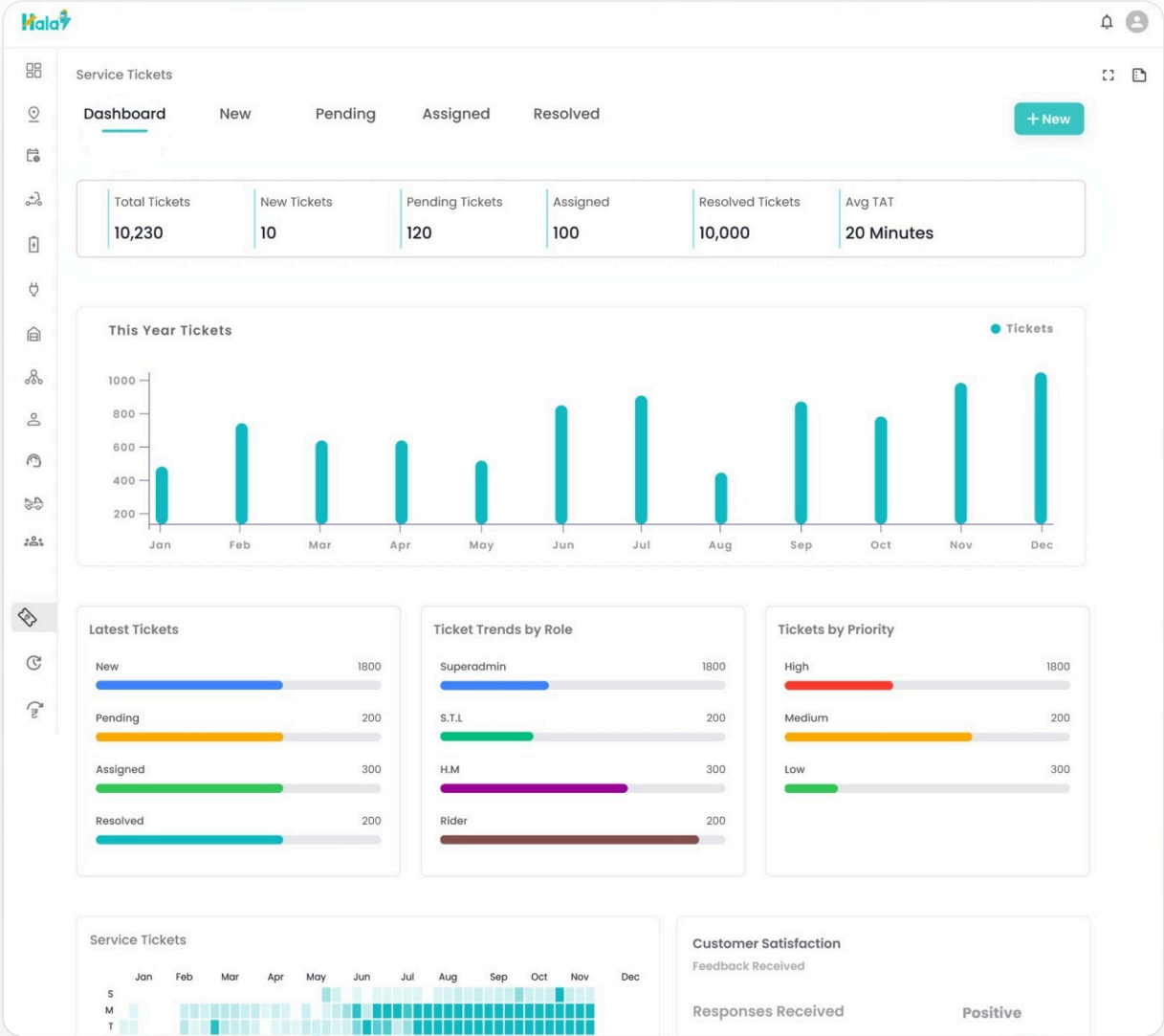


Service Ticket System

One ticket lifecycle across the rider, technician and ops apps of an EV fleet.



ROLE	PRODUCT	SURFACES	TIMELINE	RESULT
Lead Product Designer	CercleOS fleet platform	4 apps and dashboards	2024 – 2025	~60% fewer complaints



ABOUT ME

Hi, I'm Moazzam

Product designer, 3+ years across enterprise B2B, SaaS and B2C in the UAE and India. I turn complex workflows into simple products.

Gloify	Mashreq Bank’s corporate credit platform	2025–26
Hala Mobility	EV mobility app and fleet dashboards	2024–25
Delphi Consulting	Wasl, Mubadala, Emirates Steel Arkan, Tabreed	2024
Piche	SaaS and web products	2023–24

AT A GLANCE

One ticket, one owner, one history

PROBLEM

Service issues lived in phone calls, WhatsApp and Excel. Nobody owned a ticket end to end.

MY ROLE

Lead Product Designer, with a senior designer reviewing. Research through engineering handoff.

WHAT I DESIGNED

A role-aware ticket system: rider app, technician app, ops dashboard and hub manager view.

RESULT

~60%

fewer rider service complaints after launch.

FOUR SURFACES, ONE SHARED TICKET

Rider app

Raise a ticket and track its status

Technician app

Accept, fix, close with photo proof

Ops dashboard

Assign, escalate, report

Hub manager view

Volumes, owners, bottlenecks

Five roles, five channels, no owner

Hala rents EVs to riders and runs fleets for business partners. As hubs multiplied, every service issue became a relay race with no baton.

Rider	→	Phone call	no status
Support	→	WhatsApp escalation	buried
Hub manager	→	Excel sheet	out of date
Technician	→	Word of mouth	no record
Leadership	→	Asks around	no data

No owner

Nobody was clearly responsible after the first hand-off.

Idle vehicles

Every vehicle waiting on a fix was lost revenue.

No data

Leadership had no reliable view of volumes or resolution time.

What the hubs showed, and what I designed

Interviews with hub managers, technicians, riders, support and leadership · shadowed hub shifts · audited the phone, WhatsApp and Excel trail · benchmarked Fleetx, Rapido Ops, Yulu, Bounce

WHAT I FOUND IN THE FIELD

Tickets weren't lost. Ownership was.
Issues moved from phone to WhatsApp to Excel, and nobody was clearly responsible after the first hand-off.

Technicians closed jobs with almost no notes
Typing on-site, next to the vehicle, simply didn't happen.

Every team needed a different slice of the same truth
Riders wanted status, ops wanted queues, hub managers wanted bottlenecks, leadership wanted trends.

DESIGN DECISION

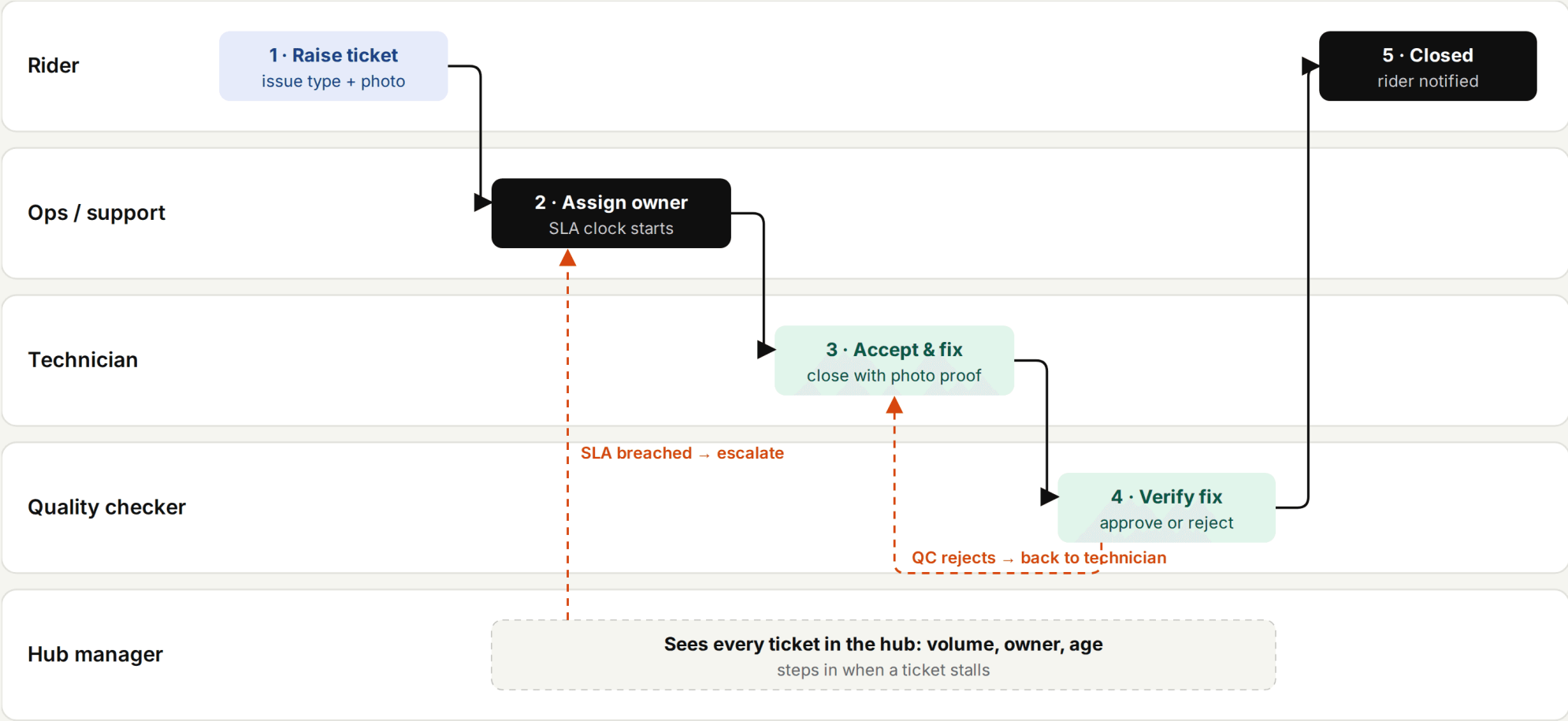
→ **No ticket without an owner**
Every ticket always has an assignee, a next step and an SLA clock. Stalled tickets escalate.

→ **Close with photos, not paragraphs**
Guided close with minimal typing, plus a quality check before sign-off.

→ **One ticket, role-based views**
One ID, one lifecycle, one history. Each role sees only the actions it can take.

One ticket moves across every role

Every hand-off reassigns the ticket explicitly, so it is never ownerless.



The cost: agreeing one lifecycle across five roles before any screen was built.

The payoff: no reconciling data between tools, and leadership reporting came almost for free.

No ticket without an owner

Owner, status and age on every ticket

Ops see who holds each ticket and how long it has waited.

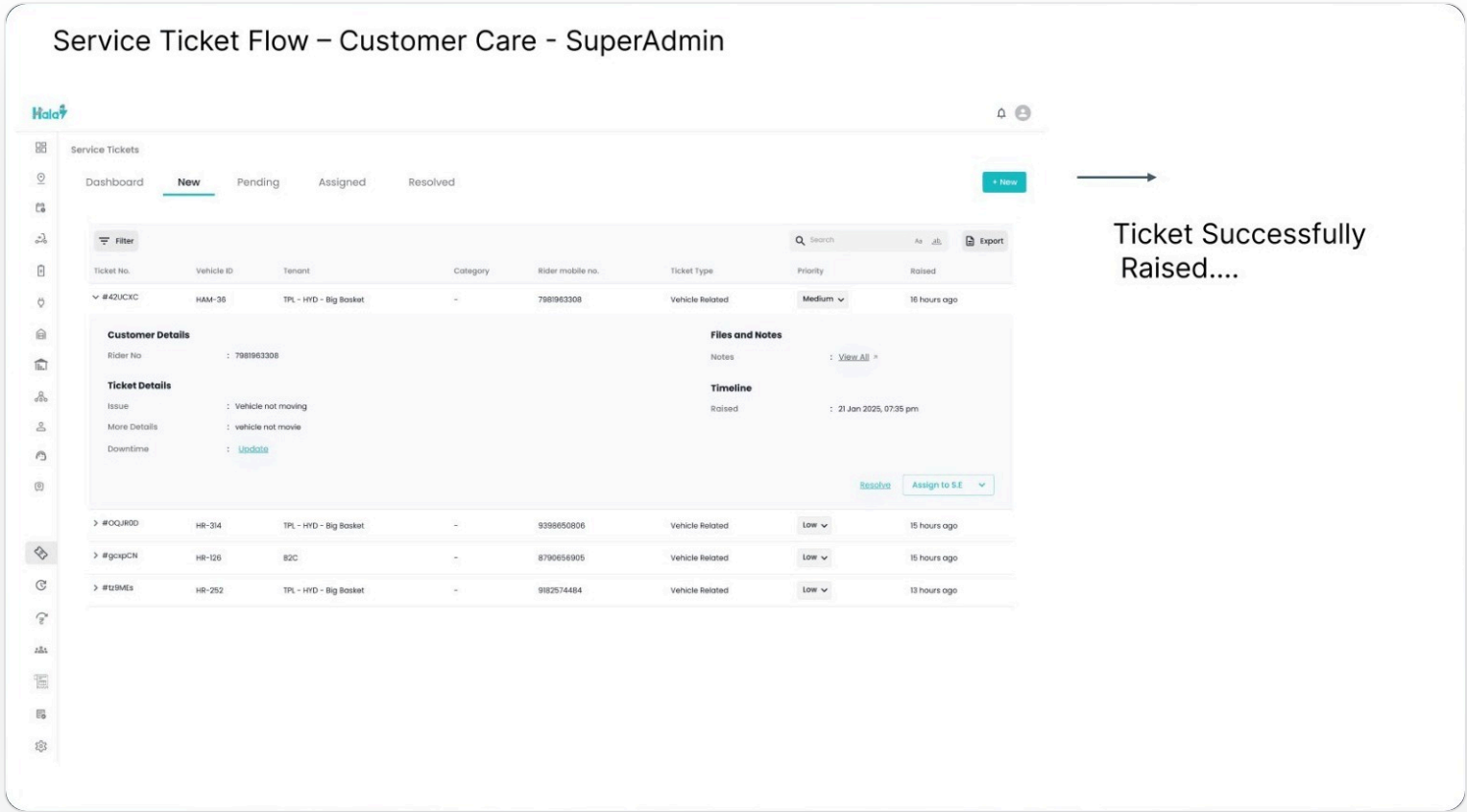
Hand-offs are explicit

Reassigning is a deliberate action, never a forwarded message.

Stalled tickets escalate

An SLA clock starts at assignment, so nothing silently ages.

The cost: a little more effort for ops at creation. It’s exactly the friction that ended the chasing.



Technicians close jobs with photos, not paragraphs

Accept, update, close

A guided flow built for someone standing next to a vehicle.

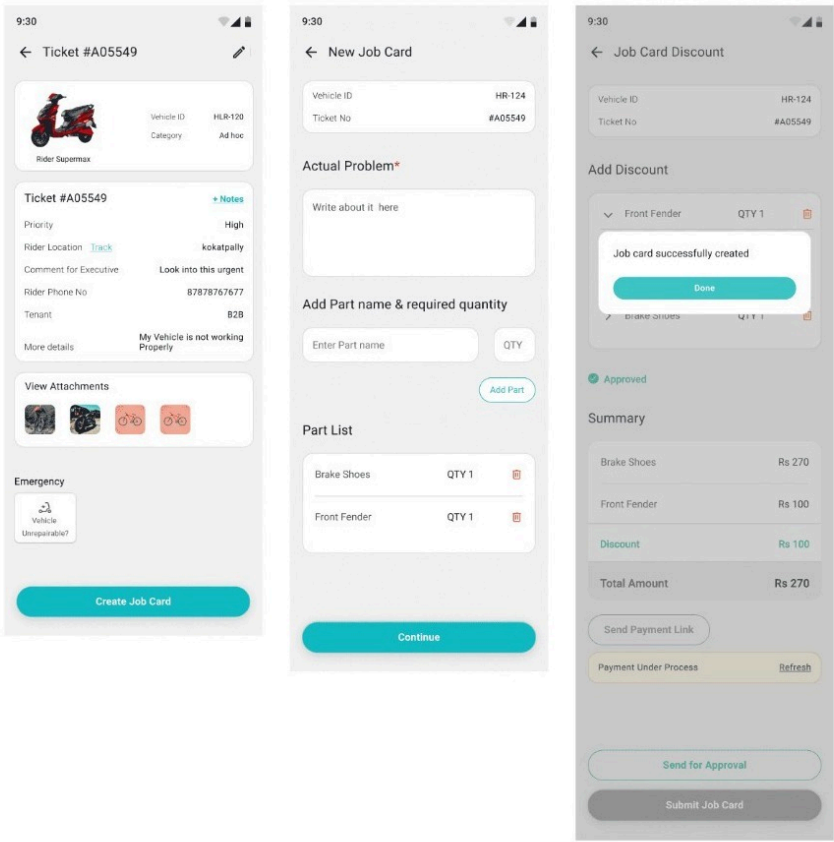
Photo proof of every fix

Documentation actually happens in the field.

Minimal typing

The fix is recorded without writing a report.

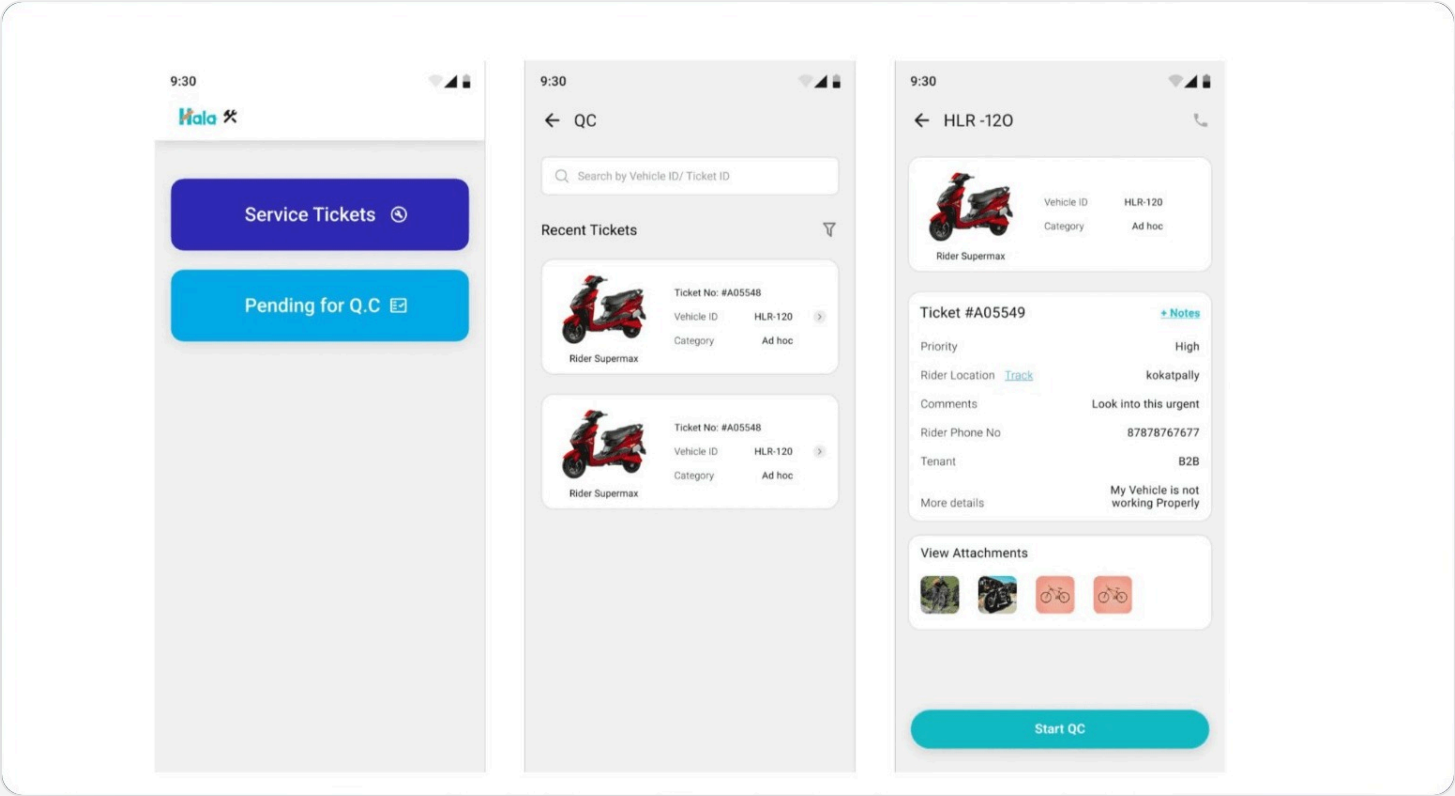
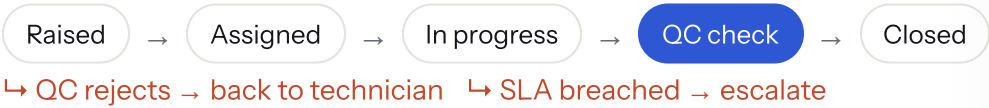
Trade-off: photo uploads lagged on hub connections, one of the engineering constraints we worked through.



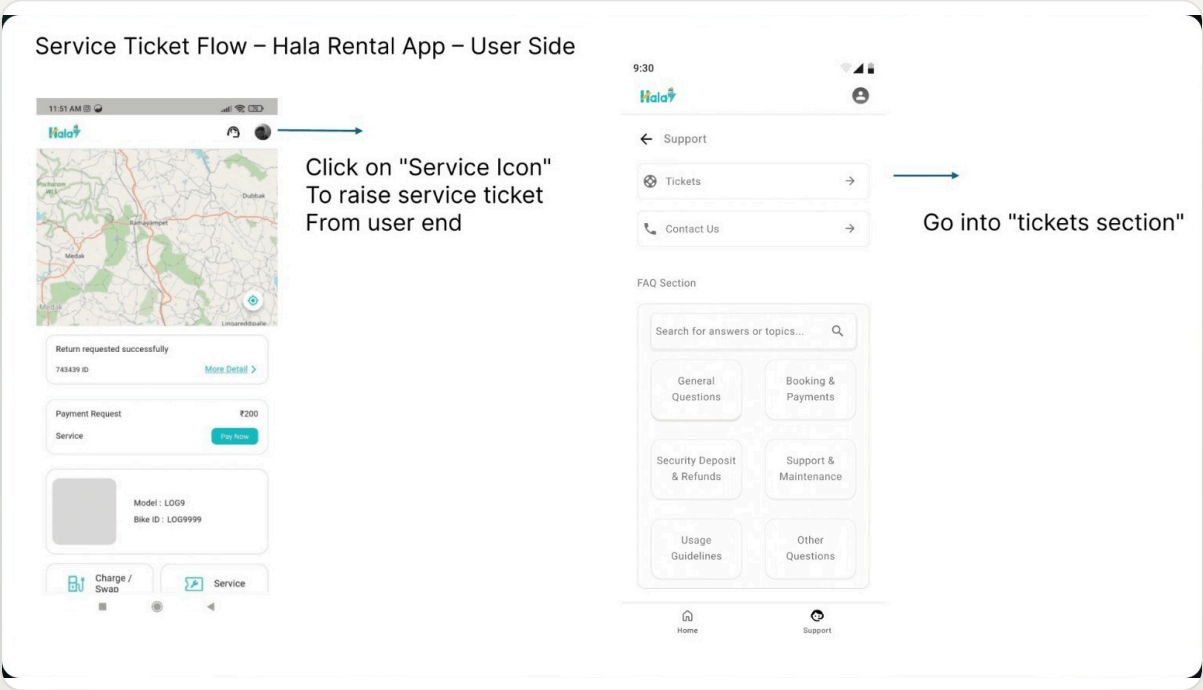
A quality check before a ticket can close

One extra step before closure

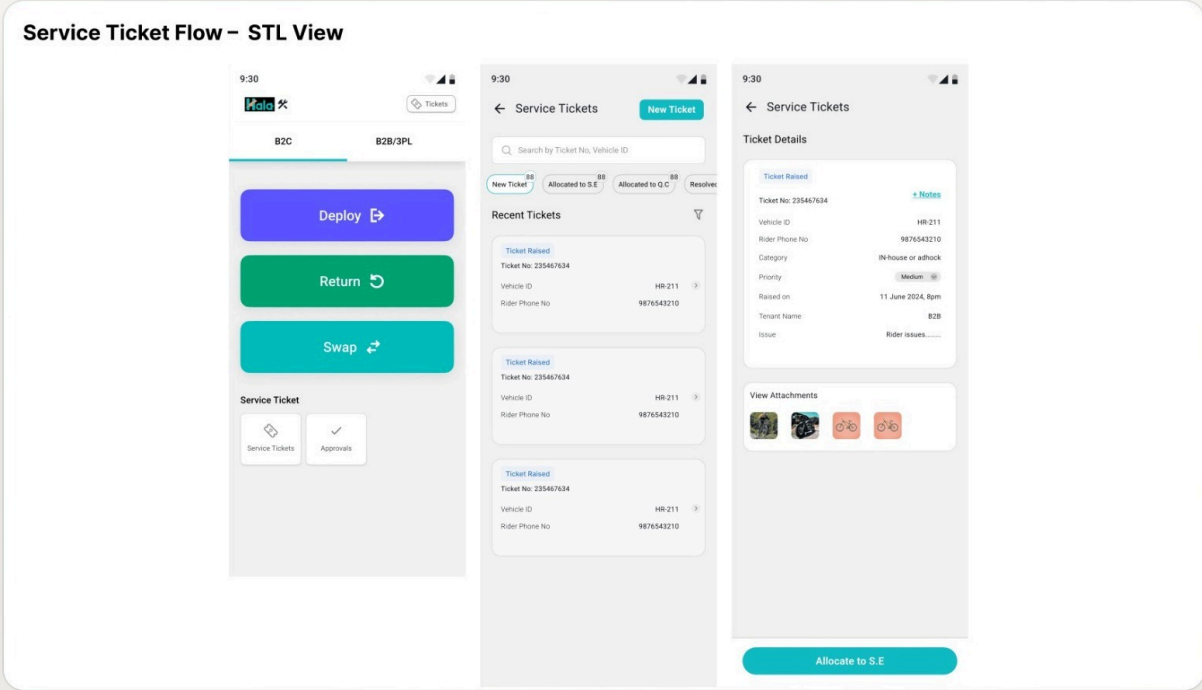
A quality checker verifies the fix before sign-off. A rejected fix goes straight back to the technician instead of the vehicle going back on the road unfixed.



Every role sees its own slice

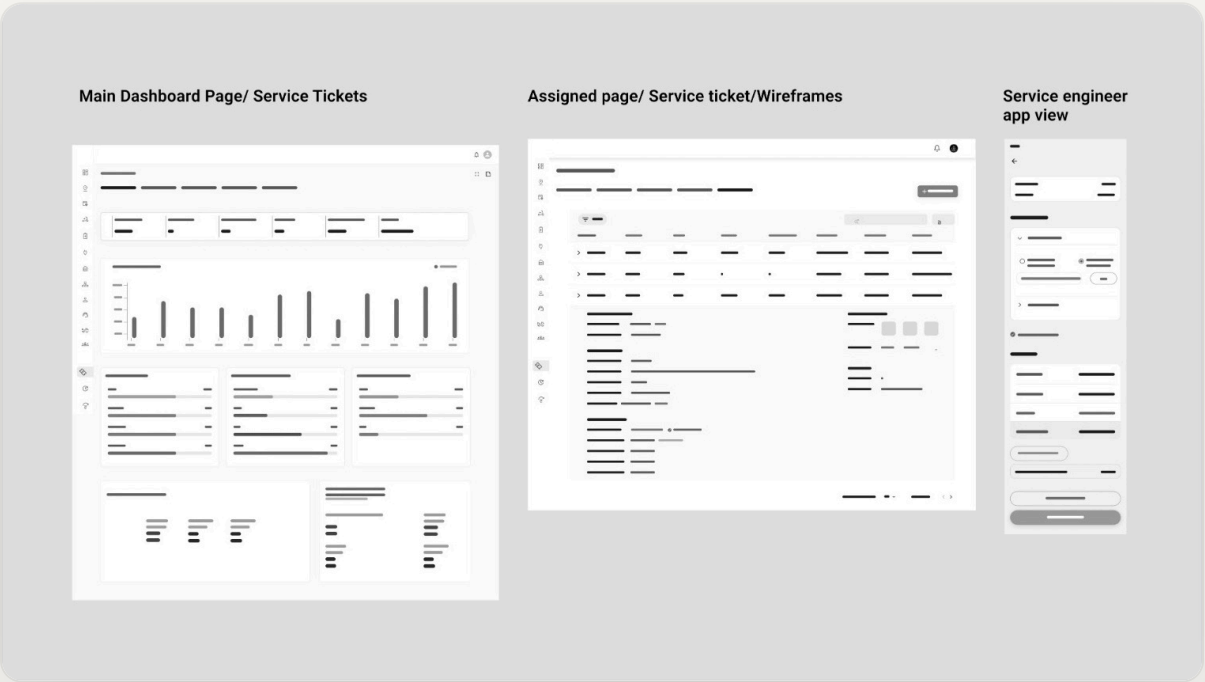


Rider app. Raise a ticket in a few taps and track its status. No more “any update?” calls.

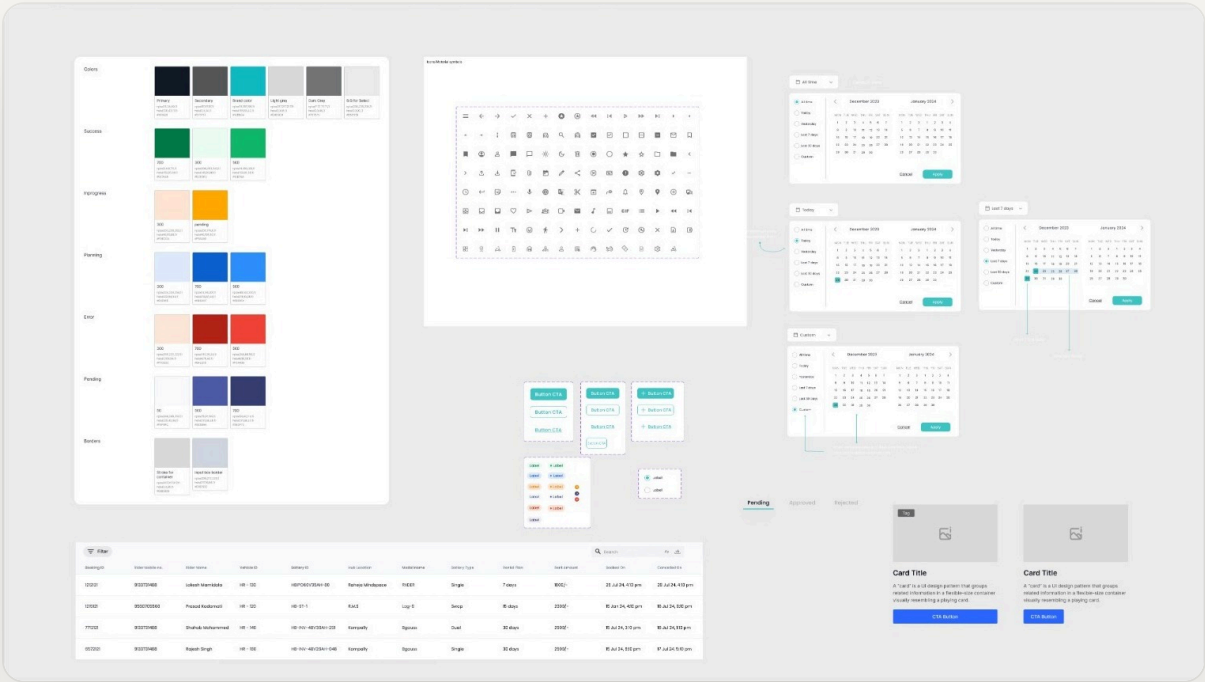


Hub manager view. Volumes, owners and ageing tickets, so they step in before a rider complains.

Tested with the people who'd use it



Low-fidelity flows for fewer steps to create, explicit assignment and field updates without typing.



One shared design system kept the rider app, technician app and dashboards consistent and faster to build.

Shipping it with engineering

Constraints we hit during build, and how the design adapted.

The backend couldn't track a ticket's timeline

→ Every status change stored in a status log, which makes resolution time measurable

Shared components were hard to change

→ Prioritised lifecycle clarity over pixel perfection

Engineering needed to build in parallel

→ Clear acceptance criteria written for every state

WHY IT MATTERED

The status log made resolution time measurable for the first time: **48h on average**.

From WhatsApp chaos to one measurable lifecycle

~60%

fewer rider service complaints

~30%

more jobs closed per technician per day

<5%

of monthly tickets left unresolved

48h

average resolution, Hala’s first reliable baseline

“This saved me at least 1 hour daily chasing executives. Great accountability.”

HUB MANAGER, AFTER LAUNCH

“I finally know who’s doing their job and who’s not.”

OPS HEAD, AFTER LAUNCH

WHAT I’D DO DIFFERENTLY

Instrument before launch

Ship the technician app first

Multi-role design is systems design

MOAZZAM MATTU · PRODUCT DESIGNER

Thank you

Happy to walk through any screen in detail.

EMAIL

moazzammattu1@gmail.com

LINKEDIN

[linkedin.com/in/moazzamm](https://www.linkedin.com/in/moazzammattu)

PORTFOLIO

moazzam.framer.ai