

Users & Roles

Admin Invite people, set what they may do, and switch them off, without ever deleting their work.

Screens Users & Roles `users-roles.html` Invite `invite-user.html` Email settings `email-settings.html`

Writes the person profile the trail `audit_log`

Key fields what they may do `profile.role` switched off? `enabled` handle `username` account email `email`

the invite wording `email_template`

1 · Given / When / Then

Human wording first; the exact table/field in `mono`.

GIVEN The signed-in person is an **admin**. In ChatMaestro the admin role is the install owner; there is no separate owner record. Sign-in is passwordless (*Sign In*): an invitation lets the person authenticate with a provider or an emailed code, so there is no credential to set here.

WHEN On the **Users & roles Screen** they invite somebody, which sends an invitation driven by `email_template`, change an existing person's `role`, or deactivate an account by setting `enabled = false`.

THEN A `profile` row is created or updated, and the change is written to `audit_log` with `before_state` and `after_state`. Role is what row-level security, the database's per-user access rules, keys off, so a role change takes effect on the person's very next query, with no redeploy and no cache. An invited enrollee gets a handle from a running counter (for example `enrollee-0001`); staff handles are readable. The handle keeps names off the screens, but it is sequential and stable across studies, so it is not anonymity — the blinding the platform enforces is `experiment.blinded`.

writes `profile`, `audit_log` **reads** `email_template`

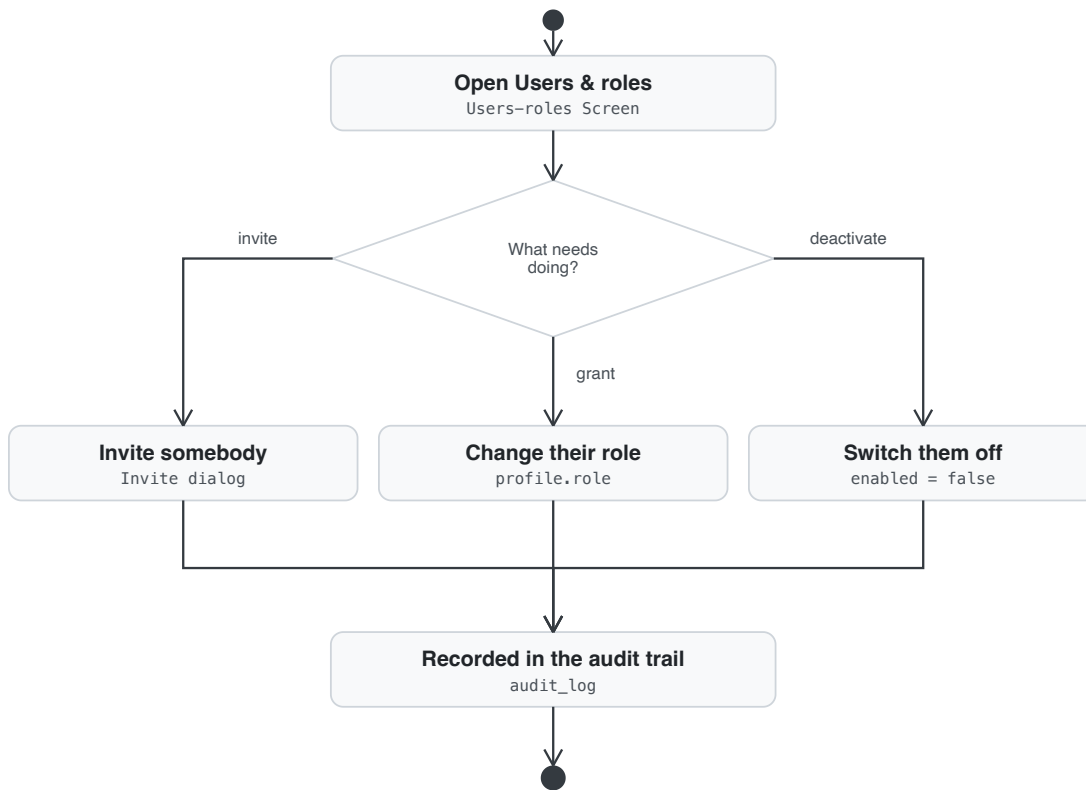
Variations and exceptions

IF...	THEN...
They demote or deactivate the last remaining admin	Refused: it would lock everyone out of the install permanently.
They want a person gone	They deactivate: <code>enabled = false</code> blocks sign-in and keeps every row the person produced, so finished studies stay intact and attributable. Hard-deleting a profile is not offered.
The email is already in use	Refused: <code>profile.email</code> is unique across the install.
They try to change a person's email	Not from here: the person changes it through the login service's verified flow, and this copy follows.

Acceptance test — *Given* two admins, *When* one demotes the other to experimenter, *Then* it succeeds and an `audit_log` entry records both states; *and* attempting to demote the now-only admin is refused.

2 · The Journey, Screen by Screen

Three actions, one audit trail.



3 · In Plain English

Everything here is invite-only, so somebody has to let people in. That somebody is an admin, and in ChatMaestro the admin is the owner; there is no separate super-user account hiding behind it.

There are three roles, and they are deliberately coarse. Enrollees take part in studies. Experimenters design and run them. Admins also manage people, models, and documents. Those roles are not just menu options; the database itself enforces them, so an enrollee cannot read another enrollee's transcript even if something in the app misbehaves.

There are two guard rails. You cannot remove the last admin, because that would lock everybody out of the install for good. And you never delete a person; you switch them off. Their finished work stays where it is, still correctly attributed.

Fits the schema cleanly. `profile.role` + `enabled` are the whole model; the last-admin lock is a rule, not a column.