Ollscoil Teicneolaíochta an Atlantaigh

Atlantic Technological University

The title of your FYP goes here

Your name and affiliation goes here. Add your GitHub URL.

Introduction

Livestock farmers often face critical decisions when animals exhibit signs of illness, yet many operate in rural areas with limited access to veterinary care and digital resources. This project addresses that challenge by developing a mobile-first, offline-compatible farm management application tailored to Irish cattle farmers.

The app features an AI-powered diagnostic engine that suggests potential diseases based on symptoms entered by the user, along with tools for tracking vaccinations, managing livestock records, and viewing weather data. The goal is to empower farmers with accessible, context-aware decision support that fits naturally into their daily workflow.

.

Problem Statement

Modern livestock farming encounters significant challenges with early disease detection and health monitoring, frequently leading to economic losses and animal welfare issues. Traditional diagnostic approaches depend heavily on veterinary expertise and physical examinations, which aren't always readily accessible to farmers. There's an urgent need for user-friendly, intelligent systems that can deliver preliminary health assessments and guide farmers toward appropriate interventions, especially in remote or resource-constrained settings.

System Architecture

A progressive web application with native mobile capabilities and component-based design for seamless performance across all devices is provided by the system's multi-tier architecture, which is based on Angular 17+ and Ionic. A strong RESTful API with integrated middleware for CORS handling, JWT authentication, and thorough request logging is offered by the Express.js backend.

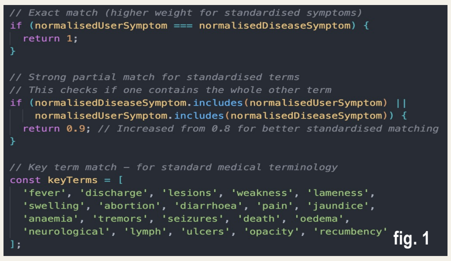
For dependable schema validation and safe user-specific data isolation, data management makes use of MongoDB Atlas cloud infrastructure with Mongoose ODM. In order to preserve system availability in the event of service outages, the platform incorporates a number of external services, such as Firebase Authentication and two weather APIs (WeatherAPI.com and OpenWeatherMap), with built-in fallback mechanisms.

The AI diagnostic engine functions as a stand-alone microservice with parallel processing and in-memory caching. In order to provide confidence-ranked diagnostic recommendations and guarantee accurate and trustworthy health assessments for livestock management, it makes use of standardised symptom databases and sophisticated fuzzy string matching algorithms.

AI Disease Diagnostic Engine

At the core of the application is an offline-capable, rule-based diagnostic engine designed to assist farmers in identifying potential cattle illnesses based on observable symptoms. The engine operates using a multi-tiered fuzzy matching algorithm that compares user-entered symptoms against a curated dataset of over 36 cattle diseases, each with an associated list of clinical indicators. The engine performs symptom normalisation—removing punctuation, lowering case, and accounting for common veterinary terminology variants—to ensure consistency and improve match accuracy.

Each input symptom is evaluated using a weighted scoring system: exact matches are scored at 1.0, strong partial matches at 0.9, and broader key-term matches at 0.85, enabling the system to handle varied expressions of similar symptoms (e.g., "laboured breathing" vs "difficulty breathing"). Additionally, a multi-factor analysis layer dynamically adjusts scores based on the animal's context, such as age group (e.g., calf vs adult diseases), sex (e.g., mastitis in females), reproductive state, and vaccination status—reducing the likelihood of returning diseases the animal is immunised against. [fig. 1]



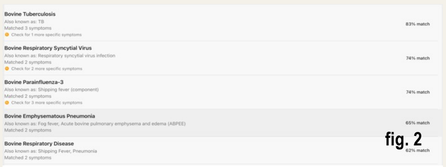
```
// Exact match (higher weight for standardised symptoms)
if (normalisedUserSymptom === normalisedDiseaseSymptom) {
  return 1;
}

// Strong partial match for standardised terms
// This checks if one contains the whole other term
if (normalisedDiseaseSymptom.includes(normalisedUserSymptom) ||
    normalisedUserSymptom.includes(normalisedDiseaseSymptom)) {
  return 0.9; // Increased from 0.8 for better standardised matching
}

// Key term match - for standard medical terminology
const keyTerms = [
  'fever', 'discharge', 'lesions', 'weakness', 'lameness',
  'swelling', 'abortion', 'diarrhoea', 'pain', 'jaundice',
  'anaemia', 'tremors', 'seizures', 'death', 'oedema',
  'neurological', 'lymph', 'ulcers', 'opacity', 'recumbency'
];
```

fig. 1

The engine outputs the five most probable diagnoses, ranked by descending confidence percentage, along with clear displays of matched vs unmatched symptoms [fig. 2][fig. 3]. Each result includes differential diagnosis notes to distinguish between similar conditions and suggests practical next steps based on best practices from veterinary literature. Built in TypeScript as an Angular service, the tool integrates seamlessly with the application's UI, handling asynchronous processing via observables. Its offline-first design and domain-specific logic ensure that farmers receive timely, trustworthy, and context-aware diagnostic support—even in remote areas without internet connectivity.



Bovine Tuberculosis
Also known as: TB
Matched 3 symptoms
Check for 1 more specific symptoms
83% match

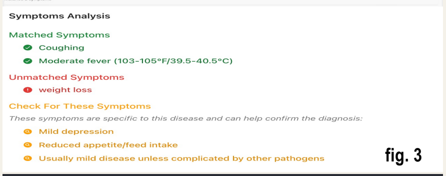
Bovine Respiratory Syncytial Virus
Also known as: Respiratory syncytial virus infection
Matched 2 symptoms
Check for 2 more specific symptoms
74% match

Bovine Parainfluenza-3
Also known as: Shipping fever (component)
Matched 2 symptoms
Check for 3 more specific symptoms
74% match

Bovine Emphysematous Pneumonia
Also known as: Fog fever, Acute bovine pulmonary emphysema and edema (ABPEE)
Matched 2 symptoms
65% match

Bovine Respiratory Disease
Also known as: Shipping Fever, Pneumonia
Matched 2 symptoms
62% match

fig. 2



Symptoms Analysis

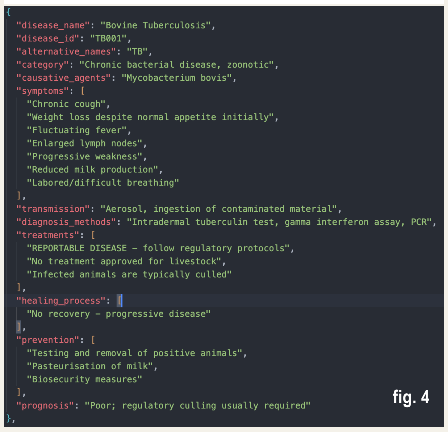
Matched Symptoms
- Coughing
- Moderate fever (103-105°F/39.5-40.5°C)

Unmatched Symptoms
- weight loss

Check For These Symptoms

These symptoms are specific to this disease and can help confirm the diagnosis:
- Mild depression
- Reduced appetite/feed intake
- Usually mild disease unless complicated by other pathogens

fig. 3



```
{
  "disease_name": "Bovine Tuberculosis",
  "disease_id": "TB001",
  "alternative_names": "TB",
  "category": "Chronic bacterial disease, zoonotic",
  "causative_agents": "Mycobacterium bovis",
  "symptoms": [
    "Chronic cough",
    "Weight loss despite normal appetite initially",
    "Fluctuating fever",
    "Enlarged lymph nodes",
    "Progressive weakness",
    "Reduced milk production",
    "Labored/difficult breathing"
  ],
  "transmission": "Aerosol, ingestion of contaminated material",
  "diagnosis_methods": "Intradermal tuberculin test, gamma interferon assay, PCR",
  "treatments": [
    "REPORTABLE DISEASE - follow regulatory protocols",
    "No treatment approved for livestock",
    "Infected animals are typically culled"
  ],
  "healing_process": [
    "No recovery - progressive disease"
  ],
  "prevention": [
    "Testing and removal of positive animals",
    "Pasteurisation of milk",
    "Biosecurity measures"
  ],
  "prognosis": "Poor; regulatory culling usually required"
},
```

fig. 4

User Interface Designs

Several design considerations were taken into account when designing this application's user interface. Mainly, how there is an evident gap in technical literacy amongst a large population of farmers. This meant that the app's user interface had to be intuitive, and provide the least resistance for the user to reach their desired goal.

The Navigation bar, for example, was designed to be as simplistic, yet aesthetically appealing as possible in order to retain user attention, and negate confusion [fig. 5]. Similar techniques were used across the entirety of the application. In sections like 'Livestock', where the user needs to be able to navigate around their herd with as little confusion as possible, coloured icons were used to indicate the location of areas of potential interest.

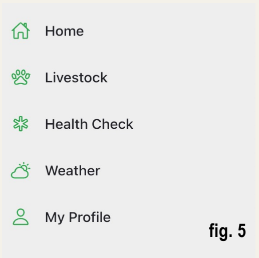


fig. 5

Building on this, spacing, iconography, and label visibility were all optimised to accommodate users who may not be comfortable reading lengthy digital text or navigating nested menus. Wherever possible, plain language was used instead of technical jargon—e.g., using "bad scours" instead of "profuse diarrhoea"—to ensure terminology remained familiar and regionally appropriate. Button sizes and tap targets were deliberately made large to suit use in rugged environments, and the app included a text modifier so that people with reading difficulties can modify the scale of the content on the screen to their liking [fig. 6].

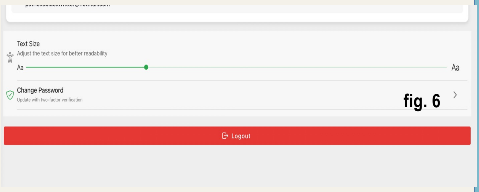
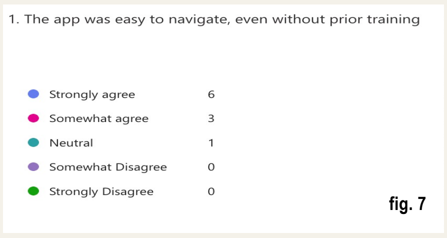

Text Size
Adjust the text size for better readability
Aa
Aa
Change Password
Update with two-factor verification
Logout

fig. 6

During testing, the UI's effectiveness was reinforced through survey feedback, which highlighted that users felt confident navigating the app without needing external guidance [fig. 7]. As a result, usability enhancements were driven by real-world usage, with features like dropdowns, autocomplete symptom entry, and confirmatory dialog boxes being refined to reduce the likelihood of user error. This approach ensured that the UI not only served a functional purpose but supported trust and confidence in the broader system.



1. The app was easy to navigate, even without prior training

● Strongly agree	6
● Somewhat agree	3
● Neutral	1
● Somewhat Disagree	0
● Strongly Disagree	0

fig. 7

Evaluation and Results

The system was evaluated through a combination of scenario-based testing and user feedback surveys. Six realistic diagnostic scenarios were developed, reflecting common health issues in Irish cattle herds. These were manually input into the application, and in all cases, the correct disease appeared within the top two suggestions—validating the accuracy of the AI scoring logic.

Ten farmers also participated in a usability survey hosted on Microsoft Forms. This survey was conducted to assess the usability and relevance of the application. Farmers were asked to rate their experiences while doing various tasks within the app, such as "The information provided by the app was clear and easy to understand". This feedback was crucial in the continuous development of the application, as is all feedback when developing with an agile methodology.

Conclusion

This project successfully demonstrates how mobile-first, offline-enabled technology can bridge the gap between veterinary expertise and real-time farm decision-making. The rule-based diagnostic engine, paired with livestock tracking and vaccination management, aims to provide a valuable support tool for farmers, especially those working in low-connectivity rural environments.

By focusing on usability, domain-specific logic, and contextual accuracy, the system allows users to make faster, more confident health diagnoses. The strong validation outcomes and positive user feedback suggest this approach would be greatly appreciated as it is further integrated into the agriculture world.

Acknowledgements

Andrew Beatty – My Project Supervisor: For his continuous reassurance throughout the development of this project

John French – Module Coordinator: For his continuous assistance whenever a problem arose

All the students and staff at ATU that made learning here a pleasure