Integration Isn’t a Tech Problem — It’s a Clarity Problem

Marc Lilienthal & **Emma** Jane Rodrigues

tonies

braze

Hi all, I’m Marc and this Emma

I'm working as **Product & Tech Lead** at **tonies** and co-founded **Chordis**, an Activation Intelligence Platform.

Usually i am the person send in projects when 'things' are not moving fast enough or projects are stuck.

I’m **Retail Lead** at **braze**. We’re a customer engagement and AI platform helping retailers drive ROI and LTV while protecting margins.

We’ve loved working with Tonies; onboarding with Marc and the team was a uniquely clear and well-structured experience.

Most loyalty and CRM projects don’t fail loudly...

They don’t crash.
They don’t get cancelled.

They just... quietly underperform.

And mostly, the reason is the same:

Integration.

That’s why this session exists.

The goal of Integration

The deliberate combination of tools and services to create one coherent customer experience.

In practice, that means:

- Systems exchange information reliably
- Logic is explicitly agreed across teams
- The customer experiences it as seamless


tonies

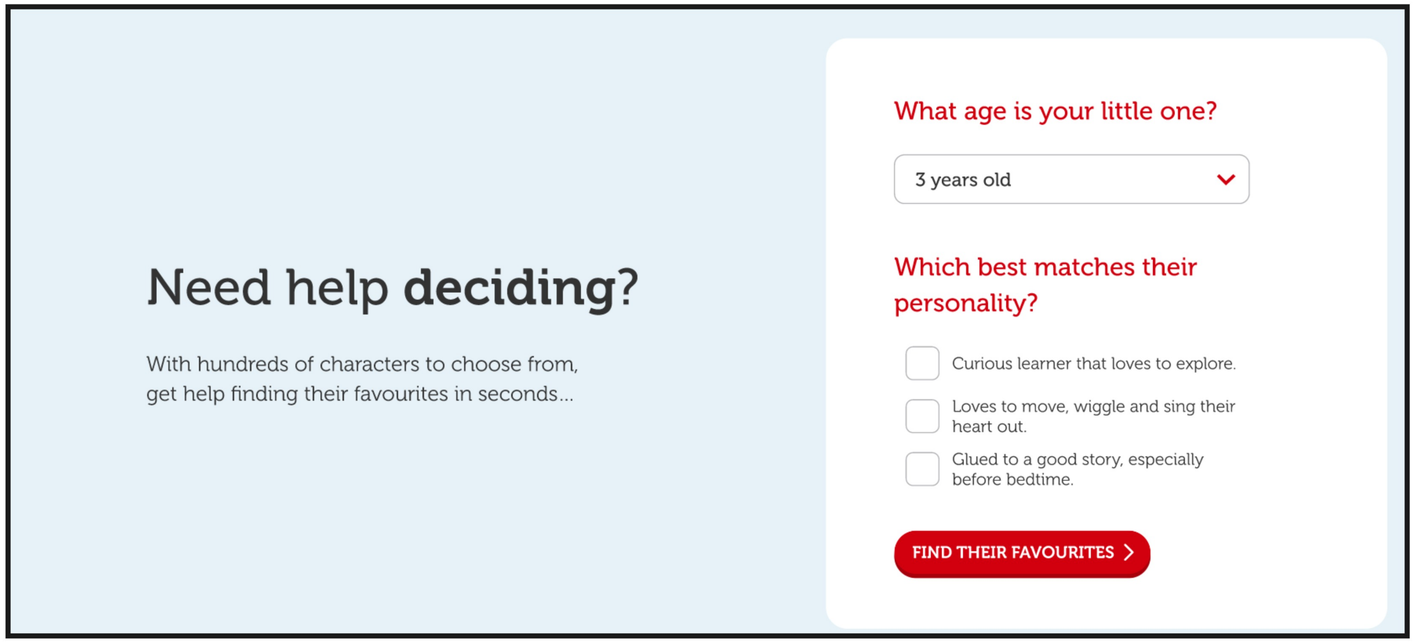

PLAY
LALALINOS

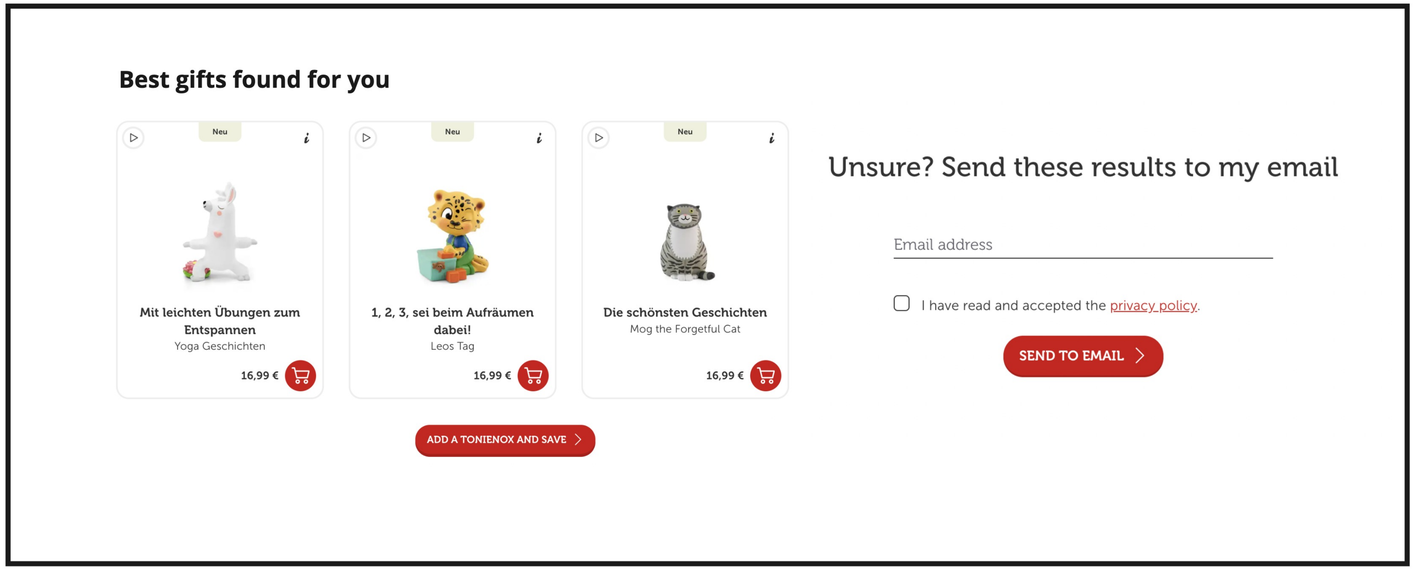

PLAY
tonies

Gift Finder
Example

Hypothesis:
Users are unable to make a decision. A small form to find the right gift injected on landing pages, will improve CVR.

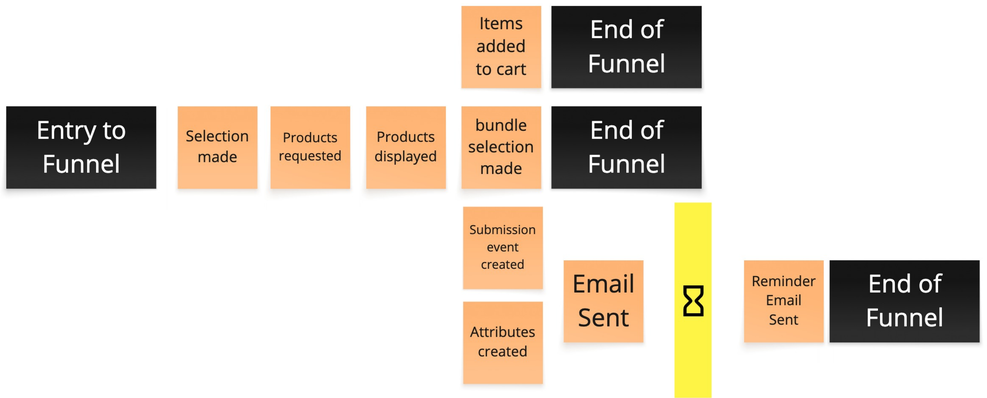
Easy feature, no?





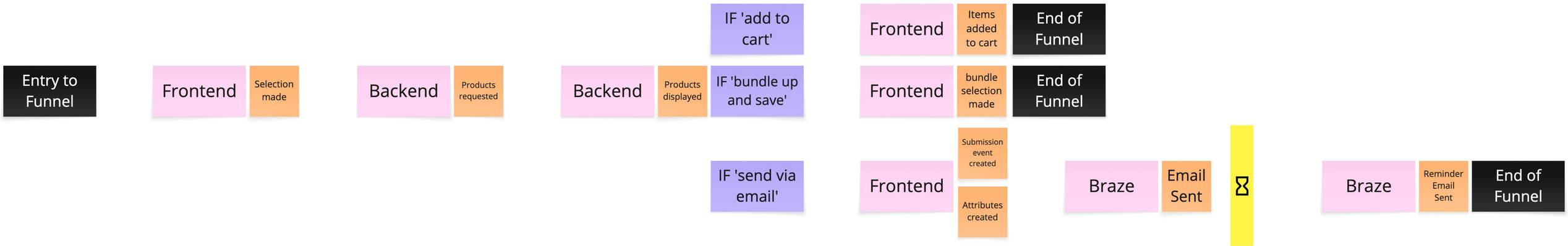
1. Discover Events

2. Create chronological order

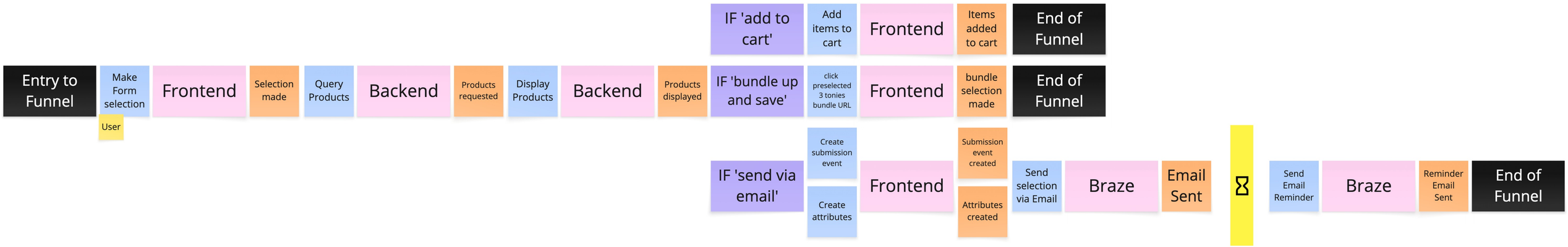


3. Give it some room

4. Add Systems and conditional logic



4. Add Commands so every system knows what to do.



YogaKitchen — a local yoga studio

Loyalty logic

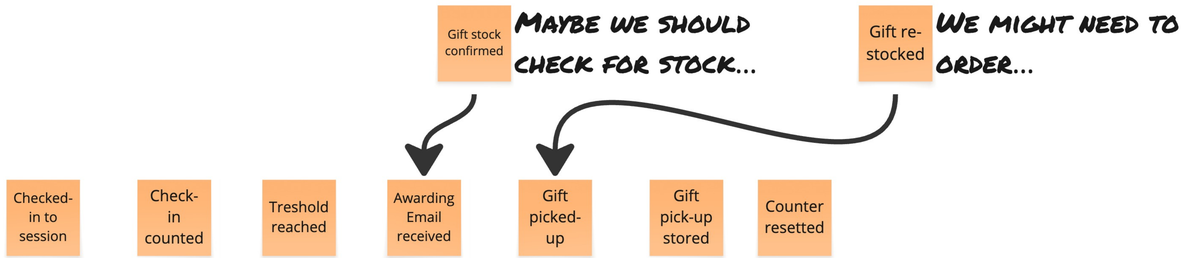
- Tiers based on class check-ins, annual reset

Rewards

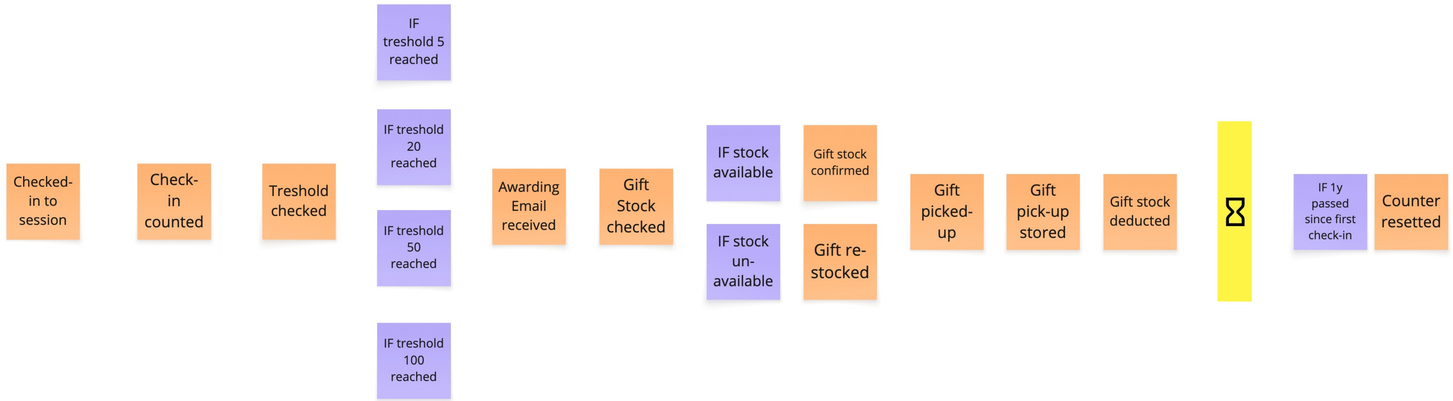
- 5 check-ins → Key Chain
- 20 check-ins → Coffee Mug
- 50 check-ins → Free Class Voucher
- 100 check-ins → Yoga Pants

We collect the Events

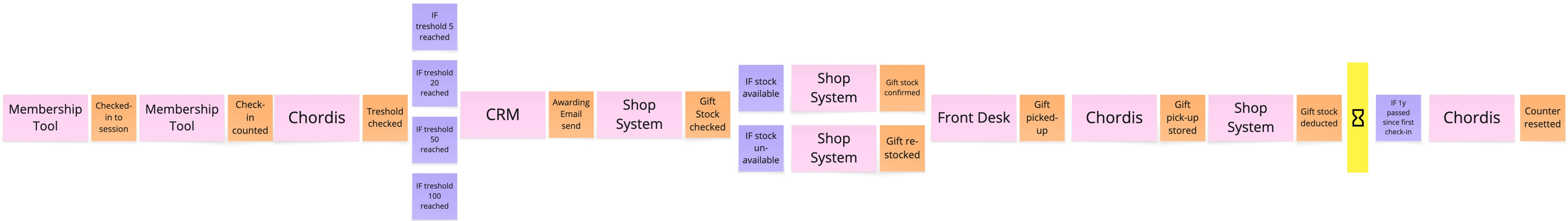
...and make sure they are complete



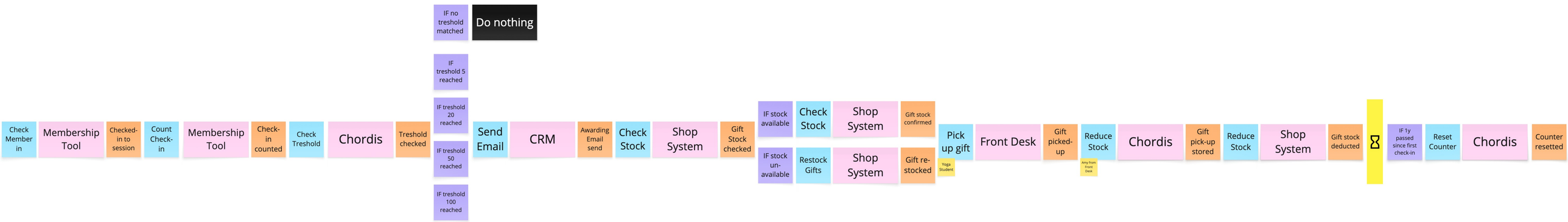
Create chronological order and conditions



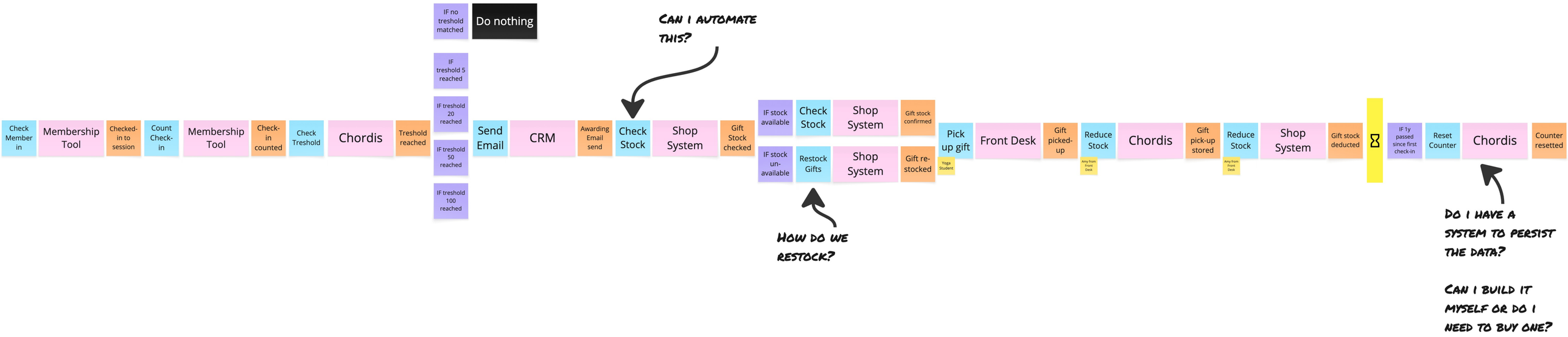
Identify Systems (think 'make or buy')



Finalize with Commands

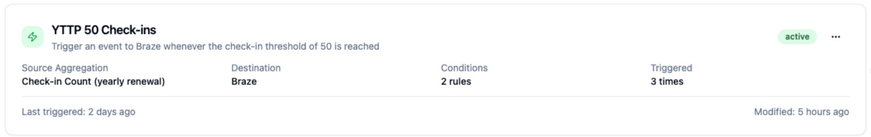


Now we have a story! What's still missing?



Now bring it to life during the project

Growth ads Trigger Logic



Tooling pushes

Code block

```
curl -X POST "https://rest.iad-01.braze.com/users
  /track" \
  -H "Content-Type: application/json" \
  -H "Authorization: Bearer YOUR_REST_API_KEY" \
  -d '{
    "events": [
      {
        "external_id": "USER_ID_123",
        "name": "issue_yoga_pants",
        "time": "2026-02-18T12:00:00Z",
        "properties": {
          "reward": "Yoga Pants",
          "threshold_checkins": 50,
          "program": "YogaKitchen",
        }
      }
    ]
  }'
```

Check Member in | Membership Tool | Checked-in to session | Count Check-in | Membership Tool | Check-in counted | Check Treshold | Chordis | Treshold reached

IF no treshold matched | Do nothing

IF treshold 5 reached

IF treshold 20 reached

IF treshold 50 reached

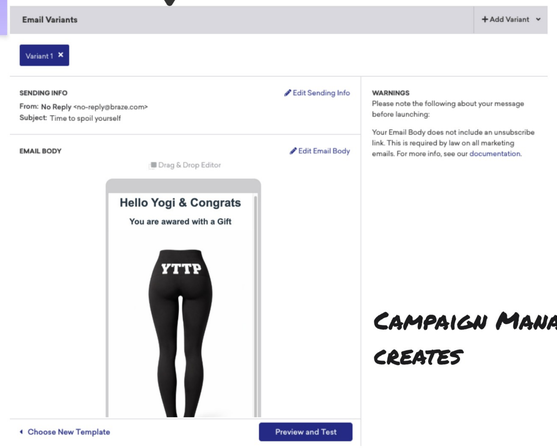
IF treshold 100 reached

Send Email | braze | Awarding Email send | Check Stock | Shop System | Gift Stock checked

IF stock available | Check Stock | Shop System | Gift stock confirmed

IF stock un-available | Restock Gifts | Shop System | Gift re-stocked

Pick up gift (Yoga Student) | Front Desk | Gift picked-up | Reduce Stock (Amy from Front Desk) | Chordis | Gift pick-up stored | Reduce Stock (Amy from Front Desk) | Shop System | Gift stock deducted | ⧖ | IF 1y passed since first check-in | Reset Counter | Chordis | Counter resetted



Campaign Manager creates

We need to create a 'HowTo' page and send it around

Now its your turn!

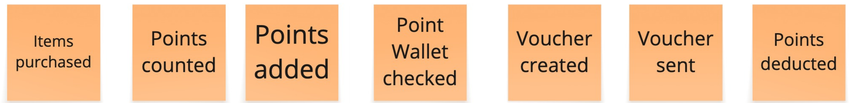
Find Post-Its and Instructions on your table.

Table assignment: Map the event flow

Scenario:
1 point per £1 · £5 voucher at 100 points · issued when threshold is reached

Your task:
1. Collect all Events as a group (orange Post-its)
   Remember: An Event is an interaction that already happened
2. Bring the Events into chronological order
3. Read the full flow out loud
   Have each team member confirm: *“Is anything missing?”*

It looks probably like this now



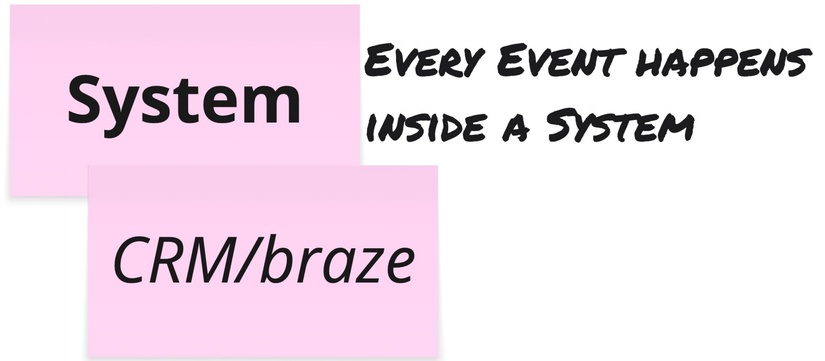
Next: Identify systems & conditions

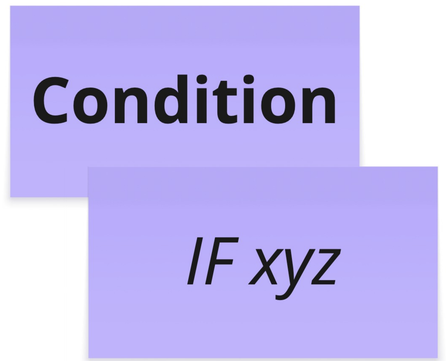
Systems (Pink Post-its)

- Add the system to each event (Shop, CRM, App, POS, etc.)
    - If unclear write “???”, his highlights integration gaps

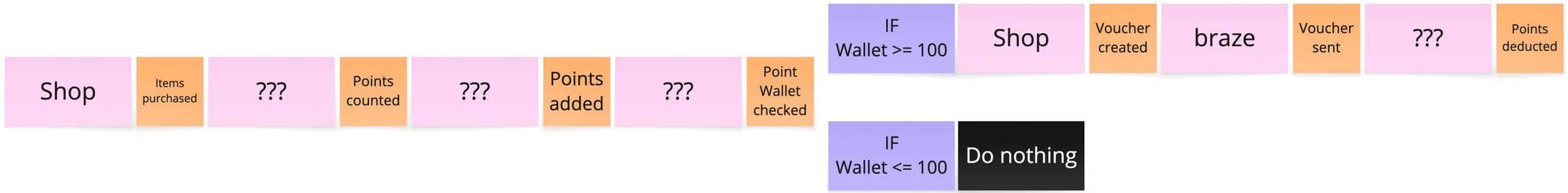
Conditions (Purple Post-its)

- Add decision points:
    - “If X happens → do Y”
- These define **logic conditions** (e.g. If balance ≥ 100 → issue voucher)





It looks sth like this now



Next: Enrich commands to have the full flow

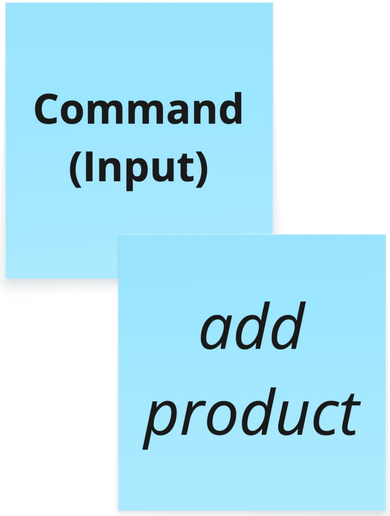
Commands (Blue Post-its)

- Every system needs instructions
- Add the Command before systems
- Be specific. Avoid vague language.

Key concept

- Command = request
- Event = confirmation

That difference prevents chaos and allows us to assign a command even to a person.

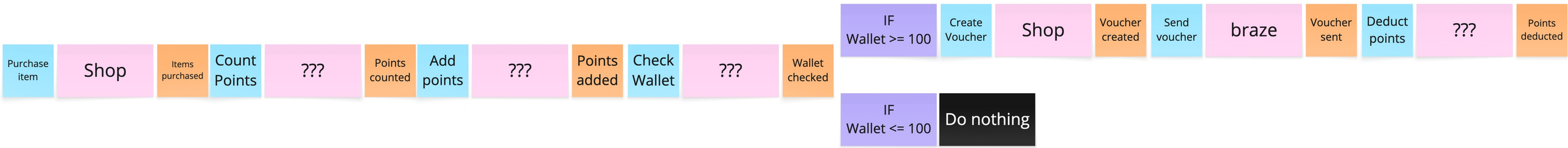


Final step

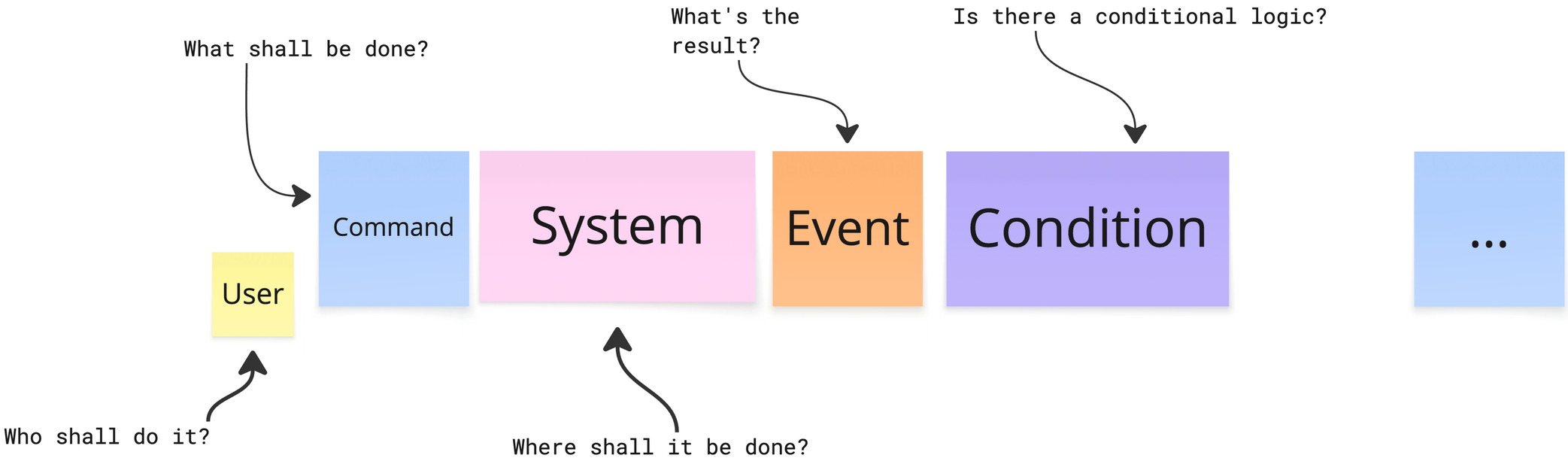
Read the entire flow out loud.

- Does every instruction have a result?
- Does every result have a clear owner?
- Do we have all Systems available that we require?
- Can we already make a 'make or buy' decision?
- Is there anything missing?

Potentially it now looks like this



At one glance



What Just Happened

1. You made logic visible
2. You surfaced hidden ownership decisions
3. You identified integration/system gaps
4. **You reduced unclarity by structuring complexity**

Three takeaways to take back to your team

Use kick-offs to create clarity — not just alignment

Don’t just share business requirements.
Sit down and map the logic together.

Don’t be afraid of integration conversations

You don’t need to be technical to talk about flows, ownership, and decisions.

Remember: this is logical thinking, not engineering

If the logic is clear, implementation becomes a lot less scary.

We hope this was fun ;) Is there any question?

Wanna stay in touch?

Marc

Emma

Thank You