ISE 307

Database System

Agritrack

Members’

Name	ID	CRN
Korkut Kaan Balta	150190203	21322

Date of Delivery
02.06.2024

Final Project Report

1.0 Introduction

1.1 Motivation / Goal

Our main goal is to develop an intuitive and comprehensive app to help farmers, empowering them to efficiently monitor and manage the organization of their animals. By enabling farmers to effortlessly record crucial data such as animal weights, tag numbers, dietary details, and associated costs, our app aims to streamline the tracking process. This tool will provide farmers with valuable insights into their animals' health, growth, and expenses. To sum up, our goal is to offer a user-friendly platform that simplifies the intricate task of animal management, empowering farmers to focus more on the well-being and productivity of their livestock.

1.2 Content

This report provides a detailed overview of the Agritrack project, including the system request, feasibility studies, project plan, risk assessments, requirements definition, use cases, process models, logical data model, interface design, and program design. Each section is divided into subsections to comprehensively address various aspects of the project.

1.3 Organization of the Document

1.4 Revision Log

- Feasibility Study: study cash flow diagram was added to the economic feasibility
- Project Plan
  - Project work plan is re-visited and added more specific tasks along with their id numbers and dependencies
  - Project work plan assignments are made more specific
  - In the staffing plan, the necessary skills for each role have been thoroughly examined
  - Added risks #3 through #7
- Use Cases : Use case 5 and 6's step numbers are fixed
- Process Models: Context Diagram is added
- Interface Design: Background color of the page is changed to white
- Physical Data Storage Design added relations added
  - For the Ration table a couple of junction tables were added (livestock_rasyon ,rasyon components, rasyon component history).
- Architecture Design added.

2.0 System Request

2.1 Project sponsor:

Korkut Kaan Balta, Mert Can Vural

2.2 Business Need:

Address the increasing demand for innovative solutions that enhance farming operations' efficiency and productivity.

- Develop a comprehensive livestock Enterprise Resource Planning (ERP) system, "AgriTrack."
- Revolutionize farm management for livestock farmers and related businesses.
- Integrate a wide range of functionalities to financial accounting into a single platform.
- Enhance decision-making, streamline processes, and elevate overall operational efficiency.
- Transition from currently used record-keeping methods, Excel sheets, to a centralized web-based platform to improve visualization and efficiency in managing financials and livestock records.

2.3 Business Requirements:

- **Technical Complexity:** Sophisticated software integration is needed for the livestock ERP solution.
- **Data Accuracy and Integrity:** The system requires precise data input to ensure reliability.
- **User Adoption and Training:** The system must be adopted by users, necessitating training, especially for those not tech-savvy.
- **Integration with Existing Systems:** The new ERP must be compatible with any systems already in use by farmers.

- **Software Quality and Performance:** The ERP system needs to be robust, user-friendly, and perform well under various conditions.

2.4 Business Value:

The requirements of the successful deployment of the ERP system would significantly improve the efficiency and decision-making process in livestock management.

- By centralizing data on a web-based platform, the errors in the data entry process will be eliminated and the operational efficiency will increase by an estimate of 15% within the first year of implementation.
- Improved livestock management through enhanced data analysis capabilities is expected to increase livestock health and this will lead to long-term sustainability of the business.

2.5 Special Issues or Constraints:

- Fixed budget constraints could lead to overspending if unexpected costs arise.
- Potential for delays due to technical issues, task underestimation, or resource unavailability.
- The ERP must adhere to agricultural and software regulations.
- The project's success is dependent on stakeholder commitment, which could change and pose risks to the project's continuity and success.

3.0 Feasibility study

3.1 Technical Feasibility

Familiarity with Application: The developers working on this project have experience building small programs, but their familiarity with web development poses a medium risk due to its complexity and differences from traditional software development.

Familiarity with Technology: Selecting the right technologies for building the web-based ERP system is crucial. While certain technologies may ease the development process, navigating their complexities is highly important to ensure the project's success.

Project Size: With a team size of 10 people, the project size is manageable.

Compatibility: The proposed project, AgriTrack, must be compatible with the existing tools and formats used by farmers, such as Excel sheets for record-keeping. Implementing features like import/export functionalities will integrate with farmers' current data formats.

Integration Requirements: AgriTrack requires sophisticated software integration to create a comprehensive livestock ERP system, combining various functionalities like financial accounting into one platform. Data accuracy and integrity are important. To ensure software quality and performance, it requires leveraging the right technologies.

3.2 Economic Feasibility

A detailed economic feasibility study has been conducted to evaluate the potential financial outcomes of implementing an ERP system for livestock management. Based on conservative estimates, the project demonstrates promising economic viability.

3.2.1 Financial Metrics:

- **Return on Investment (ROI) over 3 years:** Estimated at approximately 97.16%, indicating a strong potential for profitability relative to the initial investment.
- **Net Present Value (NPV) over 3 years:** Calculated at $400,165, suggesting that the discounted future cash flows from the project exceed its costs.
- **Break-even Point:** Expected to occur within 3 years, indicating that the project will start generating profit beyond this point.

3.2.2 Cash Flow Diagram

	Year 0	Year 1	Year 2	Year 3
Total Benefits	$0	$100,000	$250,000	$300,000
Total Costs	$200,000	$270,000	$130,000	$105,000
Net Benefits	$200,000	$170,000	$120,000	$195,000
Cumulative Net Cash Flow	$200,000	$370,000	$250,000	$55,000

Tangible Costs

- **Software Development:** Cost of programming, testing, and deploying the software, including developer salaries and software licenses.
- **Hardware:** Expenses for servers, workstations, and any necessary networking equipment.
- **Integration:** Fees for integrating existing systems with the new ERP, including middleware or custom development.
- **Training:** Costs for training staff and end-users on the new system.
- **Maintenance:** Ongoing support and maintenance expenses, including updates and security patches.
- **Operational:** Costs related to hosting, electricity, and other utilities for running the ERP system.

Intangible Costs

- **Change Management:** Efforts required to manage organizational change and ensure user adoption.
- **Opportunity Cost:** Potential benefits lost from not investing resources elsewhere.
- **Project Failure Risk:** Potential for the project to exceed budget or fail to meet objectives, impacting future investments.
- **Compliance and Regulatory Risks:** Costs associated with ensuring the system meets all legal and industry standards.
- **Reputation:** Potential impact on the organization's reputation if the project fails or performs poorly.

Tangible Benefits

The AgriTrack project promises significant tangible benefits, including cost savings and revenue growth. Initial costs encompass software development, hardware (if required), and integration with existing systems. Ongoing expenses involve maintenance, hosting for the web-based platform, and user support. Revenue projections indicate potential improvements in operational efficiency by 15% in the first year, leading to enhanced livestock health management and cost savings. These improvements are expected to translate into higher revenues for users, driving adoption and generating tangible financial gains over the project's timeline.

Intangible Benefits

In addition to tangible benefits, AgriTrack offers various intangible advantages crucial for long-term success. These include enhanced strategic planning capabilities, increased user satisfaction, improved data management, market competitiveness, and greater user engagement. As the project progresses, intangible benefits like industry leadership and recognition for innovation are anticipated, further bolstering the project's overall impact and contributing to its success.

Cost Estimation: Initial costs include software development, hardware (if any), and integration with existing systems. Ongoing costs involve maintenance, hosting (for a web-based platform), and user support.

Revenue Projections: Benefits include improved operational efficiency, estimated to increase by 15% in the first year, and better livestock health management. These improvements can lead to cost savings and potentially higher revenues for users, encouraging adoption.

3.2.3 Financial Overview (2022-2025)

- **2022:**
  - Tangible Costs: $150,000
  - Intangible Costs: $50,000
  - Tangible Benefits: $0
  - Intangible Benefits: Enhanced strategic planning capabilities
- **2023:**
  - Tangible Costs: $200,000
  - Intangible Costs: $70,000
  - Tangible Benefits: $100,000
  - Intangible Benefits: Increased user satisfaction and data management
- **2024:**
  - Tangible Costs: $100,000
  - Intangible Costs: $30,000
  - Tangible Benefits: $250,000
  - Intangible Benefits: Market competitiveness, user engagement
- **2025:**
  - Tangible Costs: $80,000
  - Intangible Costs: $25,000
  - Tangible Benefits: $300,000
  - Intangible Benefits: Industry leadership, innovation recognition

3.4 Organizational Feasibility

The AgriTrack project carries a moderate level of organizational risk.

Organizational Management: The proposed project is supported to bring visualization and better financial controls. However, entrenched farming practices may pose resistance to adopting new methods of financial and livestock record keeping, potentially hindering the adoption of the project’s ERP system.

Farmer Acceptance: Farmer adoption and integration of the AgriTrack system are crucial for its success. Efforts must be directed towards ensuring comprehensive user training and user-friendly system design principles. Without thorough farmer adoption, the project's viability could be compromised.

Stakeholder Commitment: The success of the AgriTrack project hinges on the commitment of stakeholders, including project sponsors and end-users. Ensuring their continued involvement and support throughout the project lifecycle is essential for mitigating organizational risks and maximizing project success.

4.0 Project Plan

4.1 Project Methodology

The Agritrack project is guided by an agile methodology, a dynamic approach that prioritizes adaptability and collaboration. Agile principles emphasize face-to-face communication, enabling swift decision-making and alignment among team members, stakeholders, and clients. This direct interaction fosters a shared understanding of project goals and requirements, facilitating rapid problem-solving and ensuring that the project progresses smoothly. Additionally, the iterative nature of agile development breaks down the project into manageable parts, allowing the team to deliver incremental improvements in short cycles. This iterative process not only provides stakeholders with early opportunities to provide feedback but also enables the team to respond quickly to changing requirements and priorities, ultimately enhancing the project's success.

By embracing agile methodology, the Agritrack project benefits from increased flexibility and efficiency. The iterative approach to development enables the team to continuously refine and improve the software, incorporating feedback and addressing issues as they arise. Moreover, the collaborative nature of agile methodologies encourages cross-functional teamwork and empowers team members to take ownership of their work. This collaborative environment fosters creativity, innovation, and shared accountability, leading to higher levels of engagement and motivation within the team. Overall, the adoption of agile methodology ensures that the Agritrack project can deliver value to stakeholders in a timely and effective manner, while also fostering a culture of continuous improvement and adaptability.

Main Task	Sub-Task	Task ID	Assigned To	Duration (Man.days)	Start Date	Finish Date	Dependencies
Project Management	Project Planning	1	Kaan	2	06/17/24	09/17/24	-
	Budget Management	2	Kaan	2	06/17/24	06/26/24	1
	Stakeholder Communication	3	Kaan	1	06/17/24	06/26/24	1
	Risk Management	4	Mert	2	06/17/24	06/26/24	1
Requirements Analysis	Gather Requirements	5	Kaan	2	06/17/24	06/26/24	1
	Analyze User Needs	6	Mert	2	06/27/24	07/06/24	5
	Define System Specifications	7	Mert, Kaan	3	06/27/24	07/06/24	6
Software Development	ERP System Architecture Design	8	Kaan	2	06/27/24	07/06/24	7
	Database Design & Implementation	9	Kaan	6	07/09/24	07/10/24	8
	Security Implementation	10	Mert	6	07/11/24	07/13/24	9
Backend Development	API Development	11	Mert, Kaan	28	07/15/24	07/15/24	10
	Integration of Data Management Systems	12	Mert	7	06/20/24	06/26/24	11
Frontend Development	UI/UX Design for ERP System	13	Mert	21	06/20/24	06/26/24	8
	UI/UX Design for Mobile App	14	Mert	21	06/20/24	06/23/24	13
Web Application Development	Web App Design	15	Kaan	44	06/24/24	06/26/24	13
	Integration with ERP Backend	16	Kaan	7	06/24/24	06/26/24	15
Testing & Quality Assurance	ERP System Testing	17	Mert, Kaan	14	06/24/24	06/26/24	16
	Performance Testing	18	Mert, Kaan	7	06/24/24	06/26/24	17
	User Acceptance Testing	19	Mert, Kaan	7	06/24/24	06/26/24	18
Deployment and Documentation	Deployment of ERP System and Mobile App	20	Kaan	42	06/24/24	06/26/24	19
	User Manuals and Documentation	21	Mert	4	06/24/24	06/26/24	20
	Training Sessions for End Users	22	Kaan	3	06/24/24	06/26/24	21
Project Closure	Final Reporting	23	Mert, Kaan	2	06/24/24	06/26/24	22
	Post Implementation on Review	24	Mert, Kaan	5	06/24/24	06/26/24	23

4.2 Staffing Plan

The staffing plan for the Agritrack project involves assembling a team of 10 people. The team will consist of the following roles:

1. **Project Manager**
   - **Skills:** Project management, scheduling, budgeting.
   - **Required:** 1
2. **Business Analyst**
   - **Skills:** Requirements gathering, stakeholder communication.
   - **Required:** 1
3. **Software Architect**
   - **Skills:** System design, technology evaluation.
   - **Required:** 1
4. **Database Administrator (DBA)**
   - **Skills:** Database design, optimization, security.
   - **Required:** 1
5. **Frontend Developer**
   - **Skills:** UI/UX design, HTML/CSS, JavaScript.
   - **Required:** 1
6. **Backend Developer**
   - **Skills:** API development, server-side programming.
   - **Required:** 1
7. **Security Specialist**
   - **Skills:** Data encryption, access control.
   - **Required:** 1
8. **QA Engineer**
   - **Skills:** Test planning, manual and automated testing.
   - **Required:** 1
9. **Technical Writer**
   - **Skills:** Documentation, user manuals.
   - **Required:** 1
10. **Support/Training Specialist**
    - **Skills:** User training, support.
    - **Required:** 1

4.3 Risk Assessments

RISK #1:

Farmers may not want to use the app because they are accustomed to their old tech.

Likelihood of risk:

The likelihood of this risk is medium to high.

Potential impact on the project:

- If farmers resist adopting the new app, it could hinder the project's success and delay its implementation.

- This resistance from farmers could lead to a slower adoption rate of the app, resulting in decreased usage and potentially lower satisfaction levels among users. It may also affect the project's financial viability if the anticipated benefits from increased efficiency and productivity are not realized.

Ways to address this risk:

- Highlight the benefits of using the app, such as increased efficiency, cost savings, and improved decision-making, to incentivize adoption. Use testimonials and success stories from early adopters to showcase the app's value.
- Conduct extensive user research and engage with farmers to understand their needs and concerns. Tailor the app's features and user interface to align with their preferences and workflows.
- Provide comprehensive training and support to ease the transition from old methods to the new app. Offer tutorials, workshops, and one-on-one assistance to address any issues or questions they may have.
- Continuously gather feedback from farmers and iterate on the app's design and functionality to address any pain points or usability issues. Ensure ongoing support and updates to maintain user satisfaction and engagement.

RISK #2:

Staff may have difficulty delivering the project in time because of their inexperience with the technologies used to deliver the project.

Likelihood of risk:

The likelihood of this risk is medium to high.

Potential impact on the project:

- If inexperienced staff members struggle to meet project deadlines, it could lead to delays in the project timeline, resulting in increased costs and potential budget overruns. Moreover, the quality of deliverables may be compromised, affecting the overall success and reputation of the project.

Ways to address this risk:

- Provide comprehensive training and mentorship programs for inexperienced staff members to enhance their skills and knowledge relevant to the project.
- Implement a robust project management framework with clear milestones, deadlines, and accountability measures to ensure that tasks are completed on time and within budget.
- Regularly monitor and assess the progress of inexperienced team members, offering guidance and support where necessary to address any challenges or obstacles they may encounter.
- Consider outsourcing certain tasks or seeking assistance from experienced consultants or contractors to complement the capabilities of the internal team and mitigate the risk of project delays or budget issues.

RISK #3:

Data Security Breach

Likelihood of Risk: Medium

Potential Impact on the Project:

- A data security breach could compromise sensitive information, leading to legal issues and a loss of trust among users.
- It could result in significant financial losses due to penalties, fines, and remediation costs.
- The project's reputation could be severely damaged, impacting future user adoption and overall success.

Ways to Address this Risk:

1. **Implement Strong Security Measures:**
   - Use encryption for data storage and transmission.
   - Apply multi-factor authentication for access control.
   - Regularly update and patch software to protect against vulnerabilities.
2. **Conduct Regular Security Audits:**
   - Perform routine security assessments and penetration testing to identify and address potential weaknesses.

    - Ensure compliance with industry standards and regulations (e.g., GDPR, HIPAA).
3. **Train Staff on Security Best Practices:**
    - Educate employees on the importance of data security and how to recognize phishing and other cyber threats.
    - Establish clear protocols for handling sensitive data.
4. **Develop an Incident Response Plan:**
    - Create a detailed plan outlining steps to take in the event of a data breach.
    - Regularly test and update the plan to ensure effectiveness

---

RISK #4:

Budget Overruns

Likelihood of Risk: Medium

Potential Impact on the Project:

- If the project experiences budget overruns, it could lead to a shortage of funds needed to complete the project.
- It may necessitate reductions in project scope or quality to stay within budget.
- The project's financial viability could be compromised, affecting stakeholder confidence and support.

Ways to Address this Risk:

1. **Implement Stringent Budget Management:**
    - Conduct regular financial reviews and updates.
    - Allocate contingency funds to cover unexpected expenses.
2. **Prioritize Project Tasks:**
    - Ensure essential components are completed within budget.
    - Adjust project scope if necessary.
3. **Negotiate with Vendors:**
    - Seek better rates or flexible payment terms.
4. **Monitor Expenditures:**
    - Track project costs closely and adjust plans to avoid overspending.

RISK #5:

Dependency on Third-Party Services

Likelihood of Risk: Medium

Potential Impact on the Project:

- Reliance on third-party services could lead to delays if these services are unavailable or experience issues.
- It might increase costs if third-party providers raise their prices or require additional fees.
- The quality and functionality of the project could be compromised if third-party services do not meet expectations or standards.

Ways to Address this Risk:

1. **Perform Thorough Due Diligence:**
   - Select reliable and high-quality service providers.
2. **Establish Clear Contracts and SLAs:**
   - Define expectations and responsibilities with third-party providers.
3. **Develop Contingency Plans:**
   - Prepare for switching to alternative providers if necessary.
4. **Monitor Performance:**
   - Regularly review third-party services and address issues promptly.
5. **Consider In-House Development:**
   - Develop critical functionalities internally to reduce dependency.

RISK #6:

Insufficient User Feedback

Likelihood of Risk: Medium

Potential Impact on the Project:

- If insufficient feedback is gathered from end-users, the system may not meet their needs effectively.
- Lack of user feedback can result in a product that is difficult to use or does not address critical issues, leading to low adoption rates.
- The project may miss key insights into usability and functionality improvements, potentially affecting user satisfaction and overall project success.

Ways to Address this Risk:

1. **Implement Regular Feedback Sessions:**
   - Schedule sessions with end-users to gather insights and address concerns.
2. **Create a Beta Testing Program:**
   - Release a beta version of the app to a select group of users for testing and feedback.

3. **Establish a Feedback Loop:**
    - Set up continuous feedback collection systems like in-app forms or a feedback portal.
4. **Engage with User Communities:**
    - Participate in online forums and social media groups to understand user needs and expectations.
5. **Iterate Based on Feedback:**
    - Incorporate user feedback into development iteratively and communicate updates to users.

RISK #7:

Integration Challenges with Existing Systems

Likelihood of Risk: Medium

Potential Impact on the Project:

- Integration with existing systems may be complex, leading to delays and increased costs.
- Incompatibilities could result in data inconsistencies and functional issues.
- The project might face technical difficulties that require significant time and resources to resolve.

Ways to Address this Risk:

1. **Conduct Thorough Analysis:**
    - Analyze existing systems for compatibility and integration requirements.
2. **Develop Integration Plan:**
    - Create a detailed plan outlining integration steps and potential issues.
3. **Use Standard Protocols:**
    - Utilize industry-standard protocols and APIs for seamless integration.
4. **Allocate Additional Resources:**
    - Assign experienced personnel to handle integration challenges.
5. **Test Thoroughly:**
    - Conduct extensive testing to identify and resolve issues early in the integration process.
6. **Provide Training:**
    - Train staff on integration procedures and tools to ensure smooth execution.

5.0 Requirements Definition

5.1 Functional Requirements

5.1.1 Livestock Management Module

FR 5.1.1.1 (High Priority): Herd Tracking Submodule - Develop functionalities for monitoring and managing livestock, including tracking each herd's location, weight, health status, and other relevant metrics. The system should update and display this information in real-time, with data retrieval within 2 seconds.

FR 5.1.1.2 (High Priority): Stock Management Submodule - Implement a feature for managing the stock of feed materials like straw, grains, etc. The system should track inventory levels, updating in real-time and allowing for inventory adjustments with quantities expressed in kilograms. Notifications for low stock levels should be sent out automatically.

FR 5.1.1.3 (High Priority): Meal Plan and Cost Analysis Submodule - Create a system feature to develop, apply, and manage special meal plans for different animal groups based on their nutritional needs and available feedstock. The system should calculate and update the daily cost of these meal plans per animal, incorporating this into the overall cost analysis, with updates and cost recalculations done within 3 seconds.

FR5.1.1.4 (High Priority): Slaughter Management Submodule - Implement a module to manage and track the process when animals are sent to the slaughterhouse. This includes recording the date of dispatch, weight, and other relevant details. The system should calculate the total profit for each animal by comparing the sales price with the accumulated cost data, updating and displaying this information within 3 seconds.

5.1.2 System Architecture Design

FR 5.1.2.1 (High Priority): Multi-User Support - Maintain an architecture supporting concurrent multi-user access with role-based actions, ensuring response times remain under 2 seconds.

FR 5.1.2.2 (High Priority): Integration Compatibility - Facilitate seamless integration with existing agricultural tools, databases, and specifically, ensure compatibility with Microsoft Excel formats for data import and export. The system should support batch data synchronization from Excel sheets within 5 minutes, minimizing manual data entry and ensuring data consistency.

FR 5.1.2.3 (Medium Priority): Audit Trail Framework - Design the system to log and audit all user actions, including modifications, access times, and identities, with secure, immutable logs retrievable within 4 seconds.

5.1.3 Database Design and Implementation

FR 5.1.3.1 (Medium Priority): Optimized Database Schema - Construct a database schema that reduces redundancy and enables efficient data management, targeting query response times not exceeding 3 seconds.

FR 5.1.3.2 (Medium Priority): Data Security and Recovery - Implement encryption for data at rest and in transit, and establish a disaster recovery plan to restore data within 5 minutes in case of loss.

5.2. Non-functional Requirements

2.1 Operational

NFR 5.2.1.1 (High Priority): System Reliability - Ensure the system achieves at least 99.9% uptime, excluding scheduled maintenance periods, to support continuous farm operations.

NFR 5.2.1.2 (High Priority): Availability - Design the system for high availability with a disaster recovery plan aiming for a Recovery Time Objective (RTO) of less than 4 hours and a Recovery Point Objective (RPO) of 24 hours.

NFR 5.2.1.3 (Medium Priority): Maintainability - Ensure the system allows for easy updates and maintenance without significant downtime, aiming for a system update downtime of no more than 1 hour monthly.

5.2.2 Performance

NFR 5.2.2.1 (High Priority): Response Time - The system should ensure a maximum response time of 2 seconds for user interactions under normal operational conditions.

NFR 5.2.2.2 (Medium Priority): Data Processing Speed - Ensure batch data processing, such as importing from Excel or updating livestock information, is completed within 5 minutes.

5.2.3 Security

NFR 5.2.3.1 (High Priority): Data Encryption - All stored and transmitted data must be encrypted using industry-standard methods to protect against unauthorized access.

NFR 5.2.3.2 (High Priority): User Authentication and Authorization - role-based access controls to ensure that only authorized users can access sensitive information and functionality.

NFR 5.2.3.3 (Medium Priority): Audit Trails - Securely store audit logs for at least one year and ensure they are tamper-proof to support compliance and investigative activities.

5.2.4 Cultural and Political

NFR 5.2.4.1 (Medium Priority): Usability - The user interface should be intuitive and accessible to users with varying levels of technical expertise, aiming for a learning curve that allows basic proficiency within one hour of training.

NFR 5.2.4.2 (Medium Priority): Localization - need to support Turkish and English

NFR 5.2.4.3 (Low Priority): Environmental Impact - Design the system to minimize energy consumption and reduce its carbon footprint, supporting the farm's sustainability goals.

6.0 Use cases

6.1 Monitor Livestock Status

<table>
<tr><td colspan="2">Use Case Name: Monitor Livestock Status</td><td>ID: _01</td><td colspan="2">Importance Level: High</td></tr>
<tr><td colspan="5">Primary Actor: Farmer</td></tr>
<tr><td colspan="5">Short Description: Monitoring status of livestock based on the farmer's needs.</td></tr>
<tr><td colspan="5">Trigger: A farmer needs to check status and other metrics of their livestock.
Pre Conditions:
1. Farmer must be registered to the Agritrack system.
2. Farmer must be connected to the internet.
3. The Agritrack system must be available and online.
Type: External / Temporal</td></tr>
<tr><td colspan="4">Major Steps Performed
1. Farmer logs in to the system.
2. Opens the livestock monitoring module.
3. Farmer selects a specific animal or herd for detailed status.
4. The system retrieves and displays the selected livestock's weight status, location, daily weight rate, and other metrics.
5. The farmer reviews the detailed information provided by the system.
Post Conditions:
1. Farmer has up-to-date information on the health and status of their selected livestock.</td><td>Information for Steps
→ Mail and Password
→ Selection of specific animal or herd
← Health status, location, weight, etc., of the selected livestock</td></tr>
<tr><td>Major Inputs:
Description Source
- Mail and Password farmer
- Filters farmer</td><td colspan="4">Major Outputs:
Description Destination
-Health status, location, weight, etc., of the selected livestock - farmer</td></tr>
</table>

6.2 Manage Feed Stock

Use Case Name: Manage Feed Stock	ID: _02	Importance Level: High
Primary Actor: Farm owner, Feed manager		
Short Description: Managing the stock of feed materials like straw and grains, ensuring optimal inventory levels.		

Trigger: The Feed Manager needs to update or check the stock levels of feed materials.

Pre Conditions:

1. The Feed Manager must be registered to the Agritrack system.
2. The Feed Manager must be connected to the internet.
3. The Agritrack system must be available and online.

Type: External / Temporal

Major Steps Performed 1. The Feed Manager logs into the system. 2. Accesses the ‘Stock Management’ submodule. 3. Reviews current stock levels of various feed materials. 4. Updates stock quantities or sets alerts for low stock levels. 5. Confirms the updates and adjustments made. Post Conditions: 1. Updated feed material stock levels are accurately reflected in the system.	Information for Steps → Mail and Password → Feed material stock adjustments ← Confirmation of stock level updates and alert settings

Major Inputs:

Description	Source
- Mail and Password	Farm Owner
- Stock adjustments	Farm Owner

Major Outputs:

Description	Destination
Updated stock levels	-Farm Owner
Low stock alert	

6.3 Calculate and Update Meal Plan Costs

Use Case Name: Calculate and Update Meal Plan Costs	ID: 03	Importance Level: High

Primary Actor: Farm Owner

Short Description: Updating and calculating the daily cost of meal plans per group.

Trigger: Changes in feed stock levels or prices require updates to meal plan costs.

Pre Conditions:

1. Farmer must be registered to the Agritrack system.
2. Farmer must be connected to the internet.
3. The Agritrack system must be available and online.

Type: External / Temporal

Major Steps Performed	Information for Steps
1. The Financial Manager logs into the system. 2. Navigates to the ‘Meal Plan and Cost Analysis’ submodule. 3. Selects a meal plan to review or update. 4. Inputs new cost factors (e.g., feed price changes). 5. The system recalculates the daily cost per animal based on updates. 6. The Financial Manager confirms the updated meal plan costs.	→ Mail and Password → Updated cost factors ← Recalculated meal plan costs

Post Conditions:

1. Meal plans reflect the most current costs based on feed prices and quantities.

Major Inputs:

Description	Source
Mail and Password	- Farm Owner
Updated cost factors	

Major Outputs:

Description	Destination
Updated meal plan costs	- Financial Records,
-	-Meal Plan Database

6.4 Slaughter Management and Profit Calculation

Use Case Name: Slaughter Management and Profit Calculation	ID: 04	Importance Level: High

Primary Actor: Farm Owner

Short Description: Managing livestock designated for slaughter and calculating profit.

Trigger: Livestock is selected for slaughter, requiring profit calculation.

Pre Conditions:

1. Livestock Manager must be registered to the Agritrack system.
2. Livestock Manager must be connected to the internet.
3. The Agritrack system must be available and online.

Type: [✓ External] / Temporal

Major Steps Performed

1. The Livestock Manager logs into the system.
2. Accesses the ‘Slaughter Management’ submodule.
3. Selects animals for slaughter and enters relevant data.
4. The system manages total profit for each animal from sales price minus accumulated costs.

Post Conditions:

1. The system provides an accurate profit calculation for animals processed for slaughter.

Information for Steps

→ Mail and Password

→ Selection of animals, Input of sale and cost data

← Profit calculation for each selected animal

Major Inputs:

Description	Source
- Mail and Password	Farm Owner
- Animal selection,	
- Sale and cost data	

Major Outputs:

Description	Destination
-Profit calculations per animal	Livestock Manager
	Financial Records

6.5 Import Data from Excel Sheets

Use Case Name: Import Data from Excel Sheets | ID: 05 | Importance Level: High

Primary Actor: Farm Owner

Short Description: Importing livestock and feed stock data from Microsoft Excel sheets into the Agritrack system.

Trigger: Existing farm data needs to be updated or initially populated in the Agritrack system.

Pre Conditions:

1. Data Entry Clerk must be registered to the Agritrack system.
2. Data Entry Clerk must be connected to the internet.
3. The Agritrack system must be available and online.
4. Excel sheets must be prepared in the format compatible with the Agritrack system.

Type: [✓ External] / Temporal

Major Steps Performed

1. The Data Entry Clerk logs into the system.
2. Navigates to the ‘Data Import’ function within the system architecture design module.
3. Selects and uploads the Excel sheet containing the relevant farm data.
4. The system validates the format and content of the Excel sheet.
5. Upon successful validation, the system imports the data, updating the corresponding modules (e.g., livestock management, feed stock).
6. The Data Entry Clerk reviews the import summary report for any discrepancies.

Post Conditions:

1. The Agritrack system is updated with the data from the Excel sheets, ensuring continuity and consistency in farm data management.

Information for Steps

→ Mail and Password

→ Excel sheet upload

← Validation results and import summary

Post Conditions:

Major Inputs:

Description	Source
- Mail and Password,	-Data Entry guy
- Excel sheet containing farm data	

Major Outputs:

Description	Destination
-Updated livestock and feed stock data in the system	- Agritrack System Database
-Livestock Manager	

6.6 Audit User Actions

Use Case Name: Audit User Actions	ID: 06	Importance Level: Medium

Primary Actor: System Auditor

Short Description: Logging and reviewing user actions within the Agritrack system to ensure compliance and accountability.

Trigger: The need to audit user activities for security, compliance, or investigative purposes.

Pre Conditions:

1. The System Auditor must have administrative access to the Agritrack system.
2. The Auditor must be connected to the internet.
3. The Agritrack system must be available and online.

Type: [✓ Internal] / Temporal

Major Steps Performed

1. The System Auditor logs into the system with administrative credentials.
2. Navigates to the 'Audit and Compliance' module.
3. Selects the 'User Action Logging' function to view logs.
4. Filters logs by specific criteria (e.g., date range, user, action type).
5. Reviews the filtered logs to audit user actions.
6. If necessary, exports the audit log for further analysis or reporting.
7. Takes appropriate compliance or corrective action based on audit findings.

Post Conditions:

1. The Auditor has access to detailed records of user actions within the Agritrack system, enabling compliance checks and security oversight

Information for Steps

→ Administrative credentials

→ Criteria for filtering logs

← Filtered audit logs

Major Inputs:

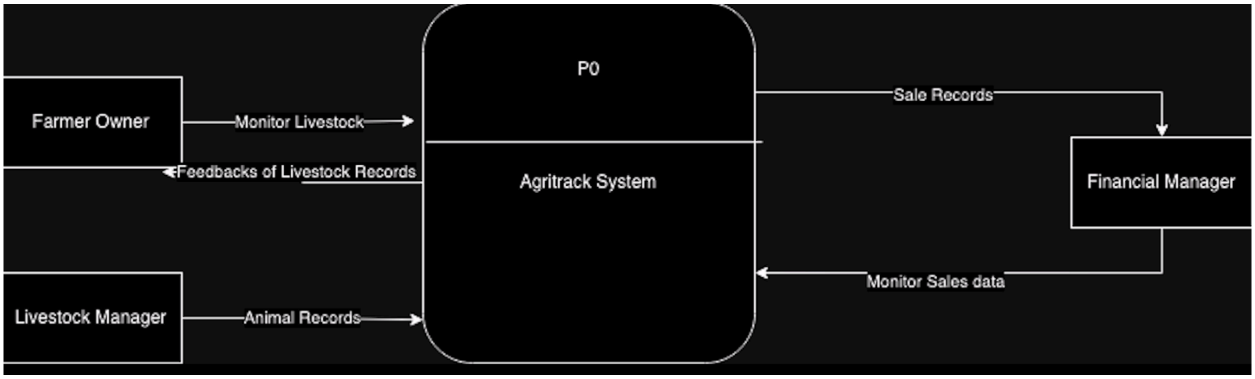
Description	Source
- Administrative credentials,	-System Auditor
- Filtering criteria (date, user, action)	

Major Outputs:

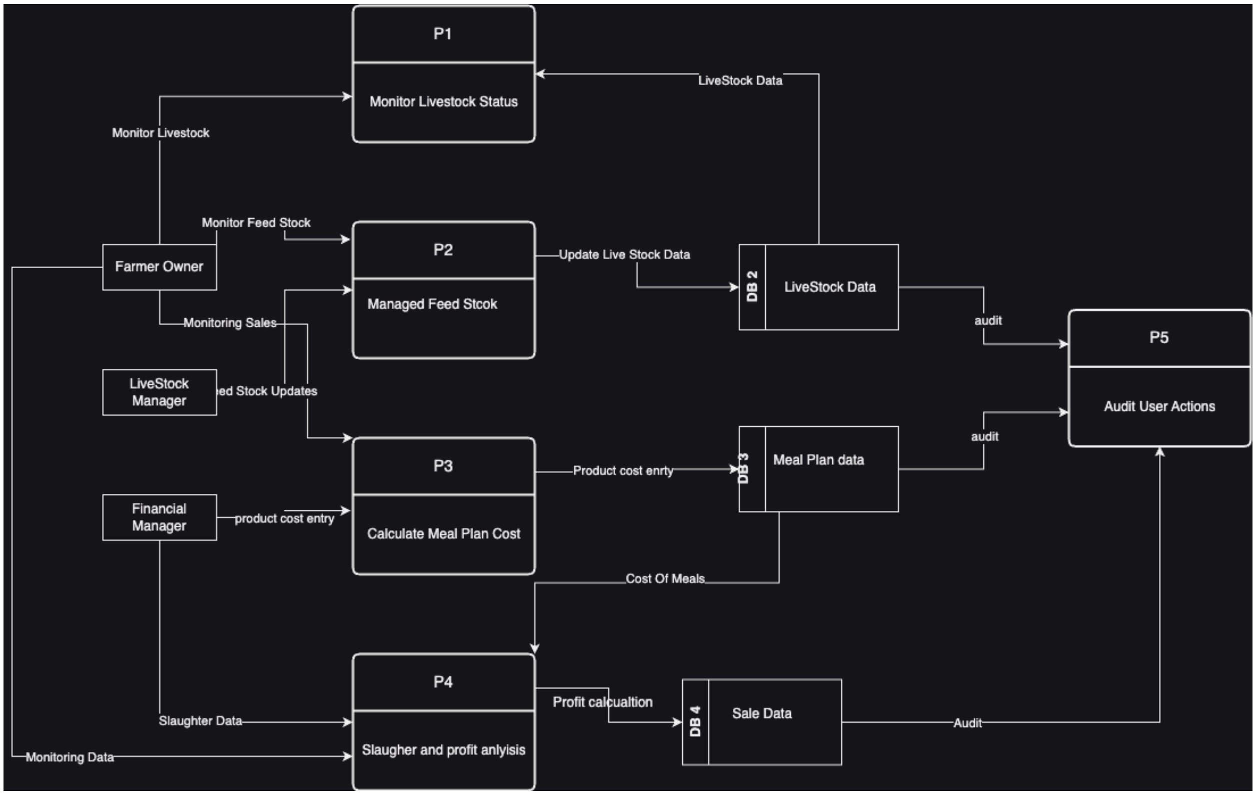
Description	Destination
Audit logs matching -	System Auditor, Compliance Reports

7.0 Process Models

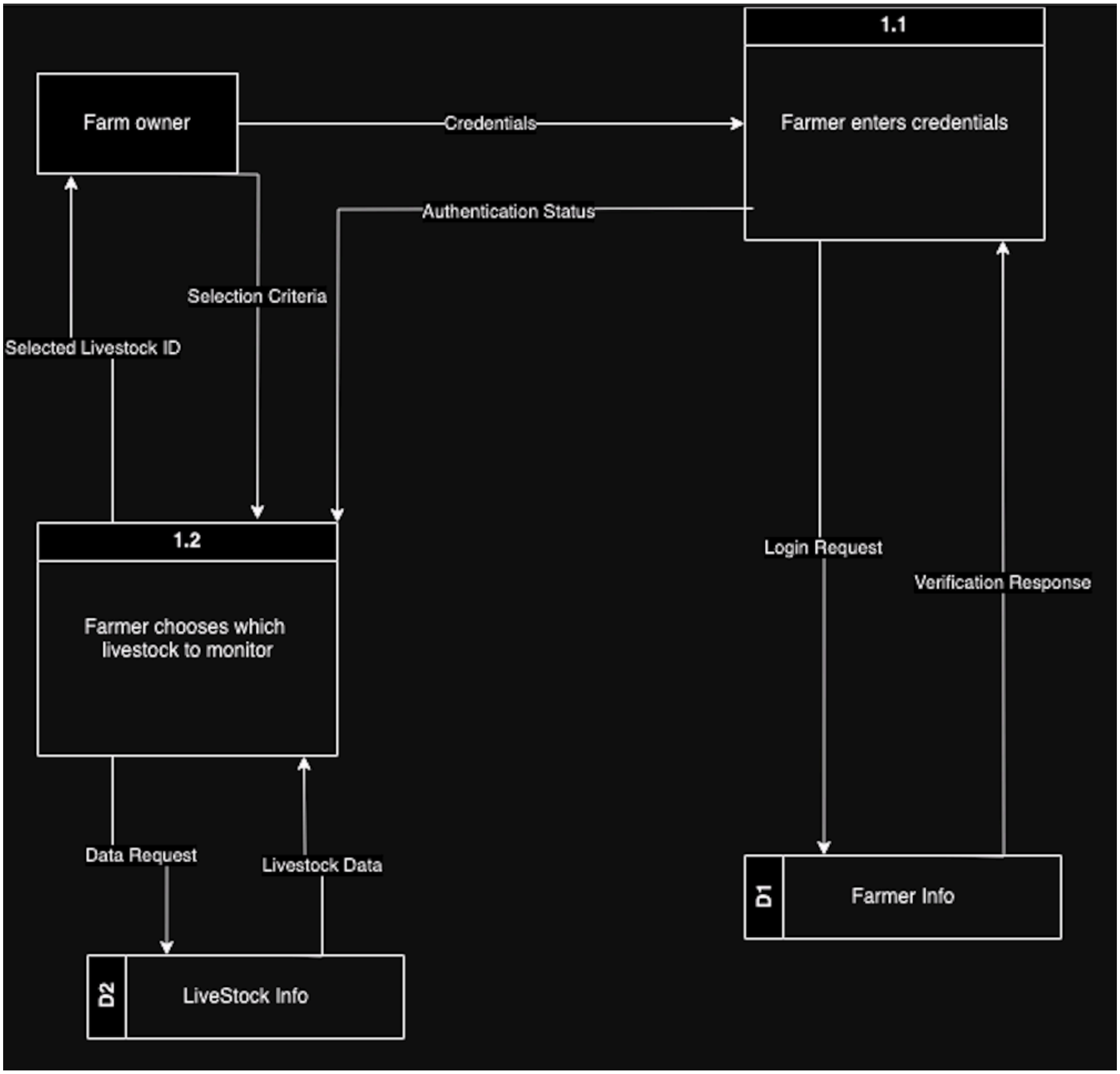
7.1 Context Diagram



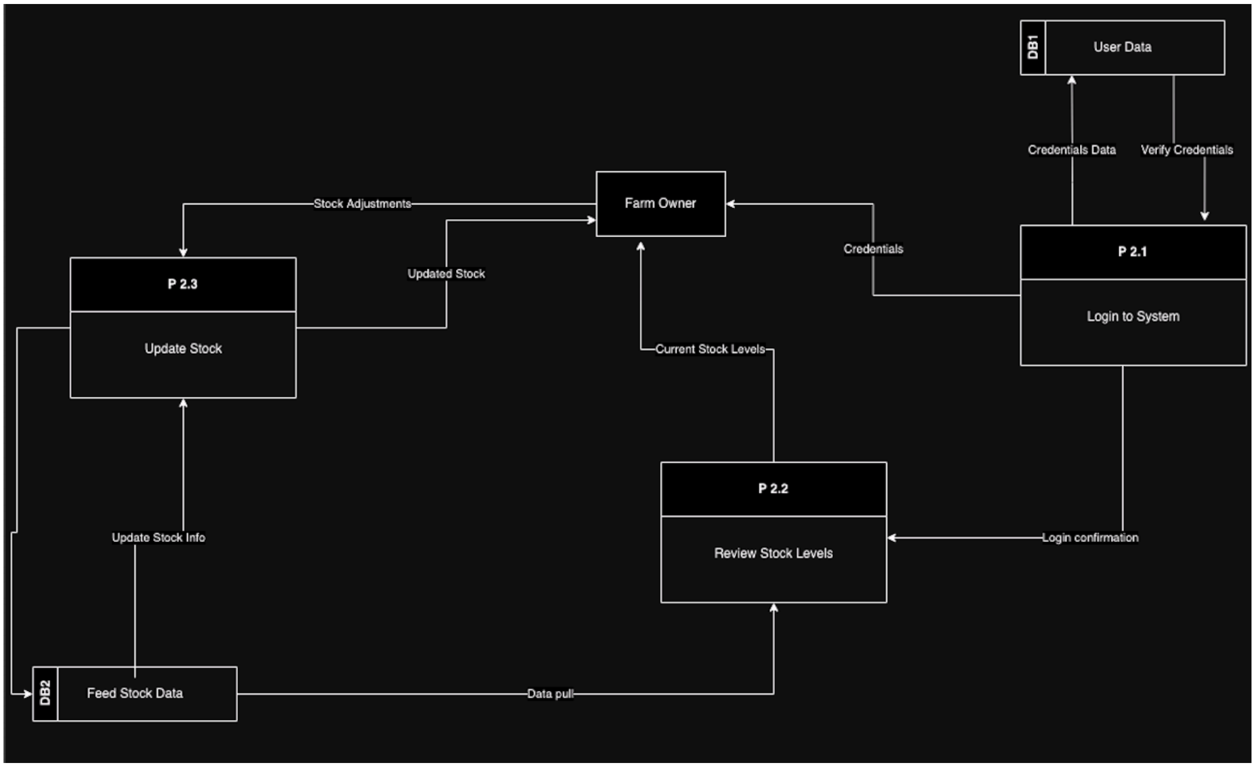
Level 0 Diagram



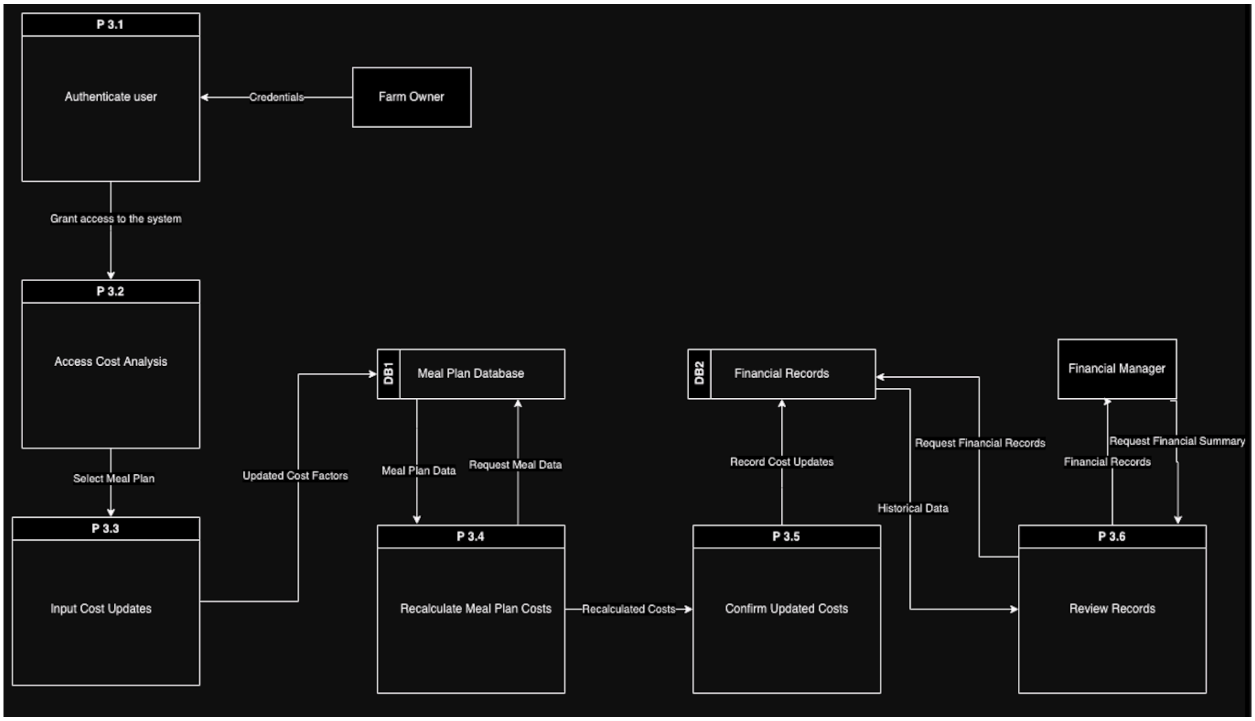
7.2 Level 1 Data Flow Diagram for the Use Case 1



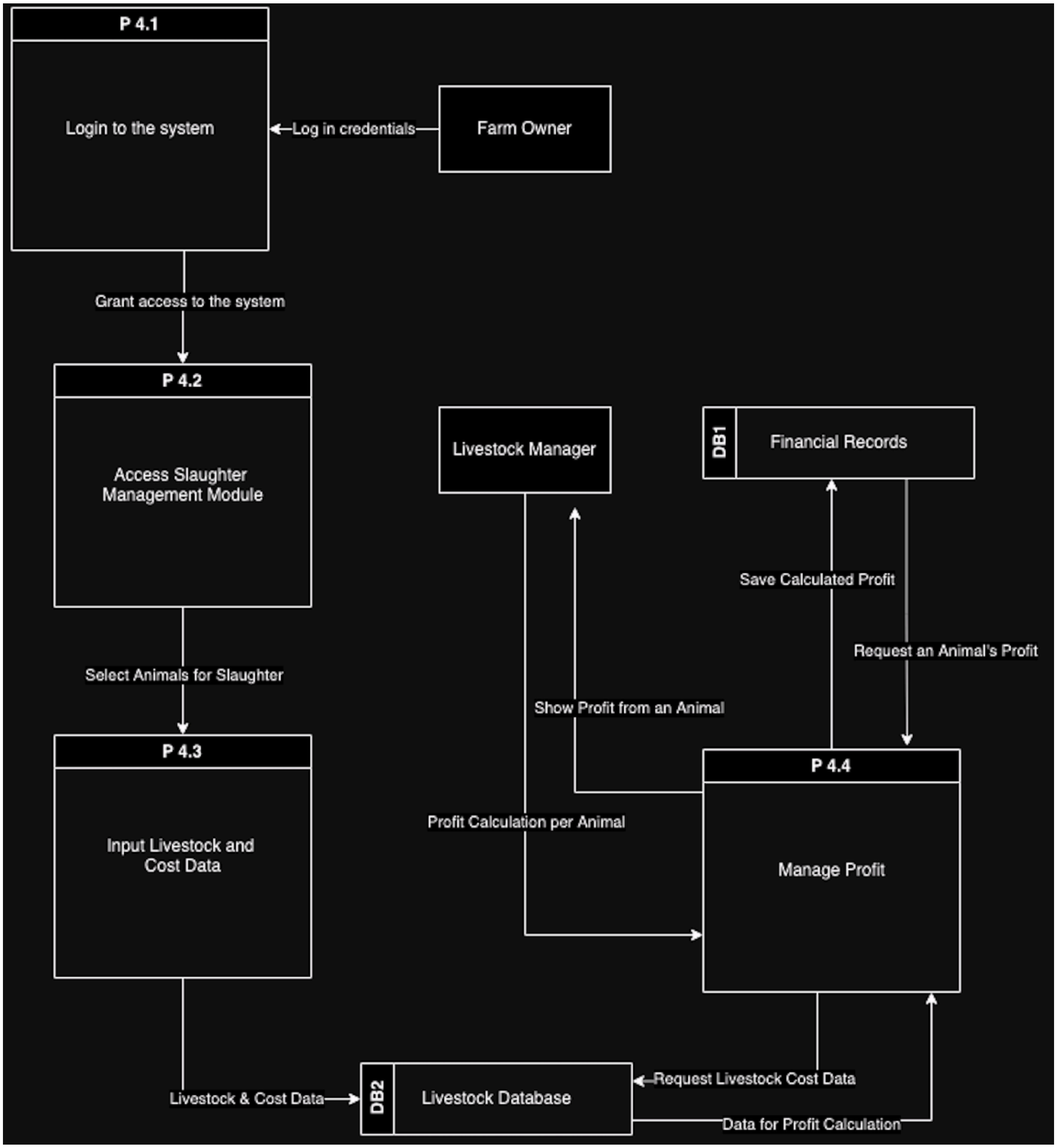
7.3 Level 1 Data Flow Diagram for the Use Case 2



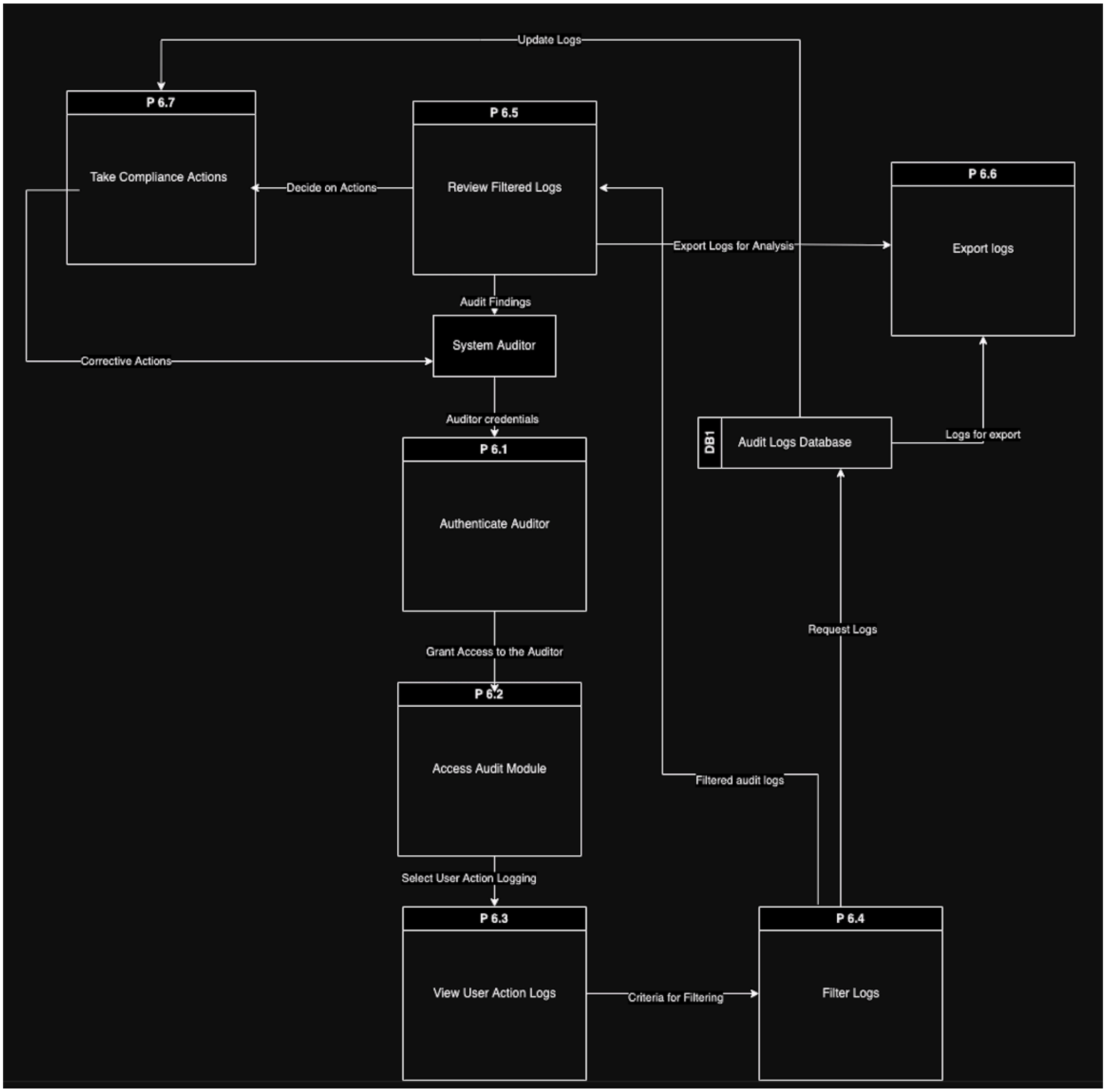
7.4 Level 1 Data Flow Diagram for the Use Case 3



7.5 Level 1 Data Flow Diagram for the Use Case 4


P 4.1
Login to the system
Log in credentials
Farm Owner
Grant access to the system
P 4.2
Access Slaughter Management Module
Select Animals for Slaughter
P 4.3
Input Livestock and Cost Data
Livestock & Cost Data
Livestock Manager
DB1
Financial Records
Save Calculated Profit
Request an Animal's Profit
Show Profit from an Animal
Profit Calculation per Animal
P 4.4
Manage Profit
DB2
Livestock Database
Request Livestock Cost Data
Data for Profit Calculation

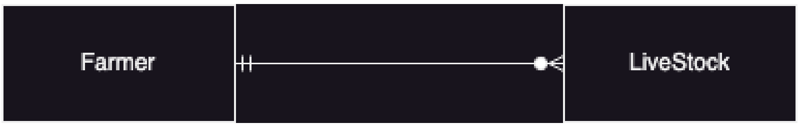
7.6 Level 1 Data Flow Diagram for the Use Case 6



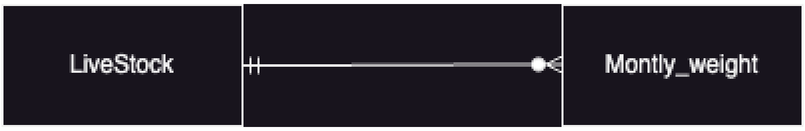
8.0 Logical Data Model

8.1 Cardinalities Relationships

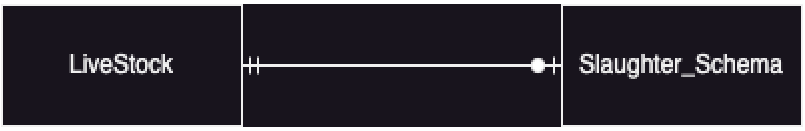
8.1.1 Farmer and Livestock



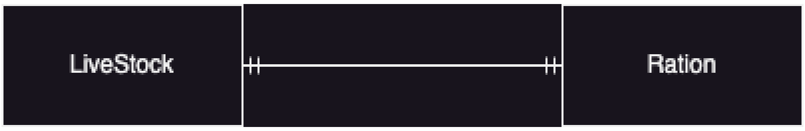
8.1.2 Livestock and Monthly Weight



8.1.3 Livestock and Slaughter Schema :

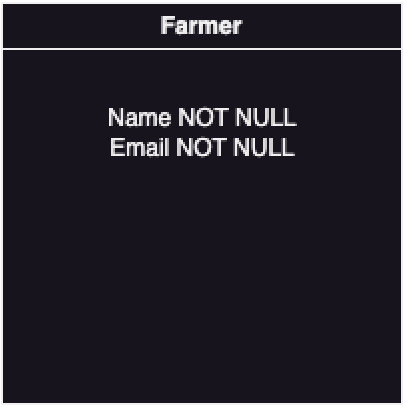


8.1.4 Livestock and Ration:



8.2 Identifier/key Attributes and Modality

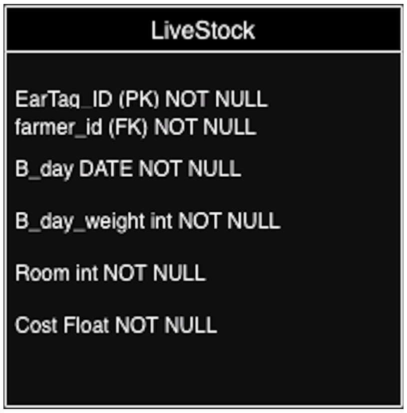
8.2.1 Farmer



Farmer

Name NOT NULL
Email NOT NULL

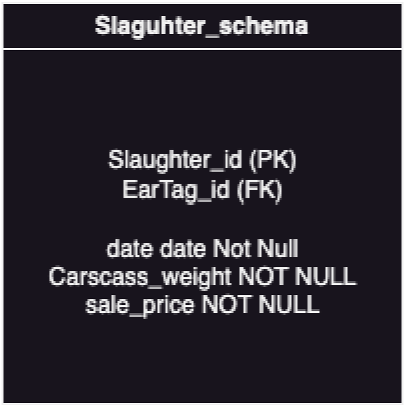
8.2.2 Livestock



LiveStock

EarTag_ID (PK) NOT NULL
farmer_id (FK) NOT NULL
B_day DATE NOT NULL
B_day_weight int NOT NULL
Room int NOT NULL
Cost Float NOT NULL

8.2.3 Slaughter Schema

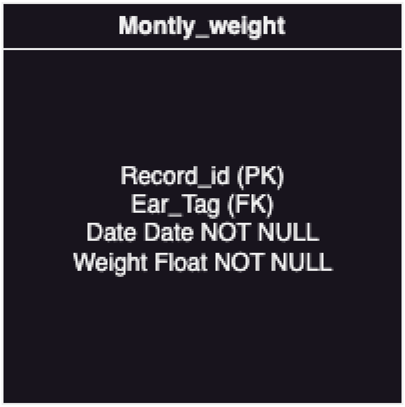


Slaguhter_schema

Slaughter_id (PK)
EarTag_id (FK)

date date Not Null
Carscass_weight NOT NULL
sale_price NOT NULL

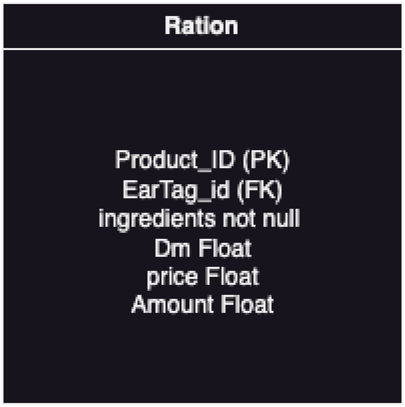
8.2.4 Monthly Weight



Montly_weight

Record_id (PK)
Ear_Tag (FK)
Date Date NOT NULL
Weight Float NOT NULL

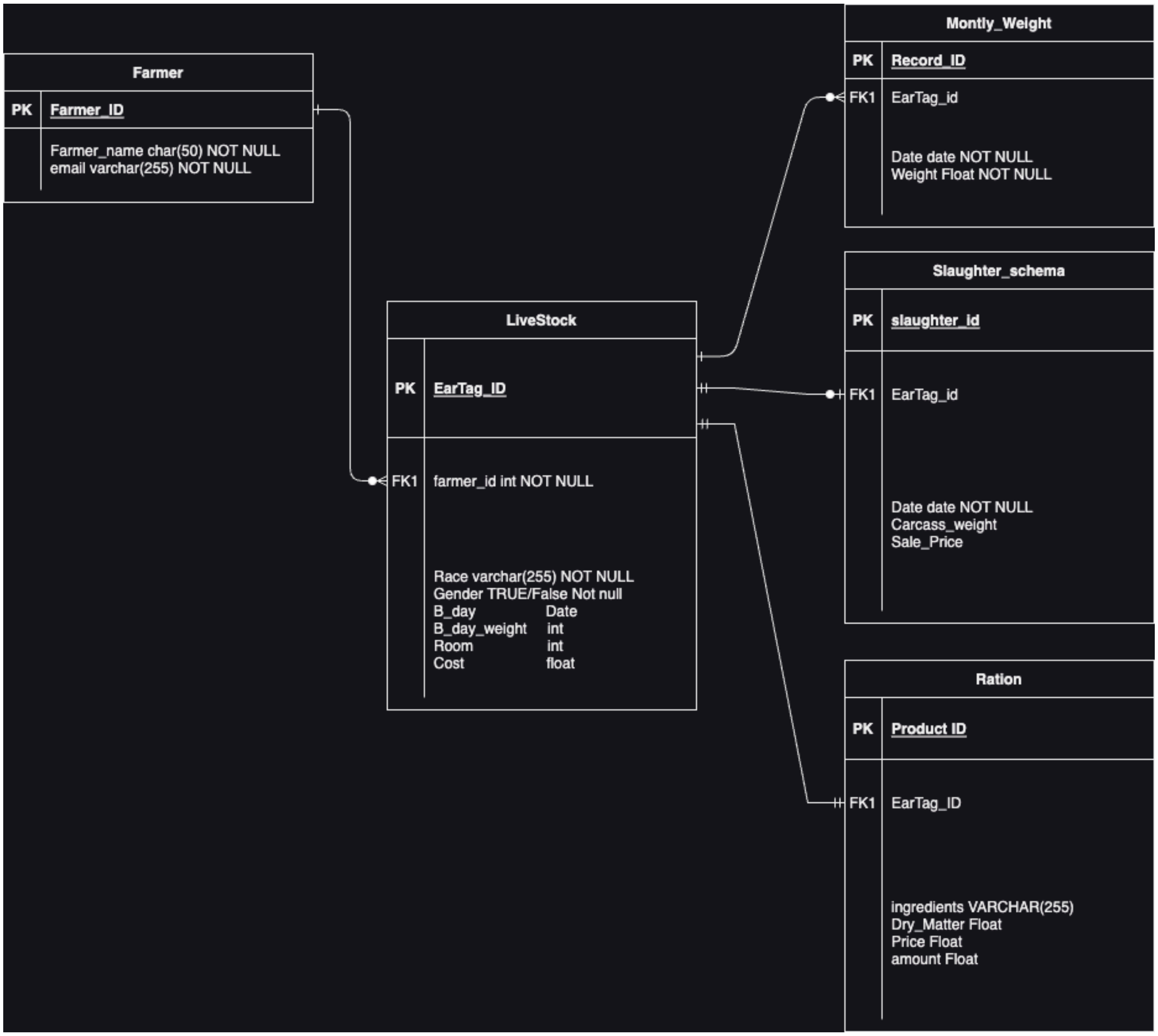
8.2.5 Ration



Ration

Product_ID (PK)
EarTag_id (FK)
ingredients not null
Dm Float
price Float
Amount Float

8.3 ERD Diagram



Farmer Table: This table is 3NF. It has a primary key (Farmer_ID), and both Farmer_name and email are non-key attributes that are directly dependent on the primary key. There are no transitive dependencies evident from the given structure.

LiveStock Table: The LiveStock table is in 3NF if every non-key attribute (Race, Gender, B_day, B_day_weight, Room, Cost) depends only on the primary key

(EarTag_ID), and not on any other non-key attribute or subset of a composite key. Ensure that Room and Cost do not depend on each other or any other attribute besides EarTag_ID.

Monthly_Weight Table: This table has a composite key (Record_ID and EarTag_id). It's in 3NF if Date and Weight depend on the entire composite key and there are no transitive dependencies.

Slaughter_Schema Table: This table meets the 3NF criteria with Slaughter_id as the primary key and EarTag_id as a foreign key. Other attributes (Date, Carcass_weight, Sale_Price) are functionally dependent on the primary key. Table is in 3NF.

Ration Table: This table meets the 3NF criteria with Product_ID as primary key and Ear_Tag_id as foreign key which means every animal has a unique meal plan. Other attributes (ingredients, Dry matter, Price, amount) are functionally dependent on the primary key.

9.0 Architecture Design

Three-Tiered Client-Server Architecture

This architecture divides the system into three distinct layers, each responsible for specific aspects of the application's operations. This separation enhances manageability, scalability, and security.

9.1 Presentation Tier (Client-Side):

- **Technology**: Utilizes React.js to deliver a responsive and interactive user interface optimized for both efficiency and user experience.
- **Features**:
    - User authentication interface.
    - Livestock management and real-time data visualization.
    - Notification system for critical updates and alerts.

9.2 Application Tier (Server-Side):

- **Technology**: Node.js with Express.js forms the backbone of the server-side application, managing API requests and business logic.
- **Microservices**:
    - **Authentication Service**: Manages user authentication and authorization, ensuring secure access.
    - **Livestock Management Service**: Facilitates CRUD operations for livestock data.
    - **Feed Stock Management Service**: Oversees inventory levels and sends notifications regarding stock.
    - **Meal Plan Service**: Updates meal plans and calculates associated costs based on current data.
    - **Slaughter Management Service**: Tracks the slaughter process and calculates profits.
- **Real-Time Data Processing**:
    - **Technology**: WebSockets enable real-time updates and notifications to clients, ensuring data synchronization without delay.

9.3 Data Tier:

- **Database Management System**: MySQL is used for its robust data storage capabilities.
- **Features**:
  - Optimized schema design to ensure quick data retrieval and storage.
  - Data encryption and secure backup solutions maintain data integrity and availability.

9.4 Cloud Deployment:

- **Platform**: AWS or Azure platforms are employed to provide a scalable and reliable infrastructure.
- **Services**:
  - **Compute**: EC2 instances or Azure VMs host the services.
  - **Storage**: S3 or Azure Blob Storage manage file storage.
  - **Database**: RDS or Azure SQL Database for efficient data management.
  - **Load Balancing**: ELB or Azure Load Balancer distributes incoming traffic to ensure smooth operation.

9.5 Security Measures:

- **Data Encryption**: Encrypts data at rest and in transit using industry standards to prevent unauthorized access.
- **Authentication and Authorization**: Implements OAuth 2.0 and Role-Based Access Control (RBAC) to manage access to sensitive information securely.
- **Audit Logs**: Maintains comprehensive logs of user actions for compliance and security audits.

Summary: The Three-Tiered Client-Server Architecture provides a robust framework for the Agritrack project. This architecture supports high scalability, enhanced security, and effective separation of concerns, making it ideal for handling complex agricultural data management tasks efficiently and securely.

10.0 Interface design

10.1 Use scenarios

10.1.1 Scenario: Focused Visitor - Slaughterhouse Management Interface

User Objective: The user aims to efficiently manage the cost and profit of animals sent to the slaughterhouse.

1. Navigation: The user navigates to the "Slaughter" tab within the management interface.
   Input Animal Details: Upon arrival at the slaughter tab, the user is presented with a clear input form. This form includes fields for the Animal ID, Carcass Weight, and Sale Price.

2. Data Entry: The user diligently fills in the required fields with accurate information regarding the animal being processed.
   Save and Calculate: After entering the data, the user clicks the "Enter" button to save the input to the system an trigger cost and profit calculations.

3. Cost and Profit Display: Instantaneously, the system computes the cost of the animal, factoring in various expenses such as feed, labor, and transportation. It then displays the profit margin, reflecting the difference between the sale price and the calculated cost.

4. Report Generation: Anticipating the need for comprehensive insights, the interface offers a feature to generate daily slaughter reports. Through a clearly labeled button or menu option, the user can request detailed reports summarizing the day's activities, including costs, profits, and any relevant trends.

10.1.2 Scenario: Focused Visitor - Livestock Management Interface

User Objective: The user seeks to effectively oversee the cost and profit of animals designated for the slaughterhouse.

1. Navigation: Upon accessing the Livestock Management Interface, the user directs their attention to the "My Stable" section. Here, they can readily view the current inventory of animals under their care.

2. Addition of New Animals: To include a new animal to the existing stock, the user initiates the process by clicking the prominently displayed "+" button. This action opens a comprehensive input form where the user can input essential details such as Ear Tag number, Gender, Cost, Location (Room), and Breed of the animal.

3. Data Entry and Validation: With precision, the user fills in the required fields with accurate information pertaining to the new addition. The system may incorporate validation checks to ensure data integrity, guaranteeing that all essential information is captured.

4. Save and Database Update: Upon completion of data entry, the user finalizes the process by clicking the "Save" button. Instantaneously, the system securely stores the newly added animal's information into the database, ensuring that the Livestock inventory remains up-to-date and reflective of the user's operations.

10.1.3 Scenario: Focused Visitor - Monthly Weight Interface

User Objective: The user intends to input animal weights into the system for record-keeping and analysis.

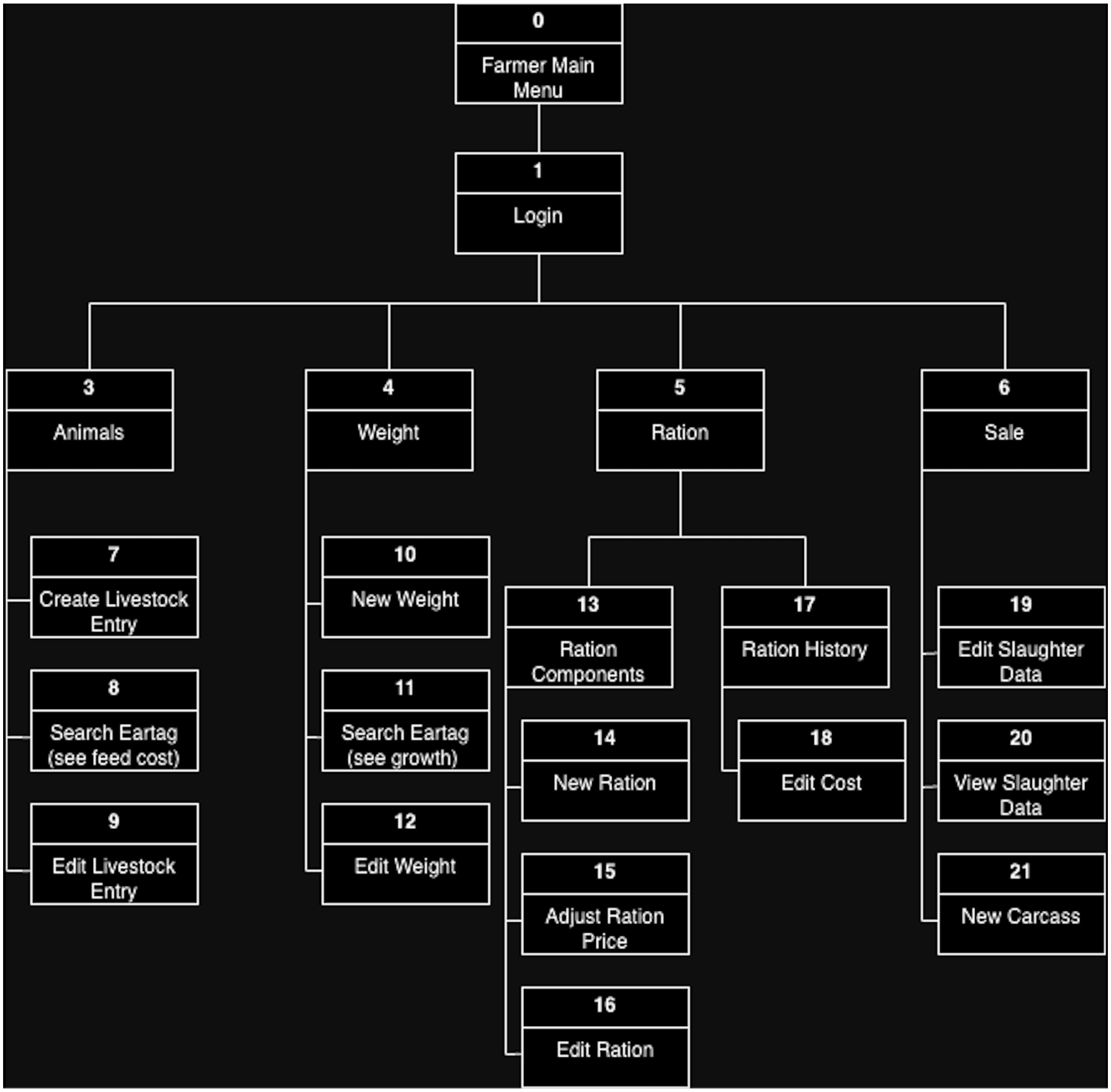
1. Navigation: The user initiates their task by selecting the "Monthly Weight List" option situated conveniently on the left-hand side of the dashboard. Upon selection, the corresponding page seamlessly loads, providing a dedicated space for managing monthly weight data.

2. Adding Weight Records: To begin the process of inserting animal weights, the user locates and clicks the prominently displayed "+" button. This action triggers the opening of an intuitive input form where the user can effortlessly enter the Ear Tag number of the animal along with its corresponding weight for the month.

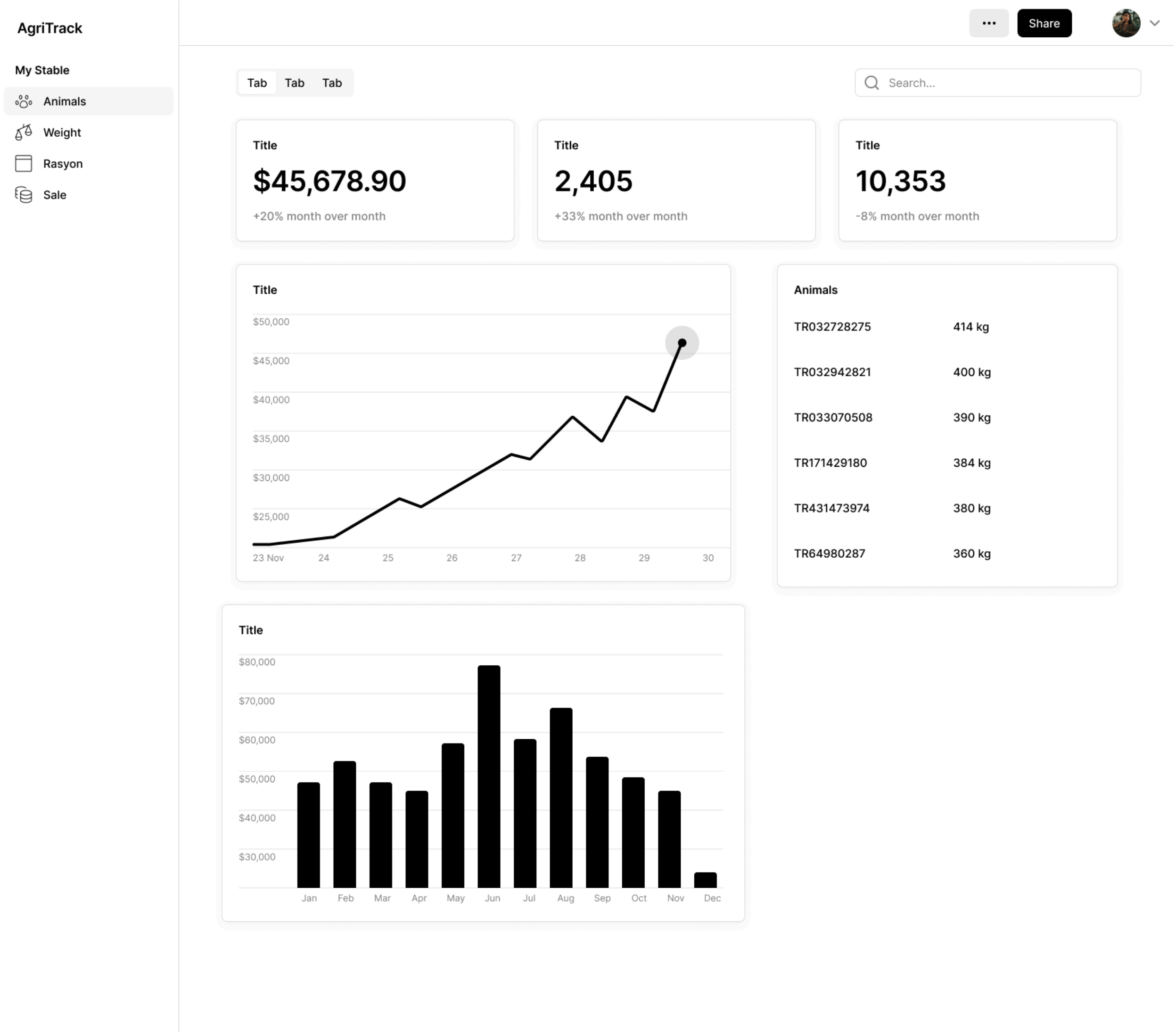
3. Data Input and Submission: With precision and accuracy, the user inputs the relevant information into the designated fields. Once satisfied, they proceed to submit the data by clicking the appropriate button or icon, indicating their completion of the weight entry process.

4. Data Integration and Analysis:Upon submission, the system saves the weight data, using it to calculate optimal prices based on feeding needs and growth trends. Continuous analysis enables informed decisions on livestock management and pricing.

10.2 Structure Diagram



10.2.1- Dashboard



AgriTrack

My Stable

- Animals
- Weight
- Rasyon
- Sale

Title

$45,678.90

+20% month over month

Title

2,405

+33% month over month

Title

10,353

-8% month over month

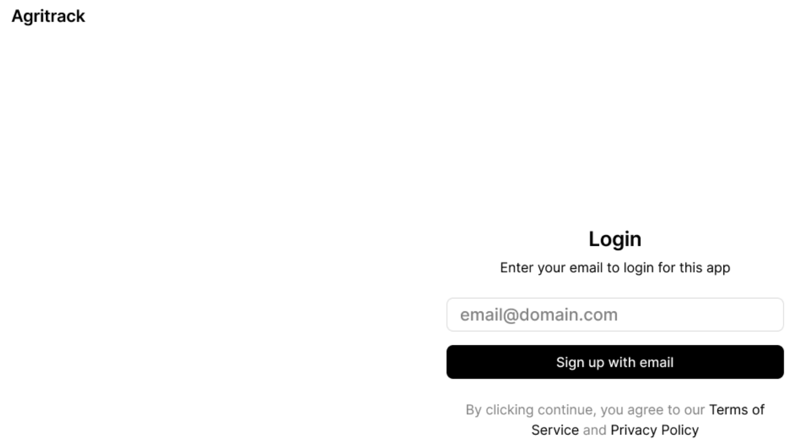
Title

Animals

TR032728275	414 kg
TR032942821	400 kg
TR033070508	390 kg
TR171429180	384 kg
TR431473974	380 kg
TR64980287	360 kg

Title

10.2.2 - Authentication



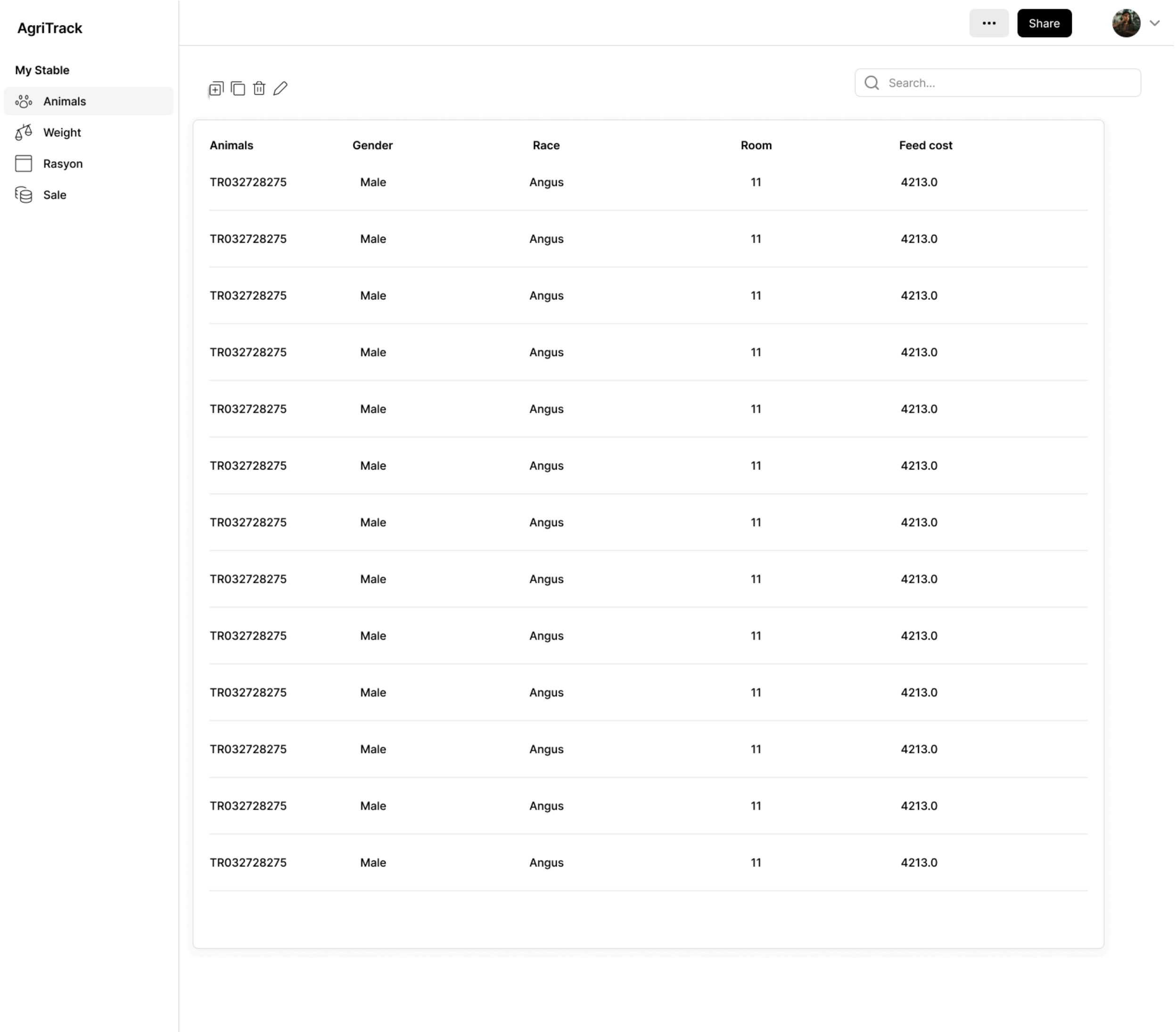
Agritrack

Login

Enter your email to login for this app

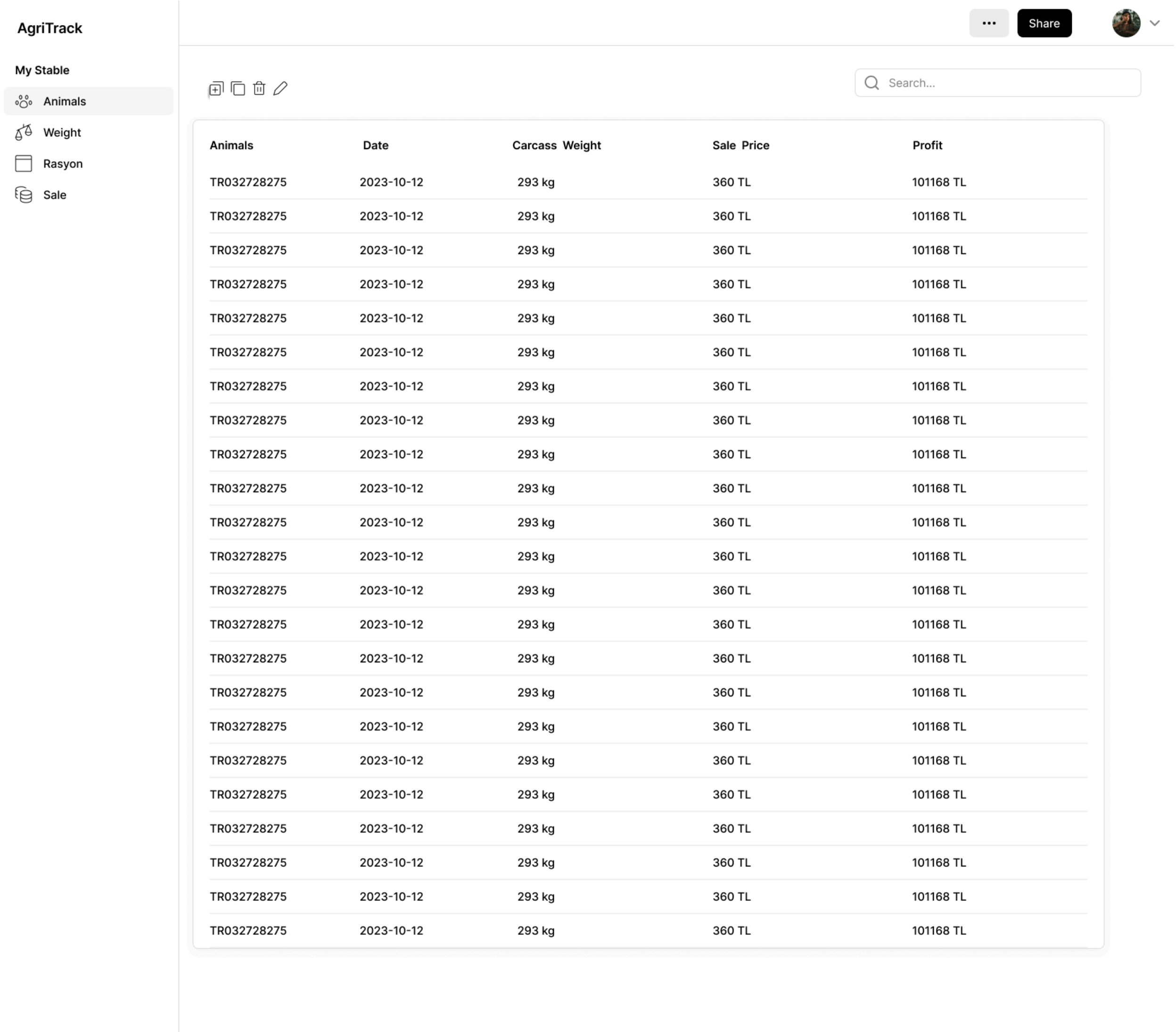
By clicking continue, you agree to our Terms of Service and Privacy Policy

10.2.3 - Animal - LiveStock Interface



Animals	Gender	Race	Room	Feed cost
TR032728275	Male	Angus	11	4213.0
TR032728275	Male	Angus	11	4213.0
TR032728275	Male	Angus	11	4213.0
TR032728275	Male	Angus	11	4213.0
TR032728275	Male	Angus	11	4213.0
TR032728275	Male	Angus	11	4213.0
TR032728275	Male	Angus	11	4213.0
TR032728275	Male	Angus	11	4213.0
TR032728275	Male	Angus	11	4213.0
TR032728275	Male	Angus	11	4213.0
TR032728275	Male	Angus	11	4213.0
TR032728275	Male	Angus	11	4213.0
TR032728275	Male	Angus	11	4213.0

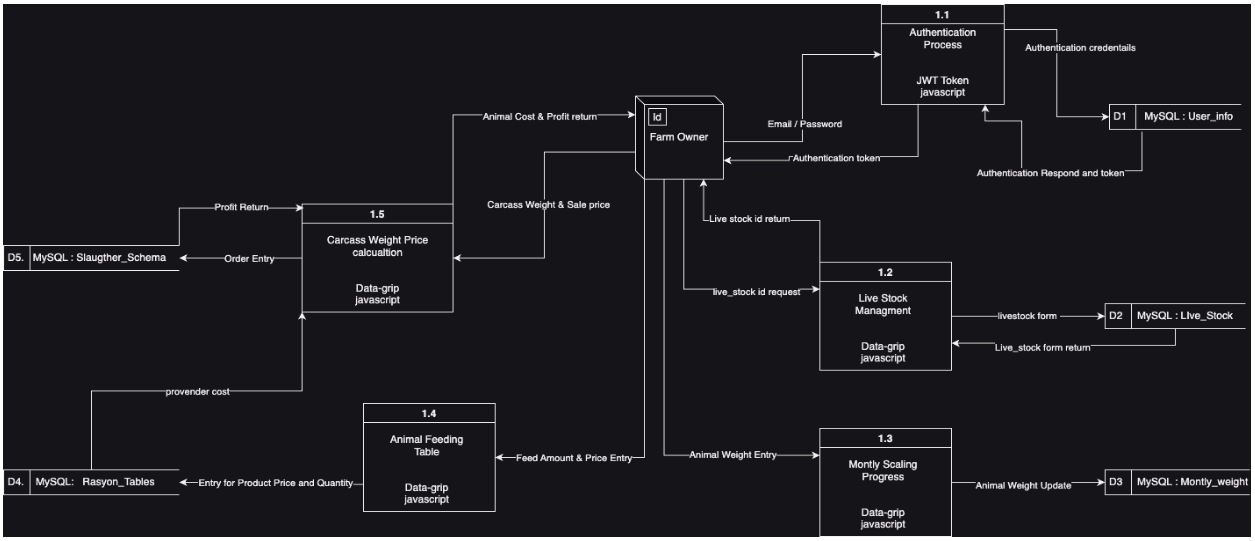
10.2.4 - Sale Interface



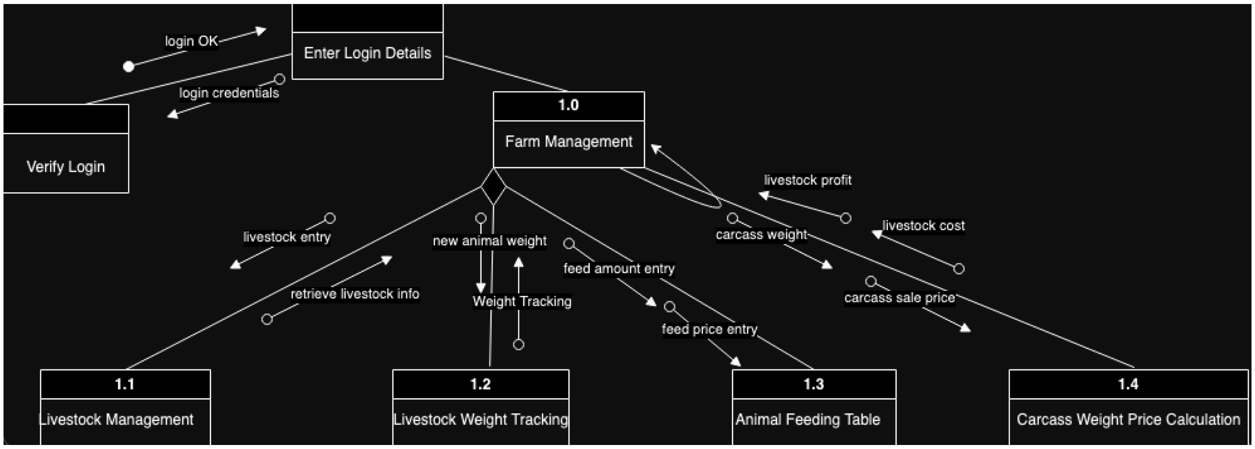
Animals	Date	Carcass Weight	Sale Price	Profit
TR032728275	2023-10-12	293 kg	360 TL	101168 TL
TR032728275	2023-10-12	293 kg	360 TL	101168 TL
TR032728275	2023-10-12	293 kg	360 TL	101168 TL
TR032728275	2023-10-12	293 kg	360 TL	101168 TL
TR032728275	2023-10-12	293 kg	360 TL	101168 TL
TR032728275	2023-10-12	293 kg	360 TL	101168 TL
TR032728275	2023-10-12	293 kg	360 TL	101168 TL
TR032728275	2023-10-12	293 kg	360 TL	101168 TL
TR032728275	2023-10-12	293 kg	360 TL	101168 TL
TR032728275	2023-10-12	293 kg	360 TL	101168 TL
TR032728275	2023-10-12	293 kg	360 TL	101168 TL
TR032728275	2023-10-12	293 kg	360 TL	101168 TL
TR032728275	2023-10-12	293 kg	360 TL	101168 TL
TR032728275	2023-10-12	293 kg	360 TL	101168 TL
TR032728275	2023-10-12	293 kg	360 TL	101168 TL
TR032728275	2023-10-12	293 kg	360 TL	101168 TL
TR032728275	2023-10-12	293 kg	360 TL	101168 TL
TR032728275	2023-10-12	293 kg	360 TL	101168 TL
TR032728275	2023-10-12	293 kg	360 TL	101168 TL
TR032728275	2023-10-12	293 kg	360 TL	101168 TL
TR032728275	2023-10-12	293 kg	360 TL	101168 TL
TR032728275	2023-10-12	293 kg	360 TL	101168 TL
TR032728275	2023-10-12	293 kg	360 TL	101168 TL

11.0 Program Design

11.1 Physical Process Model

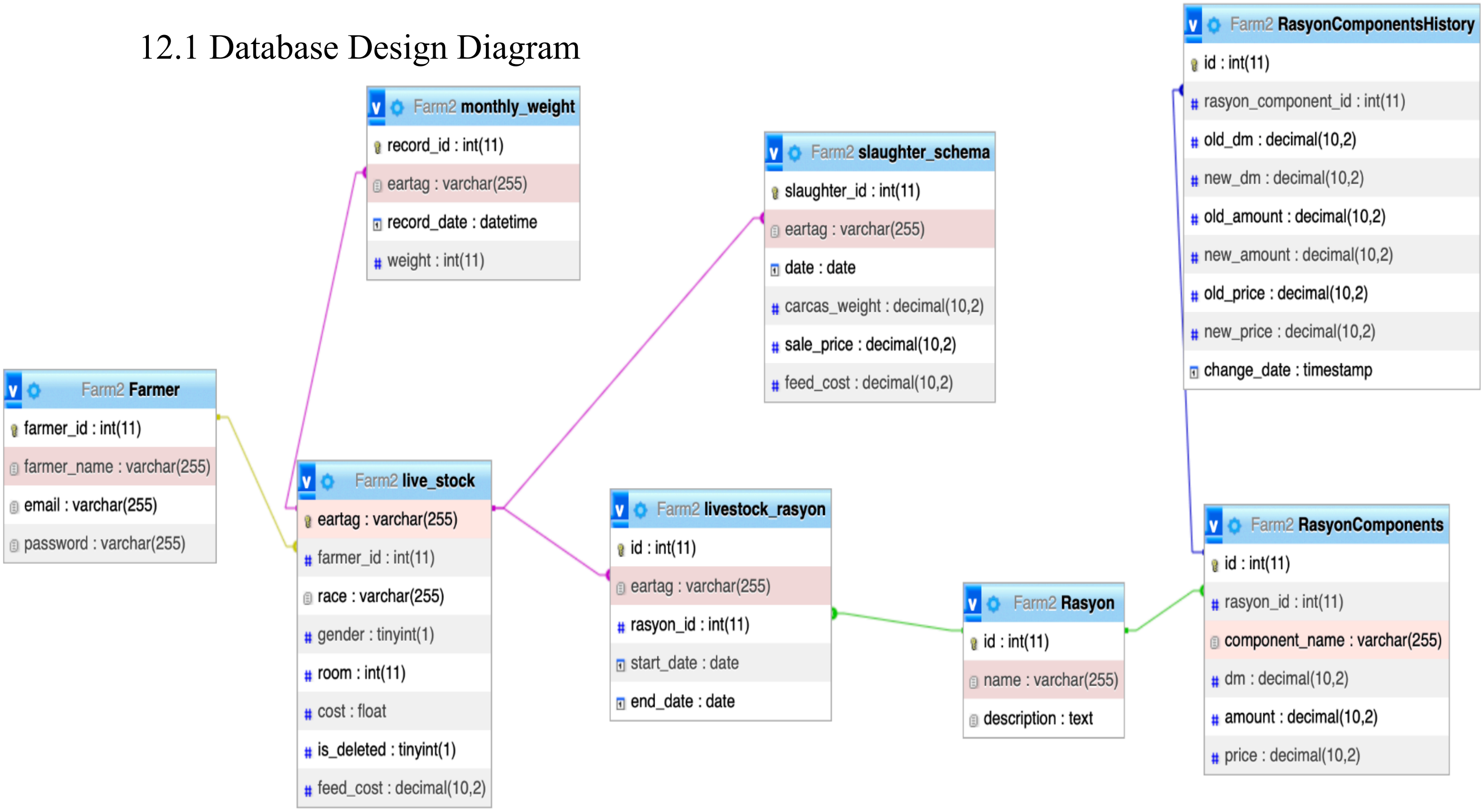


11.2 Program Structure Chart



12.0 Physical Data Storage Design

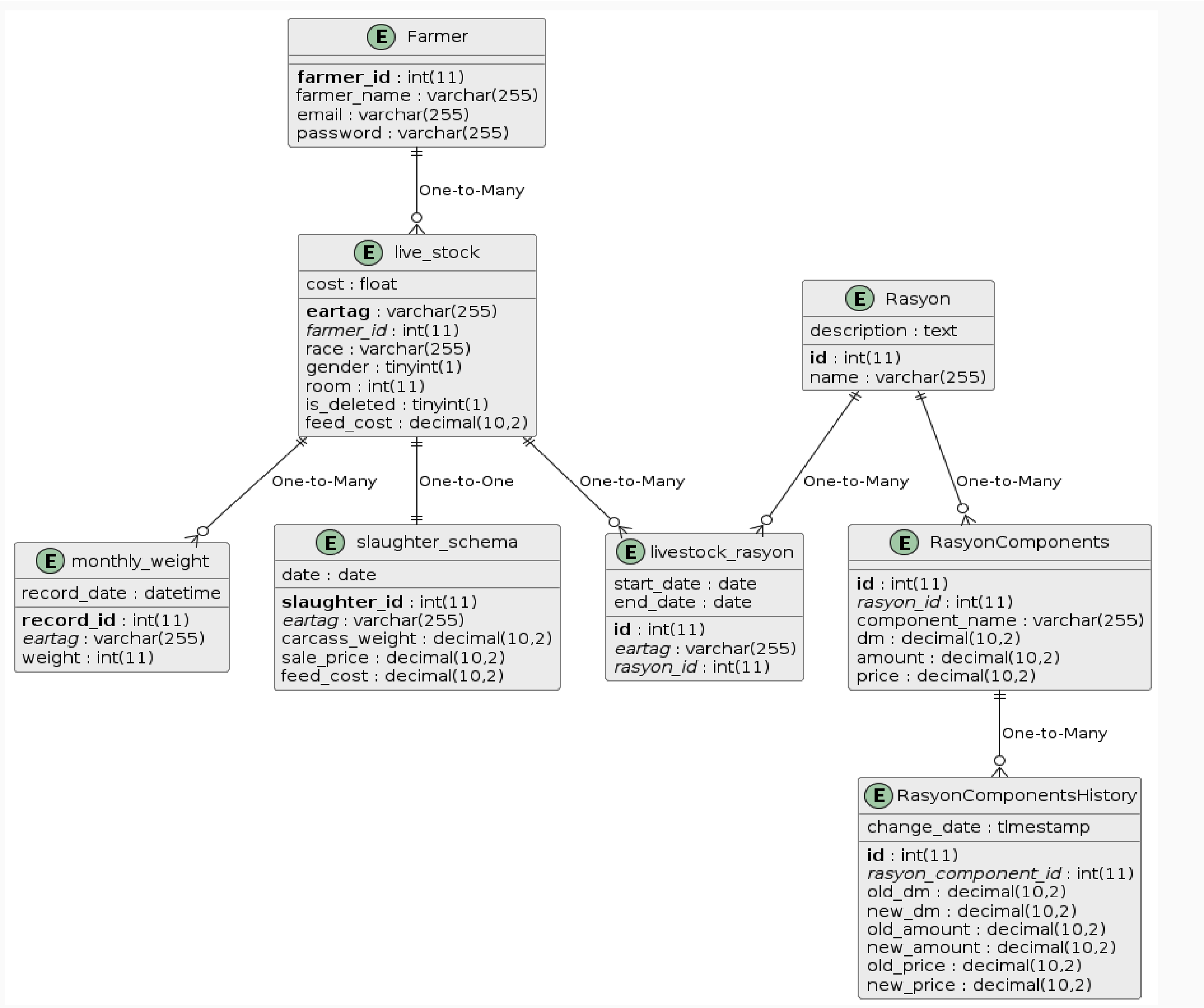
12.1 Database Design Diagram



12.2 Relationships

- **Farmer** to **live_stock**: One-to-Many (one farmer can have many livestock)
- **live_stock** to **monthly_weight**: One-to-Many (one livestock can have many monthly weight records)
- **live_stock** to **slaughter_schema**: One-to-One (each livestock can have one slaughter record)
- **Rasyon** to **RasyonComponents**: One-to-Many (one Rasyon can have many components)
- **RasyonComponents** to **RasyonComponentsHistory**: One-to-Many (one component can have many history records)
- **live_stock** to **livestock_rasyon**: One-to-Many (one livestock can have many rasyon records)
- **Rasyon** to **livestock_rasyon**: One-to-Many (one rasyon can be used by many livestock)

12.3 Database Diagram with crow's foot notation


E
Farmer
farmer_id : int(11)
farmer_name : varchar(255)
email : varchar(255)
password : varchar(255)
One-to-Many
E
live_stock
cost : float
eartag : varchar(255)
farmer_id : int(11)
race : varchar(255)
gender : tinyint(1)
room : int(11)
is_deleted : tinyint(1)
feed_cost : decimal(10,2)
E
Rasyon
description : text
id : int(11)
name : varchar(255)
One-to-Many
One-to-One
One-to-Many
One-to-Many
One-to-Many
E
monthly_weight
record_date : datetime
record_id : int(11)
eartag : varchar(255)
weight : int(11)
E
slaughter_schema
date : date
slaughter_id : int(11)
eartag : varchar(255)
carcass_weight : decimal(10,2)
sale_price : decimal(10,2)
feed_cost : decimal(10,2)
E
livestock_rasyon
start_date : date
end_date : date
id : int(11)
eartag : varchar(255)
rasyon_id : int(11)
E
RasyonComponents
id : int(11)
rasyon_id : int(11)
component_name : varchar(255)
dm : decimal(10,2)
amount : decimal(10,2)
price : decimal(10,2)
One-to-Many
E
RasyonComponentsHistory
change_date : timestamp
id : int(11)
rasyon_component_id : int(11)
old_dm : decimal(10,2)
new_dm : decimal(10,2)
old_amount : decimal(10,2)
new_amount : decimal(10,2)
old_price : decimal(10,2)
new_price : decimal(10,2)