Minutes of Meeting

Date and Time	04 December 2025 09:00 PST	Meeting type	Zoom
Organiser	Mr. Rupesh	**Client**	Citywide

Attendees (Internal)
• Kuldeep • Jaspreet • Ravinder • Ajay • Kapil • Akash • Babita
Attendees (Client Side)
• Teresa • Tom

Agenda
• **Discussions on the following:** ○ Admin Panel Phase 2 – Demo Overview (Presented by Kapil) ○ Attendance – Lock/Unlock Shift Feature Demo ○ Bulk Check-In / Check-Out Clarifications ○ Last Person Information – Auto-Population & Cloning Flow ○ Search Option in Officer List (Assignment Section) ○ Report Approval – Hold Until Approved (PSSP Requirement)

The following things are discussed:

1. **Admin Panel Phase 2 – Demo Overview (Presented by Kapil)**
    a. **Modules Demonstrated**
        i. Client Module
        ii. Subscription Module
        iii. Notification Module
    b. **Key Functionalities Shown**
        i. **Client Creation Workflow**
            1. Add basic company details:
                a. Company name, short name, domain
                b. Auto-generated unique ID
                c. Onboarding date
                d. Max number of branches
            2. Owner information + contact details
            3. Company address with Google Map integration
            4. Optional additional contact field
        ii. **Branch Creation**
            1. Branch name, short name
            2. Address with Google Map
            3. Date format, time format, timezone
            4. These settings populate into the client's profile view
        iii. **Subscription Setup (Initial Version)**
            1. Options: Trial or Paid
            2. Trial auto-handles trial period
            3. For paid: Billing date required
            4. Auto-invoicing logic applicable
            5. Billing frequency supported: Monthly, Quarterly
    c. **Major Change Request: User-Based Billing (Highest Priority)**
        i. **Requirement**
            1. The billing system must charge **per user**.
            2. Client should define:
                a. **Number of users allowed**
                b. **Per-user price**
                c. The system must restrict/notify when the user limit is exceeded.
        ii. **User Limit Logic**
            1. When creating a subscription, a field for **"Number of Users"** must be added.
            2. If users exceed the limit:
                a. The system displays a **warning message**.
                b. Clients can choose to **upgrade the number** of users.
                c. No hard restriction/lockout.
                d. System increases user count automatically upon confirmation.
        iii. **Avoid Double Counting Users Across Branches**
            1. A single employee with access to multiple branches must still count as **one user**.

        2. Branch access ≠ additional user billing.

d. **Pricing Model Enhancements**
    i. **Types of Fees**
        1. **Recurring Fees**
            a. Per User (monthly/quarterly)
        2. **One-Time Fees**
            a. Licensing Fee
            b. Branch Licensing Fee
            c. Onboarding Fee
    ii. **Branch Pricing**
        1. First branch: free
        2. Additional branches: one-time licensing fee
        3. Future capability to charge per module / package-level billing.
    iii. **Role of Future Module-Based Billing**
        1. Ability to create bundles/packages (e.g., Basic HR, Advanced HR)
        2. Each module will have its own price
        3. Admin should be able to assign/unassign modules at any time

e. **Employment Type–Based Billing Model**(Final Consensus Model (as discussed by Tom & Teresa):)
    i. **Two Chargeable User Types**
        1. **Full-Time Users**
            a. Includes: Full-Time + Contractor
            b. Charged full user rate
        2. **Part-Time Users**
            a. Includes: Part-Time + On-Call
            b. Charged lower rate
            c. **Part-time users cannot be scheduled for more than 24 hours/week**
                i. If scheduled beyond 24 hours → System prompts upgrade to Full-Time
                ii. Upon confirmation, the user automatically becomes Full-Time and billing adjusts
    ii. **Employer Flexibility**
        1. Admin & Dispatch users are treated as **full-time users**
        2. Unlimited admin users allowed; all are charged as full-time
    iii. **Subscription Input Requirements**
        1. For every client:
            a. Number of Full-Time Seats
            b. Number of Part-Time Seats
            c. Price per Full-Time User
            d. Price per Part-Time User
        2. System must:
            a. Track active users in each category
            b. Trigger warnings & auto-upgrades
            c. Prevent inconsistencies caused by overtime hour calculations

f. **Automated Branch & User Licensing Logic**
    i. **Branch Logic**

        1. Branch assignment should not affect user billing unless new branch is purchased.
        2. A user having access to multiple branches is still counted as one.
    ii. **User Creation / Onboarding Logic**
        1. System must show a warning when onboarding exceeds user limit:
            a. “You are exceeding your allotted user limit. Proceeding will increase your subscription charges. Continue?”
        2. If User Confirms:
            a. Increase user count automatically
            b. Trigger subscription update
            c. Notify CHS Admin + Company Owner

g. **Payment Gateway Integration (Future Scope)**
    i. API integration needed for third-party payment processors
    ii. Automated billing & charge capture
    iii. To be implemented after user-based billing structure is finalized

h. **Client Self-Onboarding Vision (Future)**
    i. Long-term goal:
        1. “Onboard any client in one minute.”
    ii. Automated data extraction from documents/emails for faster onboarding
    iii. AI-assisted onboarding for clients

i. **User Licensing & Usage Tracking**
    i. **Tom** emphasized that the system must accurately track **how many users are being utilized** to avoid misuse or license leakage.
    ii. The team confirmed that CommandHub will display **active, draft, inactive, and terminated users** for each client.

j. **Client Creation & Automated Onboarding:** Kapil demonstrated the end-to-end workflow:
    i. **Subscription details added** at the client level.
    ii. Client is created **within CHS system**, but not fully onboarded.
    iii. **Domain and database mapping** currently handled by development (DigitalOcean).
    iv. Future goal:
        1. Make database, domain, and client provisioning **fully manageable through the admin panel** (no backend involvement).
    v. Once “Launch Client” is triggered:
        1. System automatically assigns default modules, permissions, settings.
        2. Client instance becomes accessible with all baseline configurations.
    vi. **Jaspreet:** Confirmed that manual backend configuration of permissions will no longer be required.

k. **Client Dashboard Details:** Kapil showcased the dashboard for an individual client:
    i. **Displays:**
        1. Active users

        2. Draft users
        3. Inactive users
        4. Terminated users
        5. Blog & DigitalOcean statistics (CPU, space, memory, etc.)
    ii. **Note:** DigitalOcean stats are empty on dev server but will work once deployed.

l. **Server Usage & Resource Monitoring**
    i. The dashboard will display **CPU usage, DB size, and storage consumption** per client.
    ii. Once deployed, CommandHub admin can:
        1. Monitor resource usage per client
        2. Identify when a client is reaching space limits
        3. Trigger upgrades directly from CommandHub
    iii. **Kuldeep** confirmed:
        1. Each client has its **own database cluster**.
        2. Increasing storage for one client will **not affect others**.
        3. Increasing DB space will automatically adjust billing.
        4. DigitalOcean APIs will be integrated to allow admin-side control without logging into DO.
    iv. **CommandHub Admin Analytics (New Requirement) : Tom requested a dedicated analytics dashboard for CommandHub itself**, not just for clients.
        1. **Required metrics:**
            a. Total number of clients
            b. Average users per client
            c. Average storage usage per client
            d. Revenue metrics:
                i. Total revenue
                ii. Average revenue per user
                iii. Average revenue per client
            e. Growth metrics:
                i. Client growth
                ii. User growth
                iii. Revenue trend
        2. **The current draft dashboard already includes:**
            a. Total clients, total revenue
            b. Total users
            c. Growth graphs (client, revenue, user)
            d. Dummy data used now; real data will populate after deployment.
        3. **Action:** Remove irrelevant metrics (e.g., site/beats) from this dashboard.

m. **Environment Setup: Dev → Staging → Production**
    i. The current demonstration is on **development server**.
    ii. After finalization:
        1. Move to **staging** environment for testing (Tom & Teresa).
        2. After approval, push to **production**.
    iii. Mirrors current Core & CommandHub deployment flow.

n. **Reports & Modules**

i. Tom clarified no need for client-side report listings or patrol data on this admin dashboard.
ii. Focus only on:
    1. User statistics
    2. Usage metrics
    3. Subscription and billing
    4. Server resource consumption
iii. In the **Module Assignment section**:
    1. Currently shows default modules.
    2. In the next phase, admin will be able to **enable/disable modules per client** directly.
iv. Examples:
    1. Turn off "Parking Citation" for a specific client.
    2. Handle HRM, ERM, KPI modules as separate add-ons or part of parent module.

o. Infrastructure Compliance (SOC2, HIPAA, Gov Agencies): Tom asked whether DigitalOcean meets compliance requirements or if a migration to AWS is needed.

i. Kuleep Confirmed:
1. DigitalOcean maintains global security standards.
2. Suitable for SOC2, HIPAA, government-level requirements.
3. No need to switch hosting providers.

p. Decisions Taken

i. **CommandHub Admin Analytics** will be developed as a separate dashboard.
ii. **Module enable/disable per client** will be part of the next phase.
iii. **DigitalOcean API integration** will allow:
    1. Space upgrades
    2. Usage visibility
    3. Automated provisioning
iv. Resource monitoring will feed into **billing calculations**.
v. Unnecessary items (sites/beats, client reports) to be removed from admin panel dashboard.

q. Branch-Level Module Management

i. Discussion
1. Kapil demonstrated the *Branch Details* and *Manage Modules* sections.
2. Each branch can have its own set of modules—similar to company-level module assignment.
3. Tom asked about differences between the pages shown.
4. Kuldeep clarified:
    a. Previous page = module management at company level.
    b. Current page = module management at branch level.
    c. Different branches can have different modules based on requirement.
5. Kapil added that modules can also be assigned to the branch collectively.

ii. Outcome
1. Branch-specific module assignment works as intended.

2. No further action items noted.

r. **Multi-Business Type Support (Security, Cleaning, Fire, Law Enforcement, etc.)**

i. **Discussion :** Tom discussed future system direction:
1. The current setup supports only *security companies* (e.g., Citywide).
2. The goal is to support *any service company* (janitorial, cleaning, fire department, police, medical, etc.).
3. Dispatch forms will vary significantly for different industries.
4. The law enforcement version will not be client-based; it will be user/jurisdiction-based.
5. Question raised: *How will the system architecture handle multiple business types? Separate panels or unified?*

ii. **Kuldeep's Explanation**
1. A **Business Type** field will be added during client onboarding.
2. The entire workflow will be driven based on the selected business type.
3. Only one admin panel (CommandHub) will be used; workflows/modules will adjust dynamically.
4. For each new business type:
    a. Team will define workflow structure,
    b. Identify required modules,
    c. Decide how data will be stored and displayed,
    d. Match the workflow inside the existing CSS admin infrastructure.

iii. **Timeline Discussion**
1. Tom asked:
    a. When can they begin defining workflows for a new business type?
2. Kuldeep responded:
    a. Current Trello-board items will be finished in **~2 months**.
    b. Workflow planning for new company types can start **before March**, approx. **90 days out**.

iv. **Outcome**
1. Add **Business Type** field in client onboarding.
2. Future workflows will be handled under the same main admin panel.
3. Target timeframe for starting janitorial workflow planning: **February–March next year**.

2. **Attendance – Lock/Unlock Shift Feature Demo**

a. **Akash's Demonstration**

i. New feature added: **Lock Shift** and **Unlock Shift** in Attendance Module.

ii. Access controlled by **permissions** under Scheduling → Shift Lock/Unlock.

iii. Features include:
1. Bulk lock/unlock for selected records.
2. Icons showing locked shifts.

    3. Filters: *Locked / Unlocked / All*.
  iv. Teresa confirmed functionality is correct.

b. **Additional Logic**
  i. Locked shifts reflect immediately in the scheduling page.
  ii. Payroll Notification Logic:
    1. Automatic email notifications to users with lock/unlock permissions.
    2. Additional configurable notification recipients in Company Settings.
    3. Notifications include:
      a. Upcoming payroll run reminder (e.g., payroll runs on 15th → email on 13th).
      b. Missing punches after payroll close (post-16th emails).
    4. Payroll screen will show a **warning message** if any shifts are pending lock.

c. **Outcome**
  i. Lock/Unlock feature accepted.
  ii. Payroll notification enhancements approved.

3. **Bulk Check-In / Check-Out Clarifications**

a. **Issue Raised by Jaspreet**
  i. Current bulk operations overwrite check-ins/outs even if some users already have valid values.
  ii. Need clarification:
    1. Should bulk checkout be allowed if a user is not checked in?
    2. Should it update existing entries or only missing ones?

b. **Teresa's Decisions**
  i. **Bulk Checkout Validation**
    1. System must **NOT** check out a user who is not checked in.
    2. Only entries with an existing check-in should be updated.
  ii. **Updating Existing Entries**
    1. Bulk check-in/out **should override** even if the user has existing times.
    2. Used for cases where employees accidentally clock in/out late.
    3. Bulk action will reset times to the correct shift start & end.
  iii. **Purpose Clarified**
    1. Similar to the old *Team Express / Steliix* behavior.
    2. Designed for mass corrections when multiple employees make punch mistakes.

c. **Outcome**
  i. Implement Logic:
    1. Skip checkout for users without check-in.
    2. Overwrite all applicable records when bulk updating.

4. **Last Person Information – Auto-Population & Cloning Flow**

a. **Discussion**
  i. System should auto-populate the *last call's reporting person information* once the phone number is entered.
  ii. Clicking on the history icon should:
    1. Redirect to the call history page.
    2. Provide an option **"Clone Call"**.

        3. On clicking “Clone”, system should open a **new call form** with the cloned data prefilled.
    iii. **Assignment Rule:**
        1. No officer/person should be auto-assigned.
        2. User can assign themselves manually.
        3. **No auto-saving** is required.
    iv. Tom confirmed that if one user is marked as **Primary**, no other user should be set as Primary.
  b. **Outcome**
    i. Flow approved as demonstrated.
    ii. Primary assignment validation is correct.

5. **Search Option in Officer List (Assignment Section)**
  a. **Discussion**
    i. Tom suggested adding a **search bar** when selecting a person to dispatch, since scrolling through long lists may be challenging.
    ii. Teresa currently uses browser “Ctrl + F” but agreed a search box would improve usability.
    iii. However, Teresa wants to still **visually see all officers** to decide the best one.
  b. **Outcome**
    i. Add a **search functionality** on officer selection list.
    ii. Must allow search but still display full list simultaneously.

6. **Report Approval – Hold Until Approved (PSSP Requirement)**
  a. **Background**
    i. Current behavior:
        1. System sends only **approved reports** from the defined time window.
        2. Reports are filtered by **arrive time**.
        3. Unapproved reports remain unsent but do not block the approved ones.
  b. **Client Request**
    i. For PSSP: **Hold all reports until every report within the scheduled service hours is approved**.
    ii. They want *all reports* (for that service period) sent together as one complete batch.
  c. **Concerns Raised**
    i. If officers delay approvals for days/weeks, entire batches may get blocked.
    ii. Tracking old unapproved reports can become difficult.
  d. **Final Decision on "Hold Until Approved" Process**
    i. Block Entire Batch
        1. For a given 24-hour service window, **do not send any reports** unless *all* are approved.
        2. Even a single pending report will block the entire batch.
    ii. No “Partial Report” Sending
        1. If reports are missing/half-complete, system must send **nothing**.
    iii. Daily Notification to Account Manager
        1. If the batch fails due to unapproved reports:
            a. System will send an email to the **Account Manager**.

b. Email will include **how many reports** are unapproved.
c. Sent **daily** until the issue is resolved.
2. If the batch missed its scheduled send time, it will not be re-sent automatically even after reports are later approved.

iv. **Account Manager Responsibility**
1. They must **manually send** that batch to the client once all reports are approved.

v. **No Reminders for Old Pending Reports**
1. If users have old unapproved reports (even from a month ago), that is their internal responsibility.

vi. **Outcome**
1. Implement "Full Batch Hold" logic.
2. Add automatic daily notification emails to Account Manager.
3. No auto-resend of previous-day missed batches.

e. **Clarification on System Cron Behavior**
i. Daily cron checks only the **current scheduled window** (e.g., 4th–5th at 11 a.m.).
ii. It does **not** revisit previous unsent batches.
iii. Notification must explicitly mention the **date and site** of the failed batch.