

— FOR FOUNDERS AND COMPANIES PLANNING AN APP

The app brief template.

A workbook to write the brief you would send to any development team:
the problem, the first version, the flows, and who decides.

TIME

Two working sessions

FOR

Founders, product owners, managers

EDITION

October 2026

— HOW TO USE IT

Write the brief before anyone writes code.

App projects rarely go wrong because of code. They go wrong because the brief left the big decisions open: who the product is for, what the first version must do, and what can wait. Each open question becomes a guess, and guesses are where budgets and dates slip.

Fill this workbook in plain words. You do not need technical terms; a good development team will translate. Where you do not know an answer yet, write what you would need to find it out. A clear “not decided yet” is more useful than a confident guess.

Session 1	Sections 1 to 3: the problem, the first version and its users. Write alone, then read it to one future user.
Session 2	Sections 4 to 6: flows, data, platforms, budget and dates. Bring the person who will run the product day to day.
Before you send	Section 7 and the table at the end. Send the same brief to every team you speak to, so their answers can be compared.

01

The problem and who has it

Start with the problem, not the features. A team that understands the problem will suggest better features than any list.

What problem does the product solve, in two or three sentences?

Describe the situation without the app: what people do today and what goes wrong.

Who has this problem? Describe the first group of people precisely.

"Small clinics with one coordinator" is more useful than "businesses".

How do they handle it today?

Spreadsheets, WhatsApp groups, paper, another tool. What do they like and dislike about it?

Why now? What has changed that makes this worth building?

☐ I have spoken to several people who have this problem, and written down what they said.

Their own words are the best material you can give a team.

☐ I know who pays: the user, their employer, or someone else.

When the user and the payer differ, the product often needs two sets of screens.

02

The one job of the first version

A first version that tries to do everything ships late and teaches you little. Choose the one job it must do well.

Finish this sentence: the first version lets [who] [do what] so that [result].

One sentence. If you need two, you may have two products.

When does a user get value for the first time?

For example: the first booking confirmed, the first invoice sent, the first report read.

What must the first version prove, to you or to others?

That people sign up, that they come back, that a business will pay, that a team will stop using the spreadsheet.

☐ The one job fits in a sentence that someone outside the project understands.

☐ I can name the features I am ready to leave out of the first version.

Section 4 asks for them in writing.

03

Users and roles

Most products have more than one kind of user. Each role sees different screens and has different rights.

List each type of user and what they come to do.

For example: the customer books and pays; staff manage bookings; the owner reads reports.

Who can see or change what? Note anything a role must never see.

Prices, other clients' data, internal notes.

Who creates accounts: users themselves, an admin, or by invitation?

How many users do you expect in the first months, and later?

An order of magnitude is enough: tens, hundreds, thousands.

☐ Every role on my list appears in at least one flow in section 4.

04

Core flows, screen by screen

A flow is one thing a user does from start to finish. Describe five to eight of them, step by step, as the user would see them. Rough sketches on paper help.

Flow 1, the main one. What does the user do, screen by screen?

Start where they arrive and end where they get what they came for. Example: open link → choose service → pick a time → enter details → confirmation email.

Flow 2

Who does it, and what happens on each screen.

Flow 3

Flow 4

Flow 5

Flows 6 to 8, if you need them

Often: sign-up and password reset, settings, notifications, the admin side of the main flow.

What is left out of the first version? List it.

Written exclusions protect the dates and the budget as much as the features do.

☐ Each flow says what happens when something goes wrong: a failed payment, a missing field, a cancellation.

☐ I have examples of the real documents, messages or screens the product will replace.
Forms, spreadsheets, PDFs, screenshots. Teams estimate far better from real material.

05

Data, integrations, accounts and admin

These parts are invisible in sketches, yet they are often a large share of the work. Name them early.

What information does the product store? List the main things.

For example: customers, bookings, products, invoices, files. Mark what is sensitive.

Is there existing data to bring in? Where is it, and in what form?

A spreadsheet, an old database, an export from another tool.

Which other tools must it connect to?

Payments, email, WhatsApp, calendars, accounting, maps, an existing system.

How do users sign in, and do they pay inside the product?

Email and password, Google, phone number; one-off payment, subscription or invoice; which currencies.

What does your team need behind the scenes?

An admin area to manage users and content, refunds, exports, reports.

Which languages must the product work in? Any right-to-left language, such as Arabic?

☐ I know which data is personal or sensitive, and who is allowed to see it.

☐ I have checked whether rules in my sector or country apply to the data I plan to store.

Ask your lawyer or accountant if unsure; it changes how the product is built.

06

Platforms, budget and dates

A team cannot plan well without these. The figures are yours to set; ranges are fine.

Where must it run first: web, iOS, Android? Why?

Many first versions start on the web or as one cross-platform app. Say which users need which device.

Your budget range for the first version, and for the year after launch.

Include hosting, support and improvements after launch, not only the build.

Any fixed dates? What happens if they move?

A launch event, a funding round, a season, a contract.

Who runs the product after launch: hosting, updates, help for users?

☐ I have separated the dates that are truly fixed from the ones I would like.

☐ I have kept part of the budget for changes once real users arrive.

The first weeks of real use always teach you something.

07

Success and decisions

Agree how you will judge the first version, and who says yes, before work starts.

How will you know the first version works? Name two or three measures.

Sign-ups, completed bookings, weekly active users, hours saved, spreadsheets retired.

Who decides on scope, design and acceptance? Who is only consulted?

One final decision maker keeps a project moving.

How quickly can you review work and answer questions during the project?

A team waiting a week for an approval loses a week.

What must you own at the end?

Code, design files, domain, hosting and app store accounts, in your company's name.



Everyone who decides has read this brief.

When you send the brief

Hello, please find attached our brief for [product]. The first version should let [who] [do what]. We would like to start around [date], with a budget range of [range]. Could you tell us what you would build first and what you would leave out, how you would set the price of the full build, how we approve each stage, and who owns the code and accounts at the end? Thank you, [name]

Your three must-have flows

FLOW	WHO USES IT	DONE WHEN
1		
2		
3		



THE NEXT STEP

Turn the brief into a **working first version.**

Our Discovery Sprint takes a brief like this one and, in two to three weeks, turns it into a working MVP of the core features, on real infrastructure. It starts with workshops on goals, users and flows, and ends with a build plan and a fixed quote for the full product. If you continue with us, the sprint fee is credited to the build.

- Workshops on goals, users and the one job of version one
- User flows screen by screen, and a written list of what is left out
- A working MVP of the core features you can click through
- A fixed quote for the full build; on payment, the code and accounts are yours

[See the Discovery Sprint](#)

qafza.dev/en/mvp



Qafza designs and builds SaaS products, client portals, CRMs and custom platforms on web and mobile, starting from a working first version.