Sprint 01 Board Setup

Repositories (create first)

- **Repo 1: fieldops-mobile-flutter**
- **Repo 2: fieldops-api-dotnet**
- **Repo 3: fieldops-admin-react**

Jira project structure (recommended)

- **Epics = big modules**
- **Stories/Tasks = implementable items**
- **Labels: mobile, api, web, db, devops, qa, docs**

EPIC 0 — Project Setup & Standards

Task 0.1 — Create GitHub repos

- Create 3 repos with above names
- Add collaborators
- Add branch protection for main
- Create dev branch (optional)

Task 0.2 — Add README + run instructions (all repos)

- Local run steps
- Env variables template
- Basic folder structure

Task 0.3 — Decide branching + PR rules

- feature branches: feature/<name>
- PR template (checkbox: tested, screenshots, linked issue)

Task 0.4 — Setup Jira board

- Columns: Backlog / Sprint / In Progress / Review / Done

- Define labels + naming convention

Task 0.5 — Definition of Done (DoD) document

- Code complete, tested, reviewed, merged, demo-ready, evidence captured

---

EPIC 1 — Database & Backend Foundation (ASP.NET + PostgreSQL)

Task 1.1 — Setup PostgreSQL (local)

- Install Postgres
- Create DB: fieldops_db
- Create user + password

Task 1.2 — Create ASP.NET Core Web API project

- Project scaffold
- Add Swagger
- Add health endpoint GET /health

Task 1.3 — Add EF Core + PostgreSQL connection

- Add EF Core packages
- Add connection string via appsettings.json
- Verify migration runs

Task 1.4 — Create DB schema (Sprint 1 minimum)

Tables (minimum):

- users (Rep/BU)
- territories
- shops (lastVisitedAt, lat, lng)
- products (sku, casePack)
- settings (visitFrequencyDays)
- day_sessions
- lorry_stock_opening
- visits

- visit_osa
- orders
- order_items
- deliveries
- returns

Task 1.5 — Seed data script (dev only)

- Seed: 1 BU user, 2 reps
- Seed: 2 territories, 20 shops
- Seed: 30 products

Task 1.6 — Basic logging + error handling middleware

- Standard error response format
- Log request failures

EPIC 2 — Auth & Role-Based Access (Rep + BU)

Task 2.1 — Implement login endpoint

- POST /auth/login
- Return JWT + role + user id

Task 2.2 — JWT authentication middleware

- Protect endpoints
- Role checks: BU vs Rep

Task 2.3 — Create BU login page (Web)

- Login UI
- Save token

Task 2.4 — Create Rep login page (Mobile)

- Login UI
- Save token securely (flutter_secure_storage)

QA Task 2.5 — Auth test cases

- valid login
- invalid password
- role access restrictions

EPIC 3 — Shop Visit Frequency Rule + Today's Shop List

Task 3.1 — Settings API for visit frequency

- GET /settings/visit-frequency
- PUT /settings/visit-frequency (BU only)

Task 3.2 — Due shops API

- GET /shops/due?repId=...
  Logic:
- due if lastVisitedAt + frequencyDays <= today
- include overdue flag

Task 3.3 — Mobile: Today's due shop list screen

- List + search by name/id
- Show tags: Due / Overdue

Task 3.4 — Web: Settings screen for frequency days

- BU can change 14 → any number

QA Task 3.5 — Verify rule works

- change frequency and confirm due list changes

EPIC 4 — Start Day (Vehicle + Opening Lorry Stock)

Task 4.1 — API: Start day session

- POST /day/start
  Payload:
- vehicle id/name
- opening stock items (sku, qty)

Task 4.2 — Mobile: Start Day UI

- select vehicle
- enter opening lorry stock
- save locally (Hive) + submit online if available

Task 4.3 — Local persistence (Hive)

- Hive boxes:
    - session_box
    - pending_sync_box
    - master_data_box (shops/products cached)

QA Task 4.4 — Start day validations

- qty must be >= 0
- vehicle required

EPIC 5 — Visit Execution + OSA Capture (Core Feature 02)

Task 5.1 — Mobile: Start visit / End visit flow

- timestamps
- optional GPS capture (store lat/lng)

Task 5.2 — API: Create visit

Option A (recommended): handled inside bulk sync
Option B: POST /visits

Task 5.3 — Mobile: OSA form

For each SKU (or selected SKUs):

- On-shelf (Yes/No)
- If No → reason code:
    - no stock / backroom / not listed / competitor blocked
- Backroom qty
- expiry/damage checkbox
- optional photo attach (store locally for now)

Task 5.4 — Generate OSA issues list

- auto list of missing on-shelf items

QA Task 5.5 — OSA tests

- reason code required when No
- backroom qty numeric

EPIC 6 — Order Taking + Smart Suggestion

Task 6.1 — Smart suggestion calculation (mobile)

Formula:

- suggested = avgSales * leadTime - (shelf + backroom)
- case pack rounding
- MOQ/max warnings

Task 6.2 — Mobile: Order screen

- Accept suggestion / edit qty
- Repeat last order (optional)
- Confirm order

Task 6.3 — API: Store order + items

Option A: via POST /sync/day
Option B: POST /orders

QA Task 6.4 — Order tests

- qty cannot be negative
- warnings show correctly

EPIC 7 — Delivery Handling + Stock Deduction (Full/Partial)

Task 7.1 — Mobile: Delivery screen

- Full delivery if stock enough
- Partial delivery if stock not enough

- Next delivery date for partial
- auto backorder flag

Task 7.2 — Stock deduction logic (mobile)

- reduce lorry stock after delivery

Task 7.3 — API: Save delivery records

- full/partial status
- remaining qty
- next date

QA Task 7.4 — Delivery test cases

- full delivery deducts correctly
- partial delivery creates pending remainder

EPIC 8 — End of Day + Returns + Reconciliation

Task 8.1 — Mobile: End day stock count UI

- closing stock entry

Task 8.2 — Reconciliation summary UI

- Opening − Delivered = Expected Remaining
- show mismatch alert

Task 8.3 — Returns entry UI

- sku, qty, reason (damage/expiry/overstock)

Task 8.4 — API: Store end day + returns

- via bulk sync

QA Task 8.5 — End day tests

- mismatch alert displays
- return qty validation

EPIC 9 — Bulk Sync (Most Important) + Offline Queue

Task 9.1 — Mobile: Offline-first storage model

- Store all actions into pending_sync_box events

Task 9.2 — Mobile: Upload Day button + summary

- show counts
- upload triggers sync

Task 9.3 — API: POST /sync/day bulk endpoint

Payload includes:

- day session
- visits
- osa records
- orders + items
- deliveries
- returns
- closing stock

Task 9.4 — API: Idempotency / duplicate protection

- accept clientGeneratedId per entity
- don’t create duplicates on retry

QA Task 9.5 — Offline sync tests

- create data offline → upload later → appears in web
- retry upload doesn’t duplicate rows

EPIC 10 — BU Web Monitoring (Core Feature 03)

Task 10.1 — Monitoring APIs

- GET /monitoring/summary
- GET /monitoring/visits
- GET /monitoring/osa
- GET /monitoring/orders

- GET /monitoring/deliveries
- GET /monitoring/returns

Task 10.2 — React: Monitoring dashboard page

- filter by date range, territory, rep

Task 10.3 — React: Visits list view

- shop, rep, date, status

Task 10.4 — React: OSA Issues view

- missing SKUs + reason codes

Task 10.5 — React: Orders + Delivery status view

- completed/partial/pending counts

Task 10.6 — React: Returns summary view

- totals by reason

QA Task 10.7 — Web monitoring tests

- filters work
- data matches uploaded records

EPIC 11 — DevOps (Optional if time, but professional)

Task 11.1 — Deploy Postgres (Neon/Supabase)

Task 11.2 — Deploy API (Render/Fly/Railway)

Task 11.3 — Deploy Admin Web (Vercel/Netlify)

Task 11.4 — Update env vars for production

QA Task 11.5 — Smoke test deployed system

EPIC 12 — QA + Documentation + Evidence Pack

Task 12.1 — Write test plan + test cases (Sprint 1)

- login tests
- due shop rule tests

- sync tests
- delivery partial/full tests
- monitoring tests

Task 12.2 — Collect evidence screenshots

- mobile screens
- Swagger/Postman
- web dashboard

Task 12.3 — Sprint 1 demo script (1 page)

Task 12.4 — Update diagrams (BA)

- Use case diagram (Sprint 1 scope)
- Activity diagram: Start day → visit → upload
- Data flow diagram: Mobile → API → DB → Web