

CONTENTS

About this handbook

1. The app at a glance

2. Dashboard

3. Invoices

4. Clients

5. Compliance

6. Currency

7. Quotations

8. Email (circulars)

9. Reports

10. Timesheet

11. Settings

12. Safeguards, security and backups

Glossary

What's new

Practice App — Handbook

What the software does, module by module

Edition of 15 October 2026

About this handbook

This handbook describes **everything the practice app does**: each module, what it is for, and the rules it follows. It is a reference — look up a module to learn what it can do and why it behaves the way it does.

For step-by-step instructions (“how do I record a payment?”) see the companion **User Manual**.

Both documents are kept up to date with the software. Every time a function is added or changed, the relevant section is updated in the same release, and the change is listed under **What’s new** at the end.

Edition: see *What’s new* at the end for the date of this edition.

One convention worth knowing. A path written with arrows — **Settings** → **Company details** → **Closed up to** — is a path through *this* app, and every name in it is a name on the screen. Where another program’s menus are meant, the separator is a chevron (iPhone: Settings › General › Date & Time), so the two can never be confused. A test reads every arrow path in both documents and fails if any of them names a screen this app does not have.

1. The app at a glance

The app is the practice’s single place for invoicing, clients, compliance deadlines, the books, quotations, client mailings, time recording and reports. It runs in a web browser at **invoicing.kppanagi.com** and keeps its data in one database, so everybody sees the same figures on every device.

1.1 Modules

The top-left is the practice’s own mark, from 13 October 2026: the **KP** monogram in its gold frame, and **K.P. Panagi & Co** beside it. It read *InvoicePro* until then, which was a product name for software with one customer. The mark is drawn as geometry rather than loaded as a picture, so it follows the light/dark theme without a second file, stays sharp at whatever the text-size control is set to, and costs no request. On a narrower window — below about 1360 pixels — the name is given up and the monogram stays, which is the same order the bar has always shed things in: the signed-in person’s name first, then the brand’s words, and never a module name.

The **browser tab** says *Invoicing — K.P. Panagi & Co*, which is deliberately not what the bar says. Inside the app you know which of the three you are in; a tab strip is the other way round, since payroll and accounting are the same firm, so the title’s job is to be the half that differs.

The modules run **across the top** of the screen, name and icon together. A module with sections of its own opens them as a **menu under its name**: click the name and the list drops down, and a line with an arrow at its end has sections of its own, which open in a list beside it. The row beneath the names says which screen you are on, and holds the **Back** control and the ★.

Grouping is used where a module has too many sections for one list, and **Tools** is the only one that does. **Invoices** carries its four — *All documents*, *New invoice*, *New credit note*, *New pro-forma* — flat, from 13 October 2026: they were under a *New* › parent for a day, and a sub-menu in front of the three commonest acts in the module people have open all day is a hover and a sideways move for nothing.

Module	What it is for
Dashboard	Income at a glance, and the work queue: what needs attention today
Invoices	Invoices, credit notes, pro-formas and receipts
Clients	The client records (the CRM): details, portal credentials, statutory registers — and <i>Deadlines</i> , the filing dates the practice keeps for them (practices only)
Tools	<i>Quotations</i> and the fee list they are built from, <i>Email</i> (circulars), and <i>Currency</i> (European Central Bank euro reference rates)
Reports	Fixed reports and the report builder
Timesheet	Time recorded against clients
Settings	Company set-up, deadline rules, defaults, people, backups and your own sign-in

Ten modules would not fit on one bar, so four became sections of another: **Compliance** is now *Clients* → *Deadlines*, and **Quotations**, **Email** and **Currency** are the sections of **Tools**. Nothing was removed — every screen those four had is a line in the menu, at the depth it was before — and an address saved before the change still lands on the same screen and is rewritten to the new form as you arrive.

Search is the magnifying glass on the bar, or **Ctrl+K** (**⌘K**); it opens over the bar and closes when you click away. **Help** is the ? beside it and opens this handbook and the user manual in a new tab. Each Help page has a search box at the top that looks through both documents at once.

On a phone the module names take a line of their own beneath the company switcher and scroll sideways. The menus work the same way, except that a line with sections of its own opens them *inside* the menu, indented

under the line you tapped, because beside it there is no room. There is no burger button: the navigation is the same at every width.

1.1a Favourites

A column down the left holding nothing but shortcuts to screens. The ★ at the right of the row beneath the module names keeps the screen in view; pressing it again takes it off; the × on a row does the same without going there first.

Whose	Each person's own. Staff do not see each other's
Scope	Per company — the list follows the switcher
Order	The order they were added, so nothing moves under the cursor
Where kept	On the server (favourites), not in the browser, so the list follows you between machines
A screen that stops being offered	Not drawn. The row stays, so setting a company back to a practice brings its Deadlines shortcut back
On a phone	A scrolling strip under the module names

What is stored is the address (page , section) and a label; the name on the list is worked out from the module table each time it is drawn, so renaming a screen renames the shortcuts to it. A person given a company to **read but not change** may still keep shortcuts to its screens — a bookmark is a habit, not company data, and both the server's rule and the screen's own check say so.

Both references cascade on delete: removing a company or a person takes their shortcuts with them.

1.2 Companies and the company switcher

The app holds several **companies** — the entities that issue invoices and keep books (K.P. Panagi & Co. Ltd and the others). Almost everything you see belongs to **one company at a time**: the one chosen in the **switcher** at the top right.

- Each company has its own clients, invoices, numbering, services, payment accounts, officers, logo and mail server.
- Deadline **rules** and the shared statutory dates are global — one list for every company (see 5.1).
- A client can be a client of more than one company; each company keeps its own list.
- Switching company changes the whole screen. If you have unsaved work, the app asks first (see 1.5).
- An administrator adds new companies from the bottom of the switcher (+ **New company...**).

1.3 Kinds of company

Two settings on each company decide what the app shows for it:

- **What it invoices for** — *services, goods / products, or both*. A goods company gets a product catalogue and per-line VAT rates; a services company gets a list of services and period wording (“... for March 2026”).
- **What kind of firm it is** — a *regulated practice* (audit, law, corporate services) or a *services or trading company*. A practice sees everything: registration numbers, portal credentials, statutory registers and **Clients** → **Deadlines**. A trading company sees a leaner client record. Changing this changes only what is **shown**, never what is **stored**.

1.4 People and access

There are **five roles**, and they are the same five in all three of the practice’s applications — this one, payroll and accounting — so that a person’s standing is the same word wherever they sign in. Two of them say what somebody is **in the practice**, and three say what they are **in one company**.

In the practice	What it means
Owner	One person, and only ever one. Everything an administrator can do, plus three things nobody else can: close and reopen a period in any company, take and restore a backup , and hand ownership on. The owner’s own account cannot be touched by anybody else — not their password, not their address, not their second factor
Administrator	Runs the practice: add companies and people, read portal passwords, set rates, see the audit log. Reaches every company. Not the backup (§12.3), and not the period lock unless they are also made that company’s approver — see below
Staff	Only the companies they are given, at the level they are given them

In a company	What it means
Approver — “to change, and to close months”	Work fully in that company, and close and reopen its months. The one act this application asks a named person to stand behind
Preparer — “to change”	Work fully in that company. Prepares what the approver stands behind

In a company	What it means
Viewer — “ <i>to look at</i> ”	See it and change nothing (a banner says so). Not the company’s sensitive figures — the hourly rates and values on the Timesheet, and the full bank details of its payment accounts — unless and the figures is also ticked for that company
Switched off	Nothing, whatever the role. The account stays on the list, named, with the date — see <i>When somebody leaves</i> below

And one thing that is **not** a role, because it says who somebody is rather than what they may do:

Beside the role	What it means
The client’s own person	A director or an auditor of one of the companies you bill for, given a sign-in to see what has been invoiced to their own clients. They hold one company, at whatever level you give them, and they do not see the practice’s own settings — see <i>A sign-in that belongs to the client</i> below

Access is given company by company under **Settings** → **People & access**. A person sees only the companies they hold — nothing else exists for them in the switcher, lists or reports.

What “and the figures” covers

One tick-box, beside *Viewer* — *to look at* on **Settings** → **People & access**, and it governs **everything this application treats as a sensitive figure** for that company:

- the **hourly rates and values** on the Timesheet (since 2 October);
- the **full bank details** of the company’s payment accounts — the IBAN, the account number, the SWIFT code and the name the account is held in (since 5 October).

Unticked, a viewer still sees *which* accounts the company holds: the label, the bank’s name, the currency and the **last four digits**, which is enough to tell two accounts apart and to see which one an invoice was paid into. The card says the details are withheld, so a short row is not read as one somebody failed to finish.

Nobody else is affected. A **preparer** or an **approver** sees everything — withholding a figure from somebody who may change it is for show — and so does the owner and an administrator reaching every company.

What this does not do, deliberately: it does not change an invoice. An invoice prints the account it named, in the words it was printed with, and whoever can open the invoice reads it as the client received it. What the rule

withholds is the *list* — every account the company holds, including the ones no invoice has ever named — which is a different thing, and the only one of the two that is anybody's banking rather than a document.

Why the approver is a grant and not a rank. Administrator and Approver are alternatives, not rungs: one runs the practice, the other stands behind one company's books. So an administrator who should close a client's months is made **staff** and given that client as **approver** — which is what the People screen says in so many words (“make them staff to give them named companies instead”). The owner needs neither, and can close and reopen anywhere.

*Until 5 October this application had two roles — super admin and ordinary user — and “super admin” meant all of the above at once. Everybody who was a super admin is now an **administrator** reaching every company, and every “to change” grant is now **preparer**, so nobody's access changed on the day. What is new is that the owner is a role rather than an address, and that closing a period can be given to one person for one company.*

When somebody leaves, and when they lose their phone

Switch off their sign-in ends every session they have, **cancels any password link already in their mailbox**, and refuses any further sign-in. An attempt afterwards gets the same words as a wrong password, so the sign-in screen cannot be used to find out which addresses are accounts.

The link half is new on 8 October, and it was a real gap. An invitation lives 72 hours and a reset an hour, and switching the account off used to leave either one alone — so somebody walked out on a Tuesday with a live link in their inbox, and on the Thursday it still set a password on an account the practice believed was shut. That password would not have got them in, because “can this account sign in” is read on every request. What it would have done is *work*, quietly, and the register would carry a password being set on an account nobody was watching. The links are deleted with the sessions, in the same statement, and the register line says how many there were — a nought there is as useful as a two, because it says nothing was in flight.

And the same day, the two last steps of signing in started asking as well. Typing the password has checked this since 3 October. Answering the **code** prompt, and signing in with a **passkey** alone, did not — and both of those hand out a session. So an account switched off while somebody's sign-in was half finished, or one whose passkey was registered before the switch-off, got a cookie and a sign-in line on the register. The next request it made would have been refused, so nothing was reachable through it; what was wrong is that the practice's own log said somebody signed in after being walked out, and the person was told they were back.

The account is not deleted, and that is the change: it stays on the list with its name, the date it was switched off, and the companies it held. The audit log records the *address* that did each thing — it always survived an account being removed, because there is no foreign key to the people table and the address is copied onto each line as it is written — but “which member of staff was that, and what did they have” was only ever written down on the row itself, and deleting it took the answer. **Let them sign in** puts it back.

Clear their second factor is for a lost phone. It signs them out everywhere and asks them to set up a new authenticator next time; it leaves their password alone, so somebody who has lost both needs this *and* a password link. It is refused on your own account, which has its own screen under *Your sign-in* — and if they have no factor set up, it says so rather than appearing to work.

A **role change** and a **change to which companies somebody holds** now also end that person's sessions. Their access was always re-read on every request, so a narrowing has always taken effect at once; what ending the session adds is everything else a session carries — above all a *confirm it is you* proved minutes earlier under the role they have just lost. Saving the access screen without changing anything does **not** sign anybody out.

Each ticked company carries a level, and the dropdown names it before it explains it — **Approver — to change, and to close months, Preparer — to change, or Viewer — to look at**. *Viewer* has one further tick-box beside it: **and the figures**. Left unticked — which is how every grant starts — that person sees the company's recorded hours but not the hourly rate or the value of them. Tick it for a manager or a partner reviewing the company; leave it for somebody who has been given the company only to see what has been invoiced to their own clients, because an hourly rate is what the practice charges and not their business. The box appears only for *Viewer*: a person who may **change** a company's figures is not protected by being unable to see them, so there is nothing to decide.

Adding somebody

Two ways, under **Add somebody**, and the first is the one to use for anybody you cannot hand a password to across a desk.

Email them a link. Fill in the name and the email address, leave the password box empty, and press **Email them a link**. The account is created with *no password at all* — nobody can sign in as it, not even with a guess — and they are sent a message with a **Set your password** link. They choose their own, so nobody in the practice ever knows anybody else's. The link works **once** and lasts **72 hours** — three days; sending another one kills the first, so pressing the button twice never leaves two keys to one door.

Add with this password. The older way: you type one, press the button, and tell them what it is yourself. It is not emailed and cannot be read back off the screen afterwards. Use it when the app's own mail server is not working — which is the moment you would most want a way to let somebody in — or when they are standing beside you.

Either way, tick their companies afterwards and press **Save access**.

What the invited person does. They open the link, type a password twice, and are put on the sign-in screen to use it. A password the practice's rule refuses — too short, or built out of their own name — is refused *without* spending the link, so a mistyped one costs nothing. On their first sign-in they are asked to set up a second factor; after fourteen days an account without one stops working.

If the link has expired or been used, the page says so and tells them to ask for another. It says the same words whether the link was used, has expired, or never existed, on purpose: anything more would confirm to a stranger that a link was once real.

One prerequisite, once, on the server: the app has to know its own web address (`APP_ORIGIN`) or the link would point somewhere useless. Until it is set, **Email them a link** refuses and says so rather than sending a message nobody can use. The same setting is what passkeys need.

Under each person is a second, separate list: **may record time for**. That is the timesheet on its own — the person gets the Timesheet for that company's clients and sees nothing else of the company: no invoices, no

money, and no rate or value. It is what an audit junior or a part-time bookkeeper should hold. Ticking a company there does not give any access to the company itself, and the two lists are kept apart.

It does not, on its own, let them record anything. **A sign-in and a staff record are different things** — see 10 — and the timesheet books hours against the second. Somebody given this grant and no staff record opens the module to a sentence telling them their sign-in is not linked to one. Both are needed: the grant decides which companies' clients they may book against, the staff record is the person the hours belong to.

It asks you to confirm it is you, giving one and taking one away alike, since 8 October 2026. It is a change to what an account may do — the same class as adding somebody or changing their role — and it was the one of those that asked for nothing.

And it cannot be given to a client's own person for a company that is not theirs. Somebody marked *outside the practice* sees one company, and that is the whole meaning of the marker; the account screen refuses a second company and refuses the tick on anybody holding two. This grant went round both — it is read beside the company grants by the same rule, so it is a second company in every way that matters: that company's timesheet, its clients, its services and its rates. Their **own** company's timesheet can still be given to them, which is the ordinary use of it for a client's bookkeeper. (ENTITY-2)

Being an administrator

The role is set on each person's card under **Settings** → **People & access**, by the button reading **Make an administrator** or **Make staff**.

Making somebody an administrator gives them no companies. This changed on 11 October 2026 and it is the one thing to know before you use the button. Promoting asks first — *"An administrator runs the practice — people, teams, the register, every stored password of the clients they reach"* — and then says plainly that they reach **no company** until you tick one or press **Give them every company** on their card. Their card says the same thing in bold while they hold nothing.

So making a full administrator is now two presses rather than one: **Make an administrator**, then **Give them every company**. That is deliberate. It used to be one press, and the reason it had to change is in the panel below.

Why the second press exists. Saving a role used to set the reach with it, which meant *saving a role* was a way to **widen** somebody — and the only thing guarding the reach was a check on the reach itself. An administrator limited to two companies could send their own card back with the role unchanged at **administrator**, which they already were, and come out reaching all twenty-one. Nothing about them had changed except the thing nobody was looking at. The rule now runs one way only: saving a role can take reach away (making somebody **staff** clears it) and can never add any. Adding reach is the button, and the button asks who is pressing it.

An administrator limited to certain companies

Since 9 October 2026 an administrator need not reach every company. Two buttons sit on an administrator's card:

- **Limit to certain companies...** turns the reach off. From that moment the person reaches the companies they have been ticked into and no others, and the company ticks appear on their card like anybody else's.
- **Give them every company** turns it back on.

Both end that person's sessions, because reach is part of what a session carries: a narrowing that waited for their next sign-in would not have happened yet.

What a limited administrator keeps and what they lose:

- **They still run the practice.** The people list, the register, the reference data, adding an account, changing a role — none of that names a company, so none of it is touched. They are an administrator.
- **They do not reach a company they were not ticked into.** Its clients, its invoices, its timesheet and the stored portal passwords of its clients all answer *"You are an administrator of certain companies only, and that is not one of them."*
- **They cannot widen anybody's reach, their own included.** Turning the flag back on for themselves is refused, and so is ticking anybody — themselves or somebody else — into a company they do not hold. Only the owner, or an administrator who already reaches everything, can hand out reach.
- **They cannot give more than they hold.** Added 11 October 2026. Reach used to be compared as *"do you hold this company"* and nothing more, so a limited administrator who was only a **viewer** in a company could make somebody an **approver** there — and then be that somebody. They can now give, in each of their own companies, up to the level they hold there and no further: a viewer hands out viewer, a preparer hands out preparer, an approver hands out any of the three. The refusal says *"one of the levels named is above your own in that company"*.
- **They cannot edit their own card's companies at all.** Also 11 October. **Save access** on their own card is refused outright — *"your own companies are not yours to change"* — because the alternative is an administrator choosing their own level in the companies they already hold. The owner and an administrator who reaches everything are unaffected: their reach comes from the role, so their own ticks decide nothing.
- **They cannot take over an account that reaches further than theirs.** Also 11 October, and the most serious of the five. Setting somebody's password, clearing their second factor, sending them a reset link, changing their email, changing their role, and switching them off are all refused when that account reaches a company the limited administrator does not, or holds a higher level in one they share. Without it the limit lasted one request: set the password of an administrator who reaches everything, sign in as them, and you hold what they hold. The refusal names which of the three it was — *"reaches every company in the practice"*, *"holds a company you do not"*, or *"holds a higher level than you do in one of your own companies"*. An account holding less than they do is still theirs to manage, which is most of the practice's staff.
- **They cannot lift their own narrowing.** Also 11 October. If a **team** or a list of **clients they look after** is what holds them to part of a company's book, they cannot take themselves off that team or clear their own assignments. Adding somebody alongside themselves is fine; it is only the act that widens *them* that is refused.

- **Their linked-application keys reach no further than they do.** Also 11 October. A key reading *every company* is refused, and so is one naming a company they do not hold — on making it and on re-pointing an existing one. A key is a password that goes on working after the session, the sign-in and the employment that made it, so a limited administrator minting one for the whole practice was the quietest of the five holes.
- **A company's timesheet is reached like anything else now.** Also 11 October. The timesheet screens were the one place that still waved every administrator through to every company; they now read the same company ticks as the rest of the app, and a viewer's tick reads a company's timesheet without keying time into it. The practice's own timesheet reference data — staff, kinds of work, audit firms, rates — names no company and is unchanged.
- **The owner is never limited.** The buttons do not appear on the owner's card, and a row that somehow carried the flag down — from a restored backup, say — would make no difference: the owner's reach comes from the role, not from the column.

What narrows, and what does not. Since 9 October the rule bites on everything that is about one company. In order of how much it matters:

- a client's stored portal passwords — TAXISnet, CyLogin, TaxForAll;
- deleting a company, and deleting a client (a client billed by two of your companies needs **both**, since deleting them takes them from the other company too);
- the whole of **Settings** → **Teams** for that company: reading its teams, making one, renaming one, deleting one, and setting a team's clients or its members. Reading matters here as much as writing — a team's screen says which of that company's clients each of its people looks after;
- **Clients they look after** for that company;
- giving somebody that company's timesheet, and taking it away;
- that company's outgoing-mail settings.

Two acts belong to the practice rather than to any company in it, and those want an administrator who reaches **all** of it, not some:

- **adding a company** to the practice;
- **the practice's own outgoing mail server** — the one that sends sign-in codes and password links. A company's own mail settings still narrow as above.

And one known limit, which is not an oversight. The **register** — the audit log, the access log, the uploads log and the alerts — stays whole for every administrator, limited or not. Its rows name clients and companies a limited administrator cannot otherwise reach. Narrowing it means filtering the rows rather than refusing the screen, which is a different and larger piece of work, and a half-filtered register is worse than an honest whole one. If that matters for the person you are about to limit, do not limit them.

The rest of the administrator-only screens name no company at all and are unchanged: the people list and the accounts on it, the practice's timesheet reference data (staff, task types, audit firms and rates), the linked-application keys, the credentials key and the exchange-rate refresh. There are forty-three of those, each one

listed by name in `src/services/limited-admin.test.ts` with the reason — so a route added next year cannot quietly arrive practice-wide; the test fails until somebody classifies it.

(An earlier edition of this section said “around twenty-five” routes remained practice-wide. The real count, once they were enumerated rather than estimated, was sixteen route entries covering the seven things listed above — and all sixteen were reproduced against a running server before being closed.)

Three rules the app enforces, and it is worth knowing they exist before you need them:

- **There is always at least one.** Counting the owner, there can be any number; there can never be none. Demoting or removing the last is refused: *“This is the only super admin. Make somebody else one first, or there would be nobody who can.”*
- **You cannot remove your own sign-in.**
- **Resetting somebody’s password ends every other session they had.** That is the point of it — a session would otherwise outlive the reset — and it is why the reset is the right response to somebody leaving or an account going wrong.

Being the owner, and handing it on

One person is the owner, and the app holds to that in the database itself rather than only in a screen: a second owner cannot be written at all. The owner is whoever holds the role — not whoever matches an address in a file on the server, which is what it used to be and is the single most important thing that changed on 5 October.

What the owner has that an administrator does not:

- **Close and reopen a period in any company**, without being made its approver.
- **Hand ownership on**, which nobody else can do, for anybody.
- **An account nobody else can touch.** Their password, their email address, their second factor and their companies are refused to every other account: *“Only the owner can change the owner’s account. If ownership should move, the owner hands it on from this screen.”*

That last one is the point of the whole change. While the owner was an address, an administrator could change the owner’s email, then their own to it, and the owner’s powers followed the string — two ordinary saves. A role cannot be moved that way.

Handing it on is the button on the other person’s card, offered only to the owner. It asks you to type their name, because it cannot be undone from your side afterwards — once it is done, they are the owner and you are an administrator, and only they can hand it back. **Everybody involved is signed out**, you included, so nothing carries on running under the role it no longer has. It is refused to somebody whose sign-in is switched off: an owner who cannot sign in is a practice with nobody who can reopen a period or hand the role on again.

A sign-in that belongs to the client

Somebody outside the practice — a client’s director, their auditor — can be given a sign-in to see what has been invoiced to their own clients. The button is on their card: **They are the client’s own person.**

What it changes is small and it is all in one direction:

- the practice's own settings go — **Deadline rules** and **Defaults** are no longer offered, and are refused by the server besides. So are the practice's kinds of work and its list of audit firms;
- **their own company is untouched.** Its details, its products, its clients, its invoices and its figures are exactly what their grant gives them. That is what the sign-in was for.

Two rules the app enforces, and it enforces them in both directions:

- **They hold one company.** That is what “the client's own person” means. The tick is refused on somebody holding two, and a second company is refused to somebody already ticked — untick them first if they really are to hold more.
- **An administrator cannot be one.** The role runs the practice, so the two together would be a contradiction; the tick is refused on an administrator, and an administrator cannot be made of somebody ticked.

Ticking it signs them out, like a role change, because a session carries what it was built with and a narrowing that waits for the next sign-in has not happened yet.

Everything else such an account could once reach was closed earlier: the practice's staff list and the hourly rates on 2 October, and every screen narrowed to the company in the switcher on 5 October.

Teams, and who looks after which client

Until now a company was all-or-nothing: hold it and you hold every client it bills — their bank details, their fees, their correspondence and their filings. For a practice where one junior does six clients and another does the other fifteen, that was the whole book to each of them.

Settings → **Teams** narrows it, one company at a time.

The rule, in one sentence, and it is the one to remember:

Somebody with **no team and no client of their own** sees **every** client of the company, exactly as before. The narrowing starts the first time you put a client on a team, and only for the people in that team.

That is deliberate. The other way round — nothing visible until somebody is assigned — would blank every screen in the practice on the morning this went live and leave you assigning twenty-one clients before anybody could work. The cost is the other half of the same sentence: **forgetting to assign somebody gives them everything**, so the screen says so where you are standing when it matters.

Adding a team

1. Open **Settings** and choose **Teams** from the list on the left. It is offered to an administrator and the owner only.
2. The card is about **the company in the switcher**. Switch company first if it is the wrong one — a team belongs to one company, and two companies may each have an “Audit” without them being the same team.
3. Type a name in the box at the bottom — *Audit, Payroll, Maria's clients* — and press **Add team**.

4. The team appears in the list with **0 people · 0 clients**. An empty team narrows nobody, so nothing has changed yet.

Putting people and clients on it

1. Press **Who and what** on the team's row. Two columns open: everybody who can sign in on the left, this company's clients on the right.
2. Tick the people who are in the team, and the clients it looks after.
3. Press **Save team**. The counts on the row update.

From that moment, each person in the team sees **only** those clients in this company — on the client list, the invoice list, the quotations, the to-dos, the compliance list, the search box, the dashboard, the reports and the report builder's export. Typing a client's address in by hand is refused outright, with *"That client is not one of yours. Ask for it to be added to your team."* — refused rather than answered with an empty page, because an empty page reads as a fault.

And so is a RECORD of theirs, reached by its own address. (TEAM-2) An invoice, a quotation, a to-do or money laid out for a client outside the team is refused the same way — opening it, printing it, editing it, emailing it, deleting it, all of them. That matters because a record's address is not a secret: it is in every list the person legitimately sees and in the address bar of every record they open. Until 8 October 2026 the narrowing covered the lists and not this, so a person who was taken off a client could still reach that client's invoices one at a time if they knew where to look.

A record naming **no** client — an invoice typed against a name rather than a client, the practice's own housekeeping to-do — belongs to nobody and stays reachable by everyone in the company.

Who is looking after this client, on the client's own window

A line under the client's name says **Looked after by ...**, naming the people and the teams — or saying that nobody in particular has been put on them, which is what makes them visible to everybody holding the company.

A **client's own person** (§1.4) sees that line too, and sees the **names without the email addresses**. "Looked after by Maria" is what the line is for and tells them nothing they should not know; the practice's addresses are its own internal directory, and until 8 October 2026 that screen handed the whole of it over. (ENTITY-2)

One client, one person

"She also does this one client" does not deserve a team of its own. On **Settings** → **People & access**, each person's card has **Clients they look after** — the clients of the company in the switcher, ticked one by one, on top of whatever their teams carry.

Saving it replaces **only this company's** part of their assignments. A list sent from this screen cannot clear what they were given in another company, which matters because since 5 October this screen cannot see the other companies anyway.

Who is never narrowed

	Why
The owner	Runs the practice
An administrator reaching every company	The same
An approver of that company	They stand behind what is prepared there and close its months. Signing off on a book you may not read is not a review

So an approver on the same team as a preparer still reads the whole company, and that is the intended answer rather than an oversight.

A company they have no team in is untouched. If Maria's teams are all in the practice company and she also holds Kweller Ltd, she reads every client of Kweller Ltd as she always did. Narrowing one company never leaks into another.

A client billed by two companies is reached through the company you are **working in**. This changed on 12 October 2026 and it is worth reading twice, because the old behaviour was the opposite and it reads as the kinder one.

Say the Shared Ltd record is billed by both the practice company and Kweller Ltd. Maria is narrowed by a team in the practice company and Shared Ltd is not on her list there; in Kweller Ltd nobody has narrowed her at all. Until 12 October she reached that record wherever she was, because *any* company billing it that left her unnarrowed was enough. Now:

- with **Kweller Ltd** in the switcher she reaches it, exactly as before;
- with the **practice company** in the switcher she does not — the record is refused and it is absent from the client list.

The narrowing still does not leak between companies: she reaches the client in Kweller Ltd, where nobody narrowed her. What changed is that she has to be *in* Kweller Ltd to do it. The old rule meant a team that took a client off somebody in one company had not really taken it off them there, as long as they held any other company that billed it — which is not what anybody setting up that team believed they were doing. Since the app works on one company at a time, the company in the switcher is the honest answer to *which* company a request is about.

Switching a team off, and deleting it

Switch off suspends a team without losing what it looks after: its people fall back to whatever else they hold, which for most people is the whole company again. The app says so when you press it, because it is a **widening** and a widening should never be quiet. **Switch on** puts it back exactly as it was.

Delete removes the team for good. Its people keep their sign-ins and go back to seeing every client of the company, unless another team still narrows them.

Both are on the audit log: *Created, renamed or switched a team, Changed who is in a team, Changed which clients a team looks after, and Changed which clients somebody looks after*. The line records **how many**, never which — the names are on the screen, which is where they belong.

Where it says who looks after a client

Open the client and look at **Who looks after them, and whether they are also one of ours**. It names the people and the teams — or, when nobody has been put on them, says so in words:

Nobody in particular. Everybody who holds this company sees this client.

A client that is also one of your own companies

Sometimes the same real entity is both: you bill Kweller Ltd as a client of the practice, and Kweller Ltd also issues its own invoices out of this application as a company. Two records, one entity, and until now nothing said so — which is how the same VAT number came to sit in two places that could quietly drift apart.

On the client's own screen, under **Who looks after them, and whether they are also one of ours**, the box **This client is also our company** offers your companies. Choose one and the two records are tied together; that company's own Settings page then says *This company is also kept as the client ...*

It ties and it does not grant. Choosing it gives nobody access to anything — who may open that company is still decided on **People & access**, and nothing about a client assignment carries across. That was a deliberate choice: an assignment that silently handed over a whole set of books would undo the narrowing above in exactly the cases where it matters most.

One company can be claimed by **one** client record. A second client pointed at the same company is refused, naming the one that already has it — because a company is one entity, and two client records claiming to be it is a typing mistake rather than a shape to keep.

Resetting somebody's password

Two controls on each person's card, and they answer the same need in different circumstances.

Email a password link sends them a message with a link to set a new one. Nothing changes when you press it: their current password goes on working until they use the link, which lasts **24 hours** and works once. This is the ordinary answer — they choose their own password and you never know it.

Set a password myself asks you for one, twice, and sets it. You then have to tell them what it is. For when mail is not working, or they are beside you.

Either way, **every session that account had ends** — which is the whole point when somebody has left or an account is thought to be known. Neither one touches their **second factor**: somebody who sets a new password through a link is still asked for their code at the sign-in screen, so a mailbox on its own is not an account.

Both are an administrator's to do, and both ask you to confirm it is you first, because either one hands an account to whoever holds a mailbox or a piece of paper.

A password you typed has to be changed by them

Since 8 October, a password an administrator **types** — whether when adding somebody or later, on either of the two controls above — has to be changed by that person before they can do anything else. They sign in, set up their authenticator if they have not yet, and are then shown one screen: *the password you signed in with was typed for you by somebody at the practice, so two people know it — choose one of your own*. They type what you gave them, then their own twice, and the app opens. Until then every route is refused, and there is deliberately no way past it: an account in that state can reach the change-password screen, the screen that sets up an authenticator, and the way out, and nothing else. A “later” button would lead to a screenful of refusals.

What it is for, put plainly. A password typed for somebody exists in two places — in their head and in the head, note or message of whoever typed it. That is not a fault in the feature; it is what the feature is. Before this, the temporary password went on working for ever, so the person who typed it could sign in as them months later and leave nothing behind but a sign-in line that reads like any other. This is the end of that story.

A password LINK sets nothing. The person who follows a link chooses the password themselves, so there is no second person to worry about and no screen to show them. That remains the better door and the one the screen recommends. Using a link also **clears** this, which matters on the day somebody cannot get past the screen for any reason: send them a link and they set their own password at the end of it.

The order is the second factor first, then the password, and that is deliberate rather than incidental. The authenticator is the lock; the password is what is behind it. A password changed on an account that anybody holding the old one can still sign into with the password alone has been changed in front of whoever already knows it.

Two accounts are never asked. Somebody setting their **own** password — on *Settings* → *Your sign-in*, or through their own card — chose it, so there is nothing to change. And the **owner account seeded from the server's .env** is not marked, today or ever: it is the account that exists before there is anybody to type a password for it. Whoever looks after the server should still change **ADMIN_PASSWORD** from whatever was first set, and the owner can change their own password at any time from *Settings* → *Your sign-in*.

Nobody who already uses the app is affected. The application never recorded how an existing password was set, so nothing is assumed about one: every account that existed before 8 October carries on untouched. From that date it is recorded as it happens. Anybody whose password was handed to them before then and who ought to change it can be sent a link, or given a new typed one — which will then say so.

If you are locked out of the only administrator

Worth reading once, now, rather than on the day.

An administrator can reset anybody's **password**, and — since 3 October — can also **clear anybody else's second factor** from their card on *People & access*. That second one is the answer to a lost phone, and it used to

require somebody editing the database. It signs that person out everywhere and asks them to set up a new authenticator next time; it does not touch their password.

So the only account that cannot be rescued from inside the app is **the last administrator's own** — and, separately, **the owner's**, since nobody else may act on it at all. For those the way back in is **their own recovery codes**, because clearing your own second factor is deliberately not offered from the people list: a button that could do that would be a way past the password the enrolment screen asks for. Where the codes are gone too, no one who can sign in can help: the only remaining route is somebody with direct access to the database on the server.

Two things follow, and they are cheaper than the alternative. Keep the recovery codes somewhere that is neither the app nor the machine you sign in from — the app says with the company seal, and means it. And a **second administrator** is the cheapest insurance there is: with one, a lost phone is somebody else pressing *Clear their second factor*, and with none it is a trip to the server. (That insurance does **not** extend to the owner's own account, by design. Keep the owner's recovery codes with the seal.)

Confirming it is you

Ten acts ask you to **confirm it is you** again at the moment you do them — a code from your authenticator app, your device's fingerprint or face, or a recovery code:

Portal passwords	read them, and change them
Ways of signing in	add one, remove or replace one
Clients	delete one, and everything held for them
Companies	delete one, which destroys the clients only it holds
Accounts	create one, change what one may do, change the email address on one , email somebody a link to set their password, clear somebody's second factor , switch a sign-in off or back on , or hand ownership of the practice on
The practice	download a copy of it, restore it from a file
Outgoing mail	change the server a company's mail is sent from
Linked applications	make a key one reads this practice through, change the date and companies of one that exists, or revoke one

Revoking a key asks too, since 9 October. It used to be the one act on that screen that did not, and the reasoning was that it is the emergency brake: making a key hands out data, while stopping one only stops the accounting application importing, which would be noticed within the hour. The second review disagreed, and

on reflection it is right. What somebody with a borrowed session gets from Revoke is not “an import stops” — it is the link between the two applications broken with nothing anywhere to say why, because a revoked key looks exactly like a key somebody meant to revoke. And “noticed within the hour” is the optimistic reading: what is noticed is that figures have stopped arriving, and the first place anybody looks is the other application. The brake still works — it costs one code, the same code this application asks for everywhere else.

Why an email address is on that list, when a name is not. An address is how somebody signs in and where a password link is sent, so whoever can change it can be sent the link and take the account. That is a permission, not a detail like a spelling. Changing one therefore asks, whoever’s account it is and whatever the new address, while correcting a misspelt **name** still asks for nothing. (Deliberately: a code demanded for a typo teaches everybody to keep one permanently to hand, which is how confirming it is you becomes a formality.)

Until 5 October there was a sharper reason, and it was a weakness: the one address in the server’s `ADMIN_EMAIL` was the only account that could reopen a closed period, so moving an address moved a power. The owner is now a role, so nothing follows an address any more — and the owner’s address cannot be changed by anybody but the owner in any case.

A confirmation lasts **five minutes**, so a run of these does not ask each time. An account with no second factor cannot do any of them at all — and since MFA-1 it cannot do anything else either, so the case barely arises. The box offers **I have lost my phone** where unused recovery codes are on file, and a code is spent whether or not it turns out to be the one you needed.

Signing in does not count as confirming it is you. Since 8 October it is asked separately, so the first of these acts in a sitting asks even though you typed a code minutes earlier at the sign-in screen. That is deliberate and it is the change you will notice most. The code at the door answers *who is at this machine*, once, for a session that may last a fortnight. These ten ask something narrower and much later: *is the person at the keyboard still the one who signed in, and did they mean to do this?* Until this change, signing in answered both — so a session one second old had the whole list open for five minutes, and the five-minute window, which reads as a narrow one, started at a moment somebody who had found a signed-in machine could simply wait for. The payroll and accounting applications both stopped doing this, and the review asks for one rule across the three.

The code on your phone will be the one you have just used. Each code works once, so the box refuses it — and it says so: *“You have just used that code. Each one works once, and that was the code you signed in with — your authenticator app will show a new one within thirty seconds.”* Wait for the next one and type that. This is not counted against you: it spends none of the attempts that lock the codes, and nothing is written on the audit log, because it is not somebody guessing. A code that is simply **wrong** still counts, there as everywhere.

The sign-in screen does not say this — there the same words would be answering whoever typed them, who may be a stranger holding a password, so it gives one sentence for a wrong code and a used one alike. At the box you are already signed in as the account, so being told that your own last code was your own last code tells you nothing you did not type.

Setting up an authenticator still counts. It costs a password and a code typed on purpose for that act, so it is a proof rather than a side effect of arriving — which matters on somebody’s first morning, when they would otherwise be asked for a third code within a minute of their first two.

1.4a Light or dark

The button beside the company switcher, at the top right. Until it is pressed the app **follows the machine** — dark when your computer is dark, light when it is light, and it keeps following if you change that at dusk. Pressing it overrides that, and pressing on returns it to following. The button's tooltip always says what it is on and what the next press does.

The choice is remembered in that browser rather than on the account, so an office machine in daylight and a laptop at night can differ for the same person.

Two things never follow it. A **document** — an invoice, a credit note, a quotation, a client statement — is ink on white paper wherever it is read, on screen and in the PDF the client receives. And anything **printed** prints light.

The Help pages follow the same choice.

1.4b How big the screens are drawn

The **Aa** beside the light/dark button, with five steps: **90%, 100%, 110%, 125%, 150%**. The list is fixed rather than a number box, because each step is a rule in the stylesheet rather than a calculation.

Unlike light and dark, this is kept **on the account** and not in the browser, and that is the whole reason it exists. Staff sit at whichever desk is free, work a client from a laptop in the afternoon, and are sometimes in a fresh browser profile. A size set in the browser's own zoom travels with none of that, so somebody who needs 125% sets it again, everywhere, for ever. This one follows the person and is right the first time they sign in on a machine. It is also *on the bar*: the browser's zoom is behind a menu or a keystroke, and a control only reachable by people who already know it is there is not available to the people it is for.

On a machine somebody has used before, the size is applied before the page is drawn, so there is no resize a moment after it appears. On a machine nobody has used, the account's answer arrives a moment after the page and the screen settles once.

90% is deliberately on the list. Somebody with a small laptop and a wide document list wants *more* on screen, not less, and that is the same need read the other way. Nothing below it: at 75% the figures in a VAT return stop being legible.

Two limits worth knowing:

- **Nothing printed is affected, and no PDF is.** An invoice, a statement, a quotation and a client's credit note are documents somebody is handed and the practice files; they come out identical at 150% and at 100%. The size applies to screens only.
- **A shared desk forgets.** If the account has never chosen a size, whatever that browser was remembering is cleared rather than used — so the next person to sit down does not find the app at 150% with nothing on screen explaining why.

1.4c Reaching the screen, and asking for less movement

Two things that are invisible to anybody who does not need them, and are therefore worth writing down:

- **Skip to the screen.** The first thing **Tab** reaches on any page is a *Skip to the screen* link, shown only while it has focus. It puts focus on the work itself. The bar above the work carries the brand, seven module names, the search, Help, light/dark, the size, the company switcher, the person's name and Sign out, and then the row that says where you are — without this link, every screen begins with all of that.
- **The bar itself.** The seven module names are buttons, so **Tab** reaches them and **Enter** or **Space** opens a module's menu with its first line already focused. Inside a menu the arrow keys walk the lines, **→** opens a line's own list and **←** comes back out of it, **←** and **→** on a top-level line move to the next module's menu, and **Escape** closes the menu and puts focus back on the name it belongs to.
- **Reduced motion.** With *Reduce motion* turned on in the operating system, every transition and animation in the app is cancelled and the two places that scroll the page themselves stop gliding. Nothing is hidden, no layout changes and no control is removed: the page arrives at once instead of over two-tenths of a second.

1.5 Working safely

- **Unsaved work is never lost by a wrong click.** Leaving a half-written invoice, credit note, pro-forma or quotation asks first: *Save as a draft, Discard it or Stay here*. Closing the browser tab gets the browser's own warning.
- **Back works, and so does the address.** Every screen is a step in the browser's history, so the Back button (or the **←** beside the screen name) returns to the previous screen — and every screen has its own address in the bar, so **refreshing the page keeps your place, a bookmark opens the screen it was made on, and an address sent to a colleague takes them to the screen you were looking at**. Four addresses changed when the navigation moved to the top of the screen; an old one still lands on the right screen and is rewritten to today's form as you arrive, so it is used once and never again. Signing in is the deliberate exception: it opens the Dashboard whatever address was left in the bar, because the one left behind by an expired session is somebody else's place rather than yours.
- **Closed periods.** Once the books are closed up to a date, nothing dated on or before it can be added, changed or deleted without re-opening it with a written reason (see 12.1).
- **Everything that matters is recorded** — who changed which figure, when, and who did what in the app (see 12.5).

1.6 Searching the company you are in

The **magnifying glass on the top bar** (or **Ctrl+K** / **⌘K** from anywhere) finds things in **the company in the switcher**, without having to know which module they are in:

It finds	By
Clients	Name, legal name, VAT number, registration number, tax ID, email, contact person
Invoices, credit notes, receipts, pro-formas	Number (with or without the dashes), the client named on it, reference, or the exact amount

It finds	By
Quotations	Number, the client or prospect

- VAT and invoice numbers are found however they are typed — “cy 1000 0001 x” finds CY10000001X, and “inv045” finds INV-045-2026. Amounts can be typed as 1190, 1,190.00 or 1.190,00.
- Results are grouped by kind, best match first. Up to six of each kind are shown.
- Choosing a result opens it: the client’s record, the document or the quotation.
- It never searches portal credentials, notes or rates. Voided entries are left out.

One company at a time, and it says so. Until 5 October the box searched every company you held, naming each result’s company and switching to it when you opened one. It now searches the company in the switcher and no other; where nothing matches, the box names the company it looked in, so “nothing found” is never mistaken for “not in this system”. Switch company and search again to look there.

That is the rule the rest of the app has always kept — the lists, the reports and the dashboard are all the switcher’s company — and the review of 2 October asks all three of the practice’s applications to work the same way. It matters most for somebody given one of the practice’s companies to look at: every screen showed them that company, and the search box showed them the practice’s other clients by name, with invoice numbers and amounts beside them.

1.7 One company at a time

Promoted to a section of its own on 14 October 2026. This was a **###** subheading inside §1.6, *Searching the company you are in* — and it is not about searching. It is about a rule that closes the company you are working in after half an hour, which is the thing somebody looks up when it happens to them at two o’clock; nobody looks for that under *Searching*. The 9 October *What’s new* entry promised it as “Handbook §1.9”, a section that was never created, so the reference dangled as well.

Since 9 October 2026 the **server** has an opinion about which company you are in, and not only the screen. Each sign-in has **one company open**. A request naming any other is refused, and after **thirty minutes** with nothing happening the open company is closed until you open it again.

That is two rules, and they answer two different things.

Why the first exists. Until now the open company lived only in your browser. Every request simply named the company it was about, and the server asked one question: do you reach it? So a single sign-in could work its way through every company you hold, one request after another, and nothing anywhere treated that as unusual. A tab left on yesterday’s company, a bookmarked address, a back button, or a request replayed by somebody who had your laptop for five minutes — all of them name a company, and none of them is you deciding to move.

Why the second exists. A window left open on a shared desk at lunchtime was a window still working at two o'clock. It now stops being live after half an hour.

What you actually see:

- **Switching is immediate.** Choose a company in the switcher and you are in it. Nothing about this makes switching harder; the switcher is simply the only thing that moves you.
- **A tab that has fallen behind asks once.** If a screen asks for a company this window is not on, a box appears naming both and offering to switch. Say yes and the request it was refused is sent again, so you land where you were going. Say no and you stay put. (The one thing lost by switching is a half-written invoice, which belongs to the company it was started in — and the box says so.)
- **Coming back after half an hour asks once.** One button, nothing typed, and you are **not signed out** — the box says so, because the commonest fear on seeing it is that you have been. Everything is where you left it.
- **Opening the app uses the company your sign-in has open**, not the one your browser happens to remember. Otherwise every return visit would greet you with a question.

What is **never** refused by this rule: anything that is not about one company. The people list, the audit log and the other registers, the reference data, and the **list of companies itself** — you must be able to see what there is to switch to.

It is per sign-in, not per account. You can have one company open on the desktop and another on the laptop, and neither disturbs the other. The thirty minutes is counted per window too, so a window being used never locks while another sits idle.

And it applies to everybody, the owner included. The people most likely to leave a window open on a shared screen are the ones with the most behind it.

The thirty minutes is one of the values agreed across all three of the practice's applications, alongside the invitation link's seventy-two hours and the reset link's twenty-four. If it ever changes it changes in all three.

2. Dashboard

The Dashboard opens on the company chosen in the switcher and answers two questions: *how is income going?* and *what needs attention?*

2.1 Income

- A period choice: **This year**, **Last 12 months**, **Last year**.
- Four tiles: **Fees invoiced** (with the change against the same months last year), **This month, Collected** (as a percentage of what was invoiced), and **Owed to you** (with how much is overdue).
- **Fees by month** — bars for this year against last year. Fees are net of credit notes and exclude VAT and reimbursements.
- **Income by client** — the five largest clients and everyone else.

- **Income by service** — grouped as set in Settings → Company details → Services this company provides. Invoice lines that match no listed service count as **Other services**.
- **VAT on your sales** — output VAT for this quarter so far and last quarter, with the date the return and payment are due. Quarters follow the company's VAT quarter ends — and where none are set the card says so rather than guessing silently: *“Calendar quarters assumed (Mar, Jun, Sep, Dec). If yours end in other months, set them under Settings → Company details”*. A company with no VAT number gets *“This company is not VAT registered — it charges no VAT.”* instead of a figure. Imported invoices without a VAT breakdown are counted from their tax total.
- **Not yet invoiced this month** — clients billed in each of the last two months but not yet this month, with an **Invoice** button that starts one for them.

Only euro amounts are added up; documents in other currencies are noted but left out.

2.2 The work queue (“What needs attention”)

One list, most urgent first, each item with a coloured stripe (late, due soon, open) and a link to the thing itself:

Tag	What appears
Compliance	Filing deadlines due within 60 days, or already late
Money owed	Invoices overdue or falling due within 30 days
Pro-forma	Issued pro-formas not yet turned into invoices
To-do	Your own to-dos

To-dos: add one with **+ Add a to-do (What needs doing?,** and optionally **By when** and **For a client**). Tick it off with the circle; change or delete it by clicking the row.

Ticking and deleting are not the same act and the app says so at the moment it matters. A ticked row is replaced by “Ticked off: *title* · **Undo**”, and that notice survives exactly one redraw — so the undo is there for the press that follows the mistake, not for later. The delete confirm reads “Delete this to-do? Ticking it off keeps it; deleting does not.”: a ticked to-do stays on the record, a deleted one is gone.

3. Invoices

The module's own lines are **All documents** (the list), **New invoice**, **New credit note** and **New pro-forma**. The three *New* lines open the same form in three states — there is one writing screen, and which document it makes is settled before you start rather than by a box you might change halfway through.

3.1 Kinds of document

Document	Numbering	What it does
Invoice	INV-001-2026 (per company, per year)	Bills a client
Credit note	CN-...	Reverses all or part of an invoice, at the invoice's own VAT rate
Receipt	RCP-...	Records money received against an invoice
Pro-forma	PRO-...	A request for payment that is not yet an invoice — counts in no revenue or receivables figure until converted

Two documents can never share a number. Numbers are allocated when the document is saved.

The number is filled in for you but is an ordinary box: type over it and the typed number is used. The uniqueness rule is enforced at the point of saving rather than at the point of typing, and the refusal names the document already holding it — “*number* is already used by another document of this company.”

How the next number is worked out (NUM-1)

The next number is one past the highest number in use in that series — not one past a count of how many have ever been issued. The series is per company, per document type and per year, so each of the four kinds above counts separately and each year starts again at 001.

The consequence, and the reason this is written down: **deleting the last document in a series gives its number back**. Issue INV-029-2026, delete it, and the next invoice is INV-029-2026 again. That is deliberate — numbers are supposed to run without gaps, and until 7 October 2026 they did not: the app kept a counter that only ever went up, so a deleted number was lost for good and the series was left with a hole nothing could fill.

A hole in the **middle** is a different matter and stays a hole. Delete 027 while 028 and 029 stand and the next invoice is 030: a number already on a document somebody is holding is never approached from above, and a ledger whose numbers do not run in date order is one an auditor queries. Only the trailing number comes back.

Because of that, deleting a document that is **past draft** and carries the highest number in its series adds a line to the confirmation box: that the number will be given to the next document, and that a copy of this one may already be with the client. Deleting a draft, or a document with others after it, says nothing extra — there is nothing to warn about in either case.

The same rule governs quotations (\$6), where only a draft can be deleted at all — so the number of a draft thrown away goes to the next quotation.

Two people saving at the same moment still cannot collide. The counter row the app used to take numbers from is still there and is still locked before anything is read, so the second save waits for the first and is given the number after it. The counter itself is no longer what decides; it follows the ledger, downwards as well as up.

3.2 The document list

Invoices → **All documents** lists the company's documents, with:

- **Filters** — client, type, status (Draft, Sent, Paid, Overdue), period (this year, this month, quarters, last year, all time, or **Other** — **choose dates...**), and sort order.
- **Tiles** — Total, Outstanding, Paid and Overdue, switchable to fees, VAT or reimbursements only.
- **Row actions** — View, Email (✉), Edit, Pay, Credit, Convert (pro-formas), Delete.
- **Exports** — Excel and PDF of everything matching the filters (not only what fits on screen).
- ⚡ **Bulk Receipts** — marks every invoice except the latest as paid by bank transfer (used for catching up history). It needs at least two invoices, and it **skips** any that name no payment account — naming them in the message afterwards rather than failing. Every receipt is backdated to its invoice, so against a closed period the whole run is refused rather than half-applied.

The list opens on ***This year*** and shows at most **500 rows**. The cap is on the list alone: where it bites, the list says “Showing the first N of M documents. The totals above cover all M; only the list is shortened.” So the tiles are never short — but a document from an earlier year is simply not there until the period is widened, which is the usual reason a search “finds nothing”.

3.3 Writing an invoice

- **Client** — picking a client fills in their address, VAT number, payment terms, currency, usual fee and the right VAT treatment. A client not on the list can be typed in by hand.
- **Lines (services)** — choose a service from the company's list, a period (month, range of months, or year) and optional extra wording; the line reads, for example, “Preparation of VAT return for March 2026”. A new service can be added from the invoice itself. Each line can carry a fee and a reimbursement.
- **Lines (goods)** — choose a product from the catalogue (price, unit and VAT fill in), with optional period and waybill, quantity and VAT rate per line.
- **Variable fees** — a service can be charged from the period's own numbers (transactions over an allowance, per employee, by the hour, a percentage, or your own wording). The fee is worked out on the invoice with **Work it out...**, and the working prints as an **appendix** on a following page so the client sees how it was calculated. A client can have their own agreed terms.

- **VAT** — worked out from where the client is:
 - Cyprus: 19% (or the line's own rate for goods).
 - EU business with a VAT number: 0%, reverse charge — the VAT number should be checked with **VIES**, and the app shows when it was last checked.
 - EU client without a VAT number: 19%.
 - Outside the EU: no Cyprus VAT, and the invoice says so.
 - The rate is always printed, 0% included.
- **Discount** and **reimbursements** are shown separately in the totals.
- **Where the client should pay** — every issued invoice names one of the company's payment accounts. Without one it can only be saved as a draft.
- **Duplicate check** — if a service for the same period has already been billed to the same client, the app names the earlier invoice and asks whether to bill it again or remove the line. Products are never flagged.
- **Draft or issue** — decided by the **Status** field in *Details*, not by which button is pressed. It offers *Draft*, *Sent* and *Paid*, and it **starts on Draft**. **Save Invoice** saves whatever Status says, so an invoice saved without touching that field is a draft and stays out of the figures; the toast says so. **Save Draft** forces a draft whatever Status says.

This catches people. The two checks that guard a real invoice — a payment account, and a fee working finished — only bite once Status leaves *Draft*, so an invoice saved straight through raises no complaint and appears nowhere. “Where is my invoice?” is nearly always this.

A **credit note** has no Status and no draft stage: it is issued the moment it is saved.

- **Templates** — four print layouts (classic, bold header, minimal accent, centred header). The picker is the row of four buttons at the top of the **Details** card, which also holds **Currency** (a fixed list) and **Reference**.
- **The From card is per document**. **Business**, **Email**, **Address**, **Phone**, **VAT** and **Tax Number** are filled in from the company and can be overwritten for this one document without changing the company itself.
- **Notes & Payment** — **Notes to Client** prints at the foot. **Payment Details** is taken from the chosen payment account, and anything typed over it is what the client sees instead. Either way a saved document keeps the wording it printed: changing the company's bank details afterwards never rewrites an invoice a client is already holding. Where the invoice's currency differs from the account's, a **Check** flag says so — a prompt to confirm, not a refusal.
- **A working left unfinished pins the document to draft**. Saving as *Sent* is refused with the line named, and emailing is refused outright — “...has not been worked out. Open the invoice, finish the working, and send it again.” This is the second of the two checks above, and the one that gives no clue on the list screen.

- **One VAT message is a warning rather than a note.** Where the customer qualifies for zero-rating but a goods line still carries a rate, the box turns amber and is flagged **check this**: a goods line keeps its own rate whatever the customer's country. Set those lines to 0% where the exemption applies and the wording is added.
- **A goods line begins with a category**, not a product — *Products, Royalty fees for sales, Royalty fees for Marketing and advertising, or Other...*, which replaces the product picker with a free-text description.

3.4 Credit notes

Started from **Credit** on an invoice (or New credit note). The lines are copied from the invoice; untick what is not being credited. The VAT rate is **fixed at the rate the invoice was charged at** — a credit note always reverses at the original rate.

3.5 Pro-formas

Written like an invoice. **Convert** turns it into a real invoice with the next invoice number and today's date; the pro-forma stays in the list showing what it became. Until converted it appears on the work queue and in no income figure.

The screen calls it a pro-forma throughout — the heading, the Save button, the line at the top, the printed document, the PDF and the Excel export all say *Pro-forma*, and one opened for editing is saved back as a pro-forma rather than becoming an invoice.

3.6 Payments and receipts

- **Pay** (Record payment) — date, method, amount, reference and **which account** the money went into (or cash). This raises a receipt and marks the invoice paid (or part paid).
- **The account is one of the company's payment accounts** — the same list an invoice prints from, under Settings → Company details. It opens on the default, which is where the client was *asked* to pay; check it, because that is not proof of where the money actually turned up. Setting **Method** to *Cash* removes the question, and the receipt is recorded with no account at all. Somebody holding the company only *to look at* sees the labels and the last four digits here, not the full details — see §1.4.
- A payment cannot be recorded against an invoice with nothing left owing.
- Every payment is entered here by hand. It used to be possible to raise receipts automatically from an imported bank statement; the statement import left this app with the rest of the books.

Bulk Receipts, on the Dashboard, is the one exception and a deliberately blunt one. It raises a bank-transfer receipt for **every invoice of the company in the switcher except the latest** and marks them paid. It is for a company billing on standing order, where everything up to last month is known to have been settled and entering forty receipts by hand is an afternoon. It names the company in its confirmation, skips anything it cannot properly receipt — a draft, or an invoice in a closed period — and says how many it skipped, because each of those is an invoice somebody has to account for. There is no undo: what it makes are ordinary receipts, removed one at a time.

3.7 Printing, PDFs and logos

- **View** shows the document as the client sees it. **PDF as the client gets it** is the file the server generates (the one attached to emails).
- Each document keeps **the logo it was issued with**. Replacing a company's logo never changes documents already sent. **Use the current logo** applies today's logo to one document if needed.
- If a logo file is missing on the server, documents print the company name instead and Settings says the file is missing.

3.7a What the downloaded file is called

The name is built from the document, not from the screen you downloaded it on. An invoice numbered `INV-034-2026` for Simallin Ltd arrives as:

`Invoice INV-034-2026 - Simallin Ltd.pdf`

A credit note says *Credit note*, a receipt says *Receipt*, and a quotation is named from its own reference — `Q-007-2026 - Simallin Ltd.pdf`. The client's name is cut at sixty characters, because some mail systems refuse a very long attachment name.

A client named in Greek keeps their Greek name. `Αντρέας Γεωργίου Ltd` arrives as `Invoice INV-035-2026 - Αντρέας Γεωργίου Ltd.pdf`. Before 13 October 2026 every Greek and accented letter was deleted from the name, so that invoice downloaded as `Invoice INV-035-2026 - Ltd.pdf` — the client simply missing from it.

A very old browser gets a transliterated name instead, because the rule that carries the file name over the web is older than Greek on the web: the fallback half of it can only hold plain English letters. So one of two names arrives, and the second is the Registrar's own English spelling of the first:

What the browser supports	The name it saves
anything from the last decade or so	<code>Invoice INV-035-2026 - Αντρέας Γεωργίου Ltd.pdf</code>
older than that	<code>Invoice INV-035-2026 - Antreas Georgiou Ltd.pdf</code>

Characters that cannot be in a file name are removed. A slash, a colon, a quote mark, a question mark, a star, an angle bracket and a vertical bar are each reserved on Windows, macOS or Linux — so `A/B Ltd` becomes `AB Ltd` rather than arriving as a name one of the three will refuse. Leading and trailing dots and spaces go too, because Windows discards them silently and the file then has a different name on two machines.

If a downloaded file ever arrives with `%20` in its name, or with `20` where the spaces should be, that is this fault and it is a bug — say so. It is what happened until 13 October 2026: the name left the server correctly and was then wrapped in a web-address encoding that browsers do not undo in that position, so `Invoice INV-034-2026 - Simallin Ltd.pdf` was saved as

Invoice20INV-034-202620-20Simallin20Ltd.pdf by one download and Invoice_INV-034-2026-_Simallin_Ltd.pdf by another.

3.8 Emailing documents

☒ on any document sends it from the company's own mail server, with the PDF attached. The address is filled in from the client record (the invoice email first). Sending to an address that is not on the client's record asks for confirmation, and the record keeps a note of it. Every attempt, sent or failed, is logged.

3.9 Deleting

A document can be deleted only if nothing stands against it (an invoice with receipts or credit notes cannot be deleted until those are), and only if its date is not in a closed period. Receipts cannot be edited — delete and re-record instead.

4. Clients

Two lines: **Client list** and **Deadlines**. The second is the compliance calendar for this company's clients and is described in §5; it sits under Clients because a filing date is a fact about a client.

4.1 The client list

Clients lists the clients of the selected company, searchable by name, contact, VAT or registration number, and filtered by **Active only** / **All** / **Inactive only**. Clients marked *invoicing only* are not accounting clients (see 4.3).

4.2 The client record

Clicking a client opens their record. The **sidebar** has four entries and no more: **Details**, **Credentials**, the group **Statutory information** (Directors, Shareholders, Registered office, Share capital) and the group **Compliance** (Deadlines). Some are hidden for a company that is not a regulated practice.

- **Details** — one pane, divided into its own headed sections: **Identity** (name, legal name, group, entity status, registration number, tax ID, VAT number, location and country), **Contact**, **Billing terms** (payment days, default fee, currency), **Fees worked out each period**, **Paid on their behalf**, **Quotation provided for the services**, **Compliance & deadlines** including **Audited by**, and **Status & notes**.

Two things that live *inside* this pane rather than in the sidebar, and are easily hunted for in the wrong place: **Check with VIES**, a button shown only when the location is EU and a VAT number is present, which checks the number against the EU register and keeps the result as evidence for zero-rating; and **Paid on their behalf**, money laid out for the client and what has been recovered, recorded and added to on this pane, which is always shown — a panel that appeared only once something was in it gave nobody a way to add the first thing, so the hiding was deliberately taken out. (**Fees worked out each period**, just above it, IS hidden where the company has no such services, which is most of them.)

- **Credentials** (*practices only*) — portal logins: Taxisnet / Tax Portal, Tax For All, CY Login, RoC code, and for employers the employer number, Sisweb and Ergani. Stored encrypted, readable only by an administrator after confirming it is them, and every reading is recorded.
- **Statutory information** (*practices only*) — Directors, Shareholders, Registered office and Share capital. Each entry is dated; changes are recorded with **Record a change...** so the history is kept. Certificates and filings can be recorded and their files attached.
- **Deadlines** (*practices only*) — this client's filings, with **Mark done**, and per-client dates for the annual return and VAT returns.

4.2a A client two of your companies both bill

A client can be filed under more than one of the practice's companies, and when they are, **there is still only one record**. Both companies see the same name, the same address, the same VAT number, the same statutory register and the same filing dates.

That makes writing it a different question from reading it, and since 7 October the app treats them differently:

- **Reading needs any one** of the companies that bill the client. Being shown a client you are already entitled to see through the company you hold takes nothing from the other company.
- **Writing needs every one of them**. A change to a shared record changes what the other company sees — and its own people may not hold your company at all, so they cannot even discover that it happened.

So a preparer holding one of the two may open a shared client, read everything on it, and change nothing. The client window says so at the top before anything is typed, and the Save button is not drawn: a form that fills in and then refuses is worse than one that explains itself first.

What counts as the client's. The record itself, the statutory register and its certificates, marking a filing done or reopening it, and the VIES check, which writes its result onto the record. All of them go through one rule rather than each having its own.

What does not. The agreed fee for a service is one company's own — a service belongs to a company — so setting it needs only that company, as it always did.

What the record tells you about the other company: its existence, and nothing else. Opening a shared client shows the companies **you** hold, never the one you do not — being taken off a company is meant to make it stop existing for you, and a list quietly carrying its id was that promise half-kept. What the screen is told is that the record is shared and that you may not change it, which is what it needs to explain itself and no more.

Who is unaffected: the owner, an administrator reaching every company, and anybody holding every company that bills the client. A client billed by one company is unchanged in every way, which is almost every client in an ordinary practice.

If this gets in the way, the answer is a grant rather than an exception: give the person the other company too, or ask an administrator to make the change. Both are deliberate acts by somebody who can see both sides, which is the point.

4.3 Accounting clients and inactive clients

- **Is this an accounting client?** — *No* means the client is invoiced only: no deadlines are tracked for them anywhere.
- **Inactive** clients stay on record but drop off new invoices and default lists.
- 🗑️ **Remove client** usually takes the client off *this* company's list and nothing more: other companies keep theirs, and past invoices are unaffected.

In one case it deletes them outright. If this is the client's only company *and* they have no documents, there is nothing left to hold the record up, so the client is deleted — and with them their **portal passwords** and their **whole statutory register**: directors, shareholders, registered office, share capital and every certificate attached to them. There is no undo.

The app asks you to **confirm it is you** before that one, and never before an ordinary removal — so being asked for a code is itself the signal that this is the destructive case. The message names it: *"delete a client, and everything held for them"*. The toast afterwards says **deleted**, not *removed*.

To keep the record and stop the client appearing on new invoices, set them **Inactive** instead. That is almost always what was meant.

4.4 Who audits them

Audited by (*practices only, under Compliance & deadlines*) records the client's auditors, chosen from the list of audit firms kept under Timesheet → Staff, tasks & rates. It is a fact about the client, so it is entered once and read wherever the auditors matter. Leave it at — *Not said* — where the accounts are not audited. A firm that has been retired is no longer offered, but a client already recorded against it keeps what it says.

5. Compliance

(Shown for regulated practices only.)

5.1 What is tracked

Filing	Date	Applies to
TD4	One date for every client, per year	Every accounting client
Summary Information Table	One date for every client, per year	Every accounting client
UBO register	One date for every client, per year	Every accounting client
Annual return (HE32)	From the client's made-up date	Clients with that date set

Filing	Date	Applies to
VAT return	From each VAT quarter end	Clients with VAT quarters set

Dates are **worked out from rules** (Settings → Deadline rules) plus each client's own settings, so correcting a rule corrects every date at once.

The rules are practice-wide, not per company, which is the one thing about this screen worth saying twice. Everything else in the app works on the company in the switcher; these do not. The screen itself warns: "These apply to every client of every company — edit one and every date on the Compliance tab changes with it." So a rule changed while looking at one company moves the dates of all the others.

Track deadlines from, on the same screen, decides how far back the lists reach. Left blank it means **1 January of this year**, so anything due before that is not listed at all. Where the real date is different (an extension, for example) it can be set by hand — for everyone (Settings → Deadline rules, on the card **Deadline dates — the same for every client**) or for one client (their Deadlines pane).

There are TWO floors, and the later one wins. Beside that practice-wide setting, each client's own **Client since** date floors that client: a deadline that fell due before the practice took them on is not listed for them, however far back *Track deadlines from* reaches. This is the right behaviour — the firm cannot have missed a filing for a client it did not have — but it is also the commonest reason a deadline somebody expects to see is not there. Before concluding that a rule is wrong, check the client's **Client since** on their Details pane.

Those rules and dates are **shared by every company**. The app works one company at a time everywhere else, so this is the exception worth knowing: change a rule while you have one company open and every other company's dates move with it.

5.2 The deadline list

Filters for period (30 days to 12 months), type, client and whether to include completed ones. Anything overdue is always listed. **Mark done** / **Undo** records completion. **Clients** on the top bar carries a badge counting what is overdue or due within 30 days (red if anything is overdue).

Track deadlines from (Settings) hides anything that fell due before the practice started using the app, and a client's own **Client since** hides anything that fell due before that client was taken on, whichever of the two is later (§5.1).

6. Currency

One screen, three things on it, kept because an invoice may one day be raised in a foreign currency. The rates are the European Central Bank's euro reference rates and nothing else — the app never invents one.

- **Convert** — an amount, from one currency to another, **on a date**. The date is the point of it: a conversion is only as good as the day it is quoted for.

- **Look up a rate** — every rate as at a date, with **Find a currency** to narrow the table. Each row carries **Recent days**, which draws that currency’s recent movement against the euro.
- **Fetch the latest rates** (*administrator*) — asks the Bank now rather than waiting for the nightly job.

There is no rate for a weekend or a public holiday, because the Bank publishes none: ask for a Sunday and you get the last published day before it, and the answer says so — “Rate published 29/09/2026 — there is no rate for 04/10/2026, so the most recent published day was used.” That published date is the one to quote, not the date you asked for.

Whether the rates can be trusted is said at the top of the screen, every time — not only when something is wrong. A converter that looks identical whether its figures are from this afternoon or from last March is a converter nobody can trust, so the ordinary case still carries a line: “Rates published yesterday (01/10/2026).”

Four things it says instead, each a warning:

The download failed	The last attempt did not reach the Bank, so nothing newer has arrived — with what the server got back, for whoever has to fix it. Said whatever the rates’ date , which is the point: the Bank publishes at about four in the afternoon, so a download that broke an hour ago sits behind yesterday’s rates looking perfectly ordinary
Nothing has tried	No attempt for over a day. The job runs hourly, so either it has stopped or the server has been off
The rates are old	Attempts are working but the newest is more than four days old — so either the Bank has published nothing for that long, or something is arriving empty. Four days covers a weekend and a holiday either side, which is an ordinary week here
Nothing at all	No rates have ever been downloaded. An administrator presses Fetch the latest rates

A failing download outranks old rates, deliberately: when both are true the failure is what is said, because it is the one that can be acted on. The rule is in `services/fx.ts` (`rateHealth`) and worked out on the server, so the screen and `/api/fx/status` can never disagree about it.

This module was called **Cashflow** and held the books — the ledger, the categories, the vendor name book, the budgets, the statement import, the reconciliation and the reports on spending. All of that is in the accounting application now.

Where the accounts went. For a short while this module also kept a list of *bank accounts*, with an opening balance. That list existed for the ledger and the reconciliation, and once those had gone its only remaining job

was naming which account a payment arrived in — while the practice was *also* keeping the same bank under Settings as the account an invoice prints. The same bank, typed twice, with nothing linking the two. So there is one list now:

- **Where clients send the money** — Settings → Company details (§11). This is what an invoice prints, what the invoice's *Where the client should pay* box offers, and what Record payment now names as the account the money arrived in.

Nothing was deleted from the database. Receipts recorded before the change still carry the old account and still show it — in the report builder's **Money received** source, and on the receipt itself.

7. Quotations

Two lines under **Tools: Quotations**, the list of them, and **New quotation**, which opens the writing screen. A quotation is not a document of account — nothing on it reaches the ledger, the VAT return or any income figure until the day it is accepted and turned into an invoice.

- **A new company's fee list is empty**, and there is nothing to tick until it is filled. The screen offers **Copy the list from...** — a picker of the practice's other companies with a **Copy** button — or **+ Add a service** to build one from nothing.
- Quotations are built from the company's **fee list** (Tools → Services & fees): tick services, adjust fees, set billing frequency, transaction allowances (`{ txn}`), an ad hoc hourly rate, an opening paragraph and notes. The preview beside the form matches the PDF exactly.
- A quotation can go to **a prospect** (someone not yet a client) or an existing client. Prospects are kept off the client list until the work is won.
- Numbering: QT-001-2026.
- **Sending** — email it (PDF attached) or record a copy given by hand. **Every copy sent is kept** (v1, v2...) exactly as the client received it.
- **Totals** appear only where every line is charged on the same basis (monthly fees and one-off fees are not added together).
- **Status** — Draft, Sent, Accepted, Declined, and **Expired** (worked out from the valid-until date).
- **Accepted** quotations are locked. Accepting writes the agreed fee onto the client record; a prospect can be turned into a client with one button, and a first invoice can be prepared **as a draft** from the agreed lines.
- **The first invoice can be raised only once**. A second attempt is refused — “An invoice has already been raised from this quotation.” Only lines carrying an amount are billed, and each arrives as “*service (fee suffix)*”, so a catalogue line of “€200 / month” becomes a draft line of €200 described “... (/ month)”. It is dated today, on the client's own terms, with the quotation's number as the reference.

- **That invoice is an invoice like any other, so a closed period refuses it.** Because it is dated today, what decides is whether *today* falls in a closed month — not the quotation’s date. A quotation accepted long after it was sent still bills into today. Where your account may reopen a period you are asked for a reason in the usual way (§12.1); the quotation itself is not locked and never is, because nothing on a quotation reaches the ledger until this button is pressed.
- Only a draft that was never sent can be deleted; otherwise mark it declined. Each row under **Sent copies** keeps its own **Open** and **Download** for the PDF as that copy went out.

8. Email (circulars)

- **Write a circular** — a formatted message with merge fields (`{{client}}` , `{{contact}}` , `{{company}}` , `{{vat}}` , `{{reg}}` , `{{today}}` , `{{year}}`) and attachments up to 25 MB.
- **Every client gets their own message, addressed only to them** — never a group address or BCC.
- The recipient list can be narrowed (active, accounting clients, VAT-registered, kind of entity) and adjusted by ticking.
- Before sending, the app shows: misspelt merge fields (the send is refused), clients with no email address (named, not silently dropped), blanks a field would leave, and clients who share an address (they get one copy).
- **Send test to me** first — a copy marked [TEST], filled in from a real client.
- The send states its count and is refused if the list changed in the meantime.
- **Saved messages** are reused as templates; **What has been sent** logs every circular and each recipient’s result.

Circulars go to the client’s main address (falling back to the invoice address). All email leaves through the company’s own mail server; there is no fallback.

9. Reports

All reports cover the company in the switcher.

Report	What it shows
Build a report	Any report you want, from nine kinds of record (see 9.1)
Revenue by period	Fees, VAT, reimbursements and totals by month, quarter or year
Revenue by client	The same, by client

Report	What it shows
Client statement	Every document for one client with a running balance, opening balance brought forward. ✉ Email it sends that statement to the client
Aged receivables	What is owed, by age (current, 1–30, 31–60, 61–90, 90+ days), plus clients in credit

Most reports export to **Excel** and **PDF**. On the client statement, ✉ **Email it** sends the statement on screen — the same client and the same period — to the address on their record. The figures are in the message itself, so it reads on a phone, and nothing is attached. An address that is not the one on their record is questioned before anything is sent.

9.1 The report builder

- **Nine sources:** invoices and credit notes, invoice lines, money received, costs, clients, deadlines, time recorded, quotations, emails sent.
- **Costs** is worth reading twice before trusting a total. It holds money laid out for clients, which is still recorded here, **plus every cost entered while this app kept the books**. Spending since then is in the accounting application and this source cannot see it, so a report over a recent period will look far too small.
- **Period** (this month to all time, or chosen dates) and **filters** (is, is not, contains, at least, at most, empty...).
- **A list of records** (choose and order columns, sort) or **Totals, grouped** (up to two groupings, adding up totals, averages, highest, lowest or counts).
- **Charts** — bar, line or pie.
- **Excel and PDF** export.
- **Saved reports** — the practice's, shared by everybody **who can run one**: that is, anybody holding *Can change* on at least one company, plus an administrator. Somebody with only *Can look at* access, or only a timesheet grant, does not see the list at all. The reason is that a saved report carries its **filter values**, and those are usually a client's name or an amount — so the list is open to the people who could run the report and read the figures anyway, and closed to those who could not. It is still always run against the company in the switcher, so the same report answers differently for different people. Only the person who saved one, or an administrator, may change or delete it; and a report whose author has **left the practice** belongs to an administrator rather than to everybody, which is what it used to amount to. **New, Save, Save as new...** and **Delete** do what they say; deleting removes the definition and nothing it reported on. A saved report keeps its named period, so "This year" is always the current year.
- Credit notes count as minus, so totals are net of credits. **The starting filters are not all "leave the drafts out"**, and they are worth reading before trusting a figure: invoices and invoice lines start at *Status is not Draft* and *Type is not Pro-forma*; **clients** start at *Active is Yes*, so inactive ones are excluded; and costs and receipts start at *Status is Posted*, which silently drops anything voided. All are visible and removable.

- **Limits:** 5,000 rows or groups (beyond that the result line says “showing the first N; the totals cover them all”, so totals stay right), two groupings, twelve filters, six totals. With two groupings the chart draws only the first. A report on **clients** has no period control, because a client is not dated.
 - Hourly rates and portal passwords are never offered.
-

10. Timesheet

Three lines: **My time**, which is everybody’s own; **Everyone’s time**, which is the whole company’s and is offered only to somebody who may see it (see *Who sees what* below); and **Staff, tasks & rates**, where the people, the kinds of work, the audit firms and the hourly rates are kept.

Three sections: **My time**, **Everyone’s time** and **Staff, tasks & rates**. What the module records is time **for a person**, and the thing worth understanding before anything else is that a person is not the same as a sign-in.

A person and a sign-in are different things

They are kept apart deliberately. A subcontractor or an outside auditor can have their hours recorded without ever being given an account; an account can exist that never records time. So holding a sign-in — an administrator’s included — does not by itself give the timesheet anybody to book against, and **Timesheet** → **My time** shows no form at all until the two are linked: *“Your sign-in is not linked to a staff record yet, so there is nobody to record time for.”*

Two things have to exist before any hour can be recorded, both on **Timesheet** → **Staff, tasks & rates**:

- a row under **People whose time is recorded**, with the person’s address chosen under **Sign-in (optional)** — that dropdown *is* the link, and it is left blank for somebody who will never sign in;
- at least one **kind of work** under **Kinds of work**. With none, the **Task** box on the record form is empty and nothing can be saved.

This is the trap named in §1.4: giving somebody the timesheet grant is half the job.

What an hour is worth

Rates live on the same screen, under **What an hour is worth**, and may be set per person, per service, or per client *and* service — the most specific one wins.

With no rate, time still records; it records at zero value. Not refused, which is the right way round: hours booked before a rate is agreed are kept rather than lost, and become worth something the moment somebody sets one. The card says so itself — *“No rates yet. Time will record at zero.”*

Whoever may see rates also gets **Rate (leave blank for the agreed one)** on the record form. Typing in it overrides the agreed rate for that one entry and nothing else.

Who sees what

Seeing and changing are two different questions here, and the table answers both because they do not have the same answer.

There are three questions, not one, and they do not have the same answer: whose hours somebody sees, whether they see what those hours are **worth**, and whose hours they may **change**.

	Sees whose hours	Sees the rates and values	May change
Super admin	everyone, every company	yes	anything
Somebody holding the company <i>to change</i>	everyone's, for that company	yes	anybody's hours; may record time for another person, move an entry to somebody else, and type a rate over the one worked out
Somebody holding the company <i>to look at</i> , and the figures****	everyone's, for that company	yes	their own hours only
Somebody holding the company <i>to look at</i>	everyone's, for that company	no	their own hours only — even with the timesheet grant as well
Somebody with only the timesheet grant (§1.4)	their own	no	their own hours only

Staff, tasks & rates is an administrator's screen alone, and so is the list of everybody whose time is recorded. The **Who** column on *Everyone's time* names the people appearing in the rows on screen, which is not the same as the practice's staff list: somebody who has recorded no hours against that company is not named to anybody but an administrator.

Why to look at no longer shows the rates by default. A company given to somebody to look at used to show them everything in it, rates included. That is right for a manager or a partner reviewing a company they do not key data into. It is wrong for the other person who gets the same grant — somebody inside one of the companies, given it so they can see what has been invoiced to their own clients. The hourly rate is what the practice charges for an hour, and it was reaching them. So the grant now carries the answer: tick **and the figures** beside *to look at* on **Settings** → **People & access** for the people who should have it (§1.4). Nobody

holding a company *to change* is affected — somebody who may alter the figures is not protected by being unable to see them.

A company held *to look at* does not become a company you may change by also holding the timesheet grant. Worth saying plainly, because it did not used to be true: the two grants together let somebody edit and delete other people's hours, move an entry onto a different person, and overwrite a rate. Each grant on its own refused all of that; only the pair let it through. Fixed on 2 October, and the person holding both is now confined to their own hours exactly as they would be with the timesheet grant alone.

What the form will and will not accept

- **Bill this time** is the switch that matters. Ticked, the **Client** and the **Service** stop being optional — chargeable time has to say who it is for and name a service, so that it can reach an invoice line one day. Left unticked, internal work, training and admin can be recorded against no client at all.
- **Period this belongs to** is what the work was *about*, not when it was done: the work on March's VAT return belongs to March. Optional, and it takes a month (2026-03), a quarter (2026-Q1) or a year (2026).
- An entry cannot cover more than one day — *"An entry cannot be longer than one day. Split it across the days it covers."*
- **From** and **To** are HH:MM and the end must be after the start. For work that ran past midnight, type how long it took instead of the two times.
- **Time that has reached an invoice is frozen.** Where **Edit** and **Delete** would be, the row reads **invoiced**, and the app refuses: *"This time has been invoiced, so it cannot be changed. Raise a credit note against the invoice instead."*

10.1 The clock that runs while a client's record is open

Opening a client's record starts a clock. Closing it, switching company, signing out or closing the browser stops it. What it leaves behind is a **measurement, not recorded time** — a line under **Time measured, not yet recorded** at the top of **My time**, saying which client and how long. It becomes time when somebody says what kind of work it was, and until then it is on nobody's bill and in no report.

That is not caution for its own sake. Every entry needs a kind of work, and a clock cannot know whether the hour was a VAT return, a phone call or reading an email — guessing would put made-up detail on a record a client is billed from. So the clock measures, and a person decides.

Two limits keep the measurements honest, and both are the firm's own figures:

- **Thirty minutes.** A record left open with nobody working in it stops counting half an hour after the last thing anybody did. Without that, the first Friday somebody forgets to close the app produces a fourteen-hour draft against one client — and the one nobody notices is the one that gets billed. A measurement cut this way is marked a **guess** on the strip, and the figure should be corrected before it is recorded.
- **Five minutes.** Anything shorter is dropped without being offered. Opening a client to find a phone number is not work, and a strip full of twenty-second lines is a strip nobody reads — which would cost the lines that matter.

A browser that crashes or a laptop whose lid is closed within a minute of opening a client leaves nothing at all. The app can tell the difference between a record somebody was working in and one that simply vanished, and it would rather offer nothing than offer half an hour that never happened.

Recording one. Press **Record...** beside the line. The form below fills itself in and locks the client, the date and the two times — those came from the measurement and cannot be edited here. Set the minutes if the figure needs correcting, choose the **Service** and the **Task**, check the period, and press **Record this time**. It then becomes an ordinary entry, under the ordinary rules: chargeable time still has to name a client and a service, the rate is still worked out on the server, and it can still be edited and deleted like any other. To record a different day or a different client, dismiss the measurement and type the entry by hand.

Dismissing one. **Dismiss** throws the measurement away. It asks first, and the practice keeps a note that it was dismissed and how long it was — an hour the app measured and somebody decided not to bill is exactly the hour that gets asked about later.

Only your own. You see the clocks from your own sign-in and nobody else's, and so does an administrator — the owner included. Two records cannot be timed at once: opening a client stops the clock on the one before, because nobody works on two clients in the same minute.

10.2 Time measured in the payroll app

The payroll application runs the same kind of clock while a client's payroll is open. When whoever was in there leaves it, that application asks them to write a line saying what they were doing — "*September payroll and the new starter*" — and nothing is handed over until they have. The time then appears on this strip like any other measurement, marked **payroll**, with their sentence under the client's name. Confirm it the same way; the sentence becomes the note on the entry.

It is fetched, not sent. This application goes and asks whenever somebody opens **My time**. So the payroll app being off, or the connection being down, means the time arrives a little later — never that it is lost.

Which client the time belongs to. The two applications share no records, so an employer in the payroll app is matched to a client here on the registration number, and on the tax code where there is no registration number. A name that is merely similar is never matched: *Kweller Ltd* and *Kweller Limited* are the same company to you and two different things to a database, and an afternoon quietly filed against the wrong client is worse than one nobody has filed yet.

Where nothing matches, the time is **held in the payroll app** and a question appears under the strip for an administrator: the employer's name and numbers, how many entries are waiting, and a box to choose the client. Answering it once files every entry waiting on that employer, and the pairing is remembered — nobody is asked about that employer again. The same happens where somebody has a payroll sign-in but no staff record here: the question names the address, and adding them under **Staff, tasks & rates** brings their time through on the next check.

Nothing is ever lost while a question is unanswered. Time that cannot be filed stays in the payroll app and comes round on every check until somebody deals with it.

11. Settings

Section	Contents
Company details	Profile, VAT quarters, kind of firm, services, payment accounts, logo, mail server, directors & shareholders, notes, currency, invoice prefix, business type, VAT mode; closing periods; the record of changes to figures
Products	The product catalogue (goods companies): add, Edit and Delete . A product is the starting point for an invoice line, so changing its price or removing it leaves invoices already written exactly as they are
Deadline rules	Track deadlines from; the five rules; dates shared by every client
Defaults	Default currency, VAT % and payment terms; the app's own mail server
Teams (<i>administrator</i>)	Teams of this company: who is in each, and which clients it looks after. Nobody is narrowed until a client is put on a team (§1.4)
People & access (<i>administrator</i>)	People, their companies and access, who may record time; the key protecting passwords; the audit log
Backup & restore (<i>the owner; alerts and the file check, an administrator</i>)	Encrypted backup download, restore and the nightly emailed backup are the owner's alone (§12.3); the alert settings and the check of the uploads folder beside them stay an administrator's
Linked applications (<i>administrator</i>)	The keys the firm's other applications read this one's issued invoices through — made here, shown once, given a date they stop working and the companies they may read, changeable, revocable (§12.6)
Your sign-in	Your password, second factor (authenticator app, passkeys, recovery codes), where you are signed in

12. Safeguards, security and backups

12.1 Closing a period

Settings → Company details → Closed up to. Once closed, nothing dated on or before the date can be added, changed or deleted — invoices, credit notes, receipts, money laid out for a client, and **recorded time**.

Time counts, and it counts both ways. An hour already booked into a closed month cannot be corrected, moved or deleted; a new one cannot be booked into it either, and a measured clock cannot be confirmed into it — which matters because a clock run a fortnight ago is confirmed today. Hours are dated and carry a rate, so they are the same kind of record as an invoice.

However an invoice is raised, including from a quotation. *Raise an invoice* on an accepted quotation makes a real invoice — its own number from the same counter, its own VAT working — so it is refused by a closed period exactly as the Invoices screen is. The date that matters is **the invoice's**, which is today: a quotation sent in March and billed in May is a May invoice, and the quotation's own date does not come into it. Quotations themselves are never locked, because nothing on a quotation reaches the ledger — see §7.

What is **not** locked, deliberately: the hourly rates under **Timesheet → Staff, tasks & rates**. A rate is copied onto an entry at the moment the entry is written, so changing or removing one cannot revalue an hour that has already been recorded — there is nothing closed for the lock to protect. (Which also means the reverse: correcting a rate does *not* correct the hours already booked at the old one.)

Who may do it: the owner, and this company's approver. The card appears to the owner in every company, and to whoever was given **that** company as *to change, and to close months* on *People & access*. Nobody else sees it — **not an administrator**, however much else they run, unless they are also made that company's approver. Closing a month is the one act this application asks a named person to stand behind, and running the practice is a different answer from standing behind one client's books in a month somebody else prepared.

So an approver of one company does not see the card on another, and **with no owner and no approver granted, nothing can be closed or reopened** — which is the right direction for a lock to fail in, but worth knowing when the box is not where you expected it.

Until 5 October this was an address: the card appeared only to the account whose email matched `ADMIN_EMAIL` in the server's `.env`, which meant one answer for the whole practice and a power that moved with a string. See §1.4.

Re-opening has two shapes:

- **For one save.** Where your account may re-open, saving or deleting something inside a closed period does not simply refuse: a window headed **Reopen a closed period** asks for a sentence, naming the date the books are closed up to and saying that the reason is kept on the audit log with the record. Give one and that single change goes through; **the lock date does not move**, and everything else stays shut.
- **For good. Reopen everything** on the *Closed up to* card clears the lock entirely — or set *Closed up to* to an earlier date, which reopens everything between and asks for a reason the same way.

Where that question is asked, and why there is only one of it (PROMPT-1)

The question is asked in **one place in the whole application**: `api()`, the function every screen sends its writes through. Sixteen-plus write paths can land in a filed quarter — an invoice, a credit note, a receipt, money laid out for a client, recorded hours, an invoice from an accepted quotation, two bulk runs — and asking each screen to handle the refusal would mean fifteen getting it right and the sixteenth, written next year, not. The server refuses with `needsOverride`, `api()` asks, and sends the same request again with the reason attached.

Three properties of that window are worth knowing, because each was once wrong:

It is the app's own window, not the browser's

It was a `window.prompt` until 8 October 2026. A browser is allowed to switch those off — Chrome offers “prevent this page from creating additional dialogs” on the second one of a session — and after that the question could not be asked or answered, leaving the closed-period refusal with no way through it and nothing to say why

A reason too short keeps the window, and keeps the work

The old prompt showed a toast and discarded the save: you had to find the screen, fill it in again and press again, and the toast saying why had faded before you got there. The window now stays open and says so in the box, with the characters still needed counted underneath while you type

The minimum length is the server's number

`MIN_REASON` in `services/period-lock.ts`, sent to the browser on the refusal. The window carries no number of its own, so the box can never accept a sentence the server then refuses — the same choice the “password for somebody else” window makes about the password policy, after that floor was once written down twice

The same window serves the **lock date** itself, which until 8 October had a second `window.prompt` of its own, on Settings → Company details, with a second copy of the ten characters in it. That screen now sends the change and lets the one question ask.

Somebody who may **not** re-open is offered no box at all. A window that will be refused whatever is typed in it is worse than a sentence saying to ask whoever keeps the books, and the refusal does not pretend a better reason would have helped.

12.2 Signing in

- Email and password (at least 12 characters), then a **second factor**: a code from an authenticator app, a passkey (this device's fingerprint, face or PIN), or a one-time recovery code.
- **A second factor from the first session.** A new account signs in with its password and reaches one screen: the one that sets an authenticator up. It can do nothing else until it has one, and there is no grace period and no way to turn that off. The screen is in the sign-in card, before the app opens, in the same three steps as the accounting application's. (MFA-1)
- **Setting one up starts with a QR code.** The screen shows a square to scan with the authenticator app, and the typed key underneath it for anybody whose camera will not oblige. The square is drawn **by this server** and is part of the same reply as the key — it is not fetched from one of the free QR services, because what a QR code for an authenticator encodes is the shared secret itself, and a service would keep it in its access log beside the account's email address. (QR-1)
- Sessions end after one idle day and after 14 days in any case. **Sign out everywhere else** ends all other sessions.

Wrong passwords, and being shut out

Nobody is told this anywhere on the screen, so it belongs here.

Four wrong answers on one email address are free — mistyping a password twice is an ordinary morning. The fifth starts a lockout, and the lockouts lengthen: **60 seconds, then 5 minutes, then 15 minutes**, and 15 repeats. The refusal says how long is left (*"Too many attempts. Try again in 15 minutes."*). An address with no failure for an hour is forgotten entirely, so yesterday's fumbling is not held against anybody.

Four things worth knowing about it:

- **It counts the email address.** Somebody guessing at your address slows *your* sign-in down, not theirs. (This used to say "not the machine", and that has changed — see *And one place working through a list* below.)
- **The change-your-own-password screen shares the same counter**, because guessing there and guessing at the sign-in screen are the same act against the same account.
- **Codes have a counter of their own.** A run of wrong second-factor codes locks the codes after a handful of tries, and giving the right password does not clear it — which is the whole point: the password being right is what makes a run of wrong codes worth noticing.
- **A lockout now survives a restart**, since 11 October 2026. It used to be held in memory only, so every deploy — several a week — forgave every lockout in progress. Somebody working through a list of your addresses got a clean slate for free each time. The lock itself is now written down and read back when the server starts; the *guesses* in between are still only counted in memory, so nobody outside can make the server write by guessing.

The lockouts are deliberately short. Somebody who has genuinely forgotten their password should not lose the afternoon to it — and the way back in is an administrator sending them a link (\$1.4), not waiting.

And one place working through a list

The counter above is per email address, which means it cannot see the commonest shape of attack at all: **twenty wrong passwords across twenty different accounts locks nothing**, because no account reached five.

Since 11 October 2026 the app counts wrong sign-ins **per place** as well — the address the request came from. At **twenty wrong in an hour**, from one place, across any accounts:

- **every further attempt from that place is held back**, half a second at first and growing to five seconds. A sprayer is reduced to one guess every few seconds, which is useless to them;
- **you are emailed, once**. Not once per guess — once per run, however long it goes on;
- **and the line goes on the register** as *Many failed sign-ins from one place*.

Nobody is locked out by this, and that is deliberate. An office shares one address, so locking it would shut out everybody sitting at it — which is exactly what an earlier version of this did, and why the review of 2 October took it out. Somebody at that desk with the right password still signs in normally while a stranger is being slowed to a crawl.

A successful sign-in does not forgive it. That was the other half of the old fault: any honest sign-in anywhere cleared the shared counter, so an attacker got a fresh run of guesses for the price of somebody else’s morning.

12.2a What the app tells you about your own account

Five things about an account already raised an email — a lockout, a second factor added or removed, a password changed, an account created — and every one of them went to the practice’s **alert address** and to nobody else. That is the wrong way round for the case they exist for. Somebody who has your password uses it, and then makes it theirs; the alert address is read when somebody thinks to read it, and the only person who can tell at a glance that a sign-in was not theirs is you.

So they now also go **to the person whose account it is**:

When	What it says
A sign-in from a browser the account has not used before	Which browser, and that it has not been used before
A password changed	Whether you changed it or an administrator did
A second factor added	That one was added, and from where
A second factor removed, or cleared for you	Whether you removed it or an administrator cleared it
Four wrong passwords, then a fifth	That the account was locked, for how long, and that the guessing did not work
The address on the account changed	What it changed to, sent to the old address

When	What it says
What they may do changed — their role	What it became and what it was, and that their sessions ended
Their sign-in was switched off	That it was, that their work is not deleted, and to telephone rather than reply

The last two are new on **12 October 2026**, and they are the pair that had been missing longest. Both acts already ended the person's sessions and both already wrote a line on the register — but the register is what you read afterwards, and neither told the person it happened to. A member of staff found the practice's screens gone with no explanation; somebody made an administrator had been handed every stored password of the clients they reach, and nobody could have told them.

The role message is deliberately two-sided. It says that if they were not expecting it — and *especially if it gave them more than they had* — it is worth a telephone call today rather than a reply. A promotion nobody asked for is the case that matters, and it is the one a message worded as housekeeping would be read past. It goes only on a change to the **role itself**: marking somebody as a client's own person, or limiting an administrator's reach, also ends their sessions and sends nothing, because a mail saying "your role changed" when it did not is how people learn to ignore the next one. It is not sent to you for your own account.

The switch-off message goes to an address that can no longer sign in, which is the point rather than an oversight — the same reasoning as the address-changed message above it. Somebody who has left should be told their access ended; somebody who has *not* left needs to know within the hour that it did, and the mailbox is the only channel left. It says plainly that nothing they did has been deleted, because that is the first fear, and that the password link in their mailbox has stopped working. Pressing the button on an account already switched off sends nothing, and **letting somebody sign in again sends nothing at all** — it restores what they had, and a message saying "you can sign in again" to somebody who never knew they could not is noise.

The two voices are different on purpose. "Your password was changed" is a receipt: you did it, there is nothing to do, and the message exists so that the act leaves a trace somewhere you will see. "Your password was changed by an administrator" is a warning, and somebody who did not ask for one needs to read it within the hour — so it says who did it in the subject line, which is all anybody reads on a telephone, and it does not end with the reassuring sentence.

There is deliberately **no advice beyond "tell whoever looks after the app"**. This application has no self-service lock-my-account button, so a message ending in one would be worse than saying nothing.

How a browser is recognised

A **long-lived cookie carrying a random id**, and nothing else — no screen size, no font list, no header soup. Fingerprinting identifies a browser whether or not it wants to be identified, and the only question here is *have I seen this one before*, which a cookie answers honestly. The cookie's value is never stored: its SHA-256 is, the same reasoning as the password links and the integration keys, so a copy of the table is not a set of cookies that would pass as known browsers. The rows go when the account goes.

The table **does travel in the nightly backup**, which was a decision rather than an oversight. Nothing in it lets anybody in, and the alternative costs something real: a restore would leave every familiar browser unfamiliar, and the first week back would be a wave of *new sign-in* messages about people's own laptops — which is how a practice learns to ignore them.

Three consequences, all of them accepted:

- **The first browser an account ever signs in from raises nothing.** Otherwise everybody's first ordinary sign-in would arrive as a security message, which teaches a practice that these messages mean nothing.
- **Clearing cookies, a private window, or a second browser on the same machine each read as new.** A notice too many costs a moment's thought; a notice too few is the thing this exists to prevent.
- **A browser is mentioned once.** The cookie lasts over a year and is renewed on every sign-in, so the laptop used once at year end is still recognised.

What it will not do

- **It never writes to an address that is not an account.** The lockout counter counts the address whether or not anybody owns it — deliberately, so that a guessing run is visible as a run — so sending on every lock would mail whatever was typed into the box, from the practice's own address, and would answer *is this an address here*, which the sign-in screen refuses to answer by any other means.
- **An invitation raises nothing.** Somebody's first password on a new account is not news to them, and starting a job with a security warning about yourself is worse than silence. A **reset** does raise one: a link that was emailed has been used, and if it was not used by them this is how they find out.
- **It can never fail a sign-in.** The messages are sent without being waited for, and a practice whose mail server is not set up gets no notices, no errors and a line in the log saying why. A sign-in that waited on a mail server would be a worse application than one that occasionally fails to send a message about a sign-in.

They go through **the app's own mail server** (§12.4), not a company's — a message about your sign-in is not from one of the practice's clients.

12.3 Backups

Whose they are (BAK-4)

Taking a backup and restoring one are the OWNER's, not every administrator's — since 8 October 2026, and in line with Payroll and Accounting. An administrator sees the card saying so, and no buttons.

The reason is the restore rather than the download. **A restore does not merge, it replaces** — including the **users** table, which is where every “the owner alone” rule in this application lives. So an administrator could export the practice, change one word in the file and restore it, and be the owner. In practice two words: the database allows exactly one owner, so the same file has to demote the real one, which it may as well — being the owner is worth little while the previous one can put it back. The second security review found it, and the end-to-end test that reproduced it watched the owner come back as an administrator with their session gone.

The nightly schedule and **Send one now** went with them, because a backup somebody else can trigger is a copy of the whole practice leaving the building on their say-so.

What stays with an administrator on that screen: the **alert settings** and the **check of the uploads folder against the records**. Neither takes a copy of anything.

How they work

- **The nightly backup** is emailed automatically, encrypted. It goes through the app's own mail server, so **if that is not set up it silently does not happen** — and it is the only copy of the practice that leaves the server. See 12.4.
- **It is “after 02:00”, not “at 02:00”.** The server stops the application when nobody is using it, so there may be nothing running at two in the morning. The job asks itself once at every start and once an hour whether today has had a backup, and takes one if not — so on a quiet night the backup is taken when somebody first opens the app the next day, rather than not at all. The card's *Last one ...* line is how you check; **today** there means today, by the calendar.
- **If the backup fails, you are told twice over.** The notice goes out by email like any other alert — but it leaves through the same mail server the backup does, so the commonest failure of all takes out both. So a failed backup also puts a **banner across the top of the screen** the next time an administrator signs in, and that banner says whether the email got out as well. The failure, the reason, and whether anybody was told are all on **Settings → Backup & restore** and on `/api/diagnostics` too.
- **How big it may be.** The nightly file is emailed, and a mail server will not take an attachment past about 18 MB — so past that the app stops trying and says on the card that the practice has outgrown email as a backup channel. Restoring accepts rather more (32 MB), so anything the app is willing to send is something it will take back. The card shows the last file's size against those ceilings.
- **The two passphrases are different.** The nightly file is encrypted with the passphrase set on the server (`BACKUP_PASSPHRASE`); restoring one asks for it. The file you get from  **Download a backup** is encrypted with the passphrase you type at the time. Neither opens the other.
- **Download a backup** (administrator) — encrypted with a passphrase of your choice (at least 12 characters). Without the passphrase it cannot be opened.

- **Restore** replaces the data with the backup's; a copy of the current state is saved first. That copy is written on the server under `storage/tmp/pre-restore/`, named `before-restore-<timestamp>.json`. Only the **last ten** are kept, and **none for longer than thirty days**. Since 9 October it is **locked under BACKUP_PASSPHRASE** — the same passphrase that opens the nightly backups, and deliberately *not* the passphrase of the file that was being restored. To use one, restore it back through **Restore from a file...** like any other backup and give it that passphrase.

And from 12 October it is locked even where there is no passphrase. On a server with no `BACKUP_PASSPHRASE` the copy used to be written **in the clear**, and the server said so at startup. The reasoning was that an undo copy nobody can read is better than none at all — true, and the wrong pair to choose between. This is the one file in the application that nothing outside the application ever opens: it is not emailed, and no screen downloads it. So where there is no passphrase it is now locked with a key derived from the **server's own secret**, and the Restore screen opens it without asking you for anything. It cannot be opened anywhere else, which is the trade: it is an undo for this installation and not a copy you could carry to another machine. Setting `BACKUP_PASSPHRASE` is still the arrangement to prefer, and the startup line says which of the two you have.

Why that passphrase and not another, and what changed on 9 October. You reach for this copy in a panic, and the likeliest reason you are on that screen at all is that a file or a passphrase went wrong — so keying the undo to the **incoming file's** passphrase would lock it in exactly the case it exists for.

`BACKUP_PASSPHRASE` is held by the practice, is nothing to do with the file being restored, and cannot be worked out from the machine, so a stolen copy of that folder is worth nothing while the undo stays reachable. The server's own key is weaker than that — it sits in the `.env` beside the file, so anybody who can read the folder can read the key — which is why it is the fallback and not the choice. What it is better than is the clear.

The review's other suggestion was to delete the copy after a successful restore, and that was not available: a restore that FAILS changes nothing, because it runs in a transaction. This copy exists for a restore that SUCCEEDED on the wrong file, which is the one mistake here that nothing else undoes. Deleting it on success would remove the safety net at the moment it becomes the only one.

If `BACKUP_PASSPHRASE` is not set on the server, the copy is still written — in the clear, exactly as before — and the server says so in its startup log. That is deliberate: an undo copy that is not written at all is worse than one that can be read by somebody who already has the disk. It is also the same single setting that switches the nightly backup on, so the fix is one line in the `.env`.

- Uploaded files (logos, certificates, attachments, archived quotation PDFs, supplier invoice scans) are **not** in the backup and need copying separately. **The cards on this screen are not headed what this handbook used to call them**, which matters when somebody is being told where to look: the nightly backup is **The backup nobody has to remember**, and the alerting is **Being told when something happens**. On the latter, ☒ **Send a test alert** is the only way to prove alerting works end to end. Where no address has been set the button is **greyed out** rather than refusing — so the server's "Nothing is configured" is not something you will see; what you see is a button you cannot press, and the fix is to set `ALERT_EMAIL_TO` on the server.

- **Alerts** (Backup & restore) — the short list of things worth being told about by email, with the address they go to and, beside each, when it sends. Some send every time; others send the first and then stay quiet for a while, so one bad afternoon is one email rather than two hundred. The column says which, and how long the quiet lasts. More than one address can be told: they are set on the server, separated by commas.
- **A restore and the uploads folder have to go back to the same moment.** The server dumps the database at 02:00 and archives the files at 03:00, so the files are always at least as new as the database, which is the safe direction. After restoring the database to a given date, ask whoever looks after the server to restore `storage/uploads` from the archive of that same night.

The mailbox that hid its own failure (MAILBOX-1)

New on 13 October 2026, after it happened.

What happened. The nightly backup bounced for several nights with `452 Mailbox full`. The application did everything it was built to do: it recorded the failure, wrote `backup.failed` on the audit trail, and sent an alert about it.

It sent that alert to the mailbox that was full.

So the one message whose whole job is to say *your backups have stopped* was refused by the same server, for the same reason, every night. AUDIT-2026-09-28 §6 had already found the near-miss version of this and fixed half of it — `noticeSent` records whether the warning got out — but recording it is not the same as anybody reading it, and nothing anywhere said the two addresses were the same place.

Why they were. `alertAddress()` falls back to `BACKUP_EMAIL_TO` when `ALERT_EMAIL_TO` is unset, and that fallback is right: it is what makes alerting work at all on a server where nobody has configured a second address, and “ask somebody to configure a second address before any alerting happens” is how alerting does not happen. The fallback is kept. What was missing was anybody being told they were on it.

Three things now.

- **It is said, in four places.** A line on the Backup & restore card; `alerting.sameMailboxAsBackup` and its sentence on `/api/diagnostics`; a `warn` in the server’s own log at startup, immediately after the line naming the backup address — so a reader told where the backup goes is told in the next breath that that is also where they would be told it had stopped. The log matters most of the four, because it is the only one that does not depend on mail working.

And the sign-in banner, **only once it has actually bitten**. It is not a standing banner: one about a configuration choice, shown every session with nothing wrong, is one people learn to dismiss unread. When a backup has failed AND the refusal was the mailbox AND the alert goes to that mailbox, the banner adds that this notice is very likely the only one they will get.

All four read one sentence, from `mailboxShare` in `services/mailbox-full.ts`. The browser never judges whether two addresses are the same mailbox, and the alert address is never sent to a browser at all.

- **A full mailbox is retried.** Up to three times that day, and *only* for a mailbox that is full. Every other refusal still counts as the day's single attempt, which is the rule that stops a mail server with a wrong password being handed the whole practice twenty-four times a day — each attempt reads every table and holds the result in memory. A full mailbox is different in kind: the file was fine and the practice was fine, and somebody emptying the mailbox at nine in the morning fixes it. Under the old rule that day still had no backup, which is the day somebody would most want one.
- **Which addresses are one mailbox is decided properly.** Name `<a@b>` is unwrapped, the domain is lower-cased, and a `+tag` is dropped from the local part — `backups+nightly@` and `alerts@` sharing a base are one mailbox and one quota. The judgement leans towards warning: a false warning costs a line somebody reads once, and a false silence costs this finding again.

The 552 that is two different answers. `mailboxIsFull` decides whether the scheduler rebuilds the entire practice an hour later, so it is worth saying what it will not do. The code 552 is used both for *the mailbox is full* and for *that message was too big*, and those want opposite answers — one is worth retrying and the other never is, because the file that was refused is the file that would be sent again. So a reason naming the SIZE is not treated as a full mailbox even when it carries a 552, and the words decide the rest. Plenty of servers send the words and no code at all, which is why the match is on phrases as well as codes.

What is still a person's job. Nothing here empties a mailbox or changes an address. The fix is `ALERT_EMAIL_TO` in the server's `.env`, pointed somewhere else — ideally at a different provider, so one outage cannot take both. Until it is set, the app says so every time it starts.

12.3a Checking the files against the records

Settings → Backup & restore → Uploaded files against the records → Check the files.

The database holds file *names*, the files themselves sit in a folder on the server. Nothing keeps the two in step, and when they come apart it shows up as a blank logo or a certificate that will not open — in whichever record somebody happens to open next. Two company logos were missing for weeks before anybody noticed.

The check runs both ways, because they are different failures:

- **Records naming a file that is not there** — a broken record. Listed with the company or client it belongs to and what the file was, so it can be uploaded again or the folder copied across from wherever it still exists. One file can break several records: every document keeps the **letterhead it was issued under**, so a logo that goes missing is one line saying “and 312 more” rather than 312 lines. Where documents are among them, **Put the letterheads right** stamps the company’s current logo on every document naming a letterhead that is not on the server, in one press. A document whose mark is merely *older* is left exactly as it is — that is not a fault, it is why each document keeps the logo it was issued under. Every change is recorded in the audit log, one line per document.

One press has been used on this practice, in September 2026, on **twelve invoices** whose two letterhead files had gone from the server. It is worth knowing about, because the press is one-way: it overwrites the name of the file each document had pointed at, and there is no undo. On **11 October 2026** it was decided not to go after the two old files, so those twelve keep the company’s current logo permanently. If one of them is ever reprinted and the mark does not match a paper copy from September 2026, that is this decision and not a fault; the audit log still says which twelve they are.

- **Files no record names** — usually harmless (an upload whose record was deleted afterwards). **Nothing is deleted by this screen**, deliberately: there is no undo on that folder. A logo that has been *replaced* is not one of these: its file stays on disk because every invoice issued under it still names it.
- **Stored values that are not a file name** — one for whoever looks after the server.

Separately, the app writes a line to the server’s log the moment it is asked for a file that is not there, naming the record — so the next one surfaces the day it happens rather than the day somebody notices. It also **emails whoever keeps the server**, at the address alerts go to, once per file per day: the server’s nightly archive keeps thirty nights, so hearing the same day is the difference between pulling the file back and not having it. Once a day rather than once a request, because a screen left open on a broken record asks again every time it redraws.

12.4 Email

There are **two mail servers**, and they are separate on purpose.

The company’s (Settings → Company details → **Sending mail as this company**) sends everything a client receives: invoices, statements, quotations, circulars. Each company has its own; one with none set up sends nothing and says so, and never borrows another company’s.

The app’s own (Settings → Defaults → **Mail from the app itself**) sends everything that is about the practice rather than about a client: sign-in codes, every alert, and the nightly backup. Kept apart so that a security code never arrives dressed as one of your companies, and so a mistyped company setting can never stop you getting in — which is a thing it might be reporting.

Passwords are stored encrypted and never shown again.

Who may change either of them: an administrator, and only after confirming it is you. Anybody who may work in a company can see where its mail goes, and can send a test. Changing it is an administrator’s, and asks for a code at the moment — the same pair as replacing a client’s portal password, and for the same reason. The

save writes the server, the username, the from-address and a password; from that moment every invoice, statement and reminder that company sends leaves through them, and no screen looks any different afterwards. It is recorded either way.

When the app's own is not set up, three things fail quietly — the nightly emailed backup (the only copy that leaves the server), every alert whatever address it is pointed at, and emailed sign-in codes. **Both cards now say so.** Where the app's own mail is unset, a red block at the top of each names it and points at Settings → Defaults. Before that they reported only their own half — one the address it would send to, the other whether an address is configured — so `ALERT_EMAIL_TO` and `BACKUP_EMAIL_TO` could both be right, both banners green, and nothing had left the building for weeks. Even so: **a practice that has never pressed** ☒ **Send a test alert does not know whether it has alerting.** A warning proves a fault; only a message arriving proves a channel.

The card carries its own history — *Never tested*, or *Last proved to work ...* — and a **Send test** beside the save button. Both cards now check two things themselves, rather than leaving them to be discovered when a test fails in the mail server's own words:

- **A web address in the server box is refused on save**, naming the host name that should have been typed. It is what a hosting control panel shows, so it is what gets pasted, and it can never resolve.
- **A port and an encryption that disagree** — 465 with STARTTLS, 587 with TLS from the start — raise a note under the fields as they are chosen. A note, not a refusal: a mail server may offer encryption on whatever port it likes, and a screen that turns down a working configuration because it has not met it is worse than one that asks.

12.5 Records of what happened

- **Audit log** — on screen it is the card headed “**What has been done, and by whom**”, under People & access. Who did what, kept for two years, cannot be edited — and since 12 October 2026 **sealed**, which is §12.5d below and is what makes “cannot be edited” a claim rather than a hope. Also from 12 October, thirty acts **refuse to happen** if their line cannot be written — §12.5e. It records acts rather than figures: the line says who, what and when, and never the value, which is what the record of changes below is for. The timesheet puts four things on it that used to leave no trace at all — **an hourly rate being set or removed**, and the **permission to record time** being given or taken away. A rate is what turns an afternoon into a number on an invoice, and the line names who the rate was for and from what date; it does not name the amount, which is on the rate card itself.
- **What changed on the figures** (Company details) — the last changes to invoices, receipts, costs and recorded time, with before and after. Four kinds of record, and that is the whole list as of 14 October 2026 (§12.5i). Lines written under a kind this app no longer has stay and still read, because the log stores the SENTENCE rather than a pointer to a screen — which is the property that let two kinds be retired without touching the table.

An earlier edition said it also held “supplier invoices and supplier payments recorded while this app kept the books”. Neither was ever written, and a supplier payment could not be: its table went under migration 0059. See §12.5i.

- **Email log** — every email attempt with the server's own response.

12.5a The key that protects saved passwords

Clients' portal logins — TAXISnet, TaxForAll, the RoC code, CyLogin, SISWEB, Ergani — and the mail passwords are encrypted before they are stored, so a copy of the database does not open them. This card, under People & access, is about **which key** does that.

With no key of its own set, they are encrypted under **the same secret that signs the sign-in cookies**. That works, and it ties together two things that should not be tied: replacing that secret is an ordinary piece of housekeeping, and doing it would make every saved password unreadable at a stroke — discovered one client at a time, weeks later. Giving them a key of their own separates the two, and is done once.

The key itself lives in the server's environment and not on any screen, for the same reason the backup address does: a secret that can be changed from a browser is a secret one stolen session can change.

The number to act on is the count of unreadable values. Above nought means the key in use does not open what is on file. Nothing else can tell you: the screen that shows a password draws an unreadable one exactly like an absent one, as an empty box.

12.5b Is the database still what the app is written against?

Every screen in this application is written against a table and a set of columns. A database that is missing one of them does not fail politely — it fails wherever that table is read, and nowhere else, which can be a corner nobody visits for a week.

This happened on 6 October 2026: `password_links` was missing while the migration journal recorded it as applied. Two things had been broken for five days — the nightly backup, and **Email a password link**, which is the way somebody locked out gets back in — and the only visible symptom was the backup failing with a reference number, because the error came from the database driver and those words are deliberately never published (§12.5c).

So the question is now asked directly, and answered in **names**:

- **The nightly backup asks before it builds anything.** Where something is missing it refuses with the names in plain words — *"The database is missing the table ..."* — which appears on Backup & restore where somebody is already looking.
- `/api/diagnostics` reports it under `schema`, with `ok: true` when the database is whole and the missing tables and columns listed by name when it is not.

Columns as well as tables, and that is the half that matters most: a missing column leaves the table count correct and breaks exactly as hard, because every query this app writes names the columns it wants.

It is deliberately **not fatal**. A missing column should not take the practice off the air: most of the application would work, and a server that refuses to start is a worse morning than one that says what is wrong with it.

Nothing on any screen can repair this. It is a migration that is recorded as applied and did not take effect, and it is fixed by whoever looks after the server, with a migration rather than by hand so that what is rebuilt matches the code.

12.5c Why some failures give a reference instead of a reason

A message from the database names the statement that failed and the values bound into it — which for this application’s queries means client names, addresses and invoice lines. Publishing that on a screen, in a reply, or into a file that is kept, would put client data wherever the error went.

So an unexpected failure answers with *“The request could not be completed... (reference ...)”*, and the whole of it goes to the server’s log against that reference. Nothing is lost; it stops being published.

The cost is that the reference is useless to the person reading it, which is why every failure the app can **recognise** is worded properly instead — a closed period, a missing mail server, a missing table (§12.5b). A reference is the last resort, and seeing one twice for the same action means the log should be read.

12.5d The trail is sealed, so a changed line shows

Security audit, step 6, first point. The trail answers *who did this, and when* — and an answer anybody with the database could quietly rewrite is not much of one. Nothing in the application edits or deletes an audit line, and that was already true; what was missing is a way to tell whether something OUTSIDE the application had.

What happens now. Every line, a second or two after it is written, is given the next number in a chain and a **fingerprint** made from itself and the fingerprint of the line before it. Change a line afterwards and its fingerprint stops matching. Take one out and the next line no longer points at the one before it. The check walks the whole chain and either says it is intact or names the first lines that are not.

The fingerprint is keyed, which is the part that matters. A plain hash could be recomputed by whoever changed the line — chain and all, in one pass. This one is an HMAC under a key derived from the server’s `SESSION_SECRET`, which lives in the server’s settings and **not in the database**. So somebody who can reach only the database cannot make a changed line match again. Each line also records *which* key sealed it, as a short fingerprint of the key, so a trail restored from another installation reads as *“sealed elsewhere, not checkable here”* rather than as altered.

Three things it deliberately does not claim.

- It does not stop anybody changing a line. It makes the change **visible**. Nothing short of taking the database away from whoever looks after it could do the first, and the trail has to live in the database to be useful.
- A line is sealed a second or two **after** it is written, not as it is written. Chaining at the moment of writing would make every write wait for the one before it, under a lock, on the hot path of every act in the app. Those seconds are the one window in which a change would not be caught; the check says how many lines are waiting, and any waiting more than **five minutes** means the sealing is not running, which is itself reported as not intact.

- A gap the **two-year keeping rule** left is not tampering. Each line records when the line before it was written, so a gap whose missing line was old enough to be swept is the rule at work, and a gap where it was not is reported.

The one thing the chain cannot see by itself

Lines taken off the **end** leave a chain that is still perfect, only shorter — and nothing inside the database can tell, because whoever removed the lines could remove the record of how many there were.

So once an hour the newest seal — its number and its fingerprint, nothing else — is written to a small file on the server's disk, in `storage/audit-anchors/`, beside the uploaded files. The check then also asks that every seal written down there is still on the trail, unchanged: **lines removed from the end are caught**, an hour after the fact. Two days' worth are kept.

That folder is in the host's nightly archive of the application folder, so the seals survive losing the database — which is the point of their being outside it — and the deploy deliberately never touches `storage/`, so a release does not wipe them. The file is named for the database it belongs to, so a second installation or a test database on the same server never reads another's.

A restore does not disturb any of this, and that is worth saying because the accounting application has to start its seal file again after one. `audit_log` is the one table a backup deliberately does not carry (§12.3), so restoring a file leaves the trail exactly as it was — and the seals go on protecting it. Restoring the **database itself** from a dump is the other case: the lines really are gone, and the check reports them as removed, which is the right answer.

Where to read it

Settings → **Security events**, from 13 October 2026 — §12.5f below. The state of the trail is the first thing on that screen, in words, for the owner and every administrator.

It is also still on `/api/diagnostics` under `auditTrail`, in more detail and for whoever looks after the server: whether it is intact, how many lines are sealed, how many are waiting, the number of the newest line, and the first twenty problems if there are any. The number of the newest line is worth noting down from time to time — next time it should be higher.

The first time the server starts after this, every line already on the trail is sealed in one pass, in the order the lines were written. On a practice with a few thousand lines that is a second or two, and it happens once.

12.5e Thirty acts that refuse to happen unrecorded

New on **12 October 2026**, the third point of the sixth step of the security audit.

What it was, and why it was right

Writing an audit line used to **swallow** its own failure. If the database would not accept the line, the act went ahead and nothing was said. That is deliberate and it is still the right answer for almost everything on the trail: a sign-in that could not be logged is still a sign-in, and a document emailed is still emailed. Turning a logging

problem into a refusal would stop the practice working over something that is nobody's fault, and reporting a completed act as a failure is a lie that gets the act done twice.

Where it was wrong

It is the wrong answer for a short list where **the record is the act's condition rather than a note about it**.

Deleting a client with no line saying who deleted it is not a successful delete — it is the single outcome the trail exists to prevent. The audit named them: accounts and roles, teams, grants, integration keys, backups, payment accounts and bank details, practice settings, and company or client delete.

Thirty call sites, in nine files:

Where	Lines	What
<code>users.routes.ts</code>	8	created, changed, switched off, access changed, factor cleared, ownership handed on, password link sent, stored-password key rotated
<code>teams.routes.ts</code>	6	team created / changed / switched off / deleted, member changed, clients changed, a client assigned
<code>companies.routes.ts</code>	5	company deleted, the clients orphaned with it, and a bank account added / changed / removed
<code>integration.routes.ts</code>	3	a key made, re-scoped, revoked
<code>backup.routes.ts</code>	3	downloaded, emailed, restored
<code>clients.routes.ts</code>	2	a client deleted (two doors — on its own, and by unlinking its last company)
<code>timesheet.routes.ts</code>	2	a grant given, a grant taken
<code>settings.routes.ts</code>	1	the practice-wide defaults

They call `mustAudit` instead of `audit`. Everything else on the trail — a sign-in, a document emailed, a period closed, a client's portal passwords read — still calls `audit` and still swallows, which is the half of this change a regression would be quiet about.

The act and its line are one transaction

This is the part that matters, and the part that is easy to get wrong in a way that looks right. A `mustAudit` whose line is written **outside** the act's transaction is not the rule — it is the rule's shape with the opposite behaviour: the row is deleted, the line is then refused, the caller is told nothing happened, and the row is gone. Worse than the swallow it replaced, because it also lies.

So twenty-five of the thirty are written on the act's own transaction handle, and some handlers were restructured to make that possible. Three cases worth naming:

- **Deleting a company** had its three deletes in a transaction (AUDIT-1) and ran the orphan-client loop afterwards on the pool, because a swallowed line could not fail anything. The loop and both lines are now inside the transaction: either the company, its orphans and every line exist, or none of them do.
- **Unlinking a client's last company** did the unlink, the delete it causes and the line as three bare calls. All three are now one transaction — the unlink too, or a client could be left floating with no company and nothing explaining how.
- **Minting an integration key** wrote the key and then its companies, which was a latent fault of its own: a key could exist with none of its companies. Key, companies and line are now one.

The remaining five are on the **pool** on purpose, and each has the reason written beside it. Three are the backup routes: the export builds an archive and streams it, the mailed export sends it, and the restore has already replaced every table — none of which a transaction could hold without keeping one open across a file transfer. The export's line is written **before** the file is handed over, which is what makes it safe: a refusal means the archive never leaves. The other two, the password link and the stored-password key rotation, have already happened by the time the line is written: the mail has left the building, the passwords have all been rewritten and committed. They still refuse, because the refusal is the only way anybody learns the trail is broken.

Which route uses which shape is pinned in `src/services/audit-strict.test.ts`, by name, because getting it wrong is silent.

Payment accounts had no vocabulary at all

Everything else on the audit's list was already recorded and merely swallowed. The bank details were not recorded **anywhere** — there was no event name for them, so the three routes that add, change and remove a company's payment accounts wrote nothing.

It is the most attractive row in the database to change quietly. A swapped IBAN breaks nothing, shows on no screen anybody looks at twice, and prints on every invoice the company raises from that moment. The only way anybody finds out is a client paying on time and the practice not receiving it, weeks later, with nothing saying when the number changed.

Three new event names — `payment_account.created`, `.changed`, `.deleted` — with the account's **label** as the subject, the **field names** that moved, and flags for whether the account was, or became, the one invoices start on. The numbers themselves are not on the line: the detail vocabulary carries field names and never their contents (§12.5), so the trail says *the IBAN changed* and not what to. The account is the record of what it says now.

What the person sees

One sentence, in one place, whatever the act was:

This was not done, because the record of who did it could not be written. Nothing has changed.

Tell whoever looks after the server — the audit trail is not accepting lines, and until it is, the things that are kept on it cannot be done.

Answered as **503** and not **500**: nothing is broken about the request, and the server will do the thing again once the trail accepts lines. The sentence says the three things somebody needs in the order they need them — it was not done, nothing changed, and who to tell — and the last matters most, because this is the one refusal on the list that is nobody's mistake. It is answered in `safeErrorBody` by the error's own class rather than by sniffing its message, so a different failure that happens to mention the trail is not dressed up as this one. The event name is **not** in the reply; it is on the log line.

Two things it deliberately does not do

- **Sealing is outside the refusal.** The **line** is what the act is conditional on, and by the time sealing is attempted the line is written. A sealing problem has its own repair path — the sealer runs again every minute — so it must not turn a delete that did happen into a message saying it did not. That would be the same lie pointing the other way.
- **Four lines are deliberately left swallowed**, each named with its reason in the test. `integration.read` is read continuously by the accounting application, and refusing it on a trail fault would stop the other application; the key's own use is recorded on the key row anyway. `credentials.read` and `credentials.changed` are the super admin's ordinary daily work and are not on the audit's list — refusing them would stop the practice filing a VAT return because the trail was busy. `period.lock_changed` is already refused to everybody but the owner and the company's approver.

One event is **both**: `user.changed` is strict when an administrator edits somebody's account, and swallowed when a person changes their own password, which would otherwise lock somebody out of rotating their own password over a trail fault.

How it is kept true

`src/services/audit-strict.test.ts` reads the route layer and pins, by equality rather than containment: which thirty sites are strict, which handle each writes on, that nothing in the route layer binds the name `tx` except a `db.transaction` callback — the first deliberate break of this work replaced the transaction with `const tx = db;` and left twenty-five call sites reading `}, tx)` with the handle test still green — the four lines left swallowed, and the one event allowed to be both. Every assertion has a vacuity guard.

`tests/e2e/e2e-audit-strict.mjs` is the half only a database can show: a strict line refused inside a transaction leaves the client **still there** and nothing on the trail; the same refusal on the pool does not roll the act back, which is why the handle is pinned per route; an ordinary `audit()` line still swallows; and the bank-account line comes back through the real detail scrub with its field names and without the IBAN.

12.5f Security events — the screen the seal was built for

Settings → **Security events**. New on **13 October 2026**, the fourth point of the sixth step of the security audit, and the last of that step.

What it closes

§12.5d chained and fingerprinted every line on 12 October. §12.5e made thirty acts refuse to happen rather than go unrecorded, the same day. Neither produced anywhere to look. The only way to ask whether the trail held was `/api/diagnostics`, which since DIAG-1 is the super admin's — so for a fortnight the practice kept a sealed register that nobody could read.

Why this screen can be an administrator's when the audit log cannot

The comment at the top of `routes/audit.routes.ts` is the strongest in this codebase: `GET /api/audit-log` is `super` because it records **who read which client's portal passwords**, which is a map of where the practice keeps its secrets and who touches them.

That reasoning is intact and this screen does not weaken it. A Security events screen is a **narrower view, not a wider audience**: `credentials.read` and `credentials.changed` are not on it, and that exclusion is the only thing standing between “an administrator may open this” and the sentence above. The owner sees exactly the same screen — one screen with two truths is a rule that has to be right twice and will eventually be right once.

Every event in the trail is in exactly one of four buckets, and `services/security-events.test.ts` holds that to **equality** against `AUDIT_EVENTS`: an event added and put in no bucket would simply never appear, with nothing failing.

Bucket	For
watch	The sharp end. <code>auth.factor_failed</code> leads it, and the order is not cosmetic: reaching the second factor means the PASSWORD WAS RIGHT, and on a screen opened once a week what is first is what is seen.
changes	Who may do what — accounts, roles, grants, teams, keys, the practice's settings and its mail server.
records	Destroyed, or money moved. The payment-account events are here rather than under <i>changes</i> because of what §12.5e found: a swapped IBAN is the one act in this application where the loss is somebody else's money.

Bucket	For
NOT_SHOWN	Excluded, each with its reason in the file. The two credential events; ordinary daily traffic (<code>auth.signed_in</code> , <code>auth.failed</code> , the integration read); and the practice’s own bookkeeping, which belongs on the record history.

A limited administrator

The owner’s answer, 13 October: **practice-wide lines yes, company lines only for theirs.**

A line with no company is about the building, not a company — a lockout, an address working through a list, a second factor refused, the trail failing to seal. Narrowing those would leave a limited administrator with a nearly empty screen and no sight of the events most worth seeing. A line carrying a company is shown only to somebody who holds it, which is the ADMIN-3 ladder applied to a sixth door.

It is done by **filtering rows, not by refusing a route**, which is the only way to narrow a list — and it is exactly the “separate piece of work” the comment above the `super` policy kind says the register still needs, done here for the screen that could do it. The route reads four pages and cuts to one rather than running a count query over what will one day be the largest table in the database.

The screen **says** it is narrowed. A short list with no explanation is how somebody concludes the register is incomplete.

The weekly note

Deferred from §12.5d on purpose: the message has to be able to say **where to look**, and until this screen existed the only honest answer was “ask somebody to read a JSON route”. An email that reports a problem and cannot say what to do about it trains people to file it.

Only when something needs attention — the owner’s choice. Four things qualify: the trail not holding or lines unsealed too long; a password that was right with a second factor that was not; one address working through a list, or an account locked; a backup that failed, or a copy of the practice leaving the building. A week with none of those sends **nothing** — not a short note, nothing — so that an email from this address always means something.

What that gives up is written down rather than glossed: an email that arrives every week is its own heartbeat, and if it stops the server has stopped. This one cannot give you that. The **nightly backup is the heartbeat instead**, and it already alerts on its own absence (§12.3).

Considered **hourly** and sent at most weekly, with its own stamp on disk beside the backup’s — the SCHED-1 reasoning again: Passenger stops the process between visits, so a job that only fires on Mondays fires on the Mondays somebody happens to be working at that hour. The stamp moves whether or not anything was sent, or a quiet week would leave the clock unmoved and the first noisy hour afterwards would send immediately, which is not weekly but “as soon as anything happens” — and that is what the alerts are for.

The words are decided in `services/audit-weekly.ts`, which is **pure** — no database, no mailer — so the half worth testing is testable without either. `mail-content.ts` puts them in the same shell as every other message. It carries no link, for the reason the security alert carries none: a mail about a possible account takeover that teaches people to click links in mail about account takeovers is how the next one gets in.

12.5g One transaction per act somebody proved themselves for

New on **13 October 2026**, the second point of the sixth step of the security audit — and the shape it ended up in is not the shape it was asked in.

What was asked, and what was built

The audit asked for an `act-transaction.ts`: a general wrapper, so that any handler behind a step-up gets one transaction automatically and cannot forget. That was put to the owner with what it would cost, and not taken. Three reasons:

- **§12.5e had already done 21 of the 29 by hand**, the day before, each with its reasoning written beside it. A general wrapper replaces reasoning that is explicit with machinery that is implicit.
- **A request-level transaction holds itself open across whatever else the handler does** — a mail send, a file write, a session cookie. §12.5e's own note says why that is wrong, and the four backup routes cannot do it at all: the export streams an archive, the restore replaces every table.
- **What was left was a handful of handlers**, all second-factor ones, fixable with the pattern already used 21 times.

So what holds the property is a test rather than an indirection: `src/services/act-transaction.test.ts` reads every route behind a step-up and requires each to be **in one transaction, or named with the reason it is not**. The two lists are held to equality against the call sites, so an act added tomorrow is in neither and it fails with a message saying what to decide. That is the same bargain the wrapper offered, bought more cheaply, and it keeps the reasons readable.

The four that were fixed

All in `mfa.routes.ts`, and all the same fault: the writes that make a second-factor change coherent were split across a transaction and two or three bare calls after it.

Route	What could be left half-done
<code>POST /api/auth/2fa/totp/confirm</code>	The secret confirmed and the codes issued, with the enrolment flag unset and every other session still open
<code>POST /api/auth/passkeys/register/finish</code>	The device registered, the other sessions still open, and no recovery codes issued at all

Route	What could be left half-done
<code>DELETE /api/auth/2fa/totp</code>	No authenticator, ten recovery slips that still work against whatever is set up next, and the sessions still open
<code>DELETE /api/auth/passkeys/:id</code>	The same, for a device

Ending the other sessions is the sharpest of these, and it is not housekeeping. It is the security purpose of the act: if two-factor is being set up because something went wrong, a session from before that moment is exactly what must not survive it. On the pool it could fail after the factor had changed, leaving the person told it was done and the dangerous session still open.

`markSteppedUp` joins them for the enrolments: a session stamped as stepped-up for an enrolment that rolled back would be a standing permission earned by an act that did not happen.

The fourth was found by the test, not by reading

`POST /api/auth/passkeys/register/finish` is gated through `passwordOkForEnrolment` rather than by a literal `needsStepUp`, so the first scan walked past it and so had I. It had the same gap as confirming an authenticator, on the half of the feature a takeover actually uses (LINK-2d). The scan follows the helper now, and that is the argument for having written it rather than relying on four careful readings.

What stays outside, deliberately

The audit line and the account notice. The notice is an **email** — slow, able to hang, and impossible to roll back once sent — and the line is a swallowing `audit()`, because a self-service factor change is not on §12.5e's list of acts that may not happen unrecorded. Both would have been pulled inside by a general wrapper, which is the fourth reason it was not built.

The recovery codes for a new passkey are **hashed before the transaction opens**: hashing is deliberately slow, and a transaction held open across it holds a connection for nothing.

The helpers take an executor, and default to the pool

`endOtherSessions`, `refreshEnrolmentFlag`, `clearRecoveryIfLastFactorGone`, `enrolmentOf` and `markSteppedUp` each take an optional transaction handle, exactly as `mustAudit` does. Everything that passes nothing is unchanged — a password change, a lever being switched off, somebody signing out everywhere — and the test asserts that such callers still exist, so the default is exercised rather than merely declared.

12.5h What is checked before a change reaches the server

New on **14 October 2026**. Nothing on any screen changed; this is about the gate every change has to get through, and it is here because a gate with a hole in it looks exactly like a gate.

The three checks

A deploy is never automatic. Somebody with write access opens **Actions** → **Deploy to Plesk (manual)** → **Run workflow**, and before anything is copied the runner does this:

Check	What it runs	What it would catch
Types	<code>npm run typecheck</code>	A figure, a field or a function used in a way the code does not allow — in the application, in the scripts, and in the tests
Behaviour	<code>npm test</code>	A rule of the business that no longer holds: a VAT total, a closed period that lets a change through, a person seeing a company they are not on
Build	<code>npm run build</code>	The application failing to compile at all

If any of the three fails, nothing is copied and nothing on the server moves. The old version carries on serving.

And the last step, which is not a check

After the copy comes one more step, and it is worth knowing about separately because it fails differently: the deploy writes a timestamp into a file called `tmp/restart.txt`, and that is the signal for the application to restart on the next request. **Until it restarts, the new files are on the server and the old program is still running them.**

That step opens a *second* connection to the server, seconds after the long one has closed — and on **deploy #305, 15 October 2026**, the hosting company's own anti-flood rule refused it. Nothing was sent, the run went red with `exit code 12`, and the message read as *the deploy failed* when the deploy had in fact worked perfectly.

Two things changed on 15 October:

- **It tries three times**, ten seconds apart. That run would have passed on its second attempt. It does **not** keep trying for ever: a restart that genuinely cannot happen has to end red, because a deploy reported green while the old program is still serving is the one state nobody would go looking for.
- **If all three fail, the message says so in words** — that the application IS deployed, that the old version is still being served, and the one-line remedy, which needs no SSH and nobody's help: **Plesk** → **File Manager** → **the app folder** → `tmp/` → **open** `restart.txt` **and save it**. Any change to that file restarts the application on the next request.

Do not re-run the whole workflow to fix a failed restart: it repeats a three-minute transfer that already succeeded, to redo a two-second step.

The hole that was in it, and why nobody saw it

Until 14 October there were two checks, not three — and the missing one was invisible in a particular way. The build compiles a list of files that **excludes the tests**, on purpose: tests do not ship to the server, so there is no reason to compile them for a deploy. The consequence nobody had followed through is that a type fault *inside a test* could not fail the build, could not fail the test run (the test runner does not check types, it runs them), and so could not fail a deploy.

Seventeen had collected, in six files, over several months. Not one was a fault a person using the app could reach — every one of those tests passed, and went on proving what it was written to prove. They were tests whose types had drifted away from the code they test, which is cheap on the day and expensive later: a file full of warnings is a file nobody reads carefully, and the eighteenth one — the one that matters — arrives in a file where seventeen are already being ignored.

All seventeen were fixed on their own merits rather than silenced, and one of them turned out to be a real hole in the testing: a search test was capping a kind of result that no longer exists, so the question it was meant to ask had not been asked by anything for weeks.

What keeps the gate honest

`src/services/deploy-gate.test.ts`, which runs in the **Behaviour** check above and reads the deploy file itself. It fails if:

- any of the three checks is missing;
- a check has moved to **after** the copy — a check that runs once the files are on the server is a report, not a gate;
- a check is marked to be skipped, or marked so that failing it does not stop the run (one line, and it reads like housekeeping);
- the type check has been pointed at the **narrow** list of files, the one that excludes the tests, which would quietly restore the whole problem;
- or the narrow list is ever widened to include the tests, in which case the separate type check is doing nothing the build does not, and somebody should decide which one stays.

Eight deliberate breaks, each caught. Two of them were the test fooling itself rather than the deploy being wrong, which is the usual lesson: the first version read the explanation written above the check as though it were the check, and the second read a file path containing `/*` as though it were a comment and ate half of it.

12.5i Money arriving, and the two kinds the record stopped claiming

New on **14 October 2026**. The screen is the one in §12.5 — **Company details** → **What changed on the figures** — and nothing about how it works changed. What changed is what reaches it.

What a payment writes, and what used to be missing

Marking an invoice paid writes **two** rows, and they are halves of one fact:

Row	What it is for
A receipt document	What the client gets. The statement, the aged receivables and the Dashboard all read this
A <code>cash_receipts</code> row	That money arrived, in a named account, on a named day. The bank reconciliation reads only this

Neither got a line in the record of changes. And the consequence was not symmetrical, which is what made it worth fixing rather than noting:

- Deleting the receipt document **also deletes the `cash_receipts` row**, deliberately — they were written together and they end together, because a receipt row pointing at no document is a puzzle rather than a record.
- The deletion wrote a full before-image of the **document**. That image carries the document's total. It does not carry the date the bank saw, which account the money went into, or any sign that a `cash_receipts` row had existed at all.

So the one table a reconciliation reads could lose a row and nothing, anywhere, said it had. A set of figures that balanced in March and did not balance in April had no explanation to offer.

What is recorded now

When	The line
A payment is recorded	<i>"Receipt created"</i> , naming the invoice it settled and who paid
Forty are recorded at once	One line each , not one for the run
The receipt document is deleted	<i>"Receipt deleted"</i> , carrying the amount , the date and the status

A creation keeps no figures on purpose: the row itself still holds them, and copying them would double the table for no answer. A deletion keeps all of them, because afterwards there is nothing left to read.

One line per receipt in a bulk run is the choice worth explaining. The audit log already says "forty receipts were raised, at 14:32, by Maria" — that is the act. It cannot say *which* forty, because it holds counts and never values, which is its own rule and a good one. So the money gets a line each, and a receipt found six months later has the same history whichever button raised it.

And two kinds that were being claimed and could not happen

The record of changes knew six kinds of record. Four are written. The other two — **supplier invoices** and **supplier payments** — were declared, each with its own list of figures to keep, and neither was ever written once.

That mattered because of how it read. Somebody opening the file saw six kinds covered and would have had no reason to check; it is the same hazard as a file in the tree that nothing imports, and the fifth security review called it out.

The decision taken was to wire all three unwritten kinds. **Only one of them could be**, and finding that out was most of the work:

- **Supplier payments** have no table. Migration 0059 dropped it with the aged payables it fed. There is no row to keep a history of.
- **Supplier invoices** still have a table — it is kept because it names uploaded files — but nothing has written to it since the supplier capture queue left in September. The only place the application still mentions it is the check for whether a company is empty enough to delete.
- **Cash receipts** were the real one, and are now written on all three of their paths.

So the two were **removed** rather than left declared with an explanation beside them. It cost no database change, because the kind is stored as plain text rather than as a fixed list, and any line written under an older kind still reads.

What holds it from now on is `review-status.test.ts`, and the rule it enforces changed shape with this. It used to pin the gap — six declared, three written. It now pins the rule: **every kind declared is written somewhere, and nothing is written under a kind that is not declared**. The first stops the next unwritten declaration lasting a day; the second catches a misspelling, which would otherwise make lines no screen groups correctly.

The wiring itself is held by `receipt-history.test.ts`, which reads the source rather than the behaviour — every write to the receipts table must sit in a route that also writes the line, and the deletion must read the row **before** deleting it, since a line written afterwards has nothing left to copy. And `tests/e2e/e2e-receipt-history.mjs` proves the whole thing against a real database: it takes a payment, reads the line off the screen's own route, deletes the receipt, and checks that the amount survives in the history after the row is gone. Fifteen deliberate breaks, each caught.

12.6 Linked applications, and the key they read with

Settings → **Linked applications**, administrators only. This is where the firm's accounting application is given permission to bring in the invoices and credit notes each company issues here, so it can post them to its books.

What it is allowed to read is fixed and narrow: the issuing companies with their VAT numbers, each company's service names, and the issued invoices and credit notes between two dates it asks for, with their lines and the client each was addressed to. Nothing else — no client portal logins, no notes, no costs, no bank lines — and nothing at all in the other direction. The link cannot change a single figure in this application.

Making one

Type what the key is for, choose **when it stops working** and **which companies it may read**, press **Make a key**, and copy it. It is shown **once**: only a fingerprint of it is kept here, so a key that is lost is replaced rather than

looked up. Paste it into the accounting application on its own Invoicing link screen, together with this app's address, which the box tells you.

The app will **ask you to confirm it is you** first. A key is not like a password — it goes on working after the password is changed, after the session that made it has gone, and after a borrowed machine has been handed back. That puts it in the same class as downloading a copy of the practice, and it is gated the same way.

When a key stops working

The date box is “**Stops working at the end of**”, and the wording is exact: a key dated the 31st works through the whole of the 31st and is refused from midnight. Left empty, the key never stops — which is what every key did before this and remains the sensible answer for a permanent link between the firm's own applications. A key with a short life is for anybody else: an outside bookkeeper, a one-off reconciliation, a trial of something.

An expired key is refused **in the same words as a key that never existed**. That is a decision, not an oversight: a distinct message would tell whoever is holding an old string that it was once genuine. The consequence is that an expiry is silent on the other side — the accounting application brings in nothing and reports nothing — which is why the notices below exist and are not optional decoration.

Which companies a key may read

Either **every company**, or a named list. A key with a list is refused any company outside it, and the companies outside it are **left out of the companies feed entirely**, so a narrowed key cannot discover what else the practice has. The refusal is also decided *before* the company is looked up, so a company a key may not read answers identically whether or not it exists — the alternative would let a key enumerate the practice by the difference between “no such company” and “not yours”.

A key naming **no** company at all reads nothing, and the app refuses to make or save one: “name at least one company, or leave the key reading all of them”. The reach is held as a flag plus a list rather than as “an empty list means everything”, because that reading turns a key whose companies were never chosen — or whose rows went with a deleted company — into a key over the whole practice while looking, on the screen, like the most restricted one there is.

The list, changing a key, and revoking

Each key shows what it is for, its last four characters, who made it and when, what it **reads**, when it **stops working** and when it was last used.

A new key has to be given both. Since 9 October the server refuses a key with no date and refuses a key that says nothing about which companies it reads — it is not enough for the screen to offer them. A key with no date is a password that outlives the person who made it, the session it was made in, and their leaving the practice; and “reads every company” used to be what you got by not choosing, which made the widest key the easiest one to make. The screen now fills the date in a year ahead, which you can change, and leaves both company choices empty until you pick one. A date already in the past is refused too, because a key born expired is refused by the other application the moment it is pasted in.

The keys already in use are untouched. This is a rule about what may be *made*. A key created before this change, with no date and reading every company, goes on working — give it a date with **Change** when you next look at the screen.

Change gives an existing key a new date, or a different set of companies, without minting a new one. This is the control that makes the expiry usable: without it, “the key expires on Friday” has exactly one cure — a new key pasted into the other application by somebody who has to be at both machines. It is behind the same step-up as making a key, and it writes “*Changed a linked application’s key*” to the audit log. Changing the date also clears the record of having warned about the old one, so the warnings start again against the new date.

Revoke stops it at once; the accounting application then brings in nothing until a new key is made and pasted into it. It **asks you to confirm it is you**, since 9 October — see the note under *Confirming it is you* for why that reversed, and what the emergency-brake argument got wrong.

A revoked key stays on the list, marked, rather than disappearing — the record of what it did has to have a name to point at.

Being told, in three places

Any key with a date on it is spoken about from a fortnight before, and again once the date has passed. Three places, all drawing the same sentence from one rule in the server, so that no two screens can disagree about one key:

Where	What it does
A banner on the way in	Administrators only, drawn on both ways into the app — signing in, and reloading with a session already in hand. It names the key, says what stops, and offers Open linked applications . Not now puts it away for the rest of the session. One line only: the worst key, expired before expiring
The Dashboard’s queue	Tagged <i>Linked app</i> , sorted in among the deadlines and unpaid invoices. Unlike every other row there it is not scoped to the company in the switcher, because a key belongs to the practice and may read several companies
Email	To <code>ALERT_EMAIL_TO</code> , falling back to the backup address, like every other alert

Two pieces of news, not one: *expiring* and *expired*. A practice that read the first is still told the second, because “it stops in nine days” and “it has stopped” call for different things. Each is said **once**: the app checks hourly, and a fortnight’s warning on an hourly timer would be 336 messages and a filter rule pointed at the only channel the practice watches. The stamp that prevents that is kept in the database rather than in the running

process, because the server is stopped between visits and an in-memory counter would start empty several times a day.

A **revoked** key is never spoken about, before or after its date. It was stopped on purpose, and a warning that it is about to stop would be noise about a decision somebody already made.

What is written down

Six things reach the audit log under People & access: the key being **made**, **changed**, **revoked**, every time it is used to read the invoices — with the key's name, the range of dates asked for and how many documents went — and the two the app writes by itself, a key being **about to expire** (with how many days were left) and a key having **expired**. The last two name *the hourly check* as the person, which is true: nobody did them. The last of those is the one worth knowing about, because the *Last used* column on the screen holds one date and is overwritten: it answers “was this key used recently” and cannot answer “what has it taken”. Those lines show **nobody signed in** as the person, which is correct — the accounting application has no account here, and the key is what stands in for one.

Making a key also **sends an alert** to the app's own mail address, at once, the way a new account or a new way of signing in does. If one of those arrives and it was not you, revoke the key named in it.

12.7 The security code the three applications share

The practice runs three applications built the same way — invoicing, payroll and accounting — and they answer the same questions about signing in: what counts as a password, how long a wrong one is held against you, how a second factor is checked, what an invitation link lives for, which headers every page carries, how a stored secret is encrypted. A rule written three times is a rule that drifts, and the drift is never noticed in the place it matters.

So those answers live in **one folder, copied into all three repositories byte for byte**:

`src/shared/security/`. Invoicing took it on **12 October 2026**; payroll and accounting had it since 9 October.

How it is kept the same. Three things, each catching what the one before it cannot:

- `MANIFEST.json` records a SHA-256 of every file in the folder, and a test in each application fails if any file has changed — so an edit made without noticing the folder is shared stops that application's own test run, with a sentence saying the change belongs in all three or does not belong in the folder;
- the manifest cannot see another repository, so `check-peers.sh` compares this copy with the other applications' copies **on the same computer**, file by file, both ways. It is plain `sh` and `cmp`, with no Node, so it runs on the owner's Mac;
- `.github/hooks/pre-commit` runs that comparison before every commit, and refuses a commit that changes the shared folder while the copies are apart. (`git config core.hooksPath .github/hooks`, once per checkout.)

What invoicing uses from it today. The password rule, the authenticator, the passkey checks, the session lifetimes, the security headers every page carries, and the invitation and reset link lifetimes — 72 hours and 24,

which are the values agreed across all three.

And what it still keeps its own, with the reason. This is the honest half, and each one is a decision rather than unfinished work:

Still invoicing’s own	Why
The key that encrypts stored passwords	The shared version requires <code>CREDENTIALS_KEY</code> and will not run without it. This server has never had one, so everything encrypted here is under <code>SESSION_SECRET</code> . Switching over before the key is on the server and the Rotate button has been pressed would stop the application.
Wrong-password lockouts	The shared version locks an <i>address</i> out after twenty wrong guesses. This application slows it down and tells you instead, deliberately, because an office shares one address and locking it shuts out everybody at it.
The account emails	The shared version has ten kinds and not the three this application needs — the address-changed notice, the role change and the switch-off.
The cost levelling on a sign-in	This version pays the same work whichever kind of stored password it is checking. The shared one does not, and the difference is measurable from outside.
The rate-limit list, the link mechanism, the origin check	Same ideas under different names, and four things this application needs have no home in the shared versions.

Those five are on the list for the comparison the three applications are due, which is where a difference between them can be settled with all three in the room rather than one at a time.

Glossary

Term	Meaning
Accounting client	A client whose filings the practice handles; only they have deadlines

Term	Meaning
Company switcher	The menu at the top right choosing which company you are working in
Credit note	A document reversing all or part of an invoice
Pro-forma	A request for payment that becomes an invoice only when converted
Reverse charge	0% VAT on a supply to an EU business with a valid VAT number; the customer accounts for the VAT
VIES	The EU register of VAT numbers, which a client's number is checked against
Second factor	The second proof of identity at sign-in (authenticator code, passkey or recovery code)
Step-up	Confirming it is you again before a sensitive act
Closed period	A period whose figures are locked
Link key	The key a linked application reads this one's issued invoices with (12.6)

What's new

Date	Change
15 October 2026	A deploy that had worked reported itself as failed, and the message sent somebody to redo three minutes of work for a two-second step. Deploy #305 went red with <code>exit code 12</code>. The transfer had already succeeded — every file was on the server. What failed was the very last step, the one that writes a timestamp into <code>tmp/restart.txt</code> to make the application pick up the new version. That step opens a second connection to the server moments after the long one closes, and the hosting company's own anti-flood rule refused it: <code>0 bytes received</code>, same key and same host as the transfer that had just worked. The state it left behind is the thing worth naming: new files on the server, old program still running them, and a message that read as <i>the deploy failed</i> — so the natural response was to re-run the whole workflow, repeating a three-minute transfer to redo two seconds of work. Now it tries three times, ten seconds apart, which would have passed on the second attempt that night. It deliberately does not keep trying: a restart that genuinely cannot happen must end red, because a deploy reported green while the old program is still serving is the one state nobody would go looking for. And if all three fail, the message says it in words — the application IS deployed, the old version is still being served, and the remedy needs no SSH and nobody's help: Plesk → File Manager → the app folder → <code>tmp/</code> → open <code>restart.txt</code> and save it. Four checks in <code>deploy-gate.test.ts</code> hold it: that it retries, that it pauses between tries, that it still fails the run when it gives up, and that the message still names the state. Seven deliberate breaks, each caught — and the step itself was run four times against a stand-in server that drops the connection, to prove it recovers on the second try, on the third, and gives up correctly on the fourth. No

Date	Change
	migration, nothing on any screen. Handbook §12.5h gained a section
14 October 2026	A convention in the front end became a rule, and the count that was on the list was wrong twice over. Nothing on any screen changed and nothing was exploitable. <code>app.js</code> builds its HTML by hand, and its own convention is that anything put into the page goes through an escaper first. The September audit found one place that did not; a re-read on 13 October found two, and it went on the list as two lines to put right. It was thirty-five. Both counts had looked only at dropdown options, so every <code>data-id</code>, <code>data-user</code>, <code>data-edit</code> and <code>id</code> attribute built the same way sat outside both numbers without anybody deciding that it should. That is the part worth keeping: not that two lines were wrong, but that nobody could say how many were, because a convention has no count. Every one of the thirty-five carried a reference number this server's own database generated — which cannot break out of an attribute, so there was nothing to exploit, and that was never the reason to fix it. The reason is that <i>"is this one safe?"</i> had to be answered site by site, from where the value came from, with nothing holding the answer — and the day one of those expressions starts carrying a client's name instead, the convention is the only thing that would have caught it. All thirty-five now go through the escaper, and <code>attr-escaping.test.ts</code> makes it a rule: a reference number in any attribute of any script the browser is served must be escaped. The rule stops there deliberately. The file has 209 such expressions, most of them arithmetic that positions the charts, and a rule demanding an escaper around a subtraction is a rule somebody switches off — taking the thirty-five real ones with it. Four deliberate breaks, each caught, two of them breaking the test's own machinery rather than the code. No migration, no new screen, and no behaviour change

Date	Change
14 October 2026	Money arriving is written down, and two kinds of record that were being claimed and could not happen are gone. The record of changes — <i>Company details</i> → <i>What changed on the figures</i> — answers the question the audit log cannot: <i>this invoice says 500 now, it said 590 when the return was filed; who changed it?</i> It was written for invoices, costs and recorded time. It was never written for a cash receipt, and a cash receipt is the only row the bank side of the figures reads. The consequence was not symmetrical, which is what made it worth fixing. Marking an invoice paid writes two rows — the receipt document the client gets, and the row that says money arrived in a named account on a named day — and deleting the receipt document deletes both, deliberately, because they were written together. The deletion recorded a full before-image of the document, which carries the document's total and does not carry the date the bank saw, the account the money went into, or any sign that the second row had existed. So the one table a reconciliation reads could lose a row and nothing anywhere said it had: figures that balanced in March and did not balance in April had no explanation to offer. All three paths now write a line — one payment, a bulk mark-paid run, and the deletion — and a bulk run writes one line per receipt, not one for the run, because the audit log already records the act and cannot say which forty it was. A creation keeps no figures (the row still holds them); a deletion keeps the amount, the date and the status, which afterwards are the only copy. And the other half, which the decision did not expect. The record knew six kinds and wrote three; the instruction was to wire the other three. Only one could be. Supplier payments have no table — migration 0059 dropped it with the aged payables it fed, so there is no row to keep a history of. Supplier invoices still have a table, kept because it names uploaded files, but nothing has written to it since

Date	Change
	the supplier capture queue left in September; the only mention left in the application is the check for whether a company is empty enough to delete. So those two were removed rather than left declared, which cost no database change because the kind is stored as plain text. Three sentences were wrong and are now right: the screen's own description promised "supplier payment" figures that cannot occur, the user manual claimed cash receipts were covered before they were, and it claimed a filter on that list which has never existed — the list is the newest sixty lines, and whether to build the filter is on the practice's list to decide. What holds it: <code>review-status.test.ts</code> now pins the rule rather than the gap — every kind declared is written, and nothing is written under a kind that is not declared — <code>receipt-history.test.ts</code> holds the wiring, including that the deletion reads the row before deleting it, and <code>e2e-receipt-history.mjs</code> proves it against a real database. Fifteen deliberate breaks, each caught. No migration. Handbook §12.5i new, §12.5 corrected; manual §10.10 corrected

Date	Change
14 October 2026	The deploy now checks the types as well, because for months it could not fail on them. Nothing on any screen changed. A deploy runs three checks before a single file is copied to the server — the types, the tests, and the build — and until today it ran two. The missing one was invisible in a particular way: the build compiles a list of files that excludes the tests, which is correct (tests do not ship), and the test runner runs the tests without checking their types. So a type fault inside a test could not fail the build, could not fail the test run, and therefore could not fail a deploy. Seventeen had collected, in six files. Not one was a fault a person using the app could reach — every one of those tests passed and went on proving what it was written to prove. They were tests whose types had drifted from the code they test, which is cheap on the day and expensive later: the eighteenth one, the one that matters, arrives in a file where seventeen are already being ignored. All seventeen were fixed on their merits rather than silenced, in four unrelated groups: five were a fixture that had never been given a type, four were two helper functions nobody has called since the spending reports were removed in September, six were the compiler being right that <code>expect(x.allow).toBe(false)</code> does not tell it anything about <code>x</code> (now a <code>refused()</code> helper that throws <i>the policy allowed this and the test expected a refusal</i>, which is the sentence somebody needs at seven in the morning, rather than a cast that silences the line), and two were a real hole: a search test was capping a kind of result that left with the ledger removals, so the question it was written to ask — can a pile of invoices bury the client you were looking for — had not been asked by anything for weeks. What keeps the gate honest from now on is <code>deploy-gate.test.ts</code>, which reads the deploy file itself and fails if a check is missing, has moved to after the copy, is marked so that failing it does not stop the run, or has been

Date	Change
	pointed at the narrow list of files that excludes the tests. Eight deliberate breaks, each caught — two of them the test fooling itself rather than the deploy being wrong, which is the usual lesson. No migration, and nothing in the application changed. Handbook §12.5h new; manual §10.13 gained <code>buildId</code>, which <code>/api/diagnostics</code> has returned all along and the page had never listed
14 October 2026	The per-company dropdown names the role before it explains it. It offered the three levels in plain words alone — <i>to look at</i>, <i>to change</i>, <i>to change</i>, and <i>to close months</i> — which reads well while you are setting somebody up and left the screen and both manuals speaking different languages: everything written down says Viewer, Preparer and Approver, and nothing on the screen said which was which. Asked out loud by the owner, looking at the dropdown, which is the moment it became worth fixing. It now reads Approver — to change, and to close months, Preparer — to change, Viewer — to look at. The two practice-level roles on the same card have always been written this way round — <i>Owner — everything, everywhere</i>, <i>Administrator — reaches every company</i> — so the three per-company ones were the odd pair out rather than the rule. The dropdown is wider to suit, and still shrinks on a narrow screen rather than pushing the company name out of its row. Nothing about access changed — the values the screen sends and the server stores are the same three words they were, and nobody's reach moves. <code>roles.test.ts</code> now holds the convention: each level must lead with its name and then explain itself, and the two practice-level lines are pinned beside them so the five do not drift apart again. Four deliberate breaks, each caught. No migration. Handbook §1.4; manual §10.3

Date	Change
14 October 2026	A QR code to scan when you set up an authenticator app — and a blank page fixed on the way. Asked for because typing thirty-two characters into a phone is the worst part of somebody's first morning. The square is drawn by this server, which is the part worth knowing: the usual way to add a QR code is an <code></code> pointing at one of the free QR services, and what a QR code for an authenticator encodes is the shared secret itself — so that one line puts the second factor in a stranger's access log beside the account's email address, where whoever holds it can make that person's codes for ever. It is the commonest way a QR code quietly undoes the thing it is setting up. Here the code is built from the same URI the server already had, travels inside the same reply as the key, and makes no request at all. The key is still printed underneath for anybody whose camera will not focus, and both routes end in the same place. The code is deliberately black on white in dark mode too — a camera hunts for dark squares on a light field, and a QR inverted to suit a dark page is one many phones will not see. The square is proved rather than assumed: the test encodes it with one library and reads it back with a different, independent one, and the URI that comes out must match the URI that went in character for character — including for a Greek-named client, where the percent-encoding is several bytes per letter. A QR code has no partial failure anybody can see; it is a field of black squares either way, and the only way to know whether it says the right secret is to read it. And the blank page. Reported mid-build from the live server: signed in, reload, nothing at all — no sign-in box, no error, no way forward, and not even a way to sign out, because nothing was drawn to sign out with. The console named it exactly: a 403 from the first call the app makes, and an unhandled rejection saying the account needs a second way to prove it is you. The cause was one missing line. The start-up has always had a gate for a forced password

Date	Change
	change; the second factor had one only on the SIGN-IN path, and the note beside that gate says precisely what happens without it — <i>“an account in this state is refused everything else, so loading the practice would throw on its first call”</i> — which is the reload path, and nobody carried the reasoning across. So any account with a live session and no authenticator met an empty screen on every reload. It now opens the same <i>set up an authenticator</i> screen that signing in does, and any other start-up failure draws the sign-in box with the reason on it rather than leaving the shell empty — a blank page must never be this application’s answer to anything. <code>ui-enrol-reload.mjs</code> drives the whole sequence, and it was run against the unfixed code first and reproduced the blank page and the identical console message before the fix was written. No migration. Handbook §12.4; manual §1.2

Date	Change
14 October 2026	The two manuals read against the code, and three structural faults put right. They are the only documents in this repository that get handed to somebody, and the only two that nothing was holding to the application. The good news first, because it is the result: every figure in them was already correct. Twelve characters for a password, one idle day and a fortnight for a session, the 60-second then 5-minute then 15-minute logout ladder, half an hour before an open company closes, five minutes on a “confirm it is you”, 72 hours on an invitation and 24 on a reset, thirty days of security events, two years of audit trail — all checked against the constants the code actually uses, all right. Including the one I was most confident had rotted: §12.5e is titled <i>Thirty acts that refuse to happen unrecorded</i>, and there are exactly thirty, in eight files. My first count said thirty-five because it included the test files that scan for the pattern, and I came close to correcting a sentence that was already true. What was wrong was the structure. (1) Two sections of the user manual were both numbered 1.3a — Favourites and the Dashboard — so “see §1.3a” resolved to whichever a reader found first and the Help page’s index listed the number twice. The second is now 1.3b; neither section’s content changed. (2) Three <i>What’s new</i> rows named handbook sections that do not exist. That trailer is the only index from a change to the place it is written up, so a row pointing at nothing is a change nobody can find. §2.5 was cited for a rule about client reach, in a part that has only two subsections — it is §1.4, <i>People and access</i>. §6.3 and §6.4 were described as “renumbered” when §6 carries no numbered subsections at all; they went with the screens they described, and the rows now say so. (3) §1.9 was promised and never created — the 9 October entry on one-company-at-a-time pointed at it, and the content had been left as a subheading inside §1.6, <i>Searching the company you</i>

Date	Change
	<i>are in</i>. It is not about searching; it is about a rule that closes the company you are working in after half an hour, which is what somebody looks up when it happens to them at two o'clock, and nobody looks for that under <i>Searching</i>. It is now §1.7 One company at a time. What holds all of this from now on is <code>manual - truth . test . ts</code> : no section number may be used twice, every changelog row must name a section that exists in the document it names, and each of the ten figures above is compared against the constant the application reads. Nine deliberate breaks, each caught. One check was built and thrown away, which is worth recording: a scan of every bolded control name in the manuals against the strings the front end renders. It produced 539 candidates and 134 apparent misses, almost none of them real — bold marks emphasis as often as a label, and the real labels are often assembled as the page is drawn, so Save Invoice is <code>Save \${docWordCap(mode)}</code> and matches nothing. A check that cries wolf at that rate is one somebody turns off, and it would have been worse than nothing. No migration, and nothing in the application changed. Handbook §1.7 new, §1.4 and §6 references corrected; manual §1.3b renumbered

Date	Change
13 October 2026	A downloaded invoice is called what it should be called. Reported with two copies of the same invoice, downloaded on the same day and saved as <code>Invoice20INV-034-202620-20Simallin20Ltd.pdf</code> and <code>Invoice_INV-034-2026-_Simallin_Ltd.pdf</code>. Neither name was ever built by this application. The name was correct both times — <code>Invoice INV-034-2026 - Simallin Ltd.pdf</code> — and was then wrapped in a web-address encoding on its way into the header that tells a browser what to save the file as. That header has two halves and only one of them is decoded; the name went in the half that is not, so the browser saved the encoding literally, <code>%</code> signs and all, and whatever handled the file next either dropped the <code>%</code> (the first name) or turned each <code>%20</code> into an underscore (the second). One mistake, two manglings, which is why the same invoice arrived under two names. The encoding was there for a good reason and was the wrong tool: it was guarding against a quote mark or a newline in a client's name closing the header and letting something else be written into it — a real risk, guarded now by building that half of the header out of an allow-list, so the dangerous characters cannot be in it rather than being encoded once they are. Both halves are now sent, which is what the standard has recommended since 2011: the real name, encoded properly, in the half browsers read, and a plain-English fallback in the half they ignore. Three routes had this fault identically — the invoice PDF, the quotation PDF and a quotation's attachment — because the second and third were written by copying the first; a fourth, every stored upload, had the right shape and the same wrong value in its fallback half, so an upload came out well on a current browser and badly on an old one. All four now call one builder, and a test refuses to let any file in the application write a file-name header of its own. And a Greek-named client gets their

Date	Change
	name back. The old rule deleted every character outside plain English, so Αντρέας Γεωργίου Ltd had their invoice download as Invoice INV-035-2026 - Ltd.pdf — the client missing from the name of their own invoice. The Greek is kept now, with a transliteration beside it for any browser too old to read it, spelled the way the Registrar spells it on an English certificate: <i>Georgiou</i>, not <i>Georgioy</i>. That spelling cost two goes — ου is not ο + y , and Σταύρου is <i>Stavrou</i> because the stress sits on the epsilon — and both were found by running the practice’s own kind of client names through it rather than by reasoning about Greek. Characters no file system will accept are still removed, now by the union of the Windows, macOS and Linux rules rather than one of them, plus the leading and trailing dots and spaces Windows discards without telling anybody. The file names change for everyone from the next deploy; nothing already downloaded is renamed, and nothing inside any PDF changes. No migration. Handbook §3.7a new; manual §2.8

Date	Change
13 October 2026	Setting up or removing a second factor now happens completely, or not at all. The second point of the sixth step of the security audit — and not in the shape it was asked in, which is worth recording. It asked for a general wrapper that would give every act behind a “confirm it is you” one transaction automatically. That was put to the owner and not taken: the previous day’s work had already done 21 of the 29 such acts by hand with their reasoning written beside them, a request-level transaction would hold itself open across mail sends and file writes, and the four backup routes could not use it at all — one streams an archive out and another replaces every table. What was left was a handful of second-factor handlers. Four of them were fixed, all with the same fault: the writes that make a second-factor change coherent were split across a transaction and two or three bare calls after it. Confirming an authenticator could leave the secret confirmed and the codes issued while every other session stayed open; registering a passkey could leave the device on the account with no recovery codes at all; removing either could leave ten recovery slips that still work against whatever is set up next. Ending the other sessions is the sharpest of these and it is not housekeeping — it is the security purpose of the act: if two-factor is being set up because something went wrong, a session from before that moment is exactly what must not survive it, and on the old shape it could fail after the factor had changed with the person told it was done. The fourth was found by the test rather than by reading: registering a passkey is gated through a helper rather than by a literal step-up call, so the first scan walked past it and so had I. What holds this true from now on is <code>act-transaction.test.ts</code>, which requires every route behind a step-up to be in one transaction or named with the reason it is not, held to equality against the call sites — the same bargain the wrapper offered, bought with a test

Date	Change
	instead of an indirection. The email and the audit line stay deliberately outside : an email cannot be rolled back once sent, and a general wrapper would have pulled both in. No migration. Handbook §12.5g new; manual §1.2

Date	Change
13 October 2026	There is now a screen for the security events, and the sealed audit trail can finally be read. The fourth and last point of the sixth step of the security audit. The trail was chained and fingerprinted on 12 October and thirty acts were made to depend on it the same day — and neither produced anywhere to look: the only way to ask whether any of it held was a diagnostics route the super admin alone could reach, so for a fortnight the practice kept a sealed register nobody could read. Settings → Security events, open to the owner and every administrator. At the top, the state of the trail in words — it holds, it does not hold, or it could not be checked just now — and if it does not hold, what to do first: take a backup, keep it, tell whoever looks after the server. Below it the events, newest first, 30 days by default, in three kinds you can show one at a time: worth a look today (a second factor refused <i>after the password was right</i>, which is the sharpest signal there is; an address working through a list; an account locked; a backup taken, restored or failed), who may do what (accounts, access, teams, grants, keys, the practice’s settings), and destroyed, or money moved (a client or company deleted, a bank account changed, a closed period reopened). It is deliberately not the whole audit log. The two lines recording who read a client’s portal passwords are not on it, and leaving them out is the only reason the rest is safe to put in front of every administrator — the full log stays exactly where it was. An administrator limited to certain companies sees everything belonging to the practice rather than to a company, and of the rest only what belongs to a company they hold; the screen says so rather than leaving a short list to read as “nothing happened”. And a note goes out by email once a week, only when there is something worth looking at — the trail not holding, a right password with a wrong second factor, an address working through a list, or a backup that did not happen. A quiet week sends

Date	Change
	nothing at all, so an email from it always means something; what that gives up is a weekly heartbeat, and the nightly backup is the heartbeat instead. The note names the screen, which is why it waited for one: a message that reports a problem and cannot say what to do about it is one people learn to file. No migration. Handbook §12.5f new, §12.5d amended; manual §10.11a new

Date	Change
13 October 2026	The app carries the practice's name, not a product's. The top-left read  InvoicePro — a product name for software with exactly one customer, which is nobody's idea of a brand. It is now the firm's own mark: the KP monogram in its gold frame, with K.P. Panagi & Co beside it. It is drawn rather than loaded — the monogram is geometry in the page, not a picture file — which buys three things: it follows the light/dark theme without a second file for the dark one, it stays sharp at whatever the Aa text-size control is set to (including 150%, which is where the people that control exists for keep it), and it costs no request. The gold is its own colour in the theme, with a darker value for white and a lifted one for the dark ground, because the printed gold disappears into dark and the lifted one is washed out on white. The browser tab says <i>Invoicing — K.P. Panagi & Co</i>, deliberately not the same thing the bar says: inside the app you know which of the three you are in, but three tabs all reading “K.P. Panagi & Co” would tell you nothing, so the title's job is to be the half that differs. The favicon is the monogram reduced, not shrunk — the frame is dropped and the letters take the whole square, because a hairline double frame at sixteen pixels is four grey smudges; a gold rule under the letters is what tells this tab from the other two at a glance. On a narrow window the name is given up and the monogram stays, at 1360 pixels rather than the old 1300: measured, not guessed — the brand went from 92 pixels wide to 143, so the bar reaches the same pressure 51 pixels sooner. Nothing printed changes; an invoice has always carried the company's own letterhead and never the application's name. Handbook §1.1; manual §1.3

Date	Change
13 October 2026	The backup’s warning no longer goes to the mailbox it is warning about. After it happened: the nightly backup bounced for several nights on <code>452 Mailbox full</code>, and the alert about it was refused by the same mailbox for the same reason, every night. The app did everything it was built to do — recorded the failure, wrote it on the audit trail, sent a warning — and sent that warning into the full mailbox. Now it is said in four places. A line on the Backup & restore card whenever the two addresses are one mailbox; the same sentence on <code>/api/diagnostics</code>; and a warning in the server’s own log at startup, right after the line naming the backup address — the log matters most, because it is the only one of the four that does not depend on mail working. The fourth is the sign-in banner, and it appears only once this has actually bitten: a banner about a configuration choice shown every session with nothing wrong is one people learn to dismiss unread, so it is drawn only when a backup has failed, the refusal was the mailbox, and the alert goes to that mailbox — and then it says that this notice is very likely the only one you will get. All four read one sentence from the server; the browser never judges whether two addresses are the same mailbox, and the alert address is never sent to a browser at all. A full mailbox is also retried now, up to three times that day and only for a mailbox that is full — so emptying it in the morning means that day still gets a backup, where before a single refusal cost the whole day. Every other refusal still counts as the day’s one attempt, because each attempt reads every table in the practice and holds it in memory, and a wrong password at two o’clock is still a wrong password at three. The one case worth naming: a <code>552</code> means both “the mailbox is full” and “that message was too big”, and those want opposite answers — a reason naming the size is never retried. Nothing here empties a mailbox or changes an address: the fix is <code>ALERT_EMAIL_TO</code> in the server’s

Date	Change
	<code>.env</code> , pointed somewhere else, ideally at a different provider so one outage cannot take both. No migration. Handbook §12.3; manual §10.9
13 October 2026	The Invoices menu is one flat list again. It offered <i>All documents</i> and a New › line that opened <i>Invoice</i>, <i>Credit note</i> and <i>Pro-forma</i> in a second list beside it. That grouping was about the words — the three are one idea said three times, and under a parent they can drop the word they share — and it was not about the work: Invoices is the module that is open all day, and it put a hover, a wait and a sideways move in front of the three things most often wanted in it. Now the menu is four lines in one list: All documents, New invoice, New credit note, New pro-forma. Nothing else changes — the addresses, bookmarks, favourites and the screen a reload returns to were always read from the flat list of sections and never from the grouping. Tools keeps its menu, which is the case the grouping was really built for: seven sections there used to push the strip into a sideways scroll on an ordinary laptop. Handbook §1.1; manual §1.3

Date	Change
12 October 2026	Thirty acts now refuse to happen if their audit line cannot be written. The third point of the sixth step of the security audit. Writing a line used to swallow its own failure — if the database would not accept the line, the act went ahead and nothing was said. That is still the right answer for almost everything on the trail: a sign-in that could not be logged is still a sign-in, and turning a logging problem into a refusal would stop the practice working over something that is nobody's fault. It is the wrong answer for a short list where the record is the act's condition rather than a note about it. Deleting a client with no line saying who deleted it is not a successful delete; it is the one outcome the trail exists to prevent. So thirty call sites in nine files now call <code>mustAudit</code> instead of <code>audit</code>: somebody's sign-in created, changed, switched off, handed the practice or sent a password link, their second factor cleared, their access changed and the stored-password key rotated (8); teams and portfolios (6); a company deleted with the clients orphaned by it, and a bank account added, changed or removed (5); integration keys (3); backups downloaded, emailed and restored (3); a client deleted, by either of its two doors (2); timesheet grants (2); and the practice-wide settings (1). The act and its line are one transaction for twenty-five of the thirty, which is the part that matters and the part that is easy to get wrong in a way that looks right — a line written outside the act's transaction is not the rule, it is the rule's shape with the opposite behaviour: the row goes, the line is refused, and the caller is told nothing happened. Three handlers were restructured for it, and one of those restructures fixed a latent fault of its own: minting an integration key could leave a key with none of its companies. The other five are on the pool on purpose with the reason written beside each — three backups, which hand a whole file over and cannot hold a transaction across it, and the password link and key rotation,

Date	Change
	which have already happened by the time the line is written. Payment accounts were the one thing on the list that was not recorded anywhere at all — no event name existed, so the three routes that change a company's bank details wrote nothing. It is the most attractive row in the database to change quietly: a swapped IBAN breaks nothing, shows on no screen anybody looks at twice, and prints on every invoice from that moment, and the only way anybody finds out is a client paying on time and the practice not receiving it weeks later. Three new names record the label, which boxes moved, and whether it was the account invoices start on — never the number, as the detail vocabulary carries field names and never their contents. What the person sees is one sentence wherever it comes from, answered as 503 rather than 500: it was not done, nothing changed, and tell whoever looks after the server. Four lines are deliberately left swallowed, each named with its reason: the integration read, which the accounting application polls continuously; the two credential lines, which are the super admin's ordinary daily work; and moving the closed-period line. Sealing is attempted outside the refusal, so a sealing fault cannot report a delete that did happen as one that did not. No migration. Handbook §12.5e new; manual §10.11

Date	Change
12 October 2026	The audit trail is sealed, so a line changed in the database shows up. The first point of the sixth step of the security audit, and the same chain the accounting application has had since 9 October. Nothing in this application has ever edited or deleted an audit line — that was already true, and it is written down in one place so it stays checkable — but nothing could tell you whether something OUTSIDE the application had. Now every line, a second or two after it is written, is given the next number in a chain and a fingerprint made from itself and the fingerprint of the line before it. Change a line afterwards and its fingerprint stops matching; take one out of the middle and the next no longer points at the one before. The fingerprint is keyed, which is the part that matters: a plain hash could be recomputed by whoever changed the line, so this is an HMAC under a key derived from the server’s own secret, which is in the server’s settings and not in the database. Each line records which key sealed it, so a trail restored from another installation reads as <i>sealed elsewhere</i> rather than as altered. And once an hour the newest seal is written to a small file on the server, outside the database — <code>storage/audit-anchors/</code>, beside the uploaded files, in the host’s nightly archive and never touched by a deploy. That is what catches lines deleted from the end, which the chain alone cannot see: a shortened chain is still a perfect chain, and nothing inside the database could tell, because whoever removed the lines could remove the count as well. Three things it deliberately does not claim: it makes a change visible rather than impossible; a line is sealed a second or two after it is written rather than as it is written, because chaining at the moment of writing would make every act in the app wait on the one before it — and a line left unsealed for five minutes is itself reported as a fault; and a gap the two-year keeping rule left is not tampering, because each line records when the line before it

Date	Change
	was written. A restore does not disturb any of it: <code>audit_log</code> is the one table a backup deliberately does not carry, so restoring a file leaves the trail alone. For now the answer is read on the diagnostics page under <code>auditTrail</code> — intact or not, how many lines are sealed, the number of the newest one — and the Security events screen that puts it in front of you is the next step. The first start after this seals every line already on the trail, once, in the order they were written. Handbook §12.5d new; manual §10.11

Date	Change
12 October 2026	The three applications now share one folder of security code, and five smaller things were put right with it. The fourth step of the 12 October security audit. The shared folder. The practice's three applications answer the same questions about signing in — what counts as a password, how a second factor is checked, what a link lives for, which headers every page carries — and a rule written three times drifts without anybody noticing. Those answers now live in one folder copied into all three repositories byte for byte, with three things keeping them that way: a manifest of hashes that fails this application's own tests if a file is edited here, a comparison against the other applications' copies on the same computer, and a pre-commit hook that refuses a commit changing the folder while the copies are apart. Invoicing uses the shared password rule, authenticator, passkey checks, session lifetimes and security headers from today. Five things it deliberately keeps its own are listed in the handbook with the reason for each, and the most important is the encryption key: the shared version refuses to run without <code>CREDENTIALS_KEY</code>, which this server has never had, so switching it over before that key is in the server's <code>.env</code> would stop the application. The link lifetimes changed with it: an invitation now lasts 72 hours rather than seven days, and a password reset 24 hours rather than one — both the agreed values, and both moved for the same reason even though one is shorter and one far longer. Two new account emails. Having your role changed and having your sign-in switched off both ended your sessions and both went on the register, and neither told the person it happened to. Both now do. The role message is two-sided on purpose — if it gave you <i>more</i> than you had and you were not expecting it, that is worth a telephone call today — and it is sent only on a change to the role itself. The switch-off message goes to an address that can no longer sign in, which is the point: it says nothing you

Date	Change
	did has been deleted, that the password link in your mailbox has stopped working, and to telephone rather than reply. Being let back in sends nothing. The first owner must change its password. The account the server's file creates on an empty database is the one door where a password is typed into a file, stays in that file, and sits in the deploy's history — so anybody who could read <code>.env</code> knew it until the owner happened to change it. It is now asked for at the first sign-in, like every other password somebody else typed. The copy taken before a restore is always locked now. Where there is no <code>BACKUP_PASSPHRASE</code> it used to be written in the clear, with a warning; it is now locked with a key the server keeps for itself and opens by itself on the Restore screen. Such a copy opens on this server and nowhere else, which is the trade, and setting a passphrase is still the better arrangement. And a client billed by two of your companies is now reached in the company you are working in, not in any company that happens to leave you unnarrowed — so a team that takes a client off somebody really takes it off them there. Two of the audit's points needed no work: client bank details are already covered by the same rule as payment-account bank details, and the account the server's file creates has only ever been created on a genuinely empty database. Handbook §12.7 new, §12.2a, §1.4 <i>Teams</i>, §12.3; manual §1.5d, §10.4, §10.6

Date	Change
11 October 2026	The administrator limit now holds, in the five places it did not. The first step of the 11 October security audit, and every one of the five was reproduced against a running server before being changed. One. Saving a role set the reach with it, so saving your own card with the role unchanged at <i>administrator</i> — which you already were — handed you every company in the practice. Making somebody an administrator now gives them no companies: it is two presses, Make an administrator and then Give them every company, and only the second one widens anybody. Making somebody staff still clears the reach, because that is a narrowing. Two. Nothing compared the administrator with the account they were acting on, so a limited administrator could set the password of an administrator who reaches everything and sign in as them — the limit lasted one request. Setting a password, clearing a second factor, sending a reset link, changing an email, changing a role and switching somebody off are now all refused on an account that reaches a company the caller does not, or holds a higher level in one they share. Three. Reach was compared as <i>do you hold this company</i> and never <i>at what level</i>, so a limited administrator who was a viewer in a company could make somebody an approver there and then be that somebody. They may now give up to the level they hold in each of their own companies and no further — and Save access on their own card is refused outright, because choosing your own level in your own companies is the same hole by a shorter road. The same applies to Teams and Clients they look after: somebody held to part of a book cannot take themselves off the team or clear their own assignments. Four. A linked-application key could be minted for <i>every company</i> by an administrator limited to two — a password that goes on working after the session, the sign-in and the employment that made it. Refused now, on making a key and on

Date	Change
	re-pointing one. Five. The timesheet screens were the one place that still waved every administrator through to every company; they read the same company ticks as the rest of the app now, and a viewer's tick reads a company's timesheet without keying time into it. Nothing changes for the owner or for an administrator who reaches every company — all five refusals return before they are asked, and the whole of the test suite walks every act twice to prove it. Handbook §1.4 <i>People and access</i>; manual §10.3

Date	Change
11 October 2026	A sign-in lockout now survives a restart, and the app notices one place working through a list. The eighth item of the second security review, and both halves are narrower than the words because the review of 2 October had already removed one of them and written down why. Lockouts were held in memory only, which was a deliberate choice — a row for every wrong guess would let somebody outside make the server write on demand, which is the same kind of fault the lockout exists to prevent. The real cost of that choice was never an attacker forcing a restart; it was that the practice restarts the server on every deploy, several times a week, and each one forgave every lockout in progress. So the lock is now written down — one row at the moment somebody crosses into a lockout, and still nothing for the four guesses before it — and read back when the server starts, with the escalation carried so a lock resumes at fifteen minutes rather than restarting at sixty seconds. Expired ones are swept at every start, because an outsider can make rows there five password-hashes at a time. And wrong sign-ins are now counted per place as well as per address. The per-address counter cannot see the commonest attack at all: twenty wrong passwords across twenty different accounts locks nothing, because no account reached five. At twenty wrong in an hour from one place, every further attempt from it is held back — half a second at first, growing to five — you are emailed once for the whole run, and one line goes on the register. Nobody is locked out by it, deliberately: an office shares one address, and locking that shuts out everybody sitting at it — which is precisely what the earlier version of this did and why 2 October removed it. Somebody at the next desk with the right password signs in normally while a stranger is slowed to a crawl. And a successful sign-in does not forgive the count, which was the other half of the old fault: one honest sign-in anywhere handed an attacker a fresh run of

Date	Change
	guesses. Handbook §12.2 <i>Wrong passwords, and being shut out</i> ; manual §1.1 and §11

Date	Change
9 October 2026	One company open at a time, and a window left alone stops being live. The seventh item of the second security review. Until now which company you were in lived only in your browser: every request named the company it was about and the server asked one question — do you reach it? So a single sign-in could work its way through every company you hold, one request after another, and a tab left open on yesterday's company, a bookmarked address, a back button or a request replayed by somebody who had your laptop for five minutes all looked exactly like you deciding to move. Each sign-in now has one company open. Switching is the switcher and is immediate; a screen that asks for a company this window is not on raises a box naming both and offering to switch, and saying yes finishes what it was doing rather than dropping you somewhere. And after thirty minutes with nothing happening the open company is closed — one button, nothing typed, and you are not signed out, which the box says because that is the first fear on seeing it. A window left open on a shared desk at lunchtime was a window still working at two o'clock. It is per sign-in, not per account: one company open on the desktop and another on the laptop, each with its own half hour, so a window being used never closes while another sits idle. It applies to everybody, the owner included — the people most likely to leave a window open on a shared screen are the ones with the most behind it. Nothing that is not about one company is affected: the people list, the audit log and the other registers, the reference data, and the list of companies itself, which you must be able to read to know what there is to switch to. Opening the app uses the company your sign-in has open rather than the one the browser remembers, or every return visit would start with a question. The thirty minutes is one of the values agreed across all three of the practice's applications, beside the invitation link's

Date	Change
	seventy-two hours and the reset link's twenty-four. Handbook §1.7 <i>One company at a time</i> ; manual §1.3

Date	Change
9 October 2026	And the limit now bites everywhere it should. The follow-on to the item above, done the same day. When an administrator could first be limited to certain companies, only one practice-wide screen had been taught to work out which company a request was about — a client’s stored portal passwords — and the handbook said plainly that the rest remained open. They were then enumerated rather than estimated, and sixteen were reproduced against a running server: a limited administrator could delete a company they cannot see, delete one of its clients, read and rewrite its Teams (which is to say, read which of its clients each of its people looks after), rewrite Clients they look after for it, hand somebody its timesheet, change its outgoing mail, repoint the practice’s own mail server — the one that sends sign-in codes and password links — and add companies to the practice. All sixteen are now refused, each with a sentence that says which kind of administrator the act wants rather than telling an administrator that only an administrator may do it. Adding a company and the practice’s own mail server are the two that are not narrowings but <i>widenings</i> of the requirement: they belong to the practice rather than to anything inside it, so they want an administrator who reaches all of it. A company’s own mail settings a limited administrator may still set, for a company they hold. The register is a known limit and stays whole: the audit log, the access log, the uploads log and the alerts name clients and companies a limited administrator cannot otherwise reach, and narrowing them means filtering rows rather than refusing a screen — its own piece of work, written up rather than left silent. The remaining forty-three administrator-only routes genuinely name no company, and each is now listed by name in the test suite with its reason, so one added next year cannot quietly arrive practice-wide — the test fails until somebody classifies it. Handbook §1.4 <i>People and access</i>; manual §10.3

Date	Change
9 October 2026	An administrator can now be limited to certain companies. The sixth item of the second security review. The database has carried a <i>reaches every company</i> flag on every account since the roles were rebuilt, and the access rule has read it correctly all along — but nothing could ever turn it off, because promoting somebody to administrator set it and no screen offered it, which made the limit a column rather than a choice. Two buttons now sit on an administrator’s card under Settings → People & access: Limit to certain companies... and Give them every company. A limited administrator still runs the practice — the people list, the register, adding an account, the reference data, none of which is about a company — and reaches only the companies they have been ticked into for anything that is: that company’s clients, invoices, timesheet and the portal passwords kept for its clients, which answer “<i>You are an administrator of certain companies only, and that is not one of them.</i>” They cannot widen anybody’s reach, their own included — giving themselves every company back is refused, and so is ticking anybody into a company they do not hold; that is the owner’s, or an administrator who already reaches everything. Both buttons end that person’s sessions and say so first, because reach is part of what a sign-in carries. The owner is never limited, whatever the column says. Two things to know before relying on it: nobody’s access changes on the day this ships — the role still sets the flag, so every administrator reaches everything until you say otherwise — and the narrowing bites only where the app works out which company a request is about. Reading a client’s stored portal passwords was taught to work that out, because it is what the review named; around twenty-five other administrator-only screens name no company and remain practice-wide for a limited administrator, among them deleting a company, deleting a client, Teams, client assignments, the practice timesheet

Date	Change
	screens and the outgoing-mail settings. Narrowing those is its own piece of work. Handbook §1.4 <i>People and access</i>; manual §10.3
9 October 2026	The copy taken before a restore is now locked, and does not sit on the server for ever. The fourteenth item of the second security review. Every restore writes a complete copy of the practice to the server first, so that restoring the wrong file can itself be undone — and that copy was in the clear, which means every client, every invoice and every password hash readable by anybody who reached the folder. It is now encrypted under <code>BACKUP_PASSPHRASE</code>, the same passphrase that opens your nightly backups. Three things were weighed in choosing that one. The incoming file’s passphrase is no good: the likeliest reason you are on the restore screen is that a file or a passphrase went wrong, so keying the undo to it would lock it in exactly the case it exists for. The server’s own key is worse than no good: it sits in the <code>.env</code> beside the file. <code>BACKUP_PASSPHRASE</code> is held by the practice, has nothing to do with the file being restored, and cannot be worked out from the machine. The review also suggested simply deleting the copy after a successful restore, and that was not available — a restore that <i>fails</i> changes nothing, because it runs in a transaction; the copy exists for a restore that <i>succeeded</i> on the wrong file, so deleting it on success removes the net at the moment it becomes the only one. Copies are also now deleted after thirty days as well as being capped at ten, because “the last ten” said nothing about how long one lasts. If <code>BACKUP_PASSPHRASE</code> is not set on the server the copy is still written, in the clear as before, and the server says so at startup — an undo copy that is not written is worse than one that can be read by somebody who already has the disk. Handbook §12 <i>Backups</i>; manual §10.4

Date	Change
9 October 2026	A new linked-application key has to say when it stops and what it reads, and revoking one now asks you to confirm it is you. The eleventh item of the second security review, and it finishes what KEY-1 started on 8 October. KEY-1 made a date and a list of companies possible and left both optional, so that nothing about the keys already in use changed — which was right for those and wrong for the next one, because it made the widest key this application can mint the one somebody got by not reading the question. Both are now required, of the server and not only of the screen: a default on a form is a control right up until somebody sends the request without it. A date already gone by is refused as well, since a key born expired is an afternoon spent wondering why the other application cannot read anything. The screen offers a date a year out, which you can change, and leaves the two company choices unticked so the answer is given rather than inherited. Revoking changed for a different reason: it was deliberately the one act there that asked for nothing, as the emergency brake, and what that overlooked is that a borrowed session using it breaks the link between the two applications with nothing on screen to say why — a revoked key is indistinguishable from a key somebody meant to revoke. The brake still works and costs one code. Nothing about the keys already in use changes: a key created before today with no date goes on reading, because this is a rule about what may be made. Handbook §1.4 <i>Confirming it is you</i> and §12 <i>Linked applications</i>; manual §10.4

Date	Change
9 October 2026	Four smaller things from the second security review, in one go. (1) The headers the browser is given are now the same nine in all three applications, with two added — one stops a window this app opens from reaching back into it, the other stops another website reading an invoice or a logo as a resource with your own cookie attached — and the HTTPS-only promise goes from six months to two years, and now covers every subdomain of kppanagi.com. That last part is a decision about the whole domain and cannot be withdrawn for two years once a browser has recorded it, so it is worth knowing: anything else on a kppanagi.com address has to be served over HTTPS. If the server's own settings file turns either of those down, the app now says so at startup instead of quietly obeying. (2) The app refuses to start on a placeholder session secret — <code>changeme</code> , a row of one character, a short block repeated, or the <code><48+ random chars></code> line from the deployment notes left unfilled. That one value signs every session cookie and, until <code>CREDENTIALS_KEY</code> is set, also encrypts every stored portal password, so a famous string there is the whole practice. It says which it found and prints the command that makes a real one. (3) <code>ADMIN_EMAIL</code> and <code>ADMIN_PASSWORD</code> in the server's file now name the FIRST account and nothing after it. They used to be read on every start, so changing that one line on a practice two years old created a brand-new owner account with the password from the file — a way in that left no line on the register naming who asked for it, because nobody did. On a database that already has accounts both are now ignored and the startup log says why. A database with accounts but no owner, which is what restoring an old backup gives you, can still have an existing account promoted — that path never created anybody and still cannot. (4) Changing somebody's email address now emails the OLD address and ends every session on the

Date	Change
	account — and switching a sign-in off cancels any password link already in that person’s mailbox, while the code and passkey steps of signing in started checking whether the account may sign in at all. Handbook §1.4 <i>When somebody leaves</i>, §1.4 <i>Confirming it is you</i>; manual §10.3; DEPLOYMENT.md

Date	Change
8 October 2026	Signing in no longer counts as “confirm it is you”. The fifth item of the second security review, and the change you will notice most. Ten acts ask you to confirm it is you at the moment you do them — reading a client’s portal passwords, downloading a copy of the practice, making a key another application reads it through. Typing your code at the sign-in screen used to answer that question too, so a session one second old had the whole list open for five minutes with nothing further asked. The two are not the same question: the code at the door says <i>who is at this machine</i>, once, for a session that may last a fortnight, while these ten ask <i>is the person at the keyboard still the one who signed in, and did they mean to do this?</i> The five minutes were never the complaint; spending a proof obtained for one question on a different one was — and it mattered most in the case the whole control exists for, a machine left signed in or a cookie lifted from one, where the window started at a moment somebody could choose by waiting. So the first of these acts in a sitting now asks, and here is the part to know before it happens: the code on your phone will be the one you just used at the sign-in screen, each code works once, and the box says so in as many words rather than claiming it is wrong. Wait the few seconds for the next one. It costs you nothing — a code already used spends none of the attempts that lock the codes and writes nothing on the audit log, because it is not somebody guessing; a code that is genuinely wrong still counts. Setting up an authenticator still counts as confirming it is you, because a password and a code typed on purpose for that act is a proof rather than a side effect of arriving. Nothing about the ordinary work of the practice changed. Handbook §1.4 <i>Confirming it is you</i>; manual §10.3

Date	Change
8 October 2026	A password you type for somebody has to be changed by them before they can do anything. The fourth item of the second security review: <i>“temporary password must be changed at first sign-in.”</i> There have always been two ways to give somebody a sign-in — email them a link, where they choose their own password and nobody else ever knows it, and type one for them, which exists because a link needs a working mail server and somebody standing in the office on their first morning is a real case. What was missing was the end of that second story: the typed password went on working for ever, so a password two people knew stayed a password two people knew, and whoever typed it could sign in as them months later leaving nothing behind but a sign-in line that reads like any other. Now such an account signs in, sets up its authenticator as everybody does, and is shown one screen — <i>the password you signed in with was typed for you by somebody at the practice, so two people know it</i> — and until it chooses one of its own, every route is refused. Three things about the shape of it. A link sets nothing and clears this, because the person who follows one chose the password themselves. Somebody setting their own password is not asked, by either route, so an administrator who changes their own is not marched to a screen about it. And the second factor still comes first — the authenticator is the lock and the password is what is behind it. Nobody who already uses the app is affected: how an existing password was set was never recorded, so nothing is assumed about one. Handbook §1.4; manual §10.3

Date	Change
8 October 2026	A client’s own person stays inside their one company, by every route. The third hole of the second security review: <i>“entity users widened by timesheet grants — POST /api/timesheet/grants does not check outsidePractice and needs no step-up.”</i> The marker’s whole meaning is one company, and the account screen holds that from both sides: the tick is refused on somebody holding two, and a second company is refused while the tick is on. A timesheet grant went round both. It is read beside the ordinary company grants by the same rule, so it was a second company in every way that matters — that company’s timesheet, its clients, its services and its rates — and nothing on that route had ever heard of the marker. It is refused now, in words that say why, and their own company’s timesheet can still be given to them, which is the ordinary use of it. Two more things went with it. Giving or taking that grant now asks you to confirm it is you, like every other change to what an account may do — it was the one of those that asked for nothing. And a client’s own person is no longer shown the practice’s email addresses: the line on a client’s window saying who looks after them gave every name <i>and</i> address of the staff on that client, from a screen they are meant to see. They are shown the names; the practice’s own people still see the addresses. Handbook §1.4 and §2.5; manual §8.2

Date	Change
8 October 2026	Taking and restoring a backup is now the owner's alone. The second hole of the second security review: <i>“an administrator can become Owner through a restore — POST /api/backup/import is open to any administrator and replaces the users table with the file's contents.”</i> It did. A restore does not merge with what is here, it replaces it, and the table it replaces includes users — which is where every “the owner alone” rule in this application lives. So an administrator could download the practice, change one word in the file, restore it, and be the owner. Two words in practice: the database allows exactly one owner, so the same file demotes the real one — and the end-to-end test that reproduced this watched exactly that happen, the owner coming back as an administrator with their session gone. Export, restore, the nightly schedule and Send one now are now the owner's, as they already are in Payroll and Accounting; an administrator is shown the card saying so rather than a button that refuses, and the refusal names the owner instead of saying “administrators only” to somebody who already is one. The alert settings and the check of the uploads folder on the same screen are untouched and stay an administrator's — neither takes a copy of anything. Handbook §1.4 and §12.3; manual §10.4

Date	Change
8 October 2026	A client’s records are now out of reach by their own address, not just absent from the lists. The first hole of the second security review: <i>“team limit bypassed by record id — a preparer limited to some clients can open, edit, delete or email any invoice or quotation in the company by its id.”</i> Correct, and worth being plain about. Teams narrowed every list when they arrived, and proved it: the client list, the invoice list, the quotations, compliance, the search box, the reports. What they did not narrow was the other way in. <code>GET /api/documents/<id></code> names no client, so the check that asks “is this client theirs?” had nothing to look at and let it through — opening, printing, editing, marking paid, emailing and deleting another client’s invoice all answered. The narrowing held against the screens and not against the address bar, which is the shape of a control that looks like protection; and an id is not a secret, since it sits in every list the person legitimately sees. Fixed in the one place it belongs: a record that hangs off a client now says so when the app asks what it belongs to, so the single check in the hook sees the client on every one of those routes at once — invoices, quotations and their sends, to-dos, money laid out for a client, recorded time and its drafts. A record naming no client — an invoice typed against a name, the practice’s own housekeeping to-do — is nobody’s and stays reachable, and anybody not on a team is unaffected. Handbook §1.4 <i>People and access</i>; manual §10.9

Date	Change
8 October 2026	“Why are you reopening this closed period?” is now the app’s own window, asked in one place. The last of the 2 October review’s smaller items: <i>“nothing drives <code>window.prompt</code> in a browser. It is a <code>window.prompt</code>, so the reason box, the ten-character refusal and the silent retry have never been exercised, on any of the sixteen routes that depend on them.”</i> True, and two of those things were wrong rather than merely untested. A reason under ten characters showed a toast and threw the save away — you had to find the screen, fill it in again and press again, and the toast saying why had faded by then; the window now stays open, says so in the box, and counts down the characters still needed while you type, keeping the work behind it. And a browser can switch <code>window.prompt</code> off: Chrome offers “prevent this page from creating additional dialogs” on the second dialog of a session, after which the question could neither be asked nor answered — you got the closed-period refusal with no way through it and nothing on screen to say why. It is now the same kind of window as “Confirm it is you”, which exists for the identical problem, a question asked in the middle of something. The ten characters are now the server’s number, sent with the refusal, so the box cannot accept a sentence the server then refuses — the same choice the “password for somebody else” window already made. And the lock date itself goes through the same window: Settings → Company details had a second <code>window.prompt</code> with a second copy of that ten, and now simply sends the change and lets the one question ask. Nothing changes about who may reopen a period, or about the audit line it writes. Handbook §12.1; manual §3.2

Date	Change
7 October 2026	Deleting the last invoice now gives its number back. Reported: issue an invoice, delete it, raise another, and the numbering had moved on — INV-029 was deleted and the next invoice was INV-030, with 029 lost for good. Every number in the app came from a counter per company, document type and year, and that counter only ever went up: the catch-up to the ledger was a GREATEST and the allocation was plus one, so a deleted number could never come back. The next number is now worked out as one past the highest number actually on a document in that series, which is what it should always have been — so nothing is skipped because something was deleted. All five series behave the same way: invoices, credit notes, receipts, pro-formas and quotations. A hole in the middle deliberately stays a hole — delete 027 while 028 and 029 stand and the next is 030, because a number already on a document somebody is holding is never approached from above and a ledger out of number-and-date order is one an auditor queries. Deleting a document that is past draft <i>and</i> carries the highest number in its series now says so on the confirmation: that the number will go to the next document, and that a copy of this one may already be with the client. Nothing is blocked. Two people saving at the same instant still cannot collide — the counter row is still locked before anything is read, so the second save waits and takes the number after the first; the counter simply no longer decides, and now follows the ledger downwards as well as up. Handbook §3.1; manual §2.1 and §2.10

Date	Change
7 October 2026	A key a linked application reads with can now be given a date it stops working, and a list of companies it may read — and the app says so before and after. The security review of 2 October, in one sentence: <i>“integration keys read every company and never expire”</i>. A key is a bearer token, so whoever holds the string is the accounting application as far as this one is concerned; it outlived the password of whoever made it, the session it was made in, and the person leaving the practice, and the only thing that stopped one was somebody remembering it existed. Settings → Linked applications now takes a date — <i>stops working at the end of</i> the day named, so a key dated the 31st works through the 31st — and either every company or a ticked list, with a key naming none refused rather than quietly made. Change gives an existing key a new date or a different list without minting a new one, which is what makes the expiry usable at all: otherwise “it expires on Friday” has one cure, a new key pasted in by somebody at both machines. A narrowed key is refused the other companies and is not shown them, and the refusal is decided before the company is looked up, so a key cannot enumerate the practice by the difference between “no such company” and “not yours”. An expired key is refused in the same words as a key that never existed, deliberately — which is why the notices matter: the accounting application otherwise brings in nothing and says nothing. So a key with a date is spoken about from a fortnight before and again once it has passed, in three places that all read one rule: a banner on the way in, a row on the Dashboard tagged <i>Linked app</i> (the one row there that is not scoped to the company in the switcher, because a key belongs to the practice), and an email — once per piece of news, not once an hour. A revoked key is never spoken about. Handbook §12.6; manual §10.8

Date	Change
7 October 2026	A client record no longer hands back the id of a company you were taken off. The second of the review’s smaller things about the client handler, and the last one that is an access question: <i>“a client record returns <code>companyIds</code> — every issuing company that bills that client — to anybody who can see the client... the one query in that handler that was left out of the AC-2 filter beside it.”</i> The two lists beside it have asked <i>whose money is this</i> since 2 October; this one answered <i>whose client is this</i> to anybody who could open the client at all. Ids only, and the names are not resolvable through the app, so it was never the worst leak there has been — it was the answer to a question nobody asked, and being taken off a company is supposed to make that company stop existing for you. The record now names only the companies you hold. It still says the record is shared and that you may not change it, because that is what the screen needs in order to explain a missing Save button — the fact, not the list. Nothing read the full list but the create form, which sends the company it is creating in. Handbook §4.2a; manual §4.6

Date	Change
7 October 2026	A client that two of your companies both bill can no longer be changed from one of them. The security review of 2 October, among its smaller things: <i>“an edit grant on one company can edit a client record shared with companies the caller does not hold — the membership is protected; the content is not.”</i> A shared client is one record: both companies see the same name, address, VAT number, statutory register and filing dates. The app asked only whether you held a company that bills them, so a preparer holding one of two could rewrite what the other company sees — and that company’s own people, who may not hold yours, had no way of finding out. Reading is unchanged and still needs any one of the companies. Writing now needs every one of them, and that covers everything belonging to the client: the record, the statutory register and its certificates, marking a filing done or reopening it, and the VIES check, which writes its result onto the record. One route is deliberately left alone — the agreed fee for a service, because a service belongs to one company and its terms are that company’s own. The client window says so before anything is typed: a shared client you may only read opens with a line at the top saying so, and no Save button, rather than a form that fills in and refuses. Nobody else is touched — the owner, an administrator reaching every company, anybody holding every company involved, and every client billed by one company, which is almost all of them. The end-to-end script was written first and run before the change, where it failed on exactly the four writes it now refuses. Handbook §4.2a, new; manual §4.6

Date	Change
6 October 2026	The app now says what is missing from its own database, by name. On 6 October the live server was found to be missing one table, <code>password_links</code>, while the migration journal recorded it as applied. Two things had been broken for five days and neither announced itself: the nightly backup, which is the only part of this application that reads every table and so the only thing that noticed; and Email a password link — the way somebody locked out gets back in, which would have failed at the one moment it was needed. What the backup reported was <i>“The request could not be completed... (reference 07b273d3)”</i>, because the error came from the database driver and those words carry client data, so they are deliberately never published. The fault was found by reading a table count off the diagnostics page and comparing it against a database built from scratch; that worked, and it took two people and most of a day. The app answers the question directly now, and in names. The nightly backup asks before it builds anything and refuses with <i>“The database is missing the table ...”</i>, which lands on Backup & restore where somebody is already looking. <code>/api/diagnostics</code> reports the same under <code>schema</code>. Columns are checked as well as tables, which is the half a count could never do: a missing column leaves the count correct and breaks just as hard, because every query this app writes names the columns it wants. It is deliberately not fatal — a missing column should not take the practice off the air. Handbook §12.5b and §12.5c, both new; manual §10.13

Date	Change
6 October 2026	A pass over both manuals against the app's functions, and the gaps filled. Every module, section, screen and server route was listed and checked against both documents. Four things the app does were in neither: Bulk Receipts on the Dashboard, the record of changes to figures, the key that protects saved passwords, and the diagnostics page. All four are now written up — handbook §3.6, §12.5, §12.5a and §12.5b; manual §10.10 to §10.13. Fifteen sections that named the buttons without explaining what turns on them were rewritten with the reasoning and a worked example: credit notes and what a part credit does to the balance, pro-formas and why Convert dates the invoice today, emailing and what actually goes with it, part payments and why an invoice is never marked paid by hand, deleting and why a credit note is usually the right answer, the five ways a variable fee is worked out, what each company setting actually decides, the logo, products and why a price rise never touches an issued invoice, adding a client and why Location is the field to get right, portal credentials, marking a filing done, quotation versions, light or dark, and what the search box looks through. One finding is for the practice rather than the documents: the deadline digest has no button. The server will email a company's outstanding filings to an address, and no screen offers it — so it is in neither manual, because documenting a button that does not exist helps nobody

Date	Change
6 October 2026	Teams: who looks after which client, and a client that is also one of your own companies. The security review's phase 4 — “teams, per-client assignment inside a practice-type company, and a client record linked to its own issuing company”. Holding a company used to mean holding every client it bills: their bank details, their fees, their correspondence, their filings. For a practice where one junior does six clients and another does the other fifteen, that was the whole book to each of them. Settings → Teams is new. A team is a name with people in it and clients on it: put somebody in it and they pick up the whole portfolio, move a client between teams and everybody follows at once. For the one-off — “she also does this client” — each person's card on People & access now has Clients they look after, on top of whatever their teams carry. The rule that matters is the one that does nothing: somebody with no team and no client of their own sees every client of the company, exactly as before. Nothing was narrowed on the day this went live, and nothing is narrowed until you put a client on a team. The other way round would have blanked every screen in the practice that morning. Once somebody is narrowed, every screen agrees — the client list, the invoices, the quotations, the to-dos, the compliance list, the search box, the dashboard, the reports and the report builder's export — and typing a client's address in by hand is refused outright rather than answered with an empty page. Three people are never narrowed: the owner, an administrator reaching every company, and the approver of that company, who stands behind what is prepared there and closes its months. A company somebody has no team in is untouched, so narrowing one company never leaks into another. Switching a team off gives its people the whole book back and says so when you press it, because a widening should never be quiet. Separately, a client can now

Date	Change
	be marked as also one of your own companies — the practice bills Kweller Ltd as a client while Kweller Ltd issues its own invoices here as a company, two records for one entity. It ties and does not grant: who may open that company is still decided on People & access, and one company can be claimed by one client record. Handbook §1.4, with two new sections and a row in the Settings table; manual §10.9 and §10.14

Date	Change
5 October 2026	A sign-in that belongs to the client, not to the practice. The security review’s phase 5 asks for “a client’s own staff signing in to one entity and never seeing the staff list or firm settings” — a director or an auditor of one of the companies you bill for, given a sign-in to see what has been invoiced to their own clients. Most of it was already done ahead of this phase: the practice’s staff list and the hourly rates on 2 October, and every screen narrowed to the company in the switcher on 5 October. What was left is the practice’s own settings, and that is this. A new button on their card — They are the client’s own person — and from then on Deadline rules and Defaults are neither offered nor served to them, nor are the practice’s kinds of work and its list of audit firms. Their own company is untouched: its details, products, clients, invoices and figures are exactly what their grant gives them, which is what the sign-in was for. Two rules, each enforced in both directions, because a contradiction otherwise waits for whichever order nobody guarded: such a person holds one company — the tick is refused on somebody holding two, and a second company is refused to somebody ticked — and an administrator cannot be one, nor can one be made an administrator. Ticking it signs them out, like a role change, because a narrowing that waits for the next sign-in has not happened yet. It is deliberately not a sixth role: the five roles are shared with payroll and accounting and say what somebody may DO, and this says who they ARE. Nothing changes for anybody already on the list — every existing account is the practice’s own until somebody is deliberately ticked. Handbook §1.4, with a new <i>A sign-in that belongs to the client</i>; manual §10.3

Date	Change
5 October 2026	One company at a time, everywhere — including for the owner. The security review asks all three of the practice’s applications to work on one entity at a time. Most of this one already did: the lists, the reports and the dashboard are all the company in the switcher. Two things were not. The search box searched every company you held — type a client’s name and you were told which other companies of the practice also bill them, with invoice numbers and amounts beside them. It now searches the company in the switcher and no other, and where nothing matches it names the company it looked in, so “nothing found” is never mistaken for “not in this system”; switch company and search again to look there. Choosing a result no longer switches company, because there is nothing to switch to. And a call that named no company at all was refused to everybody who holds companies and allowed to everybody who reaches them all — so the narrowing was real for a member of staff and absent for the owner and an administrator, which is the wrong way round. Every rule in the app that is about one company now insists on being told which, whoever is asking. Nothing on any screen changes for anybody: every screen already names its company. What changes is the client list fetched without one, which answered with the whole practice. What stays practice-wide, by decision rather than by oversight: the company list the switcher itself is built from, the people and backup screens, the settings that apply to every company, and the saved-report definitions. Handbook §1.6 rewritten; manual §1.3 and §1.6

Date	Change
5 October 2026	A company given to somebody <i>to look at</i> no longer shows them its bank details. The security review asks for a Viewer who reads and exports with sensitive fields hidden unless granted; the hourly rates were dealt with on 2 October, and this is the second field it names. A viewer now sees which accounts the company holds — the label, the bank’s name, the currency and the last four digits — and not the IBAN, the account number, the SWIFT code or the name the account is held in. Enough to tell two accounts apart and to see which one an invoice was paid into; not enough to pay into or to quote. The card says the details are withheld, so a short row is not mistaken for one somebody forgot to finish. It is the same tick-box as the rates, and that is the point: and the figures, beside <i>to look at</i> on Settings → People & access, now governs every sensitive figure this application has, from one rule written in one place rather than one per screen. Nobody who may CHANGE things is affected — a preparer, an approver, the owner and an administrator reaching every company all see the accounts whole, because withholding a figure from somebody who can change it is for show. An invoice is unaffected, deliberately: it prints the account it named, in the words it was printed with, and whoever can open it reads it as the client received it. What is withheld is the LIST — every account the company holds, including ones no invoice has named. Handbook §1.4 (new <i>What “and the figures” covers</i>) and §3.6; manual §3.1 and §10.3

Date	Change
5 October 2026	Five roles instead of two, and the owner is now a role rather than an address. This application had super admin and ordinary user, which the security review of 2 October called “too few and too flat”, and it asked all three applications to share one set: Owner, Administrator, Approver, Preparer, Viewer. Two of those say what somebody is in the practice and three say what they are in one company. Nobody’s access changed on the day: every super admin became an administrator reaching every company, and every <i>to change</i> grant became preparer. What is new is the two ends of the list. The owner is one person, held in the database itself — a second owner cannot be written — and nobody else may act on the owner’s account at all: not their password, not their address, not their second factor. That is what finally closes the hole found on 2 October, where an administrator could change the owner’s email, then their own to it, and walk into the owner’s seat in two ordinary saves; a role cannot be moved that way. Hand ownership to them is the new button on somebody else’s card, the owner’s alone: it asks you to type their name, signs everybody involved out, and leaves you an administrator. And at the other end, closing a month is now a grant you can give for one company. The third level beside each company reads <i>to change, and to close months</i> — the approver, the one person this application asks to stand behind a company’s books. The <i>Closed up to</i> card appears to the owner everywhere and to that company’s approver there; an administrator does not see it unless they are also made its approver, which is a narrowing and deliberate. Before this, the card belonged to whichever account matched <code>ADMIN_EMAIL</code> in the server’s <code>.env</code> — one answer for the whole practice, and nothing to do with the app’s own screens. <code>ADMIN_EMAIL</code> now does one thing only: it names the first owner on an installation that has none, and it can never overrule

Date	Change
	one afterwards. Handbook §1.4 rewritten, §12.1 and the <i>confirm it is you</i> list; manual §3.2 and §10.3
4 October 2026	A second way of proving it is you, from the first session. The fortnight’s grace is gone: a new account signs in with its password, reaches one screen — the one that sets an authenticator app up — and can do nothing else at all until it has one. That is what the payroll and accounting applications already did, and it was the last row of the security review’s account phase where this application was the one behind. The screen is in the sign-in card, before the app opens, in the same three steps as the accounting application’s: set up, then the key and the six-digit code it produces (with your password, because a first factor needs it — otherwise anybody who found your machine signed in could put their own authenticator on your account), then the recovery codes, which are shown once. There is deliberately no “later”: a button leading to a screenful of refusals is worse than no button, and the way out is to sign out. Nothing changes for anybody already enrolled. What held this back for a fortnight was not the application — it was one constant — but the test harness: the account all seventy-five browser suites share was un-enrolled on a fresh database, and eighty-six places in them signed in by filling the login form. Those eighty-six are now one shared sign-in used everywhere, which deals with a code being asked for, an account that must enrol first, and one already signed in; and the fixture puts an authenticator on the shared account with a secret the suites know. Four suites whose whole subject is an account without a factor now take that authenticator off again at the top and say why. Handbook §1.4 and §12.2; manual §1.2 rewritten

Date	Change
4 October 2026	The app now tells you when something happens to your own account. Five things already raised an email — a lockout, a second factor added or removed, a password changed, an account created — and every one of them went to the practice’s alert address and to nobody else. That is the wrong way round for the case they exist for: somebody who has your password uses it and then makes it theirs, the alert address is read when somebody thinks to read it, and the only person who can tell at a glance that a sign-in was not theirs is you. So five of them now also go to the person whose account it is: a sign-in from a browser the account has not been used from before (naming it — “<i>Firefox on Windows</i>”), a password changed, a second factor added, a second factor removed or cleared, and a lockout after the fifth wrong password, saying how long for and that the guessing did not work. The two voices are deliberately different. “<i>Your password was changed</i>” is a receipt and says there is nothing to do; “<i>Your password was changed by an administrator</i>” is a warning, says so in the subject line — which is all anybody reads on a telephone — and does not reassure. A browser is recognised by a long-lived cookie carrying a random id and nothing else: no fingerprinting, and the cookie’s value is never stored, only its SHA-256, so a copy of the table is not a set of working cookies. The first browser an account ever uses raises nothing — otherwise every first sign-in would arrive as a security message and the messages would come to mean nothing — and a browser is mentioned once. Clearing cookies or using a private window makes a browser look new again, which is the right way for this to be wrong. Three things it will not do: it never writes to an address that is not an account (the lockout counter counts the address whether anybody owns it or not, so sending on every lock would mail whatever an attacker typed and would answer <i>is this an address here</i>); an invitation raises nothing while a reset

Date	Change
	does; and it can never fail or delay a sign-in — the messages are not waited for, and a practice with no mail server gets no notices, no errors and a line in the log saying why. Handbook §12.2a new; manual §1.5d new
3 October 2026	The sections of a module are now a menu under its name. They were a second row across the whole screen, a strip of pills beneath the module names, and a strip has a far end nobody finds: Tools carried seven pills, so <i>What has been sent</i> sat off the right of an ordinary laptop and was reached by scrolling a row most people never noticed could scroll. Click a module's name now and its sections drop down under it, and a line with an arrow at its end has screens of its own, which open in a list beside it — point at the line or click it. That lets the long lists be grouped the way they belong rather than laid end to end: Invoices offers <i>All documents</i> and a New line holding <i>Invoice</i>, <i>Credit note</i> and <i>Pro-forma</i>; Tools offers <i>Quotations</i>, <i>Email</i> and <i>Currency</i>, the first two with their three screens under them. The row beneath the names is still there and now holds only the three things it should: Back, the name of the screen you are on, and the ★. Because a shut menu cannot say where you are, the line you are on is marked inside the menu as well. Nothing moved and no link changed — every screen is at the same address it was, bookmarks and shortcuts are untouched, and a menu is one click to open where a pill was one click to use, which is the cost of it. From the keyboard: the name opens the menu, the arrow keys walk it, → opens a line's own list and ← comes back out, ← / → on a top-level line move to the next module's menu, and Escape closes it, and the names themselves are now buttons rather than labels, which is what

makes any of that reachable without a mouse. On a phone a line's own screens open **inside** the menu, indented under the line you tapped, because beside it there is no room and a list that covered the lines below would cover the half of the menu you could then not reach. Whether there is room is measured at **the size the**

screens are drawn at rather than at the window's — at 150% a 1440-pixel window has the space of a 960-pixel one, and the menu follows the size the way every other width rule has since 30 September. This is also how the firm's accounting application has always worked, so the two now behave the same way. Handbook §1.1, §1.1a and §1.4c rewritten; manual §1.3, §1.3a and §1.5c likewise || 3 October 2026 | **Somebody leaving, and somebody losing their phone, are both things the app can now handle on its own.** Three changes to the life of an account, from the security review's account phase. (1) **A lost second factor can be cleared** — *Clear their second factor* on a person's card, under **Settings** → **People & access**. It signs them out everywhere and asks them to set up a new authenticator next time, and leaves their password alone, so somebody who has lost phone and password needs this and a password link both. Until now this was a job for whoever has access to the server's database, which meant a lost phone could stop somebody working for a day. It is refused on your own account, which has its own screen. (2) **An account is switched off, not deleted.** *Remove* has become *Switch off their sign-in*: every session ends at once, no further sign-in is accepted, and the account **stays on the list** with its name, the date and the companies it held. The audit log always survived an account being deleted — the address is copied onto every line as it is written — but it records what an *address* did, and “who was that, and what did they have” was written down only on the row that the deletion removed. *Let them sign in* puts it back. (3) **A role change, or a change to which companies somebody holds, now signs that person out.** Their access was always re-read on every request, so a narrowing has always taken effect immediately; what this adds is everything else a session carries — above all a *confirm it is you* proved minutes earlier under the role they have just lost. Saving the access screen without changing anything signs nobody out. Handbook §1.4 and §12.2; manual §10.3 || 2 October 2026 | **The hourly rates are no longer part of “give them the company to look at”.** A company handed to somebody *to look at* used to show them everything in it, the Timesheet's hourly rates and values included. That is right for a manager or a partner reviewing a company they do not key data into — and wrong for the other person who gets exactly the same grant: somebody inside one of the companies, given it so they can see what has been invoiced to their own clients. An hourly rate is what the practice charges for an hour, and it was reaching them because the only question the app knew how to ask was whether they held the company at all. So the grant now carries the answer: beside *to look at* on **Settings** → **People & access** there is a tick-box, **and the figures**. Unticked — which is how every existing grant starts, so this takes something away until you say otherwise — they still see every hour recorded against the company, and no rate or value. Ticked, nothing changes for them. Nobody holding a company **to change** is affected, because somebody who may alter the figures is not protected by being unable to see them. Two smaller things went with it: the list of everybody whose time is recorded was readable by **anyone** with an account and is now the super admin's, and the **Who** column on *Everyone's time* names only the people appearing in the rows on screen rather than the whole practice. Handbook §1.4 and §10; manual §8.2 and §10.3 || 2 October 2026 | **Three things closed after a security review of all three applications.** (1) **A company given to somebody to look at no longer becomes a company they can change** if they also have the timesheet grant. Each grant on its own was right — *to look at* refused every write, and the timesheet grant confined somebody to their own hours — but the two together let that person edit and delete **other people's** recorded hours, move an entry onto a different person, and type a rate over the agreed one. It was one question being asked of a rule written to answer a different one (“may this person be shown what an hour is worth” standing in for “may this person touch somebody else's row”), and those two only ever disagreed for that exact pair of grants. Seeing whose hours is unchanged: somebody given a company to review still sees everyone's time for it, which is the point of being given it. (*Seeing what those hours are worth changed later the same day — see the entry above this one.*)

(2) **Changing the email address on an account now asks you to confirm it is you.** One address on the server is the only account that may reopen a closed period, so an address is a permission rather than a detail — and moving one was previously as unguarded as fixing a typo, which meant any super admin could quietly take the owner's seat. A **name** still asks for nothing. (3) **A write now has to come from this application.** The session cookie already refused another *site*; it could not refuse another address on **this** site, and the payroll and accounting applications live at such addresses. Nothing about using the app changes — if you ever see “*This request did not come from this application*”, sign in again and tell whoever keeps the server. Handbook §1.4 and §10; manual §8.1, §8.2 and §10.3 || 2 October 2026 | **An invoice raised from a quotation now obeys a closed month, like every other invoice.** *Raise an invoice* on an accepted quotation makes a real invoice — its own number from the same counter, its own VAT working, its own line items — and it was the one way of making one that never asked whether the books were closed. So a company closed through today refused the invoice on the Invoices screen and issued the identical one from the quotation screen; the output VAT for a quarter already filed moved, and nothing said so. It is refused now, and where your account may reopen a period you are asked for a reason in the usual way, which is kept with the record. The date that decides is **the invoice's — today** — not the quotation's: a quotation sent in March and accepted in May is a May invoice. Quotations themselves are still never locked, because nothing on a quotation reaches the ledger until that button is pressed. Handbook §7 and §12.1; manual §3.2 and §6.3 || 2 October 2026 | **A closed month now refuses new hours as well as changes to old ones, and the timesheet's money and access leave a trace.** Closing the books stopped time already recorded in a filed month from being corrected or deleted — and left a door open beside it: a **new** entry could be written straight into that month, and a measured clock could be confirmed into it, which is the case that actually arises since a clock run a fortnight ago is confirmed today. Both are refused now. The hourly **rates** are deliberately not locked, because a rate is copied onto an entry when the entry is recorded and so cannot revalue an hour already booked — which also means correcting a rate does not correct hours already booked at the old one. Separately, four things in the timesheet used to leave **no trace at all**: setting an hourly rate, removing one, and giving or taking away the permission to record time. All four are now on **What has been done, and by whom**, naming who a rate was for and from what date — never the amount, which is on the rate card. And setting the same rate twice now says so (“*A rate for ... already exists*”) instead of failing with an internal error. Handbook §10, §12.1 and §12.5; manual §8.1 and §8.2 || 2 October 2026 | **Refreshing a screen no longer loses your place, and screens can be bookmarked.** Every screen has put its own address in the bar since the navigation moved to the top — which is what makes **Back** work — and then a reload threw it away and opened the Dashboard. So pressing refresh lost the screen you were reading, a bookmark to **Clients → Deadlines** opened the Dashboard, and an address sent to a colleague did not take them where you were. It now opens the screen the address names. The four addresses that changed when the navigation moved still land on the right screen and are rewritten to today's form as you arrive, costing no extra press of Back. **Signing in is the deliberate exception** and still opens the Dashboard whatever is in the bar: the address left behind when a session ended belongs to whoever was last at that desk, and the Dashboard is the screen that says what needs attention. Handbook §1.3 and manual §1.3 || 2 October 2026 | **The Currency screen now says whether its rates can be trusted, at the top, every time.** It knew all of this and said it in the wrong place: how old the held rates were — and that a download had failed, if one had — went into a card headed *Where these come from* at the **foot** of the page, under the converter and under the whole thirty-row table, where nobody who has just read an answer off the top of the screen ever arrives. The line is now the first thing on the screen. When all is well it is quiet and still names the publication date (“*Rates*

published yesterday (01/10/2026)”) — a converter that looks identical whether its figures are from this afternoon or from last March is one nobody can trust. It turns into a warning for four things: **the download failed** (with what the server got back), **nothing has tried for over a day** (the job runs hourly, so it has stopped or the server has been off), **the newest rate is more than four days old** while downloads are working, and **nothing has ever been downloaded**. The first of those is the one that used to be invisible: the Bank publishes at about four in the afternoon, so a download that broke this morning leaves yesterday’s rates sitting there looking perfectly ordinary — which is why **a failing download now outranks old rates** when both are true. The old line at the foot of the page is gone: one statement, not two that could disagree. Handbook §6 and manual §3.1 rewritten || 2 October 2026 | **A pass over both documents against the screens — thirteen corrections and two gaps**. The handbook claimed to cover the app module by module and had **no Timesheet chapter at all**; §10 is now written — that a person and a sign-in are different things and both are needed before an hour can be recorded, that time with no rate records at **zero** rather than being refused, who sees whose hours and whose rates, and what the form refuses (one day per entry, HH:MM either way round, invoiced time frozen). **Deadlines: there are two floors, not one**. A client’s own **Client since** date hides anything that fell due before you took them on, whichever of it and **Track deadlines from** is later — the commonest reason a deadline somebody expects is not on the list, and neither document said so. **The Currency screen** is described properly: Convert on a date, Look up a rate with Recent days, Fetch the latest rates, and that there is no rate for a weekend. **Being shut out after wrong passwords** is written up for the first time — four free, then a minute, five, fifteen, counted per address. Corrected: the **search box** that has been a magnifying glass since 27 September (in *both* documents); **Settings → Company**, twice in the handbook and once inside a quotation of the app’s own VAT card; the VAT-note table, which described one note that does not exist and omitted Article 138 for goods entirely; the two revenue reports, which take no client (use **Build a report** for one client) and only one of which has a grouping box; **Linked applications**, missing from the Settings table; **Add a passkey**, which is **Register this device**; *Paid on their behalf*, which is always shown; “Four things... easy to miss” followed by three; a test-alert refusal that cannot appear because the button is greyed out; and two messages in Part 11 that left with the bank-statement import. **A convention, now stated and tested**: an arrow is a path through this app, a chevron is another program’s menus. The test that has checked the screens’ own signposts since September now reads both manuals too, and the handbook’s list of Settings sections is checked against the app’s. Handbook §1.4, §2.1, §4.2, §5.1, §6, §10 (new), §11, §12.2 and §12.3; manual §1.1, §1.2, §1.6, §2.9, §3.1, §5.3, §9.1, Part 7 and Part 11 || 1 October 2026 | **Nobody has to be told anybody else’s password any more**. Adding somebody used to mean typing a password for them and passing it on — fine while everybody works down the corridor, useless for an account at a client’s office, and it leaves the password in a telephone call or a note. **Settings → People & access → Add somebody → Email them a link** now creates the account with **no password at all** and sends them a *Set your password* link: they choose their own and nobody here ever knows it. The link works once, lasts seven days, and sending another kills the first. A password the practice’s rule refuses does not use the link up. On each person’s card, **Email a password link** does the same for an account that already exists, lasting an hour — the control both documents have described for weeks and the app did not have, which was the sharpest of the documentation faults because it is the instruction for what to do when somebody leaves. Pressing it changes nothing until they use it, so their current password keeps working while they are still at their desk. **Set a password myself** keeps the old way for when the app’s own mail server is the thing that is broken. Everything else holds: a reset signs them out everywhere, their second factor is untouched and still asked for, both acts ask you to confirm it is you, and both are on the register — including

whether the message actually left the building. There is deliberately **no “forgotten my password” box on the sign-in screen**: it would let anybody make the app send mail to a member of staff, and tell them which addresses are accounts. One setting is needed on the server first (`APP_ORIGIN`); until it is there the button refuses and says so rather than sending a link pointing nowhere. Handbook §1.4 extended; manual §1.1 and §10.3 rewritten | | 1 October 2026 | **Three gates put on the right acts. Deleting a company** now asks you to **confirm it is you** before it runs — it destroys every client only that company holds, and with them the government-portal passwords held for those clients, which is exactly what deleting a *client* has always asked about. It also writes **one line per client** on the register, naming each and saying whether passwords went with it; before, it wrote a count, so the record said two clients had been destroyed without saying which two and the rows were gone. **Changing the mail server a company sends from** is now the super admin’s, and asks for a code at the moment of saving: anybody holding one company could previously repoint every invoice, statement and reminder it sends through a server and from-address of their own choosing, silently. Seeing those settings, and pressing **Send test**, are unchanged. And the ★ on a screen now works for somebody given a company to **read but not change** — both manuals said it did, and the server was refusing it. Handbook §1.4 and §12.4 extended | | 1 October 2026 | **Four things that used to fail quietly now say so.** A backup between 10 and 18 MB was emailed and reported as successful every night and **could not be put back** — the restore refused it for being too large, and you would have found that out on the day you needed it. Restoring now accepts comfortably more than anything the app will send, and the card shows the last file’s size against both ceilings. When a backup does fail, the notice goes out through **the same mail server the backup uses**, so the commonest failure takes out both; the refusal used to be discarded and nothing was left anywhere. It is now recorded, and a **banner** tells the super admin on the way in — including whether the email got out. **Settings** → **Backup & restore** and `/api/diagnostics` both carry it. Diagnostics also answers the two questions anybody actually asks about a sick server — *did last night’s backup run* and *is the schema current* — by name, along with uptime, memory and free disk. And a migration that had been **edited after it ran** was being run a second time: it failed with *relation already exists*, every later migration was skipped in silence, and the server started anyway so the deploy stayed green. Migrations are now recognised by their name rather than by the file’s contents, an edit is reported rather than re-applied, and a database migrated by the old code is matched up by the order things ran in. Handbook §12.3 and manual §10.4 extended | | 1 October 2026 | **Saved reports are now for the people who can run one.** The list was visible to everybody signed in, and a saved report carries the **values you filtered on** — which in practice is a client’s name or an amount. So somebody holding nothing but a timesheet grant could read every filter in every saved report in the practice. The list is now shown to anybody with **Can change** on at least one company, plus the super admin — the same people who could run the report and read the figures anyway — and to nobody else. Nothing about running one has changed, and nothing was deleted. Separately: a report whose author has **left the practice** was editable and deletable by anyone, because the account had gone and the report was left belonging to nobody, which the app read as belonging to everybody. It now belongs to the super admin, who is told so in the refusal. Handbook §9.1 and manual §9.2 rewritten | | 30 September 2026 | **The size the screens are drawn at is now yours, and it follows you.** An Aa beside the light/dark button offers **90%, 100%, 110%, 125% and 150%**, and what you pick is kept **on your account** rather than in the browser. That is the point of it: staff sit at whichever desk is free, and a size set in a browser’s own zoom does not travel — somebody who needs 125% was setting it again on every machine, for ever. Now it is right the first time they sign in anywhere, and on a machine they have used before it is applied **before the page is drawn**, so nothing resizes a second later. It is on the bar rather

than in Settings for the same reason: the browser's zoom is behind a menu a lot of people have never opened.

Nothing printed changes — an invoice, a statement or a quotation comes out identical at 150% as at 100%, on screen, in the PDF and on paper. **90% is on the list on purpose**, for a small laptop and a wide document list. If an account has never chosen, whatever the browser remembered is cleared, so a shared desk does not greet the next person at 150% with no explanation. Three smaller things arrived with it: the app now has a **favicon** — its own  mark in the browser tab; **Tab** on a fresh page reaches a **Skip to the screen** link that jumps past the whole bar, which is the longest of the three applications'; and with **Reduce motion** turned on in the operating system, every transition and animation stops, with nothing hidden and no layout changed. Handbook §1.4b and §1.4c new; manual §1.5b and §1.5c new | | 28 September 2026 | **Favourites — a column of shortcuts down the left.** The bar at the top holds seven names and everything else is a section under one of them, which is right for a screen opened occasionally and tiresome for the four or five opened every day. So the ★ at the right of the section strip keeps the screen you are on, and what you keep appears down the left: click one to go straight there, press the star again or the × on the row to take it off. The list is **each person's own and per company**, so it follows the switcher and staff do not see each other's. It is kept on the server rather than in the browser, so the same shortcuts are there on another machine. A screen that stops being offered is not drawn and its shortcut is not destroyed — set a company back to a practice and its Deadlines shortcut is there again. Somebody given a company to read but not change may still keep shortcuts to it. On a phone the column becomes a third scrolling strip under the sections. Handbook §1.1a new; manual §1.3a new | | 27 September 2026 | **The menu has moved to the top of the screen, and ten modules are now seven.** The rail down the left is gone, with the **Keep open** pin and, on a phone, the menu button and the panel it slid out. In their place: the module names across the top, and the sections of whichever one is open as a strip of pills beneath — the same two levels, laid out along the screen rather than down it, which gives every page the 238 pixels the two columns were holding. Seven names fit where ten did not, so four modules became sections of another: **Compliance** is now **Clients** → **Deadlines**, and **Quotations**, **Email** and **Currency** are the three screens under **Tools**. **Nothing was removed** — every screen those four had is a pill on the strip, at the depth it was before — and **every link saved before today still works**: an old address lands on the same screen and is rewritten to the new form as you arrive. The deadline count that sat on the Compliance icon now sits beside **Clients**, so it is still visible from anywhere. **Search** gave up the box that held a third of the bar and is a magnifying glass at every width (**Ctrl+K** still opens it); **Help** is the ? beside it. On a phone the module names take a line of their own and scroll sideways, as the section strip always did. Handbook §1.1 and §5.2 rewritten; manual §1.3, Part 3, Part 5, Part 6 and Part 7 likewise | | 27 September 2026 | **Four unused tables removed from the database.** Every step of moving the books to the accounting application ended “nothing was deleted from the database”, deliberately — a screen comes back in an afternoon and a dropped table does not. Those removals have settled, so the tables that nothing can read any more have gone: the closing balance captured from a bank statement, the record of a month signed off as reconciled, what was paid against a captured supplier invoice, and the monthly budget per category group. **Bank accounts, the vendor name book and the captured supplier invoices all stay** — receipts still name the account they arrived in, the supplier invoices name uploaded files, and the vendor book is still pointed at by past entries. A backup taken before today still restores; the four tables in it are simply skipped | | 27 September 2026 | **One list of accounts, not two, and the Money module is now just Currency.** A client is told where to pay on the invoice, and that comes from **Settings** → **Company details** → **Where clients send the money**. A second list, **Money** → **Bank accounts**, was left over from when this app kept the books: the ledger coded against it, the statement was imported for it and the

reconciliation measured it. With all three gone its only job was naming which account a payment arrived in — the same bank the practice was already keeping under Settings, typed twice. **Record payment now offers the accounts an invoice prints**, opening on the default; check it, because the default is where you asked them to pay rather than proof of where it arrived. *Cash* still records a receipt with no account. Bank accounts and their screen are gone and the rail now reads **Currency**. **Nothing was lost**: receipts recorded before today keep the account they named and still show it, on the receipt and in the report builder's **Money received** source. Handbook §6 and §3.6 rewritten; manual Part 3 and §2.7 likewise || 27 September 2026 | **The expense ledger, the categories and the vendor name book have left this app, and Cashflow is now called Money**. The module kept two things: **Bank accounts** and **Currency**, the European Central Bank rates. (The line first published here said bank accounts were what an invoice prints as the place to pay. That was wrong — an invoice prints a *payment account*, kept under Settings — and the row below corrects it.) Gone with the ledger: adding or editing a cost by hand, the month selector and its tiles, expense and income categories, budget groups, the **Vendors & customers** name book, and the clearing of a whole month. Global search no longer offers cashflow entries. **Money laid out for a client is unaffected** and is recorded on the client's record (**Details** → **Paid on their behalf**); on an older entry of that kind which carries VAT the amount can no longer be changed, because the VAT on it is what was actually charged — the date, who it was paid to and what it was for can still be corrected. Handbook §6 rewritten; manual Part 3 likewise. Nothing was deleted from the database || 27 September 2026 | **Money laid out for a client is now recorded on the client record**. A filing fee or a levy paid on a client's behalf is a receivable that happens to start as a payment, so it is kept where the invoice is raised: **Paid on their behalf** on the client's Details pane, with what has come back and the difference. It is not a cost of the practice and no VAT is reclaimed on it. Entries made before this are unchanged and still appear — this is the same record with a way to add to it. The alternative was a second link to the accounting application, which was not taken: it would have made the client record go blank whenever the other app was down || 27 September 2026 | **Budgets, director's accounts and the spending reports have left this app**. Reports loses **Where the money went, Director's accounts** and **Budget**; the consolidated report across companies goes with them. All four were reports on the books, and the accounting application holds the books now. **Directors and shareholders are still kept here**, on the statutory register — it is their running account with the company that moved. Cashflow's month selector now applies to the Ledger alone. Nothing was deleted from the database || 27 September 2026 | **Purchase invoices and aged payables have left this app**. The capture queue, the reading of what suppliers send, posting them into the ledger, supplier payments and the aged payables report are all in the accounting application now, and the report builder no longer offers a supplier-invoice source. Costs already posted from an invoice **stay on the ledger**: their VAT still cannot be recomputed — it is what the invoice charged, not a percentage — but they can now be voided, which the old refusal did not allow. Handbook §6 rewritten, and what stood at §6.3 is gone with the screens it described; manual §3.1–§3.2 likewise. Nothing was deleted from the database || 27 September 2026 | **The bank statement import and the reconciliation have left this app**. Cashflow no longer takes a bank file: no statement import, no column mapping, no matching bank lines against invoices and costs, and no agreeing a month against a closing balance. The accounting application does the bank side now. **Costs are entered by hand** on Cashflow → Ledger, which is unchanged, and **recording a payment against an invoice is unchanged** — it still names the account the money came into. Handbook §6 rewritten — what stood at §6.3 and §6.4 went with the screens, and §6 now carries no numbered subsections; manual §3.1–§3.3 likewise. Nothing was deleted from the database || 26 September 2026 | **The VAT return working and the VIES statement have left this app**. The accounting application

prepares both now, from the purchases and the sales together, which is the only place a return can be prepared correctly once the purchases live there. Reports loses **VAT working — purchases** and **VIES. Nothing about VAT on a sales invoice has changed**: the rate, the reverse-charge decision and the wording printed on the invoice face are all worked out and snapshotted here exactly as before, and **Check with VIES** on a client record still checks their VAT number against the Commission's register — that is a different job on the same word. Nothing was deleted from the database || 26 September 2026 | **The key the accounting application reads with is now treated as what it is**. Making one asks you to confirm it is you — a key goes on working after the password is changed and after the session that made it has gone, which puts it in the same class as downloading a copy of the practice; it is the ninth act to ask. Making one also **sends an alert**, at once, the way a new account does. And **using** one now leaves an audit line naming the key, the range of dates asked for and how many documents went: the *Last used* column holds one date and is written over, so it could say a key was alive and never what it had taken. Revoking is deliberately left quick and unguarded — it is the emergency brake. The screen itself was in neither document; new \$12.6 of the handbook and \$10.8 of the manual || 26 September 2026 | Changing the second factor now ends your other sessions **however you change it**. Confirming or removing an authenticator app already did; **registering or removing a passkey did not**, which was the half of it most people use — so somebody told a device had appeared on their account could remove it and leave the session that added it still open. All four now end the others, say how many, and record it || 26 September 2026 | The linked-applications key table was in the schema and in no migration, so a **restore onto a fresh server** came up with that screen failing. It is now created by a migration like everything else, and the existing server passes over it untouched || 26 September 2026 | **A pass over both documents against the screens — 34 corrections**. The manual gains the **Dashboard**, which it had no section for at all, and the handbook's to-do notes gain the **Undo** that lasts one redraw. Named for the first time: the **Status** field's neighbours on an invoice (the typed-over number, the per-document *From* card, the Payment Details override), that an unfinished working pins a document to draft, the amber **check this VAT** warning, the 500-row cap and the *This year* the list opens on, the category that starts a goods line, the client record's own section headings, **Show past entries** and **Documents & certificates** on the registers, the fourth Compliance filter and the **Next 3 months** it opens on, why some deadline rows have no box to type in, that the deadline rules reach **every client of every company**, the empty fee list and its **Copy the list from...**, that a quotation bills only once, the report builder's real starting filters (clients start at *Active is Yes*) and its limits, who may change a saved report, what a missing hourly rate does, the per-entry rate override, that invoiced time is frozen, the two different **Add person** buttons, and the real names of the backup and alert cards. Corrected: the VAT-note table listed a *Standard rate 19%* message that does not exist || 26 September 2026 | **Time spent in the payroll app now reaches this timesheet**. Opening a client's payroll there starts a clock, and on leaving it that application asks what the time was for — nothing crosses without a line saying so. It then appears under **Timesheet → My time** marked *payroll*, with that sentence as its note, and is confirmed like any other measurement. Employers are matched to clients on the registration number then the tax code, never on a similar name; anything that cannot be matched is held in the payroll app and put to the super admin as a question until it is answered. New \$10.2 of the handbook and \$8.4 of the manual || 26 September 2026 | **Opening a client's record now times it**. Closing the record, switching company or signing out stops the clock, and what it leaves is a measurement waiting on **Timesheet → My time** — never recorded time, because a clock cannot know what kind of work an hour was. A record left open stops counting half an hour after the last thing anybody did, anything under five minutes is not offered at all, and a browser that crashed within a minute of opening a

client leaves nothing rather than half an hour that never happened. New §10.1 of the handbook and §8.3 of the manual || 24 September 2026 | **The alerts and nightly-backup cards now notice when the app has no mail server.** Each used to report only its own half, so both could show green while nothing could send. And Cashflow → Categories & vendors now says which of its lists belong to the company on screen and which are shared by all of them || 24 September 2026 | **Both mail cards check the settings themselves.** A web address pasted into the server box is refused on save and told what to type instead; a port and an encryption that disagree raise a note as they are chosen. Both used to be found only by sending a test and reading the mail server's own refusal || 24 September 2026 | **The nightly backup was not running, and the screen could not say so.** The jobs that run without being asked sat on an hourly timer that only starts when the app does — and the server stops the app when nobody is using it, so at two in the morning there was no timer. They now run once at boot as well. Separately, “last one ... (today)” counted elapsed hours rather than calendar days, so a backup from yesterday morning read as today's right up until the same hour came round || 23 September 2026 | **The import warns when the same lines are already under another bank account.** The account is part of what makes a line recognisable, so a statement re-read against the wrong account looked entirely new and doubled the month with nothing greyed and nothing said. The preview now names the account the lines are actually in || 23 September 2026 | **Reverse charge at 9% and 5% can now be chosen.** The server had computed and tested all three rates since it was written; only the category picker was never told, so a reduced-rated intra-EU acquisition had no treatment but the 19% one. The existing entry now reads **Reverse charge (19%)** so the three read as a set, and handbook §6.2 lists every treatment || 23 September 2026 | Six more corrections: which reports actually have an export and a period (four have neither, VIES has **Print**); the two Reports screens that ignore the Cashflow month; the client sidebar, which does not hold **VIES check** or **Paid on their behalf** — both are inside Details; and closing a period, which belongs to one email address on the server rather than to the super-admin role, with the prompt that lets a single change through without moving the lock || 23 September 2026 | **The super-admin role is written down** — how it is given and taken, that there can be any number but never none, and that resetting a password ends every other session. With it, the thing nobody had said: **nobody can clear anybody else's second factor**, so recovery codes are the only way back into an account, and a practice with one super admin is one lost phone from needing the server. The step-up list is corrected too — eight acts, not three, lasting five minutes || 23 September 2026 | **A statement closing balance can be corrected.** Once one was on file the cell was a number with no box and no button, so a mistyped figure left the month out for ever and the only remedy was to re-import the statement — the server had always accepted a correction, the screen simply never offered one. **Change** beside the figure now opens it || 23 September 2026 | **Correcting a figure on a posted supplier invoice** — the ledger said to change it on the invoice, the invoice said to change it on the ledger entry, and neither would. Both messages now give the actual route (undo payments → Take out of the ledger → correct → post again), and the manual has it as steps || 23 September 2026 | **The app has two mail servers and only one of them was written up.** New §10.7 of the manual covers Settings → Defaults, which had no section at all, and both manuals now say what nothing said before: with *Mail from the app itself* unset, the nightly backup, every alert and emailed sign-in codes fail silently, with nothing on screen to say so. Also: the nightly backup's passphrase is not the one you type on Download || 23 September 2026 | **Nobody could record time without knowing two things nobody had written down** — a staff record linked to the sign-in, and at least one kind of work. New §8.1 of the manual and a new opening to §10 of the handbook; the screen now tells the super admin to add *themselves* rather than to go and find one; and both manuals corrected where they said “may record time for” was enough on its own ||

23 September 2026 | **The VAT working is one month** and a Cyprus return is a quarter — new §9.3 says so, with what decides the month a figure falls in and why reverse charge appears on both sides || 23 September 2026 | **Importing the same statement against a different bank account brings every line in again** — the account is part of what makes a line recognisable. Both manuals said re-importing was always harmless || 23 September 2026 | Expense categories, income categories, budget groups and the Vendors & customers book are **shared by every company**, not per company as §1.2 said || 23 September 2026 | **Removing a client** — the box now says what it can destroy. In one case (their only company, no documents) the client is deleted outright with their portal passwords and statutory register; the manuals said the opposite, and so did the box || 23 September 2026 | **Draft or issued** — both manuals said *Save Invoice* issues an invoice. It saves whatever the **Status** field says, and that field starts on *Draft*. Corrected, with what to look at when an invoice is missing from the figures | 23 September 2026 | Six places where the app told you to go and look at a screen that does not exist by that name — “Staff & rates”, “Settings → Services”, “Settings → Company” — now name the real one, and a test checks every such signpost against the screens themselves || 23 September 2026 | The hint on a quotation service said to type a CSS variable where the transaction allowance goes. It says `{txn}`, which is what it always should have || 23 September 2026 | An alert email now sends you to the card as it is actually headed on screen — “What has been done, and by whom” — instead of to an “Activity” that was never there || 23 September 2026 | The alerts list on Backup & restore now says how long each alert stays quiet after it has sent, instead of calling every one of them “every time” || 23 September 2026 | A missing file is now emailed to whoever keeps the server as well as written to the log — once per file per day, so the archive it can be recovered from is still in the window || 22 September 2026 | **Light or dark**: a button beside the company switcher, following your computer’s own setting unless you say otherwise. Documents, PDFs and anything printed stay light. The Help pages follow the same choice || 22 September 2026 | **Put the letterheads right** on the file check: one press stamps the company’s current logo on every document naming a letterhead that is no longer on the server, and leaves a merely older one alone || 22 September 2026 | Uploaded files: **Check the files** on Backup & restore compares the uploads folder against the records in both directions and deletes nothing; supplier invoice scans and the letterhead each document was issued under are audited at last (both were on no list); a missing file is written to the server’s log the day it happens; and the pre-restore copies moved out of the nightly off-server archive || 22 September 2026 | Eight things the screens said or did not offer: the password rule on screen now says 12 characters, as the app has always required; the filings page is headed Compliance, as the rail always called it; a client record records **Audited by**; a client’s statement can be emailed from the report; a pro-forma is called one throughout and stays one when edited; **Open** works on the aged payables report; **may record time for** grants the timesheet on its own, company by company; and products can be edited and deleted || 22 September 2026 | Search: a search box in the top bar (Ctrl+K) that finds clients, documents, quotations, supplier invoices and Cashflow entries across every company you hold; and a search box on the Help pages covering both documents || 22 September 2026 | First edition of the handbook and user manual, covering the app as it stands: dashboard with income charts and to-dos, invoices, clients, compliance, cashflow with statement import and supplier invoices, quotations, circulars, fixed reports and the report builder (ten sources), timesheet, settings, security and backups |