|  |
| --- |
| SARApp |
| Incident Management Assistant |
| Design Document |

|  |
| --- |
| Brendan Pheley  7-31-2025 |

[ICS Command Assistant - Desktop Application 3](#_Toc204882988)

[Phase 1: Core System & User Foundation 8](#_Toc204882989)

[Phase 2: Team Operations, Personnel, and Status Boards 10](#_Toc204882990)

[Phase 3: Communications & Public Information 12](#_Toc204882991)

[Phase 4: Forms, Documentation, & Reference Library 13](#_Toc204882992)

[Phase 5: Logistics, Medical, & Safety 14](#_Toc204882993)

[Phase 6: Intel & Mapping 15](#_Toc204882994)

[Phase 7: Advanced Operations & Toolkits 16](#_Toc204882995)

[Phase 8: UI Customization & Multi-Window UX 17](#_Toc204882996)

[Phase 9: Status Boards, Automation, & Reporting 18](#_Toc204882997)

[Phase 10: Finance/Admin & Incident Closeout 19](#_Toc204882998)

[Phase 11: Mobile Integration & Future Systems 20](#_Toc204882999)

[Phase 12: Special/Planned (AI & Advanced GIS) 21](#_Toc204883000)

[Module 1: Command 22](#_Toc204883001)

[Module 2: Planning 24](#_Toc204883002)

[Module 2-1: Strategic Objectives 26](#_Toc204883003)

[Module 3: Operations 30](#_Toc204883004)

[Module 3-1: Taskings 32](#_Toc204883005)

[Module 4: Logistics 45](#_Toc204883006)

[Module 4-1: Resource Request 47](#_Toc204883007)

[Module 5: Communications 51](#_Toc204883008)

[Module 6: Medical and Safety 53](#_Toc204883009)

[Module 7: Intel 55](#_Toc204883010)

[Module 8: Liaison 56](#_Toc204883011)

[Module 9: Personnel and Role Management 58](#_Toc204883012)

[Module 9-1: Personnel Certifications 60](#_Toc204883013)

[Module 10: Reference Library 63](#_Toc204883014)

[Module 11: ICS Forms and Documentation 65](#_Toc204883015)

[Module 12: Finance/Admin 67](#_Toc204883016)

[Module 13: Status Boards 69](#_Toc204883017)

[Module 14: Public Information 71](#_Toc204883018)

[Module 15: Mobile App Integration 73](#_Toc204883019)

[Module 16: Training/Sandbox Mode 75](#_Toc204883020)

[Module 17: Search and Rescue Toolkit 76](#_Toc204883021)

[Module 18: Disaster Response Toolkit 77](#_Toc204883022)

[Module 19: Planned Event Toolkit 78](#_Toc204883023)

[Module 19-1: Event Promotions and Communications 82](#_Toc204883024)

[Module 19-2: Vendor & Permitting Coordination 83](#_Toc204883025)

[Module 19-3: Public Safety & Incident Management 84](#_Toc204883026)

[Module 19-4: Mini Tasking Module 85](#_Toc204883027)

[Module 19-5: Public Health & Sanitation Oversight 86](#_Toc204883028)

[Module 20: Initial Response Toolkit 87](#_Toc204883029)

[Module 21: UI Customization 88](#_Toc204883030)

[Module XX: AI Integration (Wishlist Phase) 89](#_Toc204883031)

ICS Command Assistant - Desktop Application

# 1. Overview

The ICS Command Assistant is a Windows desktop application designed to support emergency management, search and rescue, and incident command operations. Its goal is to streamline information flow, tasking, personnel/resource tracking, documentation, and reporting, with a flexible, modular design based on the Incident Command System (ICS) framework.  
  
Target Users:

* Incident Commanders
* Section Chiefs and ICS staff
* Logistics and Planning personnel
* Field teams (via desktop, future mobile integration)
* Emergency management staff at local, state, and federal levels

# 2. System Architecture

Frontend:

* Developed with Qt (PySide6) using QML for UI and Python for logic
* Modular UI layout with each module having its own:
  + Models/ for class definitions
  + Panels/ for PySide6 widgets (e.g., status boards)
  + Qml/ for QML interface components (e.g., detail views, creation dialogs)
* Supports:
  + Modeless windows
  + Multi-monitor layouts
  + User-selectable themes (dark mode, light mode, and custom color profiles)

Backend:

* Python FastAPI server (bundled and run locally)
* SQLite databases:
  + master.db (persistent personnel, equipment, templates, etc.)
  + Incident-specific DBs (all incident/event data, audit trail, forms)
* Local LAN/WebSocket support for real-time collaboration

File System & Data:

* Root-Level Folders
  + /data : All databases and uploads
  + /modules : All feature-specific logic and UI (subfolders for each module)
  + /qml : Shared QML ui files for program level forms and windows.
* Each module folder contains:
  + /models: Python classes for data structure
  + /panels: PySide6 table widgets and logic
  + /qml: QML UI files for forms, details, creation dialogs
* Drag-and-drop upload interface for:
  + Documents, ICS forms, media (photos, video, and audio), other attachments

Security & Perincidents:

* Role-based UI/perincidents
  + User role affects visible modules, context menu options, and edit rights
* Audit logs for all changes
  + All data changes tracked per user in the incident-specific database

# 3. Module Overview

The app is divided into major ICS sections, each with its own modules:

1. Command: Incident command, objectives, situational awareness
2. Planning: Tasking, action plans, documentation, mapping
3. Operations: Field teams, assignments, real-time tracking
4. Logistics: Personnel, check-in, resources, supplies, transportation
5. Communications: Radio logs, comms plans, chat, frequency/resource management
6. Medical and Safety: Medical plans, safety messages, incident tracking
7. Intel: Data, weather, intelligence, and clue management
8. Liaison: Agency contacts, notifications, support requests
9. Personnel and Role Management: Assignments, qualifications, accountability
10. Reference Library: ICS forms, agency forms, guides, regulations, state/county resources
11. ICS Forms and Documentation: All fillable forms, auto-fill from data, upload/digitization of custom forms
12. Finance/Admin: Time tracking, expense reporting, procurement, reimbursement
13. Status Boards: Global/module-specific boards (team, personnel, equipment, resource requests, etc.)
14. Public Information: Press releases, briefing log, public info contacts
15. Mobile App Integration: Sync/bridge with field/mobile app (future)
16. Training/Sandbox Mode: Simulated incidents, user practice environment
17. Search and Rescue Toolkit: Specialized tools, calculators, and workflows for SAR operations
18. Disaster Response Toolkit: Dedicated modules for disaster incident types (floods, wildfires, hurricanes, etc.)
19. Planned Events Toolkit: Event-specific planning and resource management tools
20. Initial Response Toolkit: Tools for hasty search, rapid tasking, and initial response workflows
21. UI Customization: End-user module for UI Template Selector, Custom Theme Editor, and Dashboard Builder (custom layouts, colors, and widget arrangements)

Special/Planned:  
- Module XX: AI Integration (future AI assistant, data automation, recommendations)  
- Module XX: GIS Integration (future advanced GIS and mapping platform)

# 4. UI Layout

## Main Window:

* Persistent header: Incident name, incident number, operational period, user role
* Section tabs (top-level): Command, Planning, Operations, etc.
* Modules appear as sub-tabs/side menu items
* Modeless windows for all modules (multi-monitor ready)
* Quick-access dashboard with widgets for key stats/alerts
* Team and Task Status boards always accessible
* Built-in search/filter and help
* Theme selector (light/dark/custom color profiles)
* **Docking Improvements**
* Center area: Establishes CenterDockWidgetArea so docks snap relative to the full window.
* Margins: Zeroed central/container layout margins and spacing to make the dock area fill the window.
* Constructors: Standardized on CDockWidget(self.dock\_manager, title) to avoid deprecated paths and ensure full ADS behavior.
* **Default Layout**
* Forced defaults: Added FORCE\_DEFAULT\_LAYOUT = True to always start with a known-good layout.
* Default docks: Creates “Mission Status” (center), “Team Status” (left), and “Task Status” (bottom).
* Reset: Debug menu adds “Reset Layout (Default)” to clear saved perspectives and rebuild.
* **Status Boards**
* Unified panels: Default docks now use the same rich panels as menu items:
  + modules/operations/panels/team\_status\_panel.py
  + modules/operations/panels/task\_status\_panel.py
* Row coloring: Applies centralized palettes from styles.py (team/task).
* Live updates: Changing a Status cell recolors the entire row instantly.
* Right‑click menu: Per-row context menus for status changes and “View Detail”.
* **Menu Behavior**
* Undocked by default: Panels opened via menu now float (open undocked). They can be docked by the user afterward.
* QML docks: When QML is used for a panel, it’s wrapped in a QQuickWidget inside a CDockWidget so it’s dockable.
* **Layout Persistence**
* Save/load: Persists perspectives to settings/ads\_perspectives.ini (“default”) on close, auto-loads on startup when not forcing defaults.
* Safe fallback: If loading fails or yields no docks, seeds the default layout.
* **Mission Status**
* Converted from fixed header to a movable “Mission Status” dock showing incident/user/role; stays in sync with title and AppState.
* **Multi‑Window Workspaces**
* New workspace: Window → “New Workspace Window” creates a floating dock container you can move to a second monitor and dock panels into.
* **Debug Utilities**
* Debug menu: “Open Default Docks” and “Reset Layout (Default)” for quick recovery during testing.
* Print active incident: Quick inspection action remains.

## Module Windows:

* Consistent layout for tables, forms, and detail views
* Drag-and-drop upload areas where relevant
* Print/export/report buttons for all major forms
* User presence/status shown throughout

# 5. Key Features

* Modular architecture, with ability to enable/disable modules per incident
* Multi-database support (persistent master, incident-specific DBs)
* Fillable ICS and agency forms (auto-filled from live data)
* Real-time LAN/WebSocket collaboration and updates
* Detailed audit trail/change logs, versioned form history
* Global and module-specific search and filters
* User presence/status indicators and login auditing
* Status boards for personnel, teams, equipment, tasks, resource requests, etc.
* Quick-entry hotkey panel and UI shortcuts
* Form dependency hints and workflow suggestions
* Multi-monitor/multi-window capability
* Drag-and-drop file uploads (media, forms, documents)
* Embedded training/sandbox mode for onboarding/practice
* Scheduled task alerts and reminders
* Time synchronization with GPS or NTP
* Mapping and GIS integration for task, team, and resource tracking
* Multiple UI themes and color profiles

# 6. Design Phases

## Phase 1: Core System & User Foundation

* QML/PySide6 UI scaffold and main window shell
* FastAPI backend integration
* Database architecture: master and incident DBs
* Role-based login & user management
* Incident creation, selection, and loading
* Top-level navigation with module placeholders

## Phase 2: Team Operations, Personnel, and Status Boards

* Personnel roster and org structure
* Role assignment, team creation, and status tracking
* Planning module (tasking, SITREP, time tracking)
* Operations dashboard (team/assignment views)
* Initial Status Boards (team and personnel)
* ICS 214 logging interface

## Phase 3: Communications & Public Information

* Chat/messaging system (IM-style)
* ICS 213 structured messaging
* Notification system for updates/alerts
* Public Information module (press log, bulletins, approvals)
* File attachments in chat/messages

## Phase 4: Forms, Documentation & Reference Library

* ICS Forms engine (213, 214, 206, 205, 205A, etc.)
* Auto-fill and digitization framework
* Custom form uploader and template management
* Versioned form/document history
* Reference Library browser (SOPs, agency docs, guides)
* Related Forms linking system

## Phase 5: Logistics, Medical, & Safety

* Logistics inventory manager (supplies, vehicles, equipment)
* Check-in/check-out system
* Medical/Safety log (ICS 206, safety messages)
* Resource tracking and assignment
* Reimbursement/cost tracking

## Phase 6: Intel & Mapping

* Intel dashboard (data, weather, intelligence, clue management)
* GIS mapping engine (KML/GeoJSON import, task/resource plotting)
* Layer toggles (search areas, hazards, units)
* Time synchronization (GPS/NTP)
* SITREP builder

## Phase 7: Advanced Operations & Toolkits

* SAR Toolkit (tools, calculators, clue management)
* Disaster Response Toolkit (incident-specific)
* Planned Events Toolkit (event planning)
* Initial Response Toolkit (hasty search, rapid tasking)
* Liaison/agency support module

## Phase 8: UI Customization & Multi-Window UX

* UI Template Selector (layout/theme options)
* Custom Theme Editor (color palette, dark/light mode)
* Dashboard Builder (custom widgets, drag/drop layouts)
* Multi-window and multi-monitor support
* System tray/background integration

## Phase 9: Status Boards, Automation & Reporting

* Dedicated status boards (resource, team, comms, task, medical, logistics)
* Workflow automation (reminders, overdue alerts, form dependency hints)
* Auto-generation of logs (ICS 214, etc.) from activity data
* Filtering, sorting, PDF export, report generation

## Phase 10: Finance/Admin & Incident Closeout

* Full Finance/Admin module (time tracking, cost centers, audit trail)
* End-of-incident tools (export, ZIP archive, printable reports)
* Embedded sandbox/training mode
* Global and module-level search/filter

## Phase 11: Mobile Integration & Future Systems

* Cross-platform sync (desktop ↔ mobile app)
* Import/export for tasks, logs, forms, and status
* Real-time mobile device reflection in boards
* Shared templates and LAN/USB/manual sync

## Phase 12: Special/Planned (AI & GIS Integration)

* AI Integration (assistant, automation, smart forms)
* Advanced GIS Integration (external systems, overlays, AVL, drone feeds, etc.)

# 7. Future Considerations

* Embedded AI assistant for data entry, search, and suggestions
* Integration with external CAD, GIS, or incident planning platforms
* Offline/online sync for mobile/field deployments
* Internationalization (multi-language support)
* Custom report generator, print-ready exports
* Scheduled upgrades based on user feedback
* Additional module/toolkit categories (as user needs evolve)
* Advanced analytics/dashboarding
* Additional theme and branding options
* Enhanced mapping (e.g., drone overlays, live AVL)

Phase 1: Core System & User Foundation

**Phase 1 Goals:**

Establish the backbone of the application, ensuring the system runs locally, supports users, and lays groundwork for all future modules. Build a stable, modular, and secure core that is easy to extend.

**Major Components to Build**

**1. QML/PySide6 UI Scaffold and Main Window Shell**

- QML UI/frontend

- Build the main window with persistent header (shows incident name, op period, user role, etc.).

- Implement top-level navigation (tab bar for Command, Planning, Operations, etc.).

- Display empty placeholders or "coming soon" for modules not yet built.

- Enable basic theming (at least dark and light mode toggles).

**2. FastAPI Backend Integration**

- Create the initial FastAPI project structure.

- Set up API routing and allow connection from the Electron frontend.

- Enable CORS to allow local frontend-backend communication.

- Add a health check endpoint for dev testing (/api/health).

**3. Database Architecture: Master and Incident DBs**

- Implement master.db for persistent records (personnel, equipment, templates).

- Set up folder for incident-specific databases (/data/incidents/).

- Create API endpoints for:

- Creating new incident DBs (from UI or CLI)

- Selecting/loading a incident (returns incident metadata)

- Ensure data folder exists and is writable.

**4. Role-Based Login & User Management**

- Create a login screen as the first UI the user sees.

- Implement ID entry and role selection (roles pulled from personnel database in master.db).

- Basic role management: restrict some features/menus by role (stubbed for now).

- Store current user/role in app state and audit logs.

**5. Incident Creation, Selection, and Loading**

- Provide UI for creating a new incident (form: name, type, description, etc.).

- Display list of existing incidents; allow selection and loading.

- Show incident info in the persistent header after selection.

- Only allow one active incident at a time.

**6. Top-Level Navigation with Module Placeholders**

- Render tabs for all major modules, even if not implemented (greyed out/“coming soon”).

- Place quick-access dashboard (widgets for: current incident, op period, alerts).

- Enable a “dummy” personnel/team status board for early testing.

**7. System Settings & Data Folder**

- Allow user to set/change /data directory (for DBs, uploads, attachments).

- Ensure the app checks for this folder on startup and prompts if missing.

**Security and User Experience**

- Store all sensitive data (user/incident) only locally—no cloud sync in Phase 1.

- Ensure user actions (login, incident select, create) are logged (audit trail foundation).

- Display current user, role, and incident context at all times.

**Phase 1 Exit Criteria (Definition of Done)**

- App launches from a single executable (or from dev environment) and shows the main window.

- User can:

- Log in with an ID and select a role.

- Create and select a incident.

- See a dashboard with incident/user context.

- Navigate top-level tabs (see placeholder screens).

- master.db and at least one incident DB exist in /data.

- No critical errors or crashes in normal use.

- All persistent data is stored locally in the correct folder.

**Optional (Phase 1.5 / Stretch Goals):**

- Add basic in-app help/about screen.

- Support for minimal user settings (theme, font size).

- Allow basic app update checking (if relevant).

Phase 2: Team Operations, Personnel, and Status Boards

**Phase 2 Goals:**

Enable robust management and real-time tracking of all teams, personnel, and their statuses. Implement organization structure, assignments, dashboards, and initial activity logging to support incident operations and accountability.

**Major Components to Build**

**1. Personnel Roster and Org Structure**

- Design and build the Personnel Roster UI: searchable, filterable table of all personnel (with columns for name, role, status, assignment, contact info, etc.).

- Implement add/edit/remove personnel workflows (data stored in master.db).

- Build the Organizational Structure Manager:

- Allow creation and editing of org units (sections, teams, branches, etc.).

- Implement drag-and-drop assignment of personnel to roles and teams.

- Show interactive org chart/tree.

**2. Role Assignment, Team Creation, and Status Tracking**

- Implement team creation and editing UI: assign name, team type, members, and supervisor/leader.

- Enable assignment of personnel to teams and ICS roles.

- Track real-time status for each person and team (Available, Assigned, Out of Service, etc.).

- Store all assignments, changes, and status updates in incident-specific DB for audit and after-action review.

- Allow for “hot swap” of roles and team assignments, preserving assignment history.

**3. Planning Module (Tasking, SITREP, Time Tracking)**

- Develop task assignment and tracking tools:

- Allow creation of new tasks with description, priority, category, and assigned team(s).

- Implement task board UI: show tasks by status (Draft, Planned, In Progress, Completed, Cancelled).

- Add time tracking functionality:

- Record shift start/stop, on-duty/off-duty status for each person.

- Store time logs for personnel, teams, and assignments.

- Enable SITREP (Situation Report) entry and display:

- Simple form and log of recent operational events, visible to Planning/Command.

**4. Operations Dashboard (Team/Assignment Views)**

- Build an Operations Dashboard displaying:

- Current teams, assignments, and statuses.

- Map/table/list views of team locations (location may be manual for now).

- Live updates for critical team changes or status updates.

- Allow filtering/sorting by assignment, role, status, team, or operational period.

**5. Initial Status Boards (Team and Personnel)**

- Implement Status Boards Module:

- Create separate boards for team status and personnel status.

- Board should be real-time (auto-update or with refresh button).

- Enable filters (by team, status, role, etc.).

- Visual indicators for status (color-coding, icons).

- Enable printing or exporting current board state.

**6. ICS 214 Logging Interface**

- Develop a basic ICS 214 Activity Log interface:

- Each team, unit, and the overall Operations section gets its own log stream.

- Allow manual entry and edit of log items (who, what, when, notes).

- Auto-populate logs from status changes, assignments, and critical events.

- Enable export/print of activity log for recordkeeping.

**Security and User Experience**

- Enforce role-based perincidents for sensitive actions (e.g., only Ops/Planning/Command can change teams or assignments).

- All personnel and assignment changes are recorded in audit trail.

- UI always shows current user, role, and incident context.

**Phase 2 Exit Criteria (Definition of Done)**

- User can view and edit personnel roster, including assignments, roles, and teams.

- Teams can be created, edited, and assigned to tasks.

- Real-time status boards for personnel and teams are visible, filterable, and color-coded.

- Operations dashboard shows assignments and status.

- ICS 214 logs are available, and logs can be exported/printed.

- All actions and changes are stored in the appropriate incident database.

- No critical errors or crashes during normal use.

**Optional (Phase 2.5 / Stretch Goals)**

- Implement drag-and-drop task assignment on dashboards.

- Add shift/relief tracker with tie-in to ICS 214 logs.

- Enable basic map visualization for team locations (if not done in Phase 1).

- Add automated reminders/alerts for overdue tasks or status changes.

Phase 3: Communications & Public Information

Phase 4: Forms, Documentation, & Reference Library

Phase 5: Logistics, Medical, & Safety

Phase 6: Intel & Mapping

Phase 7: Advanced Operations & Toolkits

Phase 8: UI Customization & Multi-Window UX

Phase 9: Status Boards, Automation, & Reporting

Phase 10: Finance/Admin & Incident Closeout

Phase 11: Mobile Integration & Future Systems

Phase 12: Special/Planned (AI & Advanced GIS)

Module 1: Command

# 1. Module Name & Description

This module provides high-level control over incident setup, incident configuration, and global status settings. It is typically used by Incident Commanders or authorized supervisors at the top of the ICS structure.

# 2. Primary Functions

* Define incident type to dynamically load relevant modules (e.g., SAR, Fire, Multi-Agency, Training)
* Designate incident as a training incident with limited or sandboxed data persistence
* Create and manage incidents/incidents
* Set incident type, name, and geographic scope
* Define operational periods
* Set incident status flags (e.g., Active, Standby, Terminated)
* Lock/unlock incident data for editing
* Archive or export incidents
* Set incident objectives (tie-in with ICS 202 and Planning's Strategic Tasks)
* Optional: Track financial allocation and spending at a incident level

# 3. Submodules / Tools

* Incident Overview Panel: Dashboard showing basic info, time, status
* Incident Configuration Form: Detailed editor for incident profile
* Incident Status Controls: Toggle incident state (Active, Paused, Standby, Terminated); includes timestamps and reason codes
* Objective Tracker: Link or manage ICS 202 objectives and feed into Planning's Strategic Tasks Module
* Finance Overview Panel (optional): Displays total budget, spend-to-date, and remaining balance
* IAP Builder: Wizard-style interface that supports ICS 201 and 202 entry and integrates data from other modules (Personnel, Operations, Logistics, Communications, Safety) to support auto-population of additional forms like ICS 203, 204, 205, 206, and 207

# 4. Key Data Modules

Operational period rollover logic can be optionally enabled per incident. If active, the system will automatically close and create a new period at a defined time (e.g., 0700 local), carrying forward uncompleted objectives.

* Incident: id (int, PK), name (str), number (str), type (str), description (str), status (str), search\_area (str), icp\_location (str), start\_time (str ISO8601 UTC), end\_time (str ISO8601 UTC), is\_training (BOOLEAN)
* OperationalPeriod: id (int, PK), incident\_id (str, FK incidents(id)), op\_number (str), start\_time (str ISO8601 UTC), end\_time (str ISO8601 UTC)
* IncidentObjective: id (int, PK), incident\_id (str, FK incidents(id)), text (str), priority (int), tags[]

# 5. UI Components

* - Incident creation/edit form
* - Incident status display (e.g., badge or label in header)
* - Operational period timeline/selector
* - Export/archive buttons
* - Dashboard card with live incident metadata
* - Finance overview widget (if enabled)

# 6. API Endpoints

* - GET /api/incidents
* - POST /api/incidents
* - PUT /api/incidents/{id}
* - DELETE /api/incidents/{id}
* - GET /api/incident/{id}/periods
* - POST /api/incident/{id}/periods
* - GET /api/incident/{id}/objectives
* - POST /api/incident/{id}/objectives
* - GET /api/incident/{id}/finance
* - PUT /api/incident/{id}/finance

# 7. Database Tables

* - incidents
* - operationalperiods
* - incidentobjectives
* - incidentfinance

# 8. Inter-Module Connections

* - Command Module: Objectives feed directly into Strategic Tasks
* - Operations: Task visibility filtered by operational period and linked objectives
* - Forms Module: 202, 203, and others rely on incident metadata
* - Finance/Admin: Aggregates incident-level financial data

# 9. Special Features / Notes

* - Incidents may be flagged as "Planned" to indicate upcoming events or operations that have not yet started
* - Incident type selection determines which modules and features are loaded and visible
* - Training incidents are isolated and may use non-persistent data for exercise or simulation purposes
* - Only one incident may be "active" at a time
* - System auto-locks editing of closed/archived incidents
* - Built-in validation to prevent overlapping operational periods
* - All timestamps stored in UTC
* - Optional finance tracker can be toggled on/off based on organization policy or role access
* - The incident status is displayed in context (e.g., header or dashboard) but does not persist as a banner across unrelated modules

Module 2: Planning

# 1. Module Name & Description

Planning Module

The Planning Module manages forward-looking operations, documentation, and coordination for all incident activities. It supports the development of objectives, tracks strategic tasks, and interfaces with time-based tools like the IAP and ICS 214 activity logs.

# 2. Primary Functions

* - Convert high-level objectives into strategic taskings
* - Manage task tracking, priority, and completion
* - Manage operational periods and planning timelines
* - Prepare planning inputs for IAP development
* - Interface with ICS 214 activity logging
* - Coordinate with other modules to populate IAP forms

# 3. Submodules / Tools

* - Pending Approval Queue: Separate interface for reviewing and approving tasks before they are assigned or published to the IAP.
* - Planning Forecast Tool: Projects anticipated personnel or resource requirements based on planned tasks in upcoming operational periods.
* - Operational Period Cloning Tool: Allows duplication of prior periods, including tasks, objectives, and structure for rapid reuse.
* - Task Metrics Dashboard: Summarizes task volume, status distribution, and completion trends across periods and sections.
* - Strategic Task Tracker: Tracks requests and objectives converted into actionable tasks. Authorized positions can promote external requests into incident objectives, which are then visible in the Command module and Planning dashboard.
* - Planning Dashboard: Overview of operational period status, task load, and pending approvals
* - IAP Input Panel: View and edit inputs for forms like 202, 203, 204, 215
* - Operational Period Scheduler: Adjust and define timeframes for each period
* - Planning History View: Displays changes to strategic tasks, objectives, and planning entries over time with timestamps and editor details
* - Task-to-Form Mapping Panel: Allows planners to link tasks to associated ICS forms (e.g., 204, 206, 215), track form status and completion, and validate documentation coverage

# 4. Key Data Modules

* - StrategicObjectives: id (int PK), incident\_id (int, FK incidents(id)), objective\_id (str), description (str), status (str), assigned\_section (str), priority, due\_time (str ISO8061 UTC), is\_external\_request (boolean), customer (int, FK agency\_contacts(id))
* - PlanningLog: id (int PK), incident\_id (int, FK incidents(id)), text (str), timestamp (str ISO8061 UTC),, entered\_by (int, FK personnel(id))
* - PlanningNote: id (int PK), incident\_id (int, FK incidents(id)), user\_id (int, FK personnel(id)), text, tags[]

# 5. UI Components

* - Task entry form and assignment table
* - Task prioritization interface (drag-and-drop or rank-based)
* - Strategic objective linkage display
* - Operational period timeline tool
* - Planning summary feed or board

# 6. API Endpoints

* - GET /api/planning/tasks
* - POST /api/planning/tasks
* - PUT /api/planning/tasks/{id}
* - GET /api/planning/logs
* - POST /api/planning/logs
* - GET /api/planning/notes
* - POST /api/planning/notes

# 7. Database Tables

* - strategicobjectives
* - planning\_logs
* - planning\_notes

# 8. Inter-Module Connections

* - Command Module: Receives linked objectives for strategic breakdown
* - Operations: Task status updated based on field execution
* - Forms Module: Data feeds into ICS 202, 203, 204, 215, and 214
* - Personnel Module: Tasks may be linked to positions/roles

# 9. Special Features / Notes

* - Pending task approvals must be cleared before integration into the IAP or Operations workflow
* - Forecasting engine helps anticipate personnel/resource needs based on scheduled or recurring tasks
* - Operational periods can be cloned to streamline multi-day event planning
* - Real-time task metrics and summaries help Planning Chiefs prepare SITREPs or time-critical reports
* - Tasks can be internal or external in origin
* - ICS 214 activity logs can optionally be auto-linked to strategic tasks
* - Users can tag, prioritize, and track task chains across periods
* - Supports comment threads or updates on task cards
* - Built for tight integration with IAP and Form Builder workflows
* - Planning history view provides version tracking and accountability for strategic decisions
* - Risk assessment logic deferred to the Safety Module

Module 2-1: Strategic Objectives

# 1. Module Name & Description

The Objective Tracker submodule provides centralized management of all objectives derived from incident objectives, external requests, or manual entry by Planning or Command staff. It tracks the full lifecycle of each objective—creation, approval, assignment, status changes, audit log, and closure—and links each objective to its originating objective or request for comprehensive accountability.

# 2. Primary Functions

* Create, edit, and manage objectives (manual, from objectives, or external requests)
* Link objectives to incident objectives
* Status and lifecycle tracking (pending, approved, assigned, in progress, completed, cancelled)
* Approval workflow: approve, reject, or return objectives for more information
* Full audit trail: every change, who, what, when
* View, filter, and report on objectives by status, priority, assignment, period, and origin

# 3. Submodules / Tools

* Objective List: Filterable, sortable list of all objectives
* Objective Detail Window: Modeless, resizable window with full detail view and editor for each objective
* Objective Linkage Panel: Visual link and reference to originating objective
* Approval Workflow Panel: Approve/reject/return controls for pending objectives
* Audit Trail Viewer: Timeline of all changes for each objective

# 4. Key Data Modules

### **Objective**

* id (int)
* incident\_id (int)
* originating\_objective\_id (nullable int)
* description (text)
* status (enum: pending, approved, assigned, in\_progress, completed, cancelled)
* assigned\_section\_id (nullable int)
* priority (enum: low, normal, high, urgent)
* due\_time (datetime)
* customer (text) // Name of customer or requesting party
* created\_by (user id)
* created\_at (datetime)
* updated\_at (datetime)
* closed\_at (nullable datetime)

### **AuditLog**

* id (int)
* task\_id (int)
* user\_id (int)
* timestamp (datetime)
* action (enum: create, update, status\_change, approve, reject, assign, comment)
* details (text)

### **Comment**

* id (int)
* task\_id (int)
* user\_id (int)
* timestamp (datetime)
* text (text)

# 5. UI Components

* Objective List/Table (filter, sort, search, batch actions)
* Objective detail window (modeless, resizable; fields: description, status, assignment, objective, audit, comments)
* Objective linkage panel (popover/modal for objective details)
* Approval workflow controls (buttons for approve/reject/return)
* Assignment dropdowns/dialogs
* Status quick-update controls
* Comments panel (threaded, with notifications)
* Audit trail timeline

## **Objective Detail Window Tabs**

### **Tab 1: Narrative**

* Read-only, sortable table with columns:
  + Date/Time
  + Entry
  + Entered By
  + Critical (rows highlight red if Yes)
* Narrative entries are text-editable only by Admin users.
* The “Critical” flag is editable by any user (Yes/No toggle; Yes highlights row red).

### **Tab 2: Strategies**

* Editable two-column table:
  + Strategy (free text; describes the specific action or approach)
  + Entered By (user who entered/updated the strategy)
* Users can add, edit, or delete strategies as needed.

### **Tab 3: Linked Tasks**

* Read-only list of tasks associated with this objective.
* Tasks can be associated with multiple objectives.
* Button at the top: [Create New Task] (automatically associates new task with this objective).
* Columns will be the same as those on the task status board.

### **Tab 4: Approvals**

* (To be defined: will handle approval workflow/status/actions.)

### **Tab 5: Customer**

* Information about the requesting customer/agency and contact details (free text fields for name, agency, contact info, request number, etc.).

### **Tab 6: Log**

* Full audit trail and comment history for this objective.
* Read-only, sorted chronologically.

# 6. API Endpoints

* GET /api/planning/objectives – List/filter all objectives
* POST /api/planning/objectives – Create new objective
* PUT /api/planning/objectives/{id} – Update/edit objective
* POST /api/planning/objectives/{id}/approve – Approve objective
* POST /api/planning/objectives/{id}/reject – Reject or return objective
* POST /api/planning/objectives/{id}/assign – Assign/reassign objective
* POST /api/planning/objectives/{id}/status – Update status
* POST /api/planning/objectives/{id}/comment – Add comment
* GET /api/planning/objectives/{id}/history – Fetch full audit log

# 7. Database Tables

* strategic\_tasks
* audit\_logs
* comments

# 8. Inter-Module Connections

* Objectives Module: Link objectives to originating objectives
* Personnel/Teams: Assign and track objective responsibility
* Forms Module: Strategic objectives populate relevant ICS forms (e.g., 202, 204)
* Status Boards: Show objective status and progress in dashboards
* Audit Trail: Log all actions system-wide

# 9. Special Features / Notes

* All changes and status transitions are audit-logged and timestamped
* Approval workflow enforces review before assignment
* Objective-objective linkage allows for comprehensive operational traceability
* Threaded comments enable collaboration and handoffs
* Objective Detail Window is modeless and resizable for side-by-side workflow
* Full export/reporting capability for incident reviews and after-action reports

Module 3: Operations

# 1. Module Name & Description

The Operations Module coordinates tactical field execution of tasks, tracks live team status, and manages assignments. It provides real-time situational awareness for operations leaders and integrates with Planning, Communications, and Personnel modules.

# 2. Primary Functions

* - View and manage task assignments from Planning
* - Track real-time task progress and field updates
* - Display personnel and unit assignments
* - Log field activity and status changes
* - Interact with communications and resource logs
* - Interface with ICS 204, 214, and 215 forms

# 3. Submodules / Tools

* - Task Action Log: Maintains a chronological activity history for each task, including status changes, comments, field updates, and attached logs
* - Team & Section Loggers (ICS 214): Each team/unit and the overall Operations section will have its own ICS 214-compatible logging view to track activities, changes, and incident actions
* - Rapid Assignment Tool: Drag-and-drop interface for assigning tasks to available units or personnel in real time. Useful for contingency response and last-minute adjustments.
* - Assignment Viewer: Displays active and upcoming tasks by section/unit
* - Task Status Updater: Allows field teams or supervisors to update status (Not Started, In Progress, Complete, Blocked)
* - Unit Status Tracker: Shows current position, availability, and team composition for each unit
* - Operations Log (ICS 214): Auto-populates based on actions, can be manually edited
* - Form Link Panel: Pulls in related ICS 204/215 data for task reference

# 4. Key Data Modules

* - TaskActionLog:
* - id, task\_id, author\_id, timestamp, action\_type, note, attachment\_url
* - TeamLog:
* - id, team\_id, entered\_by, text, timestamp
* - SectionLog:
* - id, section, entered\_by, text, timestamp
* - OpsAssignment:
* - id, task\_id, assigned\_to, start\_time, end\_time, status, notes
* - UnitStatus:
* - id, unit\_id, location, status, available\_personnel[], timestamp
* - OpsLogEntry:
* - id, incident\_id, entered\_by, text, timestamp, related\_task\_id

# 5. UI Components

* - Task board by unit or section
* - Assignment update interface
* - Unit availability dashboard
* - 214 log feed with filtering by user/task/unit

# 6. API Endpoints

* - GET /api/operations/assignments
* - POST /api/operations/assignments
* - PUT /api/operations/assignments/{id}
* - GET /api/operations/units
* - POST /api/operations/units
* - GET /api/operations/logs
* - POST /api/operations/logs

# 7. Database Tables

* - task\_action\_logs
* - team\_logs
* - section\_logs
* - ops\_assignments
* - unit\_status
* - ops\_log\_entries

# 8. Inter-Module Connections

* - Planning Module: Tasks are received and status updates are fed back
* - Personnel Module: Assignments linked to individuals and teams
* - Communications Module: Status updates may trigger alerts/messages
* - Forms Module: ICS 204 and 214 integration

# 9. Special Features / Notes

* - Every task maintains a self-contained action log with a chronological activity record
* - Each team/unit and the Operations section has a 214-based log stream to support official documentation and after-action reviews
* - Rapid Assignment Tool enables dynamic reassignment in response to emerging operational needs
* - Color-coded status tracking for quick situational scanning
* - Blocked tasks can be flagged and escalated
* - Integrated time tracking for resource management
* - Mobile-compatible views for field updates
* - Logs tied to individual personnel, teams, or specific tasks

Module 3-1: Taskings

## Module Name & Description

This submodule provides Operations personnel with a centralized interface to manage tactical and logistical taskings during an incident. It enables:

* **Task ingestion:** Receive assignment requests (reflex taskings) from Planning or Command
* **Task creation & editing:** Define new tasks with detailed attributes
* **Assignment & tracking:** Assign tasks to teams, units, or individuals and monitor status
* **Notifications & alerts:** Notify recipients via the Communications module
* **Form integration:** Quick-print relevant ICS forms (e.g., ICS 213 General Message, ICS 215 Operational Planning Worksheet, ICS 218 Resource Status Change)

## 2. Primary Functions

* **Task List Dashboard:** Paginated, sortable, and filterable grid showing all active and archived tasks
* **Task Detail View:** Full metadata display (description, priority, due time, assigned unit, creator, ICS form links)
* **New Task Wizard:** Step-by-step form to capture all required fields and suggest ICS form templates
* **Bulk Actions:** Enable batch assignment, status updates, or printing multiple form outputs
* **Real-time Status Updates:** Reflect changes instantly in the dashboard and send notifications
* **Role-Based Access:** Only Operations Officer and above can create or delete; Team Leads can update status

## 3. Submodules / Tools

1. Task Detail Window

## 4. Key Data Modules

| **Table Name** | **Purpose** | **Key Columns (non-exhaustive)** |
| --- | --- | --- |
| **tasks** | Core task records | id (PK), title, description, priority, status, task\_type\_id, category\_id, assigned\_to, created\_by, created\_at, due\_time, ics\_form\_link, updated\_at |
| **task\_types** | Lookup of task types (filtered by category) | id (PK), category\_id (FK), name |
| **categories** | Lookup of task categories | id (PK), name |
| **narrative\_entries** | Logs of narrative updates | id (PK), taskid (FK), timestamp, entry\_text, entered\_by, team\_number, critical\_flag, entered\_at |
| **task\_teams** | Which teams are assigned to each task | id (PK), taskid (FK), team\_id (FK), sortie\_number, status, assigned\_ts, briefed\_ts, enroute\_ts, arrival\_ts, discovery\_ts, complete\_ts |
| **teams** | Master list of teams | id (PK), name, leader\_id (FK→personnel), leader\_phone |
| **task\_personnel** | Personnel on a task | id (PK), task\_id (FK), personnel\_id (FK), role, organization, time\_assigned |
| **personnel** | Master list of personnel | id (PK), name, phone, organization, … |
| **task\_vehicles** | Vehicles on a task | id (PK), taskid (FK), vehicle\_id (FK) |
| **vehicles** | Master list of vehicles | id (PK), type, identifier, … |
| **assignment\_ground** | “Ground Information” subtabs for task | taskid (PK & FK), prev\_search\_efforts, time\_allocated, size, pod\_responsive, pod\_unresponsive, pod\_clues, dropoff\_instructions, pickup\_instructions |
| **assignment\_air** | “Air Information” subtabs for task | taskid (PK & FK), wmirs\_area, dep\_airport, dest\_airport, etd, ete, … (all fields from Air Information) |
| **radio\_channels** | Radios tied to a task (ICS 205 fields) | id (PK), taskid (FK), channel\_name (FK→ics205\_channels), function, zone, channel\_number, rx\_freq, rx\_tone, tx\_freq, tx\_tone, mode, remarks |
| **ics205\_channels** | Master list of ICS 205 channel definitions | channel\_name (PK & FK), zone, channel\_number, rx\_freq, rx\_tone, tx\_freq, tx\_tone, mode, remarks |
| **debriefs** | Debrief header records | id (PK), task\_id (FK), sortie\_number, debriefer\_id (FK→personnel), debrief\_type, created\_at, updated\_at |
| **debrief\_ground\_sar** | Ground-SAR debrief detail | debrief\_id (PK & FK), assignment\_summary, search\_efforts, unable\_to\_search, clues\_tracks\_signs, hazards\_encountered, suggestions, time\_entered, time\_exited, time\_spent, conditions\_clouds, precipitation, light\_conditions, visibility, terrain, ground\_cover, wind\_speed |
| **debrief\_area\_search** | Area Search Supplement | debrief\_id (PK & FK), num\_searchers, time\_spent, search\_speed, area\_size, spacing, visibility\_distance, vis\_determination, areas\_skipped, pattern\_description, additional\_comments |
| **debrief\_tracking** | Tracking Team Supplement | debrief\_id (PK & FK), likelihood, existing\_trap\_desc, erased\_trap\_desc, new\_trap\_desc, route\_taken, discontinue\_reason |
| **debrief\_hasty\_search** | Hasty Search Supplement | debrief\_id (PK & FK), visibility, attract\_methods, hearing\_conditions, trail\_conditions, offtrail\_conditions, map\_accuracy, found\_features, tracking\_conditions, hazards\_attractions |
| **debrief\_air\_general** | Air (General) Debrief | debrief\_id (PK & FK), flight\_plan\_closed, atd, ata, hobbs\_start, hobbs\_end, hobbs\_total, tach\_start, tach\_end, fuel\_used, oil\_used, fuel\_oil\_cost, receipt\_no, summary, results, weather, remarks, sortie\_effectiveness, not\_success\_reason |
| **debrief\_air\_sar** | Air (SAR Worksheet) Debrief | debrief\_id (PK & FK), area\_name, grid, nw\_lat, nw\_long, ne\_lat, ne\_long, sw\_lat, sw\_long, se\_lat, se\_long, pattern, visibility\_nm, altitude\_agl, speed\_knots, track\_spacing, terrain, cover, turbulence, pod, time\_to\_search, time\_started, time\_ended, time\_in\_area, time\_from\_area, total\_sortie\_time, crew\_remarks, effectiveness\_rating, visibility\_rating |
| **audit\_logs** | Change log entries | id (PK), taskid (FK), timestamp, field\_changed, old\_value, new\_value, changed\_by |
| **attachments** | Uploaded files and form outputs | id (PK), taskid (FK), filename, file\_type, uploaded\_by, timestamp, size, version |
| **planning\_links** | Links tasks to strategic goals | id (PK), taskid (FK), goal\_id (FK→strategic\_goals), checklist\_data (JSON) |
| **strategic\_goals** | Master list of planning goals | id (PK), name, description |

## 5. UI Components

### 5.1 Task Screen Layout

* **Main Status Board:** Full-page, row-based view where each task occupies one row; column headers allow sorting and reordering. (Columns to be defined later.)
* **Row Interaction:** Single-click selects a row; double-click opens the Task Detail window.

### 5.2 Task Detail Window

**Design Note:** Layout and element grouping should draw inspiration from the CAD-style screenshot provided, but this submodule’s UI should not be an exact copy—use our app’s theme, clean spacing, and simplified controls.

A floating, draggable window containing:

* **Window Background:** Color varies by Task Category.
* **Top Toolbar (4 dropdowns):**
  1. **Category:** Dropdown with predefined options: <New Task>(Default)/Air SAR/Damage Assessment/Ground SAR/Logistics Support/Other/Photography/Relocation
  2. **Task Type:** Dropdown list populated from a predefined CSV of task types, dynamically filtered based on the selected **Category**.
  3. **Priority:** Predefined choices: Low, Medium, High, Critical.
  4. **Status (Task):** Predefined workflow states: Draft, Planned, In Progress, Completed, Cancelled.  
     *Note:* This dropdown controls the overall task status only; the **Status** column in the Teams tab uses its own set of team-specific statuses.
  5. **Task ID:** Free-text input for entering or overriding the unique Task Identifier.
* **Location Bar**:\*\* Single-line free-text input for location or assignment name.
* **Narrative Entry Box:** Large multiline input; pressing Enter appends to the narrative log.

**Dropdown Options Reference:**

* **Categories:** <New Task>(Default)/Air SAR/Damage Assessment/Ground SAR/Logistics Support/Other/Photography/Relocation
* **Priorities:** Low, Medium, High, Critical
* **Statuses (Task):** Draft, Planned, In Progress, Completed, Cancelled Low, Medium, High, Critical

**Bottom Tab Panel**

The Task Detail window includes a tab panel with ten tabs:

1. **Narrative**  
   Read‑only, sortable table with columns:
   * **Date/Time**
   * **Entry**
   * **Entered By**
   * **Team Name** (Only populates if the person who made the entry is assigned to a team)
   * **Critical** (rows highlight red if Yes)
   * **ICS 214 Export** button  
     *Text editable only by admin; Critical flag editable by any user.*
2. **Teams**  
   Sortable table with columns:
   * **Sortie Number** (optional)(Manually entered here)
   * **Team Name**
   * **Team Leader**
   * **Team Leader Phone Number**
   * **Status:** Dropdown presenting only the statuses valid for the task’s category or type, enabling quick manual updates.
   * **Assigned:** Date/time stamp set when assigned.
   * **Briefed:** Date/time stamp set on “Briefed.”
   * **Enroute:** Date/time stamp set on “En Route.”
   * **Arrival:** Date/time stamp set on “On Scene.”
   * **Discovery:** Date/time stamp set on “Discovery/Find.”
   * **Complete:** Date/time stamp set on “Complete.”
   * **Primary:** Boolean flag that indicates if the team is primary on the task. Only one team may be primary on a task. The first team assigned to a task is automatically marked as primary.  
     **Actions:** Add Team, Remove Team, Change Status (triggers timestamp update).
3. **Personnel**  
   Table view, sortable by any column:
   * **Auto‑Populate:** Entries derive from team assignments; manual add/remove allowed.
   * **Columns:**
     + **ID**
     + **Name**
     + **Role**
     + **Phone Number**
     + **Organization**
     + **Team Name**
     + **Time Assigned**  
       **Actions:** Add/Remove Personnel, Edit Time Assigned (inline date/time picker).  
       **Features:** Filtering by Organization/Role; Bulk operations.
4. **Vehicles**  
   Table view, sortable by any column:
   * **Auto‑Populate:** Entries derive from team assignments; manual add/remove not permitted.
   * **Columns:**
     + **Vehicle ID**
     + **Type**
     + **Team Assignment**  
       **Actions:** Reassign Vehicle (inline dropdown).  
       **Features:** Live status indicator; Inventory filter.
5. **Assignment Details**  
   *Goal:* Support CAPF 109, SAR 104, and ICS 204 by capturing form-specific details not located elsewhere in the tab.  
   Contains two subtabs:
   * **Ground Information** (default)
     + Previous and Present Search Efforts in Area
     + Time Allocated
     + Size of Assignment
     + Expected POD
       1. Options of High/Medium/Low in categories of Responsive Subj, Unresponsive Subj, Clues
     + Drop off and Pickup instructions
   * **Air Information** — Aviation assignment details
     + WMIRS Area of Operations
     + Dep. Airport
     + Dest. Airport
     + ETD
     + ETE
     + Other Aircraft in Area (Location & Callsign)
     + Ground Teams in Area (Location & Callsign)
     + Sortie Objectives
     + Sortie Deliverables
     + Actions To Be Taken on Objectives & Deliverables
     + Route of Flight
     + Altitude Assignment & Restrictions
     + Airspeed Expected & Restrictions
     + Aircraft Separation (Adjoining Areas)
     + Emergency/Alternate Fields
     + Military Low Altitude Training Routes
     + Hazards to Flight
     + Sortie Search Plan
       1. Search Pattern
       2. Search Visibility (NM)
       3. Search Altitude (AGL)
       4. Search Speed (Knots)
       5. Track Spacing (NM)
       6. Terrain
          1. Flat/Rolling Hills/Rugged Hills/Mountainous
       7. Cover
          1. Open/Moderate/Heavy/Light Snow/Heavy Snow
       8. Turbulence
          1. Light/Moderate/Heavy
       9. Probability of Detection
       10. Time to Search Area
       11. Time Started Search
       12. Time Ended Search
       13. Time in Search Area
       14. Time from Search Area
       15. Time From Search Area
       16. Total Sortie Time
6. **Communications**  
   Table mirroring ICS 205 Basic Radio Channel Use; only **Channel Name** and **Function** editable, others read‑only. Data populates from ICS 205 via selected channel:
   * **Channel Name** (dropdown)
   * **Zone**
   * **Channel Number**
   * **Function** (editable)
   * **RX Frequency**
   * **RX Tone/NAC**
   * **TX Frequency**
   * **TX Tone/NAC**
   * **Mode** (A/D/M)
   * **Remarks**
7. **Debriefing**  
   Post-task review interface capturing debrief notes, lessons learned, and personnel feedback. Initial screen contains a log of completed debriefs. Button at the top labeled “Add Debrief”. This opens a window asking for the sortie number, Debriefer (Enter ID to reference name) and a checklist of different debrief types. Types include Ground (SAR), Area Search Supplement, Tracking Team Supplement, Hasty Search Supplement, Air (General), Air (SAR Worksheet). Ground (SAR) must be selected in order for area search, tracking team, or hasty search to be selected. Air (General) must be selected in order for Air (SAR Worksheet) to be selected. This check list selects which forms will be displayed. After entering sortie number, debriefer ID, and selecting forms, the user clicks create which opens up the appropriate forms. After saving the form, debriefs may be edited. Information is sent to the planning debrief module and flagged for review.
   * **Ground (SAR)**
     + Assignment Summary (Free Text)
     + Describe Search Efforts in Assignment (Free Text)
     + Describe Portions Unable to Search (Free Text)
     + Describe Clues/Tracks/Signs or any Interviews (Free Text)
     + Describe any Hazards or Problems Encountered (Free Text)
     + Suggestions for Further Search Efforts In or Near Assignment (Free Text)
     + Time Entered
     + Time Exited
     + Time Spent
     + Conditions
       1. Clouds
          1. Clear/Scattered/Broken/Overcast
       2. Precipitation
          1. None/Rain/Scattered/Snow
       3. Light Conditions
          1. Bright/Dull/Near Dark/Night
       4. Visibility
          1. > 10 Miles/> 5 Miles/> 1 Mile/< 1 Mile
       5. Terrain
          1. Flat/Rolling Hills/Rugged Hills/Mtns
       6. Ground Cover
          1. Open/Moderate/Heavy/Other
       7. Wind Speed
          1. Calm/< 10 mph/< 20 mph/< 30 mph
     + Attachments
       1. Debriefing Maps
       2. Original Briefing Document
       3. Supplemental Debriefing Forms
       4. Interview Log
       5. Other
   * **Area Search Supplement**
     + Number of Searchers
     + Time Spent Searching
     + Search Speed
     + Area Size (Actually Searched)
     + Spacing
     + Visibility Distance (Free Text)
     + How was Visibility Distance Determined (Free Text)
     + Types of Areas Skipped Over (Ie Heavy Brush, Wetlands, Cliffs) (Free Text)
     + Describe the Direction and Pattern of your Search (Free Text)
     + Comments for Additional Area Searching of this Assignment (Free Text)
   * **Tracking Team Supplement**
     + Discuss Likelihood of Finding Tracks or Sign on the Trails (Free Text)
     + Describe the Location and Nature of Existing Track Traps (Free Text)
     + Did You Erase Any Track Traps (Free Text)
     + Did You Create Any New Track Traps (Free Text)
     + Describe the Route Taken by Any Tracks You Followed (Free Text)
     + Why Did You Discontinue Following These Tracks (Free Text)
     + Attachments
       1. Individual Track Sketches Attached
       2. Track Trap Summary Sketches Attached
   * **Hasty Search Supplement**
     + Visibility During Search (Day/Dusk/Night/Other) (Free Text)
     + Describe Your Efforts to Attract a Responsive Subject (Free Text)
     + Describe Ability to Hear a Response (Background Noise) (Free Text)
     + Describe the Trail Conditions (Free Text)
     + Describe the Off-Trail Conditions (Free Text)
     + Does the Map Accurately Reflect the Trails (Free Text)
     + Did You Locate Features That Would Likely Contain the Subject (Free Text)
     + How Are the Tracking Conditions (Free Text)
     + Describe any Hazards or Attractions You Found (Free Text)
   * **Air (General)**
     + Flight Plan Closed (Boolean Yes/No)
     + ATD
     + ATA
     + Hobbs Start
     + Hobbs End
     + Hobbs To/From
     + Hobbs in Area
     + Hobbs Total
     + Tach Start
     + Tach End
     + Fuel Used (Gal)
     + Oil Used (Qt)
     + Fuel & Oil Cost
     + Receipt #
     + Summary (Free Text)
     + Results/Deliverables (Free Text)
     + Weather Conditions (Free Text)
     + Remarks (Free Text)
     + Sortie Effectiveness
       1. Successful/Marginal/Unsuccessful/Not Flown/Not Required
     + Reason (if not successful)
       1. Weather/Crew Unavailable/Aircraft Maintenance/Customer Cancellation/Equipment Failure/Other
     + Attachments/Documentation
       1. CAPF 104A SAR
       2. CAPF 104B Recon Summary
       3. ICS 214 Unit Log
       4. Receipts
       5. AIF ORM Matrix
   * **Air (SAR Worksheet)**
     + Search Area
       1. Name
       2. Grid
       3. NW Corner (Lat/Long)
       4. NE Corner (Lat/Long)
       5. SW Corner (Lat/Long)
       6. SE Corner (Lat/Long)
     + Sortie Search Actual
       1. Search Pattern
       2. Search Visibility (NM)
       3. Search Altitude (AGL)
       4. Search Speed (Knots)
       5. Track Spacing (NM)
       6. Terrain
          1. Flat/Rolling Hills/Rugged Hills/Mountainous
       7. Cover
          1. Open/Moderate/Heavy/Light Snow/Heavy Snow
       8. Turbulence
          1. Light/Moderate/Heavy
       9. Probability of Detection
       10. Time to Search Area
       11. Time Started Search
       12. Time Ended Search
       13. Time in Search Area
       14. Time from Search Area
       15. Time From Search Area
       16. Total Sortie Time
     + Crew Remarks and Notes
       1. Effectiveness
          1. Excellent/Good/Fair/Poor
       2. Visibility
          1. Excellent/Good/Fair/Poor
8. **Log**  
   System audit trail, sortable table with columns:
   * **Timestamp**
   * **Field Changed**
   * **Old Value**
   * **New Value**
   * **Changed By**  
     **Features:** Filtering by date/field; Export CSV; Keyword search.
9. **Attachments/Forms**  
   Table with columns:
   * **Filename**
   * **Type**
   * **Uploaded By**
   * **Timestamp**
   * **Size**  
     **Actions:** Upload File, Download/Preview, Annotate.  
     **Features:** Version control; File size warnings.  
     **Forms Generation:** **Generate Forms** button for CAPF 109, SAR 104, ICS 204, etc.
10. **Planning**
    * **Purpose:** Link task to strategic goals; provide checklist for planning items.
    * **Fields:** TBD.

**Close/Minimize Controls:** Standard window chrome for hiding or closing the window.:\*\* Standard window chrome for hiding or closing the window.

## 6. API Endpoints

### 1. Tasks

| **Method** | **Path** | **Description** | **Request Body** | **Response** |
| --- | --- | --- | --- | --- |
| GET | /api/operations/taskings | List tasks (filtering, paging, sorting) | Query params: status, priority, assigned\_to, date\_from, date\_to, page, page\_size, sort\_by | { tasks: Task[], total: number, page, page\_size } |
| POST | /api/operations/taskings | Create a new task | { title, description, category\_id, task\_type\_id, priority, assigned\_to, due\_time?, task\_id? } | { id: UUID, ...full Task } |
| GET | /api/operations/taskings/{taskId} | Retrieve a single task (with nested sub-resources) | — | { ...full Task, narrative:[], teams:[], ... } |
| PUT | /api/operations/taskings/{taskId} | Update all metadata on a task | { title?, description?, category\_id?, task\_type\_id?, priority?, assigned\_to?, due\_time?, task\_id?, ics\_form\_link? } | { ...updated Task } |
| PATCH | /api/operations/taskings/{taskId}/status | Update only the task’s status | `{ status: Draft | Planned |
| DELETE | /api/operations/taskings/{taskId} | Archive/delete a task | — | 204 No Content |

### 2. Narrative Entries

| **Method** | **Path** | **Description** | **Request Body** | **Response** |
| --- | --- | --- | --- | --- |
| GET | /api/operations/taskings/{taskId}/narrative | List narrative log entries | Query params: sort\_by, order, page, page\_size | { entries: NarrativeEntry[], total } |
| POST | /api/operations/taskings/{taskId}/narrative | Append a new narrative entry | { entry\_text, entered\_by, team\_number?, critical\_flag? } | { id: number, timestamp, ... } |
| PATCH | /api/operations/taskings/{taskId}/narrative/{entryId} | Update critical flag (any user) | { critical\_flag: boolean } | { id, critical\_flag, timestamp } |

### 3. Team Assignments

| **Method** | **Path** | **Description** | **Request Body** | **Response** |
| --- | --- | --- | --- | --- |
| GET | /api/operations/taskings/{taskId}/teams | List teams on a task | — | { teams: TaskTeam[] } |
| POST | /api/operations/taskings/{taskId}/teams | Assign a new team | { team\_id, sortie\_number? } | { id: number, assigned\_ts } |
| PATCH | /api/operations/taskings/{taskId}/teams/{ttId} | Change team status (auto-stamp) | { status: string } | { id, status, updated\_ts } |
| DELETE | /api/operations/taskings/{taskId}/teams/{ttId} | Remove a team assignment | — | 204 No Content |

### 4. Personnel

| **Method** | **Path** | **Description** | **Request Body** | **Response** |
| --- | --- | --- | --- | --- |
| GET | /api/operations/taskings/{taskId}/personnel | List personnel on a task | — | { personnel: TaskPersonnel[] } |
| POST | /api/operations/taskings/{taskId}/personnel | Add a person | { personnel\_id, role, organization, time\_assigned? } | { id, time\_assigned } |
| PATCH | /api/operations/taskings/{taskId}/personnel/{pId} | Update time assigned | { time\_assigned: ISODateTime } | { id, time\_assigned } |
| DELETE | /api/operations/taskings/{taskId}/personnel/{pId} | Remove from task | — | 204 No Content |

### 5. Vehicles

| **Method** | **Path** | **Description** | **Request Body** | **Response** |
| --- | --- | --- | --- | --- |
| GET | /api/operations/taskings/{taskId}/vehicles | List vehicles on a task | — | { vehicles: TaskVehicle[] } |
| POST | /api/operations/taskings/{taskId}/vehicles | Assign a vehicle | { vehicle\_id } | { id, task\_id, vehicle\_id } |
| PATCH | /api/operations/taskings/{taskId}/vehicles/{vId} | Reassign to a different team | { team\_assignment: string } | { id, team\_assignment } |
| DELETE | /api/operations/taskings/{taskId}/vehicles/{vId} | Remove a vehicle | — | 204 No Content |

### 6. Assignment Details

| **Method** | **Path** | **Description** | **Request Body** | **Response** |
| --- | --- | --- | --- | --- |
| GET | /api/operations/taskings/{taskId}/assignment | Fetch both ground & air details | — | { ground: {...}, air: {...} } |
| PUT | /api/operations/taskings/{taskId}/assignment/ground | Update ground information | { prev\_search\_efforts, time\_allocated, size, expected\_pod\_responsive, expected\_pod\_unresponsive, expected\_pod\_clues, dropoff\_instructions, pickup\_instructions } | { ground: {...} } |
| PUT | /api/operations/taskings/{taskId}/assignment/air | Update air information | { wmirs\_area, dep\_airport, dest\_airport, etd, ete, /\* all other air fields \*/ } | { air: {...} } |

### 7. Communications

| **Method** | **Path** | **Description** | **Request Body** | **Response** |
| --- | --- | --- | --- | --- |
| GET | /api/operations/taskings/{taskId}/communications | List radio channels for task | — | { channels: RadioChannel[] } |
| POST | /api/operations/taskings/{taskId}/communications | Add a radio channel | { channel\_name, function } | { id, channel\_name, function } |
| PATCH | /api/operations/taskings/{taskId}/communications/{cId} | Update function for a channel | { function: string } | { id, function } |
| DELETE | /api/operations/taskings/{taskId}/communications/{cId} | Remove a radio channel | — | 204 No Content |

### 8. Debriefing

| **Method** | **Path** | **Description** | **Request Body** | **Response** |
| --- | --- | --- | --- | --- |
| GET | /api/operations/taskings/{taskId}/debriefs | List debrief records | — | { debriefs: Debrief[] } |
| POST | /api/operations/taskings/{taskId}/debriefs | Create new debrief | { sortie\_number, debriefer\_id, debrief\_type, /\* additional fields per type \*/ } | { id, created\_at } |
| PUT | /api/operations/taskings/{taskId}/debriefs/{dId} | Update an existing debrief | { /\* fields as above, per debrief type \*/ } | { id, updated\_at } |
| DELETE | /api/operations/taskings/{taskId}/debriefs/{dId} | Delete a debrief record | — | 204 No Content |

### 9. Audit Logs

| **Method** | **Path** | **Description** | **Request Body** | **Response** |
| --- | --- | --- | --- | --- |
| GET | /api/operations/taskings/{taskId}/logs | Fetch the change history | Query params: field?, date\_from?, date\_to? | { logs: AuditLog[] } |

### 10. Attachments/Forms

| **Method** | **Path** | **Description** | **Request Body** | **Response** |
| --- | --- | --- | --- | --- |
| GET | /api/operations/taskings/{taskId}/attachments | List all attachments and forms | — | { attachments: Attachment[] } |
| POST | /api/operations/taskings/{taskId}/attachments | Upload a file or form template | multipart/form-data: file, type, generate\_forms? | { id, filename, url } |
| DELETE | /api/operations/taskings/{taskId}/attachments/{aId} | Remove an attachment | — | 204 No Content |

### 11. Planning Links

| **Method** | **Path** | **Description** | **Request Body** | **Response** |
| --- | --- | --- | --- | --- |
| GET | /api/operations/taskings/{taskId}/planning | Fetch linked strategic goals | — | { links: PlanningLink[] } |
| POST | /api/operations/taskings/{taskId}/planning | Link a new strategic goal | { goal\_id, checklist\_data (JSON) } | { id, goal\_id, checklist\_data } |
| DELETE | /api/operations/taskings/{taskId}/planning/{linkId} | Unlink a strategic goal | — | 204 No Content |

All endpoints use the incident-specific database connection and emit real-time notifications via the Communications WebSocket.

## 7. Database Tables

## 8. Inter-Module Connections

**Communications**

* Send/receive real-time notifications and alerts when tasks are created, updated, or completed
* Populate the “Communications” tab’s radio channels data from the ICS 205 channel definitions
* Use the IM/213 messaging system for cross-team alerts

 **Planning**

* Ingest reflex taskings and strategic task requests from the Planning (Strategic Tasks) module
* Link tasks to strategic goals via the Planning tab (and push debrief outcomes back into Planning’s reviews)

 **Personnel & Role Management**

* Lookup and validate personnel IDs for “Created By,” “Entered By,” Debriefer, etc.
* Auto-populate the Personnel tab when assigning teams
* Enforce Role-Based Access (e.g., only Officers can create/delete tasks)

 **ICS Forms & Documentation**

* Retrieve form templates (ICS 204, ICS 213, CAPF 109, SAR 104, etc.)
* Generate and store completed forms via the Attachments/Forms tab’s “Generate Forms” action

 **Log/Audit Trail**

* Hook into the global Audit Logs system to record every CRUD operation on tasks, narrative entries, team assignments, debriefs, etc.
* Surface filtered log views in the Log tab

 **Status Boards**

* Optionally publish high-level task summaries to any Status Boards (e.g., overall Progress board)

 **Reference Library**

* Provide lookup help text or reference links for form instructions, radio channel standards, debrief checklists, etc.

 **Data Persistence/Core**

* Use the incident-specific database in core/data/ for storing all tasking tables and relationships
* Integrate with any shared “master” lookup tables (e.g., ICS 205 channels, personnel directory, vehicle inventory)

 **Strategic Tasks (Planning Submodule)**

* Accept “tasking” requests from the Strategic Tasks module and allow planners to mark items as “reflex taskings” back into Operations

## 9. Special Features / Notes

Module 4: Logistics

# 1. Module Name & Description

The Logistics Module manages all resource, supply, and equipment operations for an incident. It supports the entire lifecycle of resource requests—from creation and approval to assignment and fulfillment—while maintaining accurate real-time inventory of equipment, supplies, vehicles, and personnel-related logistical support. This module provides transparent workflows, audit trails, and critical support for both field and command personnel.

# 2. Primary Functions

- Resource Request Management: Submit, track, approve, and fulfill requests for personnel, equipment, and supplies.  
- Inventory Tracking: Manage inventory levels, check-in/check-out equipment, and maintain up-to-date availability.  
- Assignment Workflow: Assign resources and equipment to incidents, teams, or individuals; track status and location.  
- Fulfillment & Delivery Tracking: Coordinate delivery teams, monitor fulfillment progress, and provide real-time status updates.  
- Integration with Forms: Auto-generate and populate ICS forms (e.g., 213-RR, 204, 211, 218, 220) using logistics data.  
- Notifications & Alerts: Automated notifications for status changes, overdue equipment, or request fulfillment.  
- Audit & Change Log: Maintain a full history of requests, assignments, and inventory actions for transparency and accountability.

# 3. Submodules / Tools

- Resource Request Submodule: Handles creation, approval, assignment, and fulfillment of resource requests.  
- Equipment Submodule: Tracks non-vehicle equipment inventory, check-in/check-out status, and assignment.  
- Bulk Import/Export Tool: For onboarding large inventories or generating reports via CSV.  
- Assignment Dashboard: Visual overview of resource assignments and status.  
- Notification Center: Manages in-app, email, and SMS alerts for key logistics events.

# 4. Key Data Modules

- ResourceRequest: id, incident\_id, requestor\_id, timestamp, item\_code, quantity, priority, justification, status, due\_datetime, notes  
- ApprovalRecord: id, request\_id, approver\_id, action, timestamp, comments  
- AssignmentRecord: id, request\_id, resource\_id, assigned\_to\_id, assigned\_datetime, eta, status  
- ResourceItem: item\_code, description, unit, available\_quantity, location  
- EquipmentItem: id, name, type\_id, serial\_number, status, location, current\_holder\_id, tags, created\_at, updated\_at  
- CheckTransaction: id, equipment\_id, actor\_id (user or team), incident\_id, action (check\_out/check\_in), timestamp, notes

# 5. UI Components

- Resource Request Status Board: Table view with search, filters, color-coded status, and bulk actions.  
- Resource Request Detail Window: Modal for viewing/editing requests, with tabs for Order, Approval, Attachments, Supplier.  
- Equipment Inventory Table: Sortable/filterable view of all equipment items and status.  
- Check-In/Out Wizard: Guided UI for equipment assignment and return.  
- Assignment Dashboard: Overview panel for all ongoing assignments.  
- Bulk Import/Export Interface: Stepper for mapping/importing CSV files.

# 6. API Endpoints

- /api/logistics/requests (GET, POST): List, create resource requests  
- /api/logistics/requests/{id} (GET, PUT): Retrieve/update specific request  
- /api/logistics/requests/{id}/approvals (GET, POST): Approval workflow  
- /api/logistics/requests/{id}/assign (POST): Assign resources  
- /api/logistics/equipment (GET, POST): List/create equipment items  
- /api/logistics/equipment/{id} (GET, PUT, DELETE): Manage equipment item  
- /api/logistics/equipment/{id}/checkout (POST): Assign equipment  
- /api/logistics/equipment/{id}/checkin (POST): Return equipment  
- /api/logistics/items (GET): List available inventory items  
- /api/logistics/notifications (GET): Fetch logistics-related notifications

# 7. Database Tables

- logistics\_resource\_requests  
- logistics\_request\_approvals  
- logistics\_request\_assignments  
- logistics\_resource\_items  
- equipment\_items  
- check\_transactions  
- logistics\_notifications

# 8. Inter-Module Connections

- Planning: Shares resource requirements and receives forecasts for planning.  
- Operations: Notifies field teams, updates status on completion of tasks.  
- Finance/Admin: Tracks expenditures and cost approvals.  
- Personnel & Role Management: Lookup users for assignments and delivery.  
- Forms & Documentation: Generates and auto-populates relevant ICS forms.  
- Status Boards: Displays logistics statuses, overdue returns, and request tracking.

# 9. Special Features / Notes

- State Locking & Reactivation: Requests that are complete, cancelled, or denied are locked; denied/cancelled requests can be reactivated.  
- Offline Support: Mobile/offline data entry with sync upon reconnect.  
- Bulk Actions: Multi-select and bulk update for requests or equipment.  
- Attachment Support: Add photos, spec sheets, quotes to requests.  
- Customizable Priority Levels: Routine, Urgent, Emergency.  
- Audit Trail: Full change history on all records.  
- Role-Based Access Control: Restricts critical actions to authorized users.  
- Integration with Mobile App: Supports mobile subincidents and updates.

Module 4-1: Resource Request

# Resource Request Module Design Document

## 1. Module Name & Description

\*\*Logistics Resource Request\*\* is the submodule within the Logistics section that enables personnel to create, track, manage, and fulfill requests for resources (personnel, equipment, supplies) during an incident. It streamlines the request-to-delivery workflow, ensuring timely provisioning and transparent status updates.

## 2. Primary Functions

* \*\*Request Creation:\*\* Capture details for new resource requests, including item, quantity, priority, justification, and required delivery time.
* \*\*Approval Workflow:\*\* Route requests through one or more approvers with options to \*\*Approve\*\*, \*\*Deny\*\*, or \*\*Return for Information\*\*.
* \*\*Assignment & Fulfillment:\*\* Allocate available resources, assign delivery teams, and track fulfillment status.
* \*\*Status Tracking:\*\* Provide real-time dashboards showing pending, approved, in-progress, and completed requests.
* \*\*Notifications & Alerts:\*\* Notify stakeholders of status changes, upcoming delivery windows, or issues.
* \*\*Form Integration:\*\* Automatically generate and print ICS 213-RR forms based on request data.
* \*\*Request Deletion:\*\* Enable users to delete resource requests from the system.
* \*\*Request Reactivation:\*\* Allow cancelled or denied requests to be reactivated, resetting their status to Submitted.

## 3. Submodules/Tools

* \*\*Request Form Builder:\*\* Guided UI for entering request details.
* \*\*Approval Panel:\*\* View, comment on, and action pending requests.
* \*\*Assignment Dashboard:\*\* Visual interface (list/map) for allocating resources and scheduling deliveries.
* \*\*Fulfillment Tracker:\*\* Timeline view showing each step from approval to delivery.
* \*\*Notification Center:\*\* Configurable alerts via in-app, email, or SMS.
* \*\*Report Generator:\*\* Export request logs or summaries in PDF/Excel.

## 4. Key Data Models

* \*\*ResourceRequest\*\* (id, incident\_id, requestor\_id, timestamp, item\_code, quantity, priority, justification, status, due\_datetime, notes)

- \*\*status (enum):\*\* Submitted, In Progress, Approved, Ordered, Fulfilled, Complete, Cancelled, Denied

* \*\*ApprovalRecord\*\* (id, request\_id, approver\_id, action, timestamp, comments)
* \*\*AssignmentRecord\*\* (id, request\_id, resource\_id, assigned\_to\_id, assigned\_datetime, eta, status)
* \*\*ResourceItem\*\* (item\_code, description, unit, available\_quantity, location)
* \*\*Notification\*\* (id, request\_id, recipient\_id, type, sent\_timestamp, read\_flag)

## 5. UI Components

* \*\*Global Search Box:\*\* An input field above the status board to quickly search requests by Request ID or keywords.
* \*\*Resource Requests Status Board:\*\* Full-width table where each row represents a resource request.

- \*\*Configurable Columns & Saved Views:\*\* Users can hide/show columns, reorder them, and save custom board views (e.g., "My Pending", "Urgent Only").

- \*\*Default Column Visibility:\*\* A sensible set of columns is shown by default; users may customize visibility via a column manager.

- \*\*Sorting & Filtering:\*\* Clickable column headers support ascending/descending sort; header dropdowns allow per-column filter criteria.

- \*\*Color-Coded Status Badges:\*\* Status values are displayed as colored badges (e.g., green for Complete, red for Denied/Cancelled, blue for In Progress) for quick scanning.

- \*\*Row Lock & Grey-Out:\*\* Rows with status \*\*Complete\*\*, \*\*Cancelled\*\*, or \*\*Denied\*\* are greyed out and locked, disabling selection and inline actions.

* \*\*New Request Button:\*\* Above the status board; opens the Resource Request Detail Window in modal or modeless mode.
* \*\*Inline Filters Bar:\*\* Quick-access buttons above the table for common filters (e.g., Emergency, Pending).
* \*\*Bulk Action Toolbar:\*\* Appears when rows are selected; actions include bulk approve, bulk assign, delete requests, or reactivate cancelled/denied requests. Each destructive action prompts a confirmation dialog to prevent accidental operations.
* \*\*Resource Request Detail Window:\*\*

- \*\*State Locking & Reactivation:\*\*

- If status is \*\*Complete\*\*, window is fully read-only: fields disabled and action toolbar hidden.

- If status is \*\*Cancelled\*\* or \*\*Denied\*\*, show a \*\*Reactivate\*\* button to reset status to Submitted.

- \*\*Overview Panel (Top):\*\* Read-only display of core fields, including multi-role approval timestamps:

- Request Number

- Requestor

- Date/Time Submitted

- Category

- Priority

- Current Status

- Due Date/Time

- Section Chief/Command Staff Approval: Approver Name, Date/Time

- RESL Approval: Approver Name, Date/Time

- Logistics Approval: Approver Name, Date/Time

- Finance Section Approval: Approver Name, Date/Time

- \*\*Action Toolbar: Buttons for \*\*Edit Request\*\*, \*\*Print ICS 213-RR\*\*, \*\*Add Attachment\*\* (hidden when locked).

- \*\*Tabbed Content (Bottom): Four tabs with conditional read-only behavior:

1. Order: A table with columns for Quantity, Kind, Type, Priority, Detailed Item Description, Requested Reporting Location, Date/Time, Order Number, ETA, and Cost. Read-only when status is ≥ Approved.

2. Approval: Chronological approval history with approver name, action (Approve/Deny/Return for Information), timestamp, and comments. Logistics approval can include Requisition/Purchase Order Number. Sequential rule: logistics and finance cannot approve until section chief approval.

3. Attachments: File upload gallery and notes area for spec sheets, photos, or quotes. Always editable unless request is Complete.

4. Supplier: A table listing supplier contacts with columns Name, Phone, Email, and Fax. Read-only when status is ≥ Fulfilled.

## 6. API Endpoints

| Method | Endpoint | Description |

| ------ | --------------------------------------- | --------------------------------------------------- |

| GET | `/api/logistics/requests` | List all resource requests |

| POST | `/api/logistics/requests` | Create a new request |

| GET | `/api/logistics/requests/{id}` | Retrieve details for a single request |

| PUT | `/api/logistics/requests/{id}` | Update request (status, notes) |

| GET | `/api/logistics/requests/{id}/approvals`| Get approval history for a request |

| POST | `/api/logistics/requests/{id}/approve` | Submit an approval action |

| POST | `/api/logistics/requests/{id}/assign` | Assign resources to a request |

| GET | `/api/logistics/items` | List available resource items |

| GET | `/api/logistics/notifications` | Fetch user notifications |

| GET | `/api/logistics/requests/stream` | Stream real-time request updates via Server-Sent Events |

## 7. Database Tables

* logistics\_resource\_requests
* logistics\_request\_approvals
* logistics\_request\_assignments
* logistics\_resource\_items
* logistics\_notifications

## 8. Inter-Module Connections

* Planning: Share resource requirements for forecasting.
* Operations: Notify field teams about deliveries; update status on completion.
* Finance/Admin: Send cost estimates; track expenditures per request.
* Personnel & Role Management: Lookup users (requestors, approvers, delivery teams).
* Forms & Documentation: Auto-populate ICS forms (213-RR, 220) from request data.

## 9. Special Features/Notes

* Configurable Priority Levels: e.g., Routine, Urgent, Emergency.
* Audit Trail: Full change log for each request, approval, and assignment; drill-down links show before/after data.
* Attachment Support: Upload photos, spec sheets, vendor quotes.
* Offline Mode: Mobile devices can sync offline and update when online.
* External API Integration: Optional connectors to inventory systems.
* Quick-Add Templates: Save common request profiles for reuse.
* Keyboard Shortcuts Ctrl+N for New Request, Ctrl+P for Print ICS form, etc.
* Validation Flows: Before moving to Ordered, system validates that Section Chief, RESL, Logistics, and Finance approvals are present; missing approvals trigger an error dialog listing required steps.
* Column Views & Persistence:\*\* Saved board views persist per-user and across sessions.

Module 5: Communications

# 1. Module Name & Description

The Communications Module manages all inbound, outbound, and internal messaging across the incident. It includes informal chat, structured message logging (ICS 213), and broadcast tools for alerts and status changes. It links directly with the Communications Unit Leader’s responsibilities and supports message traceability.

# 2. Primary Functions

* - Facilitate unit-to-unit and user-to-user communications
* - Log formal messages via ICS 213
* - Manage communication assets (e.g., radio frequencies)
* - Broadcast incident-critical alerts
* - Track communications activity and message acknowledgments

# 3. Submodules / Tools

* Chat Interface: Informal IM-style messaging between users or teams; includes timestamps and read status
* General Message 213 System: Structured message subincident with support for replies, approvals, attachments, and priority flags
* Notification Dispatcher: Sends broadcast alerts, task change updates, and critical system messages
* Comms Log Viewer: Displays chronological feed of communications logs, filterable by user, time, priority, or type
* Incident Communications Plan (ICS-205): Track assigned frequencies, channel names, mode, usage, and assignments
* Communications Resource Availability Worksheet (ICS-217): Saves a prebuilt list of radio channels for use in creating the ICS-205

# 4. Key Data Modules

* - Message
* - id, sender\_id, receiver\_id/group\_id, content, timestamp, read[], priority, ref\_num, attachments[]
* - ICS213Message
* - id, subject, originator, approved\_by, replied\_by, sent\_time, message\_text, reply\_text, attachments[]
* - AlertNotification
* - id, type, content, timestamp, recipients[]
* - RadioChannel
* - id, alpha\_tag, function, system, mode, line\_a, line\_c

# 5. UI Components

* - Multi-user chat interface
* - ICS 213 form generator/editor
* - Notification panel with filtering and history
* - Frequency/channel planner table
* - Read receipts and threaded replies for structured messages

# 6. API Endpoints

* - GET /api/comms/messages
* - POST /api/comms/messages
* - GET /api/comms/213
* - POST /api/comms/213
* - GET /api/comms/alerts
* - POST /api/comms/alerts
* - GET /api/comms/radio
* - POST /api/comms/radio

# 7. Database Tables

* - messages
* - ics\_213\_messages
* - alerts
* - radio\_channels

# 8. Inter-Module Connections

* - Operations Module: Message routing tied to task status changes
* - Planning Module: Critical task updates trigger alert messages
* - Personnel Module: User presence/status and group targeting
* - Forms Module: ICS 213 and 217 integration

# 9. Special Features / Notes

* - ICS 213 messages support message chain threading and read receipts
* - All communications are time-synced to incident clock or GPS/NTP
* - High-priority alerts override mute or standby settings
* - Channels can be filtered by group, location, or operational period
* - Supports optional encryption for sensitive messages

Module 6: Medical and Safety

# 1. Module Name & Description

This module consolidates all health, injury, responder safety, and medical support tracking. It enables on-site triage tracking, responder health logs, safety briefings, and medical plan management. It serves both Medical Unit Leaders and Safety Officers.

# 2. Primary Functions

* - Create and manage the Medical Plan (ICS 206)
* - Track personnel injuries or exposures
* - Log responder safety observations or concerns
* - Manage field medical unit location and availability
* - Log patient treatment and triage (optional submodule)
* - Display hazard zones or restricted areas

# 3. Submodules / Tools

* - CAP ORM Form Generator: Supports creation of Civil Air Patrol-specific ORM (Operational Risk Management) safety forms in compliance with CAP regulations. Auto-fills available incident, activity, and personnel data where possible
* - Medical Plan Builder (ICS 206): Auto-populates known hospitals, contact info, med evac, and personnel data
* - Responder Injury/Exposure Log: Tracks medical incidents tied to personnel, location, and treatment history
* - Safety Observation Reporter: Input system for logging safety issues, with flagging system for high-risk findings
* - Hazard Mapping Panel: Display and update unsafe zones with notes, linked to GIS overlay if available
* - Triage Record Tracker (Optional): Tracks patients encountered by field teams, their triage category, treatment, and disposition
* - Safety Briefing Scheduler: Tool to schedule and distribute safety messages per period/team

# 4. Key Data Modules

* - MedicalIncident
* - id, person\_id, type, time, description, treatment\_given, evac\_required, reported\_by
* - SafetyReport
* - id, time, location, severity, notes, flagged, reported\_by
* - TriageEntry
* - id, patient\_tag, location, triage\_level, time\_found, treated\_by, notes, disposition
* - HazardZone
* - id, name, coordinates[], severity, description

# 5. UI Components

* - CAP ORM form generator/editor with pre-fill from incident context
* - ICS 206 form generator
* - Safety log with filtering and severity flags
* - Injury reporting form linked to personnel
* - Patient triage input sheet
* - GIS-linked hazard zone editor
* - Safety message timeline view

# 6. API Endpoints

* - GET /api/safety/caporm
* - POST /api/safety/caporm
* - GET /api/medical/incidents
* - POST /api/medical/incidents
* - GET /api/medical/triage
* - POST /api/medical/triage
* - GET /api/safety/reports
* - POST /api/safety/reports
* - GET /api/safety/zones
* - POST /api/safety/zones

# 7. Database Tables

* - cap\_orm\_forms
* - medical\_incidents
* - safety\_reports
* - triage\_entries
* - hazard\_zones

# 8. Inter-Module Connections

* - Forms Module: ICS 206 and CAPF 160 form generation
* - Personnel Module: Injury logs tied to individual responders
* - Operations Module: Hazard zones and incidents affect task status or safety flags; allows injection of safety notes into task briefings
* - Planning Module: Safety messages integrated into IAP

# 9. Special Features / Notes

* - CAP ORM generator allows units operating under Civil Air Patrol protocols to generate and archive official safety forms
* - All injury and exposure events generate audit trail entries
* - Safety issues marked “flagged” appear in real-time alerts
* - Triage entries can optionally use barcode/NFC wristbands for field input
* - Medical plan pulls preloaded hospital and resource info
* - Works offline and syncs when connectivity is restored

Module 7: Intel

# 1. Module Name & Description

**Intel Module** — Core hub for **clue management** and **collection, analysis, and dissemination of intelligence** to staff and teams. It captures subject intel, normalizes form‑aligned data, links clues to tasks/subjects/locations, and exports official SAR & CAP forms.

# 2. Primary Functions

1. Clue Management – Create, categorize, geotag, score, and link clues; maintain lifecycle & audit.
2. Subject Intel – Tabbed subject profiles (identity, LKP/PLS & plans, clothing, experience, health, behaviors, contacts, events timeline).
3. Environmental Intel – Weather snapshots, hazards, terrain, area notes.
4. Form Center – Store all fields needed to export: SAR 134, SAR 135, SAR 301, SAR 307, SAR 306, SAR 305, SAR 132, CAPF 105, CAPF 106; inject mission header at export.
5. Intel Dissemination – Build sharable intel reports/briefs; push summaries to Operations/Planning, status boards, and comms.

# 3. Submodules / Tools

* Intel Dashboard – Overview of open clues, critical subjects, recent interviews, debrief flags.
* Clue Manager – Table + detail pane; linking to tasks, subjects, locations; attachments.
* Subject Editor (Tabbed) – Unified window for missing-person records and supplemental questionnaires.
* Environmental Panel – Weather/hazard snapshots tied to OP period.
* Form Center – Data entry/editors for each supported form; PDF export.
* Intel Reports – Composer to assemble printable/shareable intel packets.

# 4. Key Data Modules

Mission DB tables (see §7 for schemas):

* Core: intel\_clues, intel\_reports, intel\_env\_data.
* Subject set: intel\_subjects, intel\_subject\_events, intel\_subject\_plans, intel\_subject\_clothing, intel\_subject\_experience, intel\_subject\_health, intel\_subject\_behaviors, intel\_subject\_contacts.
* Form‑aligned: intel\_form\_sar134, intel\_form\_sar134\_entry, intel\_form\_sar135, intel\_form\_sar135\_segment\_prob, intel\_form\_sar301, intel\_form\_sar307, intel\_form\_sar306, intel\_form\_sar305, intel\_form\_sar132, intel\_form\_sar132\_entry, intel\_form\_capf105, intel\_form\_capf106, intel\_form\_capf106\_entry.

Shared export header (injected at print time from mission context): incident\_name, incident\_number, operational\_period, prepared\_by, prepared\_by\_id, date, time, page, page\_total.

# 5. UI Components

### 5.1 QML Windows (files under modules/intel/qml/)

* IntelDashboard.qml
* ClueListPanel.qml, ClueDetailDialog.qml
* SubjectEditorWindow.qml (tabbed)
* EnvIntelPanel.qml
* Form editors: FormSAR134.qml, FormSAR135.qml, FormSAR301.qml, FormSAR307.qml, FormSAR306.qml, FormSAR305.qml, FormSAR132.qml, FormCAPF105.qml, FormCAPF106.qml
* IntelReportComposer.qml

### 5.2 ASCII Wireframes

**Intel Dashboard**

+--------------------------------------------------------------+

| Intel Dashboard [Mission X | OP 3 | Role: Intel] |

+--------------------+----------------------+------------------+

| Open Clues (list) | Critical Subjects | Recent Interviews|

| # Time Type Loc | [⚠ Doe, J | LPB:W] | 2025-08-24 19:10 |

| 14 18:42 Track ...| [ Smith, A | ASD] | RP: Neighbor ... |

| ... | ... | ... |

+--------------------+----------------------+------------------+

| Env Snapshot: Weather, Hazards | Quick Actions: [New Clue] |

+--------------------------------------------------------------+

**Clue Detail Dialog**

+--------------------- Clue #014 -------------------------------+

| Type: Footprint Score: Likely Linked: [Task-32] [Subject-1] |

| Time: 18:42 Team: G-3 Entered By: P123 |

| Location: UTM 17T 345000 4655000 [Show on Map] |

| Description: ... |

| Attachments: [photo.jpg] [audio.m4a] |

| [Save] [Link...] [Export SAR 135] [Close] |

+---------------------------------------------------------------+

**Subject Editor (Tabbed)**

+-------------------- Subject: DOE, JANE (LPB: Wilderness) -------------------+

| DOB Sex Race Ht/Wt Eyes/Hair Nickname Photo [Upload] |

+----------------------------------------------------------------------------+

| [Plans/LKP] [Clothing] [Experience] [Health] [Behaviors] [Contacts] [Events]|

| Plans/LKP: |

| LKP Time: \_\_\_\_ Place: \_\_\_\_\_\_\_\_\_\_ [Pick Map] Intended Route: \_\_\_\_\_\_\_\_ |

| PLS: \_\_\_\_\_\_\_\_\_ Dest: \_\_\_\_\_\_\_\_\_ Expected Return: \_\_\_\_\_\_\_\_ |

+----------------------------------------------------------------------------+

| [Save] [Create Report] [Open Forms] |

+----------------------------------------------------------------------------+

**CAPF 106 – Interview Entry**

+------------------ CAPF 106 — Ground Interrogation ------------------+

| Interviewer: \_\_\_\_\_ Unit: \_\_\_\_ Date: \_\_\_\_ Time: \_\_\_\_ |

| Entries: [Add+] |

| ------------------------------------------------------------------ |

| Name | Contact | Address | Location | Confidence|

| John Smith | 555-0101 | 12 Pine Rd | Trailhead| High |

| Statement: "Saw a woman fitting desc at 18:30 heading N." |

| Follow-up Required: [x] Notes: \_\_\_\_\_\_\_\_ Links: [Clue][Task][Subj] |

| ------------------------------------------------------------------ |

| [Save] [Export PDF] [Close] |

+---------------------------------------------------------------------+

**SAR 134 – Clue Log**

+------------------------- SAR 134 — Clue Log -------------------------+

| # | Time | Resource | Location/UTM | Disposition | Link |

| 1 | 18:42 | G-3 | 17T 345000 4655000 | Held | [Clue#14]|

+---------------------------------------------------------------------+

# 6. API Endpoints

All endpoints use the active mission DB connection and write audit logs.

### 6.1 Clues

* GET /api/intel/clues — list/filter clues
* POST /api/intel/clues — create
* GET /api/intel/clues/{id} — retrieve (with links, attachments)
* PUT /api/intel/clues/{id} — update
* POST /api/intel/clues/{id}/link — link to subject/task/form
* POST /api/intel/clues/{id}/export/sar135 — generate SAR 135 PDF

### 6.2 Subjects

* GET /api/intel/subjects
* POST /api/intel/subjects
* GET /api/intel/subjects/{id} (expand tabs)
* PUT /api/intel/subjects/{id}
* POST /api/intel/subjects/{id}/event — append to timeline

### 6.3 Environmental

* GET /api/intel/env
* POST /api/intel/env

### 6.4 Forms

* GET /api/intel/forms/{type} — list instances (type ∈ sar134|sar135|sar301|sar307|sar306|sar305|sar132|capf105|capf106)
* POST /api/intel/forms/{type} — create instance
* GET /api/intel/forms/{type}/{id} — retrieve
* PUT /api/intel/forms/{type}/{id} — update
* POST /api/intel/forms/{type}/{id}/export — render PDF with injected header

### 6.5 Reports

* GET /api/intel/reports | POST /api/intel/reports | PUT /api/intel/reports/{id} | POST /api/intel/reports/{id}/export

# 7. Database Tables

Below are the key fields; actual DDL created via Alembic migrations.

### 7.1 Core

* **intel\_clues**: id, mission\_id, type, score, at\_time, geom(WKT), location\_text, entered\_by, team\_text, description, attachments\_json, linked\_subject\_id?, linked\_task\_id?, created\_at, updated\_at.
* **intel\_reports**: id, mission\_id, title, body\_md, audience, linked\_subject\_id?, linked\_task\_id?, created\_at.
* **intel\_env\_data**: id, mission\_id, op\_period, weather\_json, hazards\_json, terrain\_json, notes.

### 7.2 Subject Set (tabbed editor support)

* **intel\_subjects**: identity & LPB fields (name parts, sex, dob, race, height\_cm, weight\_kg, eyes, hair, build, lpb\_category\_code, lpb\_notes, photo\_url).
* **intel\_subject\_events**: subject\_id, at\_time, type, location\_text, geom, entered\_by, notes.
* **intel\_subject\_plans**: subject\_id, lkp\_time, lkp\_place\_text, lkp\_geom, pls\_text, pls\_geom, intended\_route, destination, expected\_return, transport\_json, vehicle\_json.
* **intel\_subject\_clothing**: subject\_id, snapshot\_time, headwear, outerwear, midlayer, base\_layer, hands, legs, feet, colors\_json, notes.
* **intel\_subject\_experience**: subject\_id, navigation, overnight, terrain, medical\_training, notes.
* **intel\_subject\_health**: subject\_id, conditions\_json, meds\_text, impairments\_json, physician\_contact.
* **intel\_subject\_behaviors**: subject\_id, habits\_json, triggers\_json, favorite\_locations\_text, critical\_attractions\_text, risk\_notes.
* **intel\_subject\_contacts**: subject\_id, name, relationship, phone, notes, notify\_on\_found.

### 7.3 Form‑Aligned Storage

**SAR 134**

* intel\_form\_sar134 (parent), intel\_form\_sar134\_entry (rows: clue\_number, time\_local, resource\_text, location\_text, utm\_text, clue\_disposition\_text, linked\_clue\_id?).

**SAR 135**

* intel\_form\_sar135: fields for clue details, urgency, probabilities; child intel\_form\_sar135\_segment\_prob (tier, segment\_code).

**SAR 301**

* intel\_form\_sar301: RP interview & subject questionnaire fields (see JSON groups in §4).

**SAR 307**

* intel\_form\_sar307: wanderer sheet (identity/desc, accessories, clothing, health, dementia Qs, habits, past incidents, links).

**SAR 306**

* intel\_form\_sar306: Blessed Dementia Scale (parts I–II) + derived totals/bands.

**SAR 305**

* intel\_form\_sar305: autism spectrum questionnaire sections.

**SAR 132**

* intel\_form\_sar132 (parent), intel\_form\_sar132\_entry (rows per address).

**CAPF 105**

* intel\_form\_capf105: radio message (standard or coded groups variant) with attachments and cross‑links.

**CAPF 106**

* intel\_form\_capf106 (parent), intel\_form\_capf106\_entry (rowed interview statements; links to clue/subject/task).

# 8. Inter-Module Connections

* **Operations**: Link clues/subjects to tasks; push summaries to task briefings and 214 logs.
* **Planning**: Feed intel into SITREPs and strategic tasks; debrief flags for review.
* **Communications**: Export CAPF 105; send alerts for high‑confidence leads; radio channel lookups for interviews if needed.
* **Forms & Documentation**: Export all supported forms with mission header injected.
* **Status Boards**: Publish high‑level intel cards (critical clue, last sighting, interview count).
* **Mapping/GIS**: All features with geom renderable on map layers.

# 9. Special Features / Notes

* **Role‑based UI & audit trail** on every write.
* **Offline‑first**; sync when LAN/WebSocket available.
* **Mission‑scoped storage** for forms; master lookups for personnel/vehicles.
* **Derived fields** (e.g., 306 score bands) recalculated on save.
* **PDF exports** maintain agency layout; header auto‑injected from mission context.
* **No evidence chain of custody** in this module (clue handling only, per project scope).
* **Shared sample data** at data/sample\_data.py for UI prototypes.

Module 8: Liaison

# 1. Module Name & Description

The Liaison Module manages relationships with assisting and cooperating agencies, external stakeholders, and mutual aid partners. It documents contacts, tracks agency-specific requests, and logs correspondence throughout the incident.

# 2. Primary Functions

* - Maintain contact list of supporting agencies and representatives
* - Log meetings, correspondence, and agreements
* - Track special considerations or operational caveats from partner agencies
* - Serve as intake for external resource offers, strategic goals, or external priorities

# 3. Submodules / Tools

* - Agency Directory: List of agencies, points of contact, and response roles with tags for coordination type
* - Interaction Log: Timestamped log of calls, emails, meetings, and briefings with agency reps
* - Request Tracker: Focused on capturing strategic goals and external priorities from outside organizations; requests are routed to appropriate modules for evaluation and tasking

# 4. Key Data Modules

* - Agency
* - id, name, type, jurisdiction, contact\_info[], notes
* - Interaction
* - id, agency\_id, time, type, summary, logged\_by
* - AgencyRequest
* - id, agency\_id, type, details, status, assigned\_to, resolution\_notes

# 5. UI Components

* - Agency contact browser with filtering
* - Interaction log entry form and timeline viewer
* - Request and offer queue dashboard

# 6. API Endpoints

* - GET /api/liaison/agencies
* - POST /api/liaison/agencies
* - GET /api/liaison/interactions
* - POST /api/liaison/interactions
* - GET /api/liaison/requests
* - POST /api/liaison/requests

# 7. Database Tables

* - agencies
* - agency\_interactions
* - agency\_requests

# 8. Inter-Module Connections

* - Planning Module: Linked through strategic task intake and agency goals
* - Operations Module: Offers and coordination affect tasking and resource movement
* - Personnel Module: Contact info tied to credentialed liaison staff or agency reps

# 9. Special Features / Notes

* - External agency offers can auto-generate planning tasks for evaluation
* - All logs are time-synced and audit-tracked
* - Designed to be usable in both real-world incidents and tabletop exercises

Module 9: Personnel and Role Management

# 1. Module Name & Description

This module centralizes personnel tracking, assignment management, credential verification, and ICS role allocation. It maintains personnel records, manages organizational hierarchy, and provides dynamic assignment tools to support field and command operations.

# 2. Primary Functions

* - Sync ICS role assignments with system access perincidents
* - Support hot-swap of assignments while preserving audit trail
* - Enable alerting when critical roles are unfilled or undermanned
* - Maintain personnel database with contact, credential, and status info
* - Assign personnel to ICS roles or custom positions
* - Display org charts with drag-and-drop role assignment
* - Track current duty status (on/off shift, deployed, available, unavailable)
* - Store training records and certifications
* - Allow bulk import of personnel and unit structure from CSV

# 3. Submodules / Tools

* - Hot-Swap Assignment Tool: Quickly reassign personnel while maintaining continuity and tracking changes
* - Role Alert Monitor: Flags when vital ICS roles are unstaffed or fall below minimum requirements
* - Personnel Roster: Searchable, filterable list of all personnel with status, rank, and assignment
* - Status Dashboard: At-a-glance view of all personnel availability and current assignment
* - Assignment Pool Filter: Tool to filter by qualification, availability, and org level for staffing tasks (shared with Planning module)
* - Import/Export Manager: Supports structured import and export of unit-level rosters

# 4. Key Data Modules

* - Person
* - id, name, callsign, rank, contact\_info, org\_unit\_id, photo\_url
* - Assignment
* - id, person\_id, role, start\_time, end\_time, location, status
* - Credential
* - id, person\_id, name, category, issue\_date, expiration, is\_trainer, is\_evaluator
* - OrgUnit
* - id, name, parent\_unit\_id, type, region

# 5. UI Components

* - Personnel profile editor
* - Interactive org chart with editable roles
* - Shift/duty tracker panel
* - CSV importer with mapping and preview

# 6. API Endpoints

* - GET /api/personnel
* - POST /api/personnel
* - GET /api/assignments
* - POST /api/assignments
* - GET /api/orgunits
* - POST /api/orgunits

# 7. Database Tables

* personnel
* Certificationtype
* Personnelcertification
* assignments
* org\_units

# 8. Inter-Module Connections

* - Operations Module: Assignment data drives tasking and field tracking
* - Medical Module: Injury/exposure reports are tied to personnel
* - Communications Module: Shows available contacts by role
* - Planning Module: Personnel pool filters used in IAP and strategic tasking

# 9. Special Features / Notes

* - Supports nested organizational structures with unlimited depth
* - Configurable role templates per incident type
* - Includes audit trail for role assignment changes
* - Supports training incidents with optional credential masking
* - Allows designation of units as inactive or virtual

Module 9-1: Personnel Certifications

## 1. Module Name & Description

The Certifications Module manages all professional credentials, trainings, and certifications for personnel within the ICS Command Assistant. It tracks certification levels and progression chains, and provides reporting for compliance and readiness.

## 2. Primary Functions

* Catalog Management: Define and maintain a library of certification types (e.g., ICS-100, First Aid, GTM1, GTM2) and their progression chains.
* Assignment & Tracking: Link certifications to personnel, including certification level (0–3) and associated documentation.
* Level Management: Support four discrete levels for each certification:
* 0: No Rating
* 1: Trainee
* 2: Qualified
* 3: Evaluator
* Reporting & Compliance: Generate reports on personnel certification levels and chain progression for audits and incident planning.
* When printing certifications to forms or status boards:
* Level 0: Not shown
* Level 1: <CODE>-T
* Level 2: <CODE>
* Level 3: <CODE>-SET

## 3. Submodules / Tools

* Certification Library: CRUD operations for certification definitions, including parent–child chain relationships.
* Personnel Certification: Interface to assign, view, and update a personnel’s certification level.
* Reporting Dashboard: Pre-built filters and export options (PDF/CSV) focused on certification levels and chain progression.

## 4. Key Data Modules

CertificationType:

{

id,

code,

name,

description,

category,

issuing\_organization,

parent\_certification\_id (nullable)

}

personnelCertification

{

id,

personnel\_id,

certification\_type\_id,

level (0–3),

attachment\_url

}

## 5. UI Components

* Certification Library View: Table listing certification types, showing chain relationships and allowing edits.
* Certification Detail Form: Modal to add or edit certification types and set chain links.
* Personnel Certification Dashboard: Grid showing each person’s certification levels with clear indicators:
* Blank: No Rating
* T: Trainee
* Q: Qualified
* E: Evaluator
* Certification Assignment Panel: Form within a personnel profile to set or update a certification level.
* Form & Status Board Rendering: When certifications are displayed outside the dashboard (e.g. printed forms, status boards):
* Level 0: Not shown
* Level 1: <CODE>-T
* Level 2: <CODE>
* Level 3: <CODE>-SET

## 6. API Endpoints

GET /api/certifications

POST /api/certifications

PUT /api/certifications/{id}

DELETE /api/certifications/{id}

GET /api/personnel/{pid}/certifications

POST /api/personnel/{pid}/certifications

DELETE /api/personnel/{pid}/certifications/{cid}

## 7. Database Tables

* certification\_types:
* id (PK)
* code
* name
* description
* category
* issuing\_organization
* parent\_certification\_id (FK, nullable)
* personnel\_certifications:
* id (PK)
* personnel\_id (FK)
* certification\_type\_id (FK)
* level (INT, 0-3)
* attachment\_url

## 8. Inter-Module Connections

* Personnel & Role Management: Display certification status next to roles and profiles to inform assignments.
* Operations & Planning: Filter or group personnel by certification level for tasking.
* Reference Library: Link to certification chain documentation and level definitions.

## 9. Special Features / Notes

* Bulk Import/Export: CSV support for certification types (including chains) and personnel levels.
* Role-Based Access Control: Admins manage certification definitions; supervisors can assign levels and view reports.

Module 10: Reference Library

# 1. Module Name & Description

This module serves as a centralized repository for doctrine, guides, agency policies, mutual aid agreements, jurisdictional maps, SOPs, and other reference materials. It allows teams to quickly locate, search, and share incident-critical documents from within the system.

# 2. Primary Functions

* - Upload, categorize, and tag reference documents
* - Store internal SOPs, ICS references, job aids, and training handouts
* - Enable full-text search and filtering by topic or agency
* - Organize documents by custom collections or folders
* - Provide access controls per document or collection

# 3. Submodules / Tools

* - Document Uploader: Drag-and-drop or browse to upload documents with category, keywords, and perincidents
* - Reference Browser: Interface to search, filter, preview, and download documents
* - Collections Manager: Create and maintain sets of related documents by topic, agency, or function
* - External Link Register: Store and access bookmarked online resources or interagency sites

# 4. Key Data Modules

* - Document
* - id, title, filename, type, uploaded\_by, tags[], access\_level, agency, category, created\_at
* - Collection
* - id, name, description, document\_ids[], created\_by
* - ExternalLink
* - id, title, url, description, added\_by

# 5. UI Components

* - Document upload and tagging form
* - Searchable and filterable document list
* - Collection view with folder-style UI
* - Quick preview modal or inline reader

# 6. API Endpoints

* - GET /api/library/documents
* - POST /api/library/documents
* - GET /api/library/collections
* - POST /api/library/collections
* - GET /api/library/links
* - POST /api/library/links

# 7. Database Tables

* - documents
* - collections
* - external\_links

# 8. Inter-Module Connections

* - Forms Module: May cross-reference job aids or SOPs related to specific ICS forms
* - Planning Module: Reference materials can be embedded in planning packets
* - Liaison Module: External agreements and policies stored for agency access
* - Training Mode / Sandbox: Pulls content from this library to guide simulations

# 9. Special Features / Notes

* - Supports upload of PDFs, DOCX, images, and video content
* - Full-text document indexing for fast searching
* - Optional annotation system to allow shared highlights/notes
* - Version control and document history supported
* - Offline access for pre-synced document bundles

Module 11: ICS Forms and Documentation

# 1. Module Name & Description

This module provides a centralized location to view, fill, manage, and archive ICS forms and custom documentation. It supports digital form filling, version control, cross-module data integration, and export to PDF/print formats. The system also allows uploading custom agency forms and automatically converting them into fillable digital templates.

# 2. Primary Functions

* - Browse and search standardized ICS forms (U.S., Canadian, Coast Guard, and other designated formats)
* - Upload custom forms and convert them into digital templates
* - Auto-fill fields from related modules (e.g., personnel, planning, comms)
* - Allow real-time collaborative editing of forms
* - Track form history and versions with audit trail
* - Export to PDF, print, or attach to IAP packet
* - Group similar form types to allow users to choose preferred print/output format (e.g., communications log to ICS 309 or CAPF 110)

# 3. Submodules / Tools

* - Form Type Selector: Filter forms by system (U.S. ICS, Canadian IMS, Coast Guard, etc.)
* - ICS Form Library: Catalog of standardized forms with metadata and instructions
* - Form Builder: Used for converting uploaded forms into fillable digital templates
* - Form Viewer/Editor: Interface for filling out, saving, and reviewing forms
* - Version & Audit Log Viewer: Displays edit history and change tracking
* - Related Forms Linker: Allows cross-linking of related documents (e.g., ICS 213 linked to a task from ICS 204)

# 4. Key Data Modules

* - FormTemplate
* - id, name, type, source\_file, fields[], version, created\_by
* - FormInstance
* - id, template\_id, filled\_by, filled\_data, status, revision\_number, timestamps
* - FormVersion
* - id, instance\_id, editor\_id, timestamp, changeset, comments
* - CustomUpload
* - id, file\_path, uploader, field\_map, linked\_template

# 5. UI Components

* - Form browser with filtering by type or module
* - Digital form editor with auto-fill and validation
* - Version history side panel
* - Upload wizard for custom form intake
* - Cross-reference sidebar for linked forms

# 6. API Endpoints

* - GET /api/forms/templates
* - POST /api/forms/templates
* - GET /api/forms/instances
* - POST /api/forms/instances
* - GET /api/forms/versions
* - POST /api/forms/upload

# 7. Database Tables

* - form\_templates
* - form\_instances
* - form\_versions
* - custom\_uploads

# 8. Inter-Module Connections

* - Planning Module: Pulls forms into IAP packets; receives auto-fill from strategy/task data
* - Personnel Module: Provides names, IDs, roles for form population
* - Operations Module: ICS 204, 215, 214 tie-in for tasking and logging
* - Communications Module: Populates ICS 205, 210, 213
* - Medical Module: Links to ICS 206, injury logs
* - Liaison Module: Agency contacts and restrictions on forms like ICS 213

# 9. Special Features / Notes

* - Every form field supports audit logging, auto-fill override, and manual locking
* - Digitized templates retain formatting fidelity for print and export
* - Form builder uses drag-and-drop interface for field mapping
* - Supports offline form filling with sync upon reconnect
* - All forms tied to incident context for sorting and archiving

Module 12: Finance/Admin

# 1. Module Name & Description

This optional module supports incident-related financial tracking, administrative documentation, cost recovery, resource timekeeping, and expense forecasting. It can be enabled or disabled per incident and is structured to align with ICS finance unit responsibilities, with optional CAP or agency-specific workflows.

# 2. Primary Functions

* - Record expenses and assign them to categories and funding sources
* - Track time for personnel, equipment, and resources
* - Generate cost summaries for operations, logistics, etc.
* - Attach receipts, approvals, and justification documents
* - Export financial summaries and logs to PDF or Excel
* - Manage reimbursements, invoices, or funding requests

# 3. Submodules / Tools

* - Time Tracking Register: Logs person/equipment hours
* - Expense Entry Tool: Input field expenditures, equipment rentals, fuel, meals, etc.
* - Funding Source Manager: Assign costs to budgets, grants, or reimbursement tracks
* - Cost Summary Dashboard: Visual reporting of total and projected costs
* - Document Attachments Panel: Upload receipts or authorizations
* - ICS Form Generator: Supports ICS 214 (activity log), 213RR, 219, 211, and CAPF financial forms

# 4. Key Data Modules

* - Expense
* - id, description, category, amount, date, resource\_id, approver, funding\_source\_id
* - TimeEntry
* - id, person\_id, hours, task\_id, date, role, notes
* - FundingSource
* - id, name, code, type, balance, agency

# 5. UI Components

* - Financial entry forms with auto-categorization
* - Tables for time logs and expense history
* - Cost summary report cards with export options
* - Form print panel (ICS/CAP/Agency-specific)

# 6. API Endpoints

* - GET /api/finance/expenses
* - POST /api/finance/expenses
* - GET /api/finance/time
* - POST /api/finance/time
* - GET /api/finance/funding
* - POST /api/finance/funding

# 7. Database Tables

* - expenses
* - time\_entries
* - funding\_sources

# 8. Inter-Module Connections

* - Personnel Module: Ties to pay/time records
* - Logistics Module: Tracks supply/resource spending
* - Operations Module: Relates costs to specific tasks/incidents
* - Forms Module: Supports auto-filling of cost and time forms
* - Command Module: Optional view-only dashboard for leadership

# 9. Special Features / Notes

* - Supports custom cost categories per agency
* - CAP-specific forms supported where applicable (CAPF series)
* - Digital signature for cost approval routing
* - Role-based finance access
* - Printable financial reports and audit logs

Module 13: Status Boards

# 1. Module Name & Description

This module provides dynamic, incident-specific status boards for real-time situational awareness across all ICS sections. Status boards are configurable and tailored to specific needs such as task status, personnel availability, communications readiness, logistics supply levels, and more. This module does not generate or store new data. Instead, it presents information produced and maintained by other modules in a visual format.

# 2. Primary Functions

* - Choose from predefined status board templates for common use cases (e.g., Air Ops, Task Tracking, Staging, Logistics)
* - Create and manage multiple custom status boards
* - Display real-time updates from other modules (tasks, teams, resources, forms)
* - Support manual and automated entry updates
* - Enable filtered views by section, unit, or incident type
* - Visual indicators for priority, readiness, and changes

# 3. Submodules / Tools

* - Board Manager: Create, edit, archive, or clone boards
* - Entry Panel: Add/update status entries with color, icons, and comments
* - Board Display Grid: Interactive, tile-based real-time board UI
* - Board Templates: Save reusable board layouts (e.g., Air Ops, SAR Tasking, Logistics Inventory)
* - Filter/Sort Tool: Show by section, date, priority, or custom tag

# 4. Key Data Modules

* - StatusBoard
* - id, name, type, section, created\_by, visibility\_scope, template\_id
* - BoardEntry
* - id, board\_id, title, description, icon, color, last\_updated\_by, timestamp, linked\_item (task\_id, team\_id, etc.)

# 5. UI Components

* - Dashboard view for multiple boards
* - Board entry cards with drag-and-drop reordering
* - Quick filters for time, section, keyword
* - Board creation and layout editor

# 6. API Endpoints

* - GET /api/status\_boards
* - POST /api/status\_boards
* - GET /api/status\_entries
* - POST /api/status\_entries
* - PUT /api/status\_entries/{id}

# 7. Database Tables

* - status\_boards
* - status\_entries

# 8. Inter-Module Connections

* - Operations Module: Link entries to tasking and 214 logs
* - Personnel Module: Show status of individuals or teams
* - Logistics Module: Track inventory and requests
* - Communications Module: Show radio status, frequency availability, etc.
* - Command Module: Present board views to leadership as dashboards

# 9. Special Features / Notes

* - Status change notifications: Send alerts (in-app, email, or SMS) for critical updates or flagged board entries
* - Template-specific color and icon sets: Maintain consistent visual language across boards (e.g., Air Ops, Logistics, Comms)
* - Offline board mode: Boards can be cached for disconnected operation and synced when reconnected
* - Board templates for standardized incidents (e.g., Search & Rescue, Public Event)
* - Reusable across incidents
* - Display on dedicated monitors or screens for command posts
* - Export snapshots or history logs to PDF or image
* - Optional refresh rates or live sync settings for LAN deployments

Module 14: Public Information

# 1. Module Name & Description

This module enables authorized personnel (e.g., PIOs) to create, manage, and publish information for internal staff, partner agencies, and the public. It supports message review workflows, media coordination, and internal press log tracking.

# 2. Primary Functions

* - Draft and review press releases, situation updates, and advisories
* - Manage a log of all outgoing public communications
* - Enable approvals and revision tracking for all messages
* - Provide exportable briefings and public update summaries

# 3. Submodules / Tools

* - Message Composer: Draft, edit, preview messages
* - Approval Workflow: Assign reviewers/approvers with status tracking
* - Media Log: Record of posted or distributed content
* - Press Summary Builder: Compile briefings for agency/public release

# 4. Key Data Modules

* - Message
* - id, title, body, type, audience, created\_by, approved\_by, timestamp, status

# 5. UI Components

* - Message queue with status badges
* - Inline message editor and formatting panel
* - Revision history viewer
* - Publish/export toolbar

# 6. API Endpoints

* - GET /api/public\_info/messages
* - POST /api/public\_info/messages

# 7. Database Tables

* - public\_info\_messages

# 8. Inter-Module Connections

* - Command Module: Coordinate approved updates or briefings
* - Communications Module: Broadcasts internal PIO alerts or notifications
* - Planning Module: Sync with ongoing situation reports or ICS 209 summaries
* - Forms Module: Support ICS 213 and ICS 209 formatting where applicable

# 9. Special Features / Notes

* - Message tagging by audience: Public, Agency, Internal
* - Role-based publishing controls (e.g., only lead PIOs can finalize)
* - Public briefing export to PDF, print, or email
* - Offline message drafting support with queued sync

Module 15: Mobile App Integration

# 1. Module Name & Description

This module enables seamless interaction between the desktop ICS Command Assistant and a separately designed mobile application. This module serves as the bridge for real-time data syncing and workflow continuity between the two systems. It provides real-time data syncing, tailored mobile interfaces, and role-based mobile access for field personnel.

# 2. Primary Functions

* - Sync incident data between Windows and mobile apps
* - Support offline mobile use with automatic re-sync on reconnect
* - Display mobile-tailored views of tasks, assignments, and messages
* - Allow mobile personnel to submit ICS 214 entries and task updates
* - Enable mobile alerts, tasking, and status notifications

# 3. Submodules / Tools

* - Sync Engine: Ensures secure bidirectional data exchange
* - Mobile UI Configurator: Define what data is accessible to different user roles in the mobile app
* - Mobile Perincidents Panel: Grant/limit access based on assignment or role
* - Mobile Activity Feed: Show incoming tasks, messages, or alerts

# 4. Key Data Modules

* - MobileSession
* - id, user\_id, device\_id, session\_token, last\_sync\_time
* - MobileAccessRule
* - id, role\_id, module\_access, view\_config

# 5. UI Components (Windows-Side)

* - Sync settings page (schedule, conflict resolution)
* - Device management panel
* - Role-based visibility matrix

# 6. API Endpoints

* - GET /api/mobile/sync
* - POST /api/mobile/push
* - POST /api/mobile/pull
* - GET /api/mobile/perincidents

# 7. Database Tables

* - mobile\_sessions
* - mobile\_access\_rules

# 8. Inter-Module Connections

* - Operations Module: Pulls tasking data and logs 214 entries
* - Personnel Module: Grants mobile access per person/role
* - Communications Module: Pushes alerts, receives mobile messages
* - Forms Module: Enables 214 creation from mobile

# 9. Special Features / Notes

* - Field entries marked as mobile-sourced in audit logs
* - Optional incident QR code or link-based mobile app onboarding
* - Configurable field perincidents and lockdowns
* - Offline-first design for austere environments

Module 16: Training/Sandbox Mode

# 1. Module Name & Description

# 2. Primary Functions

# 3. Submodules / Tools

# 4. Key Data Modules

# 5. UI Components

# 6. API Endpoints

# 7. Database Tables

# 8. Inter-Module Connections

# 9. Special Features / Notes

Module 17: Search and Rescue Toolkit

# 1. Module Name & Description

A specialized suite of planning tools designed to support Search and Rescue (SAR) operations. This module provides structured utilities and calculations to assist planning personnel in defining search areas, understanding behavioral profiles, and coordinating data relevant to locating missing persons.

# 2. Primary Functions

* Provide additional SAR specific tools and modules to assist in planning search and rescue tasks
* Provide preconfigured task fields like LKP, PLS, and POD
* Simplify UI and data flows for SAR planning incidents

# 3. Submodules / Tools

1. Missing Person Toolkit
2. Lost Person Behavior Calculator
3. POD Calculator

# 4. Key Data Modules

# 5. UI Components

# 6. API Endpoints

# 7. Database Tables

# 8. Inter-Module Connections

# 9. Special Features / Notes

Module 18: Disaster Response Toolkit

# 1. Module Name & Description

# 2. Primary Functions

# 3. Submodules / Tools

# 4. Key Data Modules

# 5. UI Components

# 6. API Endpoints

# 7. Database Tables

# 8. Inter-Module Connections

# 9. Special Features / Notes

Module 19: Planned Event Toolkit

# 1. Module Name & Description

Planned Events Toolkit: A suite of standalone modules, activated only for planned incidents (e.g., festivals, parades, marathons). Provides event-specific planning, public safety, and streamlined tasking—independent of core ICS features.

# 2. Primary Functions

* Loads dynamically when incident type is planned
* Unloads automatically upon planned event demobilization

# 3. Submodules / Tools

1. **Event Promotion and Communication**
   1. Purpose: Engage attendees and stakeholders with timely, targeted messaging.
   2. Key Features:
      1. Multi channel campaign builder (email, SMS, push, social).
      2. Scheduled & trigger based sends (e.g., event countdown, weather alerts).
      3. Geo fenced notifications for location specific updates.
      4. Subscriber management: opt in/out, segmentation by role or interest.
      5. Analytics: delivery rates, open/click metrics.
   3. UI Components:
      1. Campaign Dashboard (list, status, next send time, basic stats).
      2. Message Composer (WYSIWYG with template gallery).
      3. Geo Map Selector (draw and save zones for targeting).
      4. Subscriber List (filter by tags, engagement).
2. **Vendor & Permitting Coordination**
   1. Purpose: Onboard vendors and manage permit compliance efficiently.
   2. Key Features:
      1. Vendor Registry: profiles, document uploads (insurance, licenses).
      2. Status workflow: Pending → Approved → Active → Suspended → Revoked.
      3. Map based booth assignment.
      4. Permit issuance via customizable templates.
      5. Bulk import/export and compliance reporting.
   3. UI Components:
      1. Vendor Table (searchable, sortable, status badges).
      2. Profile Detail Form (Contact / Documents / Location / History tabs).
      3. Permit Dashboard (tiles for Active, Expiring Soon, Expired).
      4. Bulk Import Wizard (CSV upload, validation, field mapping).
3. **Public Safety & Incident Management**
   1. Purpose: Enable on-site teams to monitor safety, report incidents, and dispatch responders.
   2. Key Features:
      1. Security Patrol Manager: zone definitions, real time coverage map.
      2. Mobile Incident Logger: report events with photos/videos.
      3. Automated Dispatch: route reports, track acknowledgments.
      4. Incident categorization (medical, security, lost child, etc.).
      5. Escalation and on call rotation rules.
   3. UI Components:
      1. Patrol Map (live locations, zone overlays).
      2. Incident Feed (filter by type, status).
      3. Dispatch Board (Unassigned / En Route /Arrival / Complete).
      4. Report Form (quick entry, media attach).
4. **Mini-Tasking Module**
   1. Purpose: Provide a lightweight tasking interface for minor event duties
   2. Key Features:
      1. Quick Task Creator: title, assignee, due time, status.
      2. Nested checklists for multi-step tasks.
      3. Recurring templates (e.g., hourly safety checks).
      4. Reminders & notifications for due/overdue tasks.
      5. Filtering by assignee, status, priority.
   3. UI Components:
      1. Quick Task Sidebar (create/view tasks, progress bars).
      2. Checklist Popup (view/edit sub items).
      3. Task Calendar (due dates, recurring schedules).
      4. Notification Center (task alerts).
5. **Public Health & Sanitation Oversight**
   1. Purpose: Maintain hygiene and health standards throughout the event.
   2. Key Features:
      1. Health Inspection Forms: vendor booths, facilities.
      2. Sanitation Issue Tracker (waste overflow, supply shortages).
      3. Crew Assignment & follow up verification.
      4. Instant alerts for critical violations.
   3. UI Components:
      1. Inspection Form (customizable checklist, photo capture).
      2. Issue Dashboard (map + table of sanitation reports).
      3. Crew Scheduler (drag and drop assignments).
      4. Follow Up Tracker (resolution status, timestamps).

# 4. Key Data Modules

* PromotionCampaign: id, name, channels[], schedule, content
* Vendor: id, name, type, contactInfo, status, permitIds[]
* Permit: id, vendorId, type, issueDate, expiryDate
* SecurityPatrol: id, zoneId, personnelIds[], startTime, endTime
* IncidentReport: id, type, description, location, timestamp, reporterId
* DispatchRequest: id, incidentReportId, responderIds[], status
* QuickTask: id, title, assignedTo, dueTime, status
* TaskChecklistItem: id, taskId, description, completed
* Reminder: id, taskId, remindAt, sent
* InspectionRecord: id, inspectionType, targetId, findings, date

# 5. UI Components

* Toolkit Launcher (side‑panel menu for all planned modules)
* Dashboard Widgets (campaign summary, vendor status, task overview, incident feed)
* Notification Center (alerts across modules)

# 6. API Endpoints

* Promotions: GET /planned/promotions/campaigns, POST /planned/promotions/campaigns, POST /planned/promotions/campaigns/:id/send, GET /planned/promotions/metrics/:id
* Vendors & Permits: GET /planned/vendors, POST /planned/vendors, PUT /planned/vendors/:id, DELETE /planned/vendors/:id, GET /planned/permits, POST /planned/permits
* Public Safety: GET /planned/security/patrols, POST /planned/security/patrols, GET /planned/incidents, POST /planned/incidents, POST /planned/dispatches, PUT /planned/dispatches/:id
* Tasks: GET /planned/tasks, POST /planned/tasks, GET /planned/tasks/:id/checklist, POST /planned/tasks/:id/checklist, POST /planned/tasks/:id/reminders
* Health & Sanitation: POST /planned/inspections/health, GET /planned/sanitation/issues, POST /planned/sanitation/issues

# 7. Database Tables

* planned\_promotions
* planned\_vendors
* planned\_permits
* planned\_security\_patrols
* planned\_incident\_reports
* planned\_dispatch\_requests
* planned\_tasks
* planned\_task\_checklist\_items
* planned\_task\_reminders
* planned\_inspection\_records

# 8. Inter-Module Connections

* Public alerts fed to Communications module
* Vendor/permit statuses shared with Logistics & Personnel
* Task assignments synced with Personnel & Role Management
* Inspection data exported via Forms & Documentation (ICS 219)

# 9. Special Features / Notes

* Fully independent: no overlap with core ICS modules
* Real‑time updates via WebSocket for critical data
* Role‑based perincidents via planned\_event\_roles
* Offline support with automatic sync on reconnect

Module 19-1: Event Promotions and Communications

## 1. Module Name & Description

## 2. Primary Functions

## 3. Submodules / Tools

## 4. Key Data Modules

## 5. UI Components

## 6. API Endpoints

## 7. Database Tables

## 8. Inter-Module Connections

## 9. Special Features / Notes

Module 19-2: Vendor & Permitting Coordination

## 1. Module Name & Description

## 2. Primary Functions

## 3. Submodules / Tools

## 4. Key Data Modules

## 5. UI Components

## 6. API Endpoints

## 7. Database Tables

## 8. Inter-Module Connections

## 9. Special Features / Notes

Module 19-3: Public Safety & Incident Management

## 1. Module Name & Description

## 2. Primary Functions

## 3. Submodules / Tools

## 4. Key Data Modules

## 5. UI Components

## 6. API Endpoints

## 7. Database Tables

## 8. Inter-Module Connections

## 9. Special Features / Notes

Module 19-4: Mini Tasking Module

## 1. Module Name & Description

## 2. Primary Functions

## 3. Submodules / Tools

## 4. Key Data Modules

## 5. UI Components

## 6. API Endpoints

## 7. Database Tables

## 8. Inter-Module Connections

## 9. Special Features / Notes

Module 19-5: Public Health & Sanitation Oversight

## 1. Module Name & Description

## 2. Primary Functions

## 3. Submodules / Tools

## 4. Key Data Modules

## 5. UI Components

## 6. API Endpoints

## 7. Database Tables

## 8. Inter-Module Connections

## 9. Special Features / Notes

Module 20: Initial Response Toolkit

# 1. Module Name & Description

# 2. Primary Functions

# 3. Submodules / Tools

# 4. Key Data Modules

# 5. UI Components

# 6. API Endpoints

# 7. Database Tables

# 8. Inter-Module Connections

# 9. Special Features / Notes

Module 21: UI Customization

# 1. Module Name & Description

# 2. Primary Functions

# 3. Submodules / Tools

# 4. Key Data Modules

# 5. UI Components

# 6. API Endpoints

# 7. Database Tables

# 8. Inter-Module Connections

# 9. Special Features / Notes

Module XX: AI Integration (Wishlist Phase)

# 1. Module Name & Description

This forward-looking module explores integration of artificial intelligence to enhance decision support, situational analysis, and workload reduction throughout the ICS Command Assistant ecosystem. This phase remains conceptual until prioritized for development.

# 2. Potential Functions

* - Automated analysis of 214 logs to detect emerging trends, gaps, or coordination issues
* - Task prediction and resource matching based on past data and current conditions
* - Natural language queries across logs, forms, tasks, and messages
* - Smart form autofill based on prior entries and contextual incident data
* - AI-generated briefings or situation summaries
* - Optical character recognition (OCR) and NLP for digitizing paper forms
* - Incident risk forecasting based on operational plans, weather, and historical performance
* - Auto-sort and prioritize task queues based on urgency and context
* - Collaboration summarizer for chats, logs, and planning threads
* - Smart alert filter to reduce information overload
* - Post-incident summary generator using incident logs and task data

# 3. Submodules / Tools (Conceptual)

* - AI Assistant Panel: Query interface and context-aware suggestions
* - Form AutoComplete Engine: Fills and recommends form fields based on context
* - Pattern Watcher: Monitors logs and tasks for trends, gaps, or repeated issues
* - OCR Uploader: Import and convert handwritten or scanned documents

# 4. Required Foundations

* - Expanded metadata tagging across all modules
* - Consistent use of audit logs and timestamps
* - NLP libraries and OCR tools
* - Opt-in privacy controls and usage tracking

# 5. UI Concepts

* - Inline smart hints while filling forms or reviewing tasks
* - AI chat sidebar with search and recommendation abilities
* - Visual indicators for detected anomalies or suggestions

# 6. Dependencies

* - Integration with incident database and form digitization systems
* - Coordination with data governance and audit compliance

# 7. Special Notes

* - Not prioritized for early-phase implementation
* - Will require regulatory review for sensitive data
* - Designed for opt-in usage with explicit controls
* - Intended to support, not replace, human decision-making