

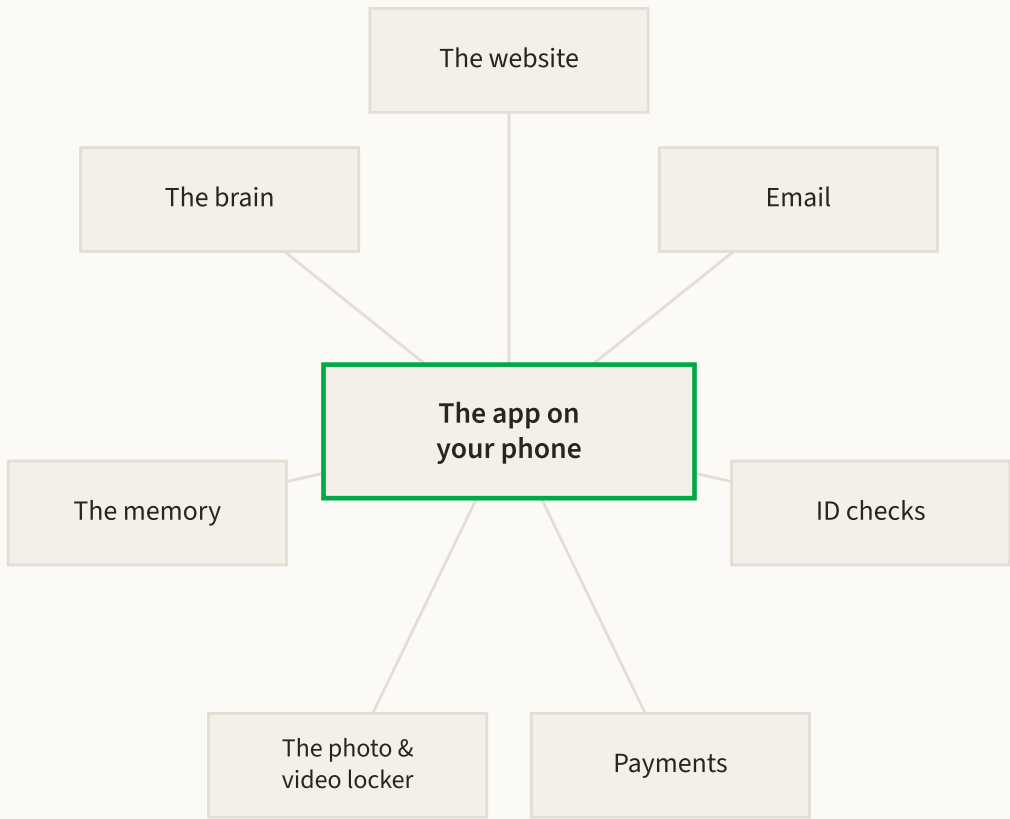
Running Greenlight

You asked what I've been doing, how available I am, and what I need help with. Here it is, with pictures.

1 OF 9

What Greenlight actually is

One app, but it leans on several other pieces to work. Here is the whole picture.



Every one of these can break. When one does, it is me who gets it back up.

2 OF 9

What I've built and kept running

430

changes since May 14

171

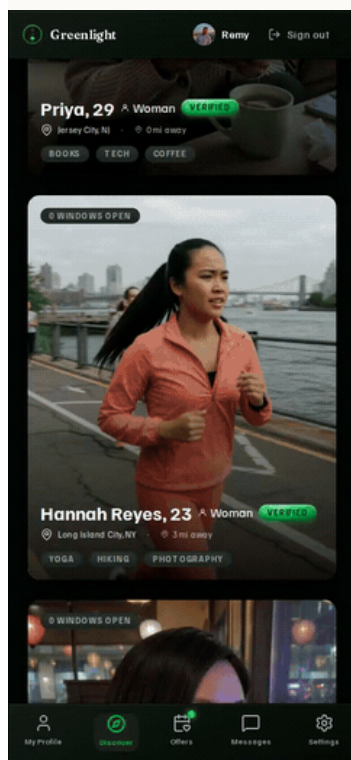
changes in the last 31 days,
about 5 a day, weekends
included

247,000

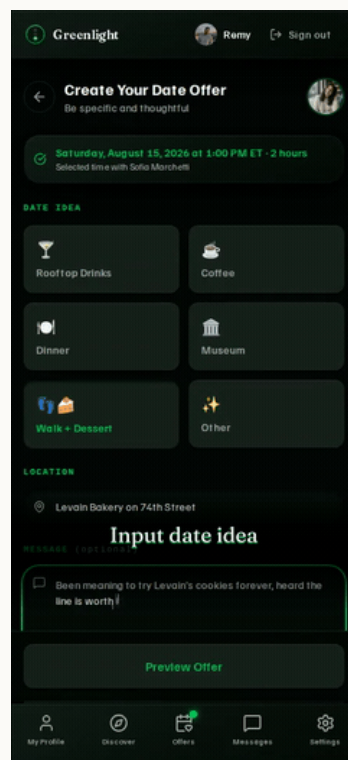
lines of code

A "change" is a piece of work that was built, tested, reviewed, and released to members.

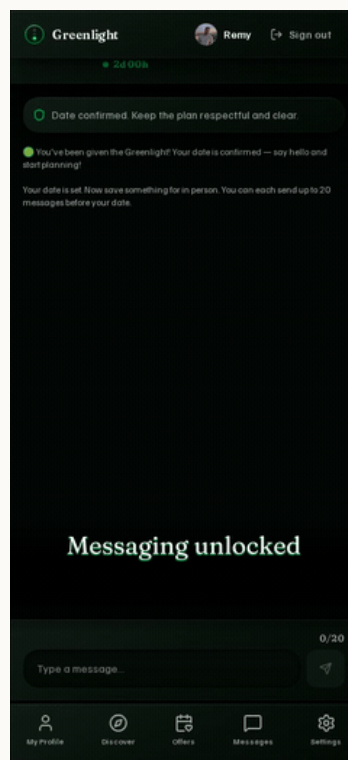
The app members use



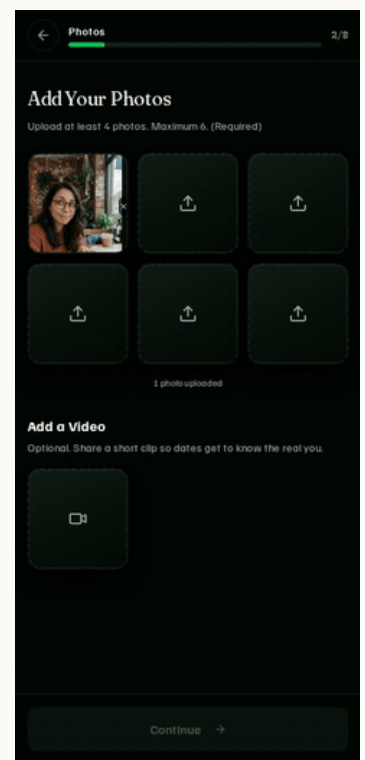
Browse nearby



Send an offer



Chat after yes

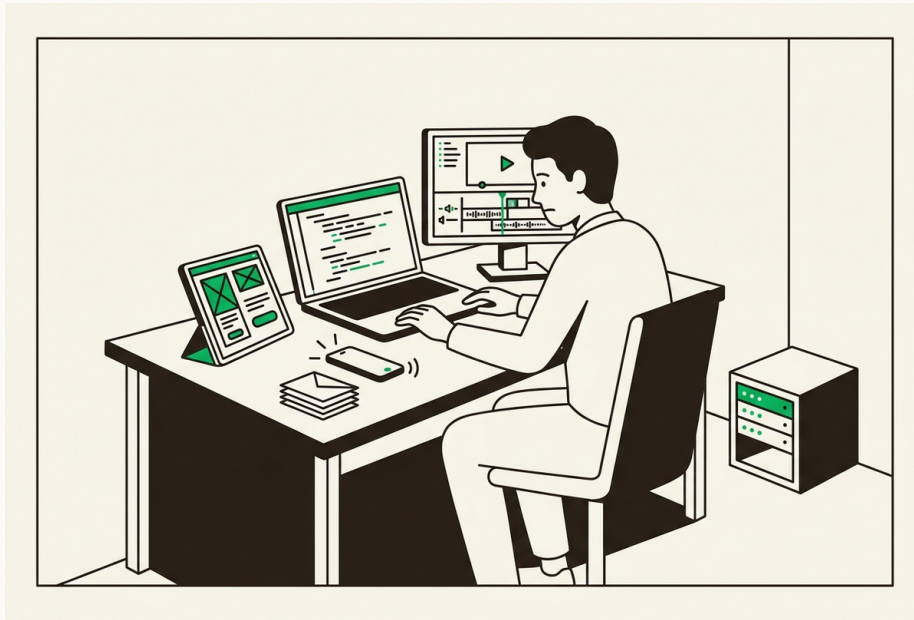


Build a profile

These four are short moving clips on the web version: <https://greenlight-running.pages.dev/>

- Browse people near you, ranked by shared interests.
- Send a date offer with a day, a time, and a place. Accept, decline, or counter.
- Chat unlocks after a yes. Chat is capped so it stays about the date.
- Build a profile with photos and a short video, trimmed right on your phone.
- Pause your profile, set your weekly availability, dark mode, report a problem.

The part nobody sees



One person keeping the pieces in section 1 running

- The brain, the memory, and the locker live in the cloud. I set them up, lock them down, and keep them running.
- Every release: build it, test it, update the memory safely, release it, check it is healthy, have a way to undo it. Several times a week.
- Error tracking so I hear about problems before members email.
- Payments through Stripe, including the billing page where members cancel.
- ID checks through Didit so verified badges mean something.

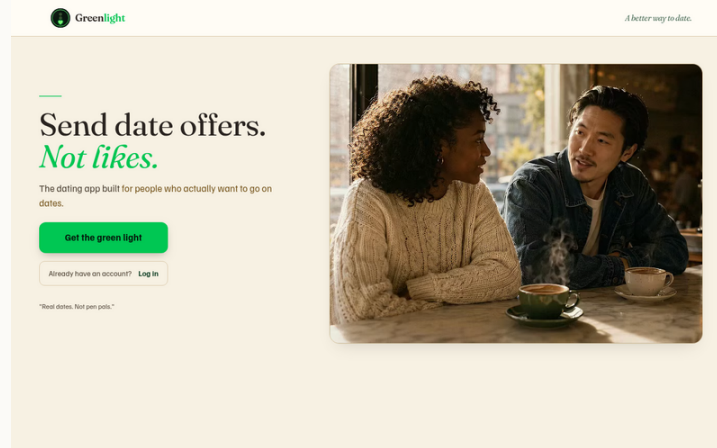
Keeping members safe and keeping us legal

- Making sure nobody can get into someone else's account or swap photos between accounts.
- Keeping evidence when someone is reported, because New York law requires it, and deleting it when it expires.
- Telling members in New York what they are entitled to when an account is banned for fraud, as the law requires.
- Written policies for copyright takedowns and for child-safety reports.
- Keeping the privacy policy true every time the app changes.
- Bot protection on signup and login.

Design

- Redesigned the whole app and website to the cream look, plus pricing and legal pages, dark mode, colour themes.
- Wrote the design rules so the app looks like one thing, not ten.

Videos and event materials



The site, same design system as the app

- 24 member videos edited: captions, end card, audio cleanup, encoding, a private review site for the team.
- 2 demo walkthrough videos with your voiceovers.
- Helped with the event banner: supplied our logo as an SVG and references.
- Signup QR codes.

Emails to members

- The email system itself, plus welcome, confirmation, reminder, and weekly availability emails.
- The post-event invite campaign: 60 members, tested on a phone in light and dark mode before sending.

Admin tools, so support does not need an engineer

- One screen per member: subscription, offers, blocks, photos, notes.
- Buttons for pause, delete, email change, refund lookup, photo review, ID review.
- Email templates and batch send to women, men, or everyone.

These exist so that anyone on the team can handle a member request without me.

Nothing was planned. The foundations were built from nothing

Greenlight did not start with a plan. It started on May 14 as an auto-generated skeleton from a website-building tool: 114 files, no working server, no rules for how the app and the server talk to each other. There was no feature plan, no written agreement for the data, no tests, no design rulebook, no way to measure usage, and no alarm for when something broke. What existed was a shell with nothing behind it. The first members were mostly close friends of the team. They were still real people with real accounts, signing up while all of that was still missing. Every item below is something a product normally has before it opens to real people. Here, each one had to be built afterwards, around live members, while also shipping features.

Days Greenlight ran without it

Usage measurement (knowing what members actually do)



Finally existed on Sep 10

Error alarms (knowing something broke before a member tells us)



Finally existed on Sep 9

Written plan and instruction manual (what the product is, how it is built)



Finally existed on Sep 1 to 13

Full walk-through test (a robot signs up, browses, pays, every release)



Finally existed on Aug 14

Design rulebook (one set of colors, spacing, type)



Finally existed on Jul 25

Tracked database changes (every change recorded and reversible)



Finally existed on Jul 19

Automated tests (code that checks the code)



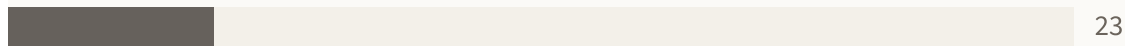
Finally existed on Jul 17

First security work (locking doors that were open)



Finally existed on Jul 17

Legal pages (privacy policy, terms)



Finally existed on Jun 6

Day 0 is May 14, 2026.

The rulebook that says how the app and the server talk to each other, what every request looks like and what comes back, existed on day one as a 36-line placeholder with nothing in it. It is now over 7,000 lines and covers sign-in, profiles, photos, offers, chat, blocking, billing, ID checks, and safety reports. That rulebook was written after the server was already running, one piece at a time, while fixing whatever the missing piece had broken.

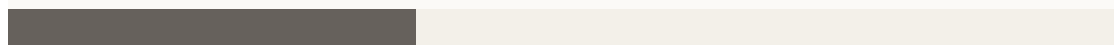
None of this is visible to members. All of it is why the app stays up.

4 OF 9

The last two weeks

The team asked what has been done recently. Here is September 1 to September 15.

Aug 15 to Aug 31: 46 finished changes



Sep 1 to Sep 15: 126 finished changes



A finished change is a reviewed, tested change accepted into the app. Going live is a separate step I also do, usually within a day or two.

What those 126 changes were

WHAT KIND	HOW MANY
Security and privacy	32
How we build and test (so mistakes get caught before members see them)	30
Fixes for things that broke or were slow	17
Design and visual consistency	16
Features members can see	10
Emails to members	8
Admin tools for the team	8
Usage measurement	4
Marketing site	1

A few examples, in plain English

- Discover now shows people with more shared interests first, and says what you have in common.
- Members can swipe through someone's photos instead of seeing one.
- If someone blocks or unmatched, any pending date offer is cancelled automatically.
- Videos filmed on iPhones now play; broken uploads show a message instead of failing silently.
- The signup city box now understands "NYC" and neighborhood names (the event bug).
- Emails were fixed three times for Gmail dark mode on phones.
- People banned for fraud get the notice New York law requires, and cannot re-register with the same ID.
- Chat messages and offer notes are scanned for scams; the privacy policy was updated before that went live.

The to-do list: done, and still to do

There was no to-do list either. I wrote one on September 5 and 13, 110 days in. Here is where it stands, checked against the live tracker on September 15.

431

finished changes since May

36

items closed

36

items still open

Signup, discover, offers, chat, settings



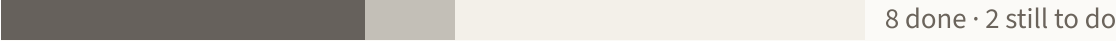
Security and privacy



Emails and notifications



Hosting and releases



ID checks and photo moderation



Usage measurement



Testing and release process



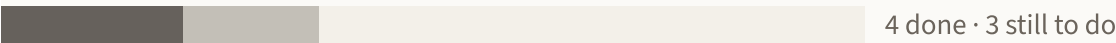
Payments



App-to-server rulebook and speed



Admin tools



Legal and compliance



3 done · 4 still to do

Design rulebook



3 done · 1 still to do

Counts are by item, not by size. One item can be an afternoon or a week.

Still to do

Members

- Blocking or unmatching does not yet cancel an offer that was already accepted, only pending ones.
- A member can turn phone alerts on but there is no way to turn them off.
- A "save your profile as a draft" feature is built but switched off until we decide to launch it.
- Eight icon-only buttons have no label for screen readers.

Safety and moderation

- No admin button yet to undo an approved ID check.
- No admin screen yet for evidence holds on reported content; today that is done directly in the database.
- A policy decision is still open: when the same person is reported repeatedly, how long do we keep the evidence?
- The scam scan on chat has no timeout, so a slow outside service could stall it.

Security

- Temporary upload links are not yet locked to one purpose and one person.
- The staff permission system exists but has no screen to use it from.
- Rate limits do not yet cover file uploads.

Emails

- Some email graphics use a green that disappears in Gmail dark mode, and the automated check that should catch it does not look at images.
- A safety-notice email keeps a member's address longer than our own deletion policy says.

Behind the scenes

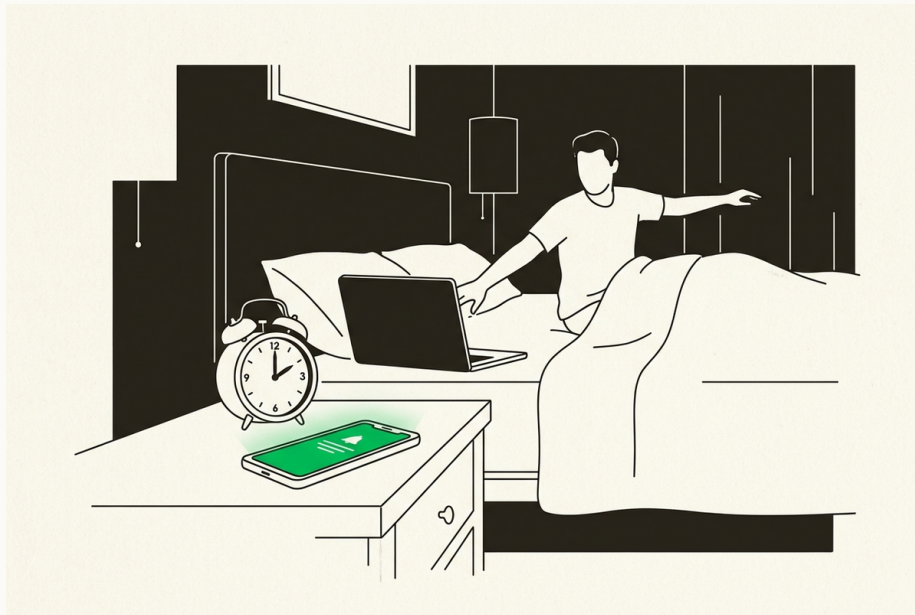
- Two speed measurements still to run on the Discover and chat screens now that the database changes are live.
- Test coverage minimum is set far too low and needs raising to what it actually is.
- One flaky test, a handful of dead files, and a large admin export that can run an unbounded database query.

Every line above needs an engineer. None of it is the support desk.

6 OF 9

When things break

Nobody sees this work. It happens at night and on weekends, and it does not wait.



An incident does not wait for business hours

The triple charge

What happened: A member who subscribed was charged three times for one subscription.

To the member: Three charges showed up on their card.

What I did: Found the cause, shipped nine fixes, had them live in production that same night, and handed refunds to Jude.

The signup lockout

What happened: Two people could not finish signup because the photo checker wrongly rejected them and deleted their photos.

To the member: "Your photo was rejected," with no way forward.

What I did: Fixed it and released the fix that same night.

The event bug

What happened: At the New York event, a member typed "NYC" as their city and the app said no such place existed.

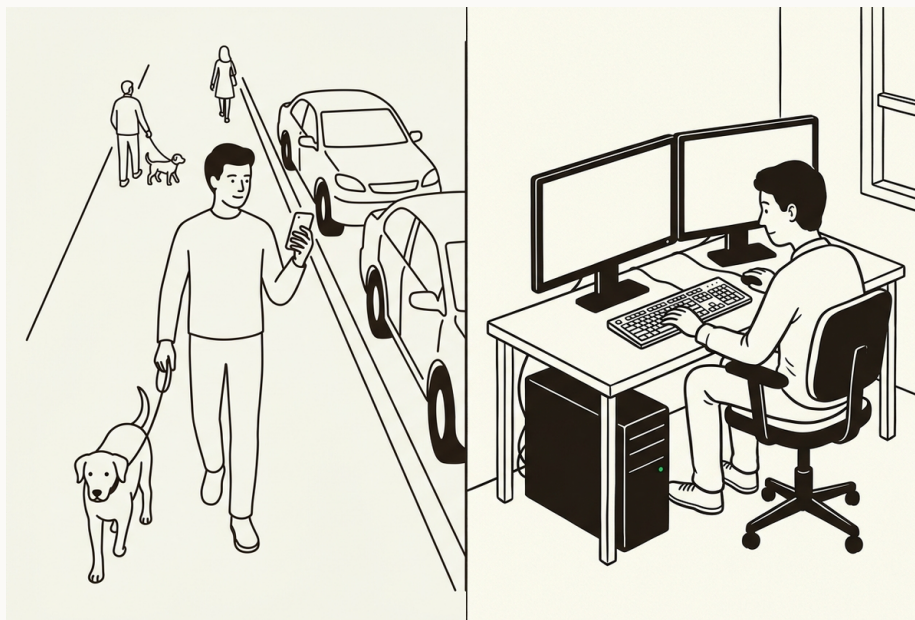
To the member: Stuck at signup, in front of us.

What I did: Fixed it that same night.

In all three cases there was no gap of days between the report and the fix. Most software teams take several days, often a week or two, to get a reported bug fixed and released. Each of these was live the same night it was found, by one person, around a day job.

7 OF 9

How available I am



Most of the job can happen from the phone, just slower. The big work needs the desk.

I am always available. I am just not always at full speed.

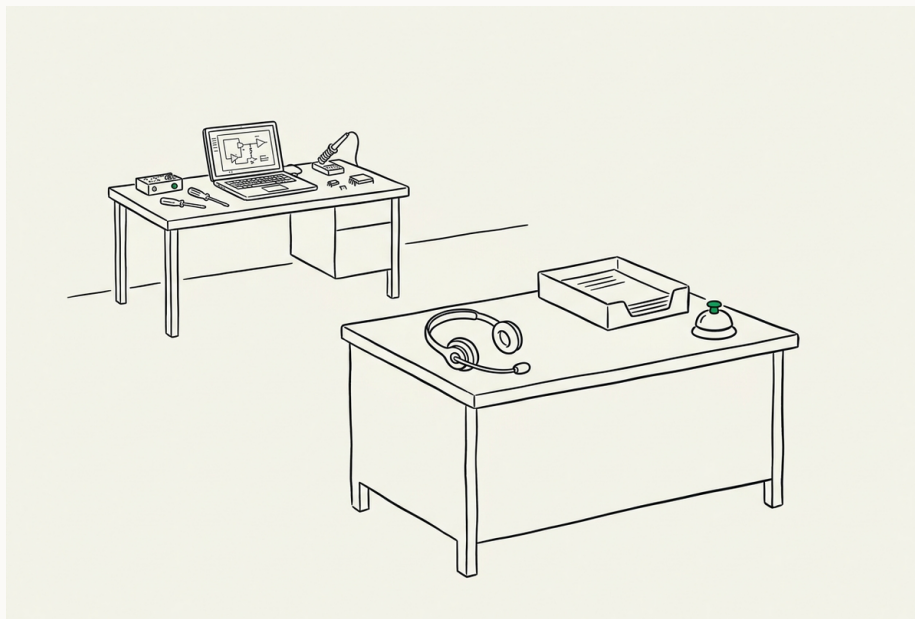
During the day I am at work, and part of it I am out walking the dogs. From my phone I can do most of the job: read messages, check a dashboard, answer a question, review a change, push a small fix. It all goes slower there. And I cannot walk the dogs and work the phone at the same time, so those stretches are dead time.

Two kinds of work cannot happen on the phone at all. Video editing needs the PC, full stop. Anything big needs it too: a major overhaul, screenshotting the app across a whole flow, building several pages for one feature. That is desk work, and it happens evenings and weekends.

Being reachable is not the same as being at full speed.

8 OF 9

The support desk is a separate job



Two different jobs, currently one person

I can be the full-stack engineer. Front end, back end, releases, fixes, security. That is the job I am doing. The support desk is its own job, and right now it lands on me alone, on top of that. I can still help with it. I cannot be the only person doing it.

WHAT	WHAT IT TAKES	WHO CAN DO IT
Reading the support inbox and replying to members	A Gmail login and a one-page list of standard answers	Anyone on the team, this week
Member account requests: pause, delete, change email, "I can't log in"	A button in the admin screen for each one	Anyone on the team, this week
Checking the ID review and photo review queues	The admin screen plus the written rules	Anyone on the team, after one walkthrough
Being the first person who answers "is this working?"	Try it on your phone, then tell me if it is really broken	Anyone on the team, this week
When a real bug shows up in that inbox, tell me. That is the handoff. Everything else in that inbox is not engineering.		

9 OF 9

The ask

Someone on the team takes on the support desk with me, starting with the inbox, so it is not one person doing both jobs.

I keep building and keep the app running.

If we want the app to move faster than it is, that is a second engineer, not more of me.

The to-do list in section 5 is the engineering work. It does not include the support desk in section 8. Both are real. Only one of them needs an engineer.