The screens of AI Personal CFO

Every screen, as the build itself renders it.

Printed from `docs/screens/SCREENS.md` and the committed Paparazzi baselines — 115 renders the build checks on every run (ADR-0082).
Rebuild with `python3 docs/screens/build_screens_pdf.py`.

The screens of AI Personal CFO

25 screens and the component gallery — 115 committed renders. Every picture below is a Paparazzi baseline rendered from the real composable and the real engines, and **the build fails if the screen stops matching it** (`verifyPaparazziDebug`, part of `./gradlew unitTests`). That is the point of this document: it is not a gallery somebody remembered to refresh, it is the test suite's own output.

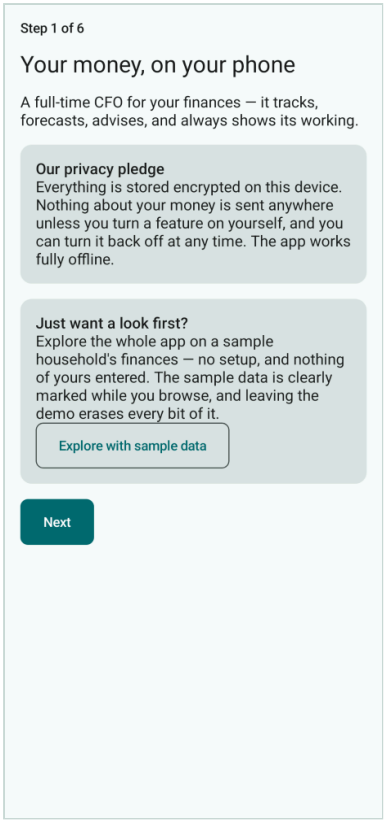
How the pictures are made	Paparazzi renders Compose on the JVM — no emulator, no device (ADR-0066)
Device	Pixel 5, `android:Theme.Material.Light.NoActionBar` pinned so an SDK bump cannot move the pixels
Configurations	Every screen in **light**, **dark** and **200% font**; the dashboard also in Hindi, Kannada and Tamil
Data	Test fixtures, computed by the real engines wherever a figure appears (P-03 — nothing here is a hand-typed number pretending to be an engine's answer)
Where they live	`**/src/test/snapshots/images/*.png`, beside the test that records them
How to refresh one	`./gradlew :feature:<module>:recordPaparazziDebug` — then **look at the images** before committing (`docs/testing/screenshot-tests.md`)

What a render is not. It is one viewport of a Pixel 5, so a screen that scrolls is shown from the top and the rest is reached in the app, not here. It is also fixture data — the amounts are realistic, not real, and no screenshot in this repository was taken from anybody's actual finances.

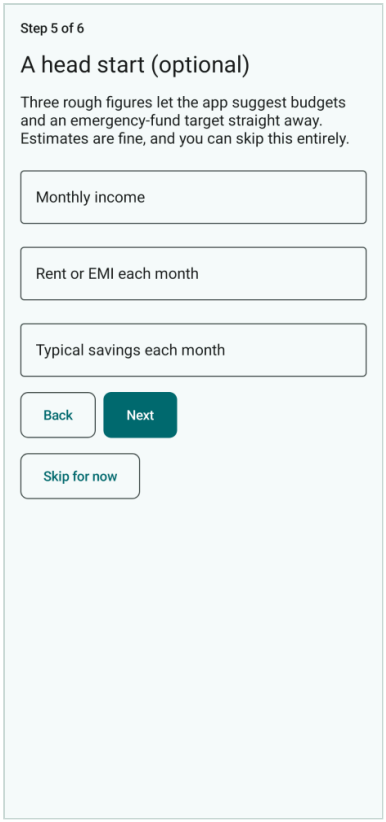
Contents

- **Getting in** — 1 Onboarding · 2 App lock
- **Home** — 3 Dashboard · 4 Order of operations
- **Money in and out** — 5 Add transaction · 6 Transactions · 7 Receipt review · 8 Bank-alert drafts · 9 Categories
- **What you own and owe** — 10 Accounts · 11 Account editor · 12 Holdings · 13 Allocation
- **Plans** — 14 Budgets · 15 Goals · 16 Goal detail · 17 Emergency fund
- **Decisions** — 18 Purchase advisor · 19 Simulators · 20 Opportunity · 21 Vehicles · 22 Assistant
- **Privacy and control** — 23 Settings · 24 Consents · 25 Erase everything
- **Appendix** — 26 Design system gallery · Coverage table · Keeping this current

1 · Onboarding



Onboarding, the welcome step



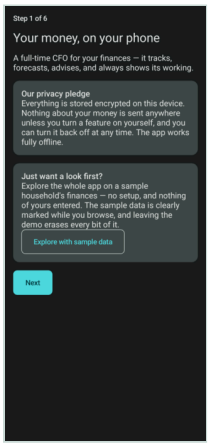
Onboarding, quick setup

`:feature:onboarding` — issue 2.1; FR-ONB-001/002/003

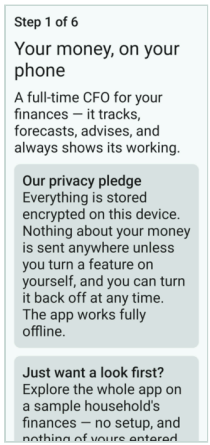
The first thing a new user sees, and the only chance to capture the profile time zone before any date in the app is computed (TIM-001). Four steps, scrollable so they survive a 200% font setting. The welcome step is the privacy pledge: the app states what it does **not** do before it asks for anything.

Quick setup (FR-ONB-002) takes three figures — income, rent or EMI, typical savings — and the `quicksetup` engine turns them into budget envelopes and an emergency-fund target, so the app has something to say on day one rather than waiting a month for history.

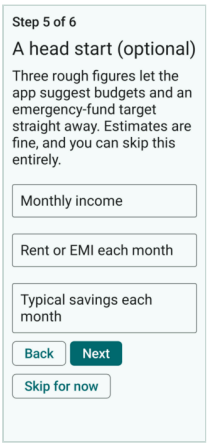
ALSO RECORDED



welcome dark

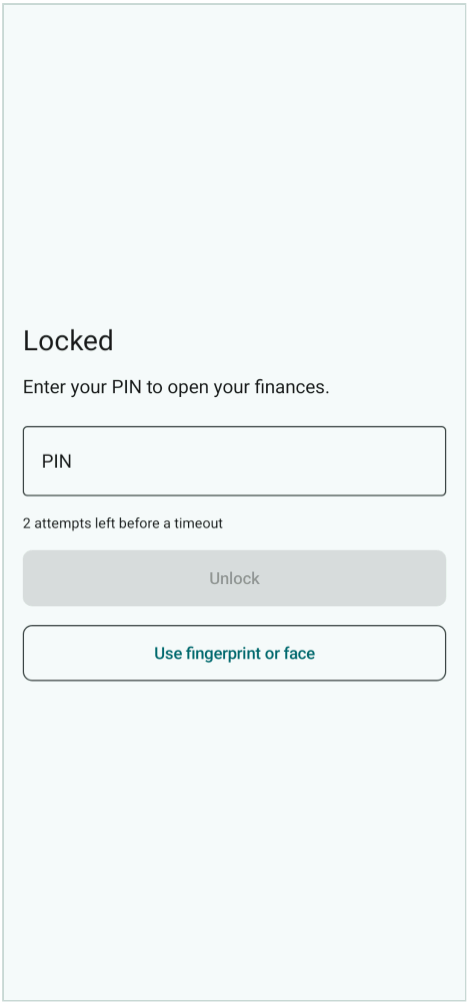


welcome 200%



quick setup 200%

2 · App lock

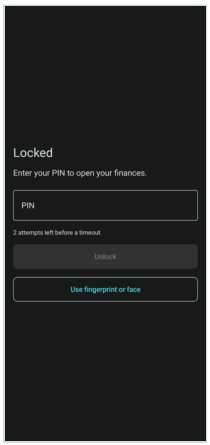


The app lock

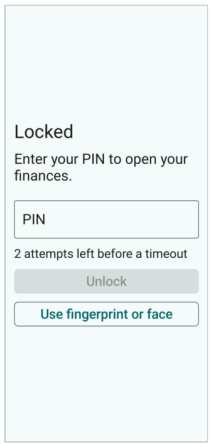
`:app` — issue 2.2; SEC-002, §23.1

The one screen whose job is to show **nothing**: no balance, no name, no figure of any kind, because the threat §23.1 names is someone else holding the phone. Biometric unlock (class 3) with a PIN fallback; repeated failures lock the gate for a cooling-off period and the screen says when it will open again rather than silently rejecting a correct PIN.

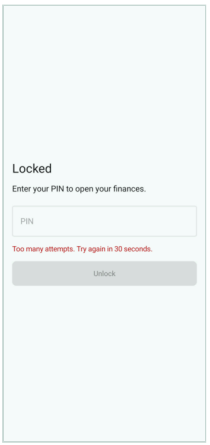
ALSO RECORDED



dark

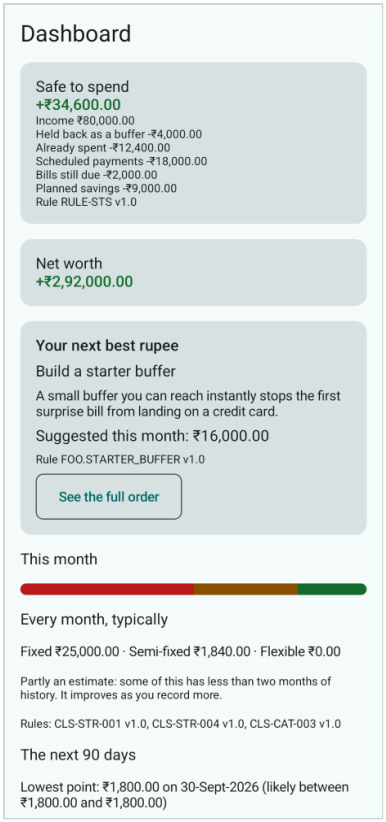


200%

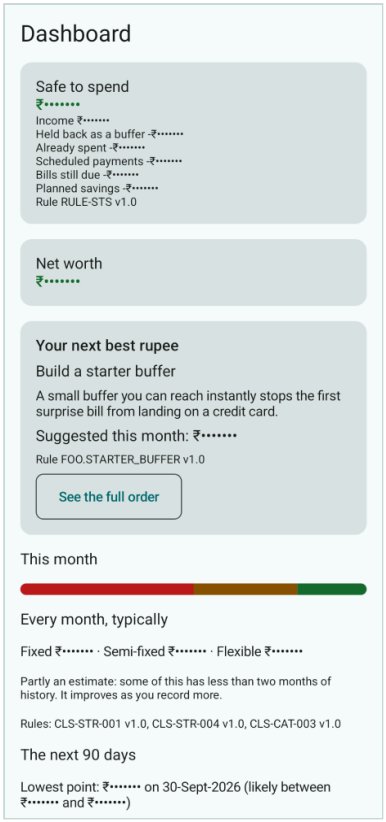


locked out

3 · Dashboard



The dashboard



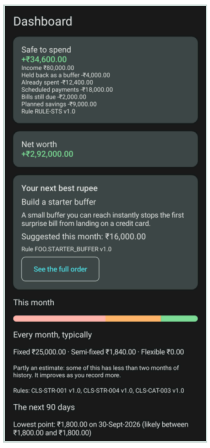
The dashboard with amounts blurred

`:feature:dashboard` — issues 5.1–5.3, 9.1–9.5; §5.2, §14, AI-STS, AI-FHS, AI-FCT

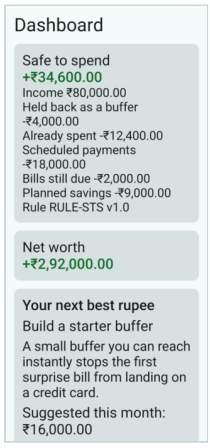
The home screen and the reference for every screen after it. Safe-to-Spend with its breakdown (§5.2), net worth, the 90-day forecast with its crunch days (AI-FCT), the financial-health score and its five pillars (AI-FHS), the stream load, the ranked insight feed (AI-ORCH) and the next-best-rupee card (§36). Every figure arrives from an engine with its provenance; the screen computes nothing (P-03).

Privacy blur (issue 5.3): one tap hides every amount on the screen without logging the user out — for reading the app on a train. The Compose test proves no amount is on screen; this render is how anyone sees what that looks like.

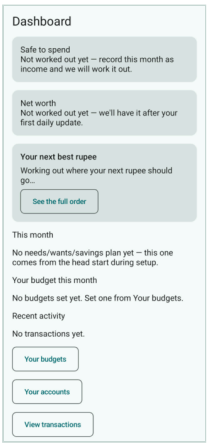
ALSO RECORDED



dark



200%



empty



हिन्दी

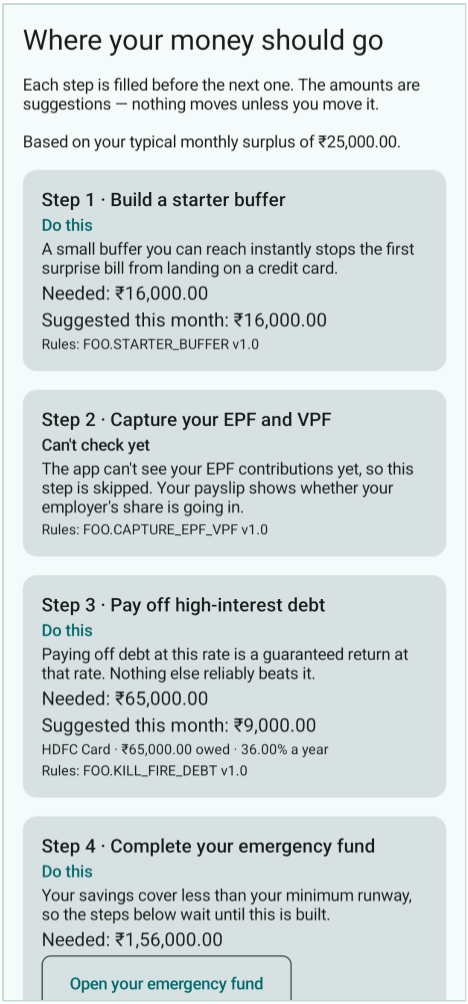


ಕನ್ನಡ



தமிழ்

4 · Order of operations

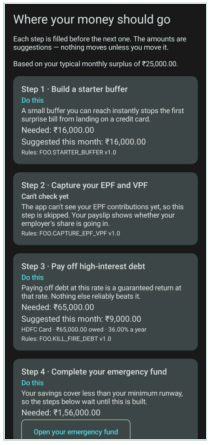


The full order of operations

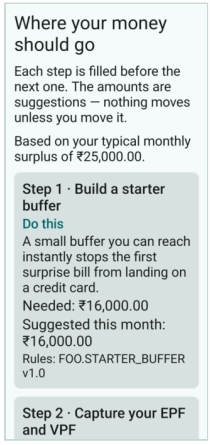
`:feature:dashboard` — issue 7.5; §36, FOO-002

The dashboard card names one action; this screen shows **all eight stages** — which are done, which is next, which are blocked and by what. Each stage carries the reason it is where it is and the rule id that put it there (P-02). The household here has a thin buffer and a 36% card, so the buffer comes first and the goal is held behind the emergency-fund gate.

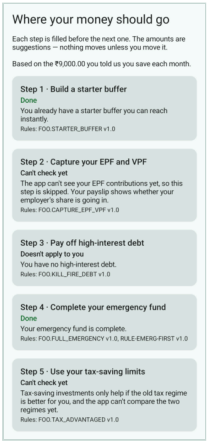
ALSO RECORDED



dark

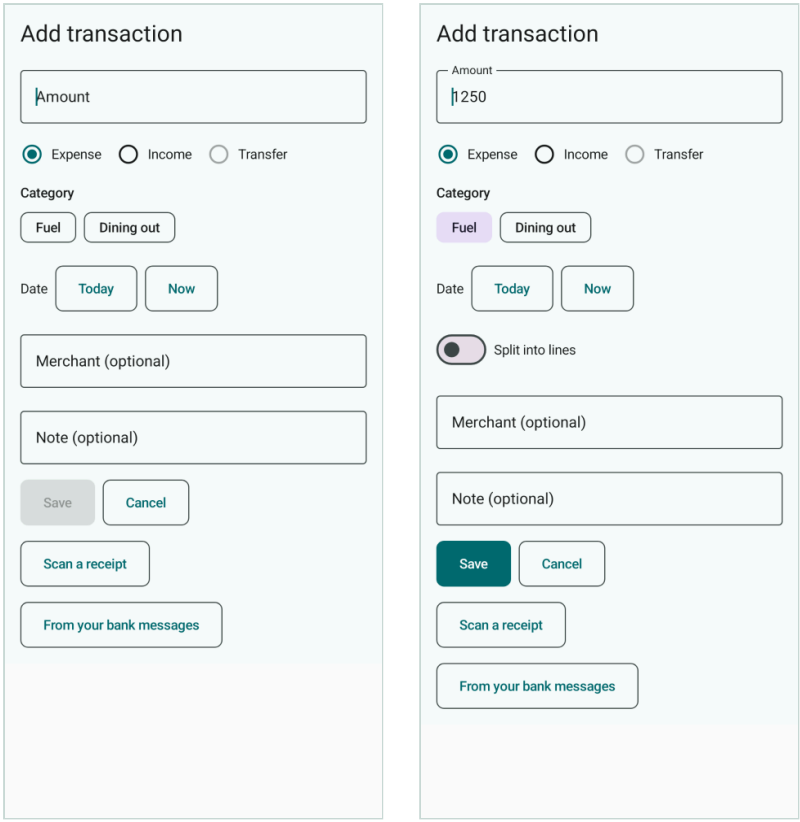


200%



past every gate, with a grey-zone choice

5 · Add transaction



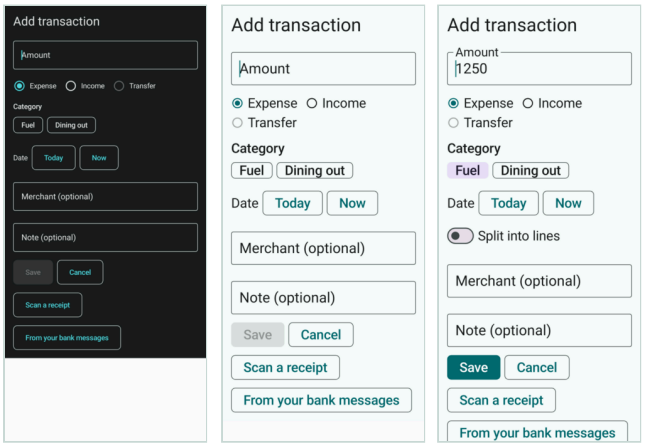
Add a transaction — Add a transaction, filled in

`:feature:transactions` — issue 3.1; FR-TXN-001, FR-TXN-002

FR-TXN-002 is a MUST with a number in it: one tap to reach, **three taps or fewer** to complete a common expense. The layout is arranged around that budget — the amount field takes focus so the keypad is already up, the account is preselected and its picker hidden when there is only one, and the category row disappears when the profile has none. Type the amount, tap Save.

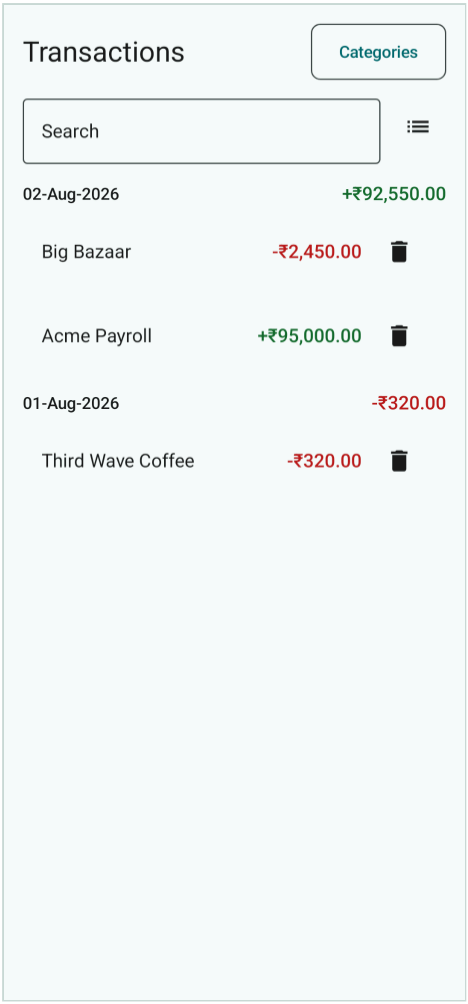
Filled in: the classifier's category suggestion with its confidence (§8.1), split lines, transfer support, scheduling, and the receipt and bank-alert routes out to their own screens.

ALSO RECORDED



dark — 200% — filled, 200%

6 · Transactions

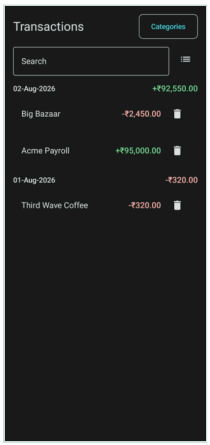


The transaction ledger

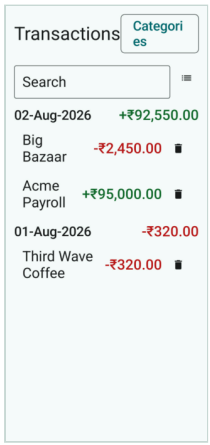
`:feature:transactions` — issues 3.1, 3.6; FR-TXN-007

The ledger, grouped by the day a transaction was booked on, each day carrying its **net** total read from the database rather than folded from the rows on screen — a page boundary can fall inside a day. Rows arrive as `PagingData`, so the list stays cheap as history grows. Search, filters, bulk selection and undo sit above and around it. A transfer collapses into one row and moves no day total.

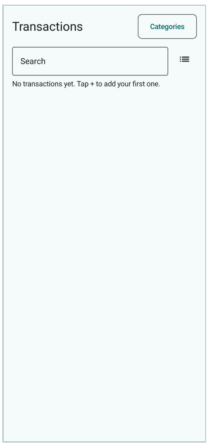
ALSO RECORDED



dark

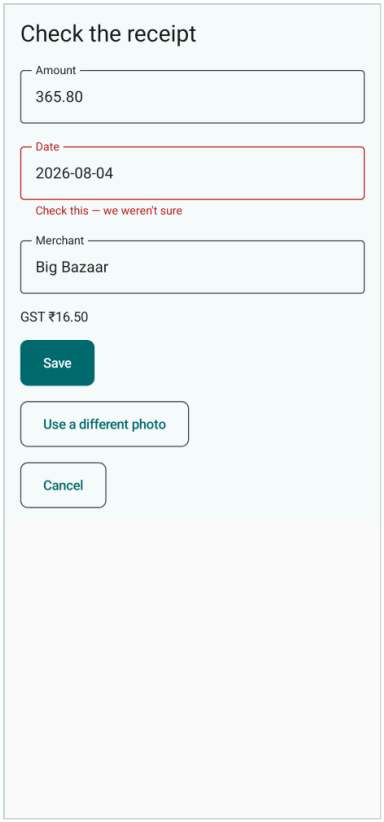


200%

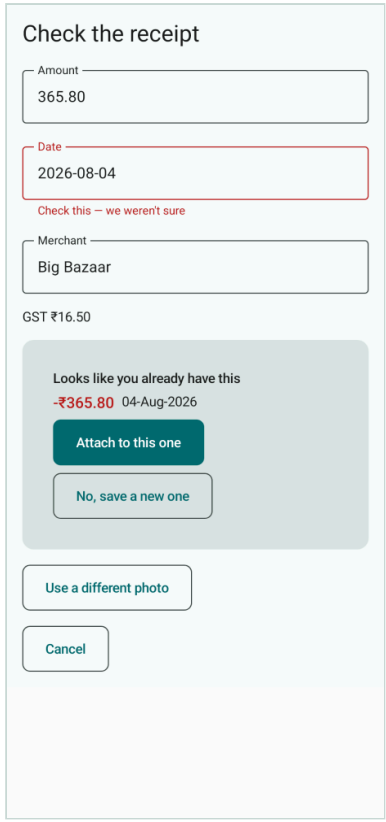


empty

7 · Receipt review



Reviewing a scanned receipt



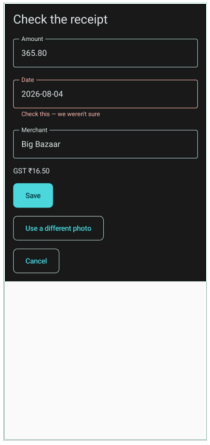
A receipt that looks like one already in the ledger

`:feature:transactions` — issue 3.8; FR-OCR-001…006

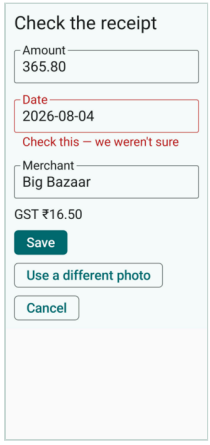
The screen captures its own photo, runs ML Kit on device, and then **asks**. Total, date and merchant come back as editable fields with the GST the scan read beneath them (FR-OCR-003), and a field the scanner was unsure of is flagged in red — *check this, we weren't sure* — so the eye goes there first. Nothing is saved until the user confirms it.

FR-OCR-006: when the scan matches a transaction already recorded, the match is shown and the user decides whether to merge or save anyway — the only thing between a scanned receipt and a double entry.

ALSO RECORDED

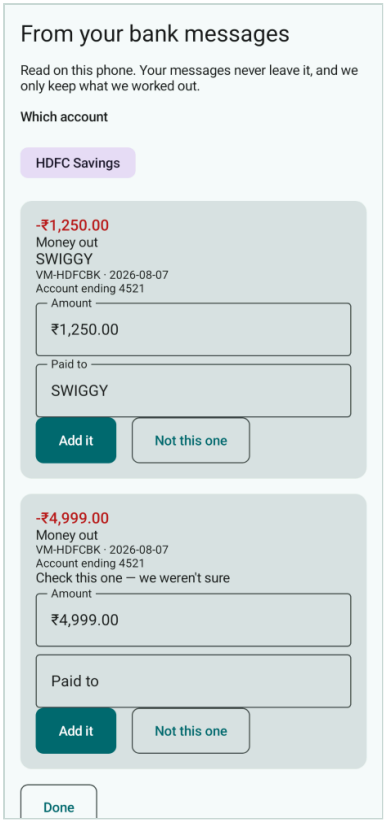


dark

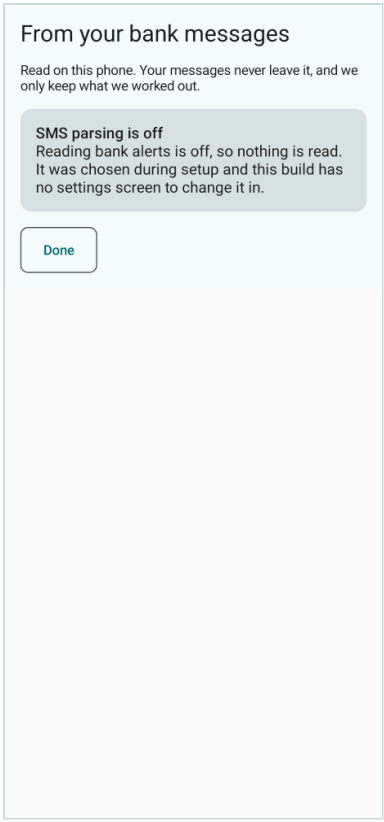


200%

8 · Bank-alert drafts



Drafts read from bank alerts



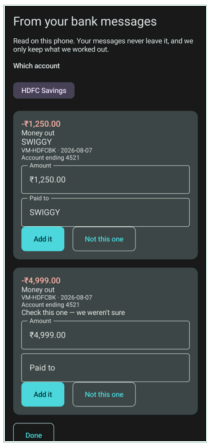
Before consent is given

`:feature:transactions` — issue 3.9; §18, §23

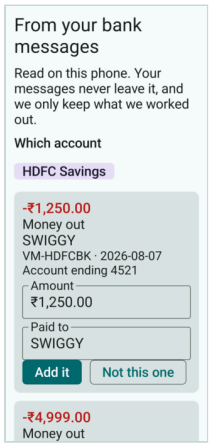
The app can read bank SMS alerts and propose transactions from them — **opt-in, on device, and never automatic** (P-01, P-07). Each draft shows the sender, the amount, the counterparty and the account tail; a draft the parser was unsure of is marked as such rather than accepted at the same weight as a confident one. Nothing becomes a transaction until the user accepts it.

The state every install starts in, and the one most users stay in: the offer explained, the permission not yet asked for. This screen is the only place in the app that requests `READ_SMS`.

ALSO RECORDED

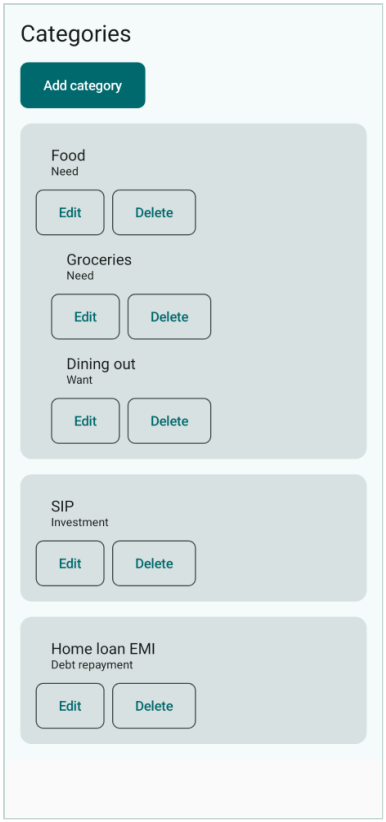


dark

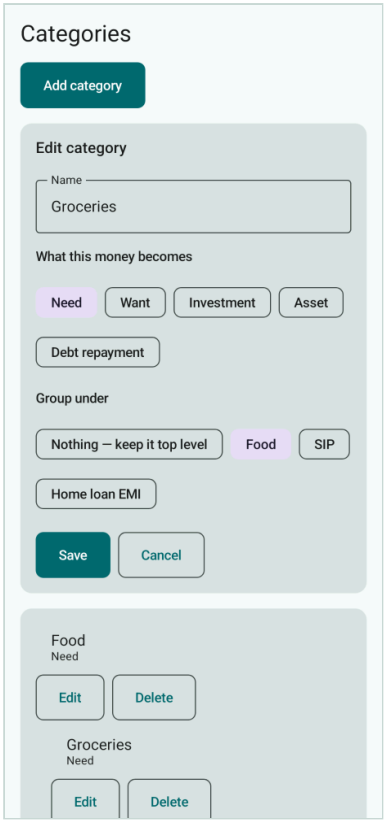


200%

9 · Categories



The category taxonomy



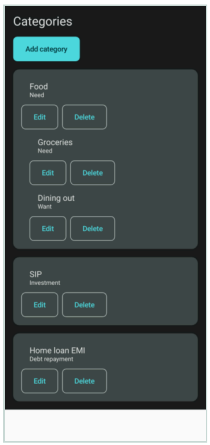
The category editor open

`:feature:categories` — issue 4.1; FR-SET-001

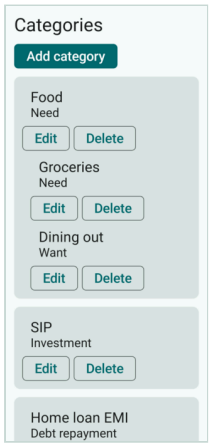
The taxonomy, two levels deep, each category carrying its **nature** — Need, Want, Invest, Asset or Liability (§8.3). The nature is what lets the app say where a rupee went rather than only what it was spent on, so it is on every row rather than hidden in an editor.

The editor opens as a sheet over the tree. The screen is a plain scrolling column rather than a lazy list for a reason recorded in its own source: with the list lazy, the sheet could not scroll, and at a 200% font setting Save was unreachable.

ALSO RECORDED

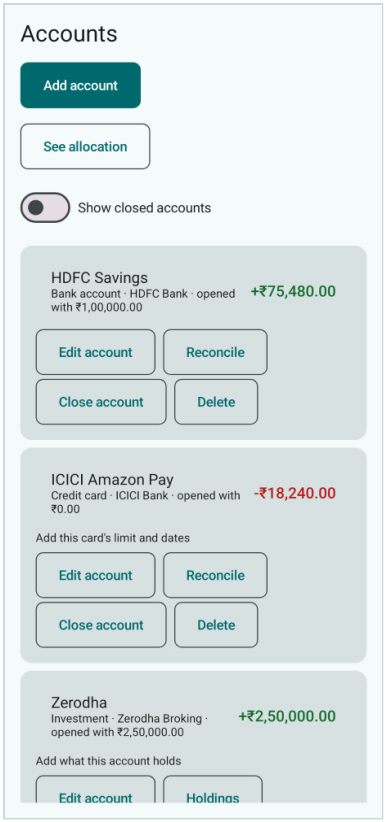


dark

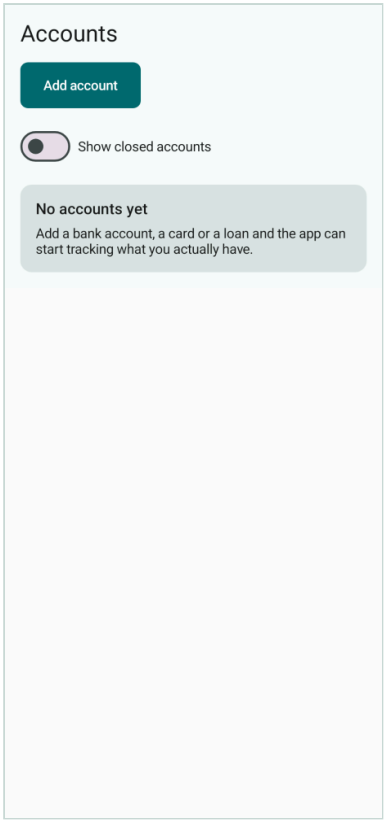


200%

10 · Accounts



The accounts list



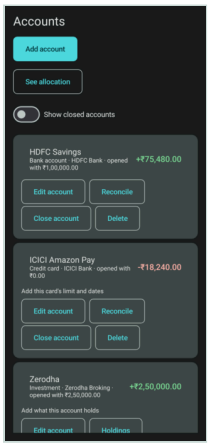
A user with no accounts

`:feature:accounts` — issue 2.5; FR-ACC-001, FR-ACC-007

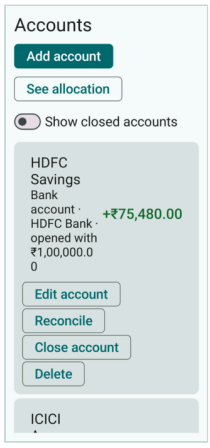
Eleven account types in one list — bank, cash, credit card, loan, investment, gold, crypto, property, vehicle, receivable, payable — each row showing the balance **derived from transactions** rather than the opening balance it was created with (DB-001). The card here has no terms entered, so the row prompts for them instead of drawing a 0% utilisation bar; the investment account holds nothing yet and says so rather than showing ₹0 across 0 holdings. That is P-03 on a list row: absence is a prompt, never a zero.

The invitation, which the state class distinguishes from both "still loading" and "the read failed" — a database that would not open must not render as a cheerful invitation.

ALSO RECORDED

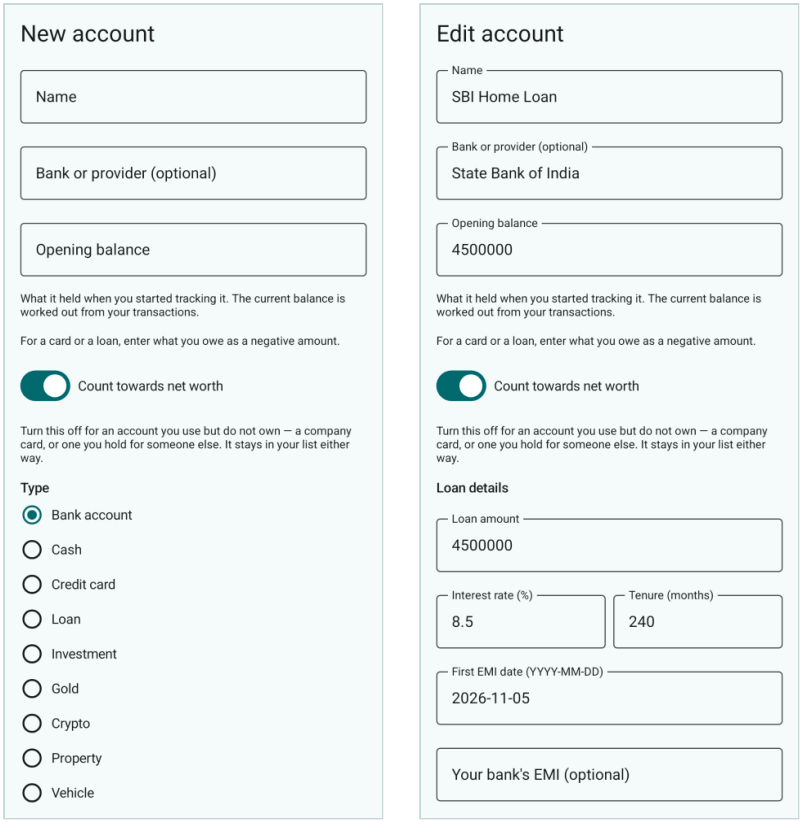


dark



200%

11 · Account editor



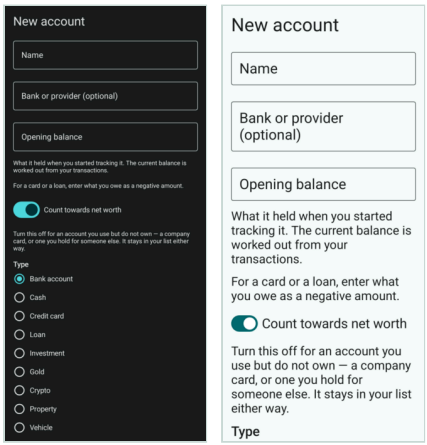
Creating an account A loan's terms

`:feature:accounts` — issues 2.5, 6.1, 6.2; FR-ACC-001/002/003

One screen for create and edit, because the fields, the validation and the save are identical and duplicating them is how two screens drift apart. Only the name is required; a blank opening balance means zero, not an error.

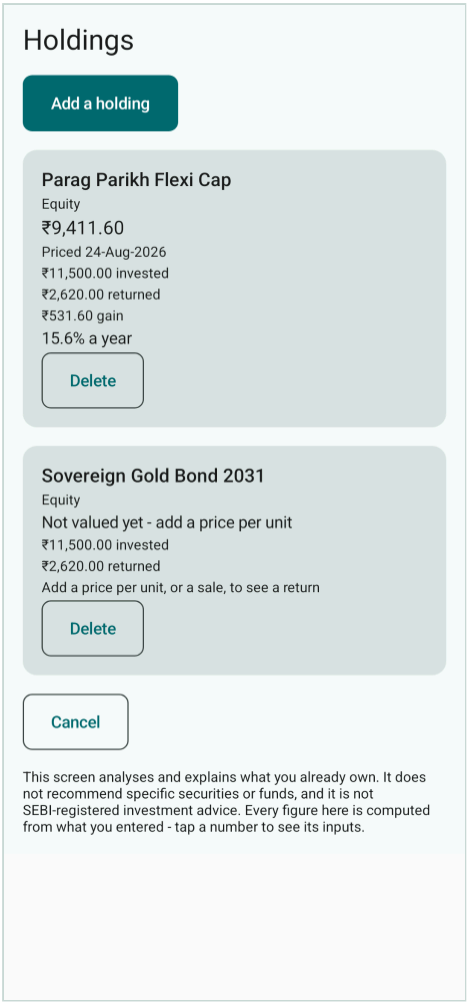
The form grows a section when the type demands one: a credit card adds a limit, statement and due days, last statement, minimum due and APR (FR-ACC-002); a loan adds principal, rate, tenure and the first EMI date (FR-ACC-003). Rates are typed in percent and stored in basis points (MNY-002), and amounts stay text until Save so a half-typed `8.` is never parsed mid-keystroke.

ALSO RECORDED



dark 200%

12 · Holdings

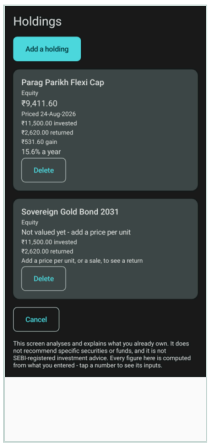


What one investment account holds

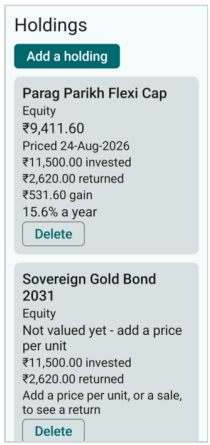
`:feature:accounts` — issue 6.3; §11, AI-INV

What is inside one investment account and what it has returned: current value, gain, and a **money-weighted return (XIRR)** rendered from integer basis points, with the age of the price it was computed from. A holding with no price shows "not valued yet" rather than a total loss, and §11.1's disclaimer sits under the list wherever a figure appears — the one place in the build where that requirement is enforced at all.

ALSO RECORDED



dark

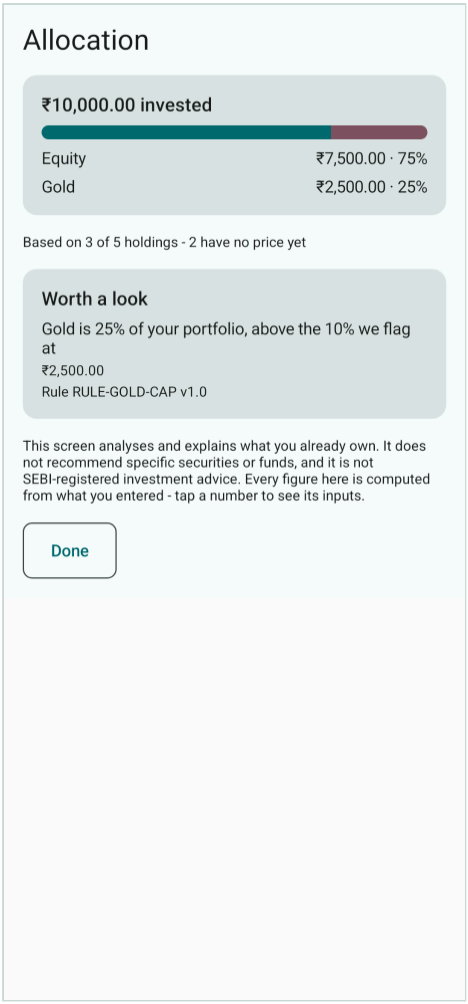


200%



nothing held yet

13 · Allocation

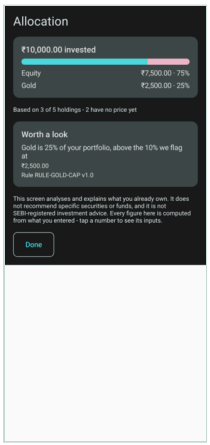


The portfolio split

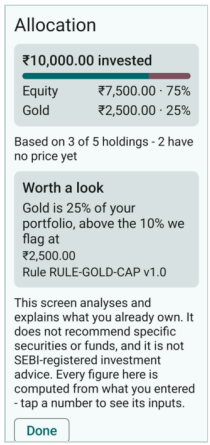
`:feature:accounts` — issue 6.4; FR-INV-002, §11.2

How the portfolio is spread across asset classes, and what about that is worth a look. The gold slice here is 25% against a 10% cap, so the flag states the measured share, the threshold **and the rule id** — a figure the app raises must be traceable to the row that produced it (§29, P-02). The coverage line says how much of the portfolio the split is actually based on when some holdings have no price.

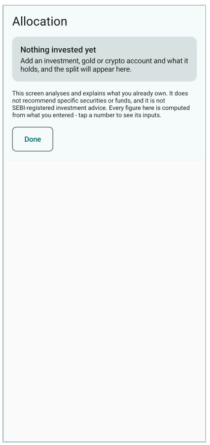
ALSO RECORDED



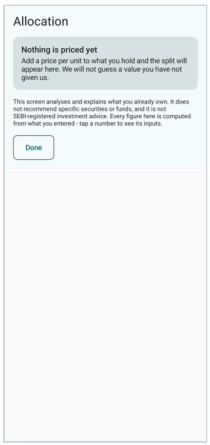
dark



200%

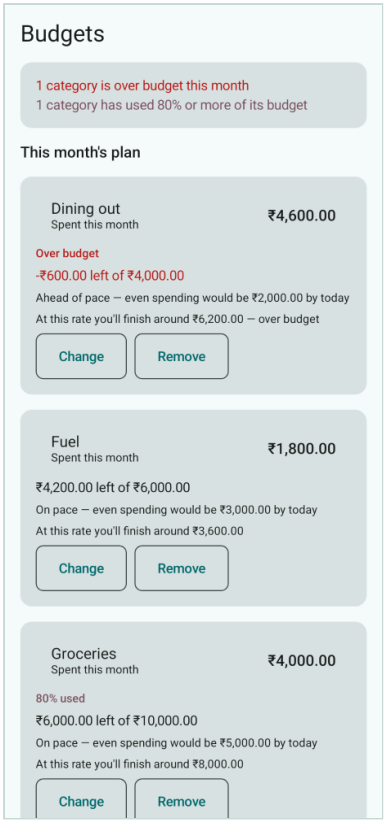


nothing invested

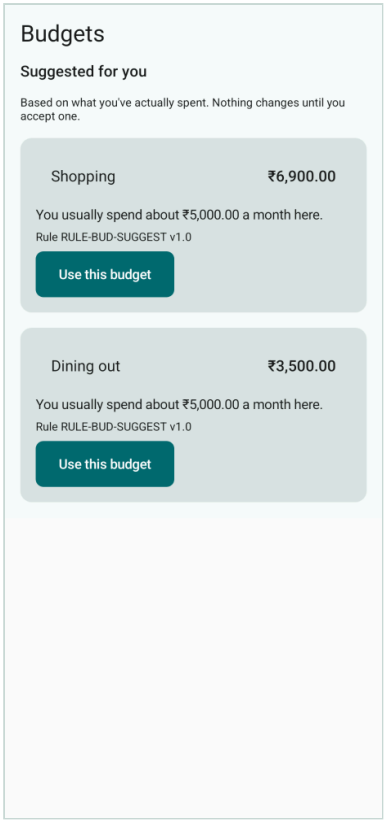


nothing priced

14 · Budgets



This month's budgets



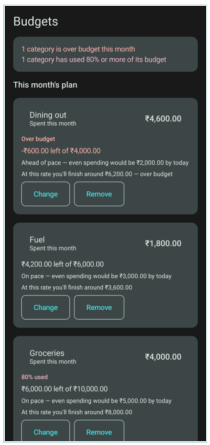
Suggested budgets

`:feature:budgets` — issues 4.4–4.6; FR-BUD-001/002/003/004

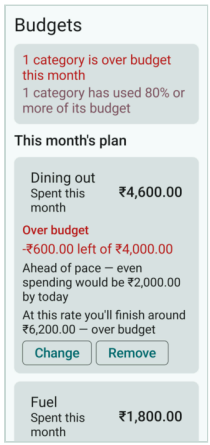
Each envelope shows what was spent, what is left, **what even spending would have been by today** ("safe pace") and where the month is projected to finish. The banner at the top counts the envelopes that are over and the ones at 80% or more (FR-BUD-004). Bands are words as well as colours, so an exceeded budget still reads as exceeded in dark mode or to a colour-blind user.

Where a budget starts (FR-BUD-002): the engine proposes a figure from the user's own median spend, adjusted for seasonality, and says so. The user accepts, edits or ignores it (P-07).

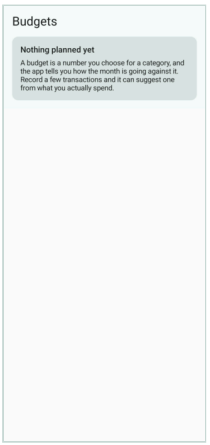
ALSO RECORDED



dark

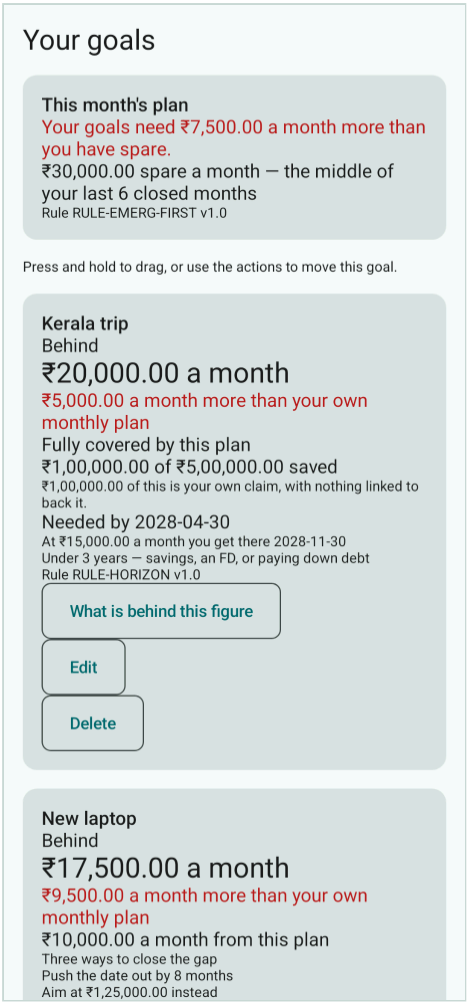


200%



nothing planned yet

15 · Goals

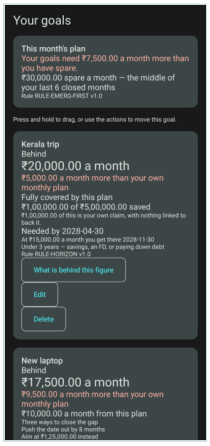


The goals list

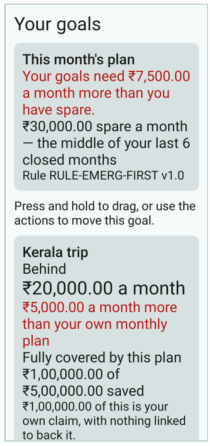
`:feature:goals` — issues 7.1, 7.3; §15, AI-GOAL

Target plus date gives a required monthly figure and an ETA (AI-GOAL), and §36's waterfall divides one month's surplus between goals that compete for it. Both engines' answers share this screen, each with the working beside it and the rule that produced it.

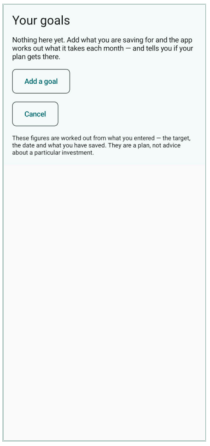
ALSO RECORDED



dark

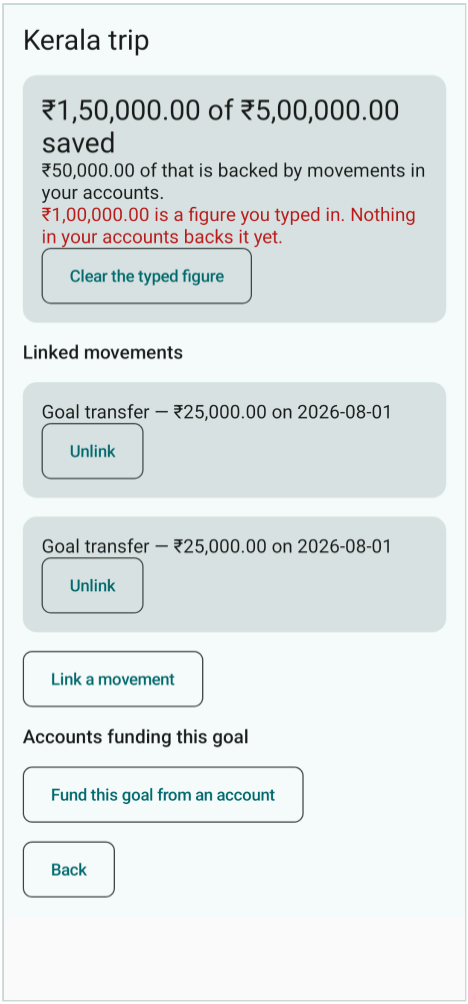


200%



no goals yet

16 · Goal detail

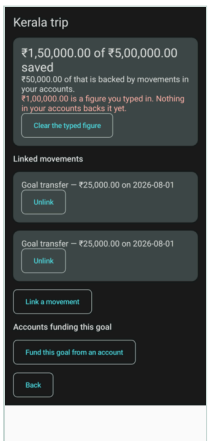


One goal's progress

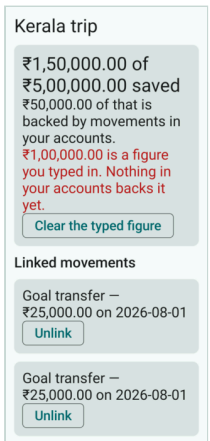
`:feature:goals` — issue 7.4; §15, FR-GOAL-002, FR-GOAL-004

§15 asks for progress that is **transaction-evidenced**, and for a manual claim to be visually distinct from money the ledger can actually see. The bar is split: the evidenced part, and the part that is the user's word. Beneath it are the linked movements and the accounts dedicated to the goal.

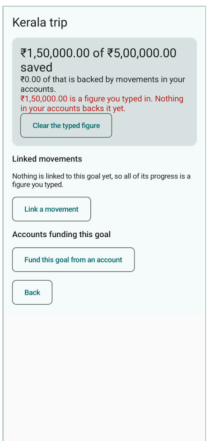
ALSO RECORDED



dark

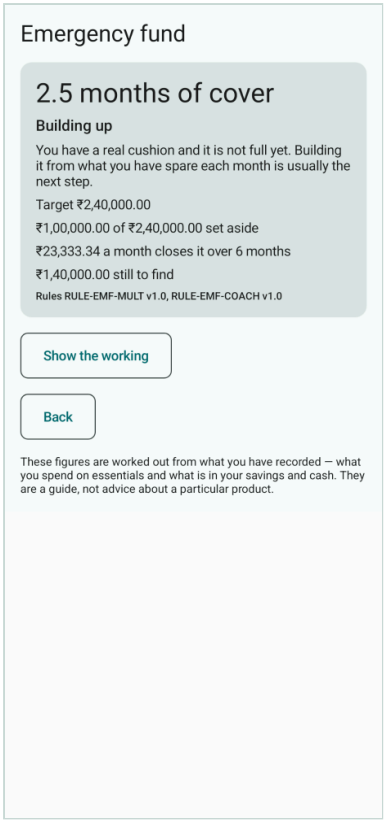


200%

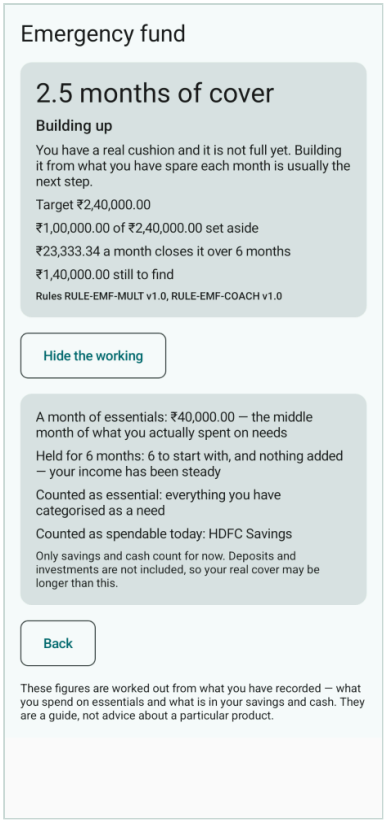


nothing linked at all

17 · Emergency fund



The emergency fund



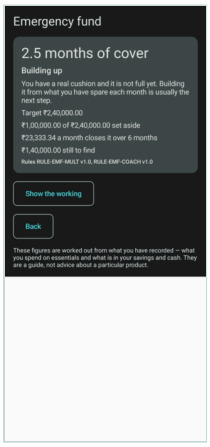
The evidence panel open

`:feature:emergencyfund` — issue 7.2; §10.1, AI-EMF

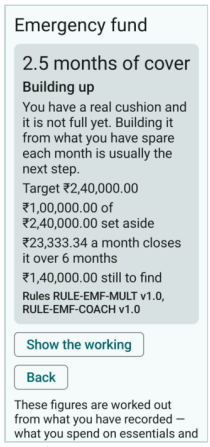
Essentials × a multiplier M gives the target; what is liquid gives the runway in months; the coach line says what to do next and never tells a funded fund to top itself up. The multiplier is not a constant — it moves with how steady the income is.

§10.1 requires every number in the explanation to link to its evidence: which accounts counted as liquid, which categories as essential, and why M came out as it did.

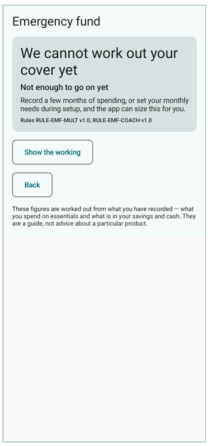
ALSO RECORDED



dark

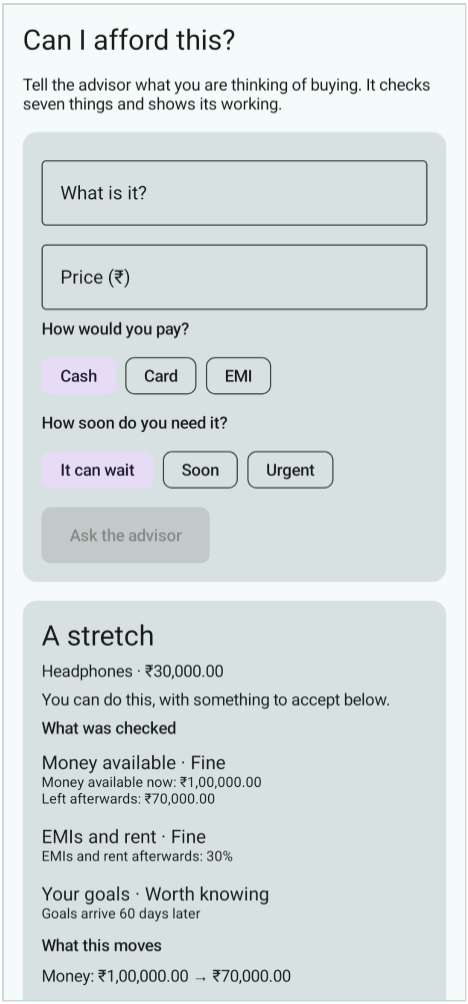


200%



essentials not known yet

18 · Purchase advisor



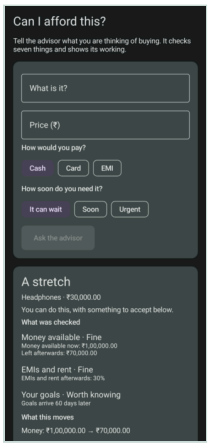
The purchase advisor

`:feature:advisor` — issue 10.1; §13.2, FR-AI-003

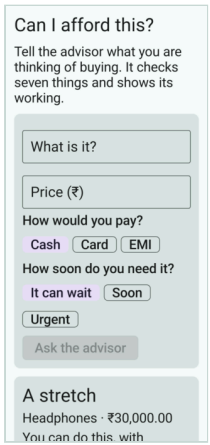
"Can I afford this?" — asked with an amount, answered with one word and a table of gates. Each gate shows its own figures, what the purchase would move, and what would change the answer. The verdict is the smallest part of the screen on purpose: it is the reasoning the user is meant to argue with (P-02).

The three verdicts are recorded separately, because each has to read differently: Stretch · Comfortable · Not now — and "not now" must read as advice with its reasons shown, never as a blocked action (P-07).

ALSO RECORDED

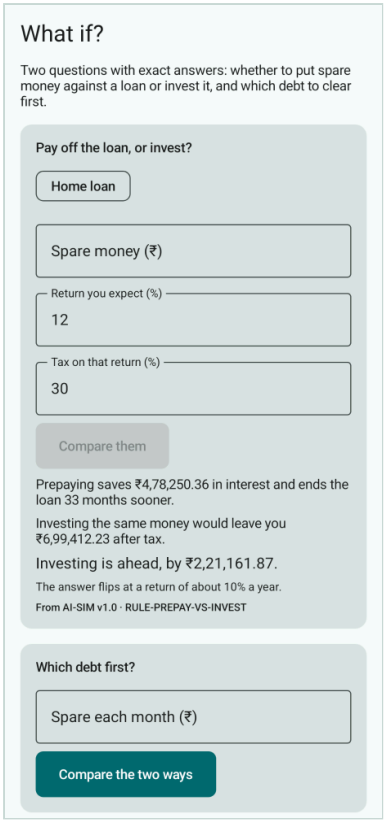


dark

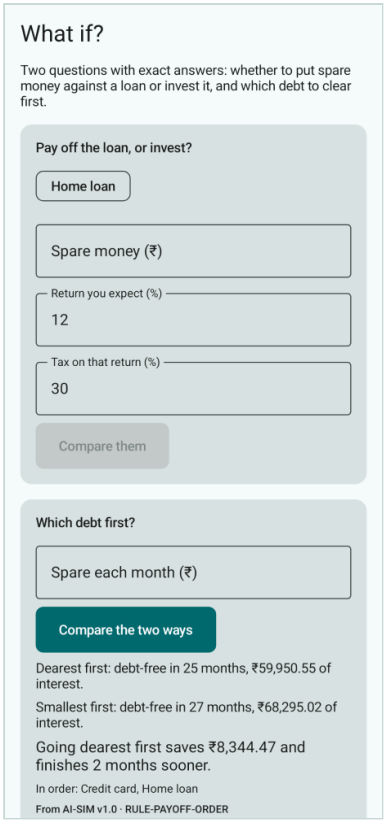


200%

19 · Simulators



Prepay or invest



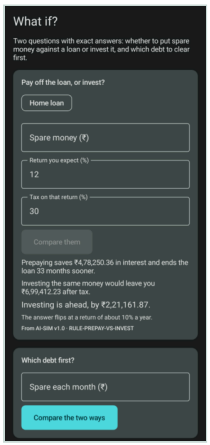
Which debt first

`:feature:advisor` — issue 10.3; §36, §40.2, AI-SIM

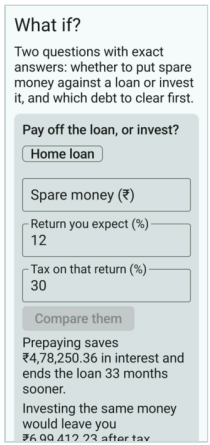
Prepay the loan or invest the money? Both sides are shown with the gap between them **and the rate at which the answer flips** — a verdict alone would be the black box P-02 forbids.

Dearest-first against smallest-first, with what the cheaper order is actually worth in interest and months. A user who picks the slower order for motivation is P-07 working as intended, which is only possible when both orders are on screen. The line at the bottom — *nothing has been paid or moved — this is arithmetic* — is why this screen is not a payment screen.

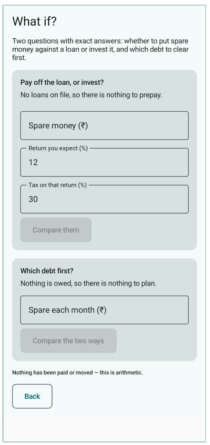
ALSO RECORDED



dark

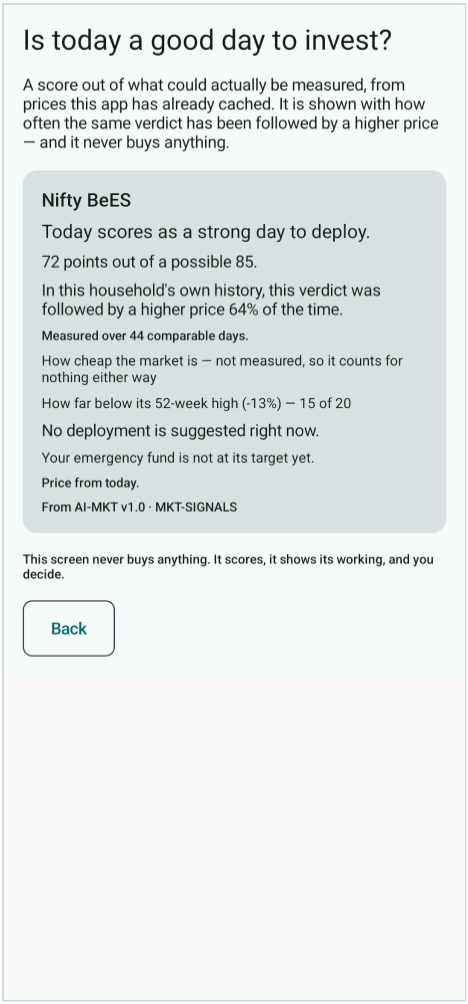


200%



no debts to simulate

20 · Opportunity

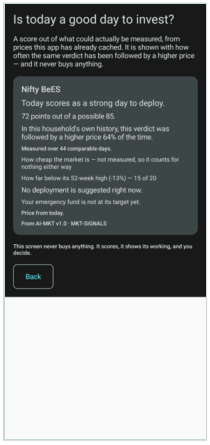


The opportunity screen

`:feature:market` — issue 10.7; §30, AI-MKT

The screen most likely to be read as a tip, so everything that stops it being one is on it: the score is out of what was **possible** rather than out of 100, each signal says whether it could be evaluated at all, the capacity gates show which ones failed, and the price's age is stated. Market data is optional and consented (P-01) — and the band never becomes an instruction.

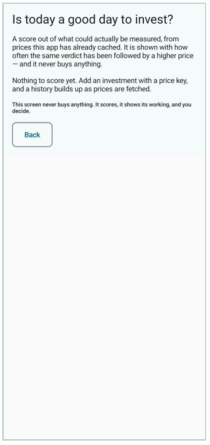
ALSO RECORDED



dark

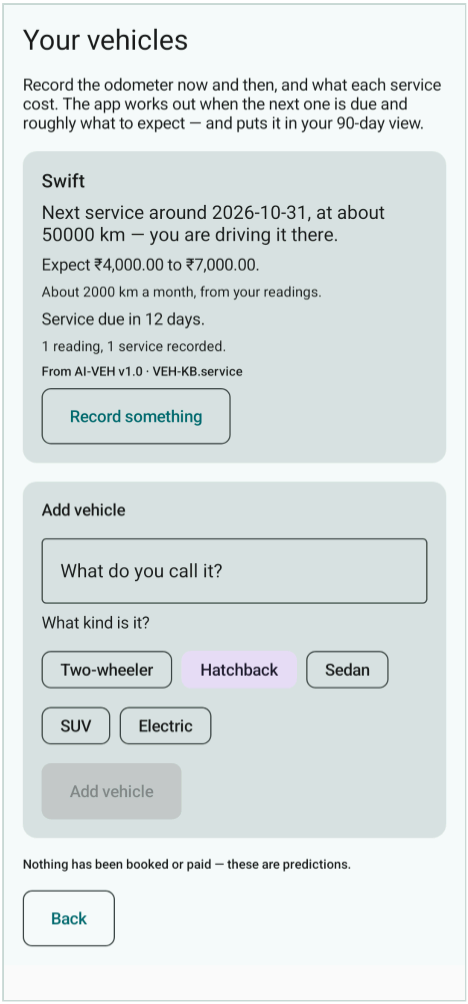


200%



nothing to assess

21 · Vehicles

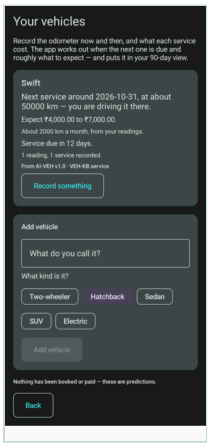


The vehicles screen

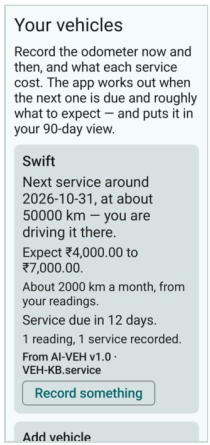
`:feature:vehicle` — issue 10.4; §12, AI-VEH

A vehicle's costs are the ones that arrive as surprises and are mostly predictable: the next service from distance driven, the renewals from their dates, and a likely cost range — widened from the knowledge base, narrowed by what this household has actually paid before, which the screen says plainly. Overdue is worded as overdue, not as something still coming. Nothing has been booked.

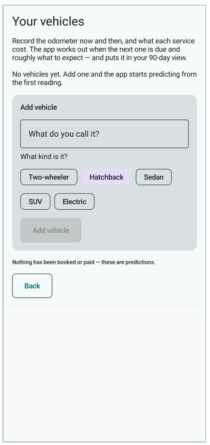
ALSO RECORDED



dark

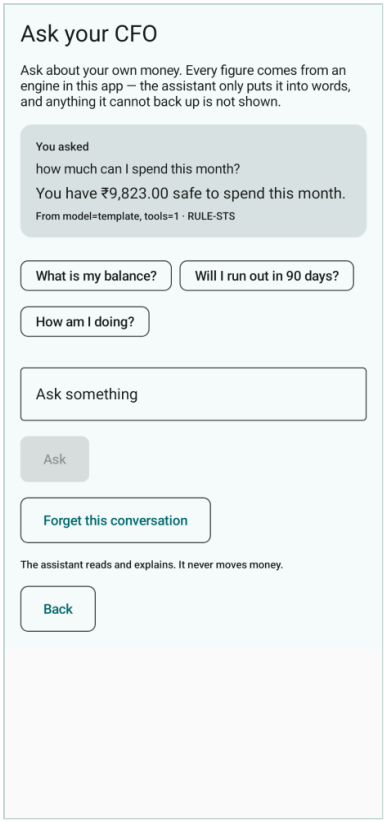


200%

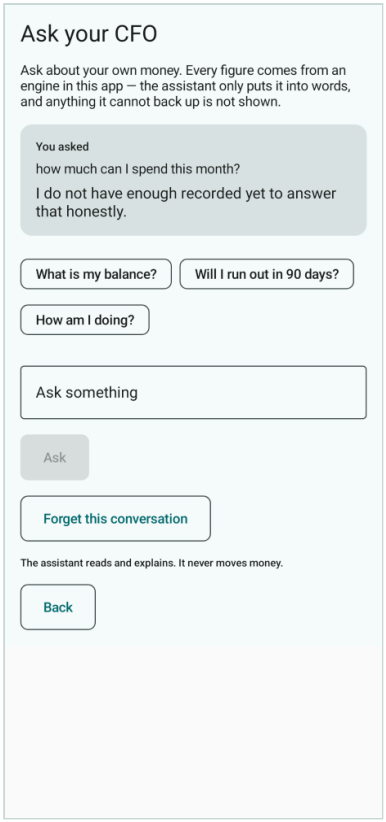


no vehicle recorded

22 · Assistant



An answered question



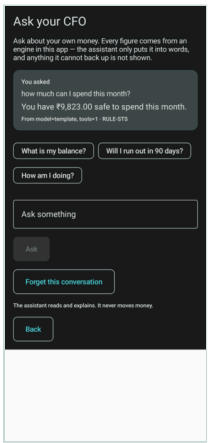
A refusal

`:feature:chat` — issue 10.5; §19, AI-CHAT

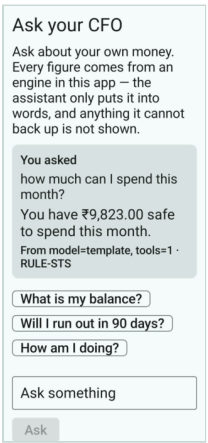
The one screen where the app speaks in sentences — and every answer carries the model that wrote it and the rules behind the figures in it. The numbers are never the model's: the deterministic engines compute them, the guardrail checks every figure in the reply against them, and the language model only puts them into words (P-03, AI-ARC-004).

When there is no data to answer from, or the guardrail blocks the reply, the screen says what happened and offers a way forward. A polite sentence with an invented number in it would be the one failure this feature cannot have.

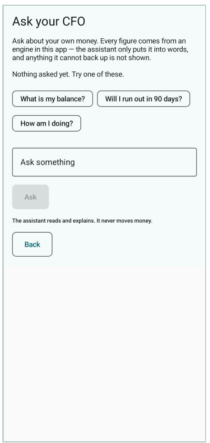
ALSO RECORDED



dark

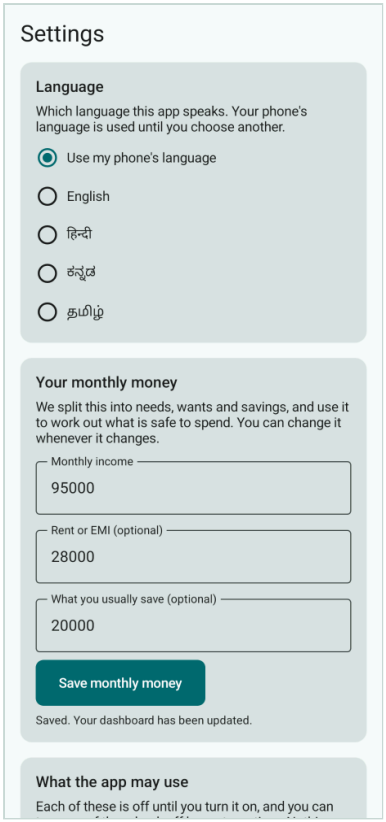


200%

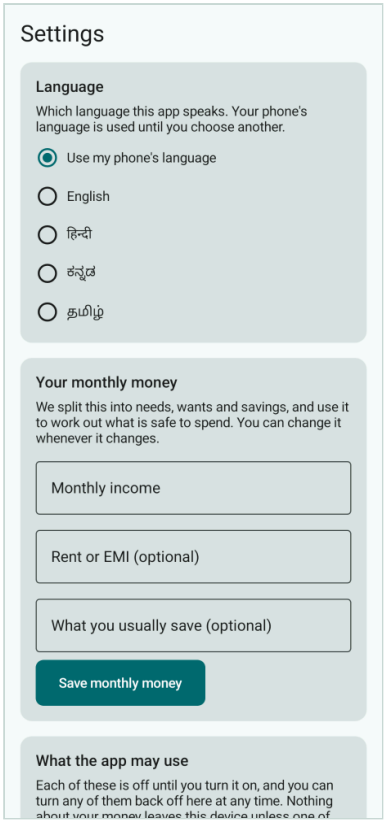


before anything is asked

23 · Settings



Settings



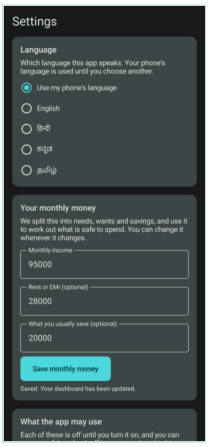
A fresh install

`:feature:settings` — issues 8.1, 8.2, 10.8, 11.x; FR-SET-001, SEC-002, SEC-005

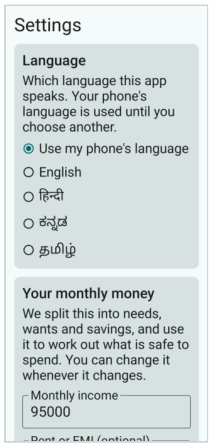
The money plan (income, rent or EMI, typical savings), the app lock and its PIN, the app language, and the **encrypted backup and restore**. SEC-005's sentence — that a backup cannot be opened without the passphrase and the passphrase cannot be recovered — sits beside the button that creates it, and the acknowledgement is a real checkbox rather than fine print.

Before anything is saved: every consent off (P-01's default), the backup unavailable, and the reason it is unavailable shown rather than a button that quietly does nothing.

ALSO RECORDED

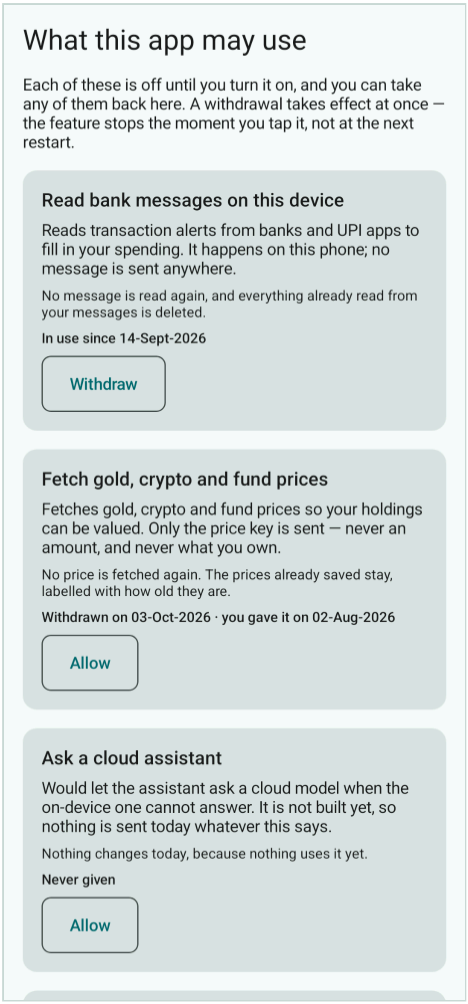


dark



200%

24 · Consents

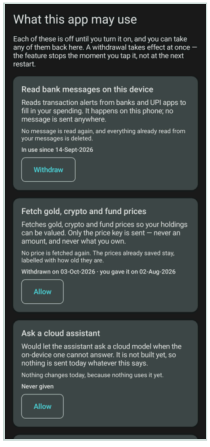


The consents dashboard

`:feature:settings` — issue 11.3; §23, P-01, DPDP

P-01 does not promise a panel of switches; it promises something a person can **audit**. Every feature that could ever use data beyond the device is listed with what the consent is for, what stops if it is withdrawn, and when it was granted or revoked. A withdrawn consent keeps both dates, which is what makes this a ledger rather than a switchboard.

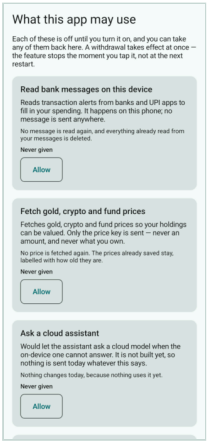
ALSO RECORDED



dark

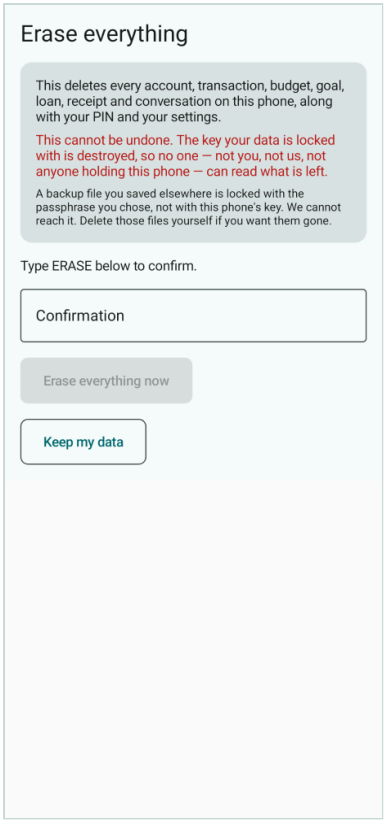


200%

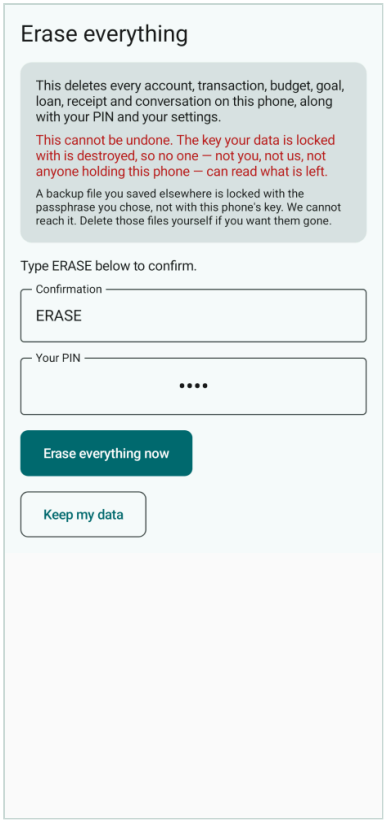


a fresh install, all off

25 · Erase everything



Erase everything



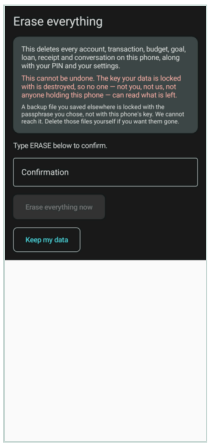
Confirmed, and asking for the PIN

`:feature:settings` — issue 11.4; §23, §34, SEC-003

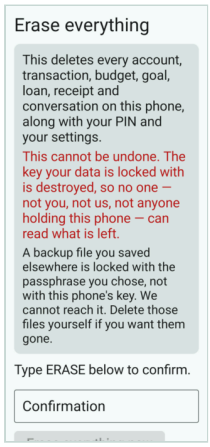
The DPDP right to erasure, done the only way a privacy-first app can do it honestly: destroy the key in the Keystore, and the ciphertext is unrecoverable (crypto-shredding). There is no undo, so the screen says so before it asks for anything.

Two barriers, not one: the confirmation word typed out, and — on a profile with the app lock on — the PIN as well.

ALSO RECORDED

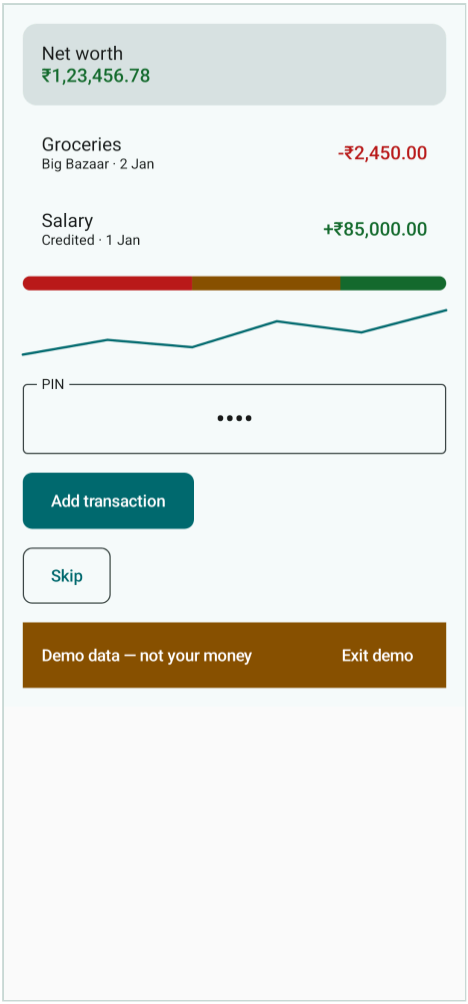


dark



200%

26 · Design system gallery

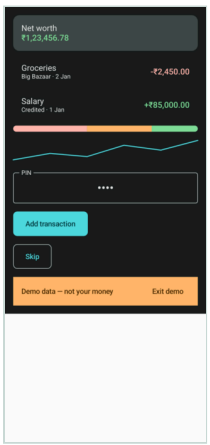


The design system gallery

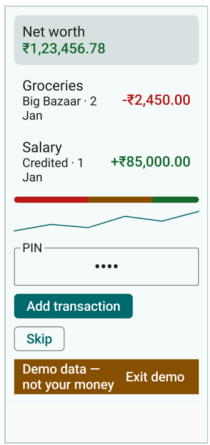
`:core:designsystem` — issue 1.8; §21.5

Not a screen a user reaches — every component and both charts rendered together, so a change to a shared token is visible as one diff instead of as twenty screens moving at once. When a baseline here changes, every screen in this document is about to change too.

ALSO RECORDED



dark



200%

Coverage

Module	Baselines	Screens covered
`:app`	4	app lock
`:core:designsystem`	3	the component gallery
`:feature:accounts`	17	accounts, account editor, holdings, allocation
`:feature:advisor`	10	purchase advisor, simulators
`:feature:budgets`	5	budgets
`:feature:categories`	4	categories
`:feature:chat`	5	assistant
`:feature:dashboard`	12	dashboard, order of operations
`:feature:emergencyfund`	5	emergency fund
`:feature:goals`	8	goals, goal detail
`:feature:market`	4	opportunity
`:feature:onboarding`	5	onboarding (welcome, quick setup)
`:feature:settings`	12	settings, consents, erase
`:feature:transactions`	17	add transaction, transactions, receipt review, bank-alert drafts
`:feature:vehicle`	4	vehicles
Total	**115**	**25 screens + the gallery, in 26 test classes**

Every *`Screen.kt`* *in* *`:feature:`* and the lock screen in `:app` has a render here. The count is checkable in one line:

```
find . -path '*/src/test/snapshots/images/*.png' -not -path '*/build/*' | wc -l
```

Keeping this current

The pictures cannot rot quietly — a screen that stops matching its baseline turns the build red (`verifyPaparazziDebug`, inside `./gradlew unitTests`). What *can* rot is the prose beside them, so:

1. **Change a screen → re-record only that module**, look at the images, and commit the baselines with the change that caused them. `./gradlew :feature:<module>:recordPaparazziDebug`. A re-recorded baseline is not a passing test; it is a new claim about what the app looks like. 2. **Add a screen → add its screenshot test and a section here.** The test is the easy half: render the stateless `*Content`, no wrapper, no padding, theme pinned, light + dark + `fontScale = 2.0`. `docs/testing/screenshot-tests.md` §3 has the checklist and the three mistakes already made. 3. **Delete a screen → delete its section and its baselines**, in the same commit.

Related documents. `docs/testing/screenshot-tests.md` is the how-to for the tests themselves; ADR-0066 records why recording is never automatic and why the theme is pinned; ADR-0082 records why coverage was taken from five modules to fifteen and what it cost in the application module. `AI_Personal_CFO_screens.pdf` beside this file is the longer, prose-led walk through the app built from device screenshots; this document is the one that cannot go stale.