

Draft IT Specification (Based on Shipment Portal Flow)

1. Project Overview

- **Purpose:** To build / update a shipment portal web application where users can log in, manage shipments, view status, etc.
 - **Objectives:**
 - Provide **role-based access** (Admin & Assistant, Accounts & Assistant, Clearance Manager & Assistant)
 - Allow users to **create, update, and monitor** shipment records
 - Maintain a **dashboard** for high-level shipment insights
 - Enforce **data validation, audit trail, and secure access**
-

2. User Roles & Permissions

Role	Description	Permissions / Access	Key Features
Admin & Assistant	Portal super-admin	Full CRUD on shipments, manage users, view audit logs	Create users, assign roles, delete shipments, escalate issues
Accounts & Assistant	Approver for shipments	Review, approve, reject or request changes on shipments & Payments updates	Approve shipments, reject shipments with reason, request changes, Payments updates and send notifications
Clearance Manager & Assistant	Operational	Create and update shipments, but limited admin-level controls	Enter shipment data, upload documents, view shipment list

Login: Each role should have separate login credentials. On login, the UI should detect the role and load the correct dashboard + menu options.

3. User Flows / Journeys

a) Admin Login Flow

1. Admin navigates to the portal login page.
2. Admin enters username + password.
3. On successful login → Admin Dashboard.
4. Dashboard shows summary: total shipments, pending shipments, user management section, reports, and logs.
5. Admin can go to **User Management** to add / remove / edit users.

b) Clearance Agent — Create a Shipment

1. Clearance Agent logs in → Agent Dashboard.
2. Clicks “Create Shipment”.
3. Shipment form opens.
 - Fields: Shipment ID (auto-generated), Exporter Details, vendor Details, Item Description, Weight, Dimensions, Value, Pickup Date, Expected Delivery Date, Mode of Transport, Status (Default = “New”), Documents (invoice, packing list for all jobs (Based on sea and air shipment have to attach BL (sea) or AWB (Air shipment)).(refer old PDF sheet)
4. Agent fills in details and uploads required documents.
5. Validation:
 - **Mandatory:** Exporter, Customer, Weight, Value, Pickup Date
 - **Format check:** Date fields must follow DD-MM-YYYY, Weight must be numeric, Value must be numeric
 - **File uploads:** Only PDF, JPEG allowed; file size limit = 10 MB
6. Agent clicks **Submit**.
7. On submit → Save shipment record → status changes to “Created” → Redirect to “Shipment Details” view page.
8. Email notification (to Manager / Admin) about new shipment (if required).

c) Accounts Manager — Approve Shipment

1. Manager logs in → Manager Dashboard.
2. Sees a list/table of shipments in “Created” status, awaiting approval.
3. Clicking a shipment row → opens Shipment Detail page.
4. On Detail page:
 - View all fields + documents.
 - Buttons: **Approve, Reject, Request Changes.**
5. If **Approve**: status changes to “Approved” → update record, send email to Agent / Admin.
6. If **Reject**: show a modal / field to ask for “Reason for Rejection” → Manager enters reason → status changes to “Rejected” → email to Agent with reason.
7. If **Request Changes**: open the shipment form in editable mode (or show fields to be changed) → Manager changes fields → “Request Changes” (or “Send back”) → status = “Changes Requested” → Agent gets notification.

d) Dashboard / Reporting

- Each role (based on permission) sees a Dashboard:
 - **Admin**: All shipments, by status, by time period, user-wise shipments.
 - **Manager**: Pending approvals, shipments approved, shipments rejected.
 - **Agent**: My shipments, status, shipments delivered / pending.
 - Reports: Export to Excel / PDF.
-

4. Functional Requirements (Screen-wise)

Here is a breakdown of important screens/modules, with field-level and action-level details:

Login Page

- **Fields:**
 - Username (text)

- Password (password)
- (Optional) 2FA Code (if implemented)
- **Buttons / Actions:**
 - **Login** → Validate credentials → Role-based redirect
 - **Forgot Password** → Opens “Reset Password” flow
- **Validation:**
 - Username: required
 - Password: required
 - If 2FA: code required

Dashboard (per Role)

- **Widgets / Panels:**
 - Summary cards: Total Shipments, Pending Approval, Rejected, Delivered
 - Recent Shipments Table (show last 5–10)
 - Action Buttons (Create Shipment for Agents; Manage Users for Admin, etc.)
- **Interactions:**
 - Clicking on any summary card or shipment opens detailed view
 - Filtering by date / status

Create Shipment Form

- **Fields:**
 - Shipment ID (auto)
 - Sender Name (text)
 - Sender Address (multiline)
 - Receiver Name (text)
 - Receiver Address (multiline)
 - Item Description (text / textarea)
 - Weight (number / decimal)

- Dimensions (L × W × H) (decimal)
- Value (currency)
- Pickup Date (date picker)
- Expected Delivery Date (date picker)
- Mode of Transport (dropdown: Air / Sea / Road)
- Status (hidden / auto = “New” or “Created”)
- Documents Upload (multiple files)
- **Buttons:**
 - **Submit** → Validate + Save
 - **Cancel** → Clear form / go back to Dashboard
- **Validation rules:**
 - Required fields (as above)
 - Numeric validation for weight, dimensions, value
 - Date validation: Pickup Date ≤ Expected Delivery Date
 - File upload limits + type check

Shipment Detail Page

- **Display Fields:** All the data from the form (read-only unless role is Manager and doing “Request Changes”)
- **Document Section:** List of uploaded files with download link
- **Buttons (depending on role):**
 - **Approve / Reject / Request Changes** (for Manager)
 - **Edit** (if allowed)
 - **Back to List**
- **Actions:**
 - Approve / Reject / Request Changes has associated status change + maybe popup for reason (if needed)
 - On changes: log history (audit trail)

5. Non-functional Requirements

- **Authentication & Security:**
 - Use HTTPS (TLS) for all data in transit
 - Use secure password hashing (e.g. bcrypt)
 - Role-based access control strictly enforced
- **Performance:**
 - Dashboard load time ≤ 3 seconds under expected load
 - Form submission response within 1–2 seconds
- **Scalability:**
 - Should support up to (specify number) shipments in future
 - Database indexing on frequently queried columns (status, dates)
- **Usability:**
 - Responsive UI — Should work on desktop and tablets
 - Clear error messages for validation failures
- **Audit Logging:**
 - Log who created / updated / approved / rejected shipments
 - Timestamp + user ID for each action

6. Integration / Data Model

- **Database Tables (suggested):**
 - Users (id, username, password_hash, role, created_at, updated_at)
 - Shipments (id, shipment_id, sender_name, sender_address, receiver_name, receiver_address, description, weight, dimensions, value, pickup_date, expected_delivery_date, mode, status, created_by, created_at, updated_at)
 - Documents (id, shipment_id, file_name, file_path, uploaded_by, uploaded_at)

- AuditLog (id, user_id, shipment_id, action, timestamp, remarks)
 - **APIs:**
 - POST /api/shipments → Create shipment
 - GET /api/shipments/:id → Get detail
 - PUT /api/shipments/:id → Update shipment (for changes / approval)
 - GET /api/shipments → List shipments (filter by role, status)
 - POST /api/users → (Admin only) Create user
-

7. Testing / Acceptance Criteria

- **Test Cases:**
 - Login with each role → Should redirect to correct dashboard
 - Create shipment with valid data → Should save + status = “Created”
 - Create shipment with missing mandatory field → Show validation error
 - Upload invalid file type → Reject upload
 - Manager approves shipment → Status changes, email sent
 - Manager rejects shipment with reason → Status updates, reason stored
 - Dashboard metrics reflect real data
 - **Acceptance:**
 - All CRUD operations working
 - Role-based views are correct
 - UI is responsive
 - Emails / notifications (if any) sent correctly
 - Security checks passed (basic auth, role-based access)
-

8. Assumptions & Constraints

- Users will have access to modern browsers.

- Internet connection is stable enough for uploads / downloads.
 - File storage: server can handle uploaded documents.
 - Email system / SMTP is already available (or to be provisioned separately).
 - No major budget constraints for building a moderate-sized web app.
-