

Roles & terminology

- **Requester:** employee who runs `REQUEST <BranchId>` to contact another branch.
- **Assignee:** employee in the target branch who receives a chat request offer and runs `ACCEPT`.
- **Session:** a single terminal connection (identified by `sessionId`).
- **Chat:** created only after an assignee accepts an offer. Identified as `CHAT-####`.

Lifecycles & timers

1. Request

- `REQUEST <BranchId> [optional note...]` queues a request from your branch to the target branch.
- If someone in the target branch is *idle*, the request is offered to them immediately.
- A dedupe key ensures you cannot spam the same branch: one active attempt per (employee, sourceBranch, targetBranch).

2. Offer (60s)

- The target employee sees: `[OFFER] Incoming chat from <branch>. Use ACCEPT...`
- They must run `ACCEPT` within **60 seconds**. Otherwise the offer expires, the employee goes back to idle, and the request offered to the next idle employee.

3. Chat created

- On `ACCEPT`, a chat is created (e.g., `CHAT-1001`). The assignee is joined and becomes **busy**.
- The requester receives: `<targetBranch> ACCEPTED] ChatID: CHAT-1001. Run: BEGIN CHAT-1001 within 60 seconds.`

4. Requester join (60s)

- The requester must run `BEGIN <ChatId>` within **60 seconds**.
- If they fail, the chat is **cancelled** and the assignee goes back to idle. The request is retried up to **2** misses; after that it is cancelled.

5. Active chat

- Once both sides are joined, use `SEND <message>` to chat.
- Anyone allowed may `JOIN <ChatId>` later (original requester/assignee by `employeeId`; shift managers of either chat branch).
- Use `CHAT_HISTORY` to print the history on your terminal.

6. Leaving / Solo timer (120s)

- `LEAVE_CHAT` removes your session from the current chat and returns you to **idle**.
- If only one side remains, the system posts a system message and starts a **120s** solo timer; if nobody rejoins in time, the chat auto-ends.

7. Ending

- **END_CHAT** ends the chat for **everyone** immediately.
- Before ending, the client may optionally save history via yes/no (prompted).

Who can join an existing chat?

- **Original requester** (by `employeeId`, `disconnect`)
- **Original assignee** (by `employeeId`)
- **Shift manager** of either participating branch.

Presence & idling

- On login, each session is registered and set **idle** by default.
- When a session joins a chat, it becomes **busy** and is removed from the idle pool.
- When leaving/ending, the session returns to **idle** if not participating in other active chats.