Architectural Document

Document ID: AD-1952
Title: Clinical Trial Packaging and Labeling Execution Application
Author: Avirup Saha
Version: 1.0
Date: June 14, 2025

---

1. Overview

This architectural document outlines the end-to-end design, component structure, and interaction flow of a web-based application developed to support **packaging and labeling activities** in a **clinical trial environment**. The application ensures accuracy, traceability, regulatory alignment, and operational control across job execution, room allocation, equipment handling, checklist compliance, reconciliation, and documentation.

---

2. Purpose

The purpose of this application is to facilitate digital execution of packaging processes for clinical trials. This includes:

- Operator login and audit trail
- Job and room allocation workflows
- Equipment registration with validation
- SOP referencing and checklist completion
- Controlled barcode-based packaging
- Reconciliation with deviation capture
- Final job completion control

---

3. Technology Stack

Layer	Technology
Frontend	React.js (with React Router, Hooks)

Layer	Technology
Styling	CSS (modular with custom components)
Backend (Data Source)	Static JSON files (simulating DB/API)
State Management	React Context API + localStorage
Notifications	react-toastify
Deployment Target	Browser-based (can be containerized later)

---

4. Component Breakdown

4.1 Operator Login

- **Function:** Authenticates operator via scan code and password.
- **Storage:** Saves logged-in operator in localStorage.
- **Features:**
    - Restriction on re-login if someone is already active.
    - Visible card with logout capability.

4.2 Job Allocation

- **Function:** Allocates a job from a predefined list (Job.json).
- **Validation:** Requires operator login before proceeding.
- **Outcome:** Saves allocated job and printer type in localStorage.

4.3 Room Allocation

- **Function:** Assigns physical room to packaging operation (Room.json).
- **Validation:** Operator credentials required.
- **Outcome:** Saves allocated room in localStorage.

4.4 Equipment Registration

- **Function:** Registers necessary equipment from Equipment.json.
- **Rules:**
    - Only one of each category (Printer, Workstation, Workcenter, Manual Pack).
    - Printer type must match the job's expected printertype.
- **Storage:** Equipment stored in localStorage and displayed in summary.

4.5 SOP Viewer

- **Function:** Displays standard operating procedures as downloadable documents.
- **Navigation:** Routes to /sopdocument and /sopdocumentcredit.

4.6 Checklist Completion

- **Function:** Displays checklist items from Checklists.json.
- **Validation:** All items must be marked "Yes" and confirmed.
- **State:** Stores completion flag in localStorage.

4.7 Packaging Process

- **Function:** Allows barcode-based packaging as per job orderqty.
- **Validation:** Requires job, room, and equipment to be confirmed.
- **Tracking:** Tracks progress against order quantity.

4.8 Reconciliation

- **Function:** Validates input/output counts.
- **Deviation Handling:** If below 100%, requires IPC reason and supervisor validation.
- **Logic:** Applies business formula for reconciliation calculation.

4.9 Other Info Panel

- **Function:** Displays snapshot of current job, room, printer, and statuses.
- **Live Refresh:** Includes refresh button to rehydrate data from JSON/localStorage.

---

5. Data Sources

File	Purpose
Operator.json	User authentication
Job.json	Job allocation and printertype info
Room.json	Physical room assignment
Equipment.json	Equipment master data
Checklists.json	Prerequisite checks
Barcode.json (planned)	Product-level packaging barcodes

All data is loaded as static JSON files and cached in browser localStorage.

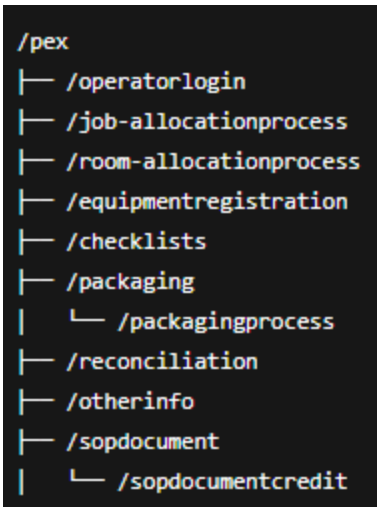
---

6. State Management

State	Storage Method	Scope
Logged-in Operator	localStorage (loggedOperator)	Global
Job & Order Info	Context (useJob) + localStorage	Global
Room Info	Context (useRoom) + localStorage	Global
Registered Equipment	localStorage	Shared
Checklist Completion	localStorage (checklistCompleted)	One-time
Packaging Progress	localStorage (progressSession)	Resettable

---

7. Navigation Flow



Flow: **Operator → SOP**(If required)**→ Job Allocation → Room Allocation → Clean Room**(If required) **→ Equipment → Checklists→ Manage SU → Label SU → Open SU → Packaging → Close SU → Reconciliation → End Job → Shutdown/ Restart Application**

---

8. Security & Validation

- Operator access is validated by scan code and password.
- Supervisor validation required for deviations.
- Equipment types are strictly validated against job requirements.
- No backend or sensitive data handled—safe for internal deployments.

---

9. Future Enhancements

- Migrate JSON files to a backend API with DB integration.
- Implement JWT-based auth for secure login sessions.
- Add audit trails and logging for all actions.
- Export job completion reports in PDF/Excel.
- Multi-role support (Operator, Supervisor, Auditor).

---

10. Contributors

- **Avirup Saha** – Lead Developer, Architect, UX Refinement & Functional Flow Design