Monitor a Live Run

Experimenter Watch a run as it happens, drill into one enrollee, and pause, resume, or abort if you must.

Screens Live-monitor `experimenter-monitor.html` Watch-drawer `watch-drawer.html`

Reads the run `experiment_run` who's in it `run_enrollment` each exchange `chat_round`

Key fields status `experiment_run.state` presence `run_enrollment.last_activity_at` withdrawn `withdrawn_at` live scores `chat_round.score_composite`

1 · Given / When / Then

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

GIVEN	A run exists with `state = running` (or `paused`), and enrollees have begun chatting.
WHEN	The experimenter opens the **Live-monitor Screen**, picks the run, and watches progress per enrollee. The roster shows two counts: everyone enrolled, and the **attendees** — those taking part right now, meaning not withdrawn and recently active, derived live from `run_enrollment.last_activity_at` rather than stored. It also shows how many exchanges each has produced and the scores landing on `chat_round`. They can open the **Watch-drawer Screen** to follow one enrollee's transcript, or intervene by pausing, resuming, or aborting the run.
THEN	Reading changes nothing. Intervening moves `experiment_run.state`: `running → paused` (and back), or `→ aborted`. Letting it finish takes it to `done`, which is what triggers the run's scorecard (`run_result`) to be built. **reads** `experiment_run`, `run_enrollment`, `chat_round` **writes** `experiment_run.state` (only when you intervene)

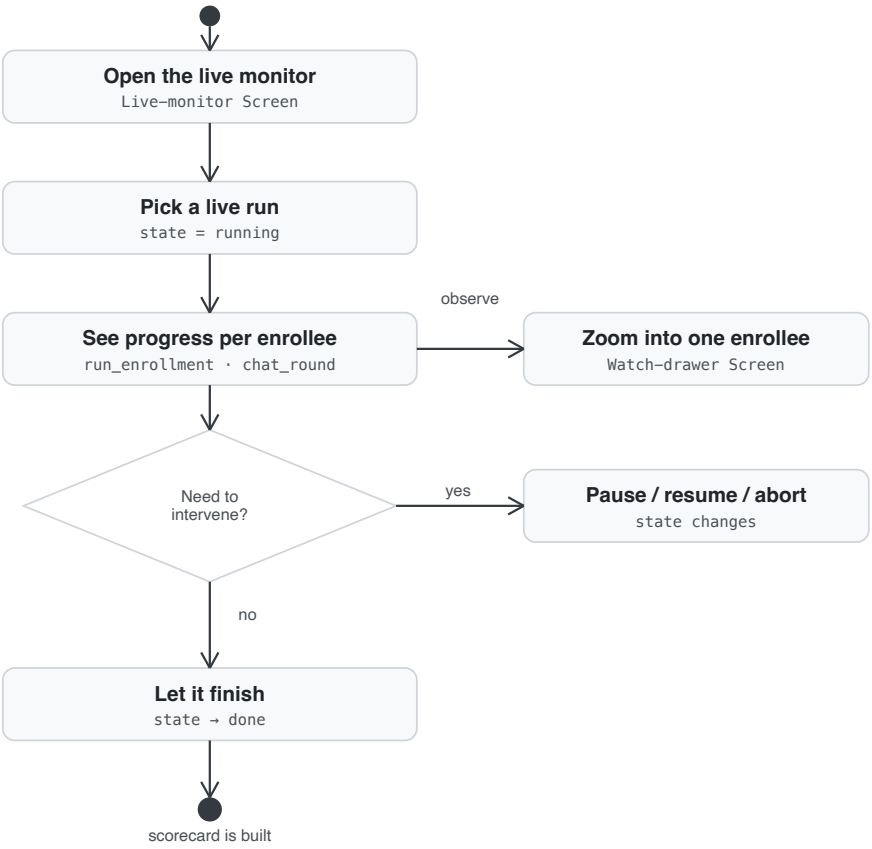
Variations and exceptions

IF…	THEN…
They try to clear a run that is `running` or `paused`	Refused: clearing is allowed only once the run is `done` or `aborted` (*Clear a Run*).
They remove an enrollee from the run	`run_enrollment.withdrawn_at` is set. The enrollee stops receiving the condition; what they already produced is kept and still counts.
They pause the run and try to launch another onto the same people	Refused: `paused` counts as live for the disjoint-cohort rule.
They abort the run	`state = aborted`; its exchanges remain until the run is explicitly cleared.

Acceptance test — *Given* a running run, *When* the experimenter pauses then resumes it, *Then* `experiment_run.state` goes `running → paused → running`, no exchange data is lost, and clearing remains blocked throughout.

2 · The Journey, Screen by Screen

Watching is read-only; intervening changes the run's status.



3 · In Plain English

While a study is live you mostly just watch. The monitor shows who is attending right now versus merely enrolled, how much they have done, and how the answers are scoring as they land.

If you want detail on one person, you open their drawer and read the actual conversation. If something is wrong, you can pause the whole thing, resume it later, or abort it outright.

Two things are deliberately not possible while a run is live: you cannot delete it, and you cannot start another study on the same people. Both protect the data you are in the middle of collecting.

Fits the schema cleanly. Every status change is a move on `experiment_run.state` — the full state machine is in *Run a nudge-comparison series*.