SMART RANCH MANAGEMENT SYSTEM

For Kenyan Smallholder Farmers

PROBLEM: 78% of Kenya's agriculture depends on smallholders struggling with:

- Manual record-keeping → poor tracking (health , weight , feeding, finances, production[milk ,meat, manure],breeding ,expenses (PnL)
- No health monitoring → delayed interventions
- No financial oversight → low profitability

Consequences:

Inaccurate expense tracking

Poor health monitoring

No data-driven decisions making

SOLUTION: Mobile app using Flutter & FastAPI

Eugine Onyango Odhiambo | EBI/61390/22

COSC 482 | Supervisor: Bernard Ongera Osero

OBJECTIVES & CORE FEATURES

GENERAL OBJECTIVE:

Design scalable mobile system to enhance efficiency, accuracy & profitability

SPECIFIC OBJECTIVES:

1. Analyze Kenyan-specific requirements
2. Build async FastAPI backend + PostgreSQL HTAP
3. Develop Flutter cross-platform app
4. Create future-ready architecture
5. Rigorous testing & evaluation

CORE FEATURES:

- Inventory (Individual + Batch tracking)
- Health monitoring with audit trails
- Feed consumption & FCR calculation
- Financial expense tracking

SYSTEM REQUIREMENTS SUMMARY

FUNCTIONAL:

1. Inventory Management (individuals + batches)
2. Expense Tracking (categorized & linked)
3. Feed Consumption (FCR calculation)
4. Health Monitoring (complete audit trail)
5. Reporting (financial + herd analytics)

NON-FUNCTIONAL:

- Usability: Simple for farmers with moderate digital literacy
- Performance: FastAPI async for low latency
- Security: Authentication + data encryption
- Scalability: Ready for IoT/ML integration

TECHNICAL INNOVATION

3-TIER ARCHITECTURE:

1. *PRESENTATION:* Flutter Mobile App
   - Cross-platform (Android/iOS)
   - Simple, intuitive interface

2. *APPLICATION:* FastAPI Backend
   - Async-first for high performance
   - Business logic & authentication

3. *DATA:* PostgreSQL with HTAP
   - Hybrid Transactional/Analytical Processing
   - Individual tracking + batch analytics simultaneously

KEY INNOVATION: Database handles real-time & analytical queries together

DEVELOPMENT APPROACH

METHODOLOGY: Agile-Scrum (2-week sprints)

6-MONTH TIMELINE:

MONTH 1-2: Foundation

- Requirements & database design
- Core API development

MONTH 3-4: Core Features

- Flutter app development
- Inventory & finance modules

MONTH 5: Testing & Polish

- User acceptance testing
- Performance optimization

MONTH 6: Delivery

- Final documentation & deployment

SCALABLE & FUTURE-READY

PHASE I (Current):

- Complete livestock management system
- Scalable 3-tier architecture

READY FOR PHASE II:

- IoT INTEGRATION: Animal sensors, GPS tracking
- MACHINE LEARNING: Disease prediction, feed optimization
- ADVANCED ANALYTICS: Geo-spatial, predictive models

DESIGN PRINCIPLE: Build foundation today, integrate AI/IoT tomorrow

WHY THIS PROJECT MATTERS

FOR FARMERS:
- Data-driven decisions → higher profits
- Better animal health → improved welfare
- Production and expense traction

FOR ACADEMIA:
- Practical software engineering demonstration
- Contribution to Kenya's agricultural digitization

FOR AGRI-TECH:
- African-context reference model(Aligns with Agriculture 5.0)
- Addresses scalability gaps in existing solutions

CAREER VALUE: Mastery of Flutter, FastAPI, PostgreSQL - highly marketable skills

Expected Deliverables

- **Software:**
  - Working mobile app (Flutter)
  - High-performance API (FastAPI)
  - Optimized database (PostgreSQL HTAP)
- **Academic:**
  - Full documentation
  - Source code with comments
  - Final report & presentation

CONCLUSION

SUMMARY:

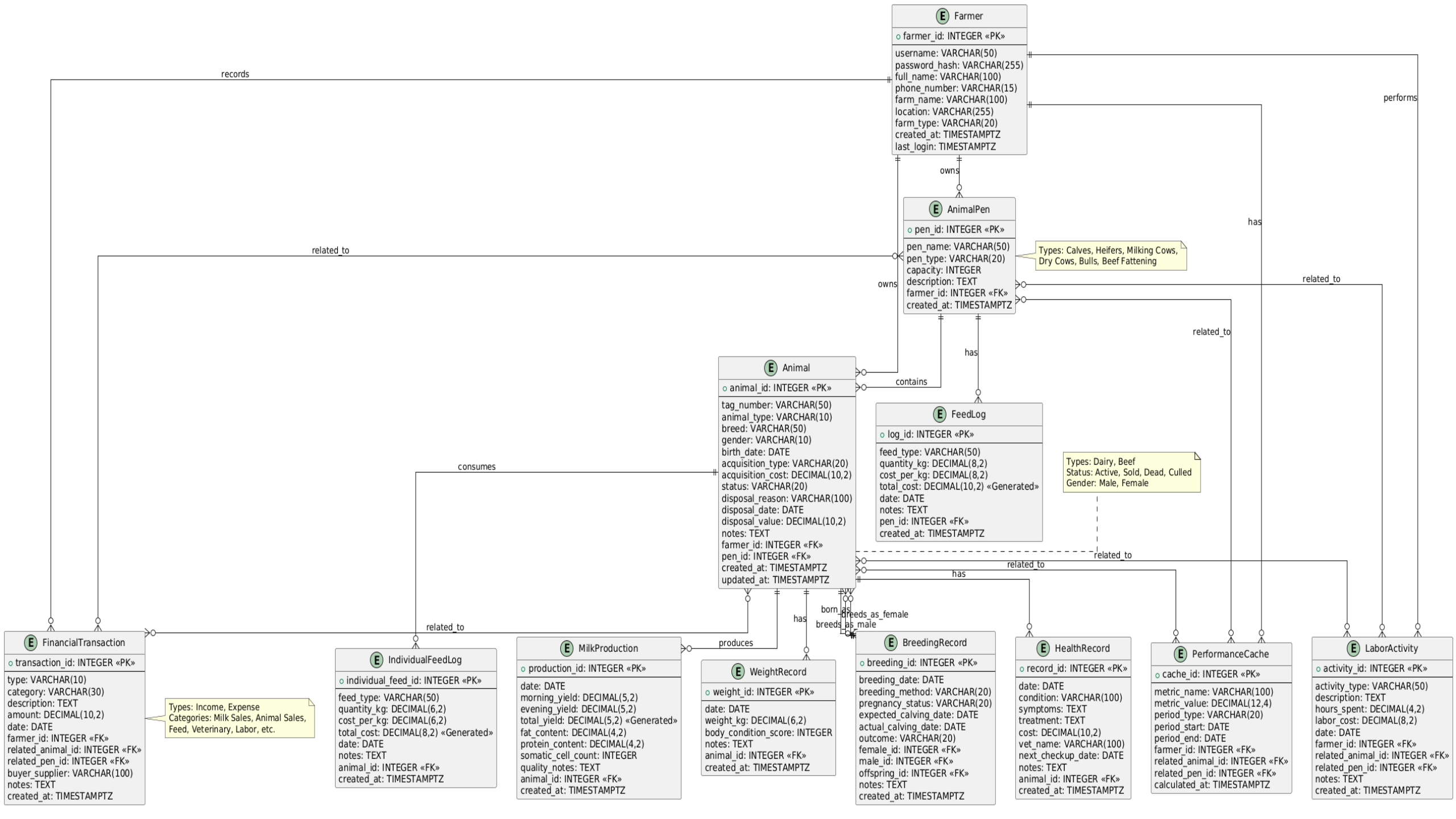
- Mobile solution for real Kenyan farming challenges
- Hybrid database for granular + aggregated tracking
- Future-ready architecture for IoT/AI integration

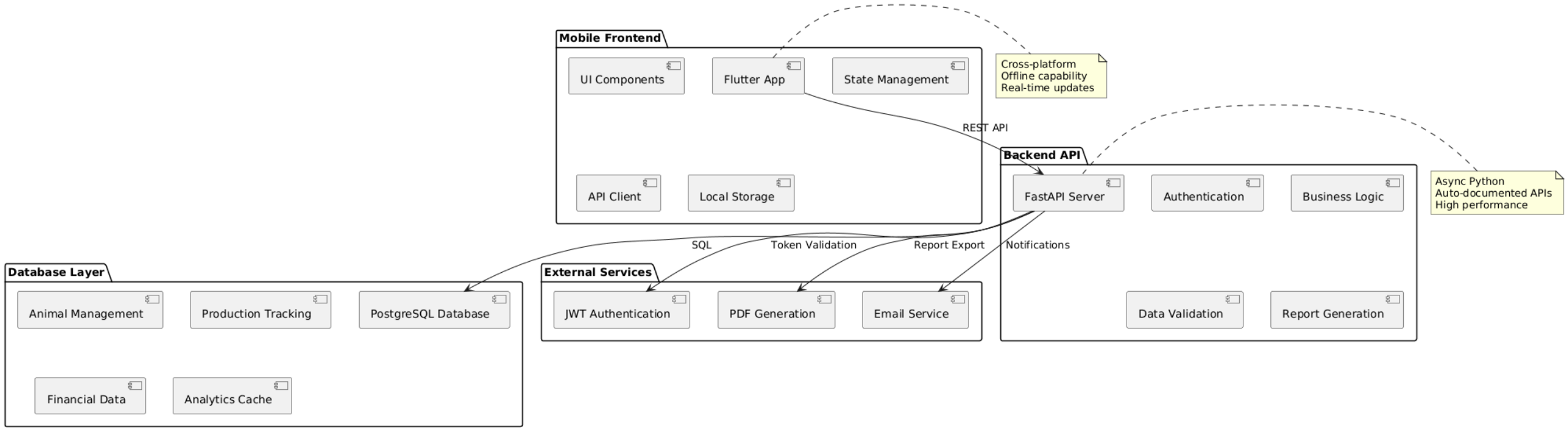
KEY ACHIEVEMENTS:

1. Solves critical record-keeping problem
2. Modern, high-performance tech stack
3. Designed for African smallholder context
4. Pathway to precision livestock farming

THANK YOU

Questions & Discussion & Answers


Farmer
farmer_id: INTEGER «PK»
username: VARCHAR(50)
password_hash: VARCHAR(255)
full_name: VARCHAR(100)
phone_number: VARCHAR(15)
farm_name: VARCHAR(100)
location: VARCHAR(255)
farm_type: VARCHAR(20)
created_at: TIMESTAMPTZ
last_login: TIMESTAMPTZ
records
performs
owns
AnimalPen
pen_id: INTEGER «PK»
pen_name: VARCHAR(50)
pen_type: VARCHAR(20)
capacity: INTEGER
description: TEXT
farmer_id: INTEGER «FK»
created_at: TIMESTAMPTZ
Types: Calves, Heifers, Milking Cows, Dry Cows, Bulls, Beef Fattening
related_to
has
owns
Animal
animal_id: INTEGER «PK»
tag_number: VARCHAR(50)
animal_type: VARCHAR(10)
breed: VARCHAR(50)
gender: VARCHAR(10)
birth_date: DATE
acquisition_type: VARCHAR(20)
acquisition_cost: DECIMAL(10,2)
status: VARCHAR(20)
disposal_reason: VARCHAR(100)
disposal_date: DATE
disposal_value: DECIMAL(10,2)
notes: TEXT
farmer_id: INTEGER «FK»
pen_id: INTEGER «FK»
created_at: TIMESTAMPTZ
updated_at: TIMESTAMPTZ
contains
has
FeedLog
log_id: INTEGER «PK»
feed_type: VARCHAR(50)
quantity_kg: DECIMAL(8,2)
cost_per_kg: DECIMAL(8,2)
total_cost: DECIMAL(10,2) «Generated»
date: DATE
notes: TEXT
pen_id: INTEGER «FK»
created_at: TIMESTAMPTZ
Types: Dairy, Beef
Status: Active, Sold, Dead, Culled
Gender: Male, Female
related_to
consumes
related_to
has
born_as
breeds_as_female
breeds_as_male
has
produces
related_to
FinancialTransaction
transaction_id: INTEGER «PK»
type: VARCHAR(10)
category: VARCHAR(30)
description: TEXT
amount: DECIMAL(10,2)
date: DATE
farmer_id: INTEGER «FK»
related_animal_id: INTEGER «FK»
related_pen_id: INTEGER «FK»
buyer_supplier: VARCHAR(100)
notes: TEXT
created_at: TIMESTAMPTZ
Types: Income, Expense
Categories: Milk Sales, Animal Sales, Feed, Veterinary, Labor, etc.
IndividualFeedLog
individual_feed_id: INTEGER «PK»
feed_type: VARCHAR(50)
quantity_kg: DECIMAL(6,2)
cost_per_kg: DECIMAL(6,2)
total_cost: DECIMAL(8,2) «Generated»
date: DATE
notes: TEXT
animal_id: INTEGER «FK»
created_at: TIMESTAMPTZ
MilkProduction
production_id: INTEGER «PK»
date: DATE
morning_yield: DECIMAL(5,2)
evening_yield: DECIMAL(5,2)
total_yield: DECIMAL(5,2) «Generated»
fat_content: DECIMAL(4,2)
protein_content: DECIMAL(4,2)
somatic_cell_count: INTEGER
quality_notes: TEXT
animal_id: INTEGER «FK»
created_at: TIMESTAMPTZ
WeightRecord
weight_id: INTEGER «PK»
date: DATE
weight_kg: DECIMAL(6,2)
body_condition_score: INTEGER
notes: TEXT
animal_id: INTEGER «FK»
created_at: TIMESTAMPTZ
BreedingRecord
breeding_id: INTEGER «PK»
breeding_date: DATE
breeding_method: VARCHAR(20)
pregnancy_status: VARCHAR(20)
expected_calving_date: DATE
actual_calving_date: DATE
outcome: VARCHAR(20)
female_id: INTEGER «FK»
male_id: INTEGER «FK»
offspring_id: INTEGER «FK»
notes: TEXT
created_at: TIMESTAMPTZ
HealthRecord
record_id: INTEGER «PK»
date: DATE
condition: VARCHAR(100)
symptoms: TEXT
treatment: TEXT
cost: DECIMAL(10,2)
vet_name: VARCHAR(100)
next_checkup_date: DATE
notes: TEXT
animal_id: INTEGER «FK»
created_at: TIMESTAMPTZ
PerformanceCache
cache_id: INTEGER «PK»
metric_name: VARCHAR(100)
metric_value: DECIMAL(12,4)
period_type: VARCHAR(20)
period_start: DATE
period_end: DATE
farmer_id: INTEGER «FK»
related_animal_id: INTEGER «FK»
related_pen_id: INTEGER «FK»
calculated_at: TIMESTAMPTZ
LaborActivity
activity_id: INTEGER «PK»
activity_type: VARCHAR(50)
description: TEXT
hours_spent: DECIMAL(4,2)
labor_cost: DECIMAL(8,2)
date: DATE
farmer_id: INTEGER «FK»
related_animal_id: INTEGER «FK»
related_pen_id: INTEGER «FK»
notes: TEXT
created_at: TIMESTAMPTZ


Mobile Frontend
UI Components
Flutter App
State Management
API Client
Local Storage
Cross-platform
Offline capability
Real-time updates
REST API
Backend API
FastAPI Server
Authentication
Business Logic
Data Validation
Report Generation
Async Python
Auto-documented APIs
High performance
SQL
Token Validation
Report Export
Notifications
Database Layer
Animal Management
Production Tracking
PostgreSQL Database
Financial Data
Analytics Cache
External Services
JWT Authentication
PDF Generation
Email Service