USER FLOW

PRIMER | ONE PAGE | NOBODY WILL ASK IF YOU READ IT

A journey map is about a person. A flow is about a system.

WHAT IT IS

Every path through a system, including the ones nobody wants: the boxes-and-arrows account of states, decisions and dead ends. A journey map is about a person; a flow is about a system.

WHEN IT EARNS ITS PLACE

Before you design screens, and again after your first probe fails in a way you did not anticipate. It is also the cheapest way to find the states you forgot.

HOW TO MAKE ONE

- Name the entry points. All of them. People arrive from a link, a search, a person telling them, a notification at a bad moment.
- Map the happy path first, quickly. It is the least interesting part.
- Then triple it. For every step ask: what if they leave? what if it fails? what if they do not have the thing it needs? what if they already did this?
- Name every state, including empty, loading, partial, expired, denied, offline.
- Mark the exits. Where can someone stop, and what happens to their work?

WHAT GOOD LOOKS LIKE

- Unhappy paths outnumber happy ones — usually about three to one
- Every terminal state is deliberate. Nothing just ends.
- Error states say what the person can do, not what went wrong
- You can point to the state your probe actually broke in

THE COMMON FAILURE

A single line from start to success with two optional branches. That is not a flow, it is a wish. If nothing in your diagram can go wrong, you have documented an intention.

WITH A MODEL IN THE LOOP

LIFT

Genuinely strong at enumeration — it will list paths and catch dead ends faster than you will, especially the boring exhaustive ones.

LIE

It gives you the happy path only, unless you ask otherwise. Ask for the unhappy paths explicitly and watch the flow roughly triple.

CATCH

You know which failures actually happen to these people, in this domain, at this moment. An exhaustive list of theoretical states is not the same as knowing which three matter.