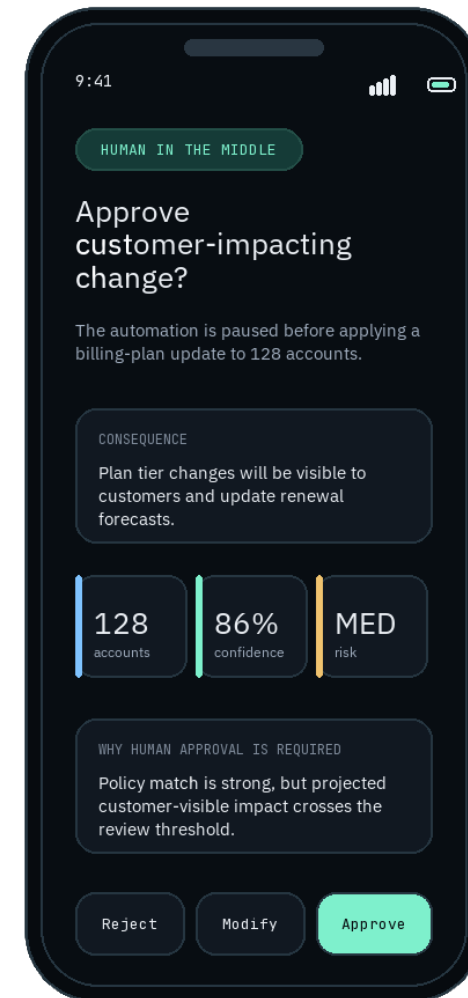


GLASSBOX

A transparent automation interface for workflows that must be inspectable, interruptible, and reversible.

Human-in-the-middle automation UX

AI workflows should expose intent before action, show progress while running, pause at consequential moments, and preserve a complete audit trail.



Executive Summary

GLASSBOX is a concept app for transparent automation. Its core claim is that AI-driven workflows should not behave like opaque magic. They should show intent before action, expose progress while running, pause at consequential moments, and preserve a complete audit trail of both human and machine decisions.

CHALLENGE Make automation legible before it acts. AI agents can move faster than users can inspect. GLASSBOX slows down only where consequence begins.	ROLE Product framing, UX model, screen architecture. The work defines the state model, semantic color rules, hi-res screen direction, and handoff logic.	OUTCOME A six-screen hi-res editable Penpot PoC. Dashboard, Plan Preview, Live Execution, Checkpoint Gate, Modify View, and Completion are rebuilt as layered screens.
--	--	--

- Brand blue identifies GLASSBOX and active system progress.
- Green marks safe/completed/reversible outcomes.
- Orange marks human attention, consequence, and review-required states.
- Red is reserved for rejection, removal, or destructive control.

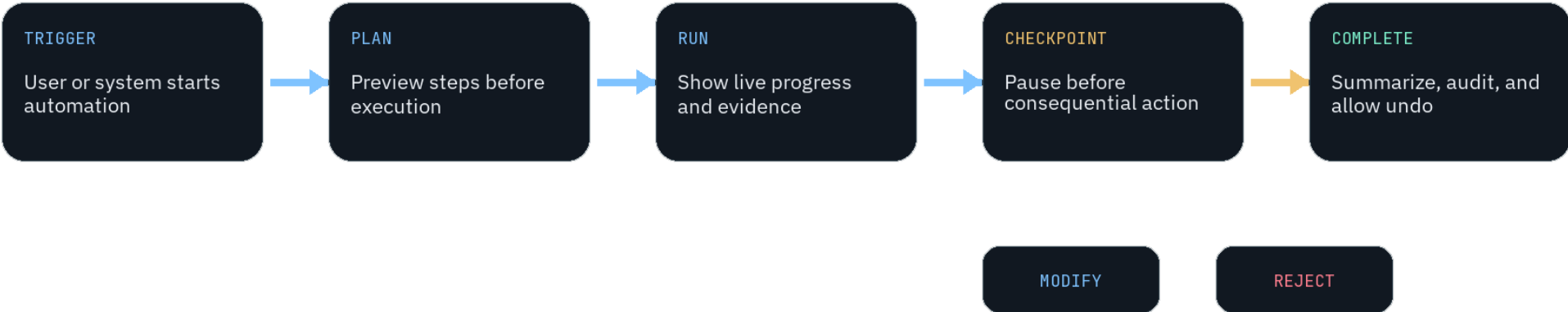
Problem Framing

The common promise of AI automation is that work becomes invisible. For consequential workflows, invisibility is the problem. When a system can modify code, change account state, send messages, or apply business rules, users need to understand what will happen before it happens.

RISK 01 Invisible intent Users often see output without understanding the plan that produced it.	RISK 02 Unclear consequence Automations can cross from harmless analysis into customer-visible or irreversible action.	RISK 03 Weak reversibility Many tools treat undo and audit as secondary, even though they are core trust mechanics.
--	--	---

Automation Contract

The product model is intentionally simple: reveal the plan, run visibly, pause at consequence, then preserve evidence. The interface is not an assistant chat. It is closer to an observability surface for automated work.



User Model

PERSONA A Operations lead Needs confidence that automations will not make customer-visible changes without approval.	PERSONA B Engineering manager Needs to inspect generated refactors, diffs, and test outcomes without reading raw logs.	PERSONA C Compliance reviewer Needs complete evidence of who approved what, when, and why.
--	--	--

The shared need is not more speed. It is accountable speed: automation that can explain its next action and wait when the stakes change.

Design Principles

PRINCIPLE 01 Plan before action The system must expose its intended steps before execution starts.	PRINCIPLE 02 Progress over animation Running work gets concrete status, file paths, and step state rather than vague motion.	PRINCIPLE 03 Consequence before choice At the gate, the user sees consequence, confidence, and rationale before action buttons.
PRINCIPLE 04 Correction is first-class Modify is not hidden behind rejection. It is a primary path in human-in-the-middle design.	PRINCIPLE 05 Audit after completion Completion does not end accountability. It reveals summary, tests, undo, and evidence.	PRINCIPLE 06 Semantic color only Blue, green, orange, and red are reserved for meaning rather than decoration.

State Machine

A production build should start from the state model, not from screen styling. The visual system exists to make the state transitions inspectable.

State	Primary user question	Allowed actions
Plan Preview	What will the system do?	Modify, Start, Cancel
Live Execution	What is happening right now?	Pause, Details, Stop
Checkpoint Gate	Should this consequential change proceed?	Reject, Modify, Approve
Modify View	What needs correction?	Prev, Next, Done
Completion	What happened and can it be reversed?	Audit Log, Undo All, Done

Implementation Notes

- Build the plan/execution/checkpoint/completion states as explicit application states.
- Store audit events for system actions and human approvals as separate event types.
- Treat Modify as a continuation of approval, not as an error path.
- Keep undo visible wherever completed automation can affect user or customer state.
- Avoid language that implies magic, autonomy, or hidden decision-making.

The product is successful when the user can answer three questions at every point: what is happening, why is it happening, and what control do I still have?

Completion

Closure surface for [GLASSBOX](#): summarize what happened, show measurable outcomes, keep tests visible, and preserve audit and undo paths.

COMPLETE

UNDO AVAILABLE

UX DECISION

Completion is not just a success message. It closes with measurable outcome, test status, review time, audit access, and an undo path, preserving accountability after the automation finishes.



UI DECISION

The success mark is calm and green, but the summary remains data-heavy. Audit Log and Undo All are kept as first-class actions so reversibility and traceability remain part of the screen.

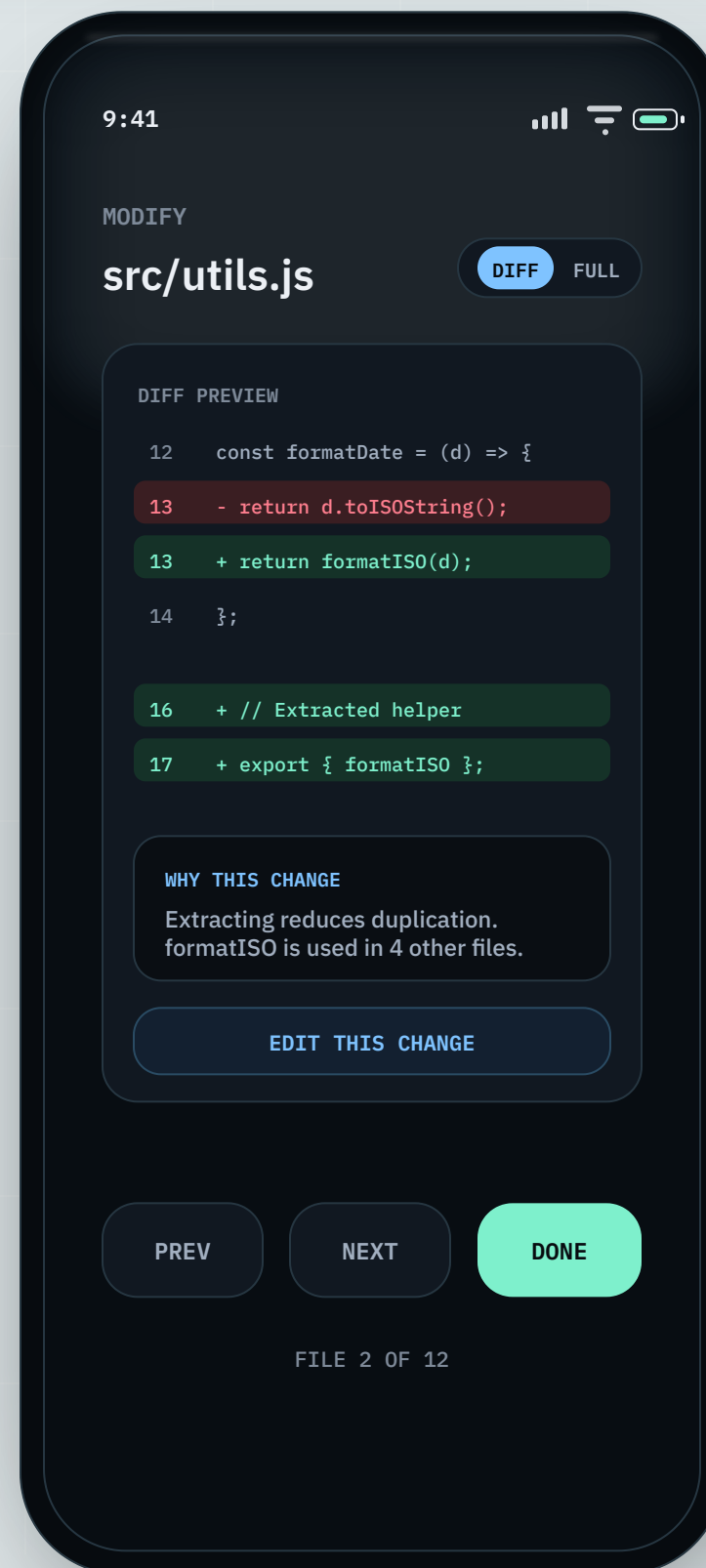
Modify View

Inline correction surface for [GLASSBOX](#): inspect proposed diffs, understand the system reason, edit the change, and resubmit the safer version.

INLINE DIFF

UX DECISION

Modification keeps the user inside the decision flow instead of sending them elsewhere. The diff, system rationale, edit affordance, and file navigation remain together so correction feels native.



UI DECISION

Diff colors are reserved for meaning: red removes, green adds. Blue identifies the active mode and edit affordance, while Done uses green with dark text to confirm the corrected proposal.

Live Execution

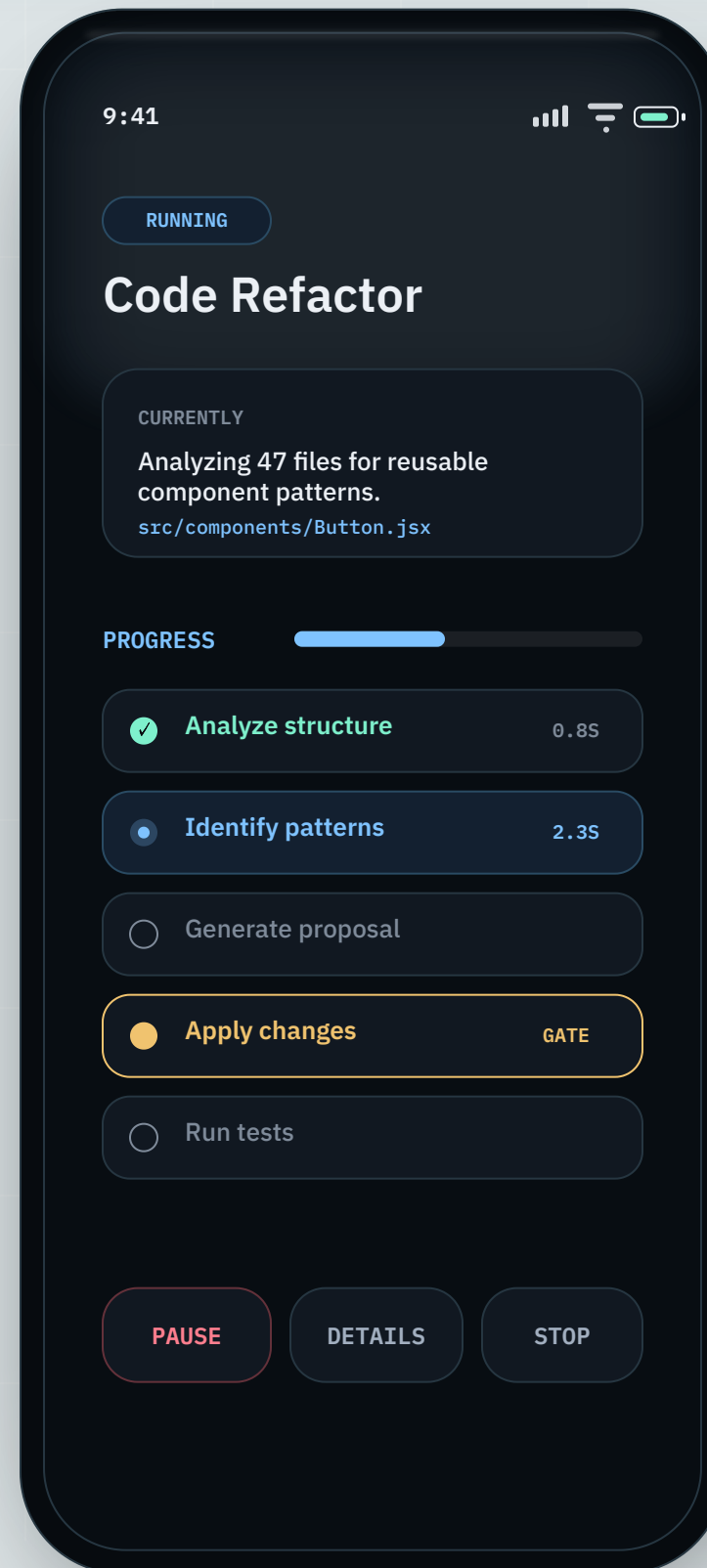
Visible system activity for [GLASSBOX](#): show the current operation, progress through the approved plan, and preserve pause/detail/stop controls while automation is running.

RUNNING

INTERRUPTIBLE

UX DECISION

Execution stays visible while it runs. The screen names the current operation and active file path, so the user can understand progress without waiting for a final explanation.



UI DECISION

Completed work turns green, the active step turns blue, and the upcoming gate remains orange. Pause, Details, and Stop stay available as equal controls, reinforcing that automation is interruptible.

Plan Preview

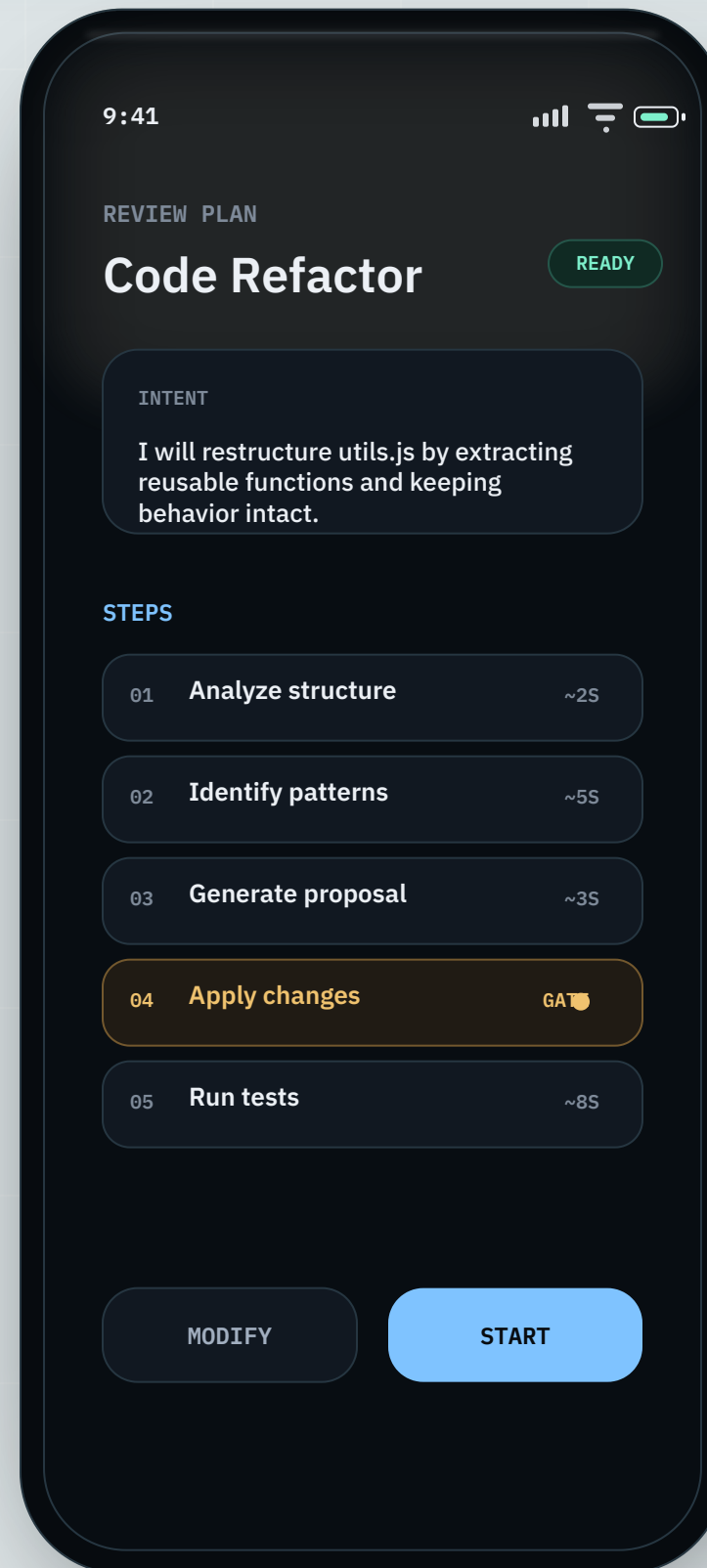
Pre-run approval surface for **GLASSBOX**: show intent, ordered execution steps, and the future checkpoint before the user starts automation.

PRE-RUN

CHECKPOINT VISIBLE

UX DECISION

Before execution starts, the user sees the system's intent and the ordered plan. This changes automation from a black-box command into an inspectable proposal that can be modified before impact.



UI DECISION

The step list is compact, numbered, and scannable. The future gate is already orange in the plan, so the user learns where **GLASSBOX** will stop before any consequential change is applied.

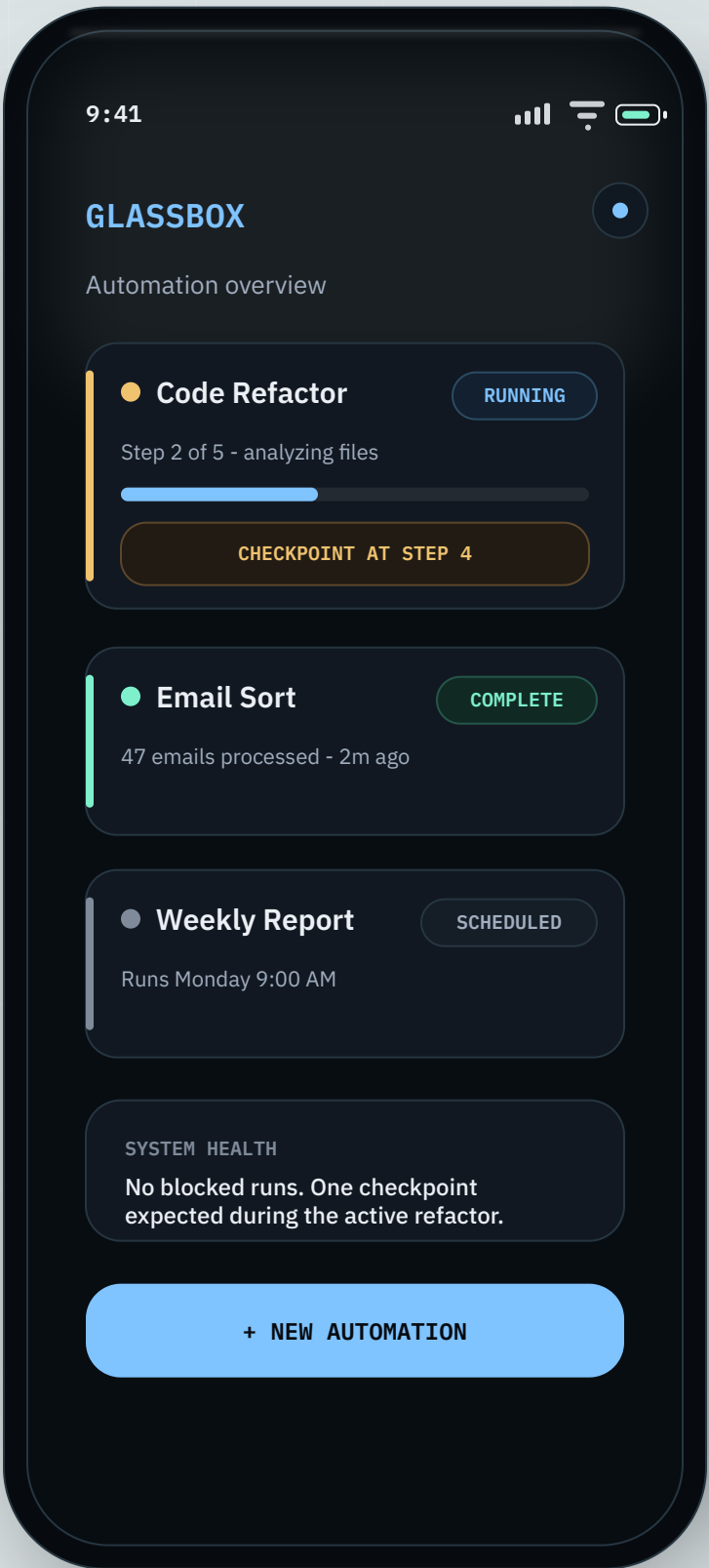
Dashboard

Entry and oversight surface for active, recent, and scheduled automations. The first screen should make system activity visible before the user drills into execution.

ACTIVE NOW

UX DECISION

The dashboard starts with observable automation instead of a prompt box. It shows what is running, what finished, and what is scheduled so the product feels accountable before the user opens a detail view.



UI DECISION

The running task is the largest object, while completed and scheduled automations stay quieter. GLASSBOX blue signals active system work, green confirms completion, and the checkpoint preview remains visible early.

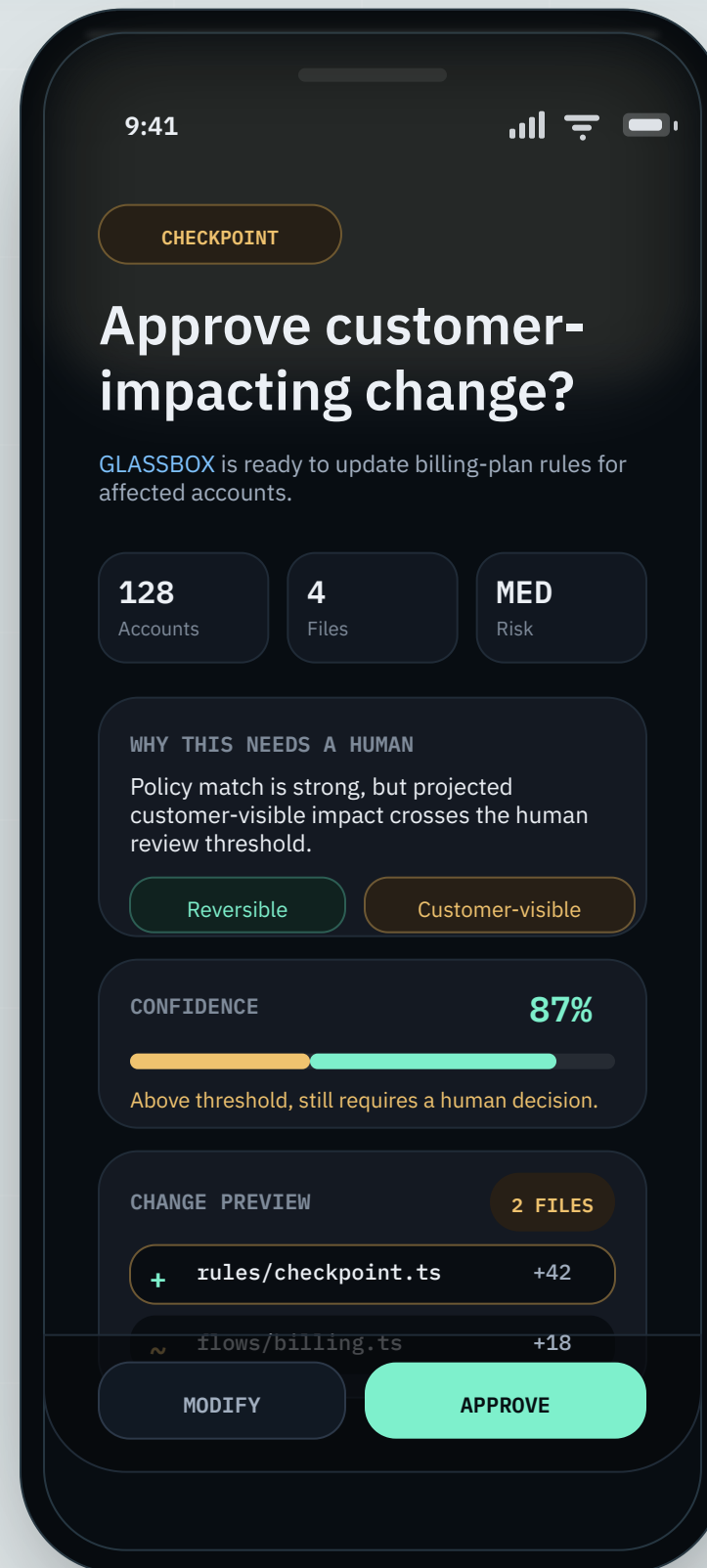
Checkpoint Gate

One native-editable screen proposal for the moment where GLASSBOX stops before consequence and asks the human to approve, modify, or reject.

SYSTEM PAUSED

UX DECISION

This screen treats human approval as a designed state, not an error or interruption. GLASSBOX pauses at the exact moment of consequence and keeps scope, risk, confidence, and reversibility visible while the user decides.



UI DECISION

The phone uses a dark operational surface so the evidence reads like a control room. Orange marks review-required information, green marks safe-to-continue outcomes, and the approval action uses dark text for legibility.