Times United – Inspection Division Software Requirements (MVP)

1. Overview

This document defines the functional and technical requirements for automating the **Inspection Division** of Times United Verifications & Inspections. It covers the complete workflow from job order creation to inspection, approval, certificate issuance, and QR-based publishing. This is a **developer-ready specification**.

2. Core Workflow

1. **Job Order Creation**
2. Created by Admin *or* Inspector.
3. Auto-generated Job Order number.
4. Fields: Client, Project, Site contact, Requester, Tentative inspection date, Scope (equipment inspections, operator certification, etc.).
5. **Assignment**
6. Assigned by Admin/Team Leader to Inspector.
7. Inspector receives notification.
8. **Scheduling**
9. Inspector coordinates with client and updates actual inspection date/time.
10. **Inspection Execution**
11. Inspector uses mobile app (offline-capable).
12. Selects **Inspection** tab.
13. Selects equipment type (search/auto-suggest).
14. Equipment form auto-populates job header + equipment details.
15. Inputs: manufacturer, year, serial no., max capacity, previous inspection date, validity period.
16. Completes checklist (per template).
17. Captures mandatory photos (predefined slots: front, side, rear, hydraulics, engine, cabin).
18. Geo-tagging: ON by default, can be toggled off.
19. Marks equipment as Safe/Not Safe.
20. Captures inspector + client signatures.
21. **Field Inspection Report (FIR)**

22. Generated per Job Order after last equipment submitted.
23. Auto-sent to client email (from job order).
24. Inspector can copy/share link (WhatsApp friendly).
25. **Approval & Certificate Generation**
26. Job sent to Approver (Technical Manager/Team Leader).
27. Approver reviews checklist, photos, signatures → Approve/Reject.
28. On approval, system generates Certificate PDF.
29. Certificates downloadable **with letterhead** (digital send) or **without letterhead** (print on pre-printed stationery).
30. **Sticker/QR Binding**
31. Each equipment gets a unique pre-printed sticker (QR + number, format: `TUVINSP-000001`).
32. Inspector enters sticker number → links sticker to equipment + certificate.
33. Scanning QR shows certificate/report.
34. Reinspection: scanning old sticker pre-fills equipment details; inspector assigns new sticker, completes new checklist, and submits.
35. **Publishing**
36. After approval, certificate is generated but **not public** until published.
37. Roles allowed to publish: Inspector, Team Leader, Technical Manager (configurable).
38. Once published, certificates are visible to clients and via QR.

3. System States

- Job Order: DRAFT → ASSIGNED → SCHEDULED → IN_PROGRESS → SUBMITTED → UNDER_REVIEW → APPROVED → CERT_GENERATED → PUBLISHED.
- Stickers: UNASSIGNED → ASSIGNED → HISTORICAL (on reinspection).

4. Database Schema (core tables)

```
clients(id, name, contact_person, email, phone, address, billing_reference)
job_orders(id, client_id, po_reference, created_by, status, site_location,
scheduled_start, scheduled_end)
job_line_items(id, job_order_id, type, equipment_id, description, quantity,
status)
equipment(id, client_id, tag_code, type, manufacturer, model, serial_number,
swl, location, next_due)
inspections(id, job_line_item_id, inspector_id, checklist_template_id,
start_time, end_time, status)
inspection_answers(id, inspection_id, checklist_row_id, result, comment,
```

```
photo_refs)
certificates(id, inspection_id, generated_by, pdf_blob, qr_code, issued_date,
approval_chain)
stickers(id, sticker_code, qr_payload, status, assigned_equipment_id,
assigned_at, assigned_by)
field_inspection_reports(id, job_order_id, fir_pdf, summary, sent_to,
share_link_token)
approvals(id, entity_type, entity_id, approver_id, decision, comment,
decided_at)
publications(id, job_order_id, published_by, published_at, status, note)
users(id, name, email, role, competence, active)
```

5. API Endpoints (sample)

- `POST /api/job_orders` – create job order.
- `GET /api/job_orders/{id}` – retrieve job order + line items.
- `POST /api/job_orders/{id}/assign` – assign inspector.
- `POST /api/inspections` – start inspection.
- `PATCH /api/inspections/{id}` – update checklist answers/photos.
- `POST /api/inspections/{id}/submit` – submit inspection.
- `POST /api/certificates/{inspection_id}/generate` – generate certificate PDF.
- `POST /api/publish/{job_order_id}` – publish approved certificates.
- `GET /api/stickers/{code}/resolve` – resolve sticker to equipment + certificate.

6. UI/UX Requirements

- Mobile tabs: Inspection • Operator Certification • Training.
- Equipment template forms with prefilled values (on reinspection).
- Photo capture matrix with predefined slots.
- Signature capture widget (inspector + client).
- Approver dashboard with side-by-side checklist/photos.

7. Compliance & Traceability

- ISO/IEC 17020 mapped: impartiality, record control, methods/procedures, reporting, NC/CAR linkage.
- Stickers + sticker history ensure 10-year record retention.
- All actions logged (audit trail).

8. Acceptance Criteria

- Inspector creates job order, completes inspection offline, syncs online.
- System generates FIR and sends to client.
- Approver reviews, approves, system generates certificate.
- Inspector/Manager publishes; certificate visible via client portal and QR.
- Reinspection via sticker scan pre-fills equipment data.

9. Workflow Improvements Incorporated

1. **Backend Data & Reporting**: All job orders, inspections, certificates, stickers, PIFs, and documents are stored and exportable. Add reporting module with Excel/CSV exports by Job Order, Equipment, Inspector, Client, Region, Date Range, Status. Reports can also be scheduled and emailed.
2. **Finance Department Integration (Basic)**: When a Job Order is Closed + Published, it is flagged as "Ready for Finance." Finance staff can view these jobs, manually enter invoice numbers from their accounting system, and mark them invoiced. Fields: invoice_number, finance_status. Later: optional API integration with external accounting.
3. **Workflow Governance**: Introduce **Closed** state after publishing so jobs clearly move from operational to finance stage.
4. **Audit Trail**: All exports, invoice tagging, and publishing events are logged. Finance entries (invoice numbers) become part of ISO 17020 evidence.
5. **Inspection Request Generation**: On creation of any Job Order, the system automatically generates an **Inspection Request record** using the customer representative info from the job order. This fulfills ISO 17020 written request requirement. Clients can also create inspection requests directly in their portal; admins decide whether to convert them into Job Orders.
6. **Self-Assignment Option**: When inspectors create Job Orders, they can assign them to self or to another inspector. Admins/Team Leads can override as needed.
7. **QR Scan Equipment Lookup**: During inspection execution, inspectors can either choose equipment from dropdown or scan an existing QR sticker to auto-populate equipment details.
8. **Tool & Calibration Module**: Maintain master list of tools, calibration records, expiry reminders. Tools are allocated permanently to inspectors. When submitting an inspection, inspector must select which assigned tools were used. FIR includes tool list used. Daily logs are system-generated by Tool Manager role for ISO 17020 compliance.
9. **Customer Notifications**: After certificate approval (pre-publication), customer receives email notification that a certificate is ready but not yet visible. Certificate becomes viewable only after publishing.
10. **Inspector UI Enhancements**: Landing page shows last 5–10 job orders with status (Open/Closed). Separate tab for Guidance Documents & Regulations (as prepared in QMS module) accessible directly during inspections.
11. **Separation of QMS**: ISO/IEC 17020 QMS module remains separate from inspection activity. Inspection checklists are pulled from controlled QMS documents (with version and control numbers) but NC/CAR workflow is handled in QMS module, not in the inspection flow.
