

Enigmaa

A password generator where you set the parameters yourself and can recalculate the same password without any record.

WHEN

From Sep 2026

PLATFORMS

Mac · iPhone

MY PART

Requirements, rules, trade-offs

01

Background and goal

BACKGROUND

Many services ask you to change your password regularly. Random strings cannot be remembered, and the combinations I made up myself ran out, and I lost track of what I had changed to.

PROBLEM

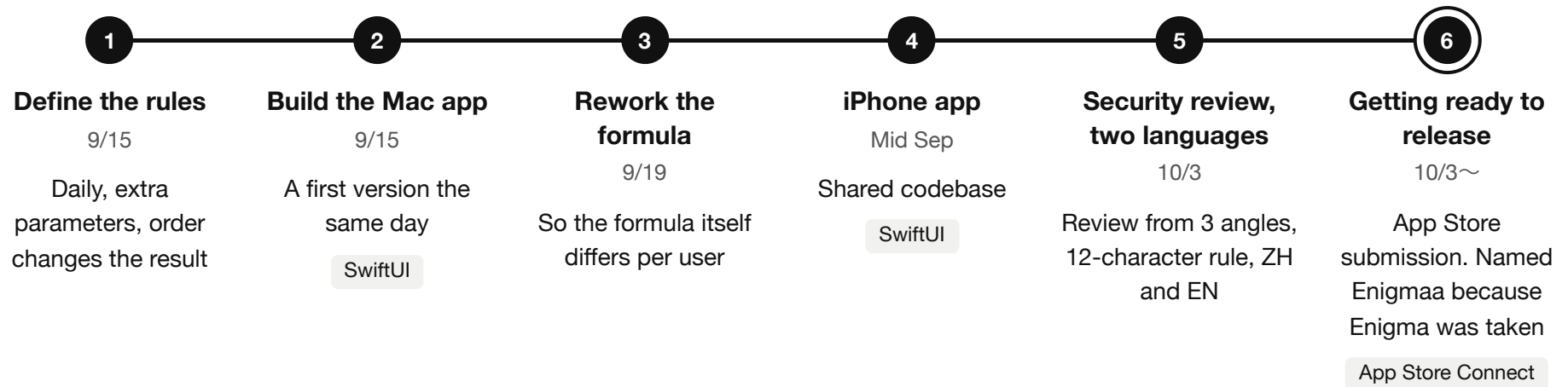
Relying on records is a problem when the records are lost. But if the formula is the same for everyone, it cannot be made public and the only protection is keeping the parameters secret.

GOAL

Not a password manager, but a generator whose parameters you control. The same conditions give the same password with no record, and the formula itself differs for each user.

02

Process



03

The idea

Recalculate, not store

Store it The usual idea

Make a password

→

Save it

→

Retrieve it when needed

Enigmaa Recalculate it

Enter the parameters



Pick a date



The same result comes out

What is kept is the conditions, not the answer. If the record is lost, the same conditions bring back the same password.

04

The hardest part

Making the formula really differ for each user

First design 9/15

Parameters

+

Date



One formula for everyone



Protection rests on keeping
the parameters secret

After the rework 9/19~

Parameters

+

Date

+

The first master
password



A formula that differs
per person



Even with the formula
public, the
parameters alone
cannot reproduce it

Who did what I asked for the formula itself to differ per user. Claude designed and built the formula.

The cost: when you change device, you need the master password you first set.

05

Trade-offs

What I did not choose, and the costs I accepted

- ☐ Put the service name in the parameters (too complex)
- ☐ A separate passphrase for the formula (too much hassle)
- ☒ **Recalculate from parameters, date and the first master password only**

COSTS (CHOSEN KNOWINGLY)

- Passwords are changed all at once about every six months, so every service shares one set in between.
- A device change needs the first master password.
- The first master password cannot be changed later.

06

Problems and fixes

Four key calls

PROBLEM

The formula was the same in every copy of the app, so protection rested only on keeping the parameters secret



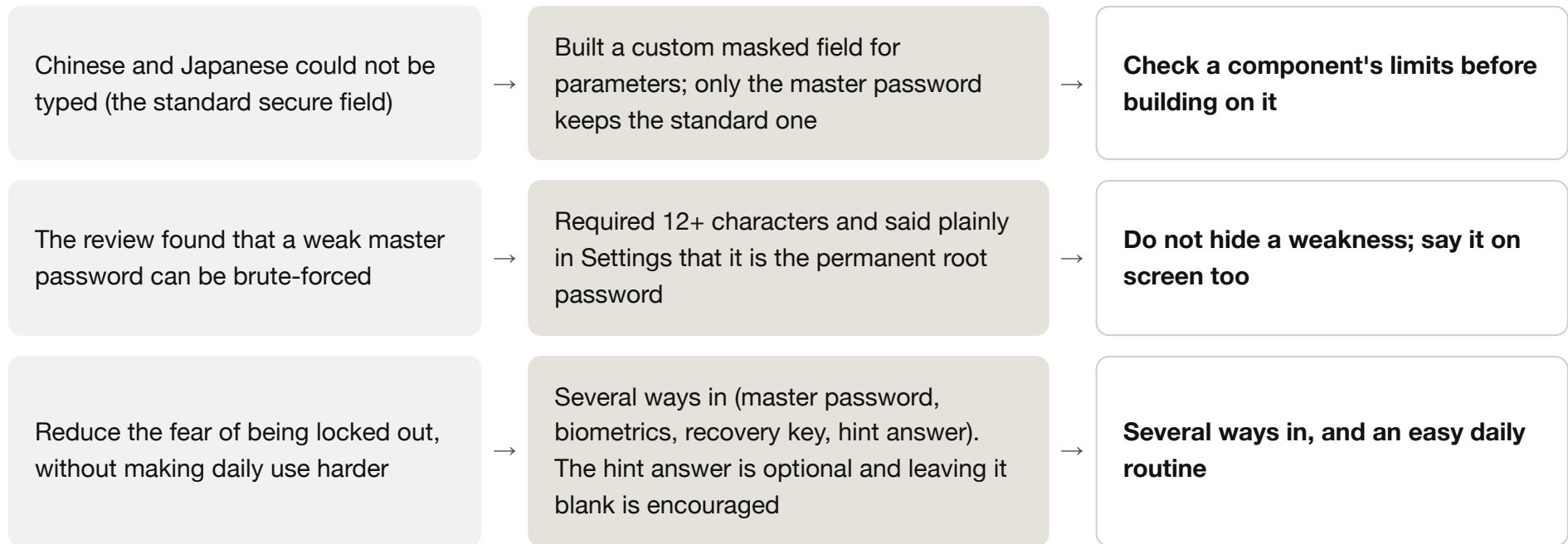
WHAT I DID

Mixed a key made from the first master password into the formula



PRINCIPLE

Build it so it stays safe even if the formula is public



07

Working split

I set requirements and trade-offs; AI designed and built the formula

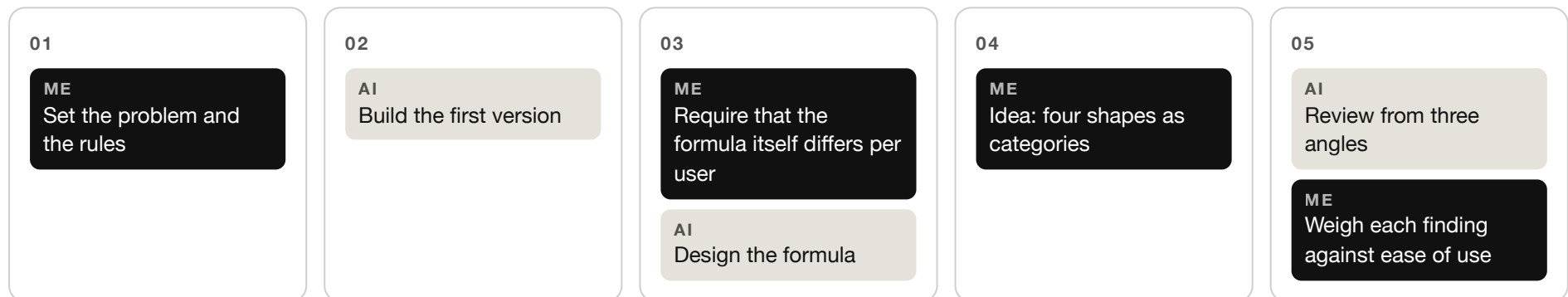
MY ROLE

Requirements

Rule design

Trade-offs

Release



06

ME

Submission, a device
walkthrough video

AI

Prepare submission
material

08

Results

2

Platforms
Mac, iPhone

2

Languages
Traditional Chinese,
English

53

Automated checks
(as of 5 Oct)

3

Review angles
crypto, App Store,
data

1.0

iPhone version
released
on the App Store

Claude's view

The collaborator's opinion, not written by the author.

Claude's view Collaborator's opinion	
WORKED WELL - Kept the definition "a generator, not a manager" to the end, and stayed with recalculation rather than storage. - Did not hide the weak point (master password strength) and said it on screen too. - Split what is required from what is optional, using ease of use as the test.	COULD BE BETTER - The formula was reviewed only by us; no third-party cryptography review. - Released only recently, so there is no record of other people using it yet. - That the first master password cannot be changed remains a cost.
AUTHOR	Judges by everyday ease of use Can say why it is needed in their own words Does not add fussy features

A personal project. The detailed specification of the formula is not published.