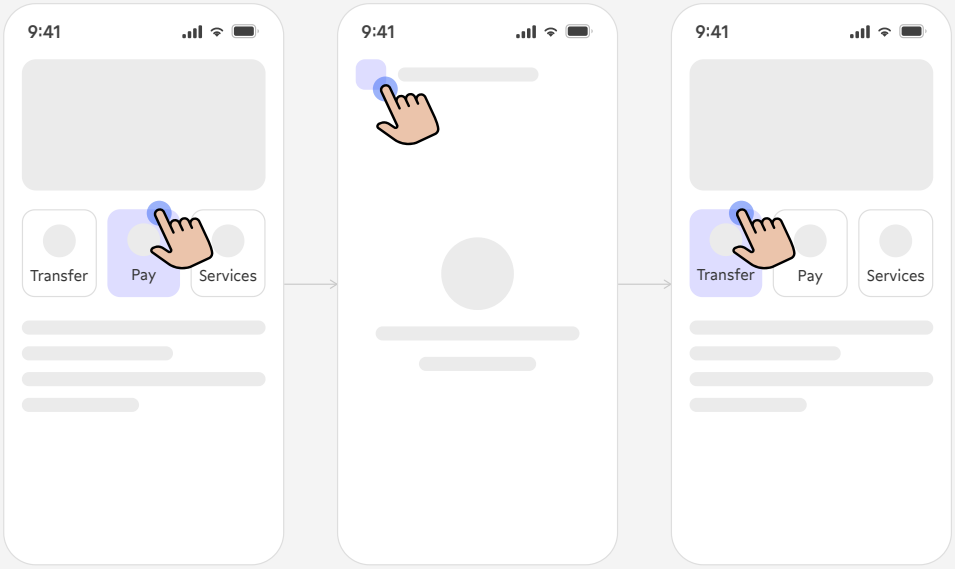
Add a simple placeholder screen for sections participants might open but that fall outside your test scope.

While testing Transfers, a participant looking for a saved recipient might try Payments or Services first. With a placeholder, they open it, realize it's not what they need, and head back — recording a valuable detour. Without one, they tap, receive no response, and you learn nothing.

Note: Every placeholder requires a functional back button so participants aren't trapped. See **section 5.1** for navigation details.

Good

Let respondents explore secondary sections and navigate back smoothly.



4.6 More than 1 route to the same task optional

Connect 2–3 alternative routes to complete the same task. Real products offer several entry points into the same feature, and people reach for different ones depending on how they think about the job.

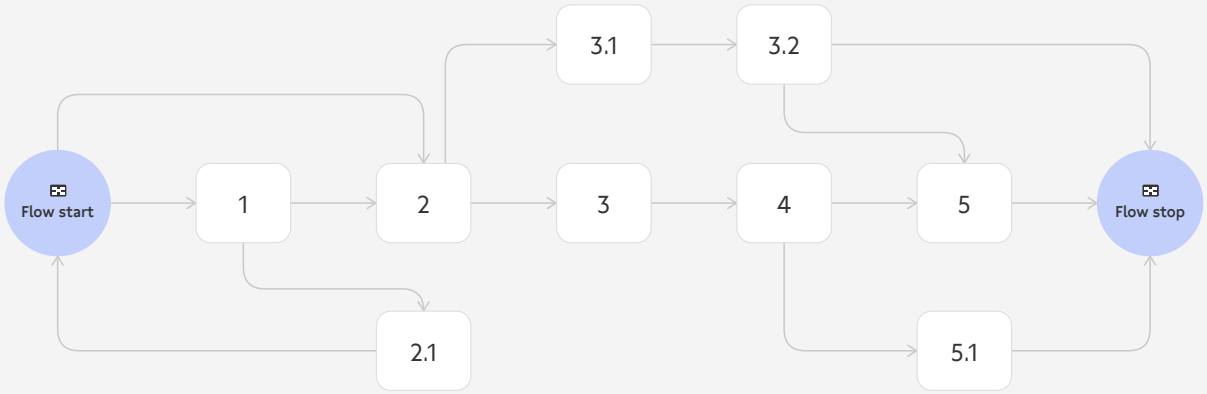
A linear prototype forces everyone down a single connected path and then reports unanimous agreement. Building multiple routes reveals which entry point people naturally reach for in real life.

Linear prototype

The user can accomplish the task in only one way.

Nonlinear prototype

Multiple ways and entry points to accomplish a task.



4.7 The real-world entry point optional

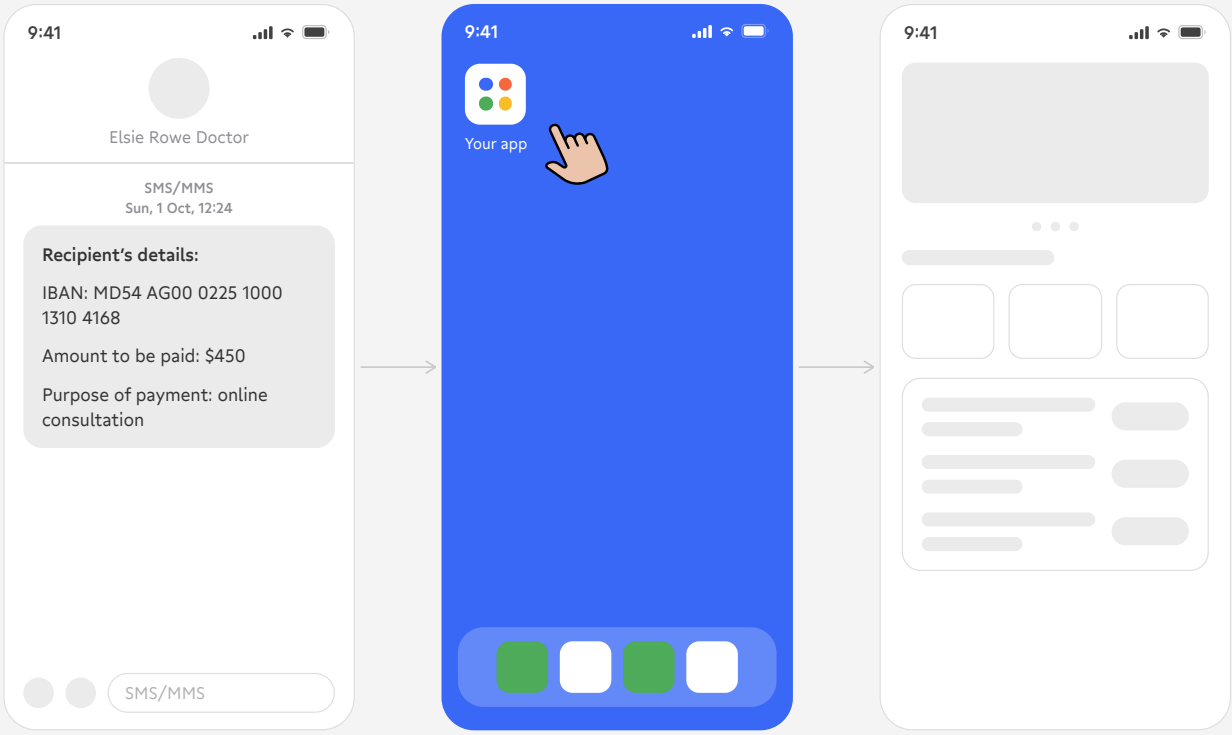
Start the flow where it starts in life. People rarely open an app for no reason. They open it because a friend sent a message, a bill arrived, or a push notification popped up.

If your flow has an origin like that, put it in the prototype: begin the respondent on the SMS, let them read the details, let them open your app themselves. It costs 2 screens and the respondent arrives with the context and intent a real user would have, instead of being asked to imagine.

Good

Flow starts with a message screen to give the user the context of the task.

Example task: Your doctor has sent you the details for paying the bill in a message. Read it and make the transfer.



5. Control: back, scroll, overlays, and notifications

Participants must always be able to retrace their steps. When someone takes a detour, a missing back button turns a natural instinct into a dead end.

Applying Controllability ↗ to your prototype ensures respondents can correct their own course, allowing you to observe how they navigate out of mistakes.

5.1 A working back control on every screen must have

Connect a working back control on every screen, including placeholders, overlays, and confirmation states.

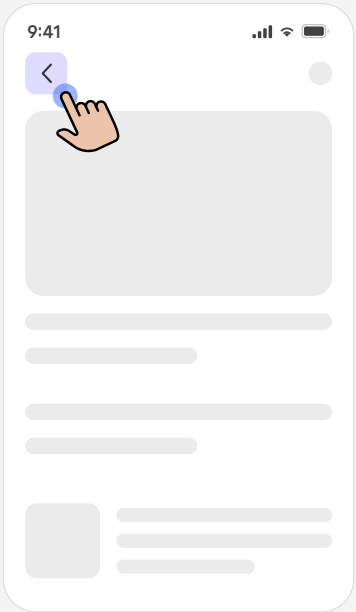
It lets the respondent recover from a wrong turn instead of stalling, and it lets you retrace their route with them: *"you went here, then came back — what were you looking for?"* is a question you can only ask if going back was possible.

It also protects you from the reconstruction problem in **section 1.2** because a respondent who walked back through their actual path can describe what they were thinking.

If your interface has a back control, connect it live on every screen. If it does not (e.g. a full-screen flow, a modal without a close, a system back you cannot render), create an invisible hotspot somewhere inert. The battery icon and the status bar work for the same reason the clock does in **3.2**.

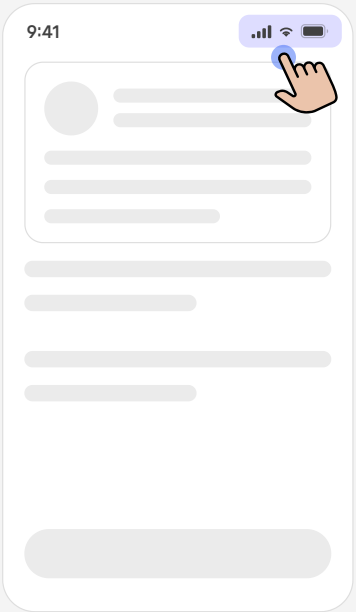
Good

Active back button to move to the previous screen.



Also Good

Navigation back using element embedded in the interface.



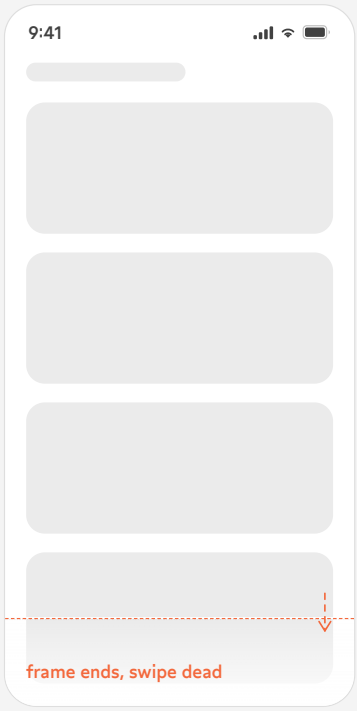
5.2 Scrollable frames actually scroll must have

Set vertical scrolling on any frame taller than the viewport, and pin fixed elements like headers or tab bars.

Taller frames without scrolling create a quiet failure: the screen cuts off cleanly so respondents assume there is nothing below. They stop looking, miss their option, and fail the task. You then log a findability ↗ problem about your product that was actually created by the prototype setup.

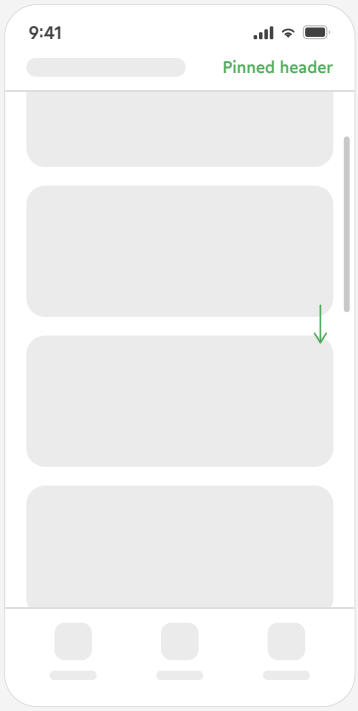
No scroll set

Scrolling does nothing. The cut feels like the end of the page.



Vertical scroll on, header and tab bar pinned

The UI behavior tells the respondent there is more below.



5.3 Overlays open and close the way they will in production must have

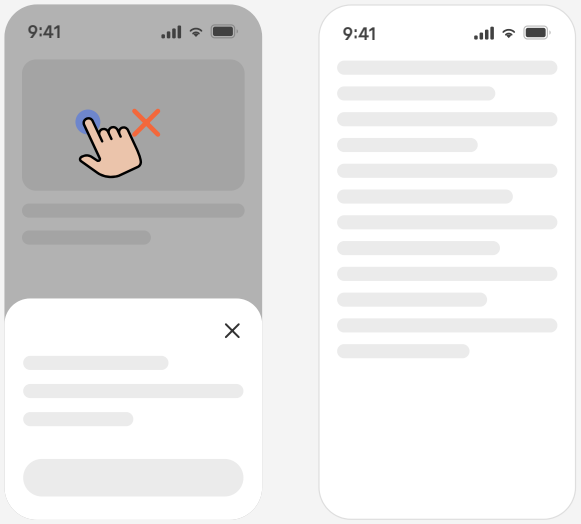
Connect the full interaction behavior of modals, sheets, and overlays.

Always wire the close button, make the background scrim dismissible, and ensure opening an overlay never resets the screen's scroll position.

Check the scrim interaction first. Tapping outside a sheet to close it is an instinctual habit; when a static scrim ignores their tap, respondents immediately assume they are trapped.

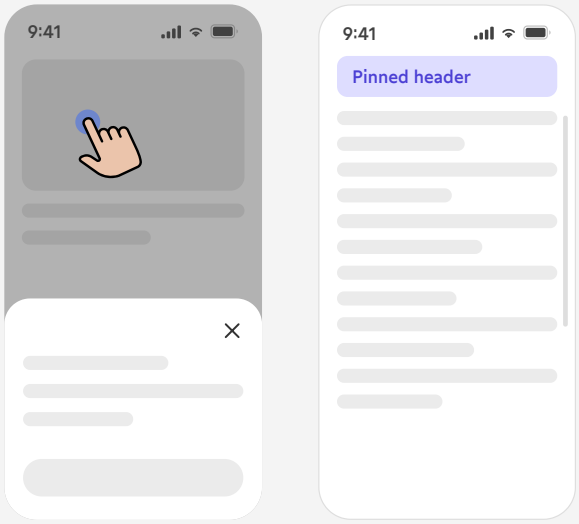
Bad

Tapping the scrim does nothing and the long screen shows no sign that there is more below.



Good

The sheet closes on the scrim, the screen scrolls, and the header stays pinned.



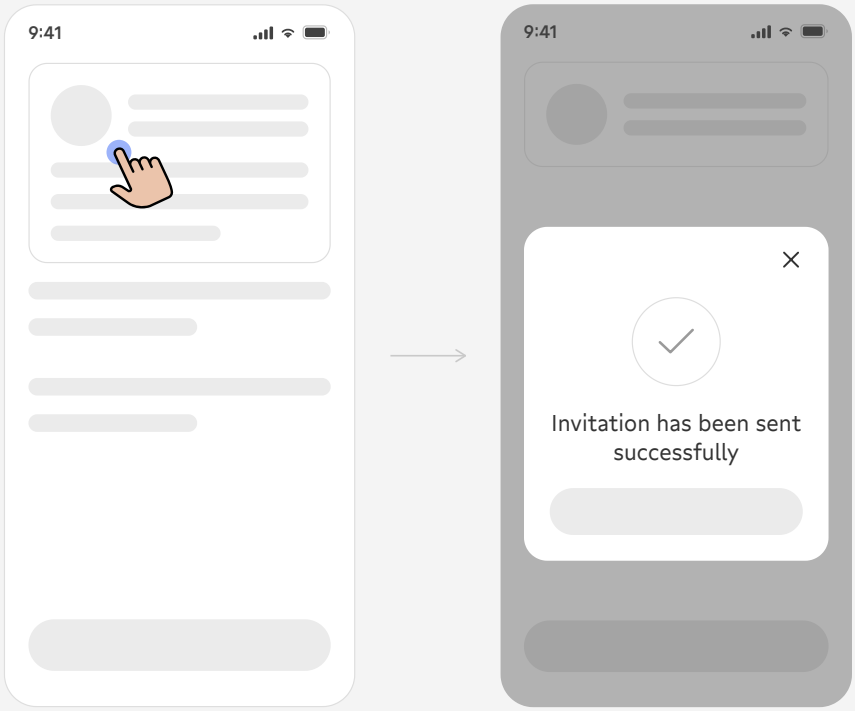
5.4 Notifications fire on a click, not on load must have

Trigger notifications, toasts, status messages, and pop-ups from a deliberate action rather than automatically on arriving at a screen. This is the only place where we recommend making the interaction slightly artificial on purpose.

Automated triggers ruin post-task debriefs. Right as you finish Task 2 and ask for feedback, an auto-firing notification for Task 3 pops up mid-sentence. The participant's attention shifts immediately, forcing you into the next task before you're ready.

Good

Modal window opening as a response to user interaction.



6. Motion, input, and states

Everything so far has been about where the respondent can go. This chapter is about whether the prototype feels like a ready-to-use app.

6.1 Transitions stay fast must have

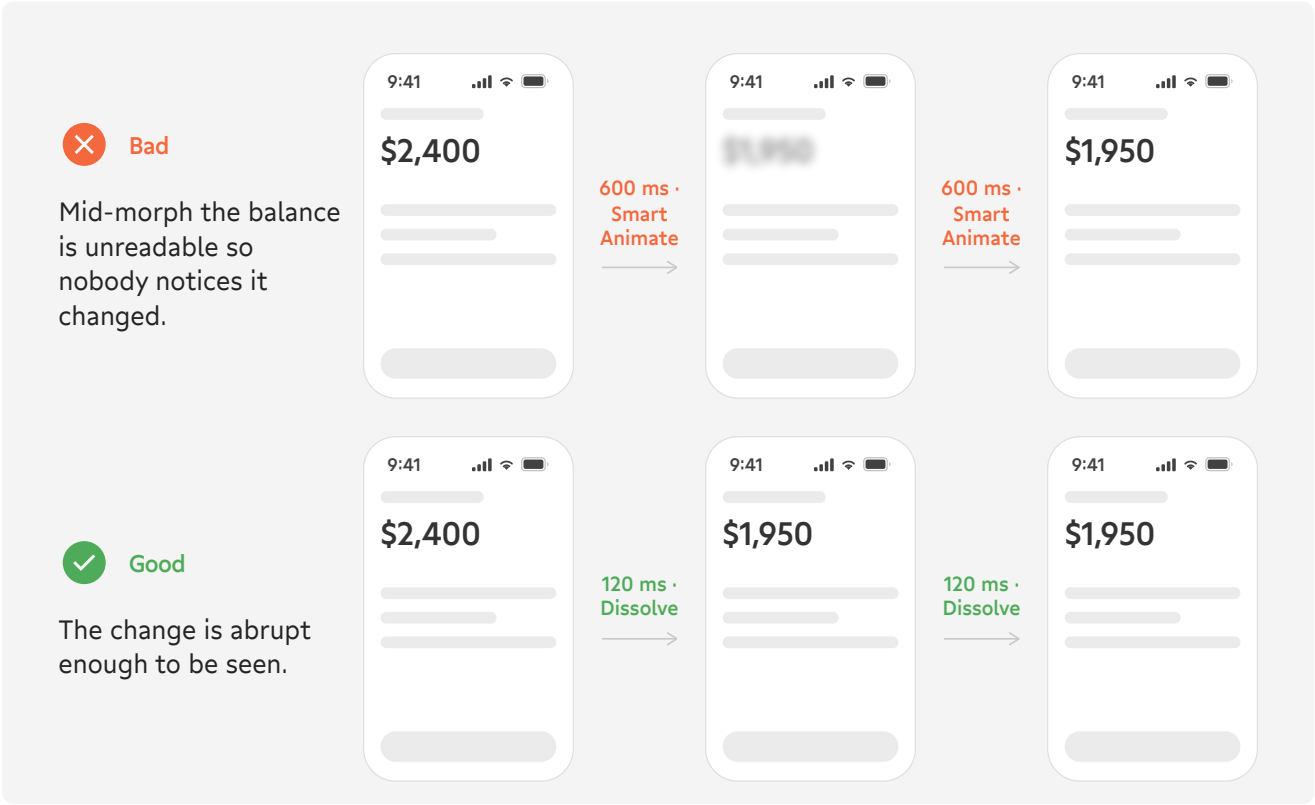
Keep transitions at or under 300 ms and remove every deliberate delay. A 600 ms transition across a 12-step flow adds more than 7 seconds to the task, and all of it lands in your time-on-task figure. Worse, respondents read slow transitions as lag, and someone who thinks the prototype is struggling starts waiting after every tap instead of moving at their natural pace.

6.2 Smart Animate should never hide a state change must have

Avoid Smart Animate. Switch to Instant or Dissolve transitions whenever a value, balance, or toggle state changes.

Smart Animate is the setting most likely to produce a false negative in your report. Smart Animate morphs between states so smoothly that respondents do not register the change. You then record that nobody noticed the confirmation and write a finding about system status feedback ↗ that is really just a defect in your prototype.

Go through every screen pair where a value changes and ask whether the change is still visible at full speed. Use Smart Animate for movement but choose something sharper for meaning.

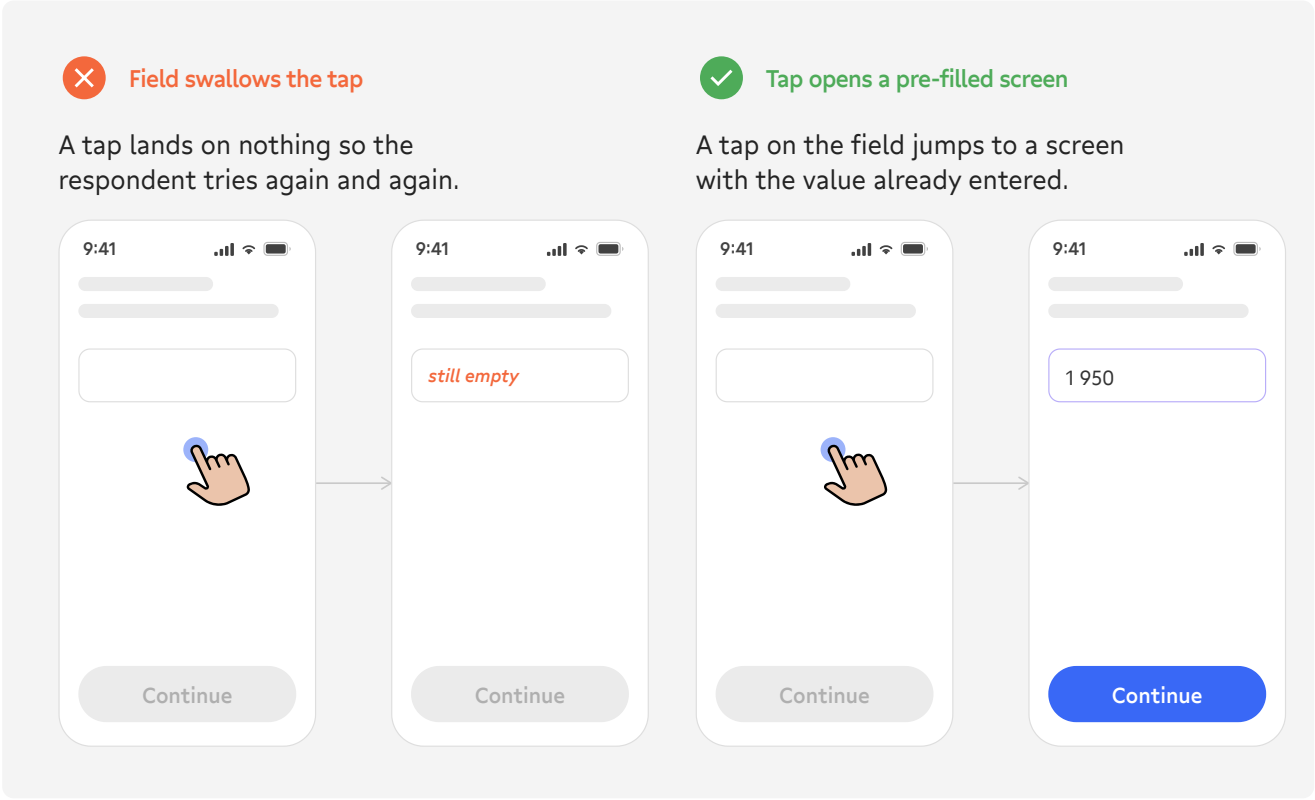


6.3 Typing does something must have

Make every input field respond to typing or taps. Best is a Figma input component that accepts real typing. Where that is impractical, connect a tap on the field to a screen with the value already filled in — artificial but far better than a field that swallows keystrokes.

A field that looks editable, accepts a tap and shows nothing gets tried 3 times before the respondent concludes the prototype is broken and looks at the moderator.

If you go the pre-filled route, say so up front. An expectation set at the start costs 5 seconds; the same discovery mid-task costs you the task.

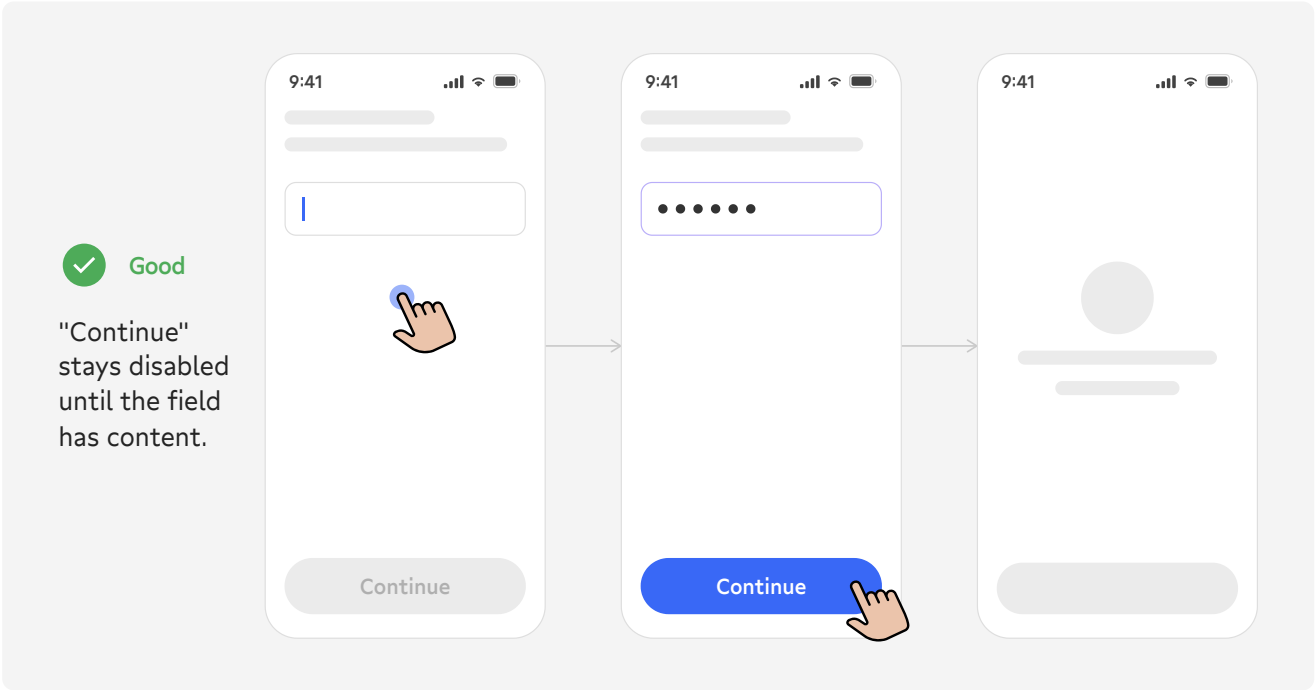


6.4 Button states respond to input must have

Connect the relationship between what the respondent enters and what the interface allows.

- The respondent types the password and taps a "Continue" button that was active the whole time.
- "Continue" starts disabled and becomes active once the field has content — which tells the respondent their input registered.

If your production interface gates the button, gate it in the prototype. This is self-descriptiveness ↗ at the level of a single control.



6.5 One realistic error state optional

Connect at least 1 path where something goes wrong: a declined payment, a failed validation, a recipient not found.

Happy-path-only prototypes tell you how your product performs when nothing goes wrong, which is the situation your users least need help in. Error states are where copy works hardest and trust is won or lost, and they are almost never tested because they are almost never connected. One is enough to learn whether your error copy tells people what to do next.

7. Data and copy consistency

Consistency is total alignment across the prototype: the same terms, names, numbers, and language on every screen.

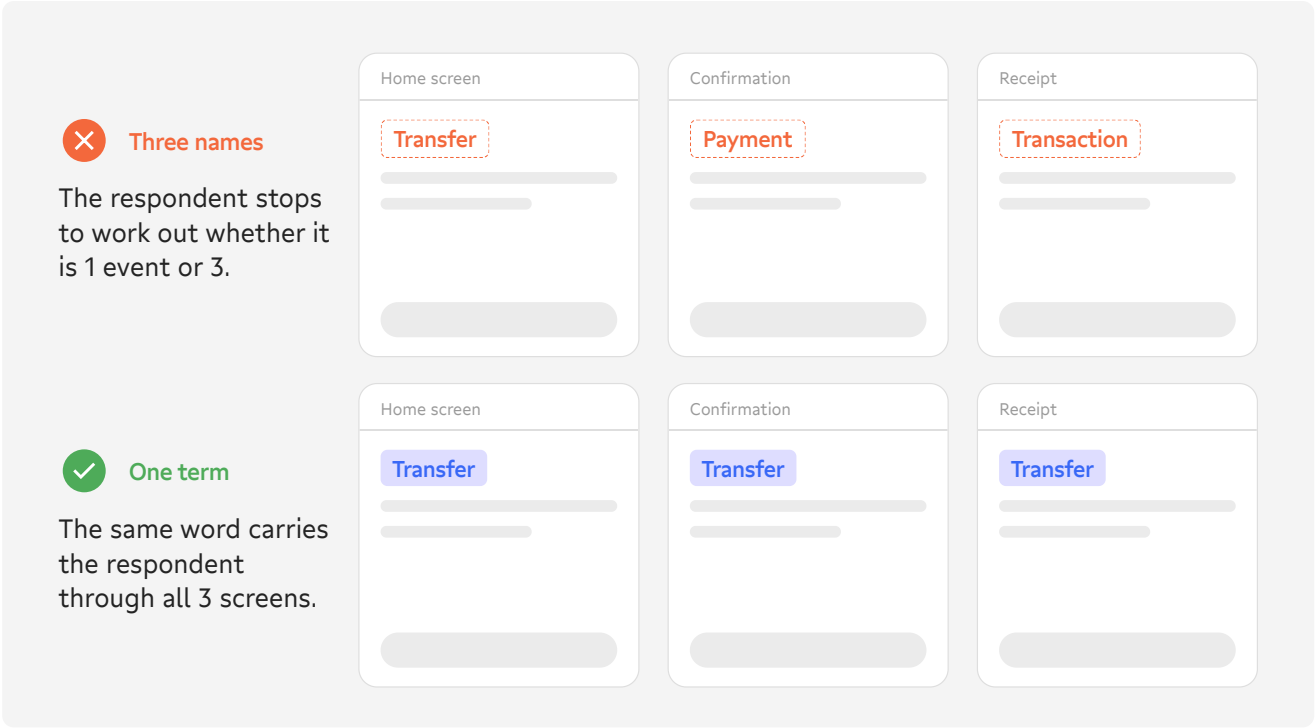
It matters because respondents read far more carefully than the team expects: they have no idea which details are meaningful and which are placeholder, so they treat all of them as real.

Internal consistency ↗ matters at prototype stage for the same reason it matters in production. This is also the chapter where AI does most of the work.

7.1 One term per concept

must have

Pick 1 word for each thing and use it everywhere. If the main screen says "Transfer", the confirmation should not say "Payment" and the receipt should not say "Transaction". Each is defensible alone. Together they make the respondent stop and work out whether all 3 refer to the same event.



AI Tip: Export every string in the file and ask AI to group them by concept and flag concepts named more than 1 way. It surfaces in seconds what a screen-by-screen review misses because inconsistency is only visible when the terms sit next to each other.

7.2 Names match across screens must have

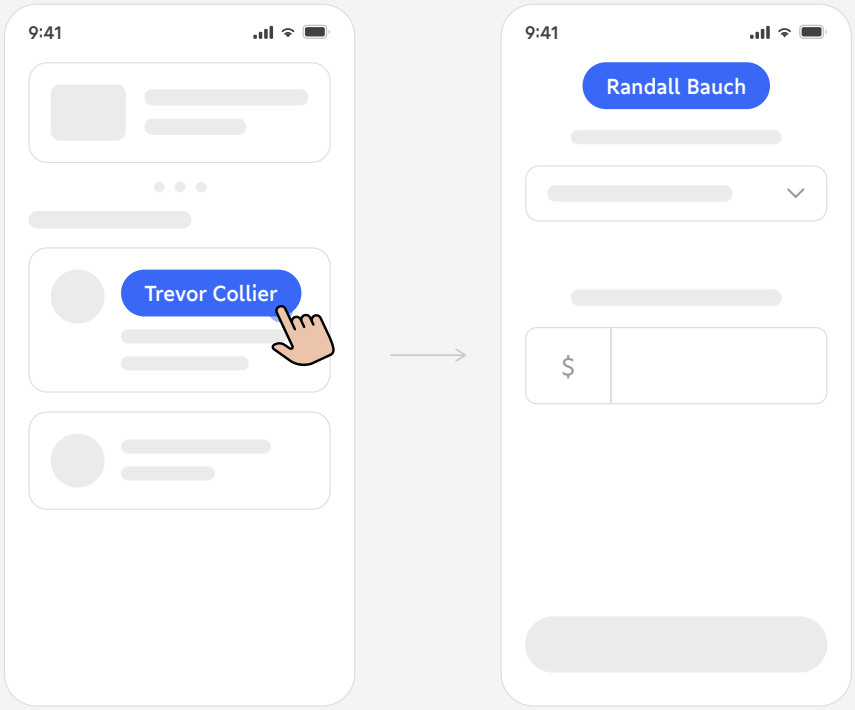
Keep recipient names and user details identical across every screen in a flow.

This is the most frequent content defect in prototypes, usually caused by different teammates reaching for different placeholder names.

If a respondent selects Trevor Collier and lands on a confirmation for Randall Bauch, they do not assume it is filler data. They assume they tapped the wrong row, go back, and hesitate, skewing every metric you record afterward.

Bad

Inconsistent user name.



7.3 Amounts, dates, and balances add up must have

Ensure all amounts, dates, fees, and balances are arithmetically accurate across every screen.

Respondents always do the math, and they immediately spot contradictions like a $450 transfer that leaves a $2,400 balance unchanged.

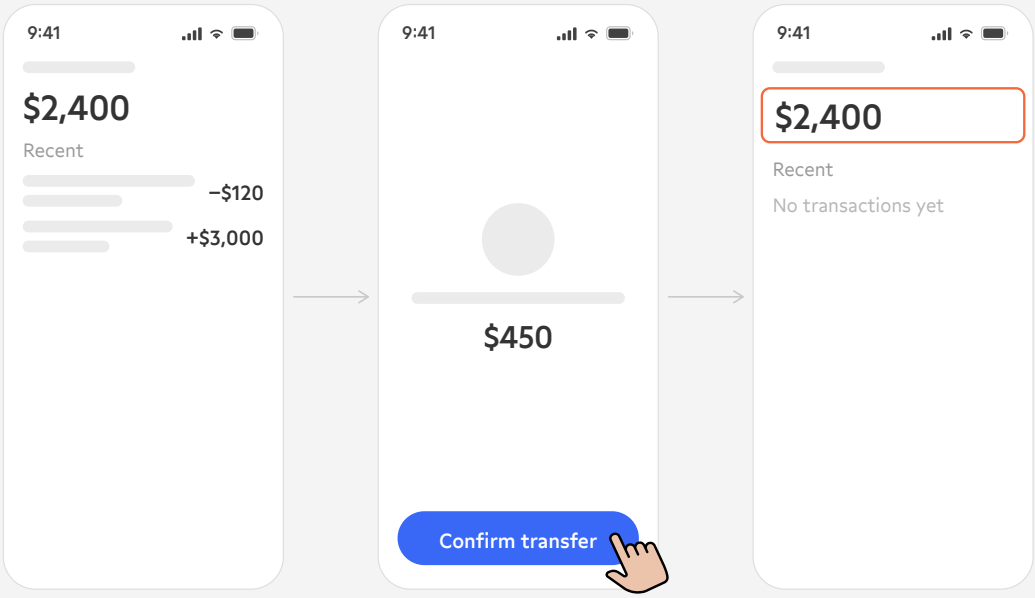
Every action must leave a clear trace in the UI. If a transfer does not update the account balance or recent history, respondents will ask if it succeeded (a question you can no longer distinguish from a real feedback defect).

Bad

The balance did not move and the transfer is missing from the history.

Good

Balance **$1,950**, and the $450 transfer at the top of Recent.



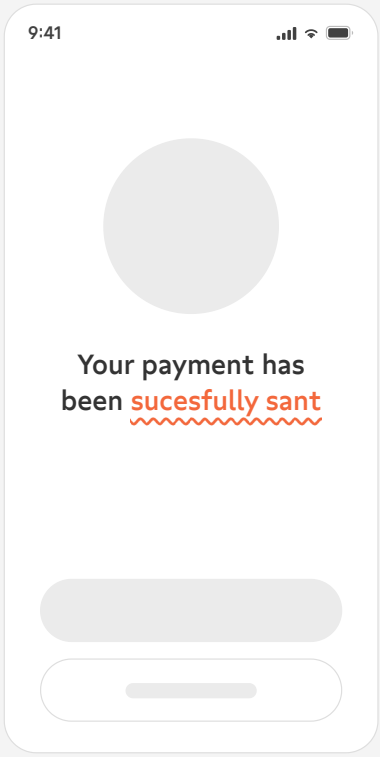
AI Tip: Give AI the screen inventory with every name, amount, date and balance, plus the intended flow order, and ask it to list contradictions. Names and arithmetic across 30 screens is exactly the tedious cross-check people skip and AI does not.

7.4 No spelling or punctuation errors must have

Run all copy through a checker: microcopy, button labels, empty states, error messages, and anything inside components. The respondent cannot tell a typo from a deliberate test of their attention. Some stop and point it out; some quietly downgrade their opinion of the whole product. Either way you spent session time on a defect that takes 30 seconds to fix.

Bad

Spelling mistakes in the prototype.



AI Tip: Paste the full string export and ask for spelling, punctuation and register inconsistencies in 1 pass. Catches the strings a spellchecker inside Figma never sees because they live inside components and variants.

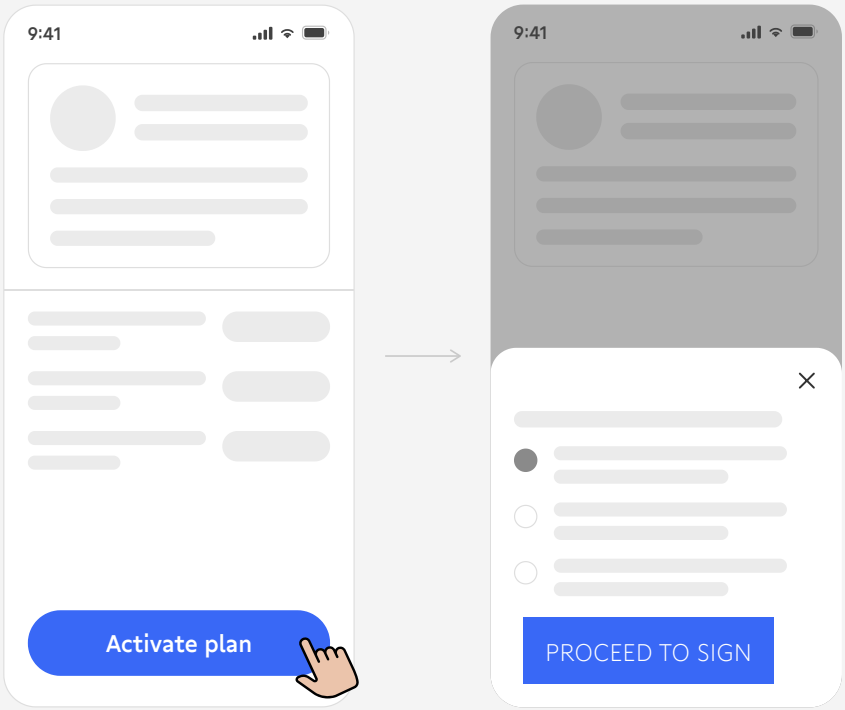
7.5 Components formatted consistently optional

Check that the same component looks the same everywhere: buttons on the same margins, cards with the same corner radius and padding, lists with consistent row heights.

Misalignment reads as sloppiness, and a respondent who has decided the product is unfinished starts reporting cosmetic issues instead of doing the task. The session turns into a design review.

Bad

Inconsistent button design.



7.6 The prototype speaks the respondents' language must have

Write every string in the language you recruited in, including assets that come from outside your design system. This breaks on borrowed assets: a screenshot of an SMS thread, an email mock, an OS dialog, a screen pulled from another project. They carry their own locale (a date format, a day-of-week abbreviation, a system label).

AI Tip: Ask AI to scan the string export for anything not in the target language, and to flag locale-formatted dates and numbers. This is the check most often skipped and the one AI closes most completely.

8. Device, access, and the dry run

A built prototype is not a test-ready prototype. This chapter covers the pre-flight checks that prevent technical glitches, broken permissions, and wasted interview time.

8.1 The prototype device matches the test device must have

Set the prototype's device preset to the hardware the session will use, and check the scaling option: Fit width, Fit height, Actual size, and Fit to screen produce different results, and the default is not always the one you want.

You cannot redraw frames for each respondent, and you do not need to. Build at the size your audience actually holds; when respondents join on their own hardware, take the most common size, then open the link once on a smaller phone to confirm nothing important drops below the fold. A prototype drawn at 390×844 and opened on a 360×800 screen scales, and scaling changes what fits on the first screen.

8.2 Open it on the actual device must have

Open the share link on the physical device the session will use and walk every task on it.

Desktop preview does not tell you what a thumb can reach, whether an 11px caption is legible at arm's length, whether your hidden status-bar hotspot is large enough to hit, or whether the file stutters on a mid-range phone. Each of those is a 5-minute fix the day before and an unrecoverable problem during the session.

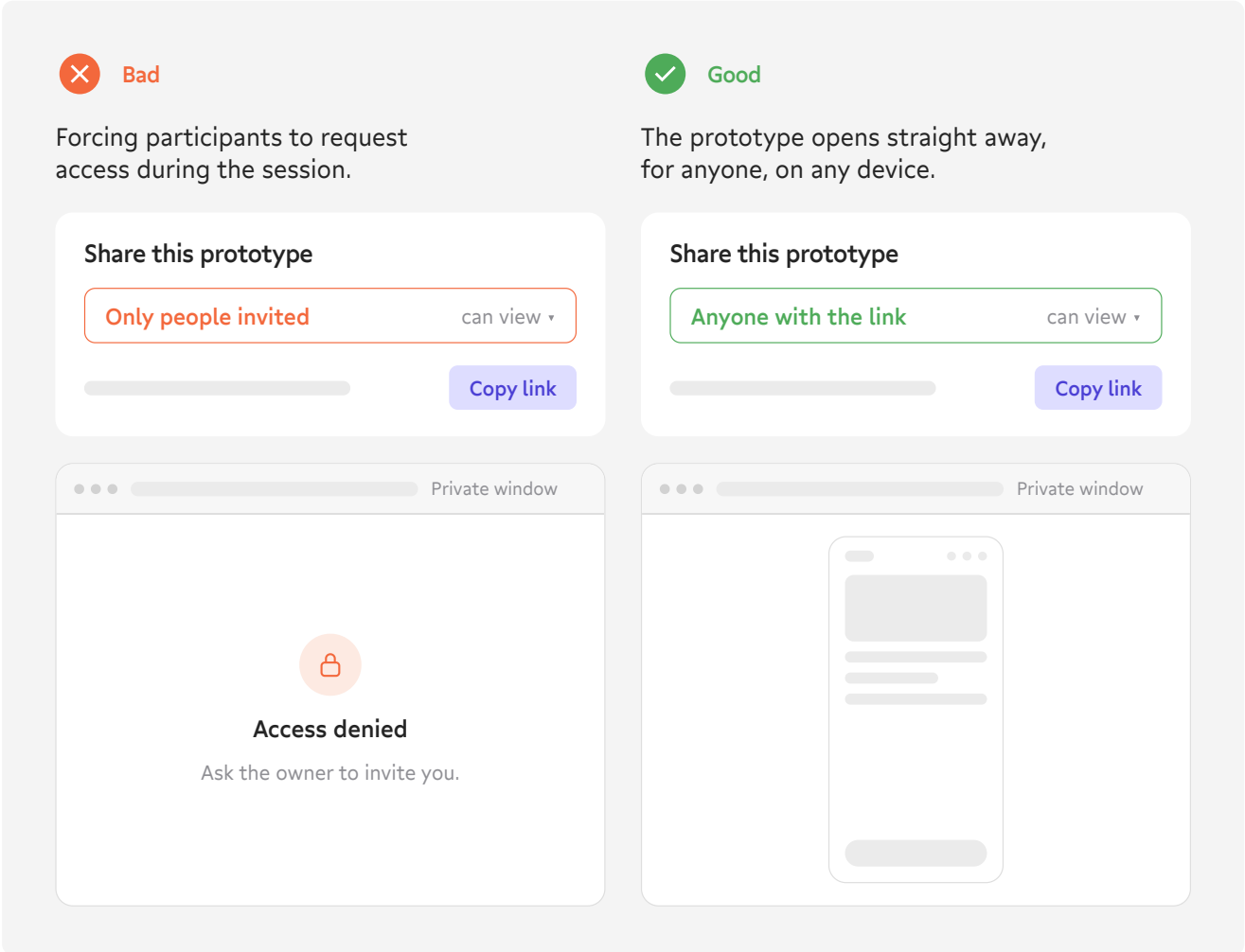
If sessions are remote and respondents use their own hardware, test on the oldest and smallest device you can find.

8.3 Share settings let the respondent in

must have

Set the link to "Anyone with the link" with view access, then open it in a private/incognito window with no Figma account signed in.

A link that works for you proves nothing — you are signed in and you own the file. This is the most common failure on session day and the most avoidable: a respondent who lands on a login wall has to be talked through it by a moderator who did not plan for it, and the session opens with the respondent feeling something has gone wrong.



8.4 The dry run must have

Run 1 full session with someone who has never seen the prototype — a colleague from another team, someone from marketing, anyone outside the project.

You cannot do this yourself.

After a day of building you know where every hotspot is and your hands go to the right places without consulting your eyes. You will walk the flow cleanly and conclude it works. Someone who has not built it finds the dead end in 90 seconds. Watch without helping. Every time you want to say "no, tap the other one" is a finding — either something to fix, or something to expect on the day.

8.5 The handover pack must have

Send the research team the prototype and a document alongside it. Format does not matter — FigJam, Notion, a Google Doc, a spreadsheet. Content does:

- The task list, worded the way the moderator will read it out.
- The intended route for each task, and the alternative routes you connected.
- Your hypotheses: what you expect to happen and where you see people struggle.
- Known limitations: which sections are placeholders, which fields are pre-filled, what does not work.
- The return-to-start hotspot and the hidden back hotspots, with their locations.

That last one is a must in practice. A moderator who does not know your return hotspot is the system clock will reload the prototype instead of using it.

> ✦ **AI Tip:** Feed it the flow map and your task list and have it draft the pack (routes, limitations, hypotheses), then correct it. Drafting this document is the step teams skip because it is dull, and a corrected draft takes 15 minutes instead of an hour.

The more the research team knows before the first session, the more of the session goes into your product, and the better the probes they can design.

8.6 Changes during fieldwork are logged must have

Log and report every prototype modification to the research team immediately.

Tell the research team every time, with 3 things:

- What changed.
- Which screen.
- Which session number it took effect from.

Never edit a prototype silently during fieldwork.

Fixing an error mid-testing is completely fine but changing things silently ruins your findings. Without a log, earlier and later respondents tested different products while your analysis treats their data as comparable.

Explicit logging lets researchers segment findings cleanly so mid-fieldwork adjustments cost you nothing.

The Prototype Readiness Check

27 must-have checks and 8 optional · for moderated usability testing in Figma

Craft Innovations
craftinnovations.global

Must-haves are pass or fail. A prototype missing any of the 27 is not ready, however good the rest of it is. **The 8 optional checks are depth** — score them out of 8 and spend your next hour on the lowest one.

BEFORE YOU BUILD

- [ ] **1.** Preparation time is booked as a task, not squeezed in
- [ ] **2.** Testing runs from a duplicated, named, frozen copy
- [ ] **3.** Comments, annotations, TODO layers and lorem ipsum are removed

START SCREEN

- [ ] **4.** A start screen lists every task, numbered 1 to N, no titles
- [ ] **5.** 1 tap returns the respondent to the start screen from anywhere

CLICKABILITY

- [ ] **6.** Every element in the research focus responds
- [ ] **7.** Every element outside the research focus responds
- [ ] **8.** "Show hotspot hints on click" is enabled

CONTROL

- [ ] **9.** A working back control on every screen, including placeholders and overlays
- [ ] **10.** Screens taller than the viewport scroll, with fixed elements pinned
- [ ] **11.** Overlays close on their control and on the scrim
- [ ] **12.** Notifications and pop-ups fire on a click, not on load

MOTION, INPUT, STATES

- [ ] **13.** Transitions are 300 ms or under, with no deliberate delays
- [ ] **14.** No Smart Animate transition hides a change in a value or state
- [ ] **15.** Every field a respondent must fill responds to input
- [ ] **16.** Button states respond to input the way they will in production

DATA AND COPY

- [ ] **17.** 1 term per concept, everywhere
- [ ] **18.** Names match across every screen in the flow
- [ ] **19.** Amounts, dates and balances are arithmetically consistent
- [ ] **20.** Actions leave a trace: history, balance and recents all update
- [ ] **21.** Copy is spell-checked, including microcopy and component text
- [ ] **22.** Every string is in the respondents' language, including borrowed assets

DEVICE AND ACCESS

- [ ] **23.** Prototype device and scaling match the test device
- [ ] **24.** The prototype has been opened and every task walked on the physical test device
- [ ] **25.** The share link opens in a signed-out private window
- [ ] **26.** A dry run with someone outside the project has passed
- [ ] **27.** The handover pack has gone to the research team

THE 8 OPTIONAL CHECKS

- [ ] **1.** A task brief screen before each task, with a Start button
- [ ] **2.** Completed tasks marked on the start screen
- [ ] **3.** At least 5 clickable zones on every screen, including the sparse ones
- [ ] **4.** Placeholder screens behind secondary sections
- [ ] **5.** More than 1 route connected to the same task
- [ ] **6.** The real-world entry point included — SMS, email, notification
- [ ] **7.** At least 1 realistic error state
- [ ] **8.** Components formatted consistently across screens

Score: 27/27 must-haves is the entry requirement. 0–2 optional functional · 3–5 good · 6–8 you will learn things the task list did not ask about.

About Craft Innovations

Craft Innovations is a global customer research and UX design firm dedicated to helping financial industry leaders innovate faster and deliver exceptional digital and physical customer experiences.

Research services:

- Discovery and product-market fit research
- Customer Experience and UX audit
- Benchmarking, CVP testing and validation
- Usability testing

Design services:

- UX/UI design: mobile, web banking, websites
- Product Value Proposition design
- Gamification design
- AI UX design

Segments:

- Retail & Corporate banking
- Landing / BNPL / Wallets / PFM
- Investech & Insuretech
- Crypto

Let's start with a friendly talk!
Book an intro meeting ↗

CRAFT INNOVATIONS

Visit website

bizdev@craftinnovations.global

+1437 4214992

REVIEWED ON Clutch ★★★★★ 20 REVIEWS