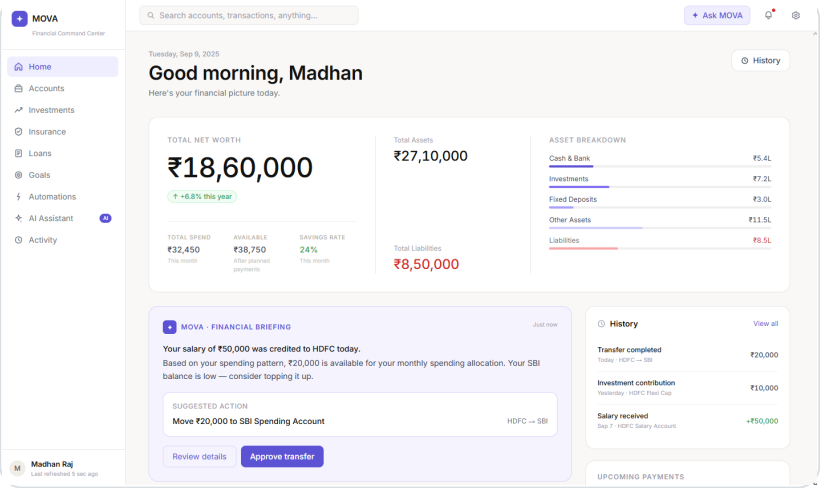


Designing a financial command center for context, intelligence and controlled action.

A self-initiated product concept exploring how AI can unify banking, investments, insurance, loans and goals - and help people turn intent into action without losing control.

ROLE	Product Designer / UX Architecture
SCOPE	Product strategy / IA / UX / AI interaction / systems / prototype
STATUS	Concept / prototype
DATA	Fictional / illustrative financial data



CORE EXPERIENCE

SEE -> ASK -> PLAN -> APPROVE

ACT -> AUTOMATE

A product-level view before the pixels.

MOVA is presented as a concept, so the case study separates design evidence from future validation.

PRODUCT OPPORTUNITY

Create a single interaction layer across a fragmented financial ecosystem.

The product combines financial aggregation with an intelligence layer that can interpret intent, prepare an action, request approval and preserve a traceable history.

WHAT I DESIGNED

- Unified information architecture across accounts, investments, insurance, loans, goals and activity.
- A cross-product experience model: SEE -> ASK -> PLAN -> APPROVE -> ACT -> AUTOMATE.
- Agentic interaction patterns for recommendation, review, approval and automation.
- Human control, trust cues, failure states and accountable activity history.
- Reusable UI components and a product-level visual language.

PROBLEM

Financial context is fragmented across institutions and applications.

OPPORTUNITY

Move from isolated views to one decision layer across the financial ecosystem.

CORE EXPERIENCE

See the picture, express intent, review a plan, approve the action, automate the repeat.

DESIGN STANCE

AI proposes and prepares; the user remains responsible for consequential approval.

WORKING HYPOTHESIS

Aggregation becomes more valuable when it reduces the work required to decide what to do next.

Different providers. Different products. One financial life.

MOVA starts with the assumption that the customer already has a financial ecosystem - the product must make it coherent.

ILLUSTRATIVE CUSTOMER CONTEXT

HDFC Bank	Salary account	Rs 1,42,500
------------------	----------------	--------------------

SBI	Spending account	Rs 18,750
------------	------------------	------------------

Zerodha	Investment account	Rs 7,20,000
----------------	--------------------	--------------------

LIC	Insurance	3 active policies
------------	-----------	--------------------------

SBI	Home loan	Rs 8,50,000 outstanding
------------	-----------	--------------------------------

THE PRODUCT OPPORTUNITY

Make the whole financial picture feel simple enough to understand - and actionable enough to do something with.

DESIGN QUESTION

What changes when the product can understand context across accounts before suggesting an action?

WORKING HYPOTHESIS

Aggregation becomes more valuable when it reduces the work required to decide what to do next.

Financial decisions are fragmented across apps.

The opportunity was not another destination. It was a decision layer across the customer's financial ecosystem.

FRAGMENTED

Salary, spending, investments, insurance and loans live in different places.

LOW VISIBILITY

Customers can see individual balances but not the whole financial picture together.

MANUAL ACTIONS

Moving money or managing recurring actions often requires repeated manual effort.

NO CONTEXT

A decision in one account may depend on what is happening across several others.

DESIGN CHALLENGE

How might AI help a customer understand their complete financial position - and safely turn intent into action?

SEE

UNDERSTAND

ACT

Design the product around usefulness, control and clarity.

PRODUCT GOALS

UNIFY

Bring fragmented financial information into one coherent view.

EXPLAIN

Help users understand what changed and why it matters.

RECOMMEND

Surface useful next steps from financial context.

CONTROL

Keep users in control of proposed and consequential actions.

AUTOMATE

Convert repeated intent into bounded, understandable rules.

CONSTRAINTS / BOUNDARIES

CONCEPT STATUS

No production data, stakeholder interviews or live integrations are claimed.

FINANCIAL CONSEQUENCE

High-impact actions require explicit user review and approval.

AI BOUNDARY

AI proposes and prepares; the user remains responsible for approval.

DATA

UI uses fictional / illustrative financial data.

VALIDATION

Testing and business metrics are proposed next steps, not achieved results.

Five principles keep an agentic product understandable and accountable.

01 CONTEXT BEFORE ACTION

Show the financial situation before asking the user to act.

02 EXPLAIN BEFORE EXECUTE

Make the recommendation and its relevant context reviewable before commitment.

03 HUMAN CONTROL OVER AUTONOMY

Increase user control as the consequence of the action increases.

04 MAKE CONSEQUENCES VISIBLE

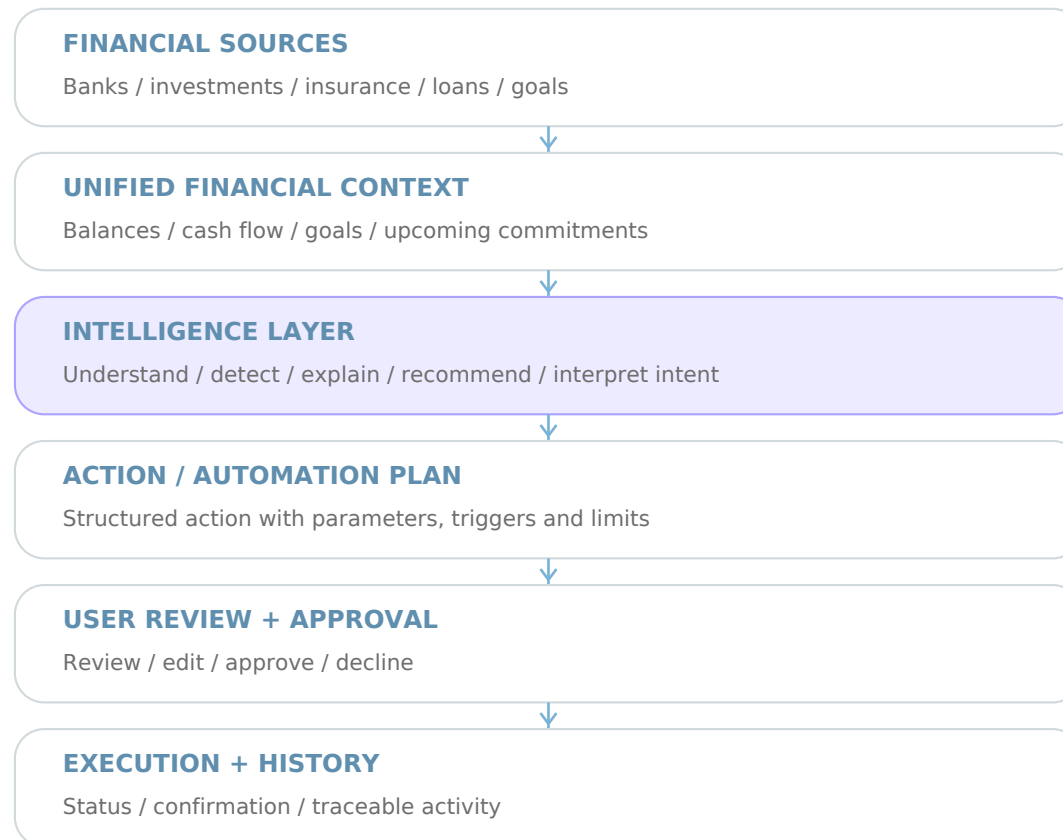
Show amount, destination, timing, conditions and limits before activation.

05 LEAVE A TRACE

Recommendations, approvals, automations and outcomes should remain traceable.

AI is the intelligence layer - not the destination.

Traditional product areas remain discoverable while AI works across them as a contextual interaction layer.

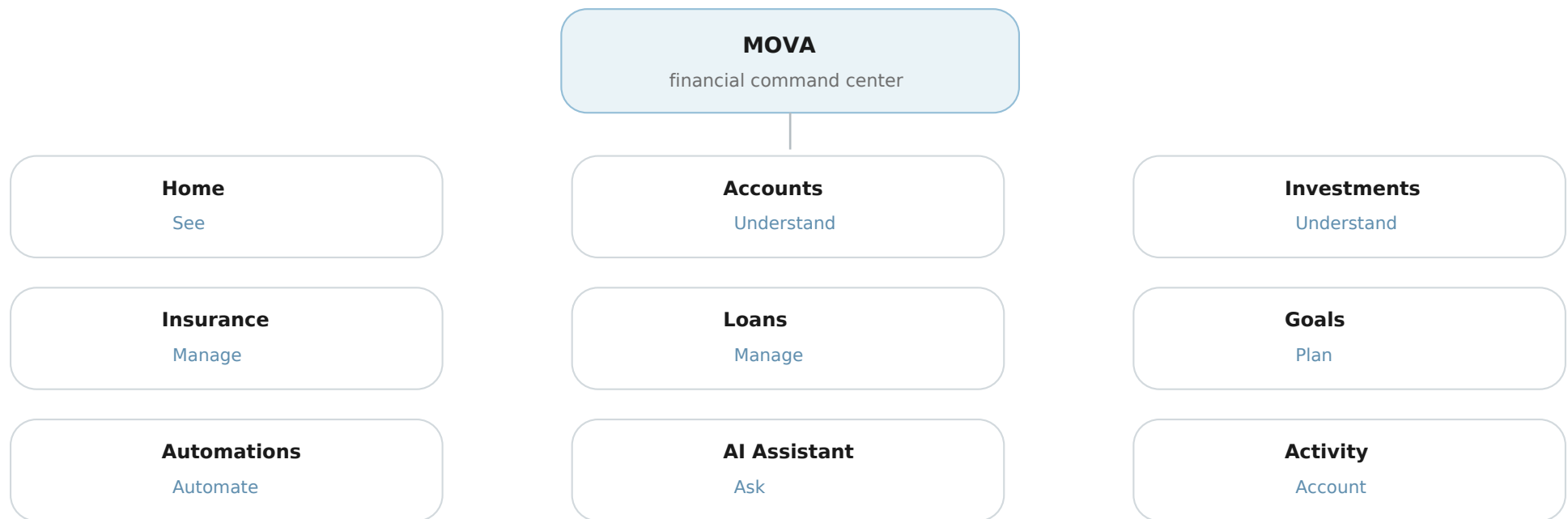


ARCHITECTURE DECISION

Keep the underlying product model visible. AI can interpret context and prepare actions, but it should not hide where the data or action comes from.

Keep the core product model visible while making AI available everywhere.

The IA separates durable product areas from the AI layer so users can inspect the source context before acting.



IA RATIONALE

AI is not a replacement for navigation. Users can still inspect the underlying account, investment, insurance, loan or goal before trusting an AI-generated next step. This supports progressive disclosure: overview first, detail when needed, action only after review.

SEE -> ASK -> PLAN -> APPROVE -> ACT -> AUTOMATE

The model turns a complex agentic interaction into a sequence users can understand.

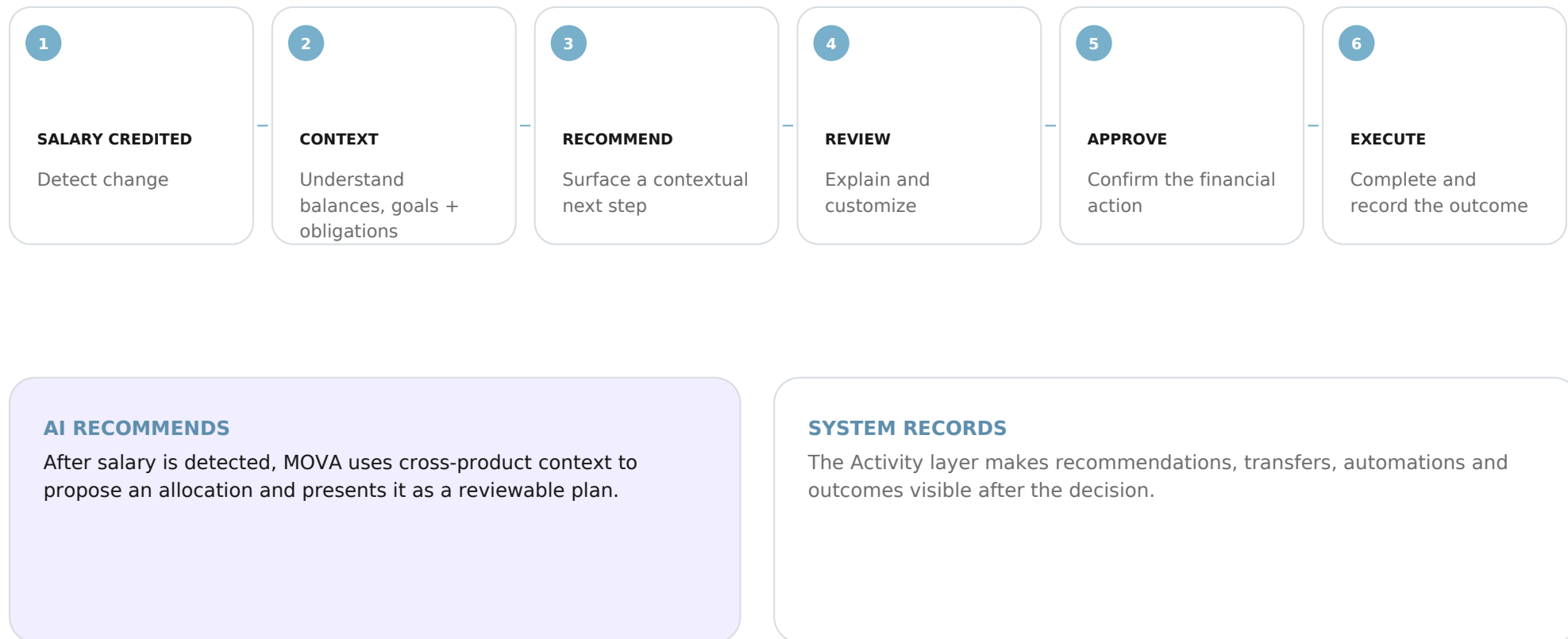


KEY UX IDEA

The transition from intent to action must be legible: AI interprets; the product structures; the user reviews; the system executes; history keeps the result accountable.

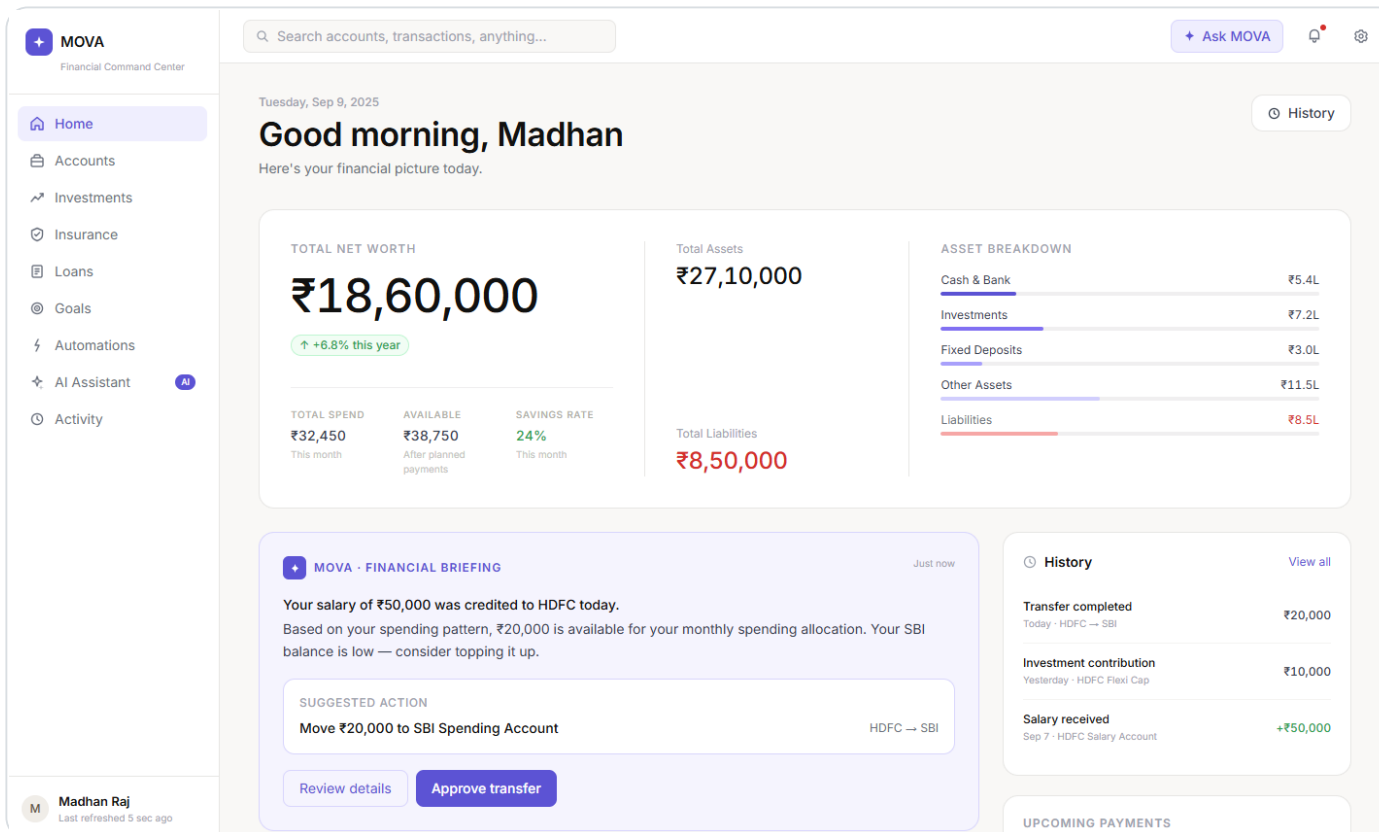
A salary-credit scenario becomes the product's anchor journey.

A concrete scenario lets the product architecture, AI behavior and control model work together in one story.



See the whole picture before taking action.

The home experience prioritizes financial position, what changed and what MOVA can help with next.



TOTAL FINANCIAL PICTURE

Net worth, assets, liabilities and supporting metrics are visible at a glance.

FINANCIAL BRIEFING

AI detects relevant changes and surfaces a suggested action.

HISTORY

Recent activity makes automated and manual actions visible.

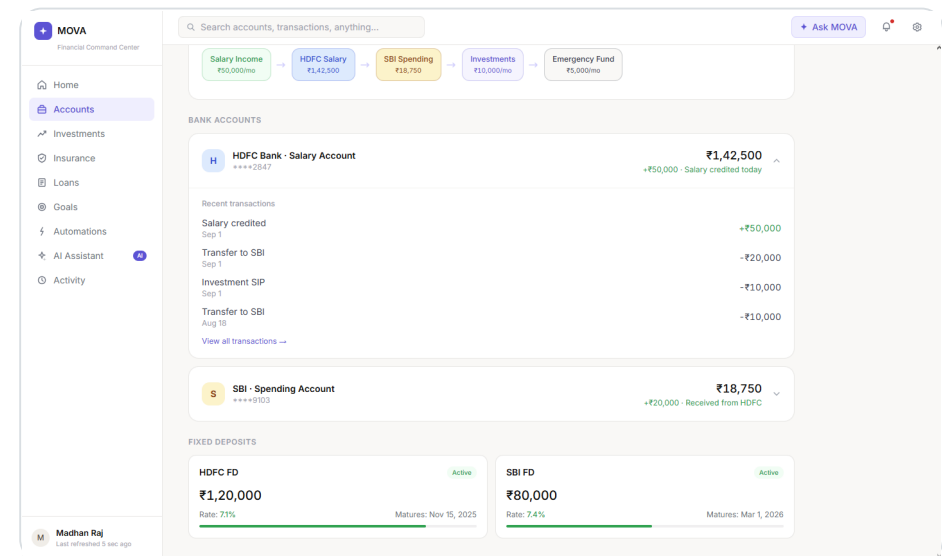
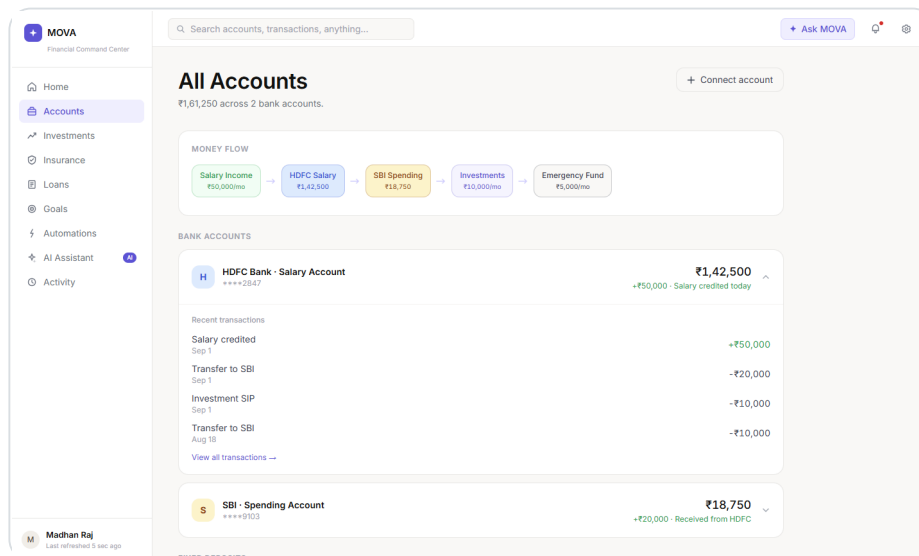
LOW COGNITIVE LOAD

DESIGN DECISION

Prioritize position -> change -> action instead of starting with transactions.

Aggregation is useful only when it leads to understanding.

Accounts are shown as connected financial context rather than as isolated balances.



MONEY FLOW

Salary -> spending -> investments -> goals.

PROGRESSIVE DETAIL

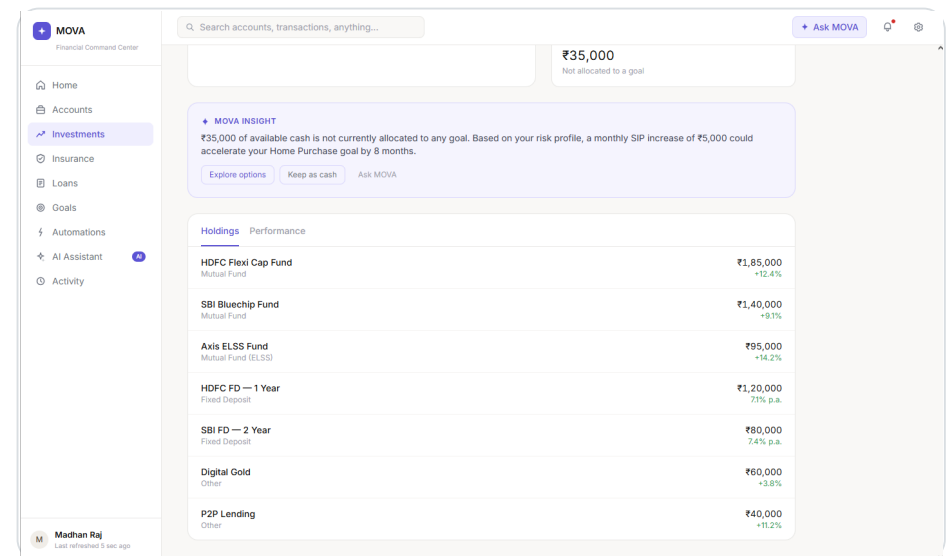
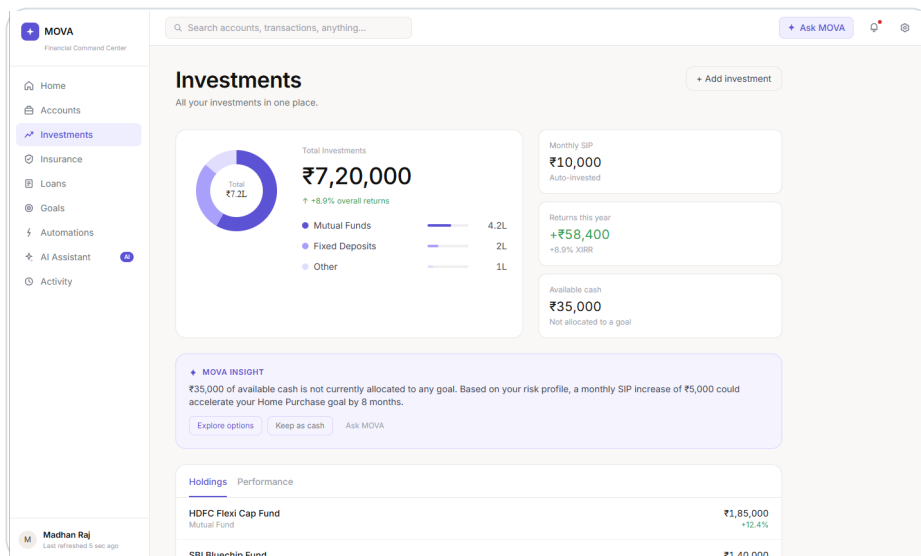
Keep the summary visible and reveal transactions or fixed deposits when users need them.

DESIGN DECISION

Represent accounts as relationships in a money-flow model rather than separate destinations.

Add context to the portfolio, not just performance.

The investment experience combines portfolio state with available cash and a bounded AI insight.



PORTFOLIO CONTEXT

Total investments, SIP, returns, holdings and available cash are shown in one mental model.

AI OPPORTUNITY

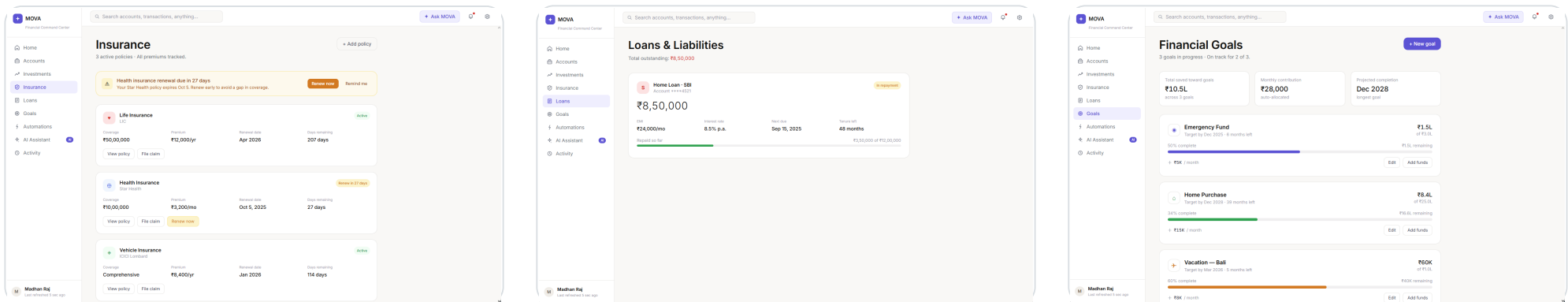
MOVA can surface a next step only after it has enough financial context to make the recommendation legible.

PRODUCT PRINCIPLE

An insight is useful when the user can inspect the source context and choose what happens next.

One experience across everyday money and long-term commitments.

Insurance, loans and goals use consistent layout patterns while preserving the semantics of each financial product.



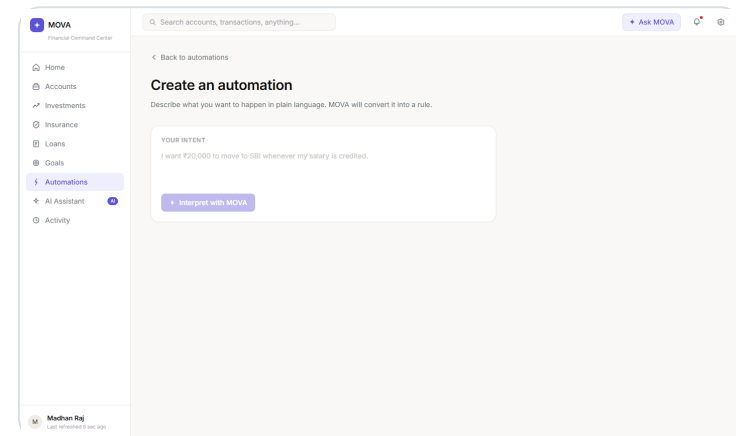
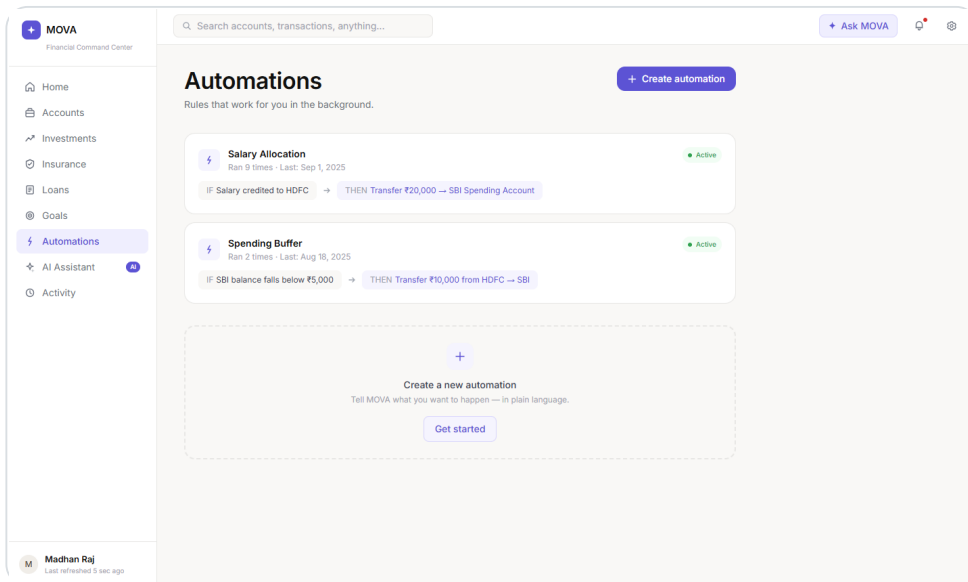
DESIGN PRINCIPLE

Keep product complexity in the system - not in the customer's head.

Use shared interaction patterns for status, alerts, progress and actions while keeping the mental model of each product intact.

From “I want this to happen” to a bounded automation.

Natural language is converted into a clear rule that the customer can review, edit and control.



EXAMPLE INTENT

“Move Rs 20,000 to SBI whenever my salary is credited.”

MOVA INTERPRETS

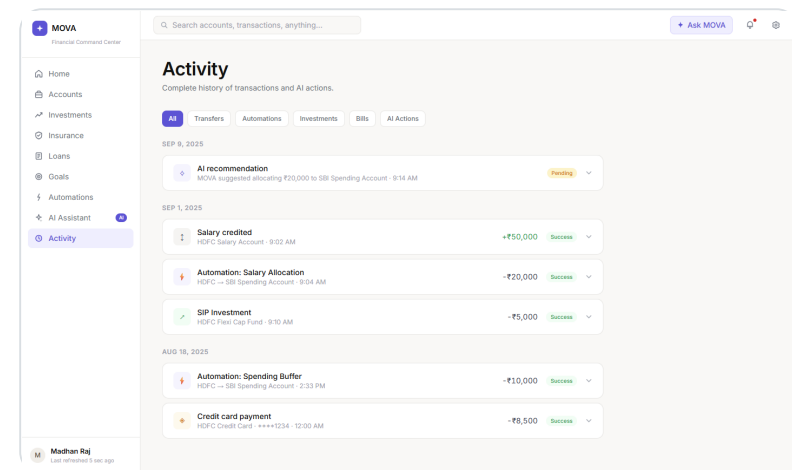
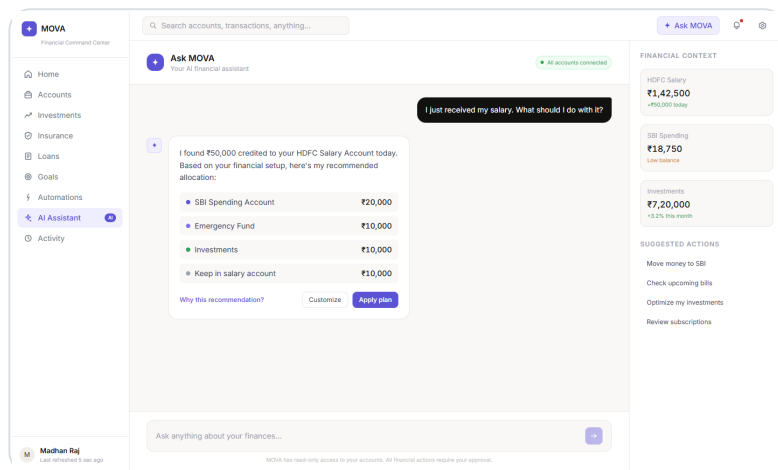
Source · amount · destination · trigger · limits

THEN

Review -> activate -> run -> record

AI becomes useful when it can explain, recommend and stay accountable.

The assistant is connected to financial context, while the activity layer makes AI-driven actions visible.



AI EXPERIENCE

Recommendation -> rationale -> customize -> approve. The action remains visible as a concrete object, not an abstract AI response.

ACCOUNTABILITY

Activity captures recommendations, transfers, automations and outcomes so users can review what happened after the interaction.

The interface uses a restrained neutral base with a focused purple action language.

Palette below is based on the supplied MOVA application screens - not the website accent.

COLOR PALETTE

	SURFACE	#FFFFFF
	CANVAS	#F9F8F6
	PRIMARY	#5C50D4
	PRIMARY LIGHT	#F0EEFF
	PRIMARY 2	#A8A0F8
	SUCCESS	#2E9840
	WARNING	#D17A1F
	DANGER	#E24B4B
	TEXT	#111110
	BORDER	#E5E3E1

COMPONENT FAMILIES

- Navigation + selected states
- Buttons + action hierarchy
- Summary / account cards
- AI insight / recommendation cards
- Inputs + natural-language prompts
- Status / alert / badge patterns
- Tabs + filters
- Progress + goal indicators
- Activity rows + history states

ACCESSIBILITY CONSIDERATIONS

- Clear visual hierarchy and readable type
- Labels in addition to color
- Visible focus and logical keyboard flow
- Adequate control size and spacing
- Explicit error / recovery messaging
- Consistent interaction patterns
- Do not claim conformance without actual testing

WCAG 2.2 considerations are used as a design reference; actual conformance should be verified through testing.

Design the system for the moments when AI is uncertain, blocked or wrong.

Agentic UX needs clear boundaries and recovery states - not just a happy-path prototype.

GUARDRAILS

- 01 Explain the recommendation**
Give users relevant context before approval.
- 02 Explicit approval**
Do not silently execute consequential actions.
- 03 Bound the automation**
Define amount, destination, trigger, limits and controls.
- 04 Make status visible**
Show pending, success, failure and paused states.
- 05 Preserve history**
Record what MOVA proposed, what the user approved and what occurred.

RECOVERY STATES

AMBIGUOUS INTENT

"How much would you like to move?"

INSUFFICIENT BALANCE

Show available balance and do not create a partial action silently.

TRANSFER FAILURE

"The transfer could not be completed. No money was moved."

LIMIT EXCEEDED

Pause the automation and surface which rule prevented execution.

USER REVERSAL

Provide edit, cancel, pause or disable controls.

Treat validation as the next product stage - not a fabricated outcome.

MOVA is a self-initiated concept using illustrative data. No live usability study, production telemetry or business impact is claimed.

USABILITY

Validate core tasks:

- Find financial position
- Understand recommendation
- Edit proposed transfer
- Create automation
- Recover from failure

MEASURES

Observe:

- Task completion
- Errors
- Time on task
- Comprehension
- Confidence
- Ability to explain the outcome

PRODUCT METRICS

After launch, define:

- Recommendation acceptance
- Action completion
- Automation success
- Trust / control signals
- Recovery rate

FIRST QUESTION TO VALIDATE

Can users explain what MOVA will do before they approve it?

From fragmented finances to one intelligent action layer.

The outcome is a product model: context -> intelligence -> controlled action -> accountability.

ONE VIEW

Bring accounts, investments, insurance, loans and goals together.

ONE INTELLIGENCE LAYER

Use AI to explain financial context and surface useful next steps.

ONE CONTROLLED ACTION MODEL

Translate natural language into reviewable actions and automations.

ONE ACCOUNTABLE HISTORY

Make recommendations, approvals, automations and outcomes traceable.

LEAD-LEVEL LEARNINGS

- Aggregation alone is not the product; value appears when context helps the user decide what to do next.
- Agentic UX is a system-design problem: intent, interpretation, action, approval and recovery must work together.
- Trust is part of the interaction model. Explainability, control and traceability are product requirements.
- The strongest abstraction in this concept is SEE -> ASK -> PLAN -> APPROVE -> ACT -> AUTOMATE.
- The next product stage is validation: test comprehension, control and recovery before introducing higher autonomy.

DESIGN FOR INTELLIGENCE - AND DESIGN THE CONTROL AROUND IT