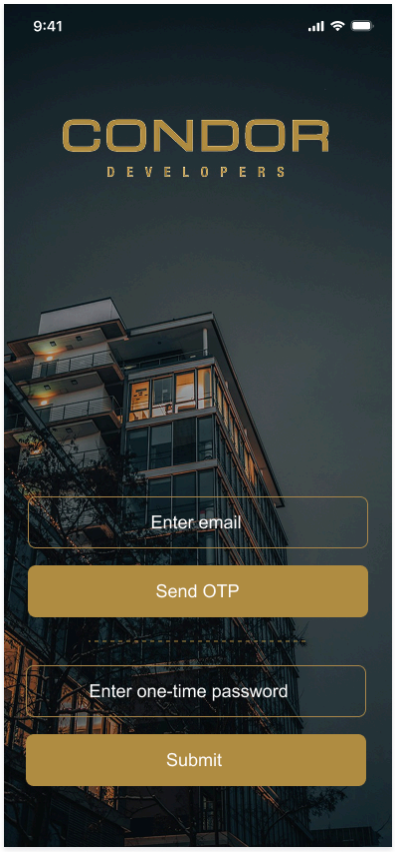
UX CASE STUDY

EMI Payment App Design for
Condor Builders' Property Owners



Project — Condor App

Role — UX Consultant

Overview

Condor Builders is a property developer whose customers pay for their units through EMI (equated monthly installment) plans over time. As that customer base grew, so did the volume of invoices, receipts, and day-to-day questions each buyer had about their account — with no dedicated digital channel to handle any of it.

I was brought on as UX Consultant to design a mobile app that gives every property owner a personalized, self-service home for their billing information: a place to check invoice status, download receipts, request documents, and reach support — without having to call or email the Condor team for routine things.

Role	UX Consultant
Client	Condor Builders
Platform	Mobile app (iOS / Android)
Scope	End-to-end screen design from client brief — IA, flows, UI

The Problem

Before this app, Condor's property owners had no self-service way to keep track of their EMI payments. Checking whether an invoice was pending or paid, getting a copy of a receipt, or raising a service request all depended on reaching someone at Condor directly — by phone, email, or in person.

For customers, that meant:

- No quick way to check what was owed, what had been paid, or when.
- Having to chase the Condor team by phone or email for a document as simple as a receipt.
- No single place to raise a service request or support query and track it.
- Low visibility into their own financial history with a significant, long-term purchase.

For Condor, this created a steady stream of repetitive, low-value support requests that a well-designed self-service tool could absorb — while also giving buyers more confidence and transparency in a relationship that often spans years of installment payments.

Goals

The client brief was intentionally simple: give each customer a personal, secure dashboard to manage their invoices and receipts. Reading between the lines of that brief, the design needed to:

- Make payment status (pending vs. paid) understandable at a glance.
- Let customers retrieve invoices, receipts, and contract documents on their own, in seconds.

- Provide a clear channel for service requests and support that doesn't rely on a phone call.
- Feel secure and personal, given the sensitivity of financial and contract data.
- Stay strictly view-only — the brief was explicit that no payments would be processed in-app, so the design had to make that boundary clear rather than implying a payment flow that didn't exist.

Process & Approach

Working from Condor's brief rather than a full discovery phase, I focused the process on translating a short set of business requirements into a clear information architecture and then into screens — treating the brief itself as the primary input and using reasonable UX judgment to fill the gaps it left open.

1. Structuring the information

The brief described a handful of distinct jobs — invoices, receipts, documents, service requests, support, FAQs — that could easily have sprawled into a cluttered app. I grouped them into two navigation layers instead of one:

- A persistent bottom tab bar for the four things a customer returns to constantly: Home, Invoice, Payment (the app's icon/hub), Receipts, and Downloads.
- A secondary top utility row — Service Request, Request Documents, Support, FAQ — for less frequent, task-based actions, kept consistent across every screen so it never has to be re-learned.

2. Designing around a boundary, not a feature

The most important constraint in the brief was a negative one: no payments in-app. Rather than leaving a payment button that does nothing, I designed the Invoice and Receipts screens purely around status and retrieval — a segmented All / Pending / Paid control, a preview (eye) icon, and a download icon — so the interface never implies an action the app can't perform.

3. Designing for a recurring, long-term relationship

Because EMI payments happen over months or years, I designed the dashboard to open on identity and status first: a profile card up top, then quick-access utility icons, then the customer's own projects and any pre-launch offers — reinforcing that this is a personal account, not a generic brochure app.

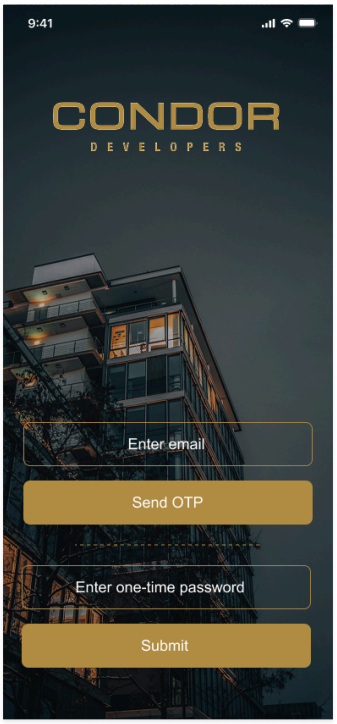
4. Keeping trust and security visible

Login uses email + one-time password rather than a stored password, which fits a low-frequency, high-trust use case where customers dip in briefly rather than staying logged in for long sessions. The Condor brand's dark, architectural login screen sets a tone of security and permanence before the customer ever reaches their data.

Key Screens

Login

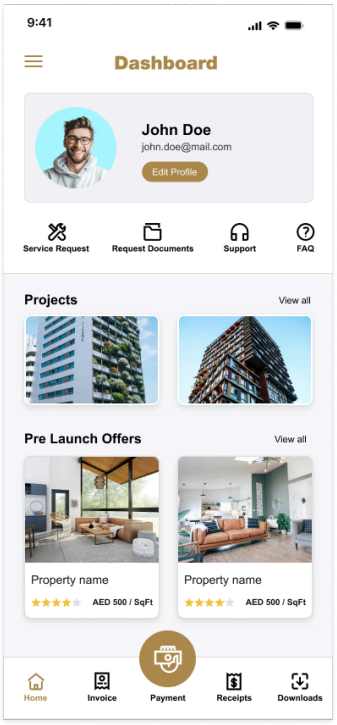
A full-bleed image of a Condor building anchors the brand before login, with a two-step email + OTP flow that keeps access secure without demanding a remembered password for an app used only occasionally.



Login — email + OTP

Dashboard

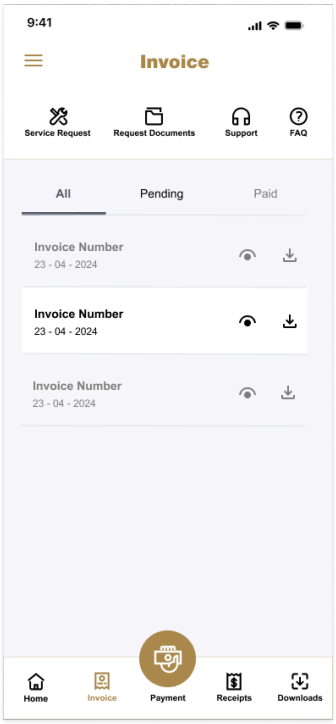
The home screen leads with the customer's identity (profile photo, name, edit-profile action), followed by the four most common actions as icon shortcuts, then the customer's own projects and current pre-launch offers — personal account information before promotional content.

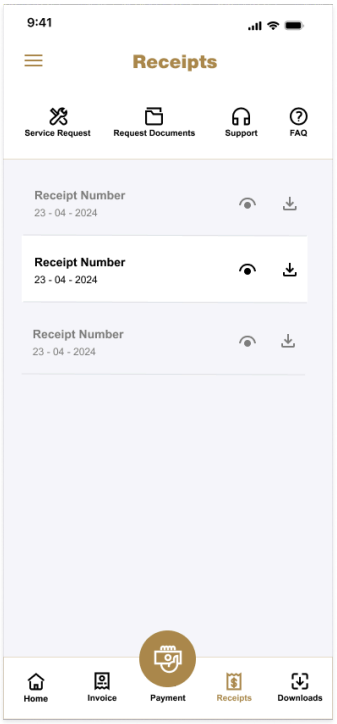


Dashboard — personalized home

Invoices & Receipts

Both screens share the same pattern: a segmented control to filter by status, and per-row preview and download actions. Repeating the pattern across Invoices and Receipts means a customer only has to learn it once.



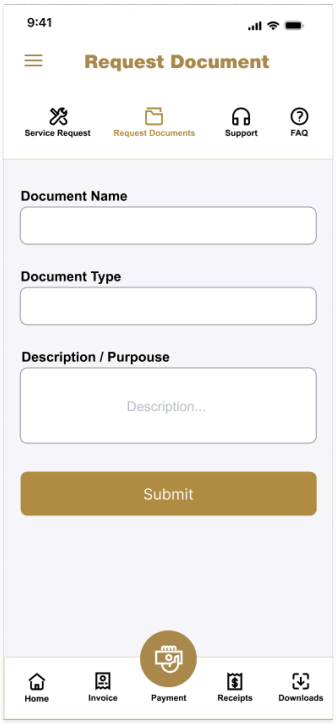


Invoice — status tabs + preview/download

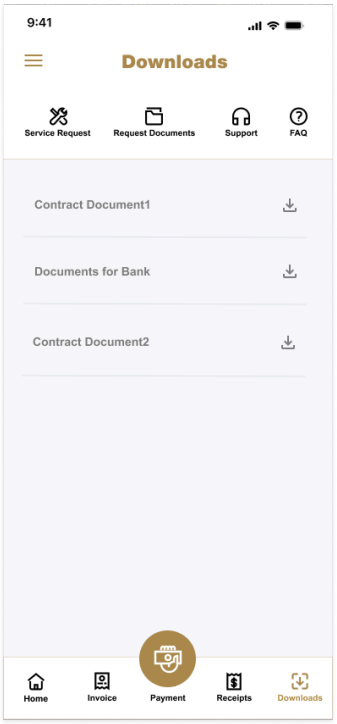
Receipts — same pattern, reused

Documents & Downloads

Request Documents gives customers a simple form to ask for a specific document, while Downloads is where anything Condor has issued — contracts, bank documents — lives for retrieval, closing the loop on that request.



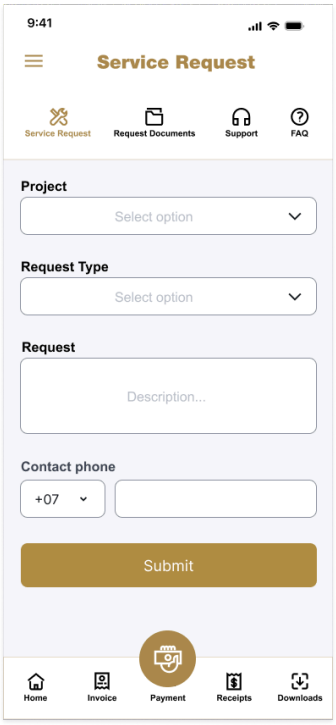
Request Documents — simple request form



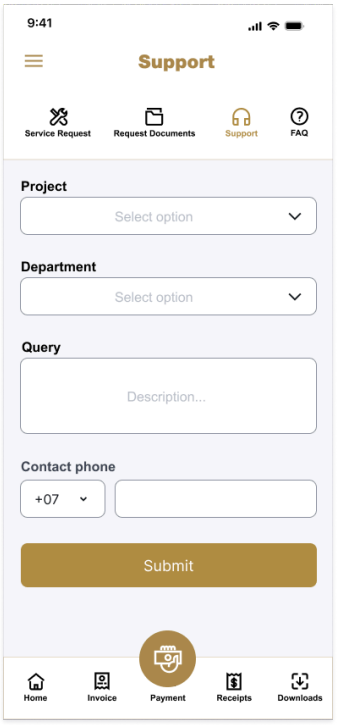
Downloads — issued documents

Service Request & Support

Both forms follow the same structure — project, category, free-text description, contact number — so a customer filling in a service request already knows how to fill in a support query. Splitting them into two flows (rather than one combined 'contact us') lets each be routed to the right team internally.



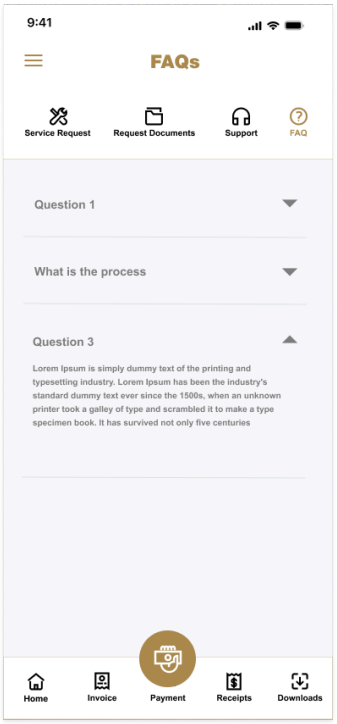
Service Request



Support

FAQs

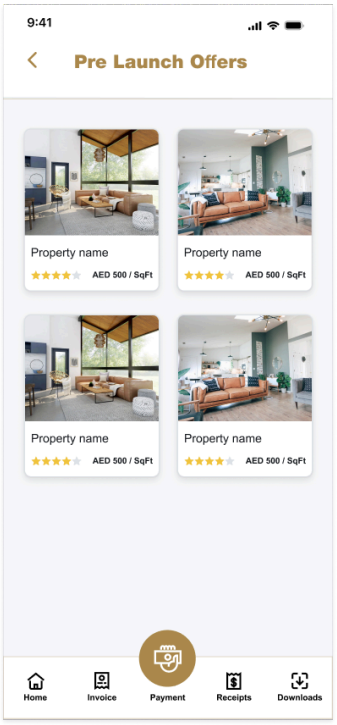
An accordion pattern keeps the FAQ screen compact and scannable, letting customers expand only the question they care about instead of scrolling a wall of text.



FAQs — accordion pattern

Pre-Launch Offers

A secondary, browsable grid for upcoming properties — visually distinct from the transactional screens, since this is discovery content rather than account management.



Pre-Launch Offers

Reflection & Learnings

This project was a good exercise in designing from a lean brief rather than a full research engagement — a common real-world constraint, especially in consulting work. A few things I'd carry forward:

- A short brief still has an implicit user in it. Reading 'view invoices and receipts' as 'a customer who wants reassurance about a long-term financial commitment' shaped decisions — like leading the dashboard with identity — that a literal feature list wouldn't have suggested.
- Consistency across similar flows (Invoice/Receipts, Service Request/Support) does a lot of the UX work when there isn't time for user testing to catch inconsistencies later.
- Designing what an app deliberately doesn't do — like payments — is as much a UX decision as designing what it does.

With more time or a follow-up phase, the natural next steps would be validating the IA with real customers (particularly whether Service Request vs. Support is an intuitive split), and instrumenting the app post-launch to see which self-service actions actually reduce inbound calls to Condor's team.