Project Report

DAIRY FIRM MANAGEMENT SYSTEM (DFMS)

Course Code: ICT 0610 5360

Submitted by:
Md. Shahidul Alam (ID: 2023822017, Batch-6th)
Program: Masters in Information Technology, IICT, SUST

Supervisor:
Professor Dr. Mohammad Abdullah Al Mumin
Department of CSE & Director, IICT,
Shahjalal University of Science and Technology (SUST)

Submission Date: 17th February 2026

Declaration

I declare that I have personally created and developed the project titled **“Dairy Firm Management System (DFMS)”** as part of my academic work. This project is fully designed, coded, and implemented by me to enhance the efficiency of dairy firm operations through a centralized digital platform. The DFMS provides multiple functional modules, including cow record and breed management, cow health records with treatments and vaccinations, feed inventory and scheduling, milk production data, and employee record management. Each module is designed to streamline daily activities, improve record accuracy, and support effective decision-making. All data, materials, and references used during this project have been properly acknowledged and cited. This project reflects my independent effort, creativity, and technical understanding.

……

Md. Shahidul Alam

Recommendation Letter from Project Supervisor

This is to certify that **Md. Shahidul Alam**, a dedicated and sincere student, has successfully completed his project work entitled **"Dairy Firm Management System (DFMS)"** under my supervision. The project has been designed and developed entirely by him with great effort, creativity, and technical understanding. The DFMS project focuses on creating a centralized digital platform that includes modules for cow record and breed management, health tracking with treatments and vaccinations, feed inventory and scheduling, milk production data logging, and employee record management. Throughout the project, he demonstrated excellent analytical and problem-solving skills. I, therefore, recommend his project report for examination and evaluation, as it fulfills the required academic and technical standards.

…….

Supervisor:

Professor Dr. Mohammad Abdullah Al Mumin

Department of CSE & Director, IICT, SUST

Acknowledgment

I would like to express my sincere gratitude to my respected supervisor, **Professor Dr. Mohammad Abdullah Al Mumin**, for his continuous guidance, support, and valuable feedback throughout the development of my project titled **"Dairy Firm Management System (DFMS)"**. His expert advice, encouragement, and insightful suggestions have been instrumental in helping me plan, design, and complete my work successfully. I am deeply thankful for his patience, motivation, and constructive criticism, which inspired me to enhance my technical understanding and problem-solving abilities. Without his supervision and kind cooperation, the completion of this project would not have been possible.

Qualification Form of Master's Degree

Student Name: Md. Shahidul Alam (ID: 2023822017, Batch-6th)
Project Title: **Dairy Firm Management System (DFMS)**

This is to certify that the project report submitted by the student mentioned above has been thoroughly reviewed and evaluated by the Examination Committee. The report demonstrates a clear understanding of system design, development, and implementation processes. Based on the assessment, the project is qualified and approved by the Examination Committee and meets the academic standards required for the Master's Degree program.

.....	.....	.....
Director, IICT	Chairman, Examination Committee	Supervisor

Abstract

The **Dairy Firm Management System (DFMS)** is a specialized software solution developed to modernize and streamline the management of dairy farm operations. This system provides a centralized platform that integrates multiple critical functions to ensure efficient and accurate management of resources and data. Key modules of DFMS include cow record and breed management, health tracking including treatments and vaccinations, feed inventory and scheduling, milk production data logging, and employee record management. By automating these processes, the system significantly reduces manual workload, minimizes errors, and ensures that all operational data is easily accessible and well-organized.

The project has been developed using modern programming techniques and database management systems, offering a user-friendly interface that allows dairy farm managers to monitor and manage daily operations effectively. The DFMS not only enhances productivity and record-keeping but also provides analytical insights for better decision-making regarding herd management, feed planning, and employee efficiency. This project demonstrates the practical application of information technology in the agricultural sector and highlights the potential of digital solutions to improve operational efficiency, transparency, and sustainability in dairy farm management.

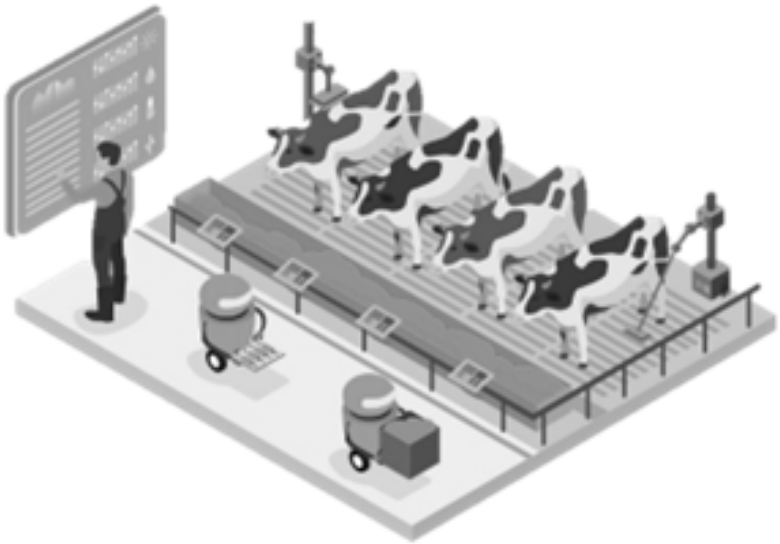
Table of Content

7. Testing

8. Conclusion

Chapter One

Introduction



1.1 Overview

The Dairy Firm Management System (DFMS) is an integrated digital platform designed to simplify and optimise the day-to-day operations of a modern dairy enterprise. Traditional farm management practices often rely on manual record-keeping, paper-based logs, or isolated spreadsheets that create inefficiencies, duplication, and limited data visibility. DFMS replaces these outdated methods with a centralised, web-based environment capable of managing livestock information, tracking feed schedules, recording milk production, and maintaining employee data in real time. By consolidating all operational activities into a unified database, DFMS ensures accuracy, transparency, and speed in managerial decision-making.

The system offers a user-friendly interface and employs robust backend technologies such as ASP.NET Core MVC and MS SQL Server to guarantee scalability and data security. It enables administrators, veterinarians, and employees to collaborate effectively while accessing information according to their designated roles. The project demonstrates how information technology can drive digital transformation in the agricultural sector, improving productivity, cost efficiency, and long-term sustainability for dairy businesses.

Project repository (GitHub): https://github.com/shahidulalam447/DFMS.git

Key Highlights

- Centralised digital management of dairy-farm operations.
- Automation of livestock, feed, and production data.
- Cloud-hosted web application with real-time accessibility.
- Role-based dashboards for admin and staff users.
- Secure data handling via encryption and SQL Server integration.
- Reduction of manual workload and human error.

1.2 Features

DFMS incorporates multiple functional modules that cover every essential aspect of a dairy firm. Each module operates within a shared database, ensuring smooth information flow between departments. The system's modular design allows administrators to manage livestock, feed, and production simultaneously without switching platforms.

Its major components include **Cow Zone** for registering animal details and breeds, **Health Management** for maintaining doctor information, vaccination and treatment records, **Feed Management** for monitoring stock- feed, feed category and consumption, **Milk Monitoring** for milk information tracking, and **Employee Management** for staff profiles information. Additionally, the system provides role-based authentication, detailed reporting, and secure backups, ensuring data integrity.

The interface is intuitive, responsive, and accessible across multiple devices. Users can view automated reports for further analysis.

Core Features

- Cow Zone – cows and cows breed management.
- Health Management – doctors, treatment and vaccine information.
- Feed Management – feed, feed category and consumption.
- Milk Monitoring – milk information.
- Employee Management – employee information, and departmental data.
- Secure Login – role-based access and authorisation.
- Dashboard – visual reports for all cows information summary, milk summary and expenses.
- Responsive Interface – mobile and desktop accessibility.

1.3 Motivation

The primary motivation behind developing DFMS arises from the need to digitise and streamline the complex workflows of dairy farming. In many farms, managers still depend on handwritten records or basic spreadsheets that are prone to loss, inaccuracy, and delay. Such limitations obstruct timely decision-making and reduce overall productivity. Recognising these challenges, DFMS was conceived as a smart, automated solution capable of addressing inefficiencies and supporting data-driven management.

Furthermore, the project was inspired by the growing emphasis on sustainable agricultural practices. Efficient resource management—particularly in feed usage, veterinary care, and milk yield monitoring—has direct implications for both profitability and environmental responsibility. DFMS aligns with these objectives by minimising wastage, improving traceability, and promoting better herd health. It also represents a meaningful contribution to Bangladesh's digital-agriculture initiatives and serves as a model for future technology-enabled farm systems.

Motivational Drivers

- Replace manual processes with automation and precision.
- Promote transparency in livestock and production management.
- Enable evidence-based decisions through real-time data.
- Reduce operational costs and resource wastage.
- Support sustainable and ethical dairy practices.
- Encourage digital transformation in agriculture.
- Contribute to national smart-farming goals.

1.4 Goal of Dairy Firm Management System (DFMS)

The overarching goal of DFMS is to deliver a comprehensive, reliable, and scalable information system that empowers dairy-farm administrators to manage all aspects of their operations efficiently. The system aims to integrate diverse functions—livestock, health, feed, milk, and workforce—within a single interface, reducing duplication of effort and enhancing organisational performance. DFMS focuses on accuracy, automation, and accessibility to create a smarter decision-support framework for farm management.

Beyond operational convenience, DFMS aspires to establish a digital ecosystem that encourages continuous improvement, accountability, and sustainability. By generating detailed analytics and performance metrics, the system supports strategic planning, cost control, and future forecasting. Its modular architecture ensures adaptability to technological upgrades such as IoT integration, AI-driven prediction models, and mobile applications, making it a future-ready platform for the dairy sector.

Project Goals

- Develop an integrated management system for all dairy-farm functions.
- Automate routine tasks to save time and minimise human error.
- Provide real-time visibility into livestock, feed, and milk production data.
- Facilitate secure, role-based user access and data integrity.
- Enhance managerial decision-making through analytics and reports.
- Support sustainable, data-driven dairy operations.
- Ensure scalability for small to large farm enterprises.
- Lay groundwork for future integration with IoT and AI technologies.
- Promote digital adoption within agricultural management systems.

Chapter Two

System Analysis

2.1 Overview

System analysis is the foundation of developing the Dairy Firm Management System (DFMS). It involves examining the existing challenges, identifying inefficiencies in traditional dairy operations, and formulating a structured approach to system improvement. The analysis explores the various processes within a dairy firm — such as livestock management, feeding, health monitoring, milk production, and workforce coordination — and defines how these can be efficiently automated through technology.

Through this phase, all system requirements were collected, interpreted, and converted into design specifications. The goal was to ensure that every stakeholder's needs, from administrators to field employees, were captured and translated into functional and non-functional system requirements. The analysis also evaluated potential risks, scalability options, and the technology stack suitable for the project. By performing detailed observation and consultation, DFMS was modelled as a practical and adaptable solution capable of meeting operational, economic, and technical expectations.

Key Analysis Points

- Study of existing manual record-keeping methods.
- Identification of core operational areas for digital transformation.
- Collection of user and administrative requirements.
- Assessment of hardware and software feasibility.
- Evaluation of process workflows in livestock and feed management.
- Analysis of security, scalability, and usability expectations.
- Alignment of design with sustainability and productivity goals.
- Creation of system models using UML and DFD diagrams.
- Risk identification and mitigation strategy formulation.

2.2 Problem Definition

Traditional dairy firm operations often depend on paper-based records, spreadsheets, and fragmented systems that lack integration. These outdated methods lead to data inconsistency, difficulty in monitoring livestock health, and challenges in managing daily workflows. Managers face delays in accessing critical information related to milk yield, feed stock, or employee performance, which ultimately hampers decision-making and profitability. Moreover, the absence of secure data handling mechanisms exposes farms to risks such as record loss and unauthorised access.

The DFMS project defines this problem as the absence of a unified, automated, and data-driven system for dairy management. It identifies inefficiencies in manual processes, poor coordination between departments, and limited analytical capabilities as major barriers. Therefore, a complete transformation towards an integrated, centralised digital system became essential for modern dairy firms aiming for growth and sustainability.

Identified Problems

- Manual records prone to loss, error, and inconsistency.
- Lack of real-time insights into livestock health or milk production.
- Poor feed management and consumption tracking.
- Inefficient employee data management.
- Time-consuming and inaccurate reporting procedures.
- Inadequate data security and access control.
- Absence of performance analytics for decision-making.
- Difficulty in monitoring multiple departments simultaneously.
- Limited scalability to accommodate farm expansion.

2.3 Existing System

The existing system before DFMS implementation consisted mainly of manual processes and basic spreadsheet tracking. In most dairy firms, employees recorded cow health details, feeding schedules, and milk yield data on paper registers or simple Excel sheets. This method lacked coordination between departments, making it hard to access updated information. For instance, when a veterinarian treated a cow, the health record was rarely synchronised with milk production data or feeding adjustments. Such fragmented systems resulted in inconsistent performance tracking and poor accountability.

Additionally, reports were generated manually, consuming valuable time and increasing the likelihood of human error. As the farm's operations scaled, the amount of data grew exponentially, overwhelming manual processes. The absence of a secure, integrated database meant data duplication, inefficiency, and limited analytical capacity. This scenario underscored the urgent need for a digital system like DFMS to unify data handling and streamline all farm activities.

Limitations of Existing System

- Manual data entry with low accuracy.
- Time-intensive record retrieval and report generation.
- Lack of centralised database or automation tools.
- No real-time data tracking or alerts.
- Weak communication between management and staff.
- Inefficient data sharing across departments.
- Difficult to maintain data security or backups.
- Inability to forecast production or analyse trends.
- High dependency on human supervision.

2.4 Proposed System

The proposed Dairy Firm Management System (DFMS) resolves the inefficiencies of the existing system by providing a unified, web-based, and automated platform. The system integrates all major operational domains—cows, health, feed, milk, and workforce—under one interface. It offers administrators a real-time dashboard to monitor performance, while employees and veterinarians can update records instantly through secure login access. DFMS enables quick data retrieval, generates analytical reports, and enhances decision-making through automation and data visualisation.

Technically, the system is built using **ASP.NET Core MVC** for the backend, **Entity Framework Core** for ORM data access, and **Microsoft SQL Server** for database management. Its responsive frontend is developed with **HTML, CSS, JavaScript, and Bootstrap**, ensuring accessibility from any device. The system design supports scalability, allowing integration of advanced technologies like IoT or AI in the future. Overall, DFMS provides an efficient, secure, and user-friendly environment that ensures accuracy, speed, and sustainability in dairy management.

Features of the Proposed System

- Centralised database with real-time updates.
- Secure role-based access for admin and employees.
- Integrated modules for cows, feed, and milk management.
- Automated reporting and performance analytics.
- User-friendly interface with responsive design.
- Cloud-based deployment ensuring reliability and uptime.
- Support for scalability and modular enhancement.
- Reduced operational workload and faster decision-making.

Chapter Three

Feasibility Study

3.1 Overview

A feasibility study determines whether a proposed system is viable from operational, economic, and technical perspectives. For the Dairy Firm Management System (DFMS), the feasibility study evaluates the practicality of implementing an integrated web-based management platform to replace manual methods. The objective was to ensure that the system can be efficiently adopted, maintained, and expanded without disrupting ongoing operations.

This stage assessed all critical parameters — including cost implications, technical resources, skill requirements, and long-term benefits — to verify whether the DFMS project aligns with both organisational goals and user expectations. After analysing existing conditions and resource availability, the study confirmed that the system is not only feasible but also essential for achieving efficiency, reliability, and scalability in dairy operations.

Key Insights

- Comprehensive evaluation of project cost, resources, and timeline.
- Identification of technical requirements and compatibility factors.
- Verification of the project's economic and operational benefits.
- Assessment of human resource capabilities and training needs.
- Determination of system sustainability and long-term adaptability.
- Risk analysis and mitigation planning.
- Confirmation of project feasibility within existing constraints.
- Clear justification for implementation and investment.
- Alignment with institutional IT goals and agricultural innovation.

3.2 Operational Feasibility

Operational feasibility examines whether the DFMS can function smoothly within the organisation's workflow and whether users can adopt it effectively. The study revealed that the system's web-based nature offers significant operational advantages. It supports multi-user access, real-time updates, and secure data handling, ensuring that administrators and employees can perform daily tasks efficiently.

The interface was designed to be intuitive, reducing the learning curve for users with minimal technical expertise. Staff can easily log milk production, update health records, or manage feed schedules from a single dashboard. Additionally, role-based access ensures that employees interact only with data relevant to their responsibilities, preventing unauthorised access. Overall, DFMS integrates seamlessly into existing operations, improving coordination, communication, and productivity across all departments.

Operational Strengths

- Web-based access allowing remote and multi-user operations.
- Minimal training required due to simple, intuitive design.
- Streamlined data entry and retrieval processes.
- Role-based access enhancing accountability and data security.
- Compatibility with existing computing infrastructure.
- Centralised data flow improving inter-departmental coordination.
- Real-time updates enabling timely decisions.
- Automated reports reducing administrative burden.
- Positive user acceptance and adaptability among staff.

3.3 Economic Feasibility

Economic feasibility assesses whether the project's financial benefits outweigh its costs. The DFMS project required an initial investment in software development, database setup, and hosting infrastructure. However, the long-term financial savings are substantial. Automation reduces the need for manual record-keeping, printing, and data reconciliation, cutting operational expenses. It also minimises losses from errors in feed distribution or milk production tracking, leading to improved profit margins.

The use of open-source and widely available technologies such as ASP.NET Core and SQL Server Express Edition further lowers software licensing costs. Cloud deployment options like AWS or Azure provide scalability, eliminating the need for expensive on-premise servers. Consequently, the DFMS project is considered highly cost-effective, with a strong return on investment within the first few operational years.

Economic Highlights

- Low-cost development using open-source tools.
- Reduction in administrative and data-entry expenses.
- Elimination of paper-based record costs.
- Increased productivity leading to higher profitability.
- Improved data accuracy reducing financial losses.
- Minimal maintenance and upgrade costs.
- Cloud hosting reducing infrastructure investment.
- Long-term operational cost savings.
- High return on investment (ROI) within short timeframe.

Category	***Description***	***Estimated Cost (BDT)***
Hardware	Basic server or hosting setup	50,000
Software	Microsoft Visual Studio 2022, Microsoft Visio 2013	00
Development	SRS, design, coding, testing	2,000
Maintenance	Annually	10,000
Others		5,000

3.4 Meeting with Expert Engineering

To ensure technical accuracy and practical feasibility, meetings were conducted with experts in software engineering, agricultural informatics, and system design. These consultations provided valuable insights into aligning the DFMS architecture with best practices in software development and agricultural automation. Experts emphasised modularity, scalability, and robust database management as key priorities.

They also recommended incorporating user feedback loops during testing phases to improve usability. The consultation process validated that the proposed technology stack—ASP.NET Core MVC, Entity Framework, and SQL Server—was appropriate for achieving system performance and stability. Moreover, engineering experts confirmed that the system design adhered to industry standards for security, accessibility, and data integrity, ensuring its sustainability and adaptability for future enhancements.

Expert Consultation Outcomes

- Validation of system architecture and development framework.
- Recommendations for security, modularity, and scalability.
- Confirmation of data integrity and performance standards.
- Guidance on integrating feedback-based iteration cycles.
- Support for Agile development methodology.
- Suggestions for optimising server and database configuration.
- Advice on user interface responsiveness and accessibility.
- Technical approval for proposed implementation plan.
- Assurance of compliance with software engineering best practices.

Chapter Four

Requirement Analysis

4.1 Overview

Requirement analysis is a crucial phase in the system development life cycle (SDLC), aimed at identifying, documenting, and validating all the essential system needs before design and implementation. For the Dairy Firm Management System (DFMS), this phase involved close interaction with end-users, administrators, and technical experts to gather comprehensive information on the operational requirements of a dairy farm.

The analysis ensured that all aspects—functional, non-functional, and environmental—were well understood and translated into actionable design elements. It covered both user expectations and technical constraints, ensuring the system aligns with practical realities. The process also involved preparing requirement specification documents that guided development and testing. Through this systematic approach, DFMS was positioned to deliver accuracy, scalability, and real-time operational efficiency while addressing every functional area of dairy management.

Key Outcomes

- Identification of all operational workflows and data dependencies.
- Categorisation of requirements into functional and non-functional groups.
- Validation of system requirements with end-user feedback.
- Documentation following IEEE SRS (Software Requirement Specification) standards.
- Prioritisation of modules based on operational impact.
- Definition of performance benchmarks and user constraints.
- Mapping of data flow and process relationships.
- Establishment of a strong foundation for system design.
- Assurance of alignment between user needs and technical feasibility.

4.2 Functional Requirements

Functional requirements define what the DFMS must accomplish to meet its objectives. Each functional module within the system is designed to automate a key operational process within the dairy firm, from livestock registration to feed monitoring, milk production tracking, and employee record management. These requirements were developed to ensure seamless integration between all functional areas, allowing users to execute operations efficiently and with minimal manual intervention.

Each function supports standard CRUD (Create, Read, Update, Delete) operations, ensuring complete control over data management. For instance, the *Cow Zone* module maintains breed and cow information; *Health Management* logs vaccinations, treatments, and veterinary visits; *Feed Management* tracks inventory, purchasing, and consumption; *Milk Monitoring* records daily output; and *Employee Management* maintains personnel and payroll records. Together, these modules form a unified and interlinked operational framework for dairy firms.

Functional Requirements

- Cow Zone: Cows, breed information tracking.
- Health Management: Doctor, treatment, and vaccine records.
- Feed Management: Category, stock, and consumption details.
- Milk Monitoring: Milk information tracking.
- Employee Management: Employee information, department, grade, and bank details.
- Authentication: Admin and employee login with role-based access.
- Reporting: Automated reports (daily, monthly).
- Dashboard: Summary of visual reports for all cows information, milk summary and expenses.
- Database: Centralised data storage with CRUD operations.

4.3 Non-Functional Requirements

Non-functional requirements describe the performance standards, usability, and quality benchmarks that ensure DFMS operates efficiently and securely. These aspects focus on *how* the system performs rather than *what* it performs. For DFMS, the design prioritised speed, data security, scalability, and ease of use. The system must provide a responsive interface, reliable uptime, and robust data protection through encryption and authentication mechanisms.

In addition, the system should handle concurrent users efficiently and offer an intuitive layout that allows even non-technical personnel to operate comfortably. Regular data backups, modular coding, and maintainability standards ensure longevity and future adaptability. These non-functional attributes ensure that DFMS remains a sustainable and secure platform suitable for real-world dairy operations.

Non-Functional Standards

- **Performance:** Response time below 2 seconds for key actions.
- **Scalability:** Expandable to support additional farms or modules.
- **Security:** Encrypted data storage and multi-level authentication.
- **Usability:** User-friendly, mobile-responsive interface.
- **Reliability:** 99.5% uptime in cloud environment.
- **Maintainability:** Modular design for easier updates.
- **Compatibility:** Works across all major browsers and devices.
- **Accessibility:** Web-based interface with multi-user access.
- **Compliance:** Conforms to modern software security standards.

4.4 Technologies

The DFMS employs a modern, secure, and scalable technology stack to ensure optimal performance and maintainability. The backend is developed using **ASP.NET Core MVC**, which supports the Model–View–Controller architecture, promoting separation of concerns and modularity. For data storage, **Microsoft SQL Server** provides structured database management and query optimisation. The ORM layer uses **Entity Framework Core**, which facilitates seamless database communication.

The frontend employs **HTML, CSS, JavaScript**, and **Bootstrap** to deliver an interactive, responsive user interface compatible with both desktop and mobile devices. The system can be hosted on cloud platforms such as **AWS** or **Microsoft Azure**, ensuring reliability and accessibility. Version control is maintained using **GitHub**, while testing is conducted through **NUnit/xUnit** frameworks. Together, these technologies form a cohesive and future-ready foundation for efficient, scalable, and secure dairy management.

Technology Stack Overview

- **Backend Framework:** ASP.NET Core MVC.
- **Database Management:** Microsoft SQL Server.
- **ORM Tool:** Entity Framework Core.
- **Frontend Tools:** HTML, CSS, JavaScript, Bootstrap.
- **Deployment Platforms:** AWS, Microsoft Azure, or IIS.
- **Version Control:** GitHub repository integration.
- **Testing Frameworks:** NUnit or xUnit for automated testing.
- **Development Methodology:** Agile and iterative approach.
- **Security Layer:** Data encryption and role-based authentication.
- **Integration Support:** Ready for future IoT and AI-based modules.

Chapter Five

System Design

5.1 Overview

System design is the blueprint phase of software development where all analysed requirements are translated into a practical structure for implementation. For the Dairy Firm Management System (DFMS), the design stage focuses on establishing the overall architecture, defining system components, data flow, and interactions between modules. The design aims to ensure efficiency, scalability, and maintainability across all operational layers of the system.

The DFMS is structured around modular and layered design principles. The system architecture divides functions into presentation, business logic, and database layers — each performing a specific role. The frontend handles user interaction through a responsive web interface, the middle layer processes logic, and the backend layer manages persistent data storage. Each module (cow management, feed monitoring, milk tracking, and employee management) interacts through secure API calls and database transactions, ensuring seamless workflow integration and data integrity.

Design Highlights

- Structured system blueprint translating requirements into implementation.
- Three-tier architecture ensuring separation of concerns.
- Modules designed for scalability and flexibility.
- Logical connections between frontend, backend, and database layers.
- Integration of data security and access control mechanisms.
- Adoption of standardised naming conventions and coding practices.
- UML-based documentation for system representation.
- High maintainability for future updates or feature expansion.
- Data consistency ensured through relational integrity rules.

5.2 Architectural Design

The DFMS architecture follows the **three-tier architecture** model: **Presentation Layer**, **Application Layer**, and **Database Layer**.

1. The *Presentation Layer* handles the user interface (UI) through HTML, CSS, JavaScript, and Bootstrap. It enables interactions such as login, data entry, and report viewing.
2. The *Application Layer* contains business logic written in ASP.NET Core MVC, processing user requests, applying validation, and coordinating between the UI and database.
3. The *Database Layer* stores and retrieves data using Microsoft SQL Server, with Entity Framework Core facilitating ORM (Object Relational Mapping).

This architecture provides separation of logic, reusability, and security while enabling performance optimisation. The system is scalable for multiple users and adaptable for cloud deployment. By isolating each functional unit, DFMS ensures smooth maintenance, better performance, and flexible upgrades without affecting other modules.

Architectural Features

- Three-tier architecture: Presentation, Logic, Database layers.
- ASP.NET Core MVC framework supporting modular design.
- Entity Framework Core for efficient ORM data handling.
- SQL Server ensuring structured and secure data storage.
- Authentication and authorisation integrated in logic layer.
- Improved fault isolation and maintainability.
- Enhanced system scalability and performance optimisation.

5.3 Logical Design

The logical design of DFMS defines how different components interact within the system from a conceptual point of view. It focuses on data flow, module dependencies, and logical relationships rather than physical implementation. The logical model illustrates the movement of data between the user interface, business logic, and database.

Each module communicates with the central database using defined operations such as Create, Read, Update, and Delete (CRUD). Logical connections are established between functional modules—for instance, the Cow Health module directly interacts with Cow Records and Feed Management modules to ensure accurate tracking of livestock health and nutrition. Logical design also defines constraints, validation rules, and control mechanisms that maintain the integrity of the system during transactions and reporting.

Logical Design Key Points

- Conceptual model describing system behaviour and relationships.
- Interconnection of functional modules via CRUD operations.
- Logical association between cow, feed, and health data.
- Consistent validation rules for accuracy and reliability.
- Integration of feedback loops for error control.
- Role-based restrictions implemented at logic layer.
- Synchronised workflow ensuring real-time updates.
- Designed to support modular and cross-functional processing.
- Focus on data consistency and inter-module dependency control.

5.4 Physical Design

Physical design translates the logical model into actual implementation specifications, including server setup, database structure, and network configuration. It defines how data will be stored, accessed, and managed in the live environment. For DFMS, physical design ensures system reliability and performance by optimising hardware and software resources.

The web application will be hosted on a cloud server (AWS/Azure/IIS) that guarantees scalability and uptime. Microsoft SQL Server manages relational databases with tables for cows, breeds, feed, treatments, milk records, and employees. Indexing and foreign key relationships ensure efficient data retrieval and referential integrity. The physical design also defines data backup strategies, access permissions, and server security configurations. Together, these elements guarantee that DFMS functions effectively in a real-world operational setting.

Physical Design Specifications

- Cloud-based hosting for high uptime and reliability.
- Microsoft SQL Server database with relational schema.
- Structured tables for each module (Cows, Feed, Milk, Employees).
- Data indexing for faster query processing.
- Role-based access and encryption for data protection.
- Load-balancing strategy for performance stability.
- Hardware configuration ensuring high scalability.
- Network protocols using secure HTTPS communication.

5.5 Use Case Diagram

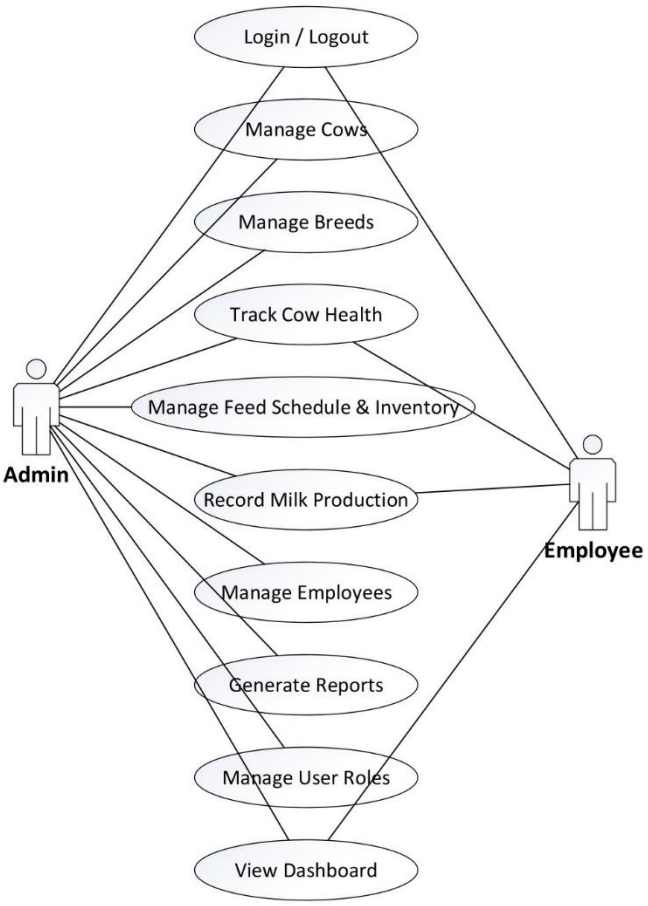
The **Use Case Diagram** represents system functionality and the interaction between users and the DFMS system. The two primary actors are **Admin** and **Employee**.

- The *Admin* actor has complete access, including system setup, data management, user authentication, and report generation.
- The *Employee* actor has limited access to perform specific tasks like updating milk production, feed usage, and health data.

This visual model demonstrates how different users engage with system modules, ensuring clarity in defining user privileges and operational flow. The diagram confirms that the system supports multi-role operations while maintaining strict access control and accountability.

Key Use Case Features

- Two main actors: Admin and Employee.
- Admin: Full privileges across all modules.
- Employee: Restricted role-based actions.
- Functional cases: Manage cows, feed, health, milk, employees.



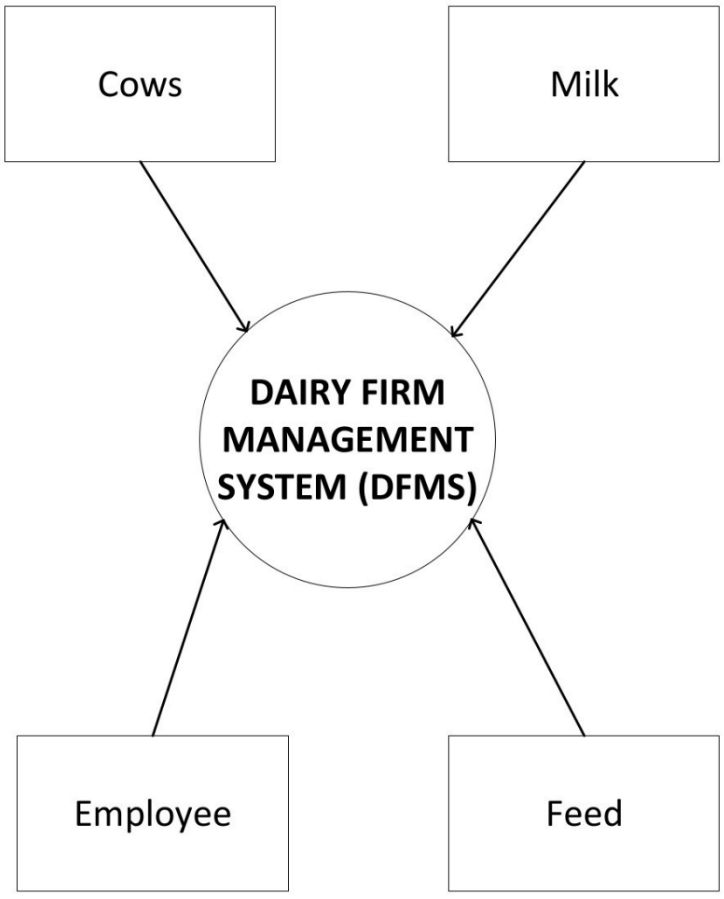
5.6 Data Flow Diagram (DFD) – Level 0

The **Level 0 DFD** provides a high-level overview of the entire DFMS system, showing data movement between external entities (users) and system processes. It presents how input data from users is processed and stored in the database, and how output is generated through reports or dashboards.

The main entities include Admin and Employee. They interact with processes like Cow Management, Feed Management, Milk Monitoring, and Health Tracking. Data flows between these processes and the central database, which stores and retrieves information as required. This level ensures that the system's functional boundaries and data dependencies are clearly identified.

DFD Level 0 Elements

- Admin and Employee as primary external entities.
- Central database storing all system information.
- Four key processes: Cows, Employee, Feed, Milk.
- Input/output flow between users and modules.



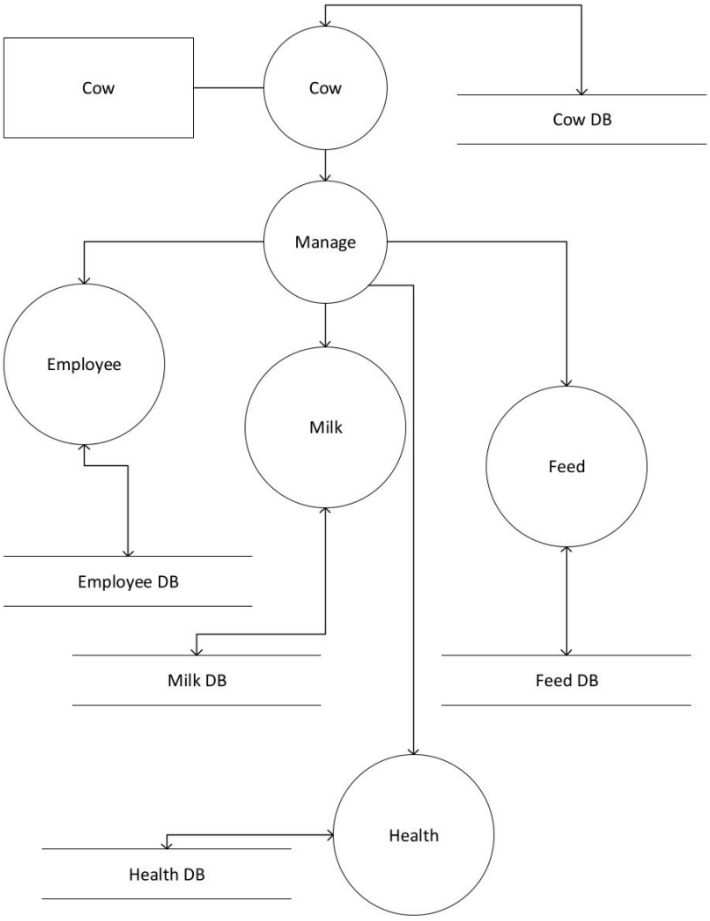
5.7 Data Flow Diagram (DFD) – Level 1

The **Level 1 DFD** expands the Level 0 diagram by illustrating individual processes in greater detail. Each major function is broken down into sub-processes, showing specific operations like “Add Cow Record,” “Update Feed Data,” or “Record Milk Quantity.”

It demonstrates how data flows internally between these sub-modules and the central database. For example, the Health module interacts with Doctor and Treatment tables, while the Feed Management module communicates with Feed Category and Feed Consumption tables. This hierarchical model ensures traceability, transparency, and accurate mapping of all operational workflows.

DFD Level 1 Highlights

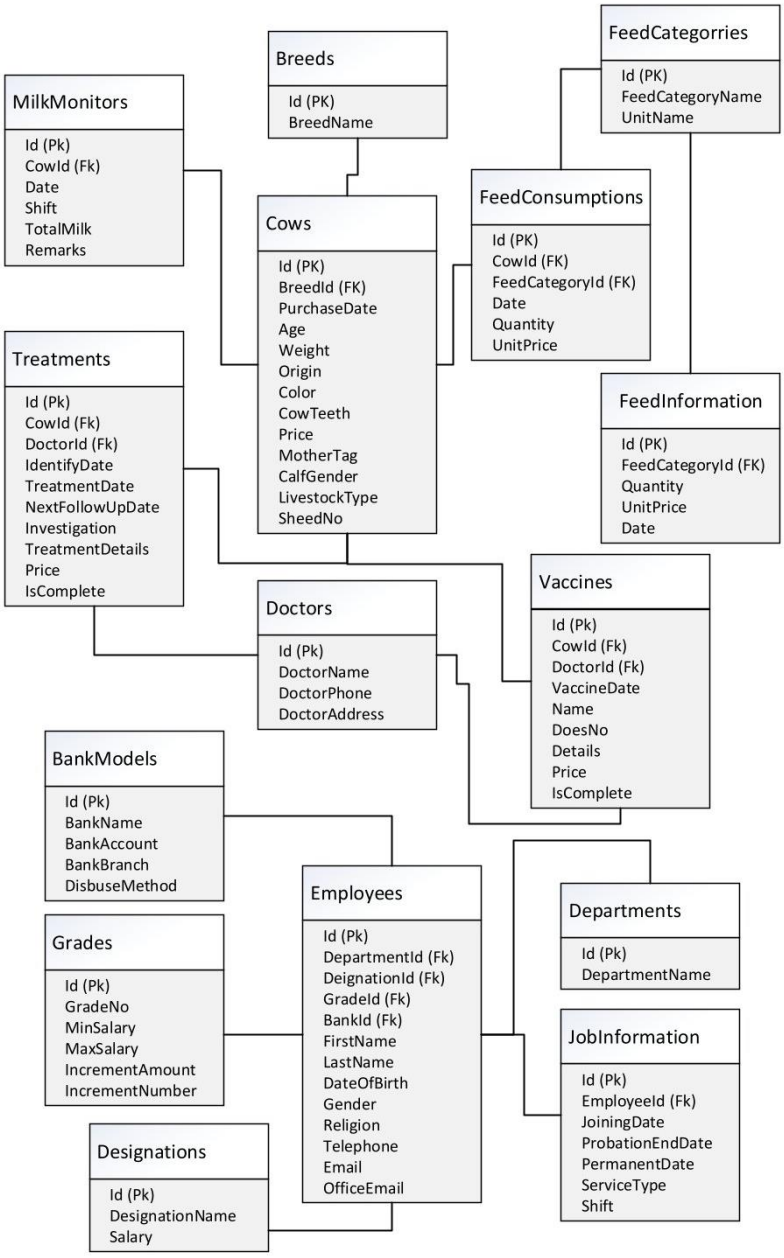
- Decomposition of main processes into sub-modules.
- Clear depiction of internal data transactions.
- Interaction among Cow, Employee, Feed, Milk, and Health subsystems.
- Real-time data updates in SQL Server database.
- Defined inputs and outputs for each module.



5.8 Entity–Relationship (ER) Diagram

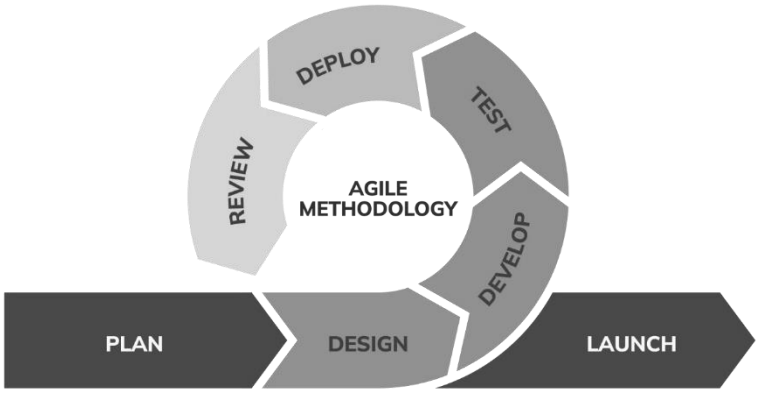
The **ER Diagram** represents the logical structure of the DFMS database, defining entities, attributes, and relationships. Each table corresponds to a functional area: Cows, Breeds, Doctors, Treatments, Vaccines, Feed Information, Feed Category, Milk Monitors, Employees, Departments, Banks, and more.

Relationships are established through primary and foreign keys, ensuring data integrity and relational consistency. For instance, the *Cows* entity links with *Breeds* and *Health Records*, while *Employees* connect with *Departments* and *Job Information*. The ER model provides a clear representation of how entities interact within the database, enabling efficient data management, query optimisation, and reporting accuracy.



Chapter Six

Implementation



6.1 Project Planning

The implementation phase of the Dairy Firm Management System (DFMS) marks the transition from system design to real-world execution. It involves converting design specifications into functional code, configuring databases, integrating modules, and deploying the application in a test or live environment. Effective project planning was essential to ensure the project remained within schedule, scope, and budget.

A structured Agile methodology was adopted, allowing for iterative development and regular feedback from stakeholders. The development cycle was divided into several sprints — each focusing on specific modules such as Cow Management, Health Tracking, Feed Inventory, Milk Monitoring, and Employee Management. Each sprint concluded with testing and review sessions to identify improvements. The project plan also accounted for risk management, data security, and scalability to support future enhancements. A detailed timeline was established, including development, testing, deployment, and maintenance stages, to ensure systematic progress and deliverables.

Planning Highlights

Phase	***Duration (Weeks)***	***Description***
Project proposal	4	Web-based platform
Requirement Analysis & Planning	4	Gathering and documenting requirements, stakeholder meetings, and system feasibility study.
System & Database Design	4	Designing architecture, database schema, and wireframes for all system modules.
Implementation	6	Developing backend and frontend modules,
Testing & Debugging	4	Conducting unit, integration, and user acceptance testing; identifying and resolving issues.
Deployment	2	Hosting DFMS on cloud platforms (AWS/Azure/IIS) and ensuring smooth performance post-launch.
Maintenance & Support	2 (Basic)	Monitoring, patch updates, performance optimisation, and user support.

6.2 Admin Panel Login

The Admin Panel serves as the core access point for system administrators, offering comprehensive control over all DFMS modules. Implementation began by designing a secure and user-friendly login interface, integrated with **ASP.NET Core Identity Framework** for authentication and authorisation. This ensured that only authorised users could access administrative features.

During login, user credentials are verified against encrypted records in the SQL Server database. Upon successful validation, the system generates a secure session token and redirects the admin to the dashboard. Multi-level authentication prevents unauthorised access, while role-based privileges define the scope of operations available to each user. The login module also includes password encryption, password reset functionality, and activity logging for accountability.

Admin Login Features

- Secure login using ASP.NET Core Identity and token authentication.
- Role-based access to system modules and functions.
- Validation against SQL Server credentials.
- Protection against unauthorised access and data breaches.
- User activity logs for monitoring and auditing.
- Responsive login interface for multiple devices.
- Logout functionality ensuring session security.

6.3 Admin Panel Activities

Once authenticated, administrators can perform a wide range of activities via the Admin Panel. This interface provides direct access to all major system modules—Cow Zone, Health Management, Feed Management, Milk Monitoring, and Employee Management—through an intuitive dashboard. Each section is built with CRUD functionalities, enabling admins to create, read, update, and delete data entries efficiently.

The dashboard also displays analytical summaries, such as total cows, recent medical expenses, milk output trends, and feed consumption patterns. Real-time visual reports assist administrators in identifying production efficiency and animal health indicators. The Admin Panel includes advanced options for managing user roles, database backups, and report exports (PDF/Excel). The integration of responsive design ensures accessibility from desktops, tablets, or mobile devices, supporting flexible operations.

Core Admin Functions

- Access to all modules through central dashboard.
- CRUD operations for cows, feed, health, and employees.
- Real-time display of key performance indicators (KPIs).
- Data visualisation through reports.
- Role management for admin and employee users.
- Report export.
- Centralised control over database entries.
- Device-responsive dashboard for accessibility.

6.4 Admin Panel Operation

The Admin Panel Operation module was developed to streamline all daily administrative processes. It acts as the control centre for monitoring data input, verifying transactions, and maintaining overall system functionality. Through this module, administrators can monitor livestock records, update vaccination and treatment details, schedule feed distribution, record milk production, and oversee employee performance.

Each function is fully integrated with the central database, ensuring that updates in one module automatically reflect across related sections. For example, when a cow's health record is updated, the system synchronises relevant data within feed and milk production records. Additionally, admin users can generate analytical reports on productivity, feed cost, and yield performance. All operations are logged for transparency and future auditing.

Operational Features

<table>
<tr><td colspan="3">Logo</td><td colspan="2">User Information</td></tr>
<tr><td colspan="3">Navigation Bar:</td><td colspan="2" rowspan="14">Dashboard:

Total Cow

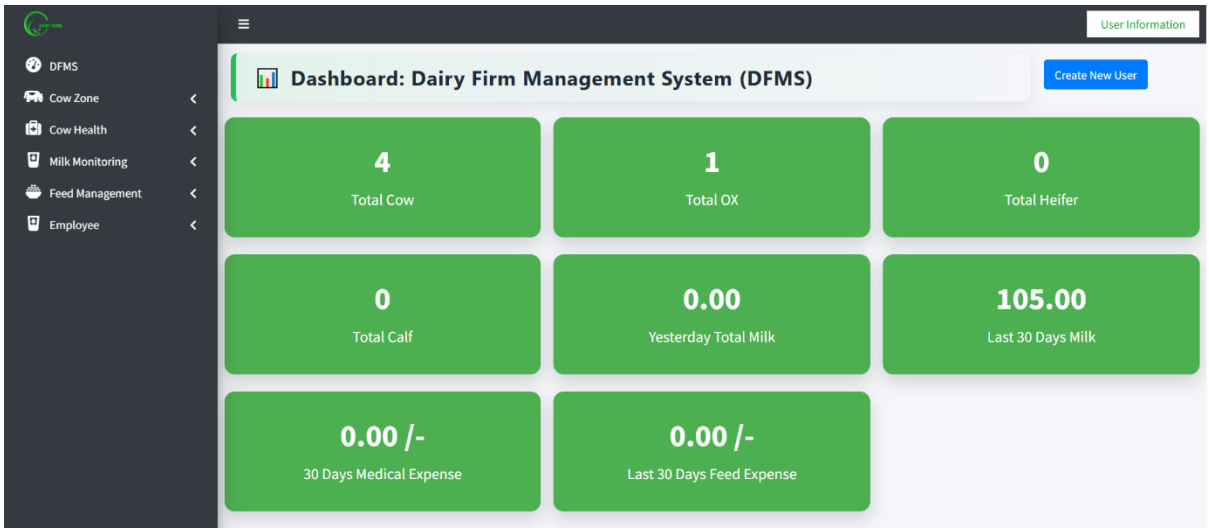
30 Days Medical Expense

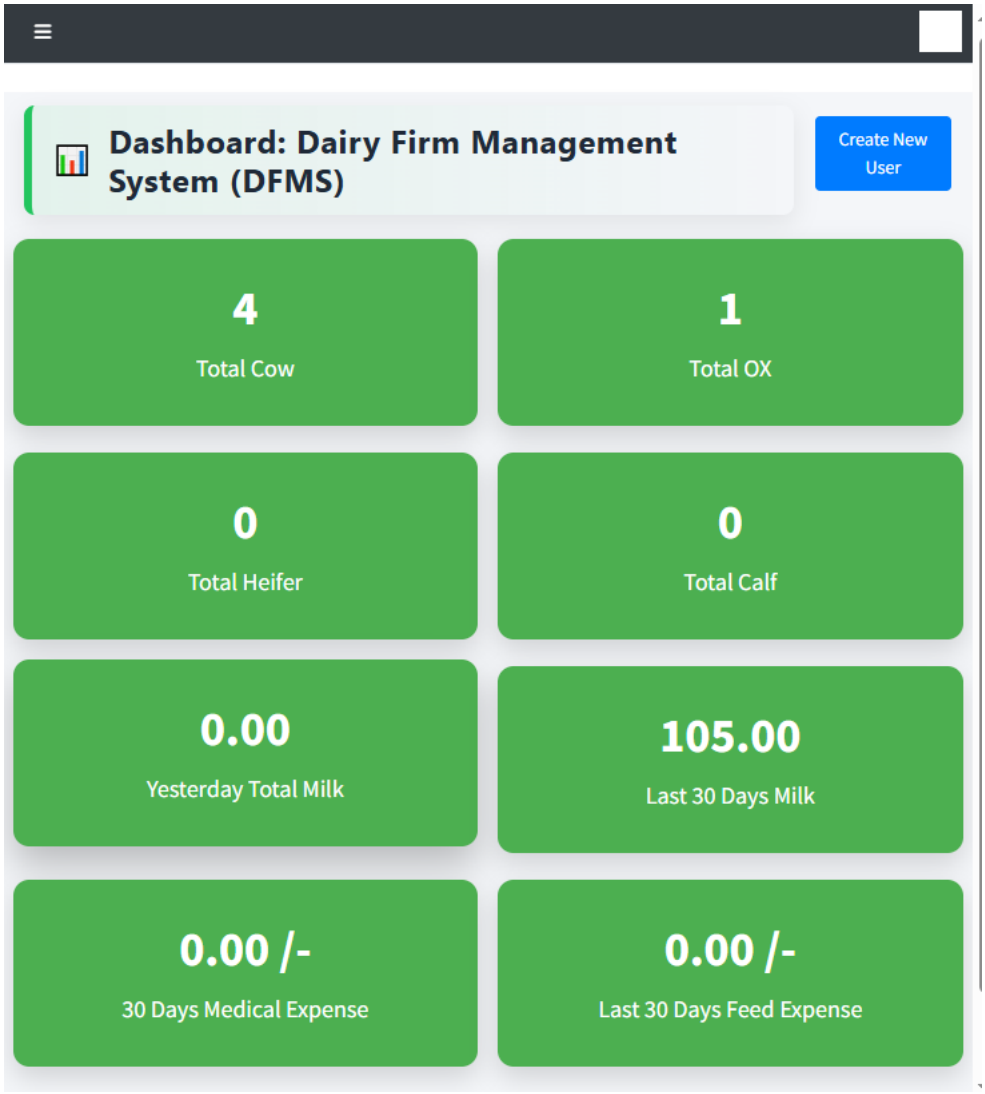
Yesterday Total Milk

Last 30 Days Feed Expense

Last 30 Days Milk</td></tr>
<tr><td rowspan="2">Cow Zone</td><td colspan="2">Cows</td></tr>
<tr><td colspan="2">Breeds</td></tr>
<tr><td rowspan="3">Cow Health</td><td colspan="2">Doctor</td></tr>
<tr><td colspan="2">Treatment</td></tr>
<tr><td colspan="2">Vaccine</td></tr>
<tr><td colspan="3">Milk Monitoring</td></tr>
<tr><td rowspan="3">Feed Management</td><td colspan="2">Feed Category</td></tr>
<tr><td colspan="2">Feed Information</td></tr>
<tr><td colspan="2">Consumptions</td></tr>
<tr><td rowspan="4">Employee Management</td><td colspan="2">Employee Information</td></tr>
<tr><td colspan="2">Bank Information</td></tr>
<tr><td colspan="2">Grade</td></tr>
<tr><td colspan="2">Department</td></tr>
</table>

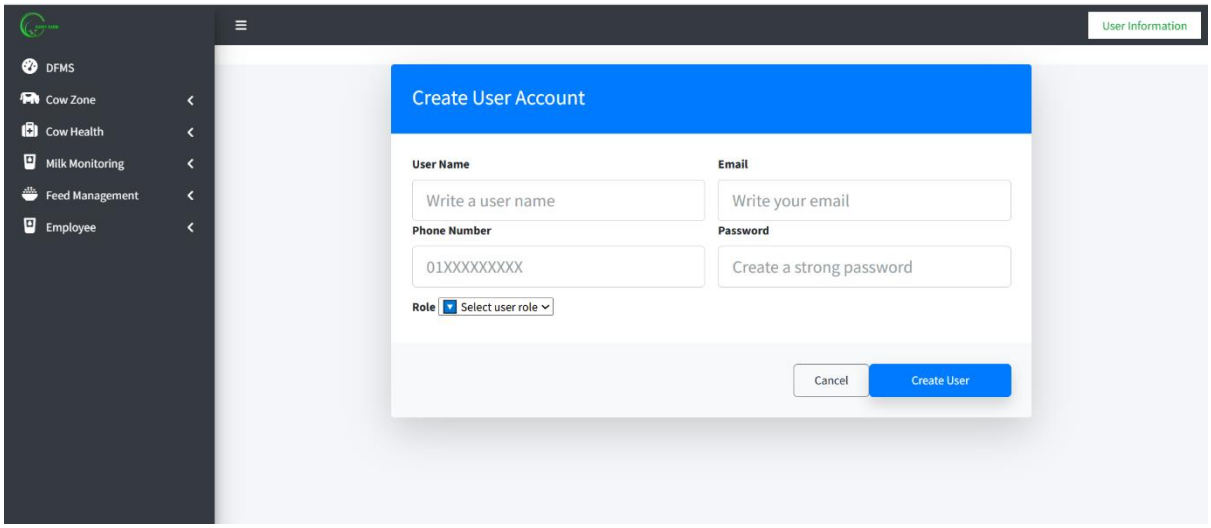
Admin Desktop View



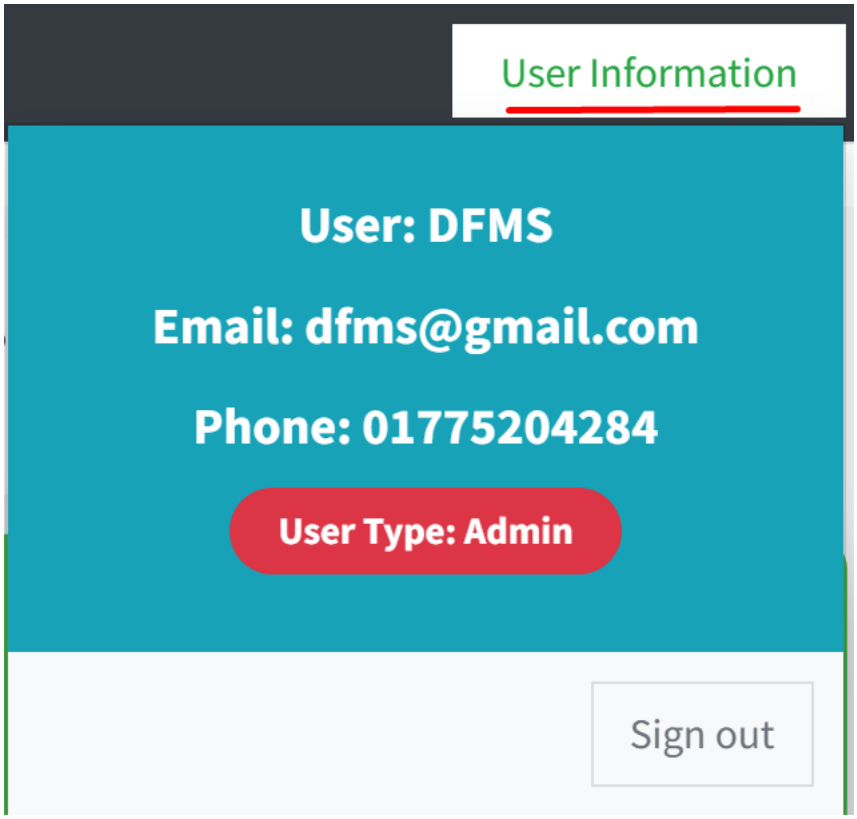


Admin Mobile View

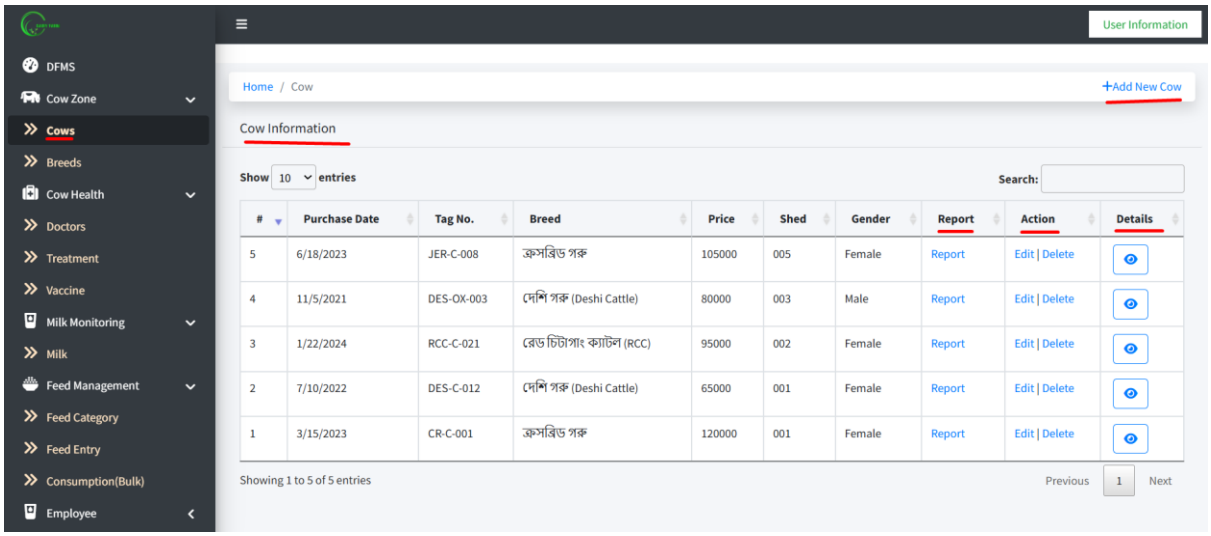
Create User Account



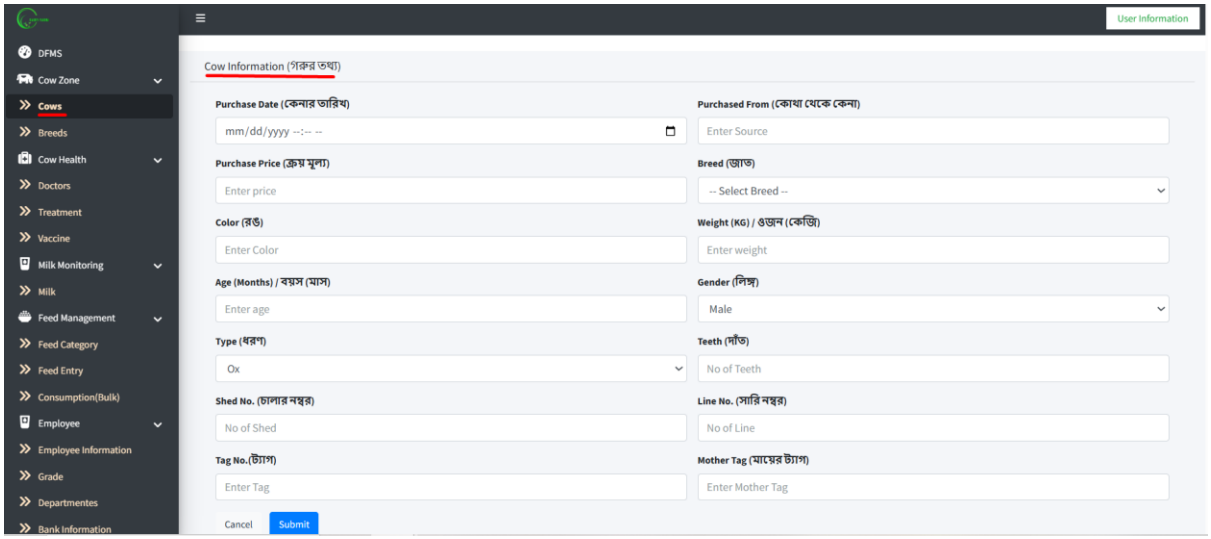
User Information View



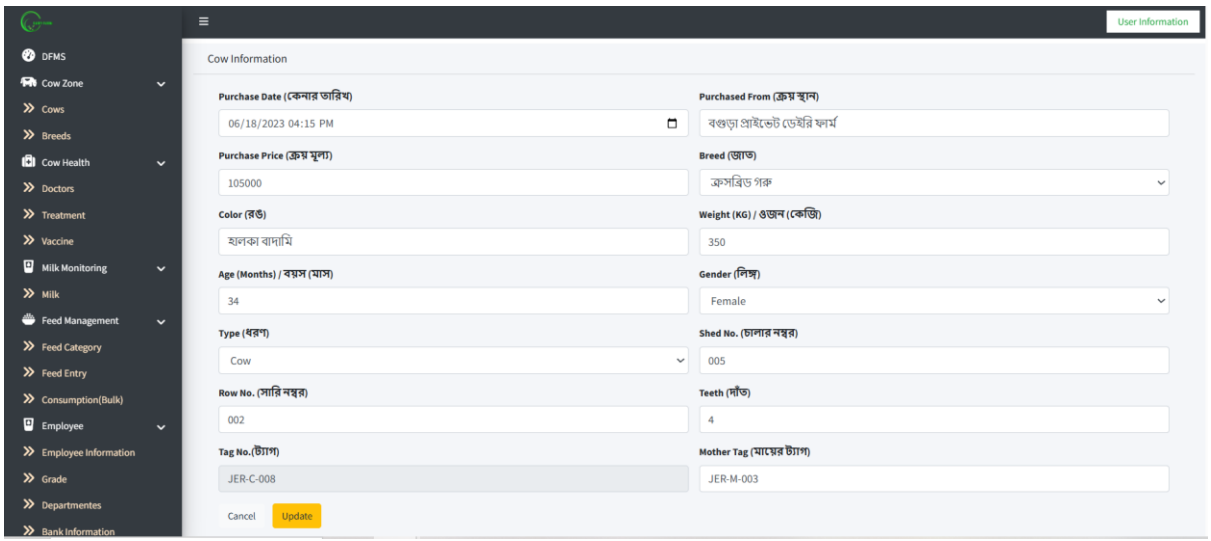
Cow Information



Cow Information Add



Cow Information Update



Cow Information Report

Print Report

DAIRY FIRM MANAGEMENT SYSTEM (DFMS)
Sylhet, Bangladesh · Phone: 01700000000
Cow Tag: **JER-C-008** · Type: **Cow** · Breed: ক্রসব্রিড গরু

Cow Summary Report
Print Date: 15 Feb 2026, 07:02 PM

SUMMARY

Buying Price (A)	Treatment Cost (B)	Vaccine Cost (C)	Feeding Cost (D)	Total Cost (E)	Milk Produced (F)
105000.00	0.00	0.00	0.00	105000.00	0 litre

MILK

Total Produced Milk
0 litres

FEEDING

Total Quantity	Cost
0	0.00

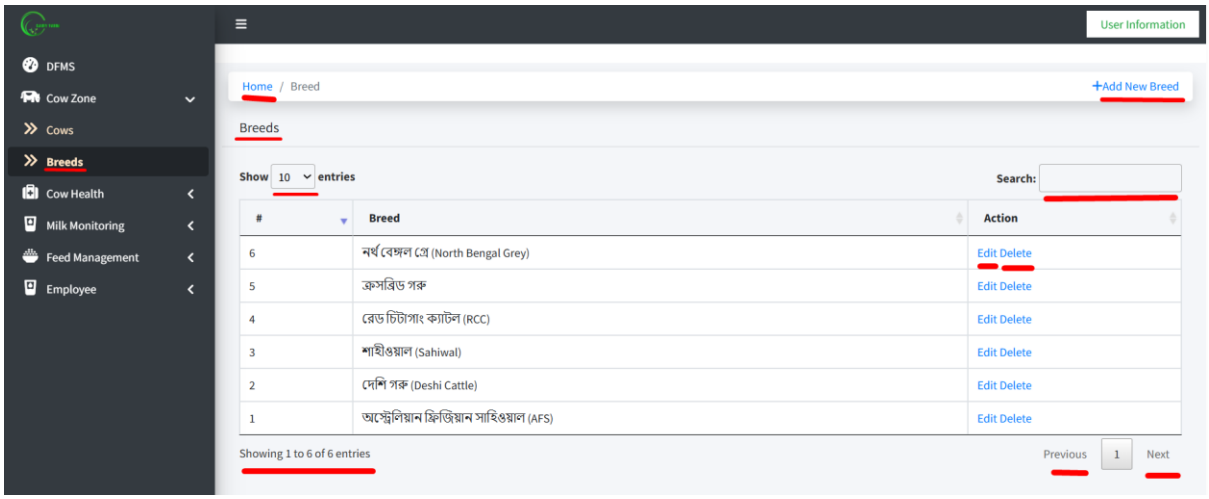
VACCINE

Date	Description	Cost
	Total Vaccine Cost	0.00

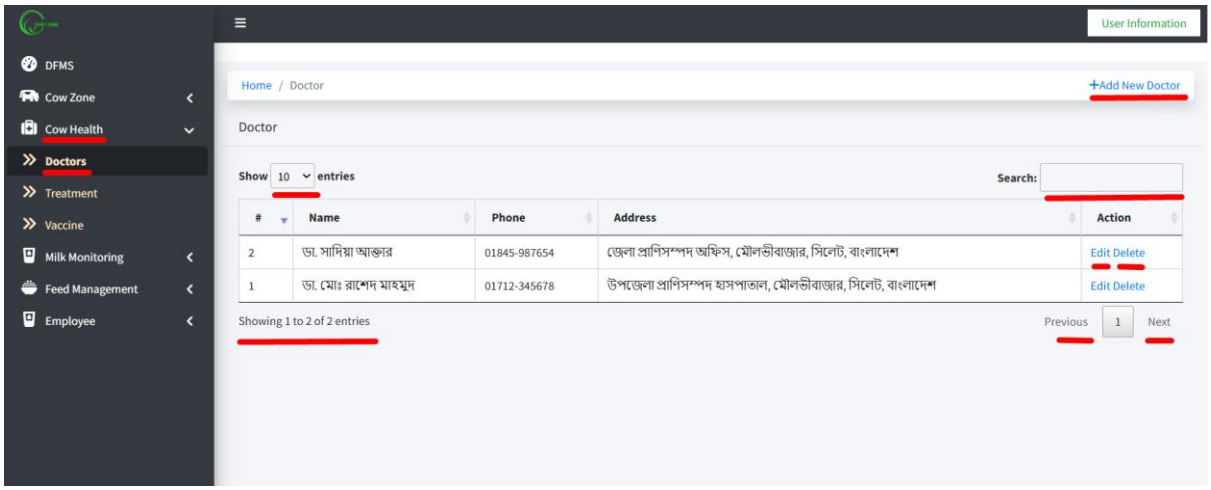
TREATMENT

Date	Description	Cost
	Total Treatment Cost	0.00

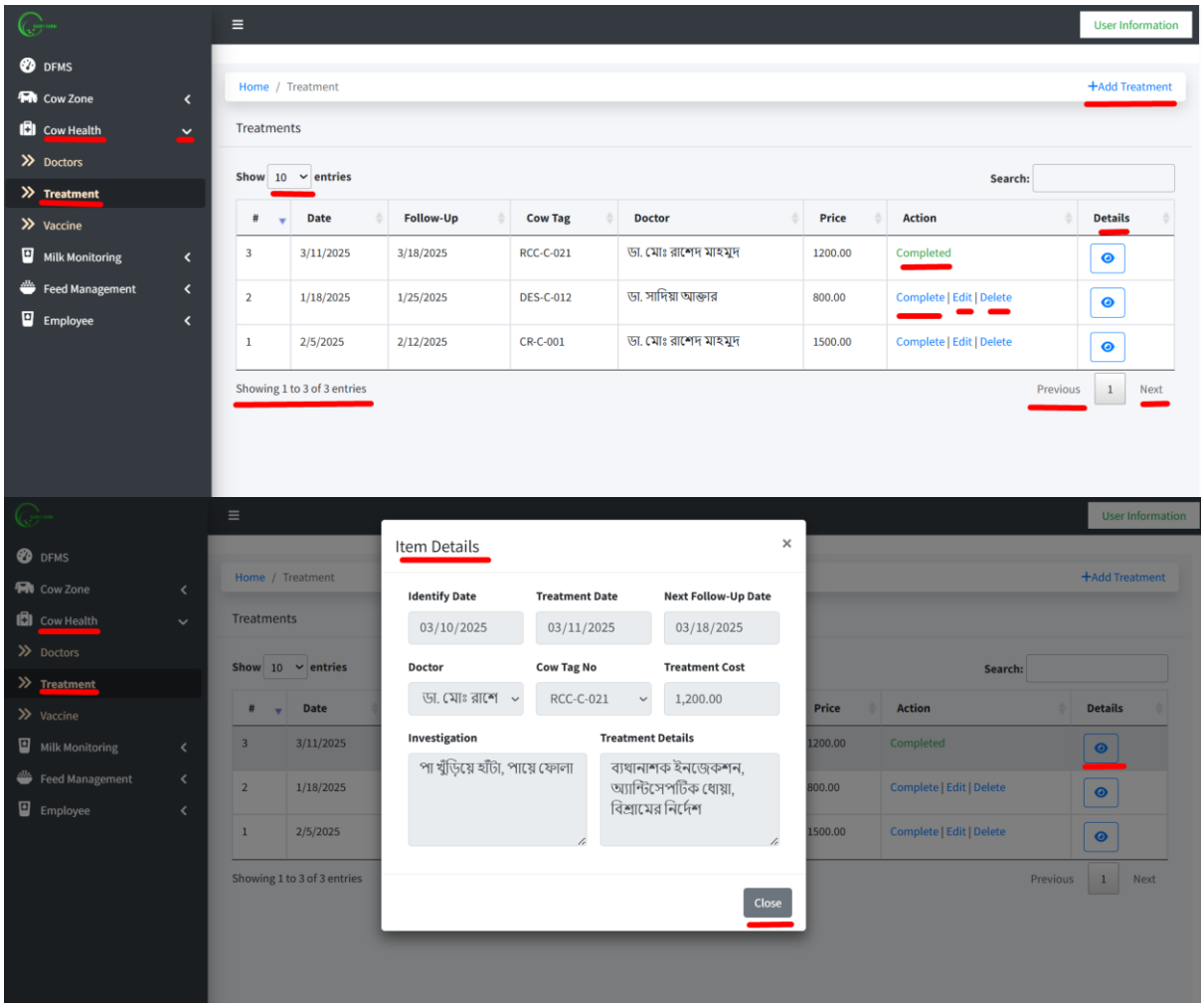
Cow Breeds



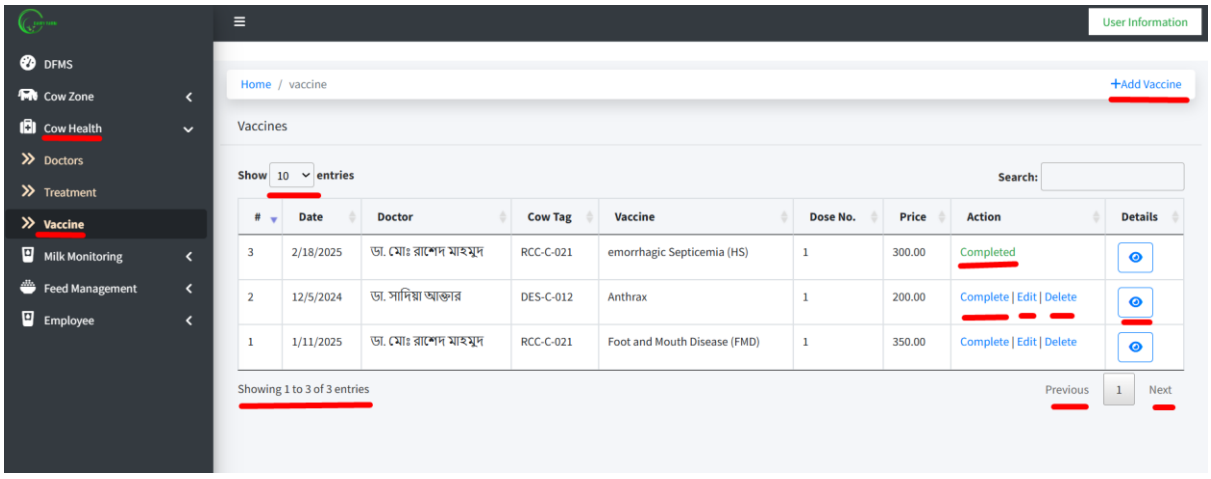
Cow Healath – Doctor

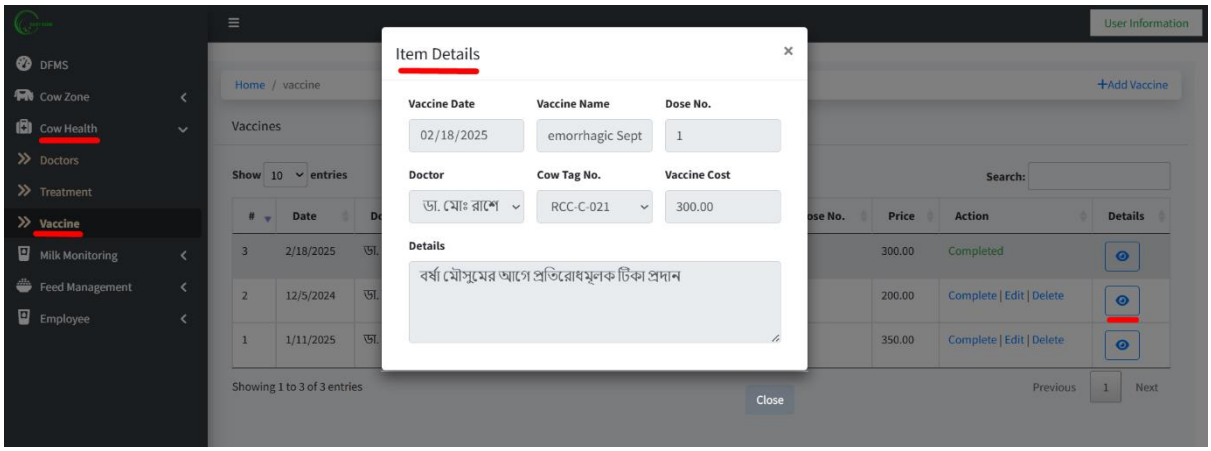


Cow Healath – Treatment

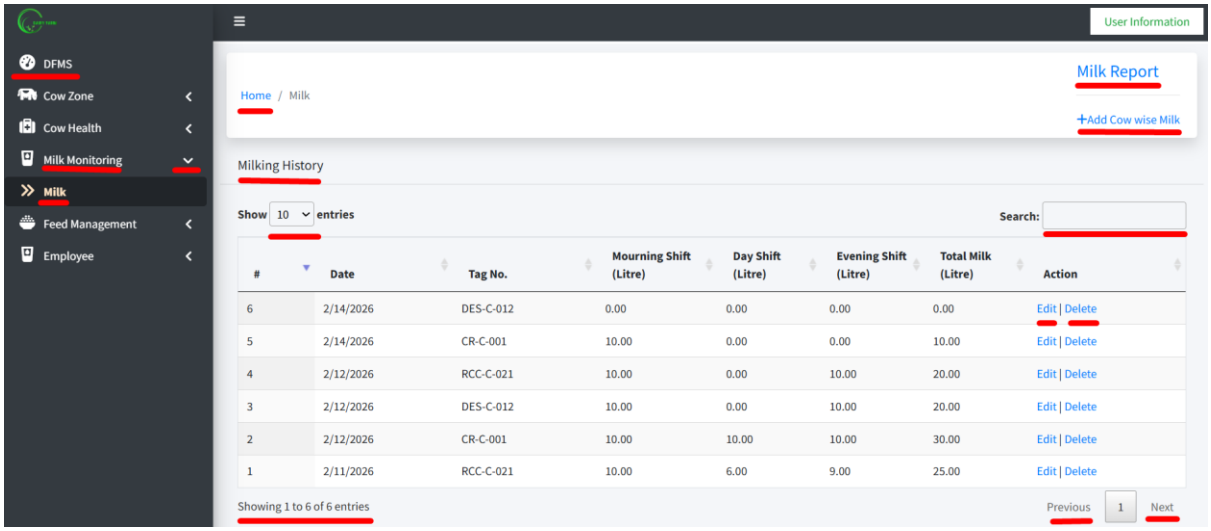


Cow Healath – Vaccine

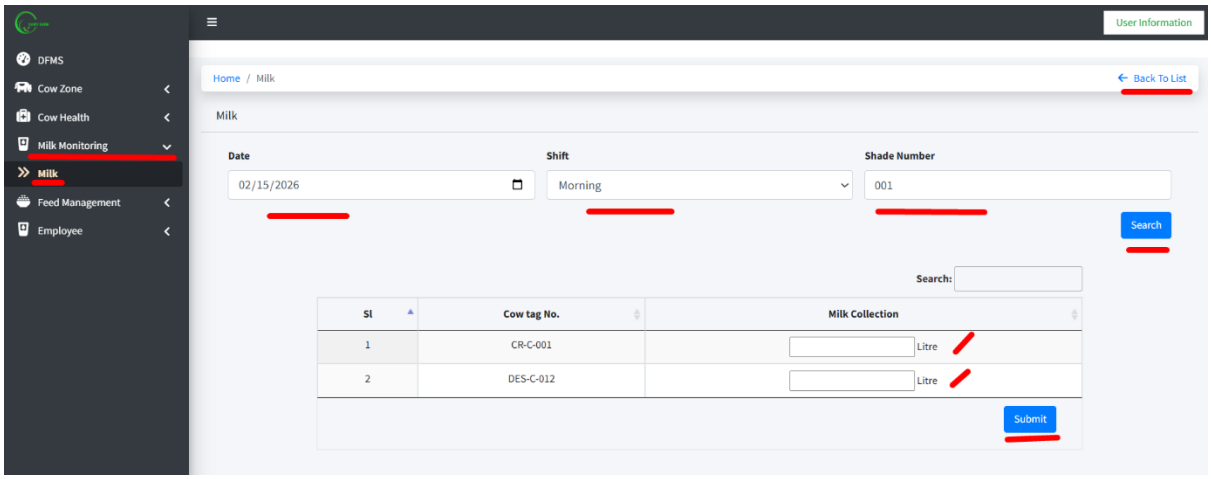




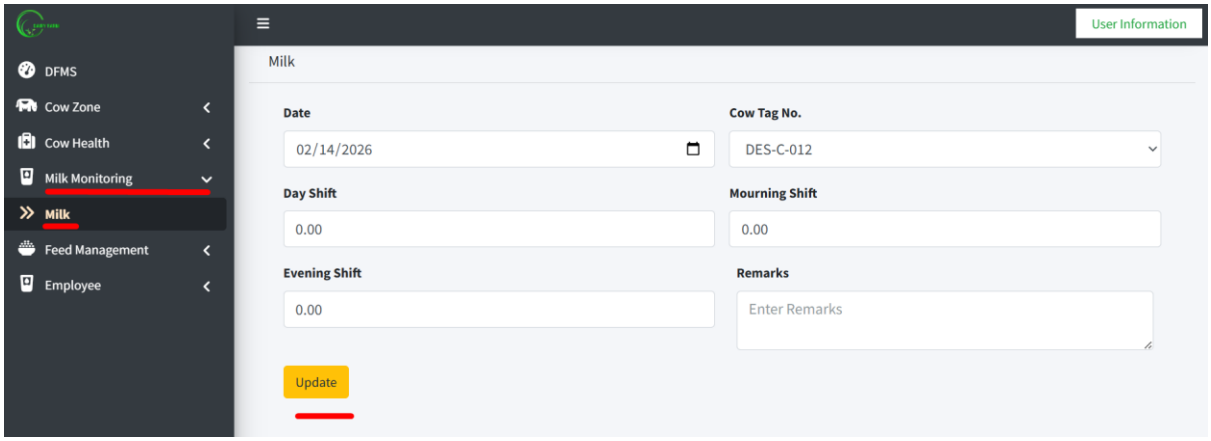
Milk Monitorring – Milk



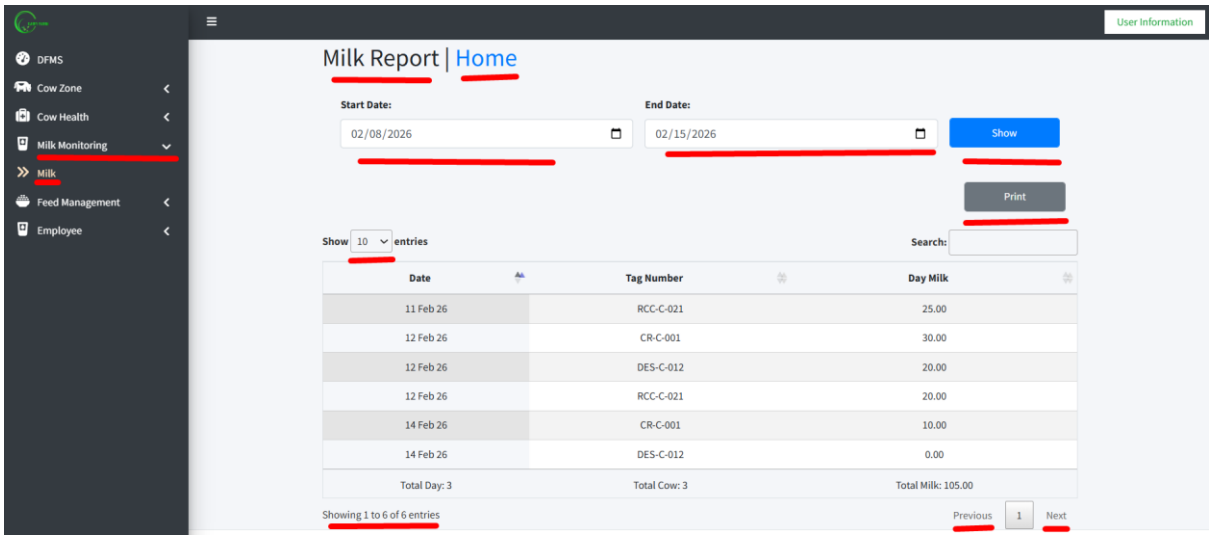
Milk Monitorring - Milk Add



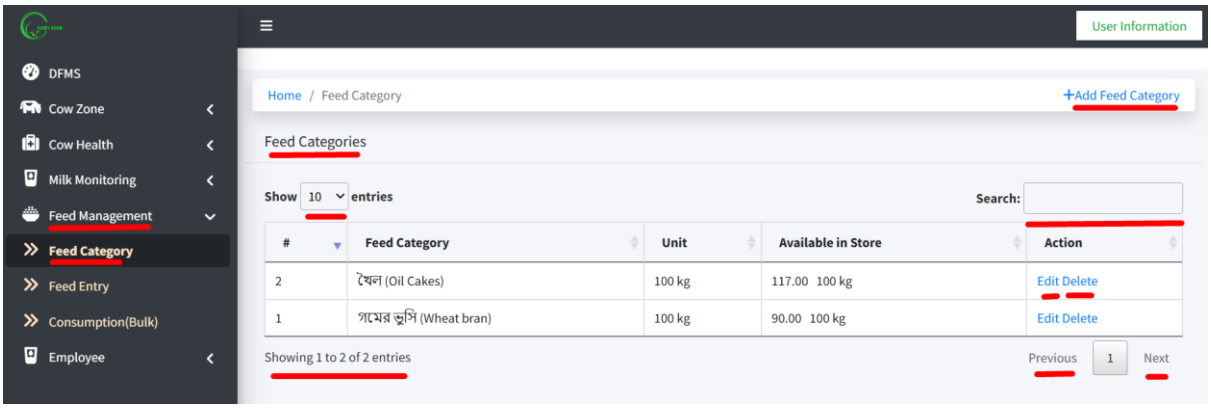
Milk Monitorring - Milk Update



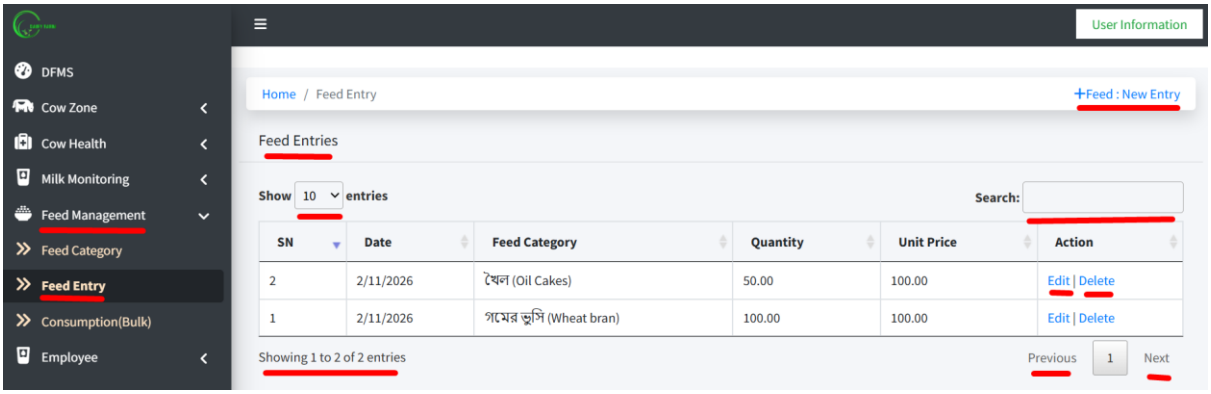
Milk Monitorring - Milk Report



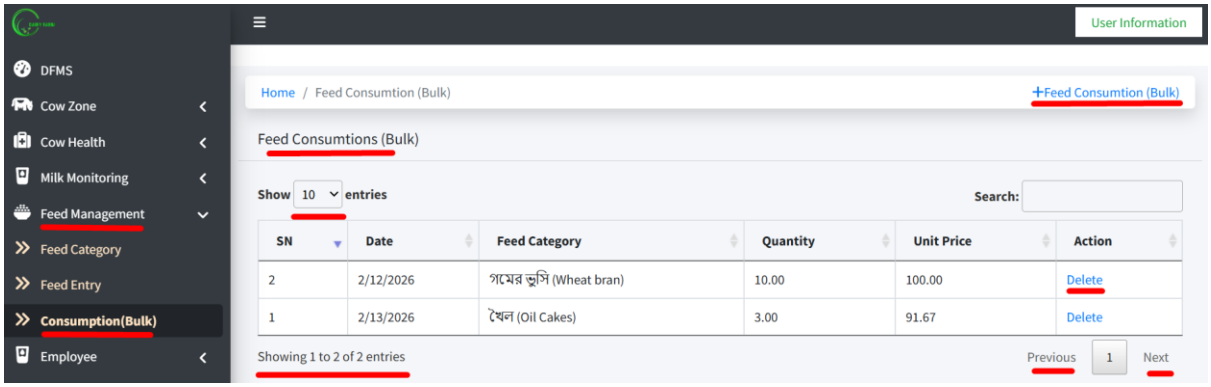
Feed Management - Feed Category



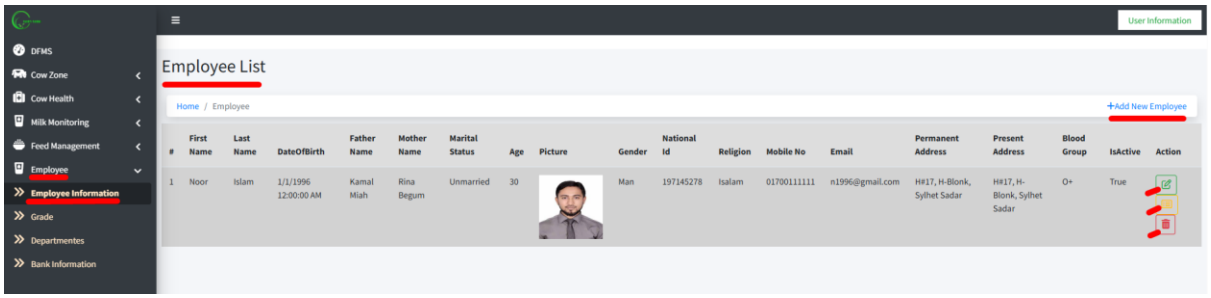
Feed Management - Feed Entry



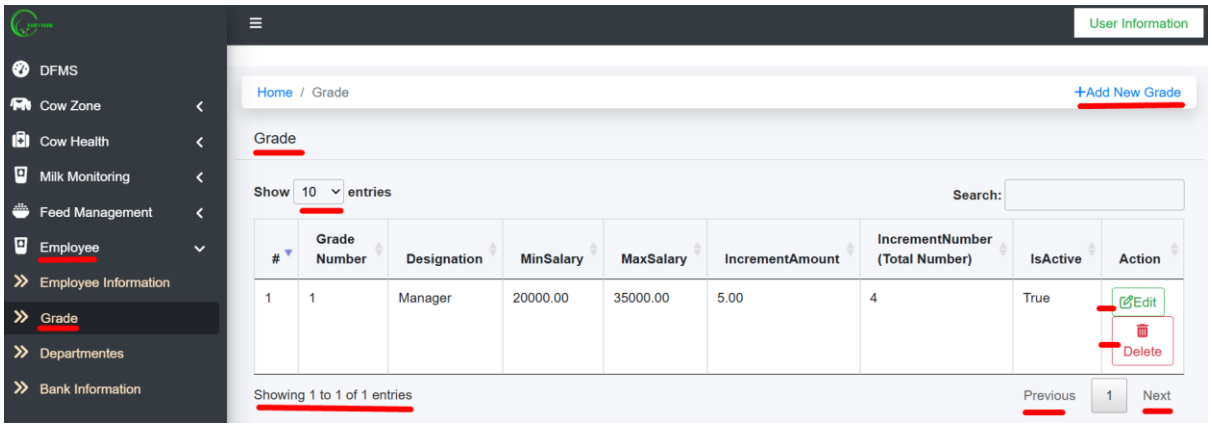
Feed Management - Consumption



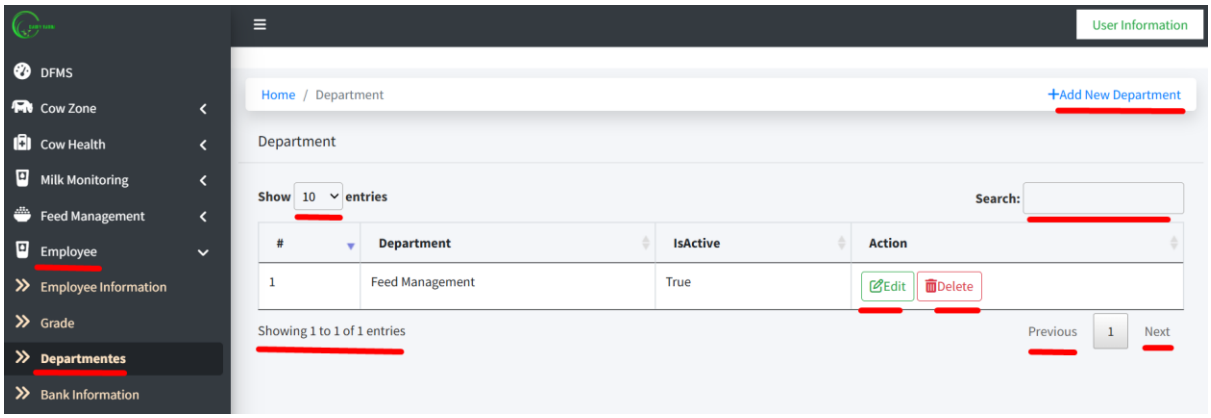
Employee - Information



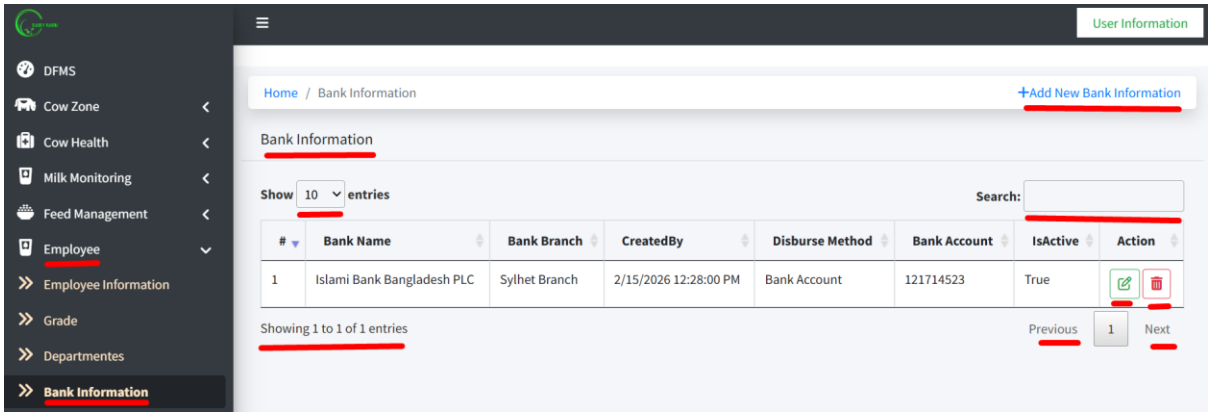
Employee – Grade



Employee - Department



Employee - Bank Ionformation



Chapter Seven

Testing

7.1 Overview

Testing is a critical stage in the software development life cycle (SDLC) that ensures the Dairy Firm Management System (DFMS) operates reliably, accurately, and securely according to defined requirements. The purpose of testing is to detect defects, validate functionalities, and verify that the system performs efficiently under various operational scenarios. This phase involves both functional and non-functional testing to confirm that the application meets user expectations and technical specifications.

Testing for DFMS was conducted in multiple phases, including unit testing, integration testing, system testing, and user acceptance testing. Each module — such as Cow Management, Health Tracking, Feed Inventory, Milk Monitoring, and Employee Management — underwent separate validation before being integrated into the final system. Both manual and automated tools (such as NUnit and xUnit) were employed to ensure reliability and repeatability. The testing process helped confirm data accuracy, security, and performance stability, ensuring a robust, production-ready platform.

Testing Highlights

- Validation of system functionality and performance.
- Multi-phase testing: unit, integration, and user acceptance.
- Both manual and automated test cases executed.
- Verification of data accuracy and workflow efficiency.
- Security, compatibility, and usability checks.
- Test-driven development (TDD) approach for reliability.
- Error logging and bug tracking during each iteration.
- Continuous testing integrated with Agile methodology.
- End-user validation before deployment.

7.2 Types of Testing

Several types of testing were implemented to ensure that DFMS meets quality and performance standards. Each testing type targets specific aspects of the system to identify potential issues and verify smooth operation.

1. **Black box testing**
2. **White box testing**
3. **Compatibility testing**
4. **Security testing**
5. **Unit Testing.**
6. **Integration Testing.**
7. **Functional testing**
8. **User Acceptance Testing (UAT)**

Each testing type contributed to building confidence in DFMS's readiness for deployment by guaranteeing smooth operation and user satisfaction.

Testing Types Overview

- Unit, integration, and system-level testing.
- UAT to confirm end-user satisfaction.
- Automated testing via NUnit/xUnit frameworks.
- Performance and stress testing for scalability.
- Regression testing after code updates.
- Validation of input/output accuracy.
- GUI testing for usability and responsiveness.
- Cross-module integration verification.
- Iterative testing aligned with Agile sprints.

7.3 Black Box Testing

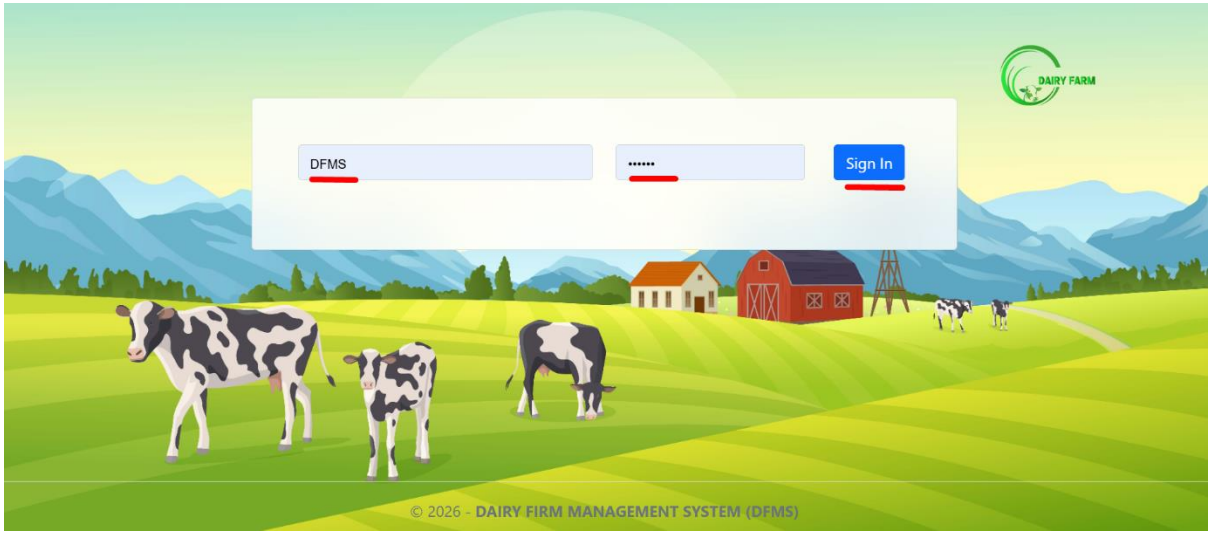
Black Box Testing focuses on examining the external behaviour of DFMS without considering internal code structures. The objective is to validate that inputs produce expected outputs across all modules. Test cases were designed based on system requirements and user scenarios, covering each function from login validation to data entry, update, and report generation.

This testing approach simulated real user interactions to detect functional errors or missing features. For example, the Cow Health module was tested by entering treatment details and verifying if the system correctly saved and displayed data. Similarly, reports were checked for accurate totals and consistency. The outcomes verified that DFMS performs as expected under realistic user operations, confirming data integrity and workflow accuracy.

Black Box Testing Summary

- Tests based on user inputs and expected outputs.
- Focus on overall system functionality.
- Validation of CRUD operations across modules.
- Verification of login, data updates, and reporting.
- No access to internal code logic.
- Identification of missing or incorrect functions.
- User-centric testing scenarios.
- Ensured compliance with SRS-defined requirements.
- Results validated system accuracy and reliability.

For example:



7.4 White Box Testing

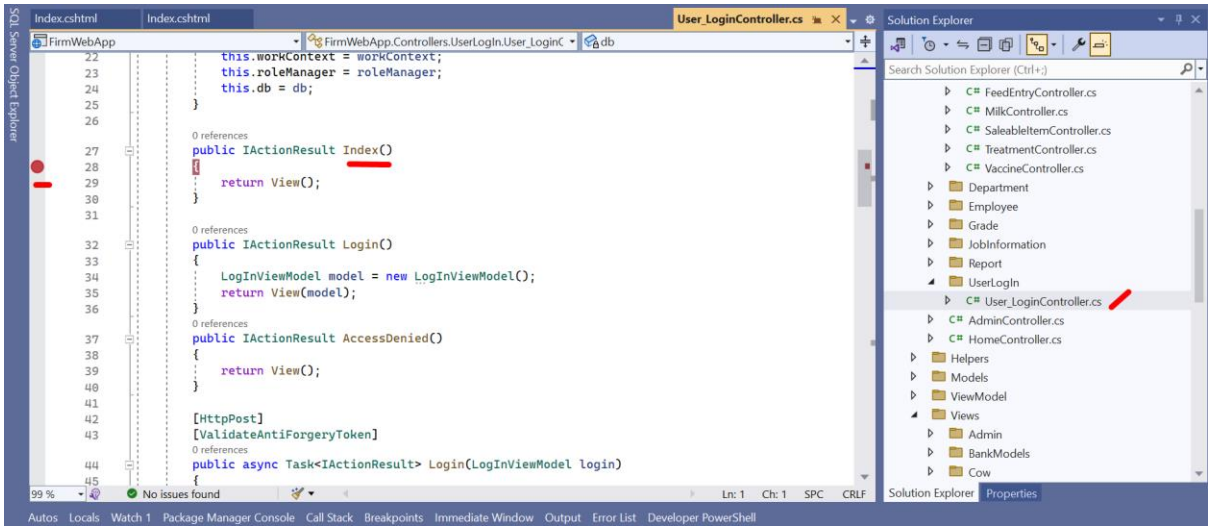
White Box Testing was conducted to evaluate the internal logic, data flow, and code structure of DFMS. The goal was to ensure that all paths, loops, and conditional statements within the code function correctly and efficiently. Developers examined the ASP.NET Core MVC controllers, services, and database interaction layers to identify vulnerabilities, dead code, or logical errors.

This testing also checked data validation routines, session management, and error handling mechanisms. Tools such as **NUnit** were used to automate and repeat tests for code-level reliability. White Box Testing confirmed that each component's internal workings aligned with expected outputs and performance benchmarks, ensuring the code was optimised, secure, and maintainable.

White Box Testing Key Points

- Internal logic and data flow validation.
- Code coverage analysis for completeness.
- Unit tests written using NUnit/xUnit frameworks.
- Path, loop, and conditional verification.
- Error-handling and exception tests.
- Database connection and query validation.
- Session and authentication tests.
- Optimised code for speed and efficiency.
- High-level maintainability confirmed through testing.

For example:



7.5 Compatibility Testing

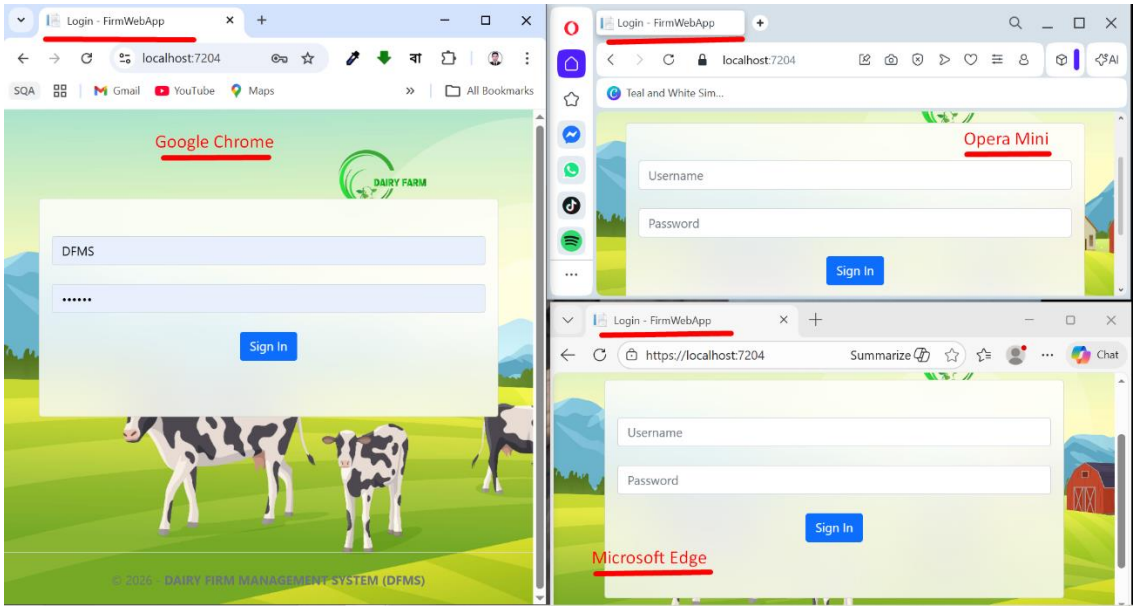
Compatibility Testing ensures that DFMS functions correctly across various environments, browsers, and devices. Since the system is web-based, this phase tested its performance on different web browsers such as Chrome, Edge, Firefox, and Safari, as well as across devices including desktops, laptops, and mobile phones.

The responsive design built using Bootstrap was verified for consistency in layout and performance across screen resolutions. Compatibility testing also evaluated database interactions and server responses under different operating systems (Windows, Linux). The results confirmed that the DFMS maintains uniform performance and user experience irrespective of the platform used, supporting accessibility and widespread usability.

Compatibility Testing Findings

- Cross-browser testing (Chrome, Edge, Firefox, Safari).
- Device testing across desktops, tablets, and mobiles.
- Validation of responsive UI layout.
- Consistent functionality under varied OS environments.
- Verified system responsiveness and load times.
- Database and backend connectivity maintained across servers.
- Browser cache and session management validated.
- No rendering errors or UI misalignment found.
- Ensured universal access and usability.

For example:



7.6 Security Testing

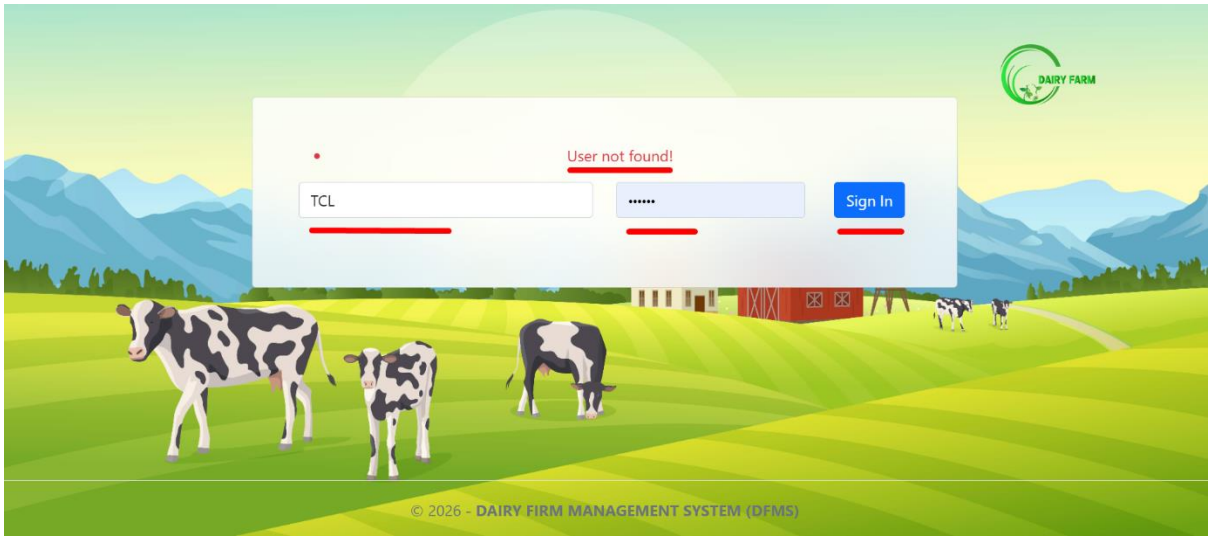
Security Testing was performed to ensure that DFMS protects user data and prevents □uthorizatio access. The system's authentication, □uthorization, and encryption mechanisms were rigorously tested to detect potential vulnerabilities. Passwords were stored using hashed encryption, and access privileges were restricted based on user roles (Admin and Employee).

Penetration testing simulated cyberattacks such as SQL injection, session hijacking, and brute-force login attempts to evaluate resistance. The system successfully prevented □uthorizatio access and maintained data confidentiality. Additionally, HTTPS protocols and secure session tokens were implemented to protect data transmission. Regular audit logs and activity tracking further enhanced accountability and transparency. The testing phase confirmed that DFMS adheres to standard cybersecurity practices, ensuring safe and reliable operation.

Security Testing Summary

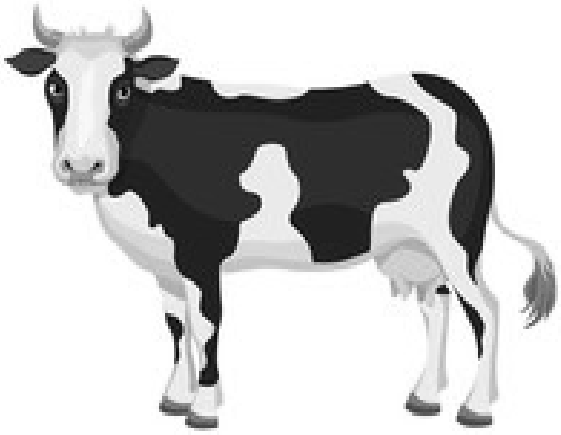
- Authentication and □uthorization validation.
- Password encryption and secure token generation.
- Protection against SQL injection attacks.
- HTTPS implementation for encrypted communication.
- Role-based access control tested and verified.
- User activity and login audits maintained.
- Overall compliance with data protection standards.

For example:



Chapter Eight

Conclusion



8.1 Overview

The Dairy Firm Management System (DFMS) represents a significant step toward the digital transformation of agricultural operations. Designed to address the challenges of manual data handling, the system successfully integrates various operational modules into one cohesive platform. Through DFMS, dairy administrators can efficiently manage livestock, monitor feed schedules, track milk production, and maintain employee records — all from a centralised dashboard.

The system delivers accuracy, speed, and transparency by leveraging a three-tier architecture built on ASP.NET Core MVC and SQL Server technologies. Its modular design, real-time reporting, and role-based security structure make it a robust, scalable, and practical solution for modern dairy firms. Ultimately, DFMS demonstrates how information technology can streamline traditional agricultural practices, reduce administrative burden, and promote sustainable productivity within the dairy sector.

Conclusion Highlights

- Complete automation of dairy-firm operations.
- Integration of livestock, feed, health, and employee modules.
- Real-time monitoring and reporting for informed decision-making.
- Reduction of manual workload and human error.
- Secure and centralised data management.
- Scalable and adaptable to future technologies.
- Promotes digital transformation in agriculture.
- Supports sustainability and operational transparency.
- Demonstrates successful application of IT in agri-management.

8.2 Limitations of the Project

While DFMS effectively meets its objectives, certain limitations were identified during development and testing. Firstly, the current system focuses primarily on desktop web access and has limited mobile application integration, which may restrict on-field data entry. Additionally, real-time IoT integration for livestock tracking and environmental monitoring has not yet been implemented, though the architecture supports future adoption.

Another limitation is dependency on a stable internet connection for remote access, which may pose challenges in rural areas with limited connectivity. Furthermore, while security features like encryption and authentication are in place, more advanced intrusion detection and multi-factor authentication systems could strengthen overall cybersecurity. These limitations provide valuable opportunities for enhancement in subsequent versions, ensuring that DFMS remains adaptive and relevant to evolving industry needs.

Project Limitations

- Limited mobile application support at this stage.
- IoT-based automation not yet integrated.
- Requires continuous internet connectivity for web operations.
- Dependency on third-party cloud hosting services.
- Limited AI-driven analytics for prediction and trend analysis.
- Advanced cybersecurity measures (e.g., MFA) pending.
- Local language support not yet implemented.
- High data volume may impact system performance over time.
- Need for broader hardware integration in future iterations.

8.3 Future Plan of the Project

The future scope of DFMS involves expanding its intelligence, automation, and connectivity capabilities to further optimise dairy farm management. Planned upgrades include integration with **IoT sensors** for real-time monitoring of animal health, temperature, and feed conditions. The addition of **AI-driven analytics** will enable predictive insights into milk production trends, disease prevention, and feed optimisation.

Development of a **dedicated mobile application** is also envisioned to enhance accessibility and allow field workers to input data instantly. Furthermore, incorporating **blockchain technology** could ensure transparency in milk supply chains, improving traceability and trust. The system will also include multilingual interfaces, enhanced reporting dashboards, and seamless integration with accounting or ERP systems. These planned advancements will position DFMS as a next-generation digital platform that continues to evolve with emerging agricultural technologies.

Future Development Plans

- Integration of IoT devices for real-time monitoring.
- AI-based analytics for yield prediction and herd health.
- Mobile app development for remote and field operations.
- Blockchain adoption for milk supply chain transparency.
- Multilingual user interface for wider accessibility.
- Integration with ERP and accounting systems.
- Enhanced dashboards and data visualisations.
- Advanced cybersecurity and cloud interoperability.
- Machine learning for predictive feed and health insights.
- Sustainable digital ecosystem for smart farming.

Progress Review Form

Student: Md. Shahidul Alam (ID: 2023822017, Batch-6th)
Program: Master of Information Technology
Project Title: Dairy Firm Management System (DFMS)
Institution: IICT, Shahjalal University of Science and Technology (SUST)

Supervisor Review Table

Progress/Work Reviewed	Supervisor's Comment	Supervisor's Signature & Date

Supervisor's Final Remarks:

_____

_____

Supervisor's Name & Signature: _____

Professor Dr. Mohammad Abdullah Al Mumin
Department of CSE & Director, IICT,
Shahjalal University of Science and Technology (SUST)