TapSure — Engineering Checklist (Step-by-step, POS Automation MVP)

Version: 0.1

Date: 2025-12-12

Scope: Convert TapSure into a POS-integrated automation service (webhook intake → case lifecycle → deterministic decision → partner callback). No fluff.

Assumptions:

- TapSure is NOT a risk carrier in MVP.
- Core output contract is: `approve | deny | needs-more-info` + reasons + required evidence.

---

0) Freeze the wedge

- [ ] Define ONE pilot program (single geography + single product class).
- [ ] Define ONE buyer (merchant program owner OR insurer ops OR TPA).
- [ ] Define ONE measurable KPI (pick 1):

- cycle time to decision
- attach rate at POS
- cost per case
- % automated without human touch

- [ ] Define failure handling: what triggers `needs-more-info` vs “human review”.

Acceptance:

- A written “Pilot Spec” exists (1 page) with schemas + KPIs.

---

1) Introduce a persistent Case model

Target files:

- `backend/app/models.py`
- `backend/app/orchestrator.py`

- [ ] Add `Case` (or `ClaimCase`) model with fields:

- `id` (uuid)
- `tenant_id`
- `status` (enum: `received|processing|needs_more_info|approved|denied|issued|closed|error`)

- `source` (`pos_webhook|upload|manual`)
- `transaction` (normalized object)
- `extracted` (OCR/parse results)
- `decision` (decision contract)
- `created_at/updated_at`

- [ ] Add `CaseEvent` model:
- `case_id`, `type`, `payload`, `timestamp`, `request_id`

Acceptance:

- Unit test: Case status transitions are valid.

---

2) Add persistence (SQLite first)

Target:

- Add `backend/app/db.py` (or similar)
- Update `backend/app/main.py` startup/shutdown

- [ ] Add a minimal DB layer (SQLite) storing:
- cases
- case_events
- tenants
- api_keys

- [ ] Add migrations strategy (simple SQL migrations folder is fine).

Acceptance:

- Restarting the API does not lose cases.
- `GET /health` includes DB connectivity check.

---

3) Tenant auth (API keys) + request IDs

Target:

- `backend/app/main.py`

- [ ] Implement `X-Api-Key` auth middleware/dependency.
- [ ] Add tenant lookup by api key.
- [ ] Add `X-Request-Id` (generate if missing) and include in logs.

Acceptance:

- Unauthorized requests return `401`.
- Authorized requests include `tenant_id` in case records.

---

4) POS webhook intake endpoint (core ingestion)

Target:

- `backend/app/main.py`
- `backend/app/models.py`

- [ ] Add `POST /api/pos/intake` (JSON) that accepts:

- partner event id (idempotency)
- merchant/store ids
- timestamp
- totals, currency
- line items (sku/upc/name/category/qty/price)
- customer contact (optional)

- [ ] Enforce idempotency:

- reject duplicates OR return existing case id for same idempotency key.

- [ ] Create a case with `status=received`.

Acceptance:

- Re-sending the same intake returns same `case_id`.

---

5) Deterministic decision contract + rule trace

Target:

- `backend/app/agents/coverage.py`
- `backend/app/models.py`

- [ ] Define decision output model:

- `decision` (`approve|deny|needs-more-info`)
- `reasons[]` (machine-readable codes)
- `missing_fields[]`
- `trace[]` (rules triggered + inputs)
- `recommended_offers[]` (optional)

- [ ] Implement rule evaluation without LLM.
- [ ] Add constraints:

- excluded categories
- max item value
- jurisdiction flags
- waiting period (if applicable)

Acceptance:

- Golden tests: given a transaction payload, decision + trace match expected.

---

6) Async processing (don’t block POS)

Target:

- `backend/app/main.py`
- new worker module (e.g., `backend/app/worker.py`)

- [ ] Intake endpoint returns immediately (<500ms) with `case_id`.
- [ ] Implement background execution:

- MVP option A: FastAPI `BackgroundTasks`
- MVP option B: lightweight queue (RQ/Redis) if available

- [ ] Processing steps:

- normalize transaction
- optional OCR/image parse if provided
- run deterministic decision
- update case + write events

Acceptance:

- Load test (simple): 50 intakes do not time out.

---

7) Partner callback (results webhook)

Target:

- `backend/app/main.py`
- new module `backend/app/integrations/webhooks.py`

- [ ] Store per-tenant callback URL + secret.
- [ ] POST results to partner:

- case_id
- status

- decision
- offers
- timestamp

- [ ] Sign callbacks (HMAC) and include request id.
- [ ] Retry with backoff; mark delivery failures in case events.

Acceptance:

- Local stub receiver gets callbacks; signature verifies.

---

8) Abuse protection + limits

Target:

- `backend/app/main.py`

- [ ] Set max request size.
- [ ] Rate limit per API key.
- [ ] Validate content-type strictly for all endpoints.
- [ ] If file uploads remain:

- accept only allowlisted image types
- scan strategy placeholder documented
- store outside repo, randomized names, no path traversal

Acceptance:

- Oversized requests return `413`.
- Burst requests return `429`.

---

9) Observability (minimum viable)

Target:

- `backend/app/main.py`

- [ ] Structured logging (json) including: request_id, tenant_id, case_id.
- [ ] Metrics counters (can be simple):

- cases_received
- cases_processed
- decision_counts
- callback_failures

- processing_latency_ms

Acceptance:

- Logs are queryable by `case_id`.

---

10) Frontend-v2: partner demo mode (optional)

Target:

- `frontend-v2/src/lib/api.ts`
- `frontend-v2/src/App.tsx`

- [ ] Add a “POS intake simulator” screen ONLY if needed for demo.
- [ ] Use API key header support.
- [ ] Show case status polling.

Acceptance:

- Demo can run with just webhook intake + polling.

---

11) Testing (non-negotiable contract tests)

Target:

- `backend/test_integration.py`
- add `backend/test_pos_intake.py`

- [ ] Tests for:

- auth required
- idempotency
- case persisted
- decision trace stable
- callback signing

Acceptance:

- `pytest` passes on clean env.

---

12) Partner deliverables

- [ ] `docs/pos-integration.md`:

- schemas

- auth
- idempotency
- retries
- callback verification

- [ ] Postman collection (or curl examples) for:

- intake
- status
- callback verification

Acceptance:

- A partner can integrate with docs only.

---

13) “Done” definition for first pilot

- [ ] One partner can send intake events.
- [ ] TapSure creates a case, processes async, returns a decision.
- [ ] Partner receives signed callback.
- [ ] You can report KPI for 50–200 pilot cases.