Ewe Pregnancy Flow - Detailed Specification (Updated)

Overview

This document describes the ewe pregnancy management flow for the farm management application. It covers sponge and hormone preparation, insemination steps, pregnancy checks, gestation windows, lambing, and per-cycle flexibility (INS2 optional). It also includes reminder logic, a simplified state machine, and UI notes.

1. SPONGE & HORMONE Timeline

- SPONGE (SPG): Inserted to synchronize cycles; typically left for 12–14 days, then removed. - HORMONE (HRM): Often administered at sponge removal (single event). - INS1: Timed ~36–48h after sponge removal. - INS2: Optional, typically ~17 days after INS1. Reminders: - After SPG insert: Remove sponge in 12–14d (overdue alert at +15d). - After SPG remove: Give HRM (optional), plan INS1 in 36–48h.

2. Pregnancy Checks

- Check #1: Earliest 28 days after INS1. - Check #2: Optional at 45–50 days, or farms may choose a single late check ~50d. - App enforces earliest CK1 ≥ INS1+28d; CK2 suggested 45–50d.

3. Gestation & Due Windows

- Gestation ≈ 150 days from conception. - If only INS1: Due = INS1 + 150d. - If INS2 also done: Due window = [INS1+150, INS2+150] (≈17-day span). - Estimated due date = midpoint of window for simple displays. - Lambing window reminders: Start 7 days before due_start, overdue alert 3 days after due_end.

4. Data Model (Entities & Fields)

- EWE: ewe_id, tag_id, dob, status, notes. - BREEDING_CYCLE: cycle_id, ewe_id, status, sponge_insert/remove, hormone_date, ins1_date, ins2_date (opt), check_plan, check1/2_date & results, preg_confirmed_date, gestation_anchor, due_window_start/end, est_due_date, notes. - BREEDING_EVENT: event_id, cycle_id, ewe_id, type (SPG_INSERT, SPG_REMOVE, HRM, INS1, INS2, CK1, CK2, PREG_CONFIRMED, LAMBING, LOSS), event_date, operator, location, ram_or_semen_batch, notes. - LAMBING: occurred_at, litter_size, male/female counts, birth_weights_json, assistance, outcome_notes. - REMINDER: reminder_id, ewe_id, cycle_id, type, scheduled_for, status, channel. - TREATMENT: drug, dose, date, cost, notes.

5. State Machine (Simplified)

NOT_PREGNANT → SPG_INSERT → SPG_REMOVE → HRM → INS1 → [INS2?] → CK1 → [CK2?] → PREGNANT → LAMBING | LOST/FAILED.

6. Reminder Logic

- On INS1: schedule INS2_SUGGEST (+17d, optional), CK1 (+28d), CK2 (+47–50d if two-check plan). - On INS2: recompute due_window_end = INS2 + 150d; optionally re-anchor CK2. - On PREG_CONFIRMED: schedule PREP_LAMBING (start−7d), OVERDUE (end+3d), cancel pending checks. - On CHECK_FAIL: close cycle as FAILED; draft next cycle in 3–7d.

7. UI/UX Highlights

- Timeline chips: SPG Insert → SPG Remove → HRM → INS1 → INS2 → CK1 → CK2 → PRG → LAMBING. - Due window card (start, end, midpoint) with anchor toggle (INS1/INS2/Ultrasound). - Batch actions, color badges, smart date pickers with minimums.

8. Lambing & Conception Resolution (NEW)

Problem: When both INS1 and INS2 are used, exact conception date is uncertain before lambing. Rule of Thumb: - If lambing occurs ≤ 153 days after INS1 → assume conception = INS1. - If lambing occurs > 153 days after INS1 → assume conception = INS2. Rationale: Gestation ~150 days (±3). Beyond 153d from INS1 is too late for INS1 but consistent with INS2+150. Model: - BREEDING_CYCLE adds: conception_source (UNKNOWN|INS1|INS2|ULTRASOUND_ESTIMATE|RESOLVED_AT_LAMBING), resolved_conception_date (date). Logic (at LAMBING): - Compute days_from_ins1 = lambing_date − ins1_date. - If has_ins2: - days_from_ins1 ≤ 153 → source=INS1, resolved_conception_date=ins1_date. - else → source=INS2, resolved_conception_date=ins2_date. Else (no INS2): source=INS1. - Update status to LAMBED and persist. UI: - Before lambing: show due window [INS1+150 … INS2+150]. - After lambing: display “Conception resolved: INS1/INS2” with computed gestation days. Analytics: - Enables success rates for INS1 vs. INS2, tighter herd KPIs.

Health & Treatments Module - Detailed Specification (Updated)

Overview

This document defines the Health & Treatments module for the farm management MVP.
It standardizes how farms log diagnoses, treatments, supplements, reminders, and medical history for each ewe.
It also enables analytics, compliance (withdrawal tracking), and integration with breeding/sales/inventory modules.

1. Goals

- Track ewe health status (Healthy, Sick, Recovering).
- Record diagnoses, treatments, supplements, and post-lambing supportive care.
- Manage reminders for next doses, follow-ups, and withdrawal end dates.
- Maintain a searchable medical history and audited event ledger.
- Provide analytics (costs, coverage, common diseases) and integrate with other modules.

2. Health Status Model

Status options: HEALTHY, SICK, RECOVERING, (optional) CULL/REMOVED.
Automatic transitions:
- New Diagnosis -> SICK
- Treatment started -> RECOVERING
- Follow-up marked success -> HEALTHY

Manual override allowed with reason capture.

3. Treatment & Supplement Workflow

Diagnosis -> Treatment/Supplement -> Follow-up -> Recovery.

Diagnosis:
- Disease, injury, or symptom log.

Treatment:
- Antibiotics, vaccines, vitamins, hormones, anti-parasitics.

Supplements (supportive care):
- Vitamins/minerals (selenium, calcium, iron, etc.).
- Hormones (Oxytocin for milk let-down, uterus contraction).
- Antibiotics (Ceftanel, Penicillin, etc.) after lambing for infection prevention.
- Seasonal vaccines or dewormers.

Post-lambing care:
- When lambing is logged, app shows supplement checklist (Oxy, antibiotic, vitamin/mineral).
- Farmer selects relevant items, app creates treatment plans and reminders.

4. Data Model (Entities & Fields)

TREATMENT_CATALOG: add drug_type values ANTIBIOTIC|VACCINE|VITAMIN_SUPPLEMENT|HORMONE|ANTI_PARASITIC|OTHER.
EWE_HEALTH_EVENT: diagnosis, treatment, supplement, follow-up, recovery, death.
EWE_TREATMENT: supports supplements as well as curative treatments.
WITHDRAWAL_TRACK: vitamins/minerals usually withdrawal_days=0, antibiotics/hormones use proper withdrawal days.

5. Reminder Logic

- Post-lambing supplement plan auto-schedules:
- Oxy: 1–3 injections (per catalog default).
- Antibiotic: 3–5 day course with withdrawal reminder.
- Vitamin/mineral: single dose, no withdrawal.
- Normal treatment reminder system still applies (doses, follow-up, withdrawal end).

6. UI/UX

- Health Tab: Status badge, Current Treatments, Supplements timeline.
- Post-lambing checklist (Oxy, antibiotic, vitamin/mineral).
- Withdrawal badge if ewe under restriction.
- Batch actions: vaccinate or supplement a group (e.g., vitamin bolus for 20 ewes).

7. Analytics

- Cost per ewe per treatment/supplement.
- Usage reports (Oxy, antibiotics, vitamins).
- Compare outcomes of supplemented vs. non-supplemented ewes.
- Vaccination coverage, withdrawal compliance.

8. Integration

- Breeding: pause breeding reminders if SICK, resume once HEALTHY.
- Post-lambing care integrated with lambing event workflow.
- Inventory: decrement medicine/supplement stock.
- Sales: block sale if ewe under withdrawal.
- Finance: roll costs into per-ewe profitability.

Feeding & Weight Module - Detailed Specification (Updated)

Overview

This document defines the Feeding & Weight module for the farm management MVP.
It covers ewe/lamb weight tracking, feed plans, feed usage & costs, lamb milk-feeding (nursing vs manufactured),
and analytics such as feed conversion ratio and average daily gain.

1. Goals

- Track each animal’s weight history and growth curve.
- Define and manage feed rations per group (ewes, rams, lambs).
- Monitor feed intake vs. weight gain to calculate efficiency and costs.
- Track lamb milk-feeding (nursing vs manufactured) from 0–35 days.
- Provide alerts for underweight/overweight animals and low stock.

2. Weight Tracking

- Manual or batch entry, or auto-import from RFID scale.
- Store: weight_date, weight_kg, operator, location, notes.
- Growth status thresholds: Underweight, Healthy, Overweight (per group rules).
- Growth curve overlays to benchmark progress.

3. Feed Plans

- Group-level ration plans: roughage, concentrate, supplements.
- Define feed items with quantity per ewe/day and cost per unit.
- Apply plan to group -> system calculates daily/weekly feed usage and costs.
- Supports supplements (minerals, vitamins, salt licks).

4. Lamb Milk-Feeding (0–35 days)

Methods:
- Natural Nursing (default):
- Lamb linked to dam ewe, no daily milk records.
- Growth tracked via lamb weights.
- Manufactured Milk Feeding:
- Record replacer brand/type, daily quantity per lamb or group.
- Costs tracked; stock deducted from inventory.
- Daily/weekly feeding logs required.

Workflow:
- At lambing: assign lamb to dam (nursing) or artificial feeding.
- Manufactured feeding: create feeding plan, auto-schedule daily logs.
- At ~30–35 days: reminder to transition lamb to starter feed.

Data Model Additions:

- LAMB_FEEDING: id, lamb_id, method (NURSING|MANUFACTURED), start_date, end_date, dam_ewe_id (nullable), milk_replacer_type (nullable), avg_daily_amount_liters (nullable).

Reminders:
- Manufactured feeding: daily feed log, low stock alerts.
- Transition reminder at 30–35 days.

Analytics:
- % lambs nursing vs. manufactured feeding.
- Costs per artificially fed lamb.
- Growth/mortality differences between groups.
- Avg weaning age and weight.

5. Data Model (Entities & Fields)

EWE_WEIGHT:
- id, ewe_id, weight_date, weight_kg, operator, notes.

FEED_ITEM (catalog):
- feed_id, name, category (ROUGHAGE|CONCENTRATE|SUPPLEMENT|MILK_REPLACER|OTHER), unit (kg, L), cost_per_unit, nutritional_values_json.

FEED_PLAN:
- plan_id, name, group_type (LAMBS|EWES_LACTATING|RAMS|ALL), start_date, end_date, active_flag.

FEED_PLAN_ITEM:
- id, plan_id, feed_id, amount_per_ewe_per_day, unit.

FEED_USAGE_LOG:
- id, plan_id, date, group_id (optional), target (EWE|LAMB|GROUP), total_feed_given, notes, cost.

LAMB_FEEDING:
- id, lamb_id, method (NURSING|MANUFACTURED), start_date, end_date, dam_ewe_id, milk_replacer_type, avg_daily_amount_liters.

6. Reminders & Alerts

- Weight check reminders (configurable, e.g., every 30 days for lambs).
- Alert: weight deviation > ±15% from growth curve.
- Low feed stock alert (including milk replacer).
- Feed cost overrun alert if above threshold.
- Lamb transition reminder: at ~30–35d move from milk to starter feed.

7. UI/UX

Animal Profile > Weight/Feeding tabs:
- Weight line chart with growth curve overlay.
- Quick Add Weight action.

Lamb Profile > Feeding tab:

- Badge: Nursing vs Manufactured.
- Nursing: show dam link; Manufactured: show replacer plan, daily logs.

Feed Plans page:
- Define rations by group; daily/monthly costs visible.

Reports:
- Feed cost per ewe/lamb/group.
- Feed conversion ratio (FCR = kg feed / kg weight gain).
- % lambs artificially fed, cost/lamb, survival rate.

8. Analytics

- Average Daily Gain (ADG).
- Feed Conversion Efficiency (FCR).
- Cost contribution of feed/milk replacer to herd profitability.
- Compare growth & mortality between nursing vs manufactured feeding lambs.

9. Integration

- Inventory: auto-deduct feed items and milk replacer from stock.
- Sales: weights used for live-weight pricing.
- Health: weight loss triggers alerts; orphan/triplet lambs flagged high risk.
- Breeding: ewe body condition linked to fertility readiness.

Inventory & Costs Module - Detailed Specification (Always-On, Best-Effort Costing)

Overview

This update redefines Inventory & Costs as **always enabled**, but calculates costs **best■effort** based on whatever data is available.
When data is incomplete (no GRN, no stock, free■text item names), the system estimates costs using fallbacks and clearly marks results as **Estimated** with a confidence score.
As data is backfilled later, historical costs are retro■recomputed and upgraded from Estimated → Actual.

1. Philosophy

- No blocking: users can log Feeding/Treatments/Supplements without inventory setup.
- Compute what we can, when we can; never lose the entry.
- Be transparent: show badges (Actual/Estimated), tooltips with how we computed it.
- Auto■improve over time: when purchases/stock are added later, recompute past entries.

2. Cost Source Hierarchy (Highest → Lowest)

For each usage line (feed, drug, supplement), pick the first available source:

1) **Actual Batch Cost (FIFO/Avg)**
- From GRN/BATCH at the issuing location & date.
2) **Last Known Purchase Price (LKP)**
- Most recent GRN price for the item (any location) prior to usage date.
3) **Catalog Default Cost**
- ITEM.default_cost in base unit.
4) **Derived Cost From Parent/Alt Unit**
- Use unit_to_base_factor; derive pack→kg/L cost.
5) **Name■Matched Heuristic**
- Fuzzy match free■text item name to catalog (≥0.85 similarity).
6) **Manual Reference Price**
- Farm■level “reference price” table (user seed data).
7) **Zero■Cost Placeholder**
- If all else fails: cost = 0, confidence = Very Low. Flag for user action.

3. Confidence Levels & Badges

Each calculated cost gets a confidence label:
- **Actual** (Batch/FIFO or Moving Avg) -> Green badge.
- **Estimated (High)** (LKP, Catalog Default) -> Blue badge.
- **Estimated (Medium)** (Derived unit, Manual Reference) -> Yellow badge.
- **Estimated (Low)** (Fuzzy name match, Zero placeholder) -> Orange/Red badge.

Reports roll up both **Actual** and **Estimated**; filters allow excluding low■confidence rows.

4. Fuzzy Matching for Free■Text Items

When users type a feed/drug name that isn't in catalog:
- Try fuzzy match to existing ITEMs (normalized strings, synonyms). Threshold default = 0.85.
- If matched: map usage line to ITEM; use cost hierarchy above.
- If not matched: create a **Pending Catalog Draft** for review with captured name, unit, and any observed costs.

5. Unit & Quantity Handling

- All costs normalized to a **base unit** (kg or L or unit).
- Support pack sizes (e.g., 1 bag = 25 kg), concentration (e.g., 100 mg/ml) for drugs.
- If entry missing unit: assume ITEM.base_unit; show warning on save.
- Partial issues from batches respect expiry & FIFO when available.

6. Retroactive Recalculation (Backfill)

When a new GRN or catalog price is added for a past date range:
- Recompute costs for affected usage lines (that were Estimated).
- Upgrade confidence & badges automatically.
- Keep an **audit trail** of previous cost vs. new cost (delta), timestamp, and reason (Backfill).
- Option to lock closed accounting periods to prevent changes.

7. UI/UX Additions

- On usage forms (Feeding/Treatment): show a **Cost pill** with badge and a tooltip explaining the source (e.g., “LKP from 2025■02■10: 1.20 $/kg”).
- In reports: toggle to **“Hide Low Confidence”**.
- **Resolve Suggestions** panel: list zero■cost or low■confidence items with quick actions (create catalog item, set reference price, link to purchase).
- **Inline add**: create ITEM from usage line without leaving the page.

8. API/Service Behavior

- Usage POST accepts either item_id or free■text name + unit.
- Costing service resolves cost using hierarchy; stores:
cost_value, cost_currency, cost_source
(BATCH|LKP|CATALOG|DERIVED|FUZZY|REFERENCE|ZERO), confidence_level.
- Recalc endpoint to reprocess a date range or item after backfill.

9. Example Scenarios

A) Small farm logs “barley 50 kg” feeding, no inventory set up:
- System matches “barley” to ITEM Barley (0.94 similarity) -> Catalog default 0.35 $/kg -> Estimated (High).

B) Antibiotic issued with batch in stock:
- FIFO selects batch #A123 at 4.80 $/vial -> Actual.

C) Milk replacer logged as free text “Prima■Milk 2kg” with no match:
- Zero placeholder cost -> Estimated (Low); appears in Resolve Suggestions to add ITEM or set price.

D) Later, GRN for “Prima■Milk” added with price 6.00 $/kg (dated before usage):

- Recalc upgrades past entries to Estimated (High) via LKP, or Actual if batch details allow.

10. Reporting & Accounting

- All cost rollups include **confidence breakdown**.
- Per■ewe/per■lamb profitability shows Actual vs Estimated portions.
- Optional monthly **revaluation** to shift Estimated to Actual once data is complete.
- Exports include cost_source and confidence_level fields for transparency.

11. Safeguards & Alerts

- Warn on issuing expired batches; block if policy requires.
- Low stock & upcoming expiries (as before).
- High cost variance (>20% vs last month) flagged for review.
- Name collision: if fuzzy match could map to multiple items, require user confirmation.

Sales & Pedigrees Module - Detailed Specification (with Milk Sales)

Overview

This document defines the Sales & Pedigrees module for the farm management MVP.
It covers animal sales, milk sales, pedigree/genealogy tracking, reports, analytics,
and integrations with other modules (Health, Inventory, Milking, Feeding).

1. Animal Sales

Sale Types:
- Live Sale: ewe, ram, lamb sold to buyer.
- Slaughter Sale: carcass weight & price recorded.
- Culling: removal due to age/health; may be sold or discarded.

Data Fields:
- sale_id, animal_id, buyer_name, buyer_contact, sale_date, sale_type,
weight_kg, price_total, price_per_kg, reason, notes.

Workflow:
- On sale -> animal status = SOLD.
- Withdrawal check prevents sale if ewe under drug withdrawal.
- Invoice/receipt generation optional.

2. Milk Sales

Sale Types:
- Raw milk: daily/weekly sale in liters × price/liter.
- Processed milk (optional future extension): cheese, yogurt.

Data Fields:
- milk_sale_id, date, buyer_name, buyer_contact,
volume_liters, price_per_liter, total_price,
batch_id (optional), notes.

Workflow:
- Milk collected -> added to MILK_USAGE pool.
- Milk Sale recorded -> deducts volume from pool.
- If ewe/batch under withdrawal -> batch flagged "Not for Sale", milk can only be discarded.

3. Pedigree / Genealogy

- Lamb birth links lamb -> dam_id (mother) and optionally sire_id (ram).
- Supports twins/triplets in birth record.
- Pedigree tree view (multi-generation).
- Inbreeding checks (shared ancestors).
- Performance tracked by bloodline (milk yield, lambing rate, survival).

4. Reports & Analytics

Animal Sales:
- Revenue per type, average price/kg, profitability per animal.

Milk Sales:
- Revenue per day/week/month.
- Liters sold vs discarded (due to withdrawal).
- Average price per liter.
- Milk revenue vs feeding costs.

Pedigree:
- Sire contribution (offspring count, performance).
- Dam contribution (lambing rate, survival, milk yield).
- Inbreeding risk %.

5. UI/UX

Sales Dashboard:
- Tabs: Animal Sales | Milk Sales.
- Charts: monthly revenue breakdown (animals vs milk).

Milk Sales Form:
- Fields: date, buyer, liters, price/liter, total auto-calculated.
- Deducts from available milk pool.

Animal Profile > Pedigree Tab:
- Sire/Dam links.
- Offspring list with birth dates.
- Pedigree tree visualization.

6. Integration

- Milking: supplies collected milk volumes for sale.
- Inventory/Costs: calculates cost per liter (feed, health, supplements).
- Health: blocks sales if under withdrawal.
- Feeding/Weight: animal weight history feeds sale pricing (live sale).
- Finance: sales revenue (animals + milk) rolls into profitability reports.

Reports & Dashboards - Detailed Specification

Overview

This document defines the Reports & Dashboards module for the farm management MVP.
It consolidates KPIs and analytics across Breeding, Health/Treatments, Feeding & Weight (incl. lamb milk-feeding),
Milking, Sales & Pedigrees (incl. milk sales), and Inventory & Costs (always-on best-effort costing).

1. Dashboard Structure

Home Dashboard (owner/manager view):
- At-a-glance KPIs (last 30 days / YTD toggles).
- Alerts & tasks (overdue checks, low stock, withdrawal, due windows, doses today).
- Quick actions: Start Cycle, Record Treatment, Add Weight, Log Milk, Record Sale.

Tabs/Sections:
- Breeding
- Health
- Feeding & Weight
- Milking
- Sales
- Inventory & Costs

2. Core KPIs by Area

Breeding:
- Active cycles, Pregnant ewes, Due windows (this week/month).
- Conception rates: INS1 vs INS2.
- Avg days open; pregnancy check pass rates (CK1/CK2).

Health/Treatments:
- Animals Sick/Recovering.
- Doses due today, overdue doses.
- Under withdrawal (milk/meat) count and days to clear.
- Treatment success rate, cost of treatments.

Feeding & Weight (incl. lamb milk-feeding):
- Average Daily Gain (ADG) by group.
- Feed conversion ratio (FCR).
- % lambs on manufactured milk; cost per artificially fed lamb.
- Underweight/overweight alerts.

Milking:
- Daily liters (trend), 7/30-day totals.
- Milk discarded due to withdrawal (%).
- Avg liters/ewe/day, lactation length distribution.

Sales & Pedigrees:
- Animal sales revenue & avg price/kg.
- Milk sales revenue & liters sold.
- Profitability per animal sold (sale minus lifetime costs).

- Bloodline performance (top sires/dams).

Inventory & Costs:
- Stock on hand by category.
- Batches expiring soon.
- Cost confidence mix (Actual vs Estimated).
- Monthly valuation and cost rollups.

3. Visualizations

- Line charts: milk yield, ADG trends, revenue over time.
- Bar charts: conception rates, treatment costs by disease/drug, sales by type.
- Pie/donut: cost confidence distribution, sales revenue split (animals vs milk).
- Heatmaps: due windows calendar, weight deviations by pen.
- Tables with conditional formatting for alerts.

4. Alerts & Tasks

- Overdue pregnancy checks, doses, weight checks.
- Ewes in lambing window today.
- Batches expiring (30/15/7 days).
- Under-withdrawal milk/animals flagged.
- Low stock vs reorder points.

All alerts are actionable: click to navigate to the relevant entry form or list.

5. Filters & Segmentation

- Time window: Today, 7d, 30d, Custom.
- Group/pen, animal type (ewe/ram/lamb), health status.
- Cost confidence filter (Actual only, Include Estimated High/Medium/Low).
- Buyer/supplier filters for sales/purchases.

6. Reports Library (MVP)

- Breeding: Conception report (INS1 vs INS2), Due windows schedule, Days open.
- Health: Treatment ledger, Withdrawal compliance, Doses due/overdue.
- Feeding/Weight: ADG by group, FCR by plan, Underweight list.
- Milking: Yield by ewe, Daily milk summary, Discarded milk.
- Sales: Animal sales ledger, Milk sales summary, Profitability per animal.
- Inventory: Valuation (Avg/FIFO), Expiry report, Usage by module.

All reports exportable to CSV/PDF.

7. Data & Performance

- Reporting DB (read-optimized) with nightly ETL from OLTP (or streaming CDC later).
- Precomputed aggregates for heavy KPIs (ADG, FCR, conception rates).
- Time-series indexes and partitions for yield/weights.
- Multi-tenant data isolation by farm_id.
- Idempotent ETL with audit for corrections/backfills.

8. API & Permissions

- /dashboard KPIs endpoints per area.
- /reports endpoints with query params (date range, filters).
- Role-based access: Owner/Manager (all), Vet (Health), Feeder (Feeding), Milk Operator (Milking), Sales (Sales), Storekeeper (Inventory).
- Respect withdrawal and privacy flags in data returned.

9. UI/UX Notes

- Cards for KPIs with color badges; click-through to detail.
- Save custom filter presets.
- Print-friendly reports.
- Mobile-friendly summaries (today's tasks, quick scan to action via RFID/QR).
- Empty states with tips to start logging data.

Lamb Selection for Replacement Ewes - Detailed Specification (Bloodline-Based)

Overview

This document defines the flow for selecting lambs to keep as replacement ewes, based on bloodline performance
(sire/dam KPIs), lamb growth/health, and farm thresholds. It includes data model fields, scoring algorithms,
UX, and reports.

1. Objectives

- Identify top ewe-lambs to retain for breeding using objective, data-driven criteria.
- Balance bloodline performance (sire/dam) with individual lamb growth and health.
- Provide transparent scoring with override capability for farmer expertise.

2. Inputs

Pedigree:
- sire_id, dam_id, ancestry (for inbreeding checks).

Bloodline KPIs (rolling 12–24 months, configurable):
- Dam: lambing rate, litter size, weaning rate, milk yield proxy, offspring survival.
- Sire: conception rate, offspring growth (ADG), offspring survival.

Lamb data:
- birth_date, birth_weight, weaning_weight, ADG to weaning, health events count/severity, twin/triplet flag.

3. Workflow

1) At lambing: register lamb with sire/dam.
2) At weaning (e.g., 60–90 days) or a chosen checkpoint:
- System computes selection score per lamb.
- Color recommendation: Recommended / Neutral / Not Recommended.
3) Farmer reviews on the Selection Screen; may override with reason.
4) Decision stored: KEEP_AS_EWE | SELL | CULL.

4. Data Model

LAMB_SELECTION:
- id, lamb_id, evaluation_date, score_total, recommendation (RECOMMENDED|NEUTRAL|NOT_RECOMMENDED),
decision (KEEP_AS_EWE|SELL|CULL|PENDING), criteria_json, operator, notes.

BLOODLINE_SCORE (aggregated KPIs by parent):
- id, parent_type (SIRE|DAM), parent_id, period_start, period_end,
conception_rate, avg_offspring_adg, weaning_rate, milk_yield_proxy, survival_rate, index_value.

CONSTRAINTS:
- Check inbreeding risk using ancestry graph before final approval; warn if relatedness > threshold.

5. Scoring Algorithm (MVP)

Total 100 points (default weights; farm-configurable):

A) Growth (0–40):
- ADG_to_weaning vs herd mean (z-score or % diff). Map to 0–40.
- Bonus if birth_weight within optimal band; penalty if underweight.

B) Dam Performance (0–30):
- Weaning rate & offspring survival (last N litters).
- Milk yield proxy (e.g., average lamb weights from this dam).
- Litter size normalized (twins ok; triplets slight penalty if poor survival).

C) Sire Performance (0–20):
- Conception rate of sire’s matings.
- Offspring ADG & survival.

D) Health (0–10):
- Fewer treatments & no severe illnesses -> higher score.
- Penalize chronic issues or >2 treatments pre-weaning.

Thresholds:
- >= 70 -> Recommended
- 50–69 -> Neutral
- < 50 -> Not Recommended
Override allowed with reason; audit retains original score.

6. Formula Options

Option 1 (Percentile-based):
- Convert each KPI to percentile within herd/year, then apply weights.

Option 2 (Z-score normalization):
- z = (value - mean) / std; clamp to [-2, +2]; scale to weighted points.

Option 3 (Rules + Bonuses):
- If dam_weaning_rate >= 90% add +5; if sire_offspring_ADG >= herd+10% add +5; etc.
Use whichever method is simpler to maintain; keep coefficients in a settings table.

7. UI/UX

Selection Screen (season/pen filter):
- Table columns: Lamb ID, Sire, Dam, ADG, Health events, Dam index, Sire index, Score, Recommendation, Decision.
- Row details: criteria breakdown with tooltips; inbreeding warning chip.
- Bulk actions: Accept all Recommended; move Neutral to Review; mark Not Recommended as Sell.
- Override modal requires reason; options to tag “keeper” or “sale lot”.

Animal Profile:
- Selection history card; show final decision & date.
- Link to bloodline metrics of sire/dam.

Pedigree View:
- Highlight lambs selected as future ewes to visualize bloodline continuity.

8. Reports & KPIs

- Replacement Rate: % of ewe-lambs kept vs born.
- Bloodline Keep Rate: % of lambs kept per sire/dam.
- Performance of Selected Ewes: later track milk yield, fertility, lamb survival vs non-selected.
- Inbreeding incidents avoided (alerts vs overrides).

9. Integrations & Rules

- Sales: lambs not kept flow into sale pipeline.
- Health: penalize selection for chronic health issues.
- Feeding/Weight: ADG inputs; ensure recent weights exist (remind to weigh at 30/60/90 days).
- Breeding: block selection if relatedness > threshold unless override with reason.

Insights & Causality - Correlating Actions with Farm Outputs

Overview

Goal: make the system smart enough to **link actions** (e.g., shearing, supplier change, ration change, vaccination campaign)
to **outputs** (e.g., sickness incidence, weight gain, milk yield, conception rate, mortality). Provide evidence,
confidence levels, and clear explanations while remaining lightweight and mobile-first.

1. Use Cases

- Shearing campaign → spike in respiratory illness 5–10 days later.
- Supplier change (feed) → ADG drops by 8–15% within 2 weeks.
- New ration formula → milk yield increases by ~10% in 7 days.
- New insemination protocol → conception rate uplift.
- Heat wave or pen change → stress markers (weight loss, sickness).

2. Data Requirements (MVP)

- Timestamped **Action Events** (who, what, when, affected group/pen).
- Clean **Outcome Metrics** time series (ADG, milk/day, sickness incidents, conception rate).
- Key **Covariates**: age class, parity, pen, season, pregnancy status; optional weather import.
- Consistent **grouping** (pens/batches) to create comparable control groups when possible.

3. Data Model Additions

ACTION_EVENT:
- id, farm_id, event_type (SHEARING|SUPPLIER_CHANGE|RATION_CHANGE|PROTOCOL_CHANGE|MOVE|OTHER), start_date, end_date (nullable), scope (GROUP|FARM), target_ids (group/pen ids), notes.

EXPERIMENT (optional tagging for planned changes):
- id, name, start_date, end_date, target_groups, control_groups, primary_metric (ADG|MILK|SICKNESS|CR|OTHER),
hypothesis, owner, status.

METRIC_SNAPSHOT (precomputed daily/weekly metrics per group/farm):
- id, date, group_id, metrics_json (adg, milk, sickness_rate, conception_rate, mortality, etc.).

4. Methods (MVP -> Advanced)

MVP (fast, explainable):
- **Pre/Post Window Uplift**: Compare metric in [t-14, t-1] vs [t+1, t+14]. Report % change and a simple t-test or nonparametric test.
- **Difference-in-Differences**: If control group exists, compare (treated pre→post) vs (control pre→post).
- **Change-Point Detection (CUSUM/BOCPD-lite)**: Flag significant shifts near action date.

- **Lagged Cross-Correlation**: Scan lags (1–21 days) to see best alignment (e.g., supplier change lag=10 → ADG drop).

Advanced (opt-in later):
- **Propensity Score Matching** to build better controls.
- **Causal DAGs** with farm-defined confounders (season, parity, disease).
- **Bayesian Structural Time Series** for impact estimation.

5. Confidence & Badges

Each insight is labeled:
- **High**: DiD with clear effect (p<0.05) or strong change-point + consistent lags.
- **Medium**: Pre/Post with p<0.1; moderate effect size; some confounders present.
- **Low**: Correlation only; data sparse; multiple overlapping actions.
Show effect size (% change), window, lag, and sample sizes. Include caveats.

6. Alerts & Stories

- **Proactive Alert**: “Possible impact detected: ADG ↓ 12% after feed supplier switch on 2025■08■01 (lag 9d), confidence: Medium.”
- **Insight Card**: narrative summary, chart thumbnail, ‘View Analysis’ button.
- **Mute/Confirm**: user can confirm the insight (becomes a labeled event) or mute/ignore to improve the model.

7. UI/UX

Insights Tab:
- Timeline of **Action Events**; selecting one shows impact analysis (pre/post charts, DiD table, lag plot).
- Filters by event type, pen/group, date range, metric.
- Export insight to PDF/CSV for co-ops/government.

Action Creation:
- When logging big actions (shearing, supplier, ration), app suggests tagging as **Experiment** with target metrics.

8. Computation & Sync

- Computation runs **server-side** nightly (or on demand); mobile shows results, works offline with cached insights.
- Uses **METRIC_SNAPSHOT** to avoid heavy recalculation (ETL aggregates daily).
- Keeps an **audit trail** of assumptions, windows, and model version.

9. Safeguards & Ethics

- Avoid over-claiming causality: explicitly distinguish **correlation vs. inferred impact**.
- Show confounders considered (season, outbreaks). Allow user to add context notes.
- Respect privacy and roles; insights scoped per farm. No cross-farm data mixing unless owner requests benchmarking.

10. Benchmarking (Optional, Privacy-Respecting)

- If a farmer opts in to **anonymous benchmarking**, compute percentile comparisons (feed cost/kg gain, sickness rates)
against aggregated peers in the region. No farm-identifying data shared.

11. APIs & Jobs (MVP)

APIs:
- POST /actions (create action event)
- GET /insights?farm_id&date;_range&type;&metric;
- GET /insights/{id} (details, charts, diagnostics)
- POST /experiments (optional planned change tagging)

Jobs:
- nightly_aggregate_metrics → populate METRIC_SNAPSHOT
- impact_detection_runner → run window uplift, DiD, change-point and lag scans
- insights_notifier → push notifications for new/changed insights

Example Insight	Action	Metric	Effect	Lag	Confidence
ADG drop after supplier change	Supplier change (2025-08-01)	ADG	-12%	9 days	Medium
Sickness spike post-shearing	Shearing (2025-03-10)	Sickness rate	+8 cases / 100 ewes	7 days	High