IT Group Project Phase 2: Planning

Group 3

Masego Khola

Vuyokazi Mooki

Kelebogile Maema

Mellomey Dzaklidzie

Lebogang Mamabolo

Tshepang Ngulube

Relesego Shabangu

Neo Phofu

Clerence Ngoepe

Bongane Maluleka

Metchis Mathome

2.1. Identification of need

Overview

The identification of need focuses on understanding the underlying reasons for implementing a farm management system. This document outlines the key challenges faced by modern farming practices and how an efficient farm management system will address these challenges, offering streamlined processes, improved decision making, and better resource management

Challenges Faced by Farms:

Resource Inefficiency

Many farms struggle with resource allocation, particularly in managing water, fertiliser, pesticides, and machinery. Without a system in place to monitor and predict usage, farms may experience waste, which leads to increased costs and environmental harm.

Lack of Data-Driven Decision Making

Farms often rely on traditional methods to make decisions regarding crop cycles, irrigation, and livestock management. Without real-time data, decisions may be poor, leading to reduced yields and lower profitability.

Inventory and Supply Chain Issues

Managing the stock of inputs such as seeds, fertiliser, and machinery parts is critical. Farms lacking a system to manage inventory are at risk of stockouts or overstocking, which can disrupt operations or lead to wasted resources.

Difficulty in Tracking Farm Activities

The diverse activities that happen on a farm—ranging from planting, irrigation, fertilisation, harvesting, and distribution—are often challenging to track. This is true for large scale operations or multi-site farms.

Regulatory Compliance and Reporting

Farms are increasingly required to adhere to strict environmental, health, and safety regulations. Proper documentation and reporting are needed for audits, inspections, and certifications. A lack of proper systems increases the likelihood of non-compliance.

Crop Yield and Production Planning

Without forecasting tools and historical data, farms may struggle with planning crop cycles, rotation, and field management, which can negatively impact yields and soil health over time.

Conclusion:

The Farming management aims to address inefficiencies in resource utilisation, productivity,reporting and financial management. It also seeks to provide farmers with better tools for decision-making, market competitiveness, and technological advancement in their operations.

By integrating various aspects of farm management into a single platform and leveraging advanced technologies like IoT and data analytics, the system has the potential to transform farming operations, making them more efficient, productive, and profitable.

2.2. Preliminary Investigation

This summary sums up the key points from our preliminary investigation of the Farming Management System. A brief overview of the main takeaways:

- **Budget**: The development budget of R616,000 is comprehensive, but long-term operational costs should be considered.
- **User Adoption**: The project addresses real needs but may face challenges in user adoption and system integration.
- **Timeline**: The proposed 10-week timeline is ambitious, a more detailed schedule is recommended.
- **Data Considerations:** Data privacy, security, and regulatory compliance are critical considerations that need to be addressed.
- **Risks and Benefits**: The project faces risks, including data security, technology dependence, and integration challenges, but also offers potential benefits in terms of optimised operations.

To move forward, we'll conduct a more detailed feasibility study, engage with potential users, develop a risk management plan, and create a more detailed project schedule.

2.3. Feasibility Study (Technical,Operational, Economical)

Technical Feasibility

We need to simplify the technology choices. The mix of programming languages (Node.js, Python, PHP) might be confusing . It's better to stick to one language (choosing either Node.js or Python for the backend not both). Technical risks include integration complexity, scalability challenges, and data security concerns, Protecting farmers' security is crucial.

Operational Feasibility

The system can work in real farm settings, but we need to convince farmers to use it and make sure it fits with their current practices. It should also work reliably, even with poor internet connections.

Economical Feasibility

The initial budget appears to be reasonable, however we need to plan for long-term costs and figure out how to price the system for farmers. We should show how the system can save farmers money to prove it's a good investment.

Conclusion

The Farming Management System project appears feasible, but its success will depend on addressing several key challenges:

- Optimising the technologies for better integration and scalability.
- Developing a strong strategy for user adoption and training.
- Creating a clear long-term financial model, including a revenue strategy and total cost of ownership analysis.
- Implementing robust security measures to protect sensitive farm data.
- Ensuring the system can integrate smoothly with a variety of existing farm practices and technologies.

2.4. Project Planning

The project is scheduled to run for 10 weeks, from August 19 to October 25, 2024, with a budget of R616,000.

1. The plan is divided into six main phases:
    - Project Initiation and Planning
    - Requirements Gathering and Analysis
    - System Design
    - Development
    - Testing and Quality Assurance
    - Deployment, Training and Project Closure

Project Timeline

Milestone	Date
Project Kick-off	19 August 2024
Sprint 1: Requirement gathering	22 August - 28 August
Sprint 2: System Design	30 August - 11 September
Sprint 3: Development Phase 1	12 September - 18 September
Sprint 4: Development Phase 2	12 September - 18 September
Sprint 5: Testing and Quality Assurance	19 September - 25 September
System Deployment and Training	26 September - 9 October
Project close Out	25 October 2024

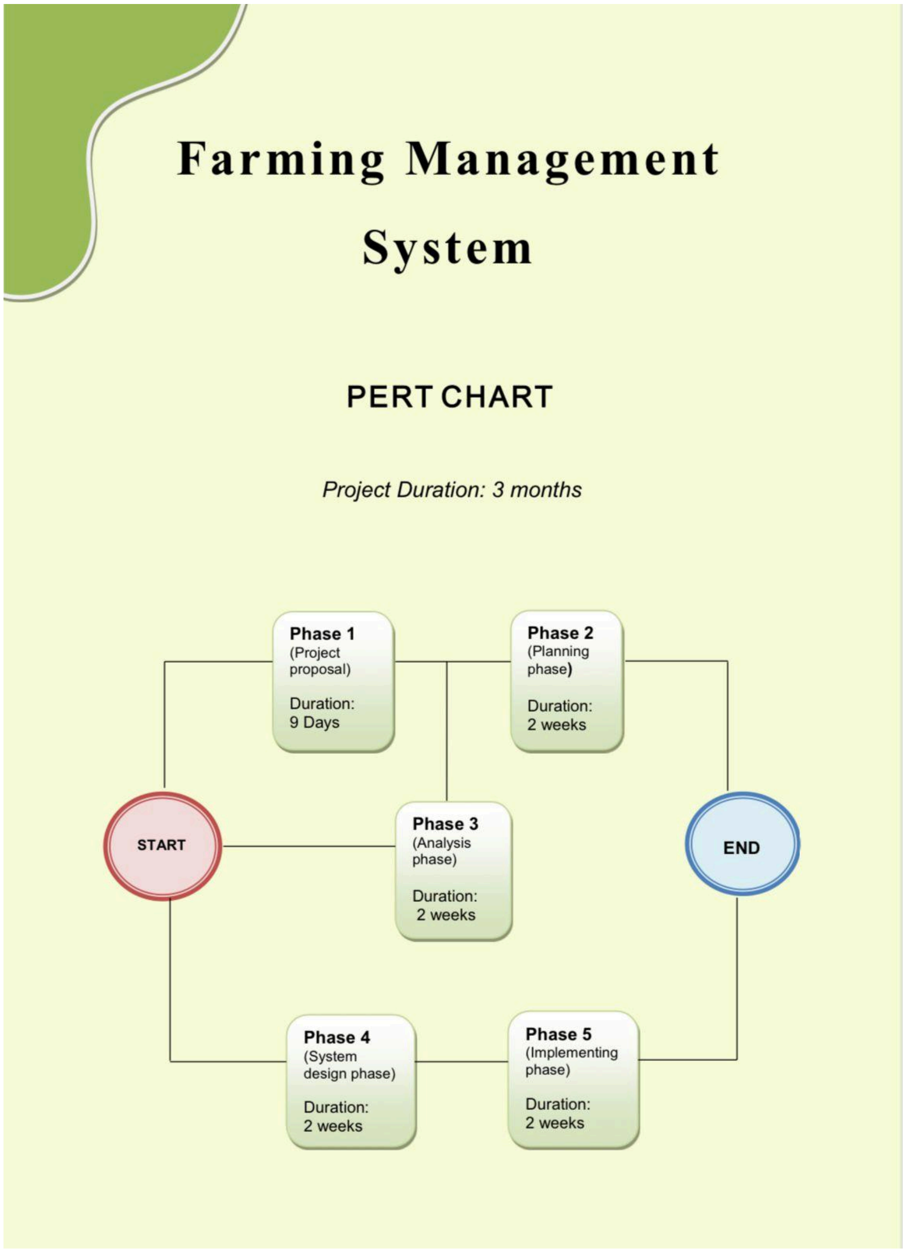
2. Key deliverables include various documentation and adherence to the project schedule.
3. The plan allocates specific roles for the project, including a Project Manager, developers, system administrators, technical lead,business analyst and quality assurance engineer

Roles	Assignment	Responsibilities
Developers, Designers	Clerence Ngoepe Bongane Maluleka Masego Khola Lebogang Mamabolo Metchis Mathombe Tshepang Ngulube	Responsible for building the system.
Business Analyst	Mellomey Dzaklidzie Tshepang Ngulube Vuyokazi Mooki	Gathers requirements and ensures alignment with business needs.
Quality assurance engineer	Relesego Shabangu Neo Phofu	Ensures that the software meets the required quality standards. Writes and executes test cases, identifies bugs, and works with the development team to resolve issues.
System administrator	Relesego Shabangu	Manages the IT infrastructure and ensures that the development and production environments are stable, secure, and scalable.
Project manager	Kelebogile Maema	Oversees the project, ensuring it is completed on time, within scope, and on budget. Manages the project team, and coordinates activities
Technical lead	Bongane Maluleka	Defines the technical architecture of the project, makes key technical decisions, and ensures that the technical team adheres to the architectural guidelines

4. Risk management, communication planning, and quality assurance are integrated throughout the project lifecycle.
5. The plan also includes provisions for user training to ensure smooth adoption and ongoing improvement of the system.

2.5. Project Scheduling (PERT Chart and Gantt Chart)

PERT Chart



Farming Management System

PERT CHART

Project Duration: 3 months

START

Phase 1 (Project proposal)
Duration: 9 Days

Phase 2 (Planning phase)
Duration: 2 weeks

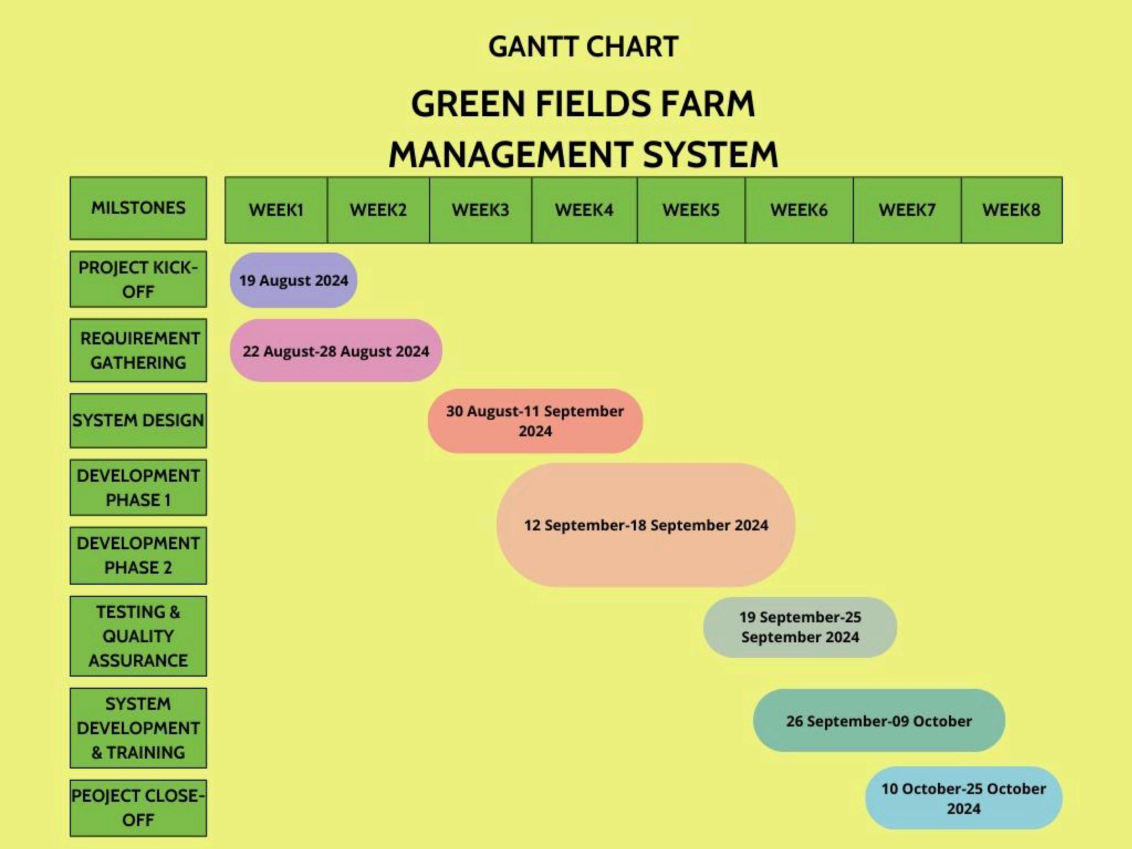
Phase 3 (Analysis phase)
Duration: 2 weeks

Phase 4 (System design phase)
Duration: 2 weeks

Phase 5 (Implementing phase)
Duration: 2 weeks

END

Gantt Chart



GANTT CHART

GREEN FIELDS FARM MANAGEMENT SYSTEM

MILSTONES	WEEK1	WEEK2	WEEK3	WEEK4	WEEK5	WEEK6	WEEK7	WEEK8
PROJECT KICK-OFF	19 August 2024							
REQUIREMENT GATHERING	22 August-28 August 2024							
SYSTEM DESIGN			30 August-11 September 2024					
DEVELOPMENT PHASE 1				12 September-18 September 2024				
DEVELOPMENT PHASE 2								
TESTING & QUALITY ASSURANCE						19 September-25 September 2024		
SYSTEM DEVELOPMENT & TRAINING						26 September-09 October		
PEOJECT CLOSE-OFF							10 October-25 October 2024	

2.6. Software Requirement Specification

1. Introduction

1.1 Purpose

This section specifies the software requirements for the Farming Management System (FMS), a comprehensive digital platform designed to enhance efficiency, productivity, and profitability in farming operations.

1.2 Scope

The FMS will integrate various aspects of farm management including crop and livestock management, financial tracking, inventory control, labour management, data analytics, and reporting. It will be accessible via web and mobile platforms and will integrate with IoT devices for real-time data collection.

1.3 Definitions, Acronyms, and Abbreviations

- **FMS**: Farming Management System
- **IoT**: Internet of Things
- **UI**: User Interface
- **API**: Application Programming Interface
- **SaaS**: Software as a Service

1.4 References

- IEEE Standard 830-1998: IEEE Recommended Practice for Software Requirements Specifications

1.5 Overview

The sections of this document provides a detailed description of the FMS functionality, interfaces, performance requirements, design constraints, and quality attributes.

2. Description

2.1 Product Perspective

The FMS is a new, self-contained SaaS product. It will integrate with IoT devices and may interface with existing farm equipment and systems through APIs.

2.2 Product Functions

- Crop planning and management
- Livestock tracking and health management
- Financial management and reporting
- Inventory and equipment management
- Labour management and scheduling
- Real-time monitoring through IoT integration
- Reporting and dashboard customization

2.3 User Classes and Characteristics

- **Farm Owners/Managers**: Full system access
- **Farm Workers:** Limited access based on assigned roles
- **Accountants/Financial Managers**: Access to financial modules
- **Agronomists/Crop Specialists**: Access to crop management and analytics
- **Livestock Managers:** Access to livestock modules
- **System Administrators**: Full access for system maintenance

2.4 Operating Environment

- **Web application:** Compatible with latest versions of Chrome, Firefox, Safari, and Edge
- **Mobile application**: iOS 13+ and Android 9+
- **Backend**: Cloud-hosted with capability for limited offline functionality
- **Database**: Relational database (e.g., PostgreSQL) with possible NoSQL integration for sensor data

2.5 Design and Implementation Constraints

- Must comply with local and international agricultural data protection regulations
- Should operate in areas with limited internet connectivity
- Must integrate with agricultural IoT devices
- User interface must be intuitive and accessible to users with varying levels of technical proficiency

2.6 User Documentation

- Comprehensive online help system
- Printable user manual
- Video tutorials for key features
- Quick start guide for new users
- Regular webinars for advanced features and best practices

2.7 Assumptions and Dependencies

- Users have basic computer/smartphone literacy
- Farms have at least intermittent internet connectivity
- IoT devices are compatible with standard protocols (e.g., MQTT)

3. Specific Requirements

3.1 External Interface Requirements

3.1.1 User Interfaces

- Responsive web interface adapting to desktop, tablet, and mobile screens
- Native mobile applications for iOS and Android
- Accessibility features compliant with WCAG 2.1 guidelines

3.1.2 Hardware Interfaces

- Compatibility with RFID livestock tracking systems
- Interface with weather stations
- Possible integration with automated farming equipment

3.1.3 Software Interfaces

- RESTful API for third-party integrations
- Integration with popular accounting software (e.g., QuickBooks, Xero)
- Weather data API integration
- Possible integration with government agricultural databases
- Export functionality to common formats (CSV, PDF, Excel)

3.1.4 Communications Interfaces

- HTTPS for secure web communication
- MQTT protocol for IoT device communication
- WebSocket for real-time updates in web and mobile apps

3.2 Functional Requirements

3.2.1 Crop Management Module

- Create, edit, and delete crop plans
- Track crop growth stages
- Generate alerts for scheduled farming activities
- Calculate estimated yield based on current conditions and historical data

3.2.2 Livestock Management Module

- Maintain individual animal records with health history
- Track animal lineage and breeding records
- Manage vaccination and treatment schedules with automated reminders
- Monitor feed inventory and consumption rates
- Generate alerts for health checks and other livestock-related activities

- Integrate with RFID systems for automated animal tracking

3.2.3 Financial Management Module

- Record and categorise income and expenses
- Generate financial reports (profit/loss)

3.2.4 Inventory Management Module

- Track inventory of supplies with real-time updates
- Manage equipment maintenance schedules and service records
- Barcode/QR code scanning for easy inventory updates

3.2.5 Labor Management Module

- Create and assign tasks to workers with priority levels
- Track work hours, breaks, and overtime
- Measure and report on worker productivity
- Manage shift schedules with conflict checking
- Support payroll calculation based on hours worked and pay rates
- Provide a mobile check-in/check-out system for workers

3.2.6 Data Analytics and Reporting Module

- Provide customizable dashboards with key performance indicators
- Generate comprehensive reports on all aspects of farm operations
- Support data visualisation with charts and graphs
- Allow for custom report creation with a user-friendly interface

3.3 Non-Functional Requirements

3.3.1 Performance Requirements

- Support at least 1000 concurrent users without performance degradation
- Ensure all pages load within 2 seconds under normal network conditions

- Process and display IoT data with a maximum delay of 30 seconds
- Support farms with up to 10,000 livestock records or 10,000 hectares of crops
- Perform database backups without system downtime

3.3.2 Safety Requirements

- Validate all data inputs to prevent errors and data corruption
- Provide clear warnings before executing irreversible actions
- Implement a rollback mechanism for critical data changes
- Ensure that critical alerts (e.g., livestock health emergencies) are delivered through multiple channels

3.3.3 Security Requirements

- Implement multi-factor authentication for user accounts
- Maintain detailed audit logs of all system actions
- Ensure compliance with General Data Protection Regulation (GDPR) and other relevant data protection regulations
- Perform regular automated security scans and penetration testing

3.3.4 Software Quality Attributes

- Achieve 99.9% system availability (excluding scheduled maintenance)
- Ensure scalability to accommodate farms from small family operations to large industrial farms
- Develop with a modular architecture to allow for easy updates and addition of new features
- Provide a consistent user experience across web and mobile platforms
- Support internationalisation and localization for major agricultural regions

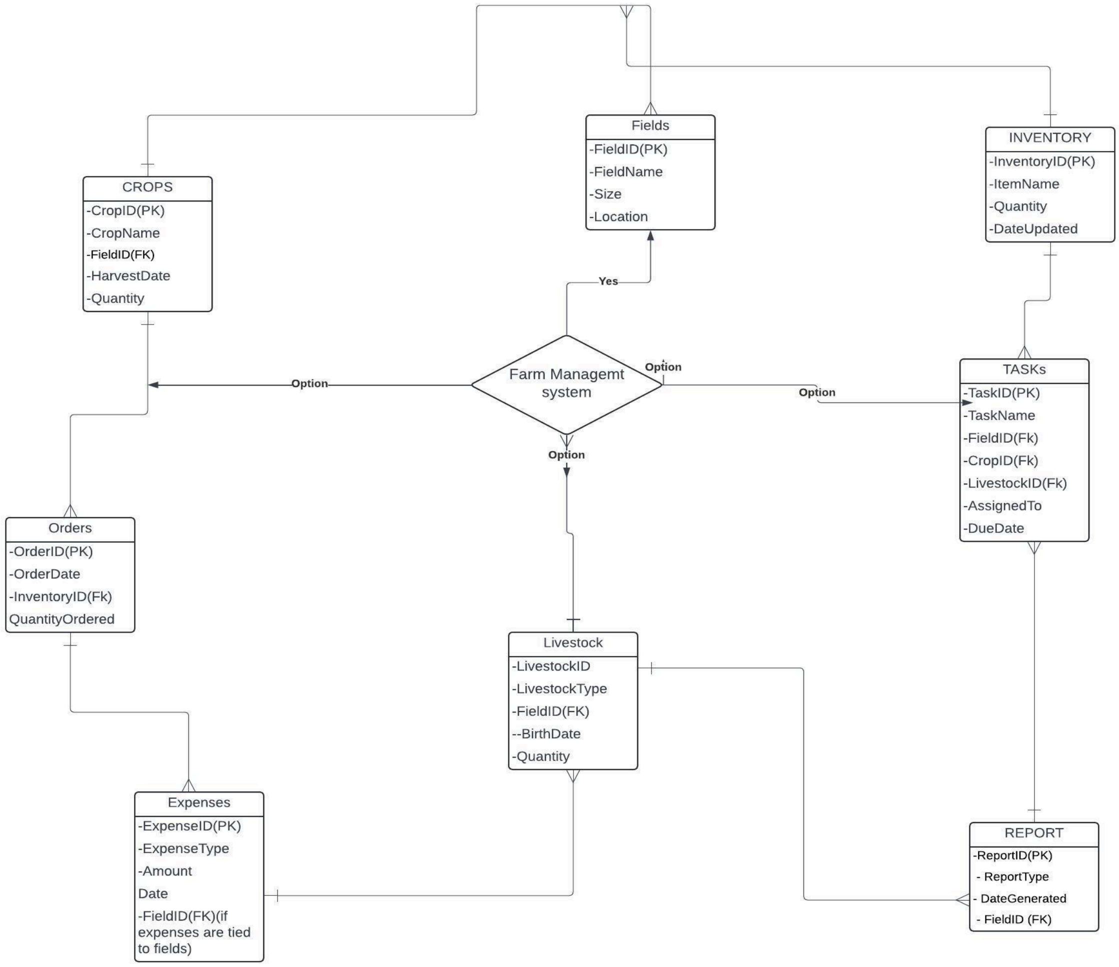
3.3.5 Business Rules

- Comply with local and international agricultural regulations
- Adhere to data protection and privacy laws in all operating regions
- Ensure all financial calculations comply with international accounting standards

2.7. Data Models

The ERD diagram covers the main entities and relationships described in the Farming Management System project proposal.

ERD Diagram


Fields
-FieldID(PK)
-FieldName
-Size
-Location
INVENTORY
-InventoryID(PK)
-ItemName
-Quantity
-DateUpdated
CROPS
-CropID(PK)
-CropName
-FieldID(FK)
-HarvestDate
-Quantity
Yes
Farm Managemt system
Option
Option
Option
Option
TASKs
-TaskID(PK)
-TaskName
-FieldID(Fk)
-CropID(Fk)
-LivestockID(Fk)
-AssignedTo
-DueDate
Orders
-OrderID(PK)
-OrderDate
-InventoryID(Fk)
QuantityOrdered
Livestock
-LivestockID
-LivestockType
-FieldID(FK)
--BirthDate
-Quantity
Expenses
-ExpenseID(PK)
-ExpenseType
-Amount
Date
-FieldID(FK)(if expenses are tied to fields)
REPORT
-ReportID(PK)
- ReportType
- DateGenerated
- FieldID (FK)