02 · B2B SAAS · PROPERTY OPERATIONS

Coordinating property maintenance across approvals and handoffs

Helping property managers identify the next step, carry the right context into approvals and coordinate work across tenants, owners and trade providers.

At JRC Software, I led UX design across four modules of Bricks+Agents, a responsive property-maintenance platform. The central design challenge was to make the next step clear as a job's status, approval requirements and responsible person changed.

ROLE

UX Designer

PRODUCT

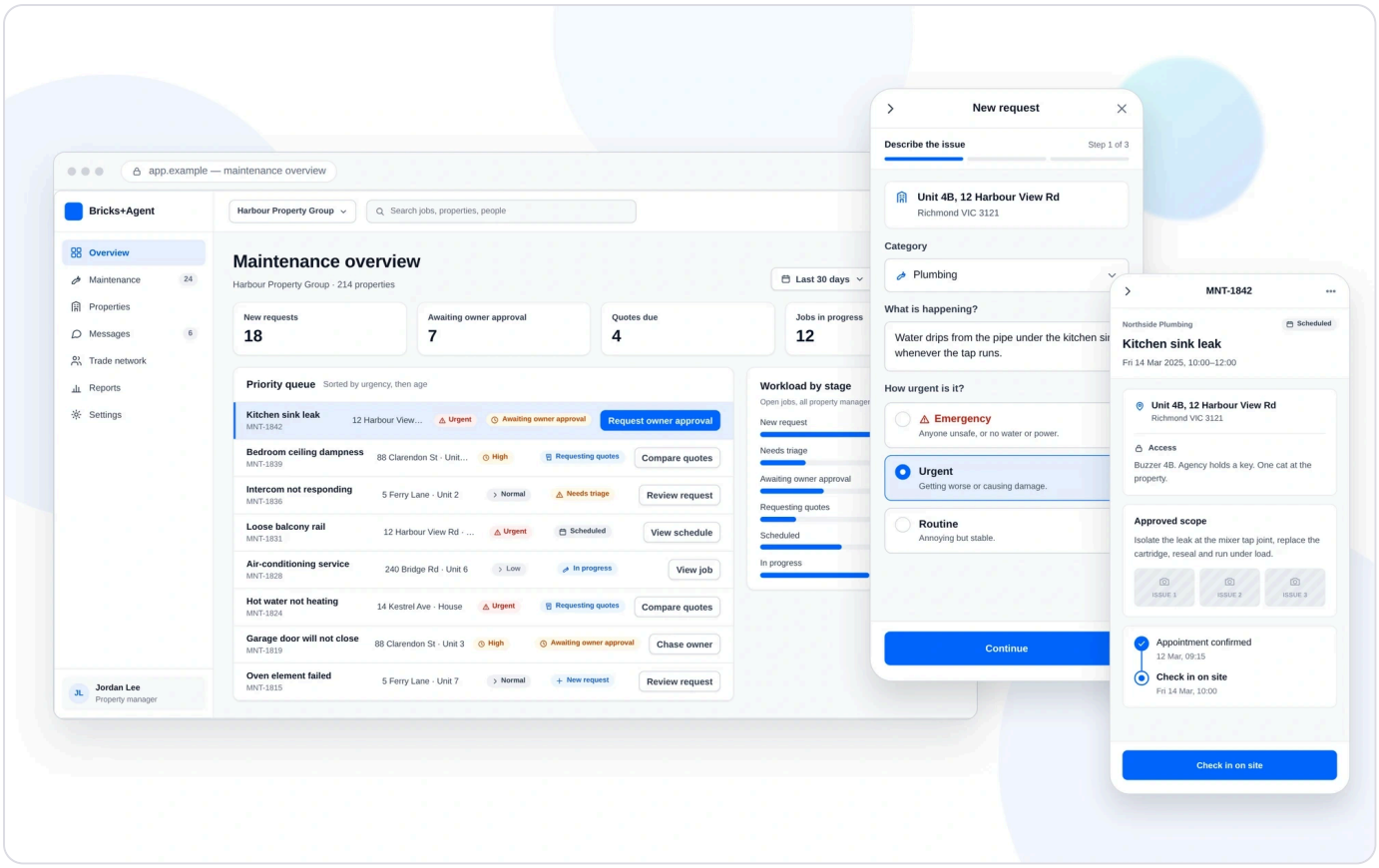
Bricks+Agents
at JRC Software

TIMELINE

Aug 2019 — Feb 2021

PLATFORM

Responsive web & mobile



The platform connects office coordination with reporting and work away from a desk. The interfaces show different parts of the same maintenance process.

4
Product modules

70+
Responsive screens

8
Concepts evaluated

15+
Features supported through delivery

---

01 — PROBLEM AND CONTRIBUTION

A job status does not explain what can happen next

An urgent request may still need owner approval. An approved budget may not cover the selected quote. A provider may have the right skills but different availability from the tenant. A maintenance job can therefore be understandable without being ready to progress.

The practical problem was helping a property manager decide what to do next without losing the conditions behind that decision: who is responsible, what work is proposed, what has been approved and what the other roles need to know.

I organised the experience around one connected job record, with stage-specific actions and views appropriate to each role. The design goal was to make responsibility and approval conditions visible

at the point where someone acts.

My contribution and how the work developed

I independently led UX design across four modules. I mapped workflows, organised the information architecture, created responsive wireframes and interactive prototypes, and developed approved directions into final UI. Business stakeholders helped clarify requirements, and developers reviewed feasibility and implementation.

Eight concepts were evaluated before development through workflow reviews, wireframes and prototypes. I stayed involved during delivery, answering design questions and reviewing layouts and interactions.

Four roles with different decisions

Tenant

Report the issue, explain access requirements and follow progress.

Property manager

Identify the next step and coordinate approvals, providers and work.

Property owner

Understand the proposed scope and cost before approving it.

Trade provider

Review the scope, quote, schedule and document the completed work.

The job information stays connected, while the detail and controls change with the responsibility of the person using it.

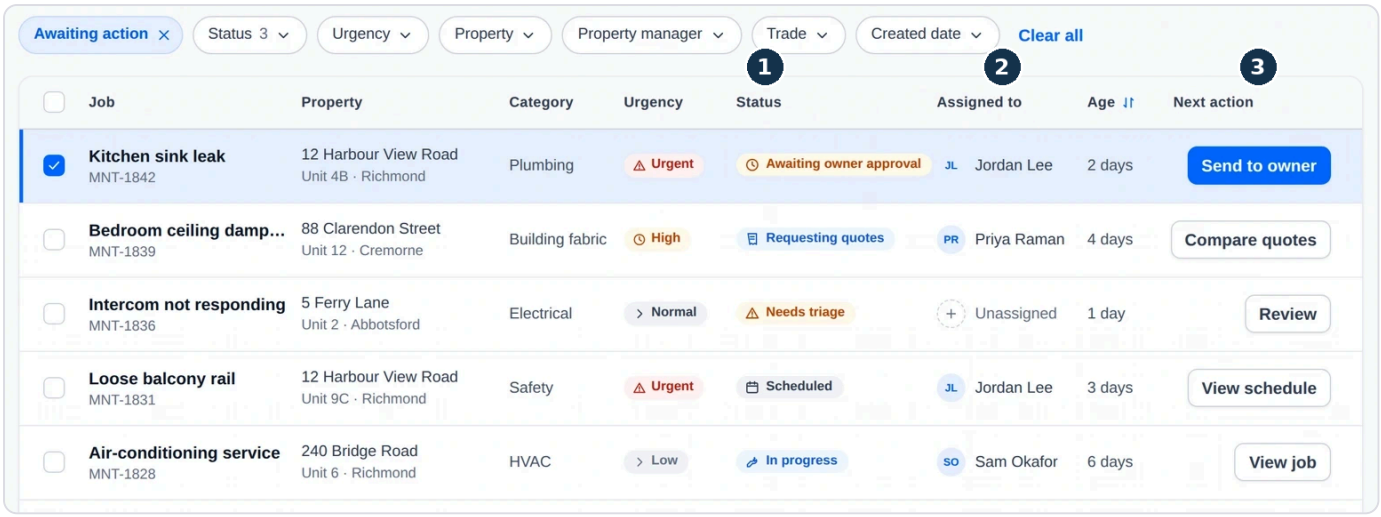
02 — MAINTENANCE QUEUE

Make the next action visible in the queue

A queue organised around status alone would still leave the manager interpreting what each label means for the next task. I placed urgency, status, assignment and the next action alongside each request.

The action changes with the stage: review an incoming request, compare quotes, follow up on approval or request an invoice. Filters narrow the workload before the manager opens a record.

Maintenance queue FIG. 02



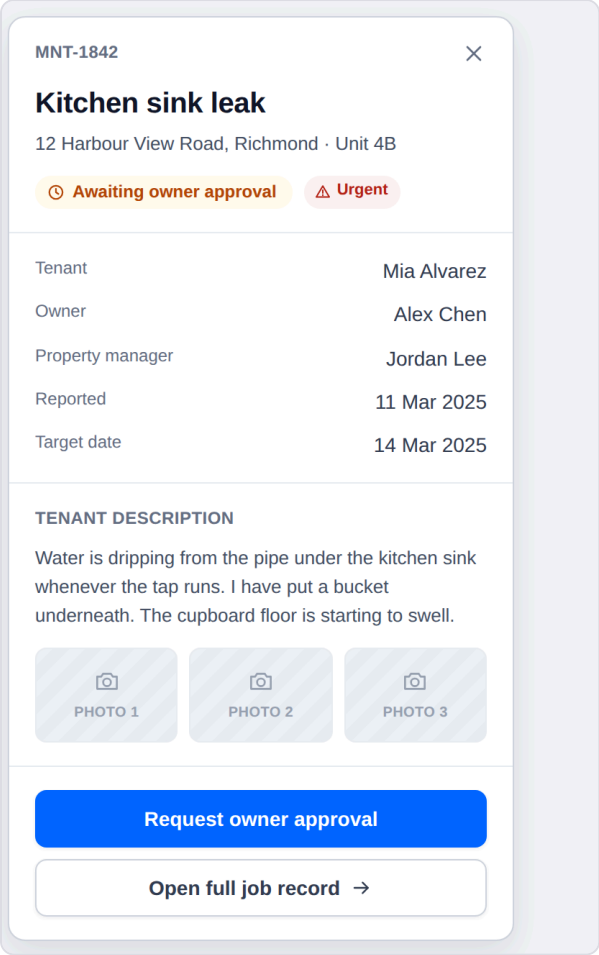
1 Current state. 2 Assigned person. 3 Next action. These appear together so the manager can connect the condition of a job to responsibility and the available task.

Keep the list compact and supporting detail accessible

The queue keeps a compact set of fields visible. A row preview adds the request description and people involved, and the full record carries the history and supporting documents. This uses progressive disclosure to preserve list context while providing more detail when a decision needs it.

Request preview

FIG. 03



The preview supports a quick review of one request while preserving access to the full job record.

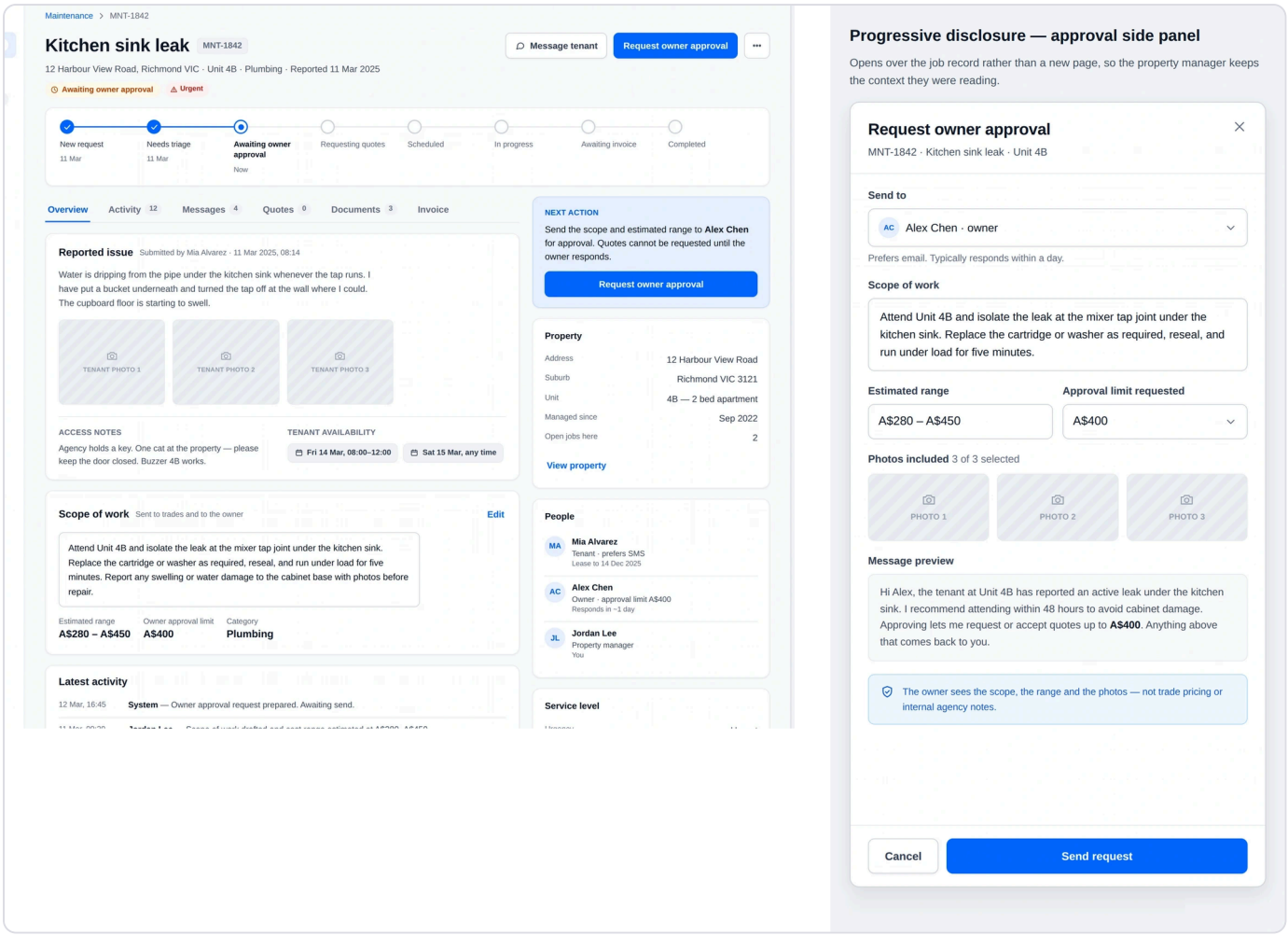
03 — OWNER APPROVAL

Keep approval scope and spending limits together

Approval determines which work can proceed. The manager needs to communicate the proposed work and requested spending limit before the owner can make a decision.

I kept scope, estimated cost range, photographs and approval limit together in the request panel. The job record retains the request and activity history as the manager prepares that message.

Job record and approval request FIG. 04



The record supplies context; the request panel focuses on the information needed for approval. The owner receives the relevant scope, range and photographs, while internal agency notes remain separate.

Share the job record while tailoring each role’s view

Each role needs context from the same job, but does not need every operational control. The owner’s decision context is narrower than the manager’s workspace. Keeping those views distinct balances continuity with information relevance.

---

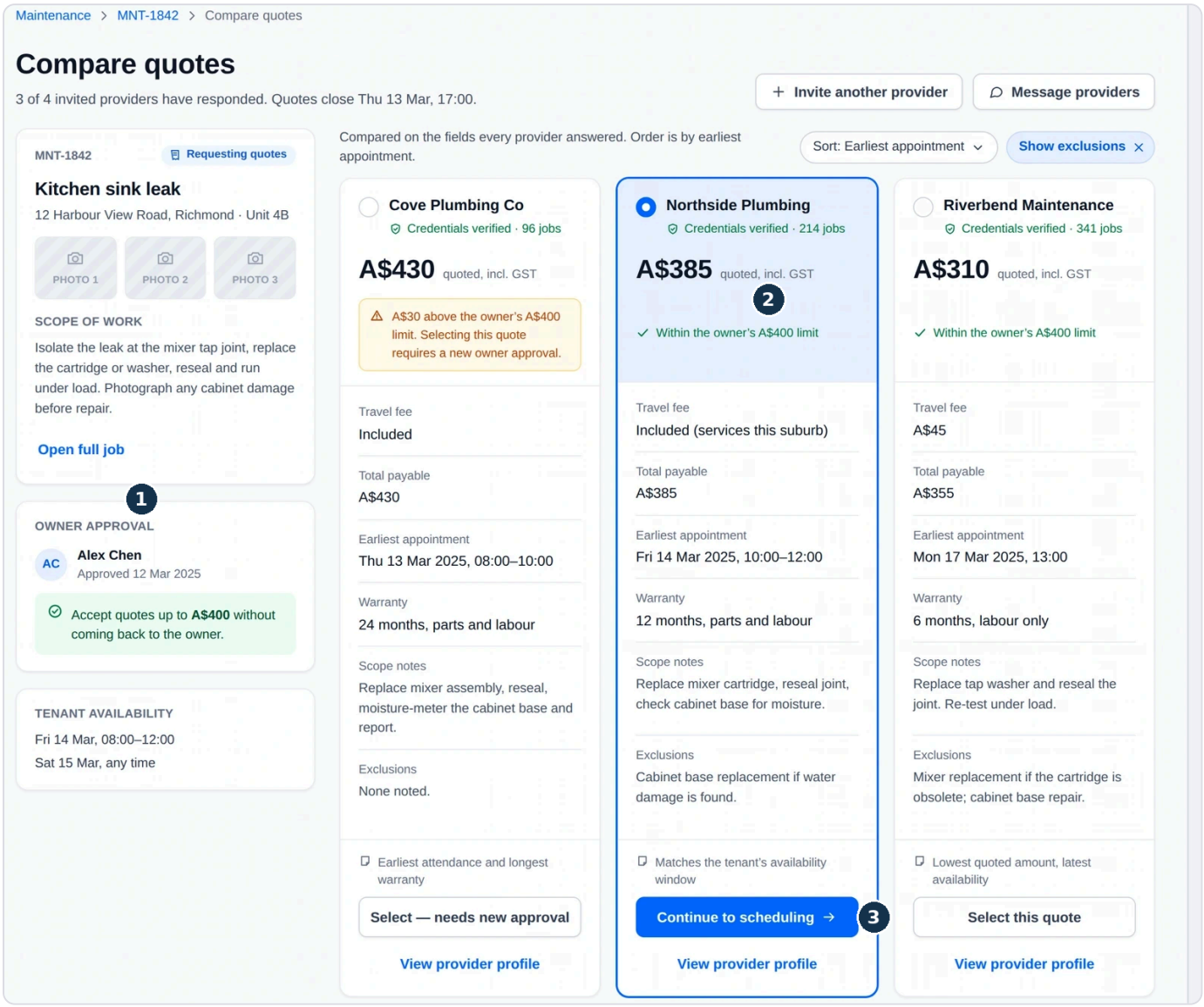
04 — QUOTE COMPARISON

A quote can change what has already been approved

The illustrated workflow requests budget approval before gathering quotes. It then checks proposals against that limit. A quote above the limit requires another owner approval; a quote within the limit can continue towards scheduling.

I kept the request scope, approved limit and tenant availability alongside the comparison. Consistent fields expose total payable, travel fees, appointment timing, warranty and exclusions.

Quote comparison FIG. 05



1 Approved spending limit. 2 Selected proposal and its approval status. 3 The action that follows the comparison. The job's conditions remain visible while the manager reviews options.

In this screen example, the A$400 limit makes the A$430 proposal require renewed approval. The A$310 headline quote includes a separate A$45 travel fee, giving a total of A$355. The selected A$385 proposal also shows a match with the tenant's availability. These are interface examples, not measured project results.

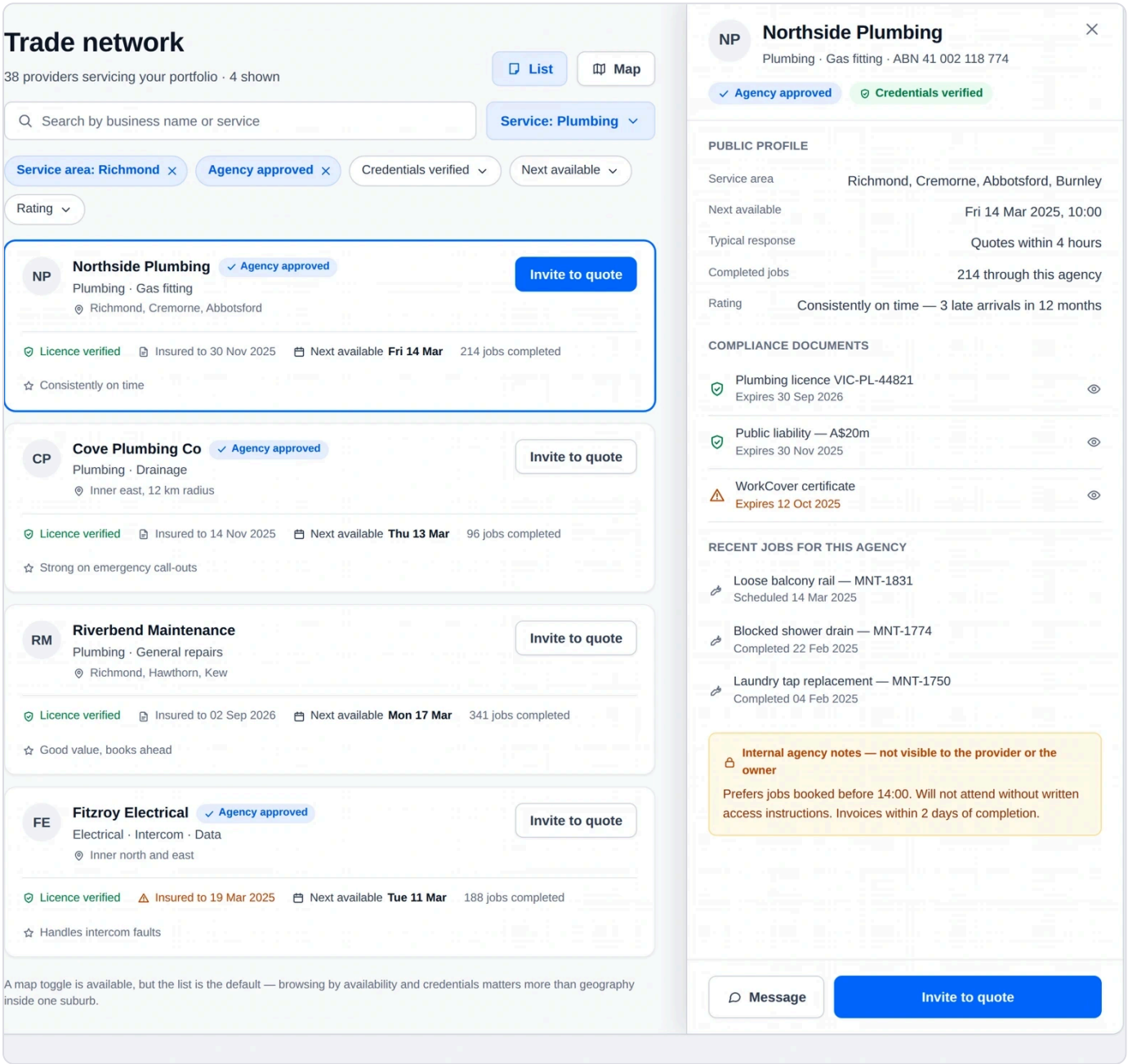
05 — PROVIDER SELECTION

Review provider suitability before inviting a quote

The provider choice adds another set of conditions: service area, availability, credentials and the scope of work. I brought those details into a filtered directory with a profile panel for further review before an invitation to quote.

The list supports shortlisting. The panel adds compliance documents and supporting information. Public provider details and internal agency notes remain distinguishable.

Provider selection FIG. 06



The list keeps comparable provider information visible. Further detail appears in the profile panel, allowing the manager to review a provider before inviting a quote.

Separate shortlisting from detailed provider review

Showing every document and note in the list would compete with the information needed to shortlist providers. The panel gives those details a place without turning each list item into a complete profile.

---

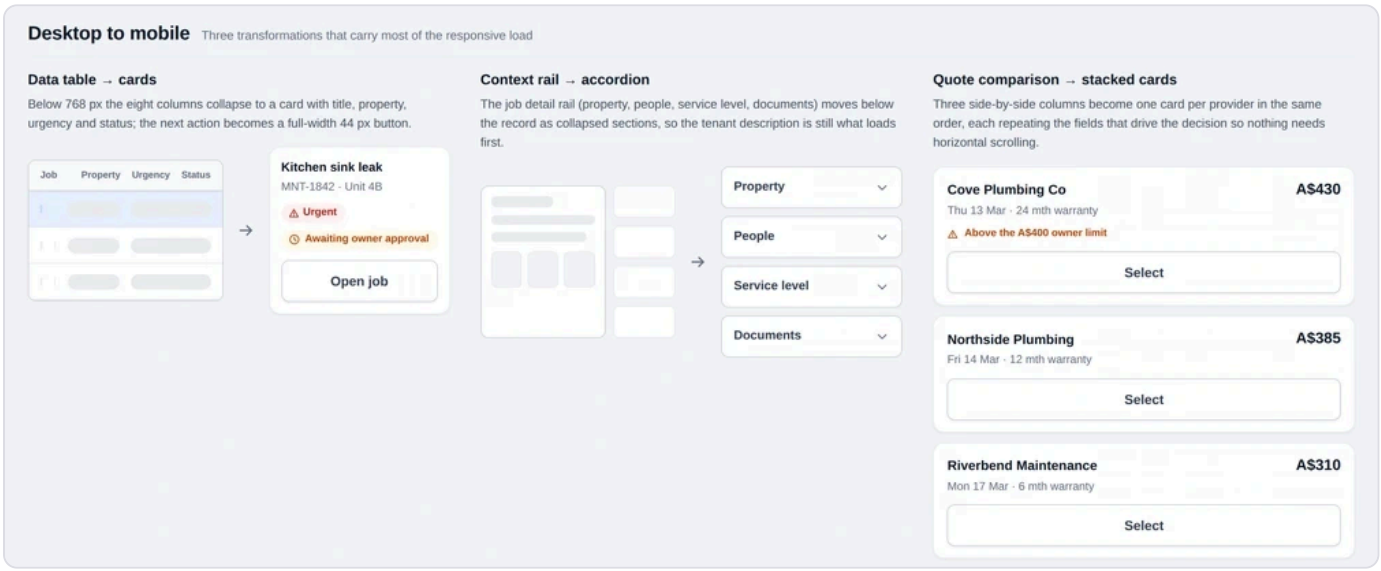
06 — MOBILE AND STATES

The same decisions needed to work on mobile

Responsive design had to preserve the relationship between the job's condition and its next action. I prioritised the current job and task for people working away from a desk.

The component guidance shows how queue rows become cards, supporting job information becomes collapsible sections, and quote columns become stacked cards. These changes preserve decision context while adapting the layout to the screen.

Responsive transformations FIG. 07

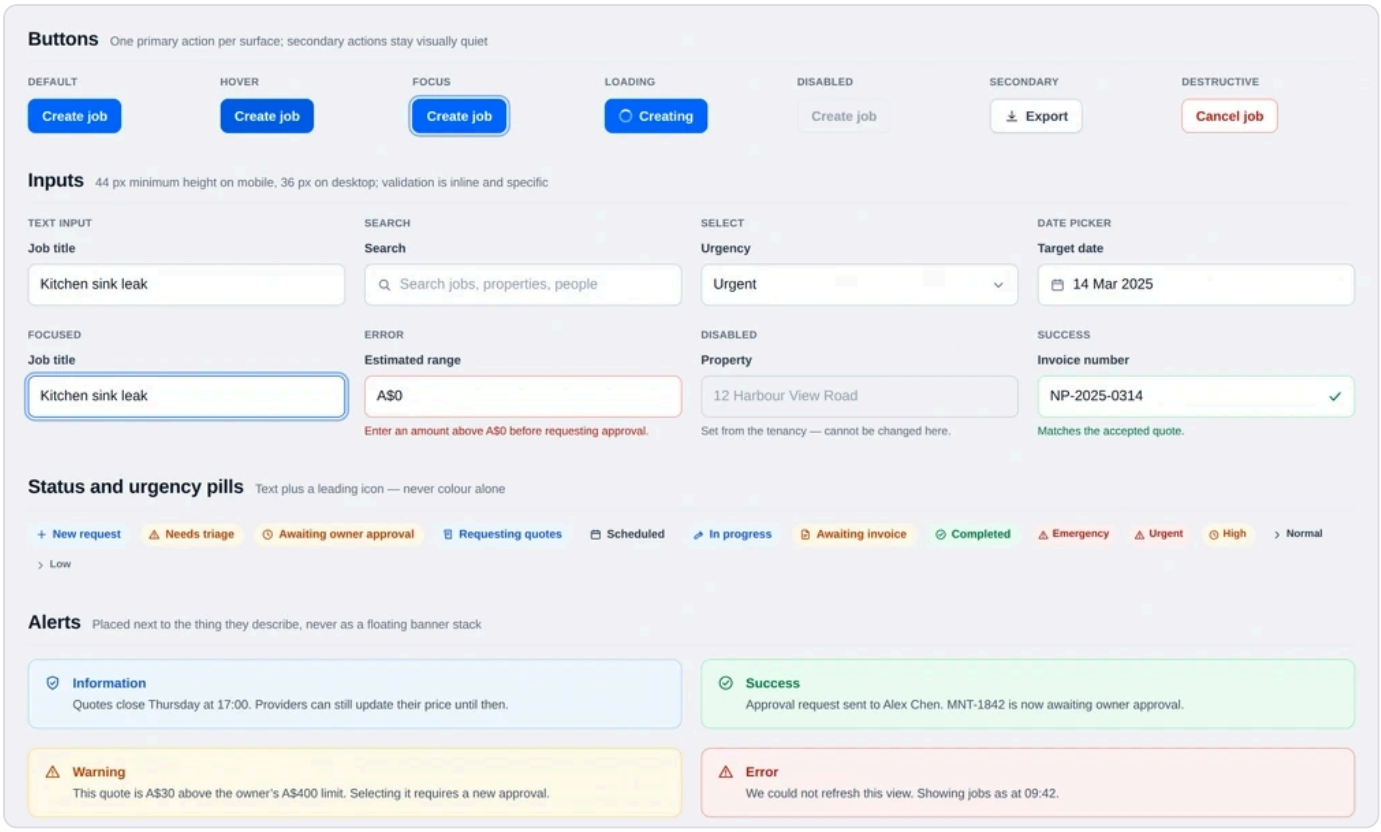


The mobile hierarchy keeps job status and actions accessible, while supporting information becomes available on demand.

Patterns also needed to cover interrupted work

I created reusable patterns for forms, status labels, actions and job information, including focus, loading, disabled, validation and error states. The component sheet includes inline validation, approval warnings and recovery messages alongside the standard controls.

Component states FIG. 08



These states define how the interface communicates incomplete input, unavailable actions and approval conditions beyond the ideal path.

---

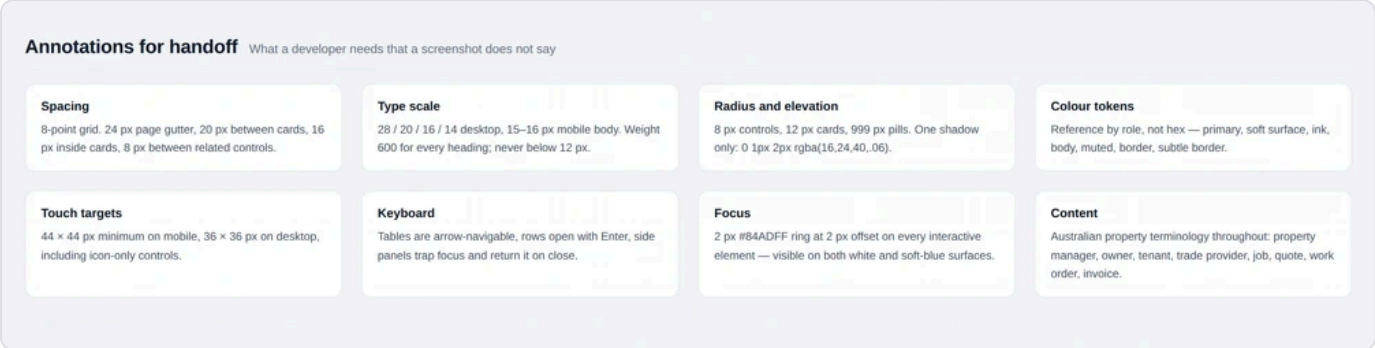
07 — DELIVERY AND REFLECTION

Reviewing the workflow through implementation

Workflow reviews, wireframes and interactive prototypes were used to evaluate eight concepts before development. The reviews helped identify unclear actions, missing states and responsive issues while the directions could still change.

I stayed involved during implementation to answer design questions and review the layouts and interactions. Reusable patterns gave the team a shared reference for behaviour across the modules.

Developer handoff FIG. 09



The handoff guidance makes interaction rules explicit, including focus treatment, keyboard behaviour, touch targets and consistent property-maintenance terminology.

What the work delivered

The delivery covered four modules and 70+ responsive screens, with 15+ features supported through implementation. The designs linked the queue, shared job record, approval requests, quote review and provider selection through a consistent maintenance workflow.

The case demonstrates how the interface represents responsibility, approval conditions and the next action. The available results describe delivery scope; they do not establish changes in completion time, support volume or business performance.

What I would validate next

Task-based evaluation should check whether managers can identify blocked jobs, interpret approval limits and compare total costs without overlooking access requirements. I would track correct next-step decisions, task completion and requests for clarification against the existing workflow.

What I took from the project

A useful operational interface needs to explain what a status allows someone to do. Organising the design around that relationship helped me connect the screens to the decisions behind the work.

I also carried forward a more deliberate approach to responsive design: preserve the information needed for the decision, then adapt how supporting detail is accessed.
