University of Puerto Rico at Mayagüez
Department of Computer Science and Engineering

Capstone: Progress Report

Group A

CIIC/INSO4151-001#

Dr. Wilson Gallegos
Axel J. Perez Rodriguez
Edgar J. Sanabria Soto
Eduardo Martinez Calvo
Mark A. Alvarez Nieves

May 12, 2025

Table of Contents

Introduction

Livestock farming is vital to global food production, yet farmers face persistent challenges in animal health monitoring, theft prevention, and resource management. Traditional methods, such as manual health assessments and GPS collars, are often unreliable, costly, or impractical for small and medium-scale farms, leading to economic losses, reduced productivity, and compromised animal welfare. There is an increasing need for cost-effective, technology-driven solutions to improve livestock management.

One of the biggest concerns for farmers is animal theft, which can result in significant financial losses. While collar-based GPS tracking has been used, these devices are easily removed, making them ineffective for theft prevention. Additionally, health monitoring is challenging, as diseases like parasite-induced anemia often go undetected until symptoms become severe. The FAMACHA system is a useful tool for assessing anemia in small ruminants, but its manual nature limits scalability and early intervention [3]. Similarly, tracking water intake is essential for livestock health, yet farmers lack affordable automated solutions to detect drinking behavior changes that may indicate health issues [2].

Advancements in IoT, artificial intelligence (AI), and sensor-based monitoring offer promising solutions. Machine learning models can help predict and detect health issues, allowing for earlier diagnosis and reduced veterinary costs [1]. RFID-based systems have been successfully used to track feeding and drinking behaviors, offering valuable health insights [2]. GPS tracking equipment will be implemented to help to prevent theft, loss and to always know where the sheeps are. In the future, a bolus-based GPS tracking system is going to be implemented, which involves implanting GPS devices inside the animal, providing a more secure and tamper-resistant alternative to collars.

Beyond academic research, consumer assessments show that cost and ease of use are the most critical factors for farmers when adopting new livestock monitoring technologies [4]. Many existing solutions are either too expensive or require specialized technical knowledge, making them inaccessible for small-scale farmers. As a result, there is a clear demand for an integrated, affordable system that provides real-time data on animal location, health, and water consumption without adding unnecessary complexity.

This project proposes a comprehensive livestock monitoring system that integrates GPS tracking, automated FAMACHA-based health assessments [6], and RFID-enabled water intake monitoring. Unlike existing solutions that focus on one aspect at a time, this system combines these functionalities into a single, easy-to-use platform. By leveraging IoT connectivity, AI-powered analysis, and real-time alerts, the proposed solution will enhance livestock security, improve monitoring, and optimize farm operations.

Problem Statement

Farmers and livestock owners often face significant problems such as animal theft, health problems, and inefficiencies in monitoring food and water intake. Current methods, such as tracking collars or manual health monitoring, are either unreliable or too expensive for many small-scale livestock owners. Moreover, accurately monitoring livestock health and ensuring sufficient access to food and water remains an ongoing challenge. Addressing these issues is essential to improving livestock management and overall farm productivity.

Target Domain and Client

This project focuses on the agricultural sector, specifically small to medium-scale livestock owners who need affordable and practical solutions. The primary users include farmers in rural and semi-urban areas who may lack access to advanced technological tools but require reliable systems to track and care for their livestock effectively.

Proposed Innovation

Our proposed solution integrates multiple technologies into a unified livestock management system. It includes:

- A GPS-enabled tracking device embedded securely into animals to prevent theft or loss.
- A FAMACHA-based health monitoring and annotation function to identify and manage health risks effectively [6].
- A water and food monitoring system to ensure each animal receives adequate nutrition and hydration.
- A machine learning model to estimate the age of sheep by analyzing images of their teeth.
- Sheep weight estimation from side view photo

While similar technologies exist independently, there are currently no solutions that integrate all four functionalities into a single, cohesive system. What makes our proposal innovative is the custom integration of these components, paired with our own machine learning model for health prediction and a mobile platform designed for ease of use and real-time alerts.

These features are powered by Internet of Things (IoT) technologies, enabling real-time monitoring, automation, and remote access through a connected network. This approach provides a comprehensive, affordable, and practical solution for modern livestock management.

Project Objectives

Project General Goal

> Develop an integrated livestock monitoring system for sheep that combines GPS tracking, age classification via machine learning, and RFID-based water intake measurement, with supporting training tools to ensure successful adoption by farmers.

Objective 1: GPS Tracking for Livestock Location 90

> Implement a GPS tracking system using a real time device that provides real-time location and movement data, achieving a minimum location accuracy of 90% across at least on 1 tracked animal.

Objective 2: Machine Learning-Based Teeth Classification 98

> Develop and deploy a machine learning model capable of estimating sheep age using teeth images, achieving an accuracy of at least 98% using a dataset of 30 or more labeled images.

Objective 3: RFID-Based Water Intake Monitoring 98

> Install and test an RFID-based system that records individual sheep's water intake at automated drinking stations, ensuring a minimum tag detection rate of 95% under regular use conditions.

Objective 4: User Training and System Adoption 80

> Design and deliver a set of training sessions and digital tutorials to farmers, ensuring that at least 80% of participants demonstrate proper use of the system prior to deployment, as verified through a usability assessment.

Objective 5: Sheep Weight Estimation 80

> To develop a photo-based system that accurately estimates sheep weight by detecting body dimensions using computer vision and scaling measurements with a known-size collar.

Solution Approach

We propose an integrated livestock management system that combines GPS tracking, health monitoring through the FAMACHA system, age estimation and water consumption tracking for sheep. The solution includes:

- **GPS tracking system with Tracki:** Each sheep will be equipped with a Tracki real-time global GPS tracker to monitor its location continuously. This allows farmers to track sheep movement, monitor grazing patterns, and ensure that no animal is lost or strays from the herd**.**
- **Machine learning classification model for age estimation**: A machine learning model will be developed to estimate the age of sheep by analyzing images of their teeth. Users will capture a photo of the sheep's lower front teeth, and the system will classify the sheep into an age group based on tooth eruption patterns, wear, and development. Traditionally, age estimation in sheep is performed manually by inspecting the eruption and wear of their incisor teeth. This method relies on identifying which permanent teeth have emerged and assessing the degree of wear, typically effective up to around four years of age. However, this approach is subjective, depends on the skill of the observer, and requires physically handling the animal. By automating this process with machine learning, the model provides a non-invasive, consistent, and scalable alternative to traditional age estimation techniques.
- **RFID-based water consumption monitoring**: Each sheep will be tagged with an RFID tag, which will communicate with RFID readers placed at water sources. The system will track individual water intake, allowing farmers to detect any changes in drinking behavior, which could be indicative of health issues. Additionally, a water level sensor will be used to log the total amount of water consumed.
- **Mobile App**: A user-friendly interface for farmers to visualize data, receive alerts, and access reports on animal health and welfare.
- **Sheep weight estimation from photo**: an image-based solution for estimating the weight of a sheep using a single photograph. The estimation process leverages computer vision techniques to detect and analyze the sheep's body dimensions with reference to a known object (the collar) and applies an empirical formula to calculate the weight in kilograms.
    1. **Reference Object: Sheep Collar for Scaling**

       To estimate real-world measurements from a 2D image, a sheep collar of a known physical size is used as a reference object. The collar is uniquely colored, allowing it to be easily identified in the image using OpenCV's `cv2.inRange()` functionality. By calculating the pixel span of the collar in the image and comparing it to its known real-world size, a pixel-to-centimeter scaling factor is derived.

    2. **Sheep Detection and Image Preprocessing**

       The sheep is detected using the **YOLOv8m-seg.pt** model, which provides both bounding boxes and segmentation masks. The bounding box is used to crop the image tightly around the sheep. To prevent background noise from interfering with collar detection, the background of the cropped image is replaced with a solid white background using the segmentation mask provided by YOLO.

    3. **Measurement and Weight Estimation**

Once the collar has been detected and the scaling factor calculated, two key measurements are extracted from the image:

- **Thoracic perimeter** (measured around the chest)
- **Diagonal body length** (from shoulder to tail)

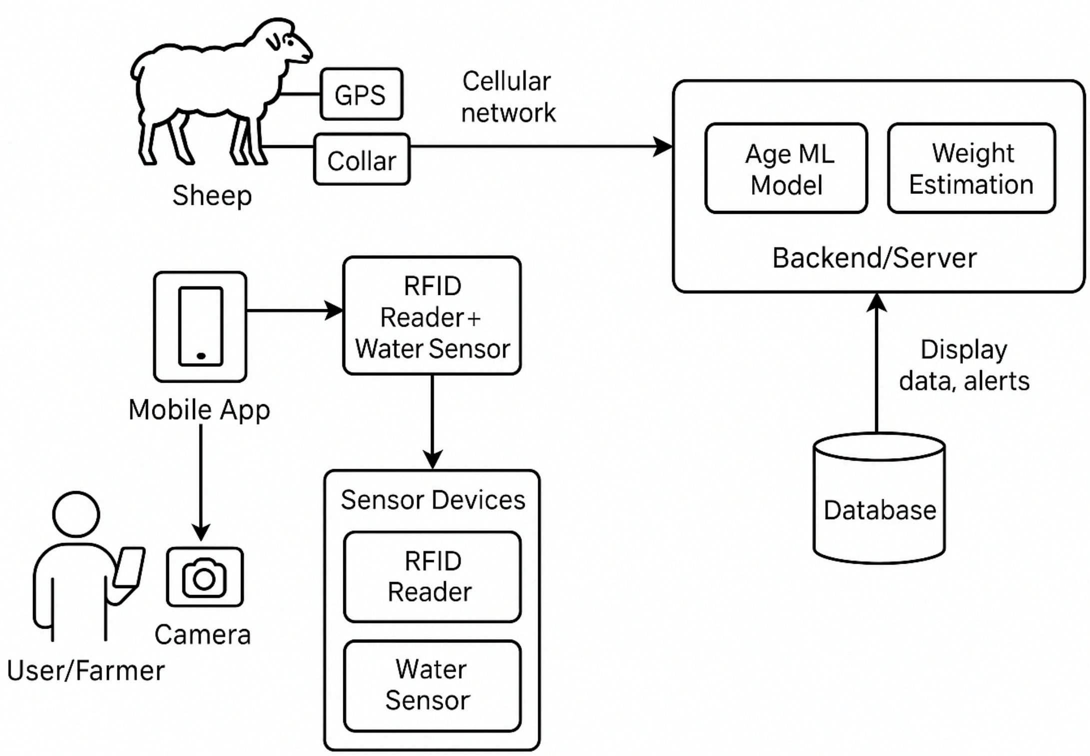
These values are used in the following empirical formula to estimate the sheep's weight:

$$Weight\ (kg) = \frac{(Thoracic\ Perimeter)^2 \times Diagonal\ Body\ Length}{10838}$$

4. **Output**

The final output is the estimated weight of the sheep in **kilograms**, based on the dimensions derived from the image and scaled appropriately using the known size of the collar.

Representative Diagram



Existing Solutions

Current livestock monitoring technologies often focus on a single function, such as GPS tracking or health diagnostics. However, they lack integration across key areas like real-time location, health prediction, and hydration monitoring. Additionally, many rely on external collars or manual assessments, which limit effectiveness and automation.

To date, there are no commercial systems that integrate GPS tracking, teeth-based anemia detection using machine learning, and RFID-based water intake monitoring in a unified solution.

Our proposed system addresses this gap by offering:

- A secure real-time global GPS tracker device for tamper-resistant location tracking.
- Automated health analysis using teeth images and AI models.
- RFID-based hydration tracking, personalized per animal.

This integration allows for real-time, data-driven decisions and early interventions – capabilities not offered by existing tools.

Value Proposition

The proposed solution offers several key advantages:

- **Improved animal welfare:** By monitoring health indicators like parasite levels and water intake, the system may help prevent diseases and ensure that the sheeps receive timely medical care.
- **Increase operational efficiency:** The automated tracking of location, health status, and water intake reduces the need for manual monitoring, saving time and resources for farmers.
- **Minimized losses:** GPS tracking ensures that sheeps do not get lost, which minimizes the financial losses associated with missing livestock.
- **Data-driven decisions:** The machine learning model offers predictive insights that can assist farmers in making informed decisions about flock health and resource management, optimizing farm operations.

Benefits and limitations

Benefits

- **Real-time monitoring**: The system offers continuous, real-time monitoring of sheep health, location, and water intake.
- **Prevention of disease spread**: Early detection of health issues, especially parasites, can prevent more severe outbreaks, improving overall herd health.
- **Resource optimization**: By tracking water consumption, the system can help ensure optimal resource use and alert farmers when animals are not drinking enough water, possibly indicating health issues.

- **Age estimation**: The age estimation functionality offers farmers a range of practical advantages, including improved flock management, optimized breeding and culling decisions, and more accurate record-keeping. By automating age assessment through non-invasive image analysis, the system reduces labor, handling stress, and reliance on guesswork, especially valuable for large or inherited flocks. It also supports better health monitoring, informed pricing, and compliance with regulatory requirements. When integrated with other data like RFID, GPS, and health records, age information enhances precision livestock management and enables smarter, data-driven decisions on the farm.

Limitations:

- **Cost of implementation**: The initial setup for GPS tracker, RFID tags, and sensors can be costly, which may be a barrier for some farmers.
- **Dependency on technology**: The system's effectiveness depends on the reliability of the technology. Any malfunction or failure in sensors or tracking systems could lead to inaccuracies in data.
- **Integration challenges**: Integrating multiple technologies (GPS, RFID, machine learning) into one seamless solution may require significant technical expertise and could face challenges in real-world implementation.

Commercial Potential

The commercial potential is significant, given the growing global demand for smart farming technologies. Livestock farmers, particularly in regions where animal health is a major concern and operational costs are high, would benefit from this integrated solution. Potential revenue streams could include:

- **Sales of the system**: Direct sales to farmers or agricultural companies looking to implement smart farming solutions.
- **Subscription model**: Ongoing access to software updates, machine learning model improvements, and data analytics could be offered as a subscription service.
- **Partnerships with veterinary services**: Collaborations with veterinary services could open up additional revenue streams, as the system could be marketed as a tool for disease prevention and herd management.

Resources

The following resources are required to implement the solution effectively:

Hardware

- Collars: Adjustable anti-lost collars for attaching tracking and sensor equipment to livestock.
- GPS Tracker: Tracki GPS tracker for monitoring real-time location and preventing livestock theft.

- RFID System:
  - Reader: Long-range UHF RFID Reader (125kHz/13.56MHz)
  - Tags: UHF RFID Animal Ear Tags Animal ID Management
- Water Level Monitoring:
  - Waterproof ultrasonic sensor JSN-SR04T / AJ-SR04M for detecting water levels in drinking stations
- Processing Components:
  - Raspberry Pi 3 Model B+: For processing RFID and water level data
  - Channel Level Converter: For logic voltage shifting
  - Breadboard & Jumper Wires: For prototype assembly and testing

Sensors

- Water Level: JSN-SR04T / AJ-SR04M ultrasonic sensor
- Identification: RFID ear tags with long-range reader
- Location: Tracki GPS unit with a Real-time global tracker.

Software Development

- Backend: Supabase with PostgreSQL for data processing and storage
- Frontend: Flutterflow-based mobile app for real-time monitoring and alerts
- AI Model: A custom machine learning model trained to estimate age based on sheep teeth images with the use of PyTorch.

Data Collection

- Sheep will wear collars equipped with GPS and RFID components
- Cameras will be used to capture teeth images for the age estimation classification model dataset
- RFID tags log water intake per sheep at designated drinking stations
- Water level sensors collect total consumption data
- All data is transmitted via Raspberry Pi and stored in a cloud-connected backend (e.g., Google Cloud + Supabase) for analysis, model training, and mobile visualization

Intellectual Property

Some aspects that can be subject to intellectual property are:

- The core application, including its user interface, backend logic, and image processing pipeline, can be protected through copyright and potentially as a software patent.
- The distinctive name and logo can be registered as trademarks, helping to build brand recognition and safeguard market identity.
- Machine learning model to estimate age
- The unique combination of GPS, RFID, age estimation and water consumption monitoring for individual sheep could be eligible for patent protection

- The dataset used to train the classification model holds significant value. It can be protected as a trade secret and licensed for research or commercial purposes. Proper data handling and user agreements will ensure ownership rights are maintained.

By developing and protecting these intellectual properties, the project can establish a competitive advantage and enable future commercialization.

Technical Description

System Architecture

1. System Overview

The system consists of:

- IoT Devices: GPS-enabled tracker devices for livestock tracking, RFID-based water monitoring systems, and cameras for sheep teeth image collection.
- Communication Network: LoRaWAN, cellular (4G/5G), or Wi-Fi to transmit data from IoT devices to the cloud.
- Cloud Infrastructure: A cloud-based backend that processes, stores, and analyzes data (Google Cloud Service).
- User Interface: A web and mobile application for real-time monitoring (FlutterFlow).

2. Layered System Architecture

The system follows a three-layer architecture:

A. Edge Layer (IoT & Sensor Devices)

This layer is responsible for data collection from sensors and devices attached to the livestock.

- Tracki GPS tracker:
  - Each sheep has a GPS-enabled device to track real-time location.
  - The device includes a low-power GPS module witch a 3-band tracker (4G, 3G, 2G)
  - Sends geolocation updates every few minutes to conserve battery.
- RFID Water Intake Monitoring System:
  - Sheep are tagged with RFID ear tags.
  - RFID readers at water stations detect when a sheep drinks.
  - Water-level sensors measure overall consumption.
  - Data is sent via a LoRa or Wi-Fi gateway to the cloud.
- Power Considerations:
  - GPS device: Uses low-power sleep cycles to extend battery life (e.g., Tracki).
  - RFID readers & water sensors: Powered by solar panels or battery packs.

B. Network Layer (Communication & Connectivity)

This layer ensures reliable data transmission from IoT devices to the cloud.

- Primary Communication Options:
  - LoRaWAN (Long Range Wide Area Network) – For remote areas with limited connectivity.
  - Cellular (4G/5G) – Where coverage is available, for real-time updates.
  - Wi-Fi – Used at farms with existing internet infrastructure.
  - RFID – Radio Frequency Identification used in farms.
- Gateway Device:
  - LoRaWAN and RFID devices send data to a LoRa Gateway.
  - The gateway forwards data via MQTT or HTTP to the cloud backend.
- Security Measures:
  - AES-256 encryption for GPS and health data.
  - Authentication tokens for IoT devices to prevent unauthorized access.

C. Cloud & Application Layer (Backend, Processing & User Interface)

This layer provides data storage, analysis, and visualization.

- Cloud Backend (Server & Database):
  - Runs on AWS, Google Cloud, or a dedicated server (e.g., Supabase + PostgreSQL).
  - Stores:
    - Livestock tracking data (GPS coordinates, timestamps).
    - FAMACHA classification scores and health records.
    - Water consumption logs (RFID detections, intake volume).
  - Uses Node.js (Express.js) for API handling and PostgreSQL for data storage.
- Machine Learning Model (FAMACHA Classification Processing):
  - Runs in a Cloud AI service (Google Cloud AI, AWS SageMaker, or local server).
  - Processes images from edge cameras and classifies anemia severity using CNN (Convolutional Neural Networks).
  - The trained model is updated periodically with new labeled data.
- User Interface (Mobile App):
  - Will be developed using Flutterflow.
  - Displays GPS tracking on a map (Google Maps API).
  - Alerts farmers when health anomalies are detected (Machine Learning Software).
  - Allows farmers to view water intake patterns and FAMACHA scores.

3. Data Flow in the System

1. GPS tracker & IoT sensors collect data (location, water intake, FAMACHA images).
2. Data is transmitted via LoRaWAN/Cellular/Wi-Fi to a cloud gateway.
3. The cloud backend processes and stores the data in PostgreSQL.
4. Machine Learning models analyze FAMACHA images and classify the sheep's health.
5. Processed results are sent to the mobile & web dashboard for farmers.

6. Farmers receive alerts for sick animals, water intake issues, or location problems.

4. Security & Data Protection Measures

- End-to-End Encryption (E2EE) for data in transit & at rest.
- Authentication & Access Control:
  - Role-based permissions (Admin, Veterinarian, Farmer).
  - Two-factor authentication for critical actions.
- Tamper-proof GPS logging to prevent data manipulation.

5. Scalability Considerations

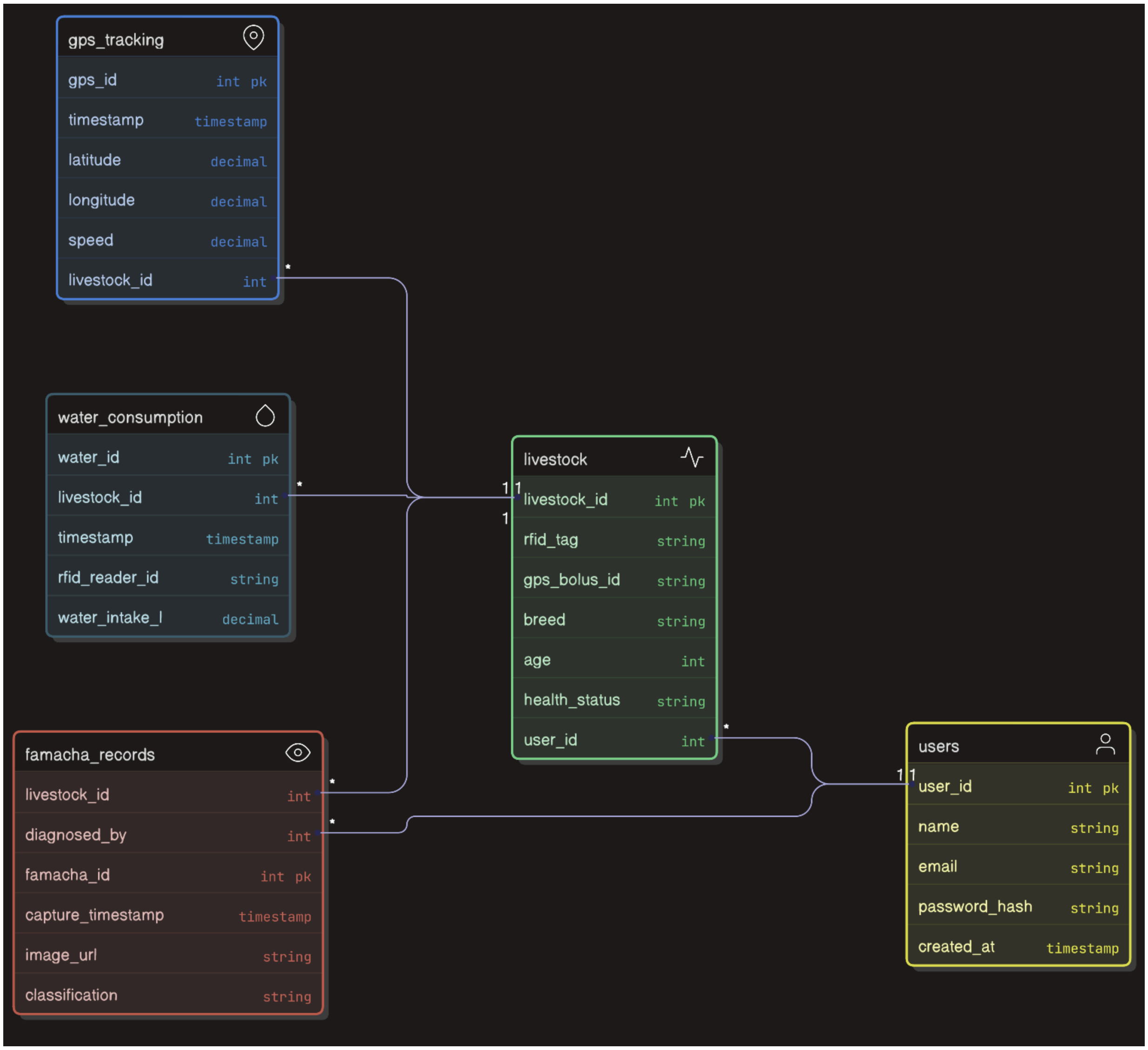
- Modular IoT architecture: Easily add more sensors or tracking devices.
- Cloud scalability: Can process thousands of GPS updates simultaneously.
- Edge AI processing: Reduces server load by handling image processing locally.

Modules Description

Module	**Function**
IoT Device Module	Collects GPS, health, and water intake data from sheep.
Communication Module	Sends sensor data to the cloud via LoRaWAN, 4G, or Wi-Fi.
Backend Processing Module	Stores, analyzes, and manages livestock data.
User Interface Module	Provides dashboards, maps, and alerts for farmers.
Reporting & Analytics Module	Generates insights, predictions, and reports.

Schemas or ER Diagrams Description

ER Diagram



1. Users Table

Stores information about users who interact with the system (livestock farmers, veterinarians, administrators).

```
CREATE TABLE Users (
        User_ID SERIAL PRIMARY KEY,
        Name VARCHAR(100) NOT NULL,
        Email VARCHAR(100) UNIQUE NOT NULL,
        Password_Hash TEXT NOT NULL,
        Created_At TIMESTAMP DEFAULT CURRENT_TIMESTAMP);
```

Constraints: User_ID is the primary key, Email must be unique.

2. Livestock Table

Records information about each monitored animal.

```
CREATE TABLE Livestock (
    Livestock_ID SERIAL PRIMARY KEY,
    User_ID INT REFERENCES Users(User_ID) ON DELETE CASCADE,
    RFID_Tag VARCHAR(50) UNIQUE NOT NULL,
    GPS__ID VARCHAR(50) UNIQUE NOT NULL,
    Breed VARCHAR(50),
    Age INT,
    Health_Status VARCHAR(50) DEFAULT 'Healthy');
```

Constraints: Each animal must be associated with a user (User_ID), RFID_Tag and GPS_ID must be unique.

3. GPS Tracking Table

Stores the geographic location of each animal in real time.

```
CREATE TABLE GPS_Tracking (

    GPS_ID SERIAL PRIMARY KEY,
    Livestock_ID INT REFERENCES Livestock(Livestock_ID) ON DELETE CASCADE,
    Timestamp TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
    Latitude DECIMAL(10, 6) NOT NULL,
    Longitude DECIMAL(10, 6) NOT NULL,
    Speed DECIMAL(5, 2));
```

Constraints: Each GPS record is associated with a single animal (Livestock_ID as FK).

4. FAMACHA Records Table

Records the results of the FAMACHA analysis to assess the health of animals.

```
CREATE TABLE FAMACHA_Records (

    FAMACHA_ID SERIAL PRIMARY KEY,
    Livestock_ID INT REFERENCES Livestock(Livestock_ID) ON DELETE CASCADE,
    Capture_Timestamp TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
    Image_URL TEXT NOT NULL,
    Classification VARCHAR(10) CHECK (Classification IN ('1', '2', '3', '4', '5')),
    Diagnosed_By INT REFERENCES Users(User_ID) ON DELETE SET NULL);
```

Constraints: Classification values will be from 1 to 5 (FAMACHA standard), Diagnosed_By refers to the user who made the diagnosis.

5. Water Consumption Table

Record each animal's water intake with RFID technology.

```
CREATE TABLE Water_Consumption (
    Water_ID SERIAL PRIMARY KEY,
    Livestock_ID INT REFERENCES Livestock(Livestock_ID) ON DELETE CASCADE,
    Timestamp TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
    RFID_Reader_ID VARCHAR(50) NOT NULL,
    Water_Intake_L DECIMAL(5, 2) CHECK (Water_Intake_L >= 0));
```

Constraints: Water_Intake_L can’t be negative. RFID_Reader_ID represents the water station where the animal's tag was read.

UI Wireframe

1. Login & Authentication

Purpose: Secure user login.

Elements:

- Email & Password fields
- "Forgot Password?" link
- Login Button

2. Dashboard

Purpose: High-level overview of farm activities.

Elements:

- Summary Cards (Total Livestock, Active Alerts, Recent GPS Movements)
- Real-time GPS Map
- Recent Health & Water Intake Reports

3. Livestock Management

Purpose: Manage individual animals.

Elements:

- Search Bar (RFID, Name, ID)
- Table View (ID, GPS Status, Health, RFID, Water Intake)
- "View Details" Button

4. GPS Tracking

Purpose: animal movement monitoring.

Elements:

- GPS Map
- Search by ID

5. Health Monitoring

Purpose: Displays FAMACHA scores and health alerts.

Elements:

- FAMACHA Image & Classification Score
- Health Alerts (Severe cases marked)
- "Request Veterinary Help" Button

6. Water Consumption Tracking

Purpose: Monitor livestock water intake.

Elements:

- Water Trends Graph
- Daily Intake Logs (ID, Date, Liters)
- Low Intake Alerts

7. Reports & Analytics

Purpose: Generate livestock reports.

Elements:

- Graphs & Charts (Health, Water, GPS)
- Filters by Date, Animal, Category

8. Settings & User Management

Purpose: Profile management and system customization.

Elements:

- Change Password & Email
- Manage User
- Enable/Disable Notifications

Algorithms Flowcharts Description

1. User Login & Authentication Flowchart

Purpose: Ensures secure login and access control.

Steps:

1. Start
2. User enters email and password
3. Validate credentials
    - If valid → Check user role
    - If invalid → Display error & retry
4. End

2. Livestock GPS Tracking Flowchart

Purpose: Monitors real-time location and detects straying livestock.

Steps:

1. Start
2. Collect GPS data
3. Check if the location is within the geofence
    - Yes → Continue monitoring
    - No → Generate alert
4. Send GPS data to the cloud
5. Display location on dashboard
6. Repeat process at intervals
7. End

3. Health Monitoring (FAMACHA Classification) Flowchart

Purpose: Detects anemia in livestock based on eye color classification.

Steps:

1. Start
2. User analyzes the sheep eye and select the appropriate category
3. Determine category (1-5)
    - Score 1-2 → Don't deworm unless there is other evidence of parasitic disease such as the presence of diarrhea, poor body condition, dull hair coat or abnormal fleece
    - Score 3 → Deworming is recommended if more than 10% of the flock has a score of 4 or 5, in lambs and kids, as well as in pregnant or lactating sheep and goats. It is also advisable in animals with poor body condition or with health

problems, such as additional diseases. When in doubt, it is always better to be safe and act with caution.
    - Score 5 → Always deworm sheep & goats in categories 4 & 5.
4. Store results and notify user
5. End

4. Water Consumption Monitoring Flowchart

Purpose: Tracks livestock hydration levels using RFID and sensors.

Steps:

1. Start
2. Detect RFID tag at drinking station
3. Record water intake in liters
4. Store data in the database
5. Check if intake is below threshold
    - Yes → Generate Low Water Intake Alert
    - No → Continue monitoring
6. Display data on dashboard
7. End

5. Alerts & Notifications Flowchart

Purpose: Triggers alerts based on GPS, health, and water data.

Steps:

1. Start
2. Monitor GPS location, FAMACHA score, and water intake
3. Check for geofence violations
    - Yes → Generate Lost Livestock Alert
    - No → Continue monitoring
4. Check FAMACHA score
    - If Score ≥ 4 → Generate Health Alert
    - If not → Continue monitoring
5. Check for low water intake
    - Yes → Generate Hydration Alert
    - No → Continue monitoring
6. Send notification to user
7. Store alert in the database
8. End

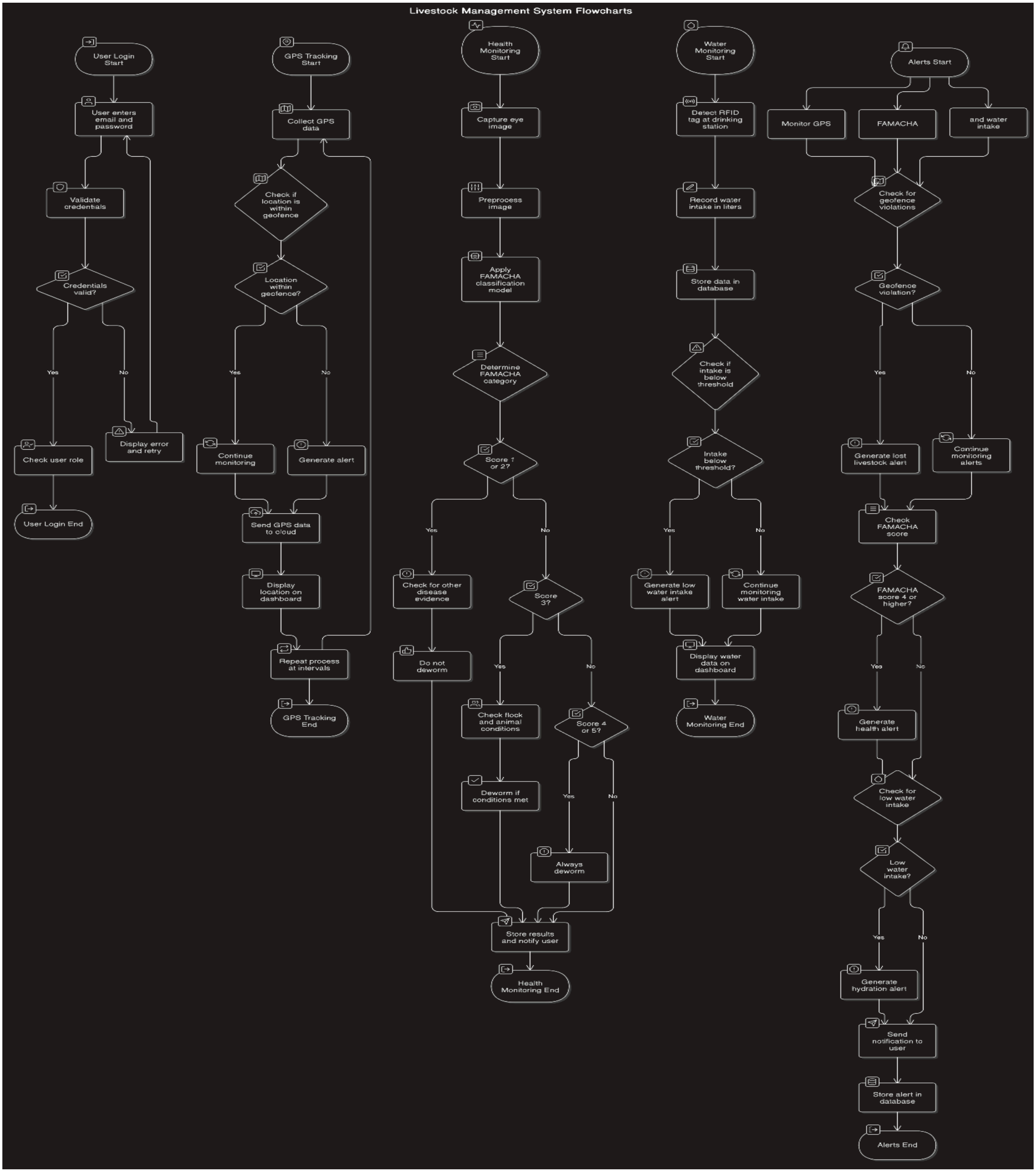
6. Age estimator Flowchart

Purpose: Detects age in livestock based on their teeth.

Steps:

6. Start
7. Capture teeth image
8. Preprocess image
9. Apply teeth classification model
10. Determine age
    - Below 1 year
    - 1 - 1.5 years
    - 2 - 2.5 years
    - 3 - 3.5 years
    - 4 - 4.5 years
11. Notify user and optionally store results
12. End

Flowchart Description



Engineering Constraints

Constraint Type	Challenges	Proposed Solution
Hardware	Power limitations, size restrictions	Low-power MCUs, optimized GPS updates
Connectivity	Limited network coverage, bandwidth constraints	LoRaWAN, edge AI processing
Data Storage & Processing	Large data volumes, latency	Compression, prioritizing critical data
Security	Data breaches, unauthorized access	AES-256 encryption, RBAC
Cost	Hardware & cloud expenses	Open-source software, cost-effective hardware
Regulations	Animal welfare laws, GDPR compliance	Biocompatible materials, data anonymization
Scalability	Adding more sensors increases load	Multi-gateway LoRa, serverless cloud
User Training	Farmers' tech expertise varies	Simple UI, video tutorials

Engineering Standards

To ensure our Smart Livestock Monitoring System meets high industry standards in software development, hardware integration, security, and usability, we will follow internationally recognized standards. Below is how each standard applies to our project.

1. Software Development Standards
    a. ISO/IEC/IEEE 12207:2017 - Software Life Cycle Processes
        i. Our software development will adhere to structured processes for planning, implementation, testing, and maintenance.
    b. ISO/IEC 25010:2011 - Software Quality Requirements and Evaluation (SQuaRE)
        i. Ensures the system meets quality attributes like reliability, usability, and maintainability.
    c. ISO/IEC JTC 1/SC 7/WG 29 - Agile Methodology
        i. We will use an Agile approach to enable incremental updates, fast iterations, and user feedback integration.
2. Authentication and Security Standards
    a. JSON Web Token (JWT) Standard RFC 7519
        i. Secure authentication and session management for users accessing the livestock monitoring dashboard
    b. ISO/IEC 27001 - Information Security Management
        i. Protects livestock location data, health records, and water intake logs from unauthorized access.

   c. ISO/IEC 18033-3 - AES-256 Encryption
      i. Used to encrypt transmitted and stored data, ensuring sensitive information remains secure.
3. IoT and Hardware Standards
   a. IEEE 1451 - Smart Transducer Interface for Sensors
      i. Ensures our GPS, RFID, and water level sensors integrate seamlessly with the cloud-based monitoring system.
   b. ISO 11784/11785 - RFID Livestock Identification
      i. Standardizes RFID-based tracking of individual livestock for accurate water intake monitoring.
   c. ISO 24631 - Performance Testing of RFID Devices
      i. Guarantees accuracy and durability of RFID-based livestock monitoring.
   d. ISO 15118 - Power Management for IoT Devices
      i. Optimizes power consumption for remote GPS and RFID trackers.
4. Networking & Data Communication Standards
   a. LoRaWAN 1.0 Specification
      i. Enables long-range, low-power communication for tracking livestock in rural environments.
   b. IEEE 802.15.4 - Wireless Communication for IoT
      i. Used for low-energy sensor-to-cloud communication.
   c. ISO/IEC 20922 - MQTT Protocol for IoT
      i. Ensures lightweight, real-time data transmission from sensors to the cloud.
   d. IPv6 (RFC 8200) - Addressing for IoT Networks
      i. Provides scalability and enhanced security for connected devices.
5. AI & Machine Learning Standards
   a. ISO/IEC 20546 - AI Terminology & Concepts
      i. Standardizes terminology for our FAMACHA-based health monitoring system.
   b. IEEE P7001 - AI Transparency & Explainability
      i. Ensures our machine learning model for anemia detection is interpretable by farmers.
   c. ISO/IEC 23894 - AI Risk Management
      i. Guides risk assessment for AI-driven health analysis.

By following these engineering standards, we ensure that our Smart Livestock Monitoring System is reliable, secure, scalable, and compliant with industry best practices.

Project Plan

Project Activities & Milestones

Phase 1: Research & Requirement Analysis (Feb 1 – Mar 1, 2025)

The first phase focuses on gathering information, defining system requirements, and selecting suitable hardware and software components and methods.

Key Steps:
- Research smart farming practices, with emphasis on GPS-based tracking boluses, IoT water monitoring sensors, and software applications for collecting and storage data.
- Analyze existing systems such as **Moonsyst Smart Monitoring System,** which offers real-time health tracking and GPS capabilities for livestock [5].
- Identify viable communication methods for both short and long-range transmission using **Raspberry Pi** and **RFID systems** integrated with a **Supabase** backend.
- Define overall system requirements, including hardware specifications, software stack, and data management protocols compatible with **Raspberry Pi**.

Deliverables:
- Report on research of smart farming technologies and systems.
- System requirements documentation detailing hardware and software requirements.

Phase 2: System Design & Planning (Mar 1 – Mar 15, 2025)
This phase focuses on designing the system architecture and defining how individual components will work together based on gathered information from the previous phase.

Key Steps:
- Select and configure microcontrollers (e.g., ESP32) to interface with water and health sensors
- Design mobile application UI in FlutterFlow for real-time data visualization.
- Develop a Supabase-based database schema to store animal health data, GPS location and sensor readings.

Deliverables:
- System architecture diagram all components and data flows.
- UI wireframes for the FlutterFlow application.
- Database schema for structured storage & analysis.

Phase 3: Hardware and Software Development (Mar 16 – Apr 1, 2025)
For this phase, all system components will be developed, integrated and tested.

Key Steps:
- Configure and integrate GPS trackers, water consumption sensors (e.g., JSN-SR04T, AJ-SR04M), and health monitoring systems.
- Develop the FlutterFlow app frontend, bind Supabase queries to UI and implement push notifications.
- Build backend services for data collection, processing and storage using Supabase.

Deliverables:

- Integrated microcontroller setup with GPS and sensor connectivity.
- Initial version of the mobile application with basic functionalities.
- Operational backend and Supabase database integration.

Phase 4: Testing and Optimization (Apr 2 – Apr 15, 2025)
This phase ensures system reliability, accuracy, and performance before deployment.

Key Steps:
- Test GPS tracking accuracy and sensor data reliability.
- Evaluate mobile app performance under different conditions.
- Improve user experience.

Deliverables:
- Test reports on system performance and reliability.
- Optimized mobile application and backend systems.

Phase 5: Deployment & Evaluation (Apr 16 – May 1, 2025)
The final phase involves deploying the system in a controlled environment and collecting feedback.

Key Steps:
- Deploy water monitoring sensors, RFID tag/readers and GPS trackers on a test group of animals.
- Monitor system performance and data collection in real-world conditions through the FlutterFlow application and Supabase database.
- Evaluate sensor data and mobile app usability.

Deliverables:
- Deployment report.
- Final project report including practices, issues and possible improvements.

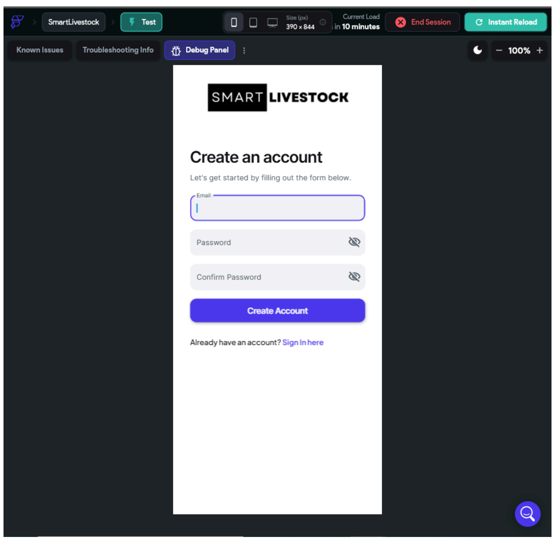
Team Roles & Responsibilities

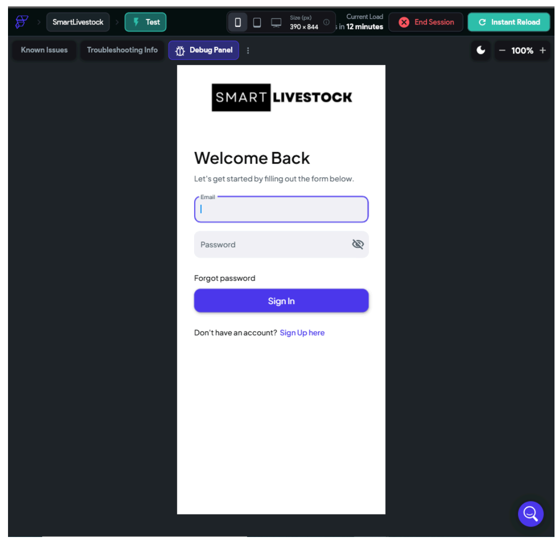
Team Member	Responsibilities
Axel J. Perez Rodriguez	Integrate wearable GPS tracking devices for livestock animals. Implement real-time tracking data in the FlutterFlow application UI, using Supabase, fire-base and google cloud.
Mark A. Alvarez Nieves	Water monitor sensor implementation and data collection. Develop FlutterFlow application UI and Supabase data storage corresponding to water monitor sensors.
Eduardo Martinez Calvo	RFID tag/reader and microcontroller implementation.

	Implement animal identification software for health monitors.
Edgar J. Sanabria Soto	Implement machine learning software to determine sheep age by analysing their teeth. Implement software using visual data to identify an animal's weight.

Performance

SignUp/LogIn





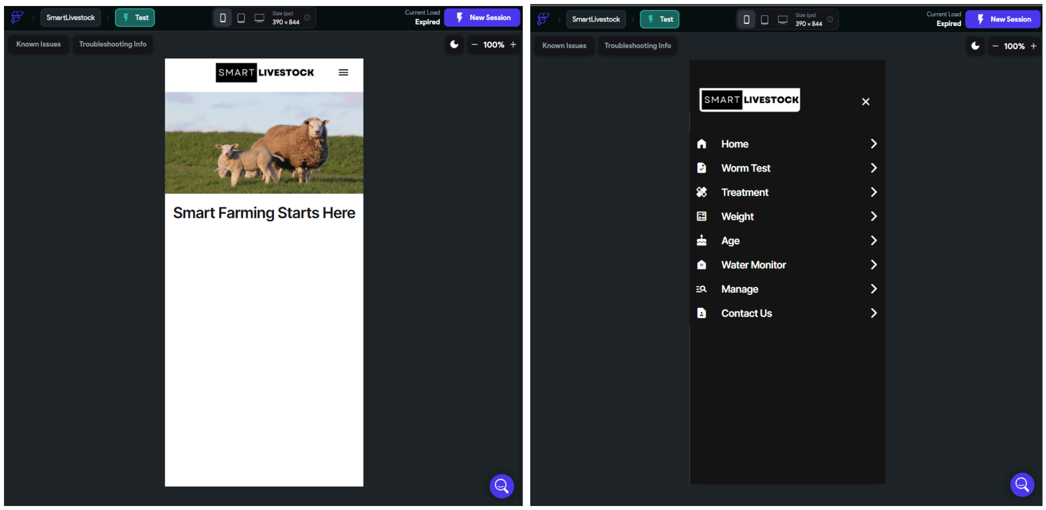
What is it?

- In this page of the application the user can input their account information to create a new account or enter a pre-existing account in the sign up page or log in page respectively

How does it work?

- On initial use, the app will automatically open the Sign Up page, where the user can input their information in the prompt and the application will save that data in the Supabase Backend allowing the user to access more of the applications functions.

Home



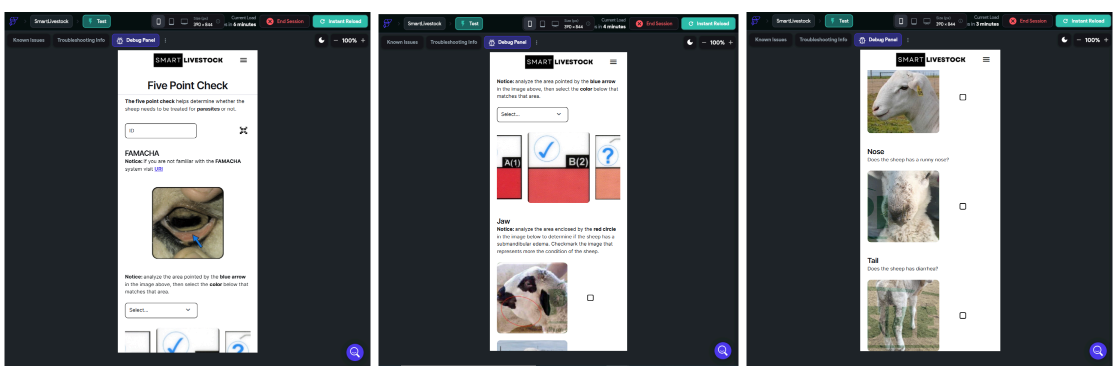
What is it?

- In this page of the application the user can select different functions in the application.

How does it work?

- FlutterFlow creates a mobile and web application Frontend using Supabase Backend. Here the user can select Home, Worm Test, Treatment, Weight, Age, Water Monitor Manage and Contact Us.

Worm Test



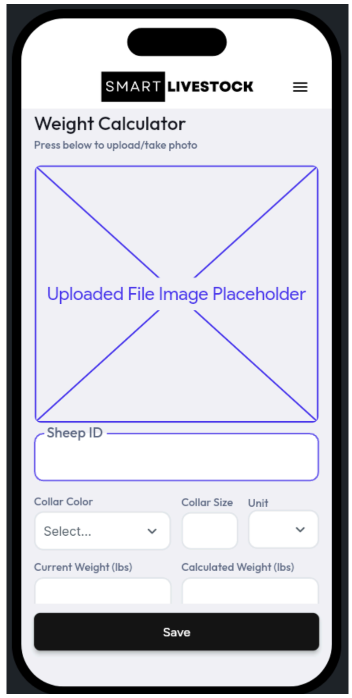
What is it?

- In this page the user can follow a series of steps to determine if the sheep needs parasite treatment.

How does it work?

- The application shows a series of images of healthy and sick specimens focusing on identifying physical anomalies. The user can select the option that closely resembles the animal’s symptoms and the application will give guidance to treatment or to inform a medical expert if the case is too severe.

Weight Estimator



What is it?

- In this page of the application the user can upload an image of the animal's profile (side view) to the application and input its identification number, collar color, collar size and unit. The application will output an estimated weight of the animal and save it to the Supabase database.

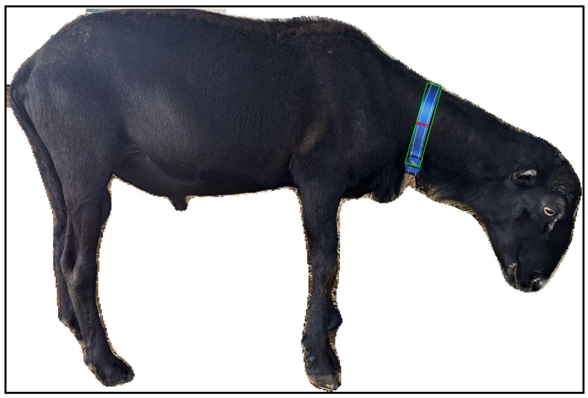
How does it work?

- This page runs python code that has constant data input to determine an approximate weight range using predetermined information pertaining to the animal and information given by the user. The program will analyze the given image and identify the collar of the animal in the image as a basis to calculate the dimensions of the animal using the image's pixel measurements, from there the program will calculate an approximate weight.

Results

- Image uploaded

- Image processed

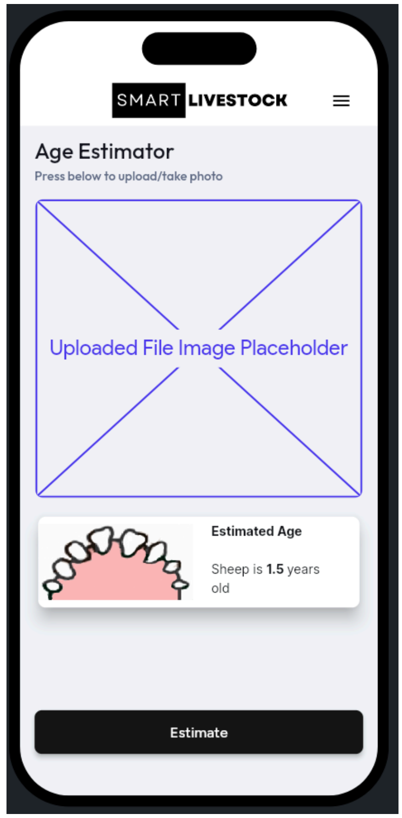


- Weight estimation

```
Sheep Weight Estimator


image 1/1 C:\Users\edgar\Documents\Courses\INSO4151\App\
Speed: 9.4ms preprocess, 570.5ms inference, 15.4ms postp
Sheep detected at (1023, 1363), width=4230, height=2808

Estimated collar width: 89.00 pixels
Estimated thoracic perimeter: 97.8067415730337
Estimated diagonal body length 86.45292454920607

Actual weight: 169 pounds
Estimated weight: 168 pounds

Process finished with exit code 0
```

- Live test
  - The demo provided shows the functionality working as intended.

Age



What is it?

- In this page of the application the user can upload an image of the animal's mouth to estimate its age using machine learning and provide that information to the user and save that data to the Supabase database.

How does it work?

- The application runs image oriented machine learning software that uses example data from confirmed data points of an animal’s mouth and their exact age as a basis. Using that data and the image the user provides the application, the software can give an approximate age range and save that information to the Supabase database through Backend Queries.

Results

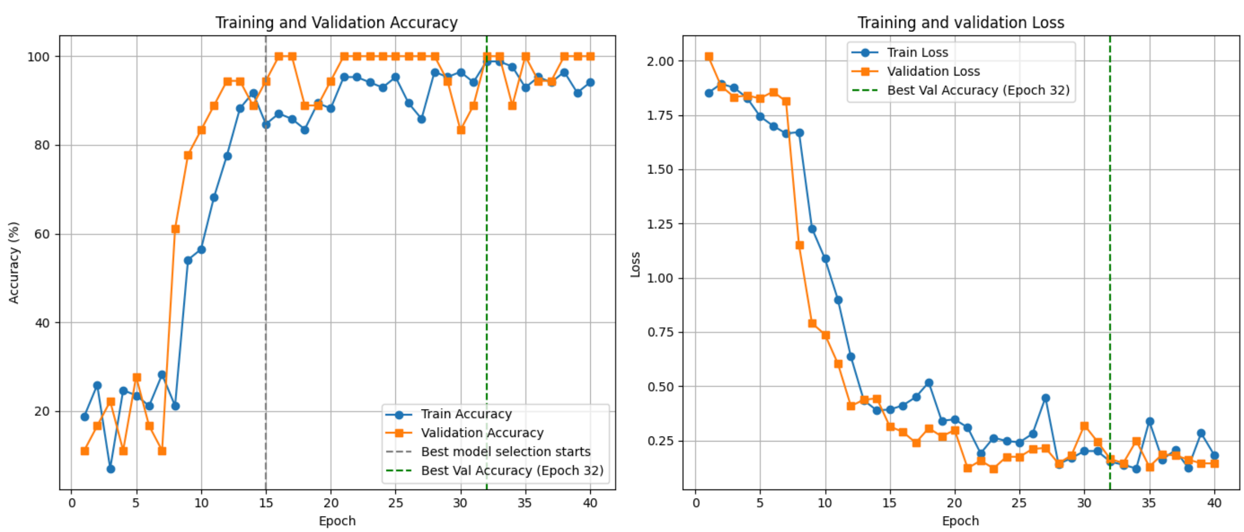
To evaluate the model's performance, we conducted many training runs. Once the model started showing good results, we conducted five independent training runs using a ResNet18-based architecture with transfer learning. Initially, all convolutional layers were frozen, and only the final classifier was trained. From epoch 8 onward, the layer4 block was unfrozen for fine-tuning. Training was performed for 40 epochs with early stopping criteria in place.

The key performance metrics are summarized below:

- **Best Run Accuracy**:
    - **Training Accuracy**: 98.82%
    - **Validation Accuracy**: 100.00%
    - **Achieved at Epoch**: 32 (Run 1), 39 (Run 5)
- **Mean Training Accuracy (Epochs 8–40)**:
    - Range across 5 runs: **85.46% – 88.98%**
    - **Standard Deviation**: Ranged from **5.67% to 10.94%**, indicating consistent convergence post-fine-tuning.

Full training logs were recorded and analyzed, with performance plotted and tracked per epoch. Results consistently showed strong generalization after unfreezing layer4, confirming the benefit of progressive fine-tuning. Each model checkpoint and training curve is stored and documented. Two files are provided one shows the precise results per epoch and the other shows the confidence values of each validation test.

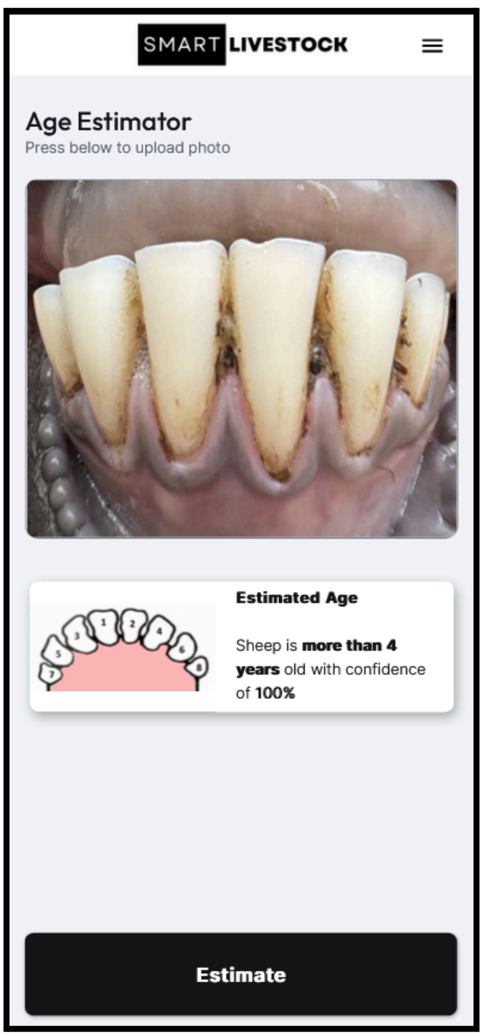
The following graph shows the training and validation curve of accuracy and loss per epoch of the selected model training run.

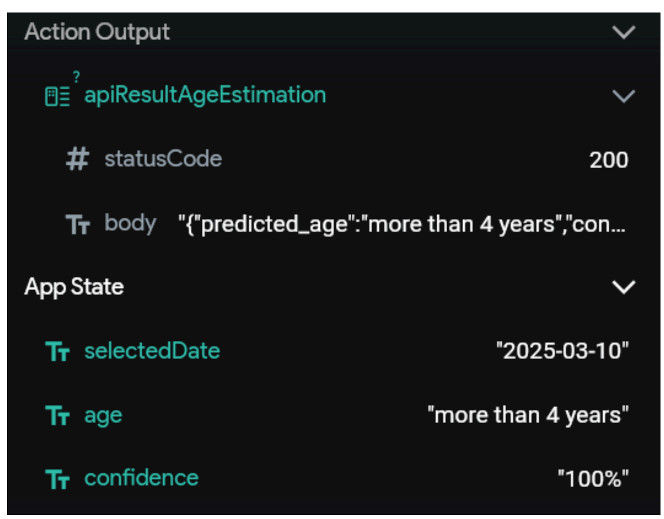


The following shows the results of the five runs.

Metric	Run 1	Run 2	Run 3	Run 4	Run 5
Mean Accuracy	85.91%	85.46%	87.83%	88.98%	88.65%
Std Deviation	10.94%	9.34%	6.38%	5.67%	5.83%
Max Accuracy	98.82%	97.65%	96.47%	98.82%	98.82%
Min Accuracy	20.00%	28.24%	30.59%	21.18%	21.18%
Stability	Least	Low	Moderate	Highest	High

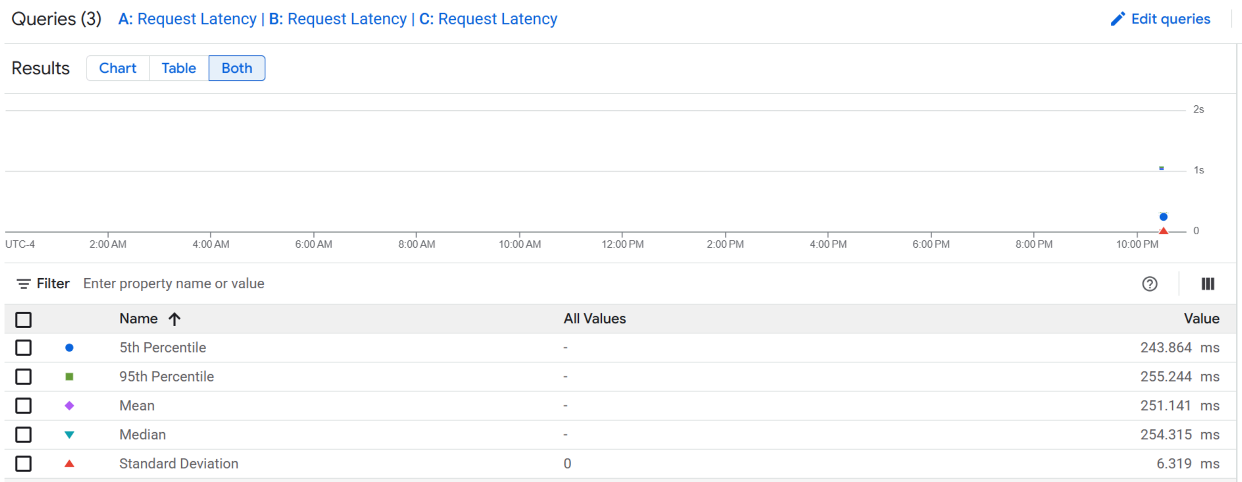
Live test





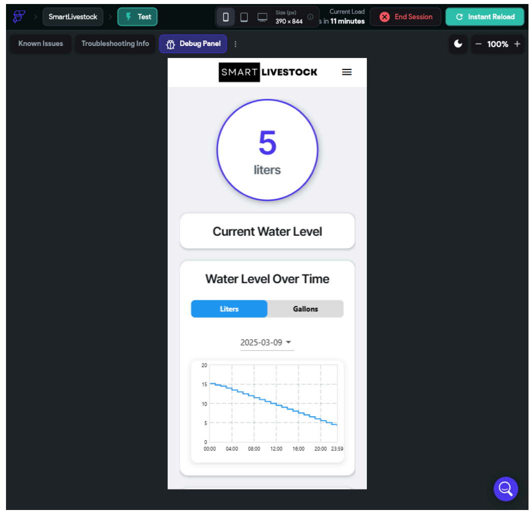
Google Cloud Run Service Latency

Overall, the Cloud Run service is fast and stable. Most requests fall in a tight range (244ms - 255ms). The low standard deviation means consistent performance. Users are waiting less than 300ms per prediction, which is excellent.



Metric	Value (ms)	Interpretation
5th Percentile	243.864	Very fast, 5% of requests were faster than this.
95th Percentile	255.244	Almost all requests were under 255ms.
Mean (Average)	251.141	Typical requests take about 251ms.
Median	254.315	Half of requests take more than 254ms.
Standard Deviation	6.319	Very little variability requests are consistently fast.

Water Monitoring



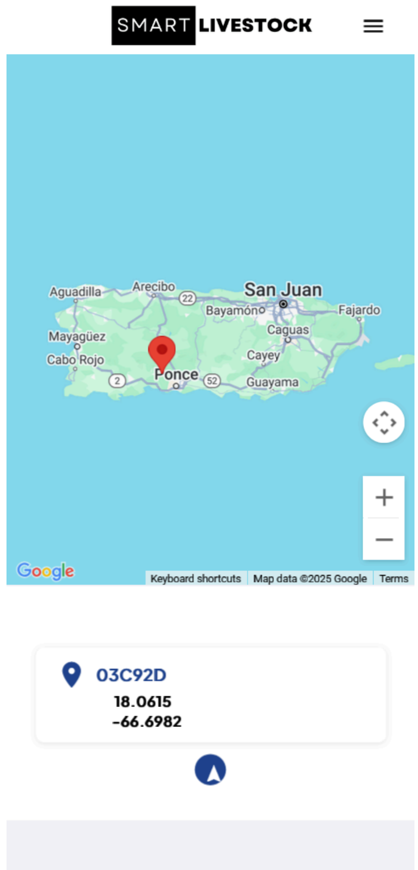
What is it?

- In this page of the application the user can monitor the most recent amount of water currently in the water container, followed by an interactive graph which shows the summarized change in water level of the container across a 24 hour period and the user can select what day to check stored data and the preferred unit of measurement.

How does it work?

- The amount of water registered on the graph is taken from a Backend Query in FlutterFlow taken from a Supabase table, which stores data collected from the Waterproof ultrasonic sensor JSN-SR04T / AJ-SR04M and sent to the database using the Raspberry Pi 3 Model B+.
- The FlutterFlow app collects and arranges this data in a format the user can better analyze and watch in real-time data collection.

GPS



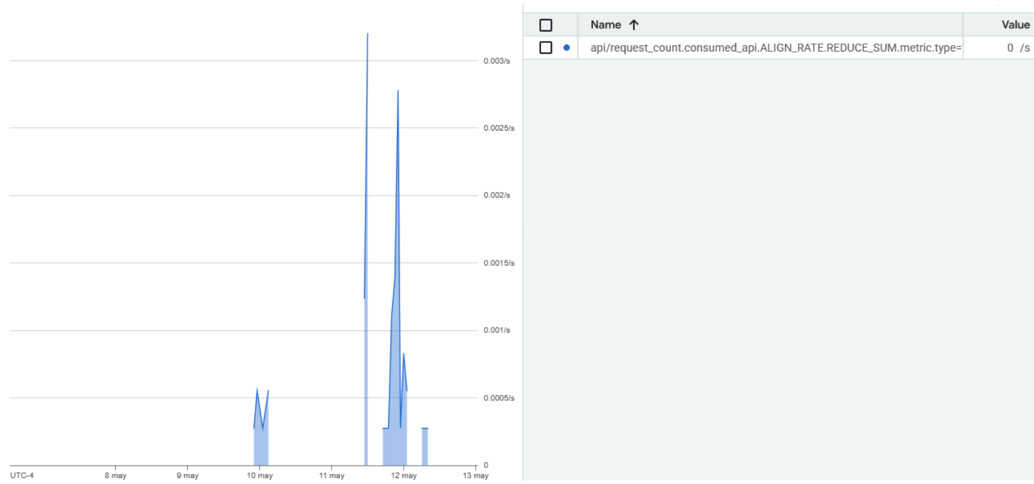
What is it?

- This section of the application allows the user to view the precise real-time location of the animal, using data provided by a Tracki GPS tracker attached to the animal's collar.

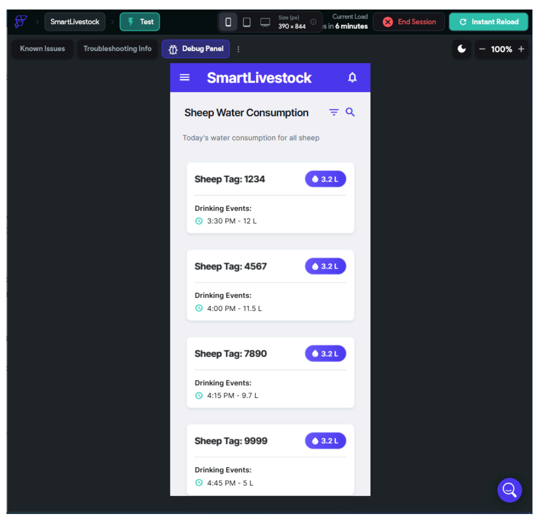
How does it work?

- The in-app GPS system works by obtaining the device's real-time location using the geolocation services built into FlutterFlow. This information is captured and stored in a Firebase database for persistence. The Google Maps API, enabled through Google Cloud, is then used to display the captured coordinates directly on the map within the app. This integration allows users to see precise, dynamically updated locations, all from a developed interface.

Performance



Manage



What is it?

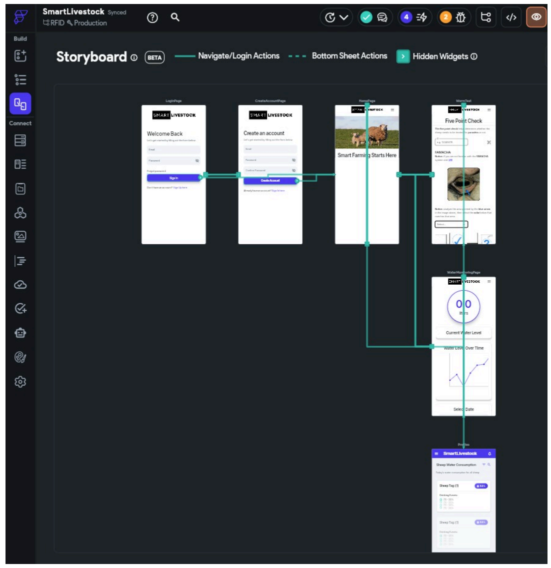
- In this page of the application the user can monitor the amount of water consumed by which animal and the specific time when the data was stored to keep real-time updates.

How does it work?

- The amount of water consumed is registered in a Supabase table, which stores data collected from the Waterproof ultrasonic sensor JSN-SR04T / AJ-SR04M and sent to the database using the Raspberry Pi 3 Model B+ to send data to our backend.
- We identify what animal consumed the water by using a Long-range UHF RFID Reader (125kHz/13.56MHz) to scan the RFID tags on the animal within a given range of the water container to accurately determine the animal consuming the amount of water at any given time.

Project Documentation

Mobile app



Repositories

Mobile application

- https://github.com/Edgar-0/SmartLivestock/branches

Sheep Weight Estimator

- Local: https://github.com/Edgar-0/SheepWeightEstimator
- Production: https://github.com/Edgar-0/sheepDetectorAPI
- Production: https://github.com/Edgar-0/collarDetectorAPI

Sheep Teeth Model

- Local: https://github.com/Edgar-0/SheepTeethAgeEstimatorModel
- Production: https://github.com/Edgar-0/sheepAgeEstimatorAPI
- Dataset: https://drive.google.com/drive/folders/1hrbQXcK2DULKjwuldIJZmW95Rp8x7NFb?usp=sharing

References

1. C. N. Belage, B. R. McEwen, A. C. MacNicol, A. P. MacKay, and J. DeVries, "Machine learning for the prediction of lameness in dairy cattle: A scoping review," *Frontiers in Veterinary Science*, vol. 8, p. 658763, 2021. [Online]. Available: https://www.frontiersin.org/journals/veterinary-science/articles/10.3389/fvets.2021.658763/full. [Accessed: Feb. 5, 2025].
2. J. Chen, X. Liu, and Y. Zhao, "A novel deep learning approach for livestock monitoring using IoT-enabled sensors," in *2022 IEEE International Conference on Smart Agriculture Technologies (ICSAT)*, pp. 1-6. [Online]. Available: https://ieeexplore-ieee-org.uprm.idm.oclc.org/document/9980682/metrics#metrics . [Accessed: Feb. 5, 2025].
3. R. Kaplan, J. Burke, and A. Terrill, "FAMACHA©: A tool for selective targeted treatment of barber pole worm in sheep and goats," University of Florida, 2018. [Online]. Available: https://animal.ifas.ufl.edu/media/animalifasufledu/small-ruminant-website/fact-sheets/FAMACHA-Factsheet.pdf. [Accessed: Feb. 5, 2025].
4. Serket, "Technology," Serket Tech, 2025. [Online]. Available: https://www.serket-tech.com/technology. [Accessed: Feb. 5, 2025].
5. Moonsyst, "Moonsyst Smart Monitoring," Moonsyst, 2025. [Online]. Available: https://moonsyst.com/. [Accessed: Feb. 5, 2025].
6. USDA, "Why and How To Do FAMACHA© Scoring." [Online]. Available: https://web.uri.edu/wp-content/uploads/sites/241/FAMACHA-Scoring_Final2.pdf
7. Google, "Smart Livestock Project – Google Cloud Console," [Online]. Available: https://console.cloud.google.com