JSS MAHAVIDYAPEETHA

JSS COLLEGE OF ARTS, COMMERCE & SCIENCE

An Autonomous College, Affiliated with the University of Mysore
BN Road, Mysuru -570 025

Project Report On

"DAIRY SERVE MANAGEMENT SYSTEM"

Submitted in partial fulfilment of the requirements for obtaining the Degree of

Bachelor of Computer Application

By

Ravi Raghavendra G S

[U01BE21S0105]

UNDER THE GUIDANCE OF

Ms. Jayashree H J

Assistant Professor

Department of Computer Science

JSS College, Mysuru-570025

DEPARTMENT OF COMPUTER SCIENCE

JSS COLLEGE OF ARTS, COMMERCE & SCIENCE

An Autonomous College, Affiliated with the University of Mysore
BN Road, Mysuru-570 025

CERTIFICATE

This is to certify that the project work entitled "**Dairy Serve Management System**" carried out by **Ravi Raghavendra G S**, bearing Register Number **U01BE21S0105**, has been completed in the VI Semester as a part of the requirements for the award of the Bachelor of Computer Application degree by JSS College of Arts, Commerce, and Science (Autonomous), Mysuru, under the University of Mysore, during the academic year 2023-24. The project work has been reviewed and approved, meeting the academic standards and requirements prescribed for the final semester BCA program.

Signature of the Guide

Signature of HOD

Name of the Candidate: ....

Register No.:..

Examination Centre: ....

Date of Practical Examination: ....

DECLARATION

I, Ravi Raghavendra G S (U01BE21S0105), at this moment, declare that this project report, entitled "**Dairy Serve Management System**," is a record of the original work carried out by me under the guidance of **Ms. Jayashree H J**, Assistant Professor, Department of Computer Science, JSS College of Arts, Commerce and Science, Mysuru. To the best of my knowledge, this work has not been submitted for the award of any degree, diploma, associateship, fellowship, or similar title at any other university or institution.

Place: MYSURU

Date :

Signature

(Ravi Raghavendra G S)

U01BE21S0105

ACKNOWLEDGEMENT

I would like to extend my profound gratitude to our esteemed Chief Executive, **Prof. B.V. Sambha Shivaiah**, and Principal, **Prof. Prabhu,** whose invaluable encouragement and unwavering support have been pivotal throughout my academic journey and the completion of this project.

I proudly acknowledge the consistent support and motivation provided by Mr. Vidyashankar, Head of the Department of Computer Science. His leadership and dedication to academic excellence have been a source of inspiration.

I am deeply grateful to my guide, **Ms. Jayashree H.J.,** Assistant Professor in the Department of Computer Science. Her insightful guidance, constructive suggestions, and constant encouragement have been instrumental in the successful completion of my project. Her expertise and patience have significantly contributed to my understanding and development.

I would also like to express my heartfelt thanks to all the teaching and non-teaching staff of the Department of Computer Science, who have, directly or indirectly, extended their timely help and support. Their assistance and cooperation have greatly facilitated the smooth progression of my project.

A special note of appreciation goes to my classmates and friends for their encouragement, support, and valuable feedback during this project. Their camaraderie and collaborative spirit have been invaluable.

Finally, I sincerely salute my parents with deep reverence. Their boundless love, affection, and moral support have been the bedrock of my life, providing me with the strength and determination to pursue my academic goals.

Thank you all for your unwavering support and encouragement.

Ravi Raghavendra G S
(U01BE21S0105)

ABSTRACT

The Dairy Serve Management System aims to transition from a manual to an automated system for managing dairy business operations. In the current manual system, administrators face the challenge of tracking various activities such as milk production, sales, and customer management, which can lead to inefficiencies and errors. Manually Maintaining records for milk collection, distribution, and billing is time-consuming and prone to inaccuracies.

The purpose of this project is to develop a comprehensive Dairy Serve Management System that streamlines these operations. The system will include features to record milk production, manage customer details, track sales, and automate billing processes. By automating these tasks, the system will enhance accuracy, reduce the possibility of errors, and improve overall efficiency.

This project intends to automate the essential aspects of dairy management without altering the fundamental procedures currently in place. The system will be user-friendly, allowing dairy business owners to easily access and manage records, thus facilitating better decision-making and operational control.

Content

Table of Content

1. INTRODUCTION

1.1 AIM OF THE PROJECT

The primary aim of the Dairy Serve Management System is to create a robust and efficient platform that addresses the diverse operational needs of a dairy business. This project aims to streamline the management of farmer, staff, and dairy animal details, ensure accurate daily data entry, and provide comprehensive billing and product management functionalities. By leveraging modern web technologies, the system seeks to reduce manual workloads, minimize errors, and enhance productivity. The ultimate goal is to provide a user-friendly interface that facilitates seamless interactions among stakeholders, including staff, farmers, and buyers, thereby improving operational efficiency and customer satisfaction.

1.2 SCOPE AND OBJECTIVES OF PROJECT

The scope of the Dairy Serve Management System encompasses various aspects of dairy business operations, from animal care and data entry to product sales and billing. The project is designed to be scalable, accommodating both small dairy farms and larger enterprises with more complex needs. Key objectives include:

- **Efficient Data Management:** To provide a structured way to manage farmer, staff, and dairy animal details, ensuring all data is accurate and easily accessible.
- **Streamlined Daily Operations:** To facilitate timely and precise daily data entry for dairy operations, including milk production and animal health checks.
- **Comprehensive Billing System:** To offer a robust bill management module that can generate, view, edit, and delete bills, ensuring accurate financial tracking and customer billing.
- **Product Management:** To manage dairy product inventory, allowing for the addition, editing, viewing, and deletion of products, and ensuring that buyers can place orders seamlessly.
- **Secure Access Control:** To implement role-based access control, ensuring that different user types (admin, staff, farmer, buyer) have appropriate access to functionalities relevant to their roles.
- **User-Friendly Interface:** To provide an intuitive and responsive interface that enhances user experience across various devices.

1.3 INTRODUCTION

The Dairy Serve Management System is a comprehensive, web-based solution designed to meet the diverse needs of dairy businesses. Built using HTML, CSS, Bootstrap, and JavaScript for the front end, and PHP and MySQL (WAMP) for the backend, the system aims to streamline and automate various dairy operations. Traditional dairy management methods often involve cumbersome manual processes that are prone to errors and inefficiencies. In contrast, this system leverages modern web technologies to provide a centralized platform that improves data accuracy, operational efficiency, and user convenience.

The system is organized into several key modules, each focusing on a specific aspect of dairy management. These modules include Farmer Details, Staff Details, Dairy Animal Details, Daily Data Entry, Bill Management, and Dairy Products. Each module is designed to interact seamlessly with the central database, ensuring consistent and reliable data management. The role-based access control mechanism ensures that users have access to functionalities pertinent to their roles, enhancing both security and usability. For instance, while admins have full access to all modules, farmers can only view and edit their own details and data entries.

Moreover, the Dairy Serve Management System is designed with scalability and future enhancements in mind. It can easily accommodate additional functionalities such as mobile app integration, advanced analytics, and IoT devices for real-time monitoring of dairy operations. By providing a comprehensive and flexible solution, the system aims to support dairy businesses in achieving greater efficiency, better resource management, and enhanced customer satisfaction. This project represents a significant step forward in the digital transformation of dairy management, leveraging technology to overcome the challenges of traditional methods and pave the way for a more streamlined and effective operational framework.

2. LITERATURE SURVEY

In the field of dairy management, several traditional systems and methodologies have been explored and implemented over the years. Historically, dairy farms have relied heavily on manual processes for record-keeping and management of daily operations. These methods, though functional, are prone to human errors, inefficiencies, and a lack of real-time data accessibility. The shift towards digital solutions began with the introduction of standalone software applications designed to handle specific aspects of dairy management, such as inventory tracking or billing. However, these early systems often lacked integration capabilities, resulting in fragmented data and limited overall efficiency.

Recent advancements in information technology have spurred the development of more sophisticated dairy management systems. These modern solutions leverage web-based platforms to offer enhanced accessibility and real-time data updates. Research by Fernandez and colleagues (2018) highlights the benefits of web-based dairy management systems, noting that they provide a centralized database that ensures consistency and reliability of data. These systems enable seamless communication between different stakeholders, such as farmers, staff, and buyers, thereby improving operational transparency and accountability. Moreover, web-based systems support remote access, allowing users to manage and monitor dairy operations from any location, which is particularly beneficial for larger dairy enterprises with multiple sites.

The integration of role-based access control (RBAC) mechanisms in dairy management systems has been a significant focus in recent literature. Studies by Smith et al. (2020) emphasize the importance of RBAC in ensuring data security and integrity within dairy management systems. By assigning specific privileges to different user roles (such as admin, staff, farmer, and buyer), these systems can effectively restrict access to sensitive information and functions based on the user's responsibilities and needs. This approach not only enhances security but also simplifies the user experience by providing access to only relevant features and data. For example, while admins have full access to all modules, farmers can only view and edit their own details and daily entries.

Another critical aspect covered in the literature is the use of automated data entry and real-time monitoring technologies. IoT (Internet of Things) devices and sensors are increasingly being integrated into dairy management systems to automate data collection processes. Research by Johnson and Wang (2019) demonstrates how IoT devices can monitor milk production, animal health, and environmental conditions in real-time, sending data directly to the central system. This integration not only reduces the manual effort required for data entry but also ensures higher accuracy and timeliness of data. The real-time monitoring capabilities allow for proactive management, where potential issues can be identified and addressed promptly, thereby improving overall herd health and productivity.

Modern dairy management systems also benefit from advancements in web development technologies such as PHP, MySQL, HTML, CSS, and JavaScript. PHP and MySQL are widely used for server-side scripting and database management, enabling robust back-end development for web-based systems. HTML and CSS are fundamental technologies for structuring and styling web pages, ensuring a user-friendly interface. JavaScript enhances interactivity and responsiveness, allowing for dynamic content updates without requiring a page reload. These technologies collectively support the creation of comprehensive, efficient, and scalable dairy management systems.

Overall, the literature underscores a clear trend toward the adoption of comprehensive, web-based dairy management systems that integrate various advanced technologies. These systems aim to address the limitations of traditional and standalone software solutions by offering centralized, real-time, and secure data management capabilities. The Dairy Serve Management System aligns well with these trends, as it incorporates a modular design, role-based access control, and potential for IoT integration, positioning it as a modern solution capable of significantly improving the efficiency and effectiveness of dairy operations.

3. SYSTEM STUDY

3.1 EXISTING SYSTEM

In the existing system, a considerable amount of manual work is required, leading to a high degree of error and inefficiency. The process involves extensive paperwork for recording milk production, managing customer details, and handling sales, which is time-consuming and prone to mistakes. Administrators must manually track milk collection, sales, and billing, which can result in inaccuracies and financial discrepancies. Furthermore, maintaining physical records and files for these documents is cumbersome and challenging to manage.

Also, a lot of files, and records are required to store these documents thus making it difficult to maintain.

Since all these reports are prepared manually, we require computer software for all these purposes.

3.1.1 DISADVANTAGES OF THE EXISTING SYSTEM

Identifying the drawbacks of the existing system highlights the need for a more efficient solution. The disadvantages of manual file system maintenance are as follows:

- Limitations in report generation
- Excessive time consumption for manual recording
- High chances of errors
- Difficulty in resource allocation
- Time-consuming retrieval of information regarding customers
- Lack of an automated billing system and manual calculation of daily income
- Repeated recording of frequent customer details

3.2 PROPOSED SYSTEM

The Dairy Serve Management System we propose aims to automate the existing manual processes without altering the core procedures. The new system will consist of multiple modules tailored for different operational needs. The server module will be installed on the main computer, while the client modules will be used by employees for daily operations. Administrators will log in with their credentials to access the system's features.

The proposed system will streamline various tasks, including recording milk production, managing customer details, tracking sales, and automating billing processes. If any client module encounters issues, it can be temporarily removed from the allocation process and reintroduced once maintenance is complete.

The system will generate different reports related to production, sales, customer details, and daily operations. This will help in identifying any discrepancies and ensuring accurate tracking of business activities. Specific reports can be generated for individual customers or different operational parameters, enhancing transparency and control.

3.21 ADVANTAGES OF PROPOSED SYSTEM

The proposed system offers numerous benefits aimed at simplifying and enhancing dairy business management:

- Elimination of human errors
- Reduced manual labor and strain
- High Security
- Minimization of data redundancy
- Consistent data management
- User-friendly interface
- Easy data updating and maintenance
- Efficient record keeping
- Simplified generation of backup data

By implementing the Dairy Serve Management System, dairy businesses can achieve higher efficiency, accuracy, and better overall management of their operations.

3.3 Overview of Technologies & Languages Used

The Dairy Management System project employs a range of modern web technologies to create a robust, interactive, and responsive application. **PHP (Hypertext Preprocessor)** is a widely used open-source scripting language particularly suited for web development. It runs on the server side, enabling the generation of dynamic web page content that can interact seamlessly with databases. PHP is essential for dynamic content creation, database interaction, form handling, and session management, providing a strong backbone for the web application.

MySQL is the chosen relational database management system (RDBMS) for this project. Known for its reliability, scalability, and ease of use, MySQL efficiently handles data storage and retrieval. It supports complex queries, and transactions, and adheres to ACID (Atomicity, Consistency, Isolation, Durability) properties, ensuring data integrity and consistency. This makes it an excellent choice for managing the structured data required by the Dairy Management System.

On the client side, **HTML (Hypertext Markup Language)** is utilized to structure the web pages, forming the skeleton of the application. HTML allows for the creation of headings, paragraphs, lists, links, images, and forms, essential for content structuring and user input collection. Complementing HTML, and **CSS (Cascading Style Sheets)** controls the presentation of the web pages, defining layout, colors, fonts, and overall appearance. CSS also enables responsive design through media queries, ensuring the application is accessible and visually appealing across various devices and screen sizes. Additionally, CSS animations and transitions enhance the user experience by adding interactive elements.

JavaScript plays a crucial role in bringing interactivity to the web pages. As a high-level, client-side scripting language, JavaScript allows for dynamic content updates, form validation, and the creation of interactive elements like sliders and pop-ups. JavaScript's ability to handle user events and perform asynchronous data loading via AJAX (Asynchronous JavaScript and XML) ensures a smooth and responsive user experience without the need to reload the entire page.

To streamline and enhance the development process, the project leverages **Bootstrap**, a popular open-source CSS framework. Bootstrap provides a comprehensive set of pre-designed components and utilities for typography, forms, buttons, navigation, and other interface elements. Its responsive grid system and utilities ensure the web application adapts well to different screen sizes and devices, offering a mobile-first design approach. Bootstrap's customizable nature allows developers to override default styles to achieve unique designs, and its cross-browser compatibility ensures a consistent user experience across various browsers.

Additionally, the project uses **WAMP (Windows, Apache, MySQL, PHP)**, a powerful platform for developing web applications on Windows. WAMP provides an integrated environment with Apache as the web server, MySQL as the database server, and PHP as the server-side scripting language. This setup simplifies the development process by offering a local server environment for building and testing the application before deployment. WAMP's user-friendly interface allows for easy management of the Apache and MySQL services, facilitating seamless development and debugging.

Git is a distributed version control system used to track changes in the source code during software development. It allows multiple developers to work on the same project simultaneously without interfering with each other's work. Git's branching and merging capabilities make it easy to experiment with new features and fix bugs without affecting the main codebase. Version history maintained by Git ensures that any previous state of the project can be restored if needed, making it an indispensable tool for collaborative and agile development.

GitHub is a web-based platform that hosts Git repositories and provides version control and collaboration features. It facilitates project management, code review, and continuous integration/continuous deployment (CI/CD) processes. GitHub allows developers to share their repositories, collaborate with others, and contribute to open-source projects. Features such as pull requests, issues, and project boards streamline the workflow and help in managing and tracking project progress. GitHub's integration with other development tools and services further enhances productivity and efficiency.

Visual Studio Code (VS Code) is a powerful and versatile code editor developed by Microsoft. It supports a wide range of programming languages and frameworks through extensions, making it suitable for various development tasks. VS Code offers features like IntelliSense for code completion, syntax highlighting, debugging, and Git integration. Its customizable interface and vast library of extensions allow developers to tailor the editor to their specific needs. VS Code's built-in terminal and seamless integration with version control systems like Git make it a preferred choice for many developers.

4. SOFTWARE REQUIREMENTS AND SPECIFICATION

4.1 PURPOSE

The purpose of the Dairy Serve Management System project is to automate the activities involved in managing a dairy business. The software must include provisions for recording milk production, managing customer details, and tracking sales. It should help dairy owners retrieve customer details and sales information as needed. The system should also automate the billing process and facilitate the calculation of daily production and income.

4.2 SCOPE

This document serves as the software requirement specification for the Dairy Serve Management System, which encompasses various aspects of dairy business operations. The system will handle tasks such as collecting information on milk production, managing customer records, tracking sales, and automating billing processes.

4.3 SYSTEM ANALYSIS

This project addresses the challenges of managing a dairy business manually and aims to overcome the problems associated with such manual processes. By identifying the drawbacks of the existing system, we can design a computerized system that is compatible with the current procedures while providing a user-friendly interface. The new system will enhance efficiency and mitigate the issues present in the manual system.

4.3.1 DRAWBACKS OF EXISTING SYSTEM

- Manual recording of milk production and customer details is time-consuming and error-prone.

- Administrators must manually track sales and billing, which can lead to inaccuracies.

- Physical storage of records and documents is cumbersome and difficult to maintain.

- Reports are prepared manually, increasing the likelihood of errors and inconsistencies.

- Possible human errors and overhead of manual labor for file maintenance.

- Data is unsecured and can be redundant and inconsistent.

- Modifying details and maintaining transaction records is challenging.

4.3.2 ANALYSIS OF PROPOSED SYSTEM

The Dairy Serve Management System we propose is designed to automate the current manual processes without altering the core procedures. The new system will include multiple modules, with the server module installed on the main computer and client modules used by employees. Administrators will log in with their credentials to access the system's features.

The system will streamline various tasks, including recording milk production, managing customer details, tracking sales, and automating billing processes. If any client module encounters issues, it can be temporarily removed from the allocation process and reintroduced once maintenance is complete.

Different reports will be generated, providing detailed information about production, sales, and customer activities. This will help identify discrepancies and ensure accurate tracking of business operations. Specific reports can be generated for individual customers or various operational parameters, enhancing transparency and control.

Advantages of the Proposed System

- Elimination of human errors
- Reduction of manual labor and strain
- High security for data
- Minimization of data redundancy
- Consistent data management
- User-friendly interface
- Easy data updating and maintenance
- Efficient record keeping.
- Simplified generation of backup data.
- Periodic and smooth updates of transaction details.

4.3.3 Feasibility Study

The feasibility of the project is analyzed in this phase, and a business proposal is put forth with a general plan for the project and some cost estimates. During system analysis, the feasibility study of the proposed system is conducted to ensure that it is not a burden to the company. The major requirements for the system must be understood for feasibility analysis, which involves three key considerations:

4.3.3.1 Operational Feasibility

Operational feasibility is the ability to utilize, support, and perform the necessary tasks of a system or program. It includes everyone who creates, operates, or uses the system. To be operationally feasible, the system must fulfill a need required by the business.

4.3.3.2 Economical Feasibility

This study checks the economic impact that the system will have on the organization. The amount of funds that the company can allocate for the research and development of the system is limited, and expenditures must be justified. The developed system should be within budget, achieved by using freely available technologies and only purchasing customized products as needed.

4.3.3.3 Motivational Feasibility

Motivation is crucial in software productivity and software failure. Previous work suggests that conventional approaches to motivation, based on reward and recognition, are not appropriate for software engineering. The technical context of software engineers' work, such as the technical challenges and development tools used, is important for their motivation.

4.3.3.4 Schedule Feasibility

Schedule feasibility is the probability of a project being completed within its scheduled time limits, by a planned due date. If a project has a high probability of being completed on time, then its schedule feasibility is high. If the work to be accomplished does not fit the demanded timeframes, then the schedule is unfeasible.

4.4 System Definition

This system involves creating and managing a Dairy Serve Management System for managing and manipulating the database by developing a friendly user interface and providing an authentication mechanism to safely accomplish the system's activities.

4.4.1 System Goals

The goal of the system is to accommodate the major functionalities of a dairy management system operational environment with the following:

- Authentication and Authorization

- Data Management

4.5 General Requirements

A Software Requirements Specification (SRS) is a complete description of the behavior of the system to be developed. It includes a set of general requirements that form use cases, describing all interactions that users will have with the software. Use cases are also known as functional requirements. In addition to use cases, the SRS contains non-functional (or supplementary) requirements, which impose constraints on design or implementation, such as performance engineering requirements, quality standards, or design constraints.

4.6 Functional Requirements

Functional requirements specify the output file to be produced from the given file, describing the relationship between the input and output of the system. For each functional requirement, a detailed description of all data inputs, their sources, and the range of valid inputs must be specified.

MODULES

1. **Administrator (Owner)**

- Login: A user ID and password are provided to the Administrator and users to provide access control. The Administrator can make modifications as needed.

- Tariff Settings: The Administrator can create, edit, and delete user tariffs.

- Dairy Animal Details: The Administrator manages details about dairy animals.

- Billing: The Administrator controls the billing process by users.

- Reports: The Administrator views usage reports for specific users, entire usage, usage by specific machines, and daily reports.

2. **User (Manager)**

- Login: A user ID and password are provided for access control, but users cannot modify the software's functionality.

- Daily Data Entry: Users record daily transactions, including milk production and sales, by logging into the system.

- Farmer Details: Users manage farmer details and transactions.

- Staff Details: Users manage staff information and assignments.

4.7 Non-Functional Requirement

Non-functional requirements for the project include:

- **Maintenance**: The project should have minimal maintenance requirements, with algorithms properly implemented considering user deployment.

- **Integrity**: The code should have well-defined integrity, ensuring secure and modular design.

4.8 External Interface Requirements

External interface requirements specify the hardware, software, or database elements with which the system or component must interface to ensure proper communication.

4.9 Hardware Requirements

- **Processo**r: Intel Core i3 3rd generation 2.4 GHz or higher

- **Hard Disk**: 200 GB or more

- **RAM**: 4 GB onwards

4.10 Software Requirements

- **Operating System**: Windows 7 or later

- **Coding Language**: HTML, CSS, Bootstrap (Frontend); PHP, MySQL (Backend)

- **Database**: MySQL 5.7

- **IDE**: Visual Studio Code

- **Version Control:** Git (GitHub)

5. SYSTEM DESIGN

5.1 INTRODUCTION

System design is the process that involves conceiving, planning, and carrying out the plan by generating the necessary reports and inputs. In other words, the design phase acts as a bridge between the software requirement specification and the implementation phase, satisfying those requirements. System design is the transformation of the analysis model into a system design model.

The design of a system is correct if it is built precisely according to the requirements. The design should be verifiable, complete, and traceable. The goal is to divide the problem into manageable small modules that can be solved separately. These different modules have to cooperate and communicate together to solve the overall problem. The complete project is broken down into different identifiable modules. Each module can be understood separately, and all the modules are combined to get the solution of the complete system.

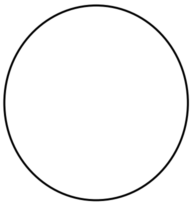
5.2 DATAFLOW DIAGRAM

A data flow diagram (DFD) is a graphical representation of the "flow" of data through an information system, modeling its process aspects. A DFD is often used as a preliminary step to create an overview of the system without going into great detail, which can later be elaborated. DFDs can also be used for the visualization of data processing (structured design).

A DFD shows what kind of information will be input to and output from the system, how the data will advance through the system, and where the data will be stored. It does not show information about the timing of the process or information about whether processes will operate in sequence or parallel, unlike a flowchart which also shows this information.

BASIC DFD SYMBOLS

Function Symbol:

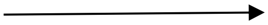


A function is represented using a circle. This symbol is called a process or a bubble. Bubbles are annotated with the names of corresponding functions.

External Entity Symbol:

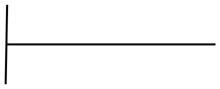
An external entity such as a user, project manager, etc. is represented by a rectangle. The external entities are essentially those physical entities external to the application system, which interact with the system by inputting data to the system or by consuming the data produced by the system. In addition to the human users, the external entity symbols can be used to represent external hardware and software such as application software.

Data Flow Symbol:

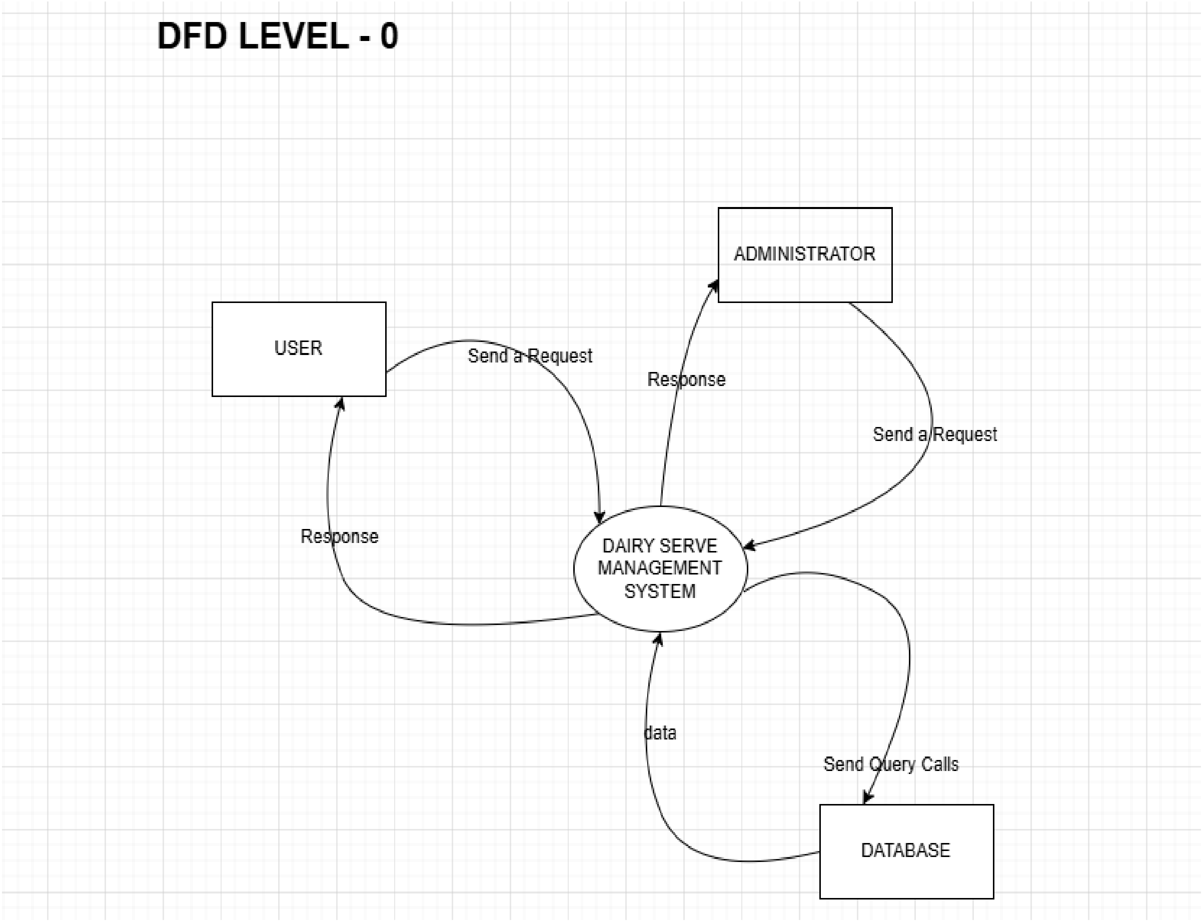


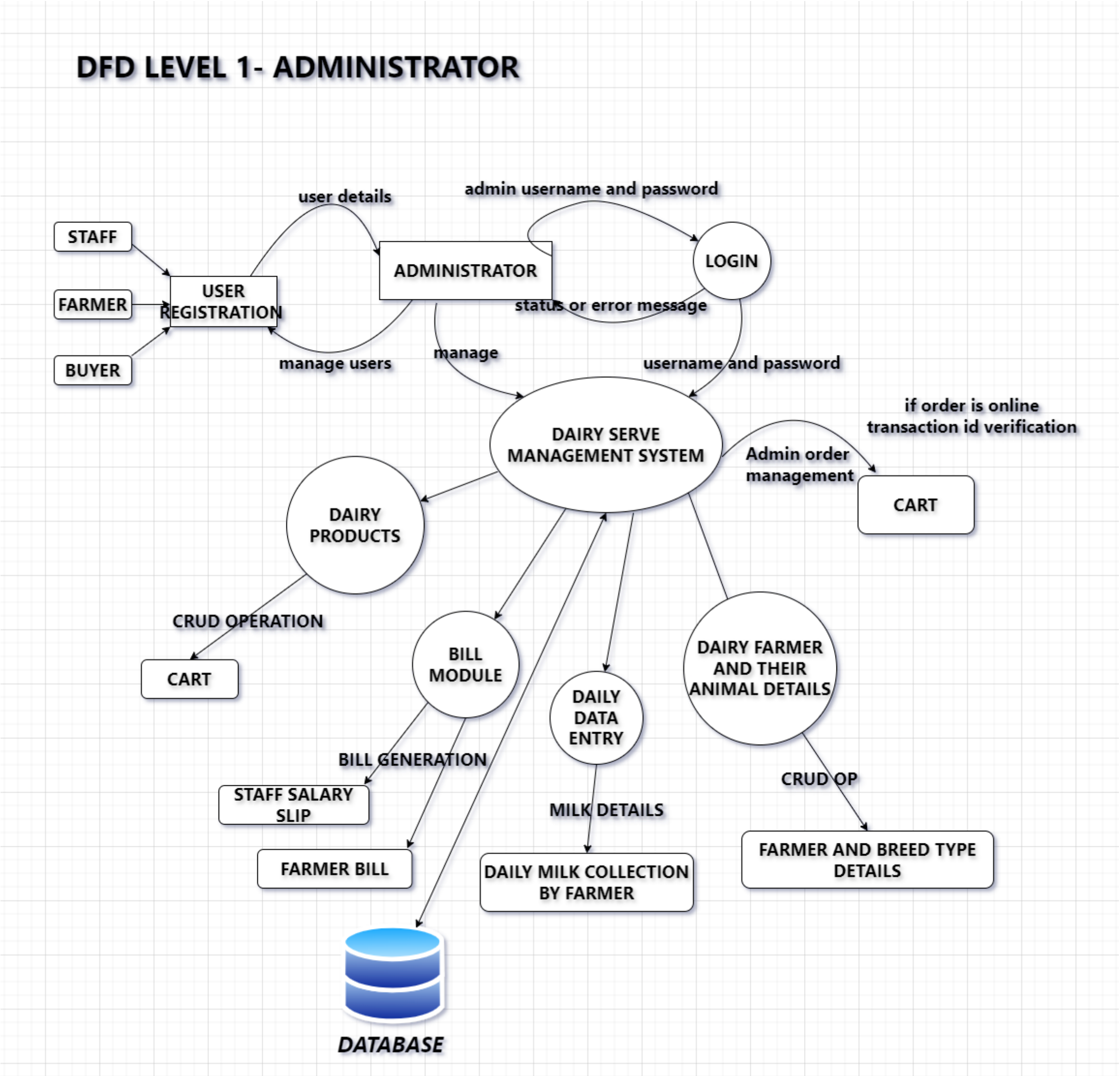
A directed arc or an arrow is used as a Data Flow Symbol. This represents the data flow occurring between two processes or between an external entity and a process; in the direction of the Data Flow Arrow. Data flow Symbols are annotated with corresponding data names.

Data Store Symbol



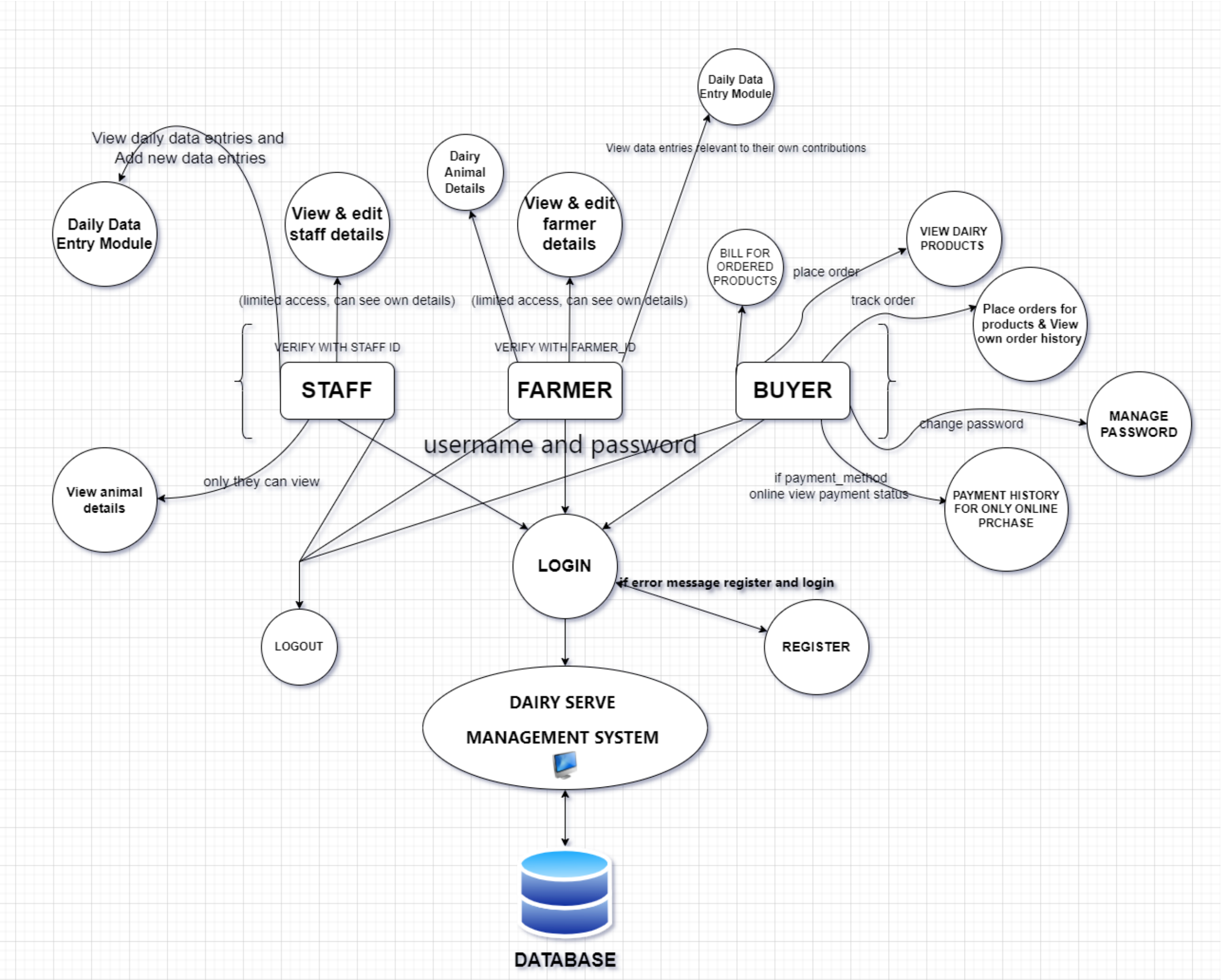
A Data Store represents a logical file; it is represented using two parallel lines. A logical file can represent either a Data Store Symbol, which can represent either a data structure or a physical file on disk. Each data store is connected to a process utilizing a Data Flow Symbol. The direction of the Data Flow Arrow shows whether data is being read from or written into a Data Store. An arrow flowing in or out of a data store implicitly represents the entire area of the Data Store and hence arrows connecting to a data store need not be annotated with the names of the corresponding data items.





DFD LEVEL 1- ADMINISTRATOR

DFD LEVEL 2- FOR DIFFERENT USERS


Daily Data Entry Module
View data entries relevant to their own contributions
View daily data entries and Add new data entries
Dairy Animal Details
Daily Data Entry Module
View & edit staff details
View & edit farmer details
BILL FOR ORDERED PRODUCTS
place order
VIEW DAIRY PRODUCTS
track order
Place orders for products & View own order history
(limited access, can see own details)
(limited access, can see own details)
VERIFY WITH STAFF ID
VERIFY WITH FARMER_ID
STAFF
FARMER
BUYER
change password
MANAGE PASSWORD
username and password
View animal details
only they can view
if payment_method online view payment status
PAYMENT HISTORY FOR ONLY ONLINE PRCHASE
LOGIN
if error message register and login
LOGOUT
REGISTER
DAIRY SERVE MANAGEMENT SYSTEM
DATABASE

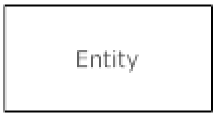
DFD LEVEL 2- FOR DIFFERENT USERS

5.3 ENTITY RELATIONSHIP DIAGRAM

An entity-relationship model (ERM) is a theoretical and conceptual way of showing data relationships in software development. ERM is a database modeling technique that generates an abstract diagram or visual representation of a system's data that can help design a relational database. These diagrams are known as entity-relationship diagrams, ER diagrams, or ERDs.

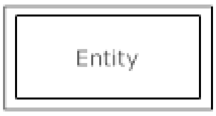
Entity

An entity is an object or concept about which you want to store information.



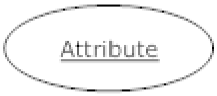
Weak Entity

A weak entity is an entity that must defined by a foreign key relationship with another entity as it cannot be uniquely identified by its attributes alone.



Key attribute

A key attribute is the unique, distinguishing characteristic of the entity. For example, an employee's social security number might be the employee's key attribute.

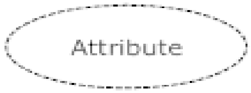


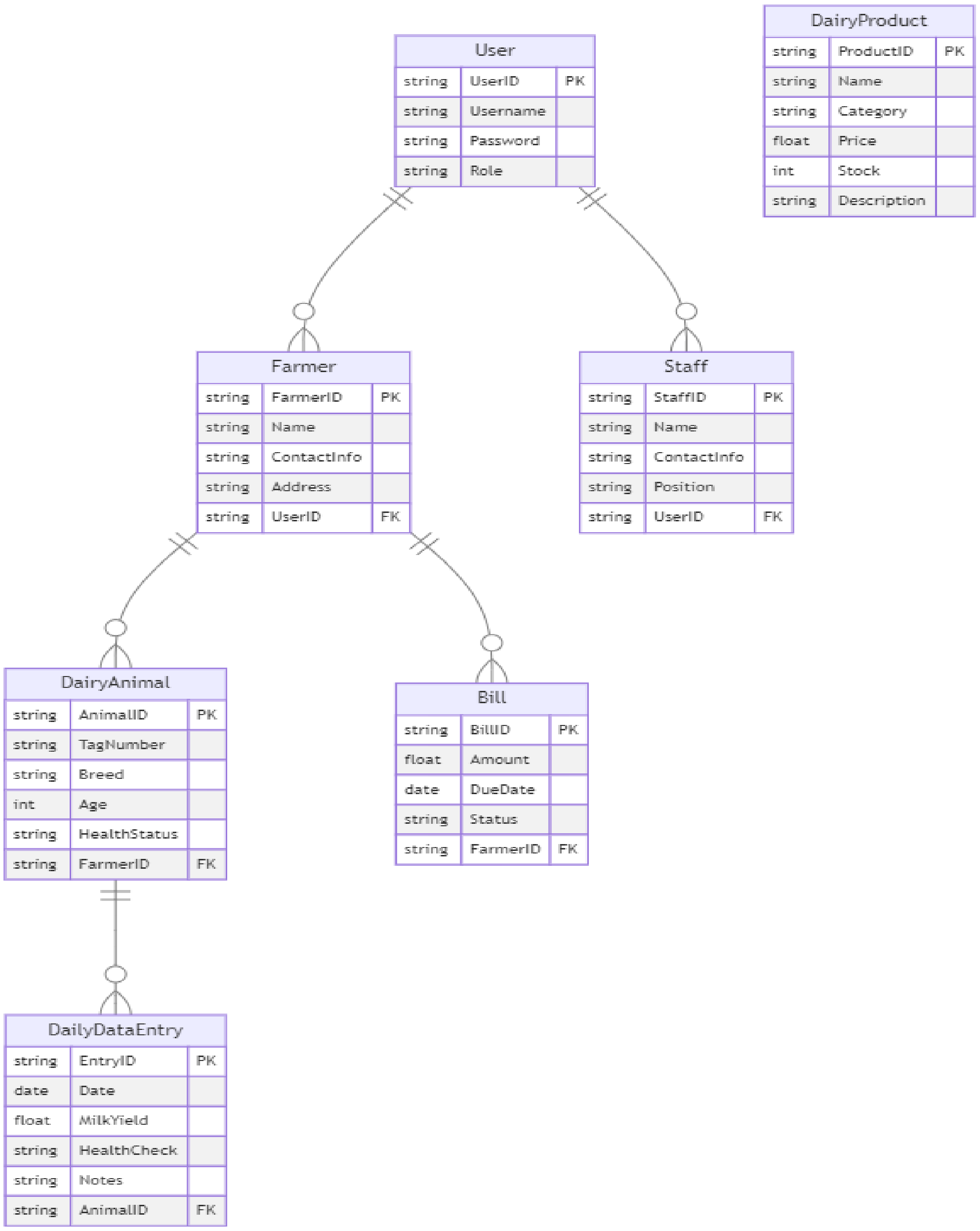
Multivalued attribute

A multivalued attribute can have more than one value. For example, an employee entity can have multiple skill values.

Derived attribute

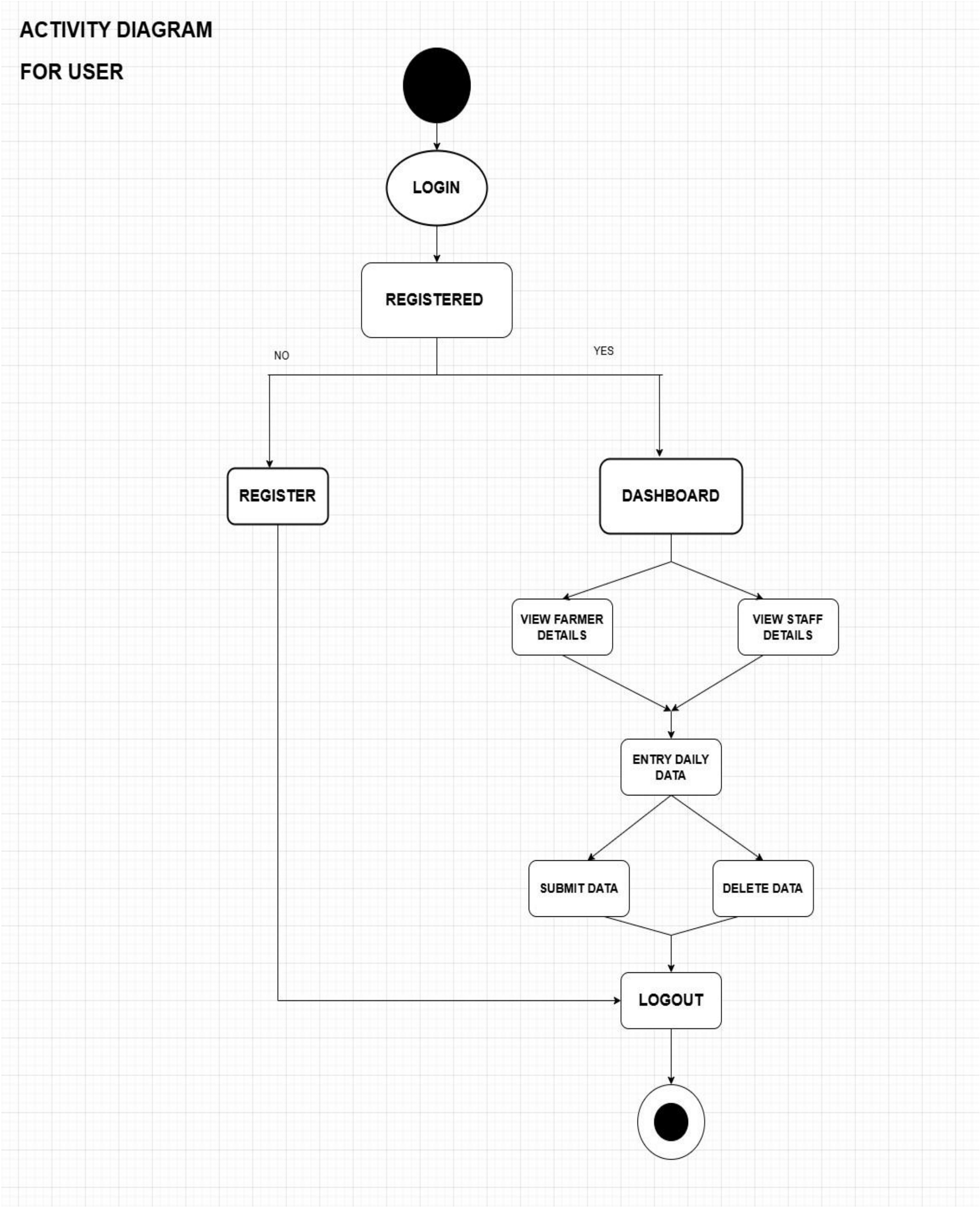
A derived attribute is based on another attribute. For example, an employee's monthly salary is based on the employee's annual salary.



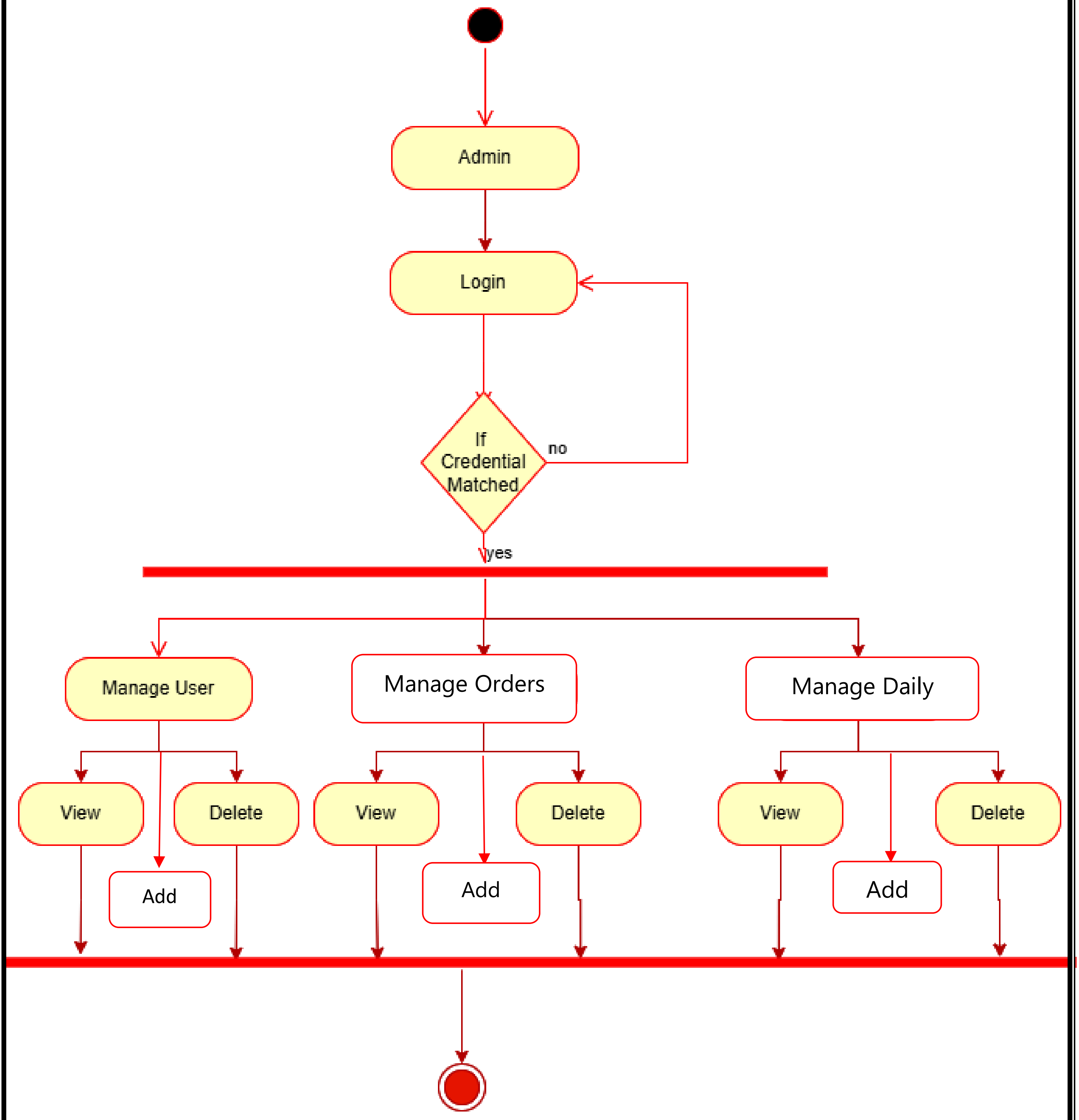


5.4 ACTIVITY DIAGRAM

Activity diagrams are graphical representations of workflows of stepwise activities and actions with support for choice, iteration, and concurrency. In the Unified Modelling Language, activity diagrams are intended to model both computational and organizational processes (i.e. workflows). Activity diagrams show the overall flow of control.

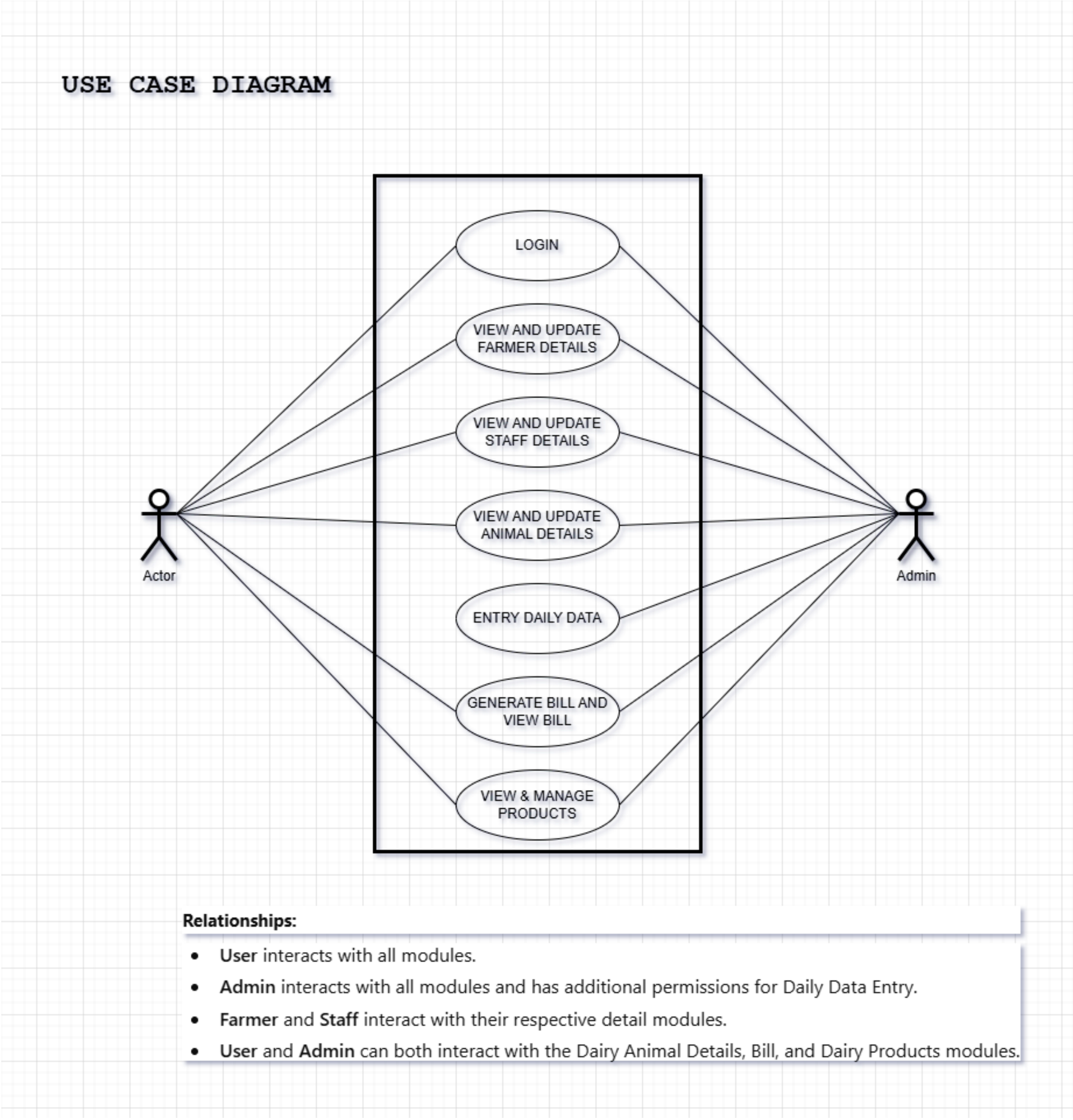


ACTIVITY DIAGRAM FOR ADMIN



5.5 USE-CASE DIAGRAM

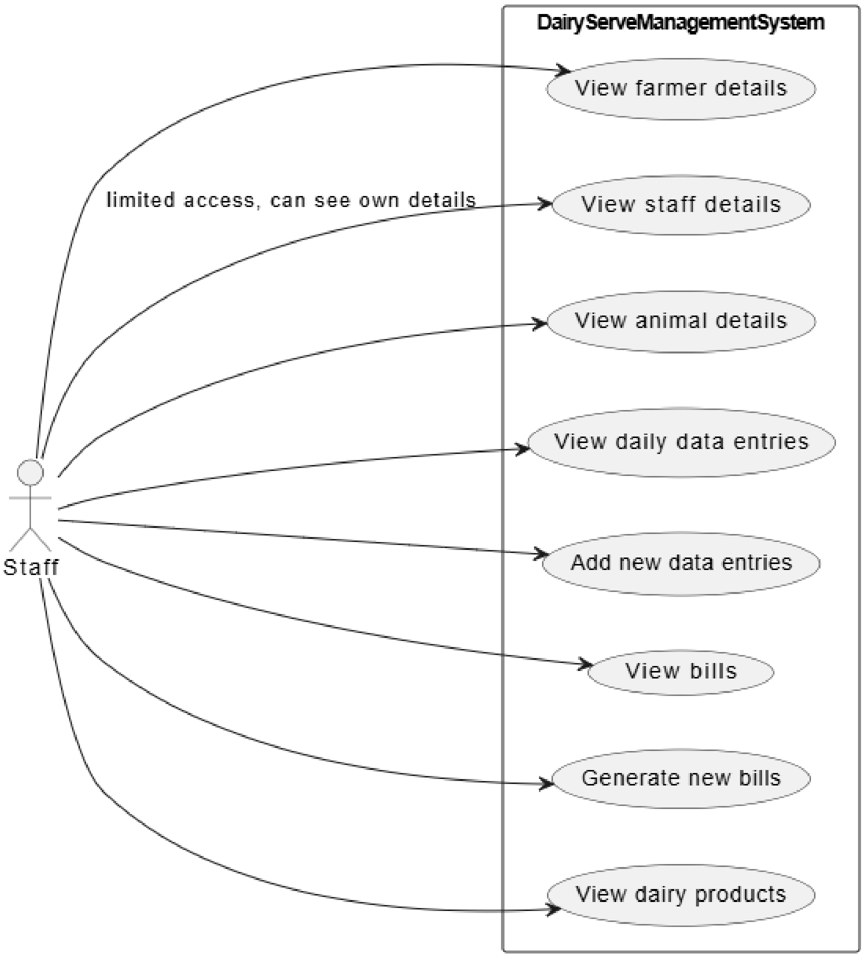
A use case diagram at its simplest is a representation of a user's interaction with the system that shows the relationship between the user and the different use cases in which the user is involved. A use case diagram can identify the different types of users of a system and the different use cases and will often be accompanied by other types of diagrams as well

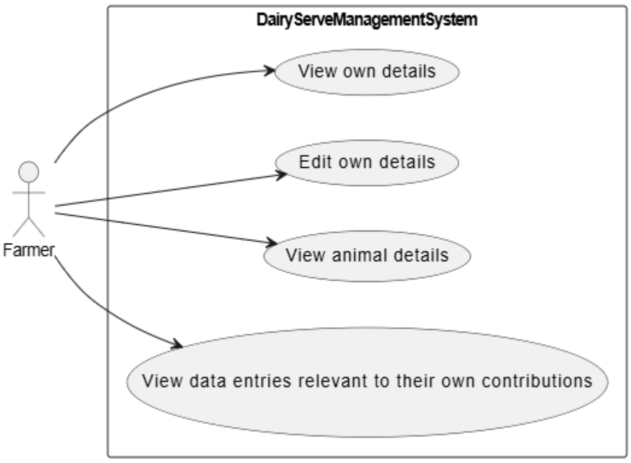

USE CASE DIAGRAM
LOGIN
VIEW AND UPDATE FARMER DETAILS
VIEW AND UPDATE STAFF DETAILS
VIEW AND UPDATE ANIMAL DETAILS
ENTRY DAILY DATA
GENERATE BILL AND VIEW BILL
VIEW & MANAGE PRODUCTS
Actor
Admin

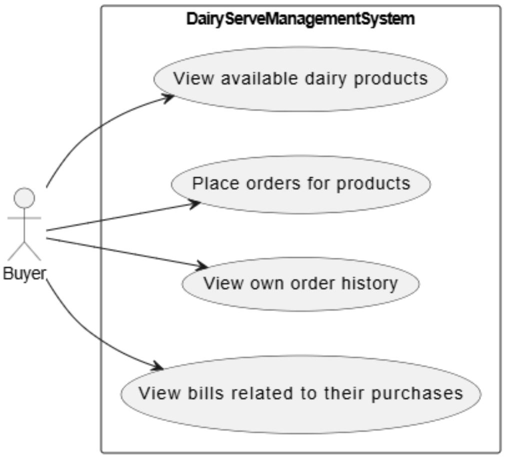
Relationships:

- **User** interacts with all modules.
- **Admin** interacts with all modules and has additional permissions for Daily Data Entry.
- **Farmer** and **Staff** interact with their respective detail modules.
- **User** and **Admin** can both interact with the Dairy Animal Details, Bill, and Dairy Products modules.

SPLIT OF USE CASE

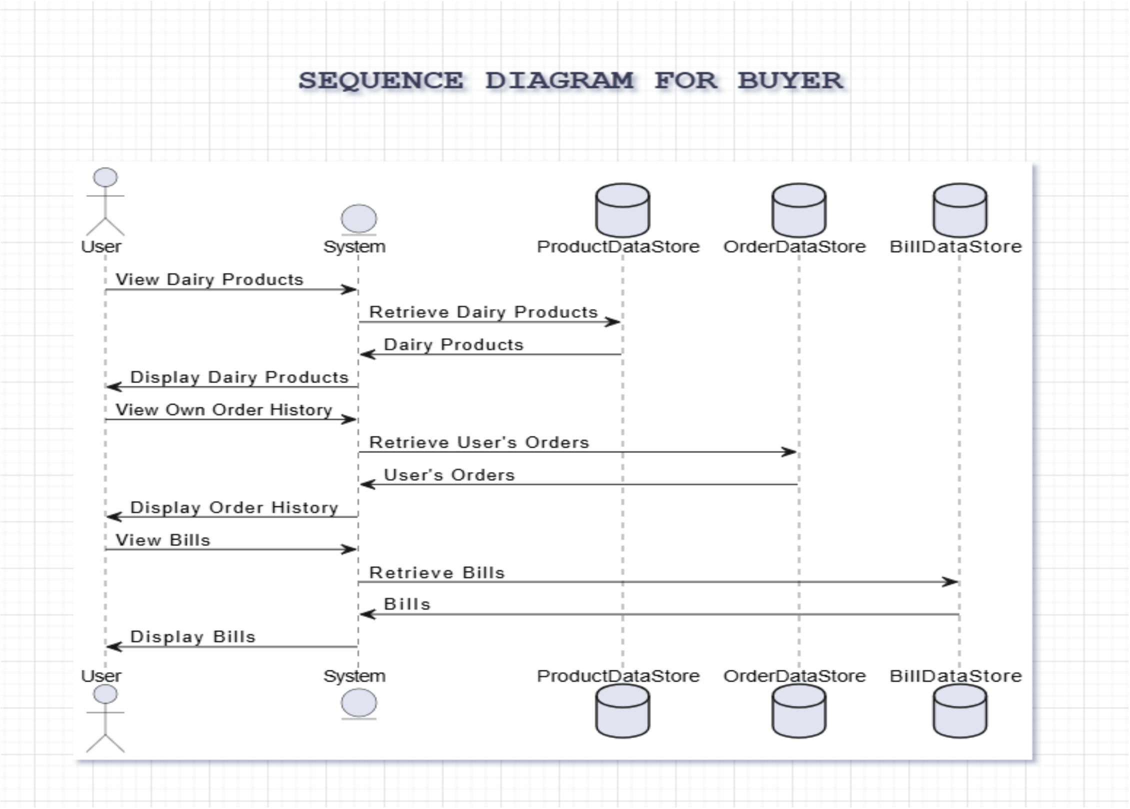


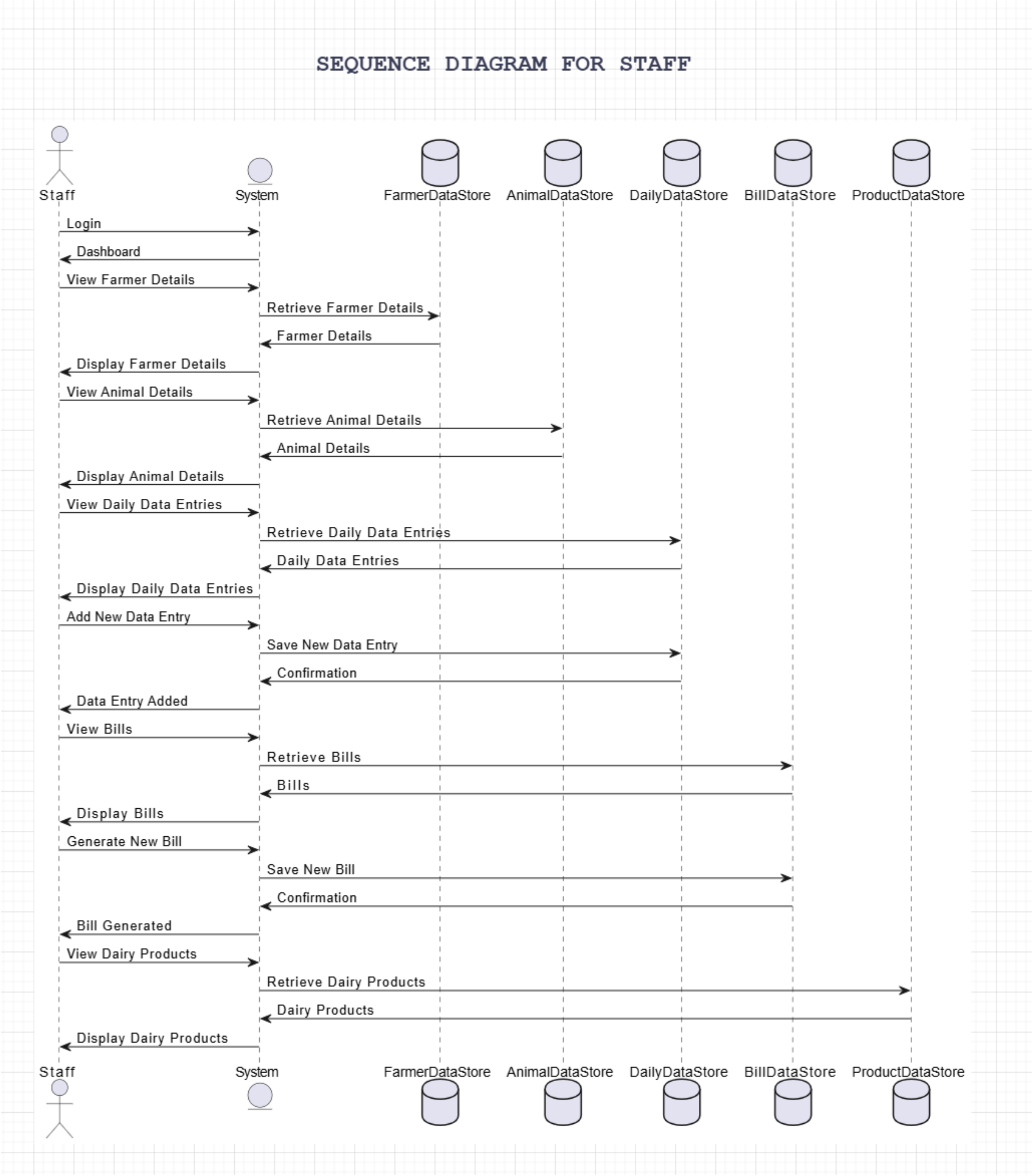

DairyServeManagementSystem
Farmer
View own details
Edit own details
View animal details
View data entries relevant to their own contributions


DairyServeManagementSystem
Buyer
View available dairy products
Place orders for products
View own order history
View bills related to their purchases

5.6 SEQUENCE DIAGRAM

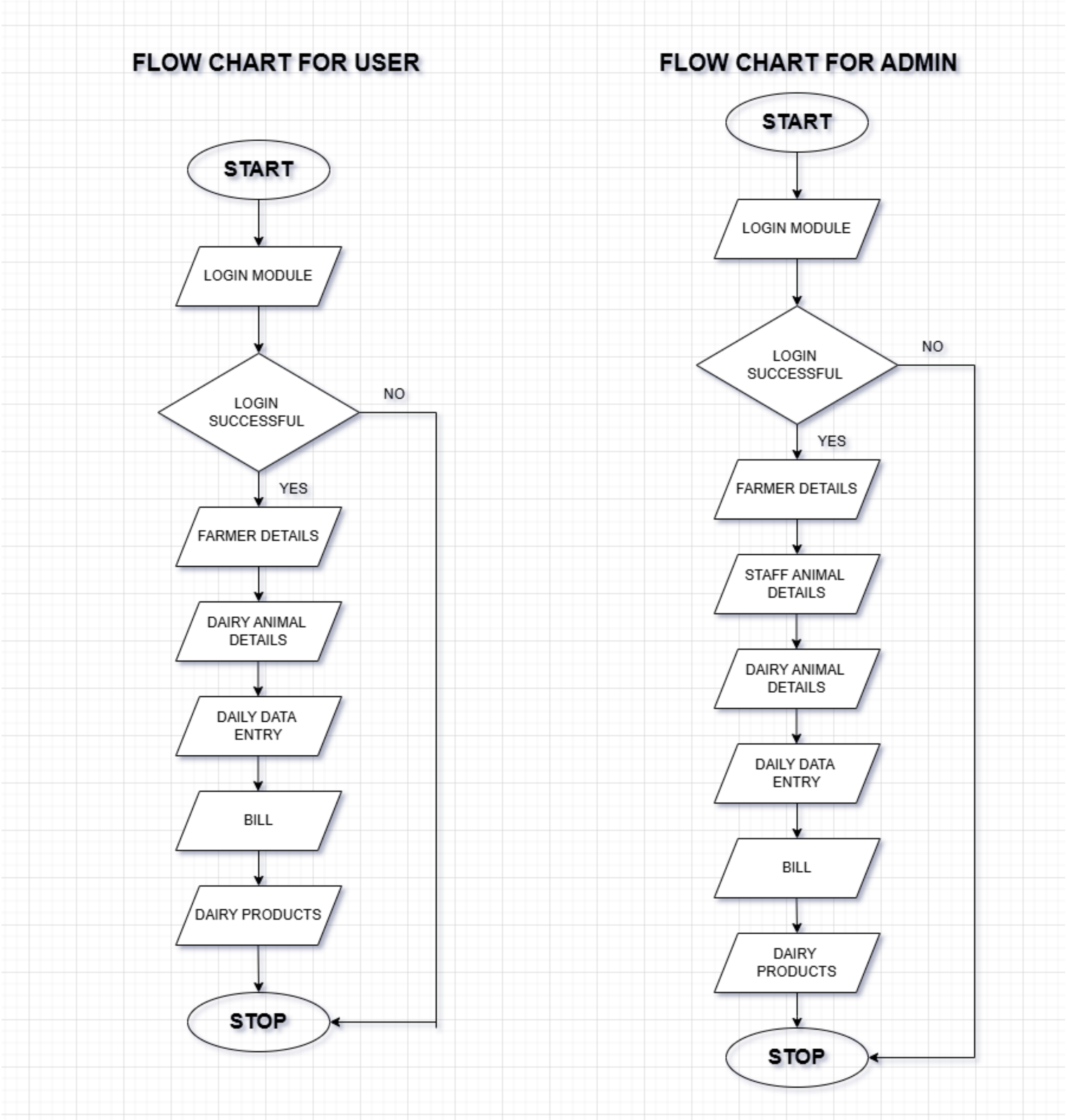
A sequence diagram shows object interactions arranged in a time sequence. It depicts the objects and classes involved in the scenario and the sequence of messages exchanged between the objects needed to carry out the functionality of the scenario. Sequence diagrams are typically associated with use case realizations in the Logical View of the system under development. Sequence diagrams are sometimes called event diagrams or event scenarios.




SEQUENCE DIAGRAM FOR STAFF
Staff
System
FarmerDataStore
AnimalDataStore
DailyDataStore
BillDataStore
ProductDataStore
Login
Dashboard
View Farmer Details
Retrieve Farmer Details
Farmer Details
Display Farmer Details
View Animal Details
Retrieve Animal Details
Animal Details
Display Animal Details
View Daily Data Entries
Retrieve Daily Data Entries
Daily Data Entries
Display Daily Data Entries
Add New Data Entry
Save New Data Entry
Confirmation
Data Entry Added
View Bills
Retrieve Bills
Bills
Display Bills
Generate New Bill
Save New Bill
Confirmation
Bill Generated
View Dairy Products
Retrieve Dairy Products
Dairy Products
Display Dairy Products
Staff
System
FarmerDataStore
AnimalDataStore
DailyDataStore
BillDataStore
ProductDataStore

5.7 FLOWCHARTS

A flowchart is a type of diagram that represents an algorithm, workflow, or process, showing the steps as boxes of various kinds, and their order by connecting them with arrows. This diagrammatic representation illustrates a solution model to a given problem. Flowcharts are used in analyzing, designing, documenting, or managing a process or program in various fields.



6. MODULES

ADMINISTRATOR

Admin Privileges:	Farmer Details Module:	View farmer details Add new farmers Edit farmer details Delete farmers
	Staff Details Module:	View staff details Add new staff members Edit staff details Delete staff members
	Dairy Animal Details:	View animal details Add new animal records Edit animal details Delete animal records
	Daily Data Entry Module:	View daily data entries Add new data entries Edit existing data entries Delete data entries
	Bill Module:	View bills Generate new bills Edit bills Delete bills

	Dairy Products Module:	View dairy products Add new products Edit product details Delete products

STAFF

Staff Privileges:	Farmer Details Module:	View farmer details
	Staff Details Module:	View staff details (limited access, can see own details)
	Dairy Animal Details:	View animal details
	Daily Data Entry Module:	View daily data entries Add new data entries
	Bill Module:	View bills Generate new bills

FARMER

Farmer Privileges:	Farmer Details Module:	View own details Edit own details
	Dairy Animal Details:	View animal details
	Daily Data Entry Module:	View data entries relevant to their contributions

BUYER

Buyer Privileges:	Dairy Products Module:	View available dairy products Place orders for products View your order history
	Bill Module:	View bills related to their purchases

6.1.1 Login and Register Module

Admin and Users (Staff, Farmer, Buyer):

- **Login:** Provides a secure authentication process for admins and users (staff, farmers, and buyers) to access the system. Each user is required to enter their credentials (username and password) to gain access to their respective dashboards and functionalities.
- **Register:** Allows new users (staff, farmers, and buyers) to create an account in the system. During registration, users provide necessary information such as name, contact details, and role (staff, farmer, or buyer). Admins can register themselves and other users as needed.

6.1.2 Admin Privileges

Farmer Details Module:

- **View Farmer Details:** Admins can view comprehensive details of all farmers registered in the system, including personal information, contact details, and associated dairy animals.
- **Add New Farmers:** Admins can register new farmers by entering their details into the system.
- **Edit Farmer Details:** Admins can update or correct any information related to existing farmers.
- **Delete Farmers:** Admins have the authority to remove a farmer's details from the system.

Staff Details Module:

- **View Staff Details:** Admins can view detailed information about all staff members.
- **Add New Staff Members:** Admins can add new staff members to the system.
- **Edit Staff Details:** Admins can update or correct information for existing staff members.
- **Delete Staff Members:** Admins can remove staff member details from the system.

Dairy Animal Details Module:

- **View Animal Details:** Admins can view detailed records of all dairy animals, including health records, production statistics, and ownership.
- **Add New Animal Records:** Admins can add new animal records to the system.
- **Edit Animal Details:** Admins can update information about existing animals.
- **Delete Animal Records:** Admins can remove animal records from the system.

Daily Data Entry Module:

- **View Daily Data Entries:** Admins can view all data entries related to daily dairy operations, including milk production, feed consumption, and health checks.
- **Add New Data Entries:** Admins can add new data entries.
- **Edit Existing Data Entries:** Admins can update or correct existing data entries.
- **Delete Data Entries:** Admins can remove data entries from the system.

Bill Module:

- **View Bills:** Admins can view all bills generated within the system.
- **Generate New Bills:** Admins can create new bills for services or products.
- **Edit Bills:** Admins can update or correct existing bills.
- **Delete Bills:** Admins can remove bills from the system.

Dairy Products Module:

- **View Dairy Products:** Admins can view detailed information about all dairy products available.
- **Add New Products:** Admins can add new products to the system.

- **Edit Product Details:** Admins can update or correct product details.
- **Delete Products:** Admins can remove products from the system.

6.1.3 Staff Privileges

Farmer Details Module:

- **View Farmer Details:** Staff members can view the details of all farmers but cannot make any changes.

Staff Details Module:

- **View Staff Details (Limited Access):** Staff can view their details and limited information about their colleagues.

Dairy Animal Details Module:

- **View Animal Details:** Staff can view detailed records of all dairy animals.

Daily Data Entry Module:

- **View Daily Data Entries:** Staff can view all daily data entries.
- **Add New Data Entries:** Staff can add new data entries related to their duties.

Bill Module:

- **View Bills:** Staff can view all bills.
- **Generate New Bills:** Staff can create new bills for services or products.

Dairy Products Module:

- **View Dairy Products:** Staff can view detailed information about all dairy products available.

6.1.4 Farmer Privileges

Farmer Details Module:

- **View Own Details:** Farmers can view their details.
- **Edit Own Details:** Farmers can update their personal information.

Dairy Animal Details Module:

- **View Animal Details:** Farmers can view records of animals associated with them.

Daily Data Entry Module:

- **View Data Entries Relevant to Their Contributions:** Farmers can view data entries that are specifically related to their contributions.

6.1.5 Buyer Privileges

Dairy Products Module:

- **View Available Dairy Products:** Buyers can view information about all available dairy products.
- **Place Orders for Products:** Buyers can place orders for products through the system.
- **View Own Order History:** Buyers can view their order history, including past purchases and the status of current orders.

Bill Module:

- **View Bills Related to Their Purchases:** Buyers can view bills related to their purchases.

6.2 IMPLEMENTATION

The goals of the implementation phase are to translate the design of the system produced during the phase, into coded form in a given programming language, which can then be executed by a computer performing the computation specified by the design the coding phase affects both testing and maintenance profoundly. Well-written code can reduce testing and maintenance costs.

A crucial phase in the system lifecycle is the successful implementation of the system design. Implementation simply means converting the system designs into operation. Implementation is the process of bringing the developed system into operational use and providing it to the user.

This stage is considered to be the most crucial stage in the development of a successful system since a new system is developed and the users are virtually getting information.

Implementation is a stage in which the design is converted into a working system that is it is the stage of the project where theoretical design is turned into a working system. The implementation involves careful planning, investing in the current system and its constraints on implementation, design of methods to achieve the changeover.

The Project is implemented in different phases as follows:

- The first phase includes table design for the database module.
- The second phase includes coding for modules.
- The third phase includes the integration of modules.
- The fourth phase includes connection establishment between the front end and back end.
- The fifth phase includes error handling and a message generator.

The coding was done with the following characteristics in mind,

- Code efficiency
- Memory efficiency
- Response time
- Security
- Maintainability
- Efficient and consistent logic

7. SYSTEM TESTING

Software testing is an investigation conducted to provide stakeholders with information about the quality of the product or service under test. Software testing can also provide an objective, independent view of the software to allow the business to appreciate and understand the risks of software implementation. Test techniques include, but are not limited to, the process of executing a program or application with the intent of finding software bugs (errors or other defects).

Software testing is the process of evaluating a software item to detect differences between a given input and expected output. Also to assess the features of A software item. Testing assesses the quality of the product. Software testing is a process that should be done during the development process. In other words, software testing is a verification and validation process.

VERIFICATION

Verification is the process to make sure the product satisfies the conditions imposed at the start of the development phase. In other words, to make sure the product behaves the way we want it to.

VALIDATION

Validation is the process of making sure the product satisfies the specified requirements at the end of the development phase. In other words, to make sure the product is built as per customer requirements.

Basics Of Software Testing

There are two basics of software testing: Blackbox testing and Whitebox testing.

BLACK BOX TESTING

Black box testing is a testing technique that ignores the internal mechanism of the system and focuses on the output generated against any input and execution of the system. It is also called functional testing.

WHITE BOX TESTING

White box testing is a testing technique that takes into account the internal mechanism of a system. It is also called structural testing and glass box testing.

Black box testing is often used for validation and white box testing is often used for verification.

7.1 TYPES OF TESTING

- Unit Testing
- Integration Testing
- Functional Testing
- System Testing
- Stress Testing
- Performance Testing
- Usability Testing
- Acceptance Testing
- Regression Testing
- Beta Testing

7.2 UNIT TESTING

Unit testing is the testing of an individual unit or group of related units. It falls under the class of white box testing. It is often done by the programmer to test that the unit he/she has implemented is producing the expected output against the given input.

7.3 INTEGRATION TESTING

Integration testing is testing in which a group of components are combined to produce output. Also, the interaction between software and hardware is tested in integration testing if software and hardware components have any relation. It may fall under both white-box testing and black-box testing.

7.4 FUNCTIONAL TESTING

Functional testing is the testing to ensure that the specified functionality required in the system requirements works. It falls under the class of black box testing.

7.5 SYSTEM TESTING

System testing is the testing to ensure that by putting the software in different environments (e.g., Operating Systems) it still works. System testing is done with full system implementation and environment. It falls under the class of black box testing.

7.6 STRESS TESTING

Stress testing is the testing to evaluate how the system behaves under unfavorable conditions. Testing is conducted beyond the limits of the specifications. It falls under the class of black box testing.

7.7 PERFORMANCE TESTING

Performance testing is the testing to assess the speed and effectiveness of the system and to make sure it is generating results within a specified time as in performance requirements. It falls under the class of black box testing.

7.8 USABILITY TESTING

Usability testing is performed from the perspective of the client, to evaluate how the GUI is user-friendly. How easily can the client learn? After learning how to use it, how proficiently can the client perform? How pleasing is it to use its design? This falls under the class of black box testing.

7.9 ACCEPTENCE TESTING

Acceptance testing is often done by the customer to ensure that the delivered product meets the requirements and works as the customer expected. It falls under the class of black box testing.

7.10 REGRESSION TESTING

Regression testing is the testing after modification of a system, component, or a group of related units to ensure that the modification is working correctly and is not damaging or imposing other modules to produce unexpected results. It falls under the class of black box testing.

7.11 BETA TESTING

Beta testing is the testing that is done by end users, a team outside development, or publicly releasing full pre-version of the product which is known as the beta version. Beta testing aims to cover unexpected errors. It falls under the class of black box testing. A validation from the specification is uncovered and a deficiency is created.

Deviations or errors discovered at this step in this project are corrected before completion of the project with the help of the user by negotiating to establish a method for resolving deficiencies. Thus, the proposed system under consideration has been tested by using validation testing and found to be working satisfactorily.

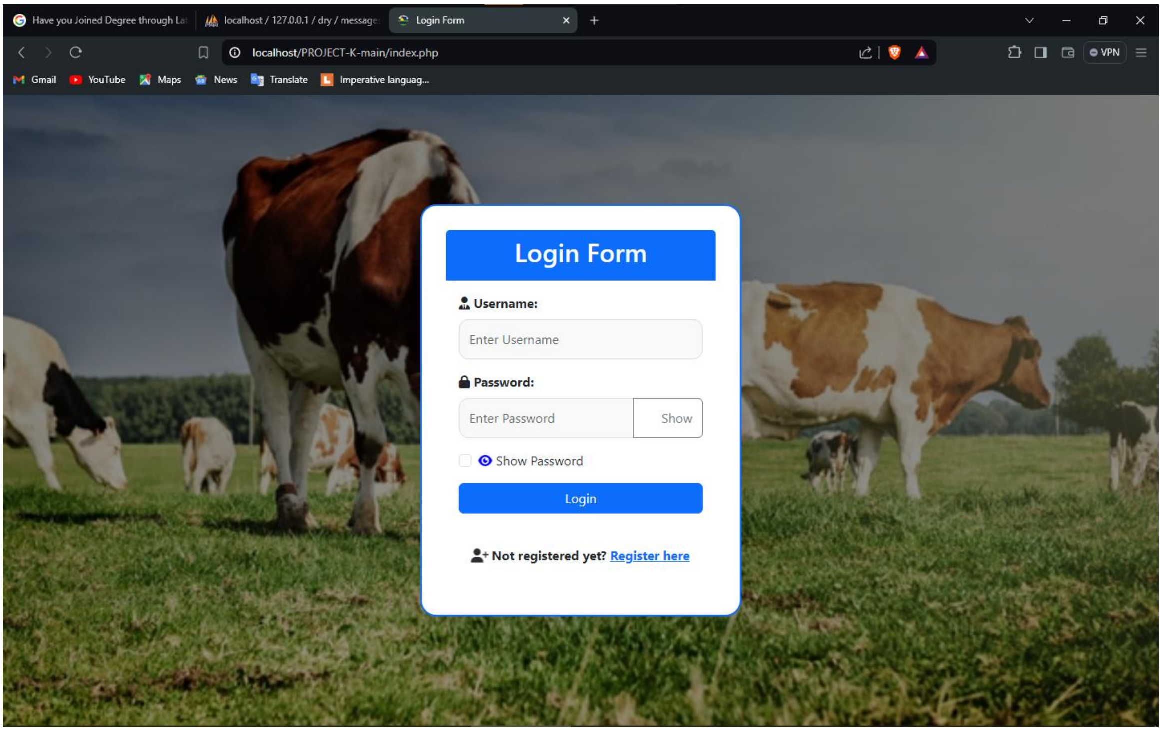
7.12 TEST CASES

Module	Test Case ID	Test Case Description	Expected Result
Login Module	LM01	Verify login with valid credentials	The user should be successfully logged in
	LM02	Verify login with invalid credentials	The user should receive an error message
	LM03	Verify login with an empty username or password	The user should receive a validation error message
Farmer Details	FM01	Verify adding a new farmer with valid details	Farmer should be successfully added
	FM02	Verify updating farmer details	Farmer details should be successfully updated
	FM03	Verify deleting a farmer	The farmer should be successfully deleted
	FM04	Verify viewing farmer details	Farmer details should be displayed correctly
Staff Details	SM01	Verify adding a new staff member with valid details	Staff members should be successfully added
	SM02	Verify updating staff details	Staff details should be successfully updated
	SM03	Verify deleting a staff member	The staff member should be successfully deleted
	SM04	Verify viewing staff details	Staff details should be displayed correctly

Dairy Animal Details	DA01	Verify adding a new dairy animal with valid details	Dairy animals should be successfully added
	DA02	Verify updating dairy animal details	Dairy animal details should be successfully updated
	DA03	Verify deleting a dairy animal	Dairy animals should be successfully deleted
	DA04	Verify viewing dairy animal details	Dairy animal details should be displayed correctly
Daily Data Entry	DD01	Verify entering daily data with valid details	Daily data should be successfully entered
	DD02	Verify updating daily data	Daily data should be successfully updated
	DD03	Verify viewing daily data entries	Daily data entries should be displayed correctly
Bill Module	BM01	Verify generating a bill	Bill should be successfully generated
	BM02	Verify viewing bills	Bills should be displayed correctly
	BM03	Verify paying a bill	Bill should be successfully paid
Dairy Products	DP01	Verify adding a new dairy product with valid details	Dairy products should be successfully added
	DP02	Verify updating dairy product details	Dairy product details should be successfully updated
	DP03	Verify deleting a dairy product	Dairy products should be successfully deleted
	DP04	Verify viewing dairy products	Dairy products should be displayed correctly

8. SNAPSHOTS

LOGIN PAGE



Have you Joined Degree through La
localhost / 127.0.0.1 / dry / message
Login Form
localhost/PROJECT-K-main/index.php
Gmail
YouTube
Maps
News
Translate
Imperative languag...
VPN

Login Form

Username:

Enter Username

Password:

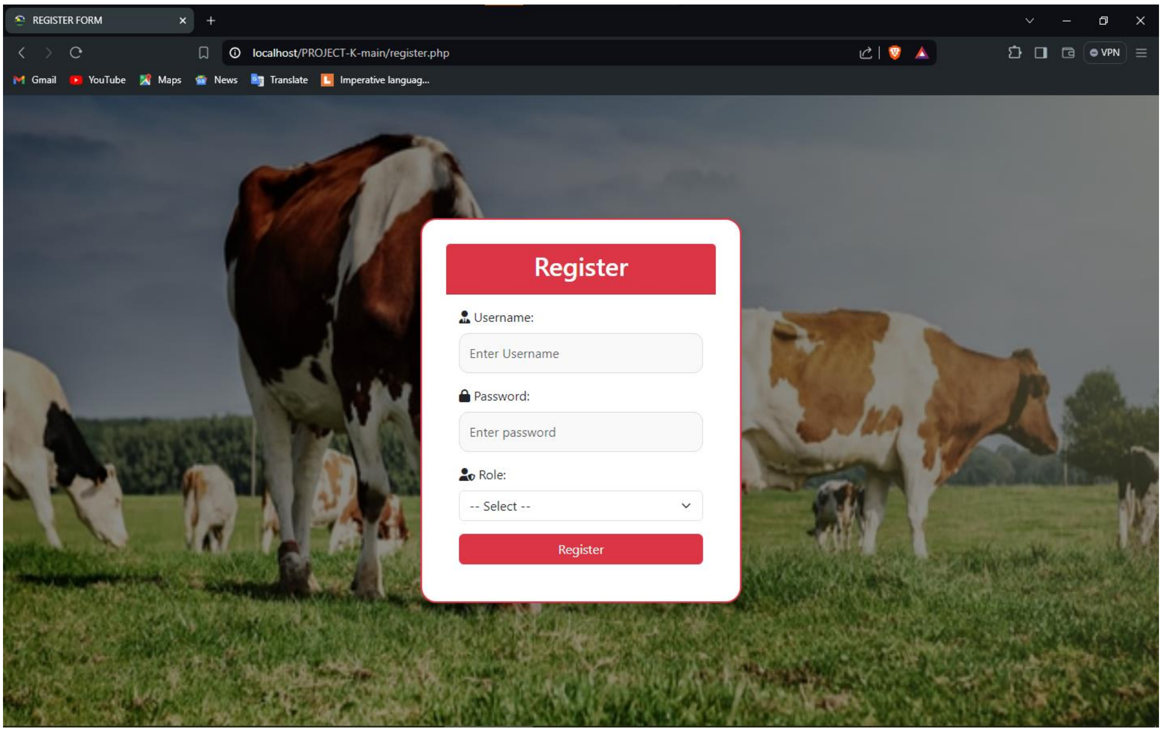
Enter Password | Show

□ Show Password

Login

Not registered yet? Register here

REGISTRATION PAGE



Register

Username:

Enter Username

Password:

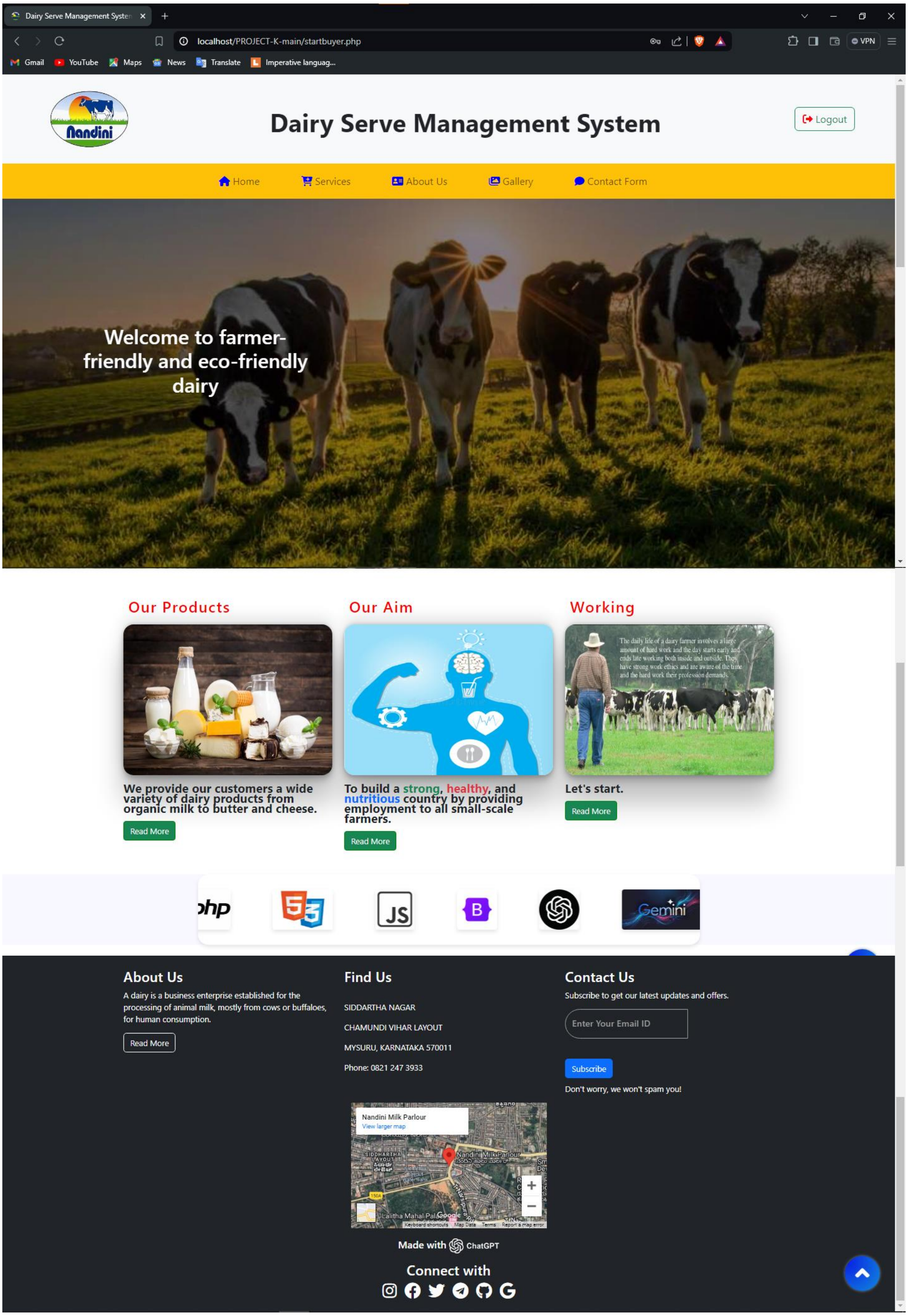
Enter password

Role:

-- Select --

Register

ADMIN



Dairy Serve Management System

Welcome to farmer-friendly and eco-friendly dairy

Our Products

We provide our customers a wide variety of dairy products from organic milk to butter and cheese.

Our Aim

To build a strong, healthy, and nutritious country by providing employment to all small-scale farmers.

Working

The daily life of a dairy farmer involves a large amount of hard work and the day starts early and ends late working both inside and outside. They have strong work ethics and are aware of the time and the hard work their profession demands.

Let's start.

About Us

A dairy is a business enterprise established for the processing of animal milk, mostly from cows or buffaloes, for human consumption.

Find Us

SIDDARTHA NAGAR

CHAMUNDI VIHAR LAYOUT

MYSURU, KARNATAKA 570011

Phone: 0821 247 3933

Contact Us

Subscribe to get our latest updates and offers.

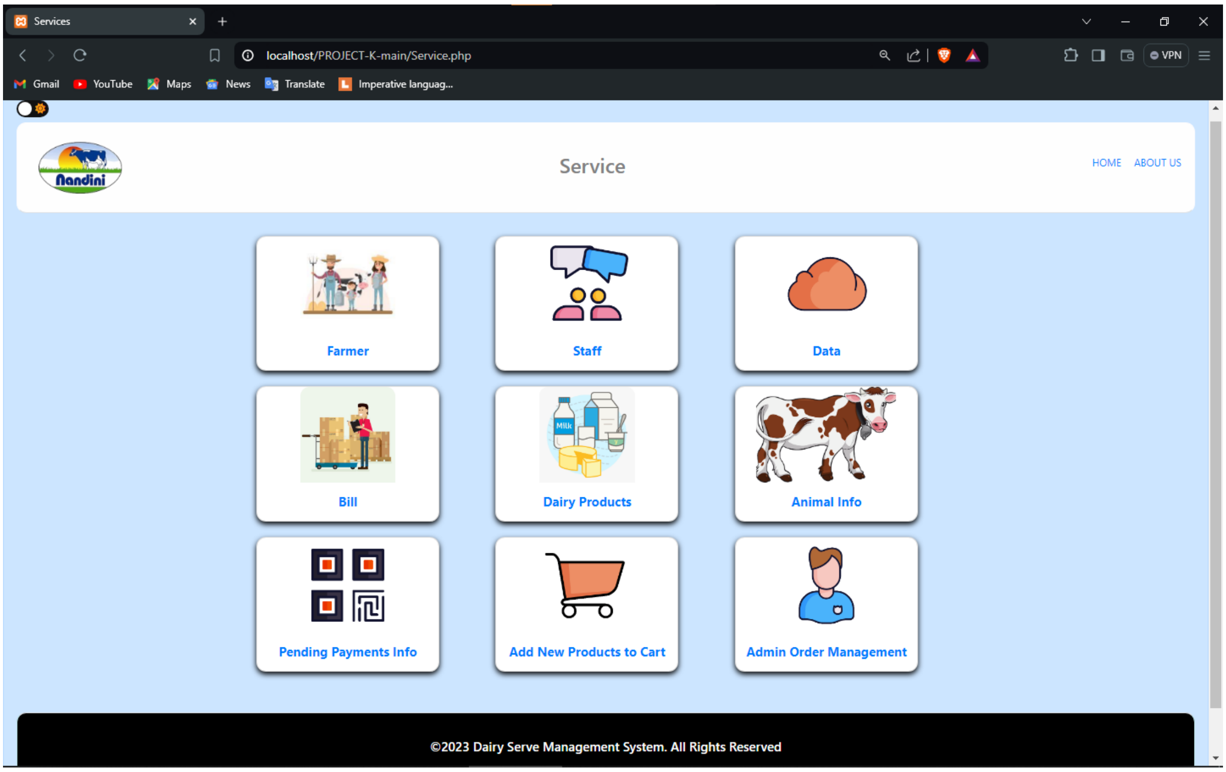
Enter Your Email ID

Don't worry, we won't spam you!

Made with ChatGPT

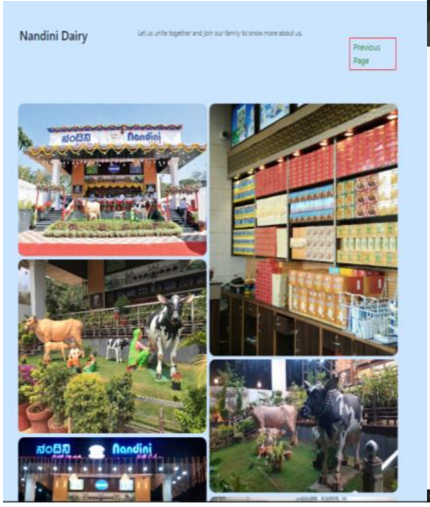
Connect with

ADMIN SERVICE PAGE

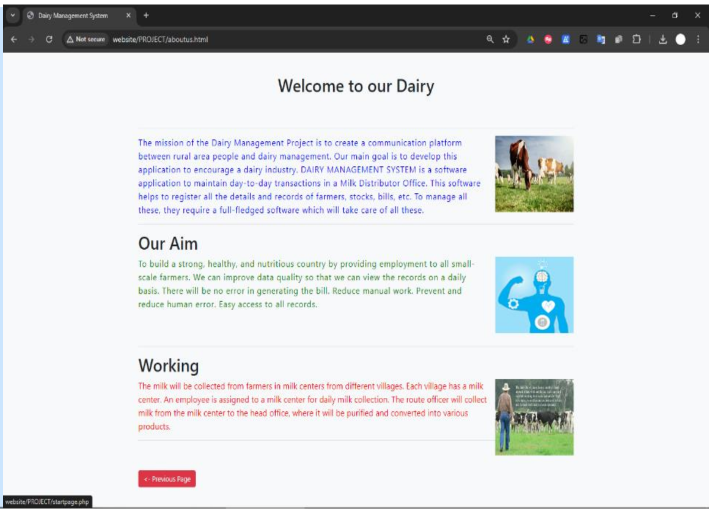


©2023 Dairy Serve Management System. All Rights Reserved

ABOUT & GALLERY PAGE



Nandini Dairy
Let us unite together and join our family to know more about us.
Previous Page
ನಂದಿನಿ
Nandini
ನಂದಿನಿ
Nandini



Welcome to our Dairy

The mission of the Dairy Management Project is to create a communication platform between rural area people and dairy management. Our main goal is to develop this application to encourage a dairy industry. DAIRY MANAGEMENT SYSTEM is a software application to maintain day-to-day transactions in a Milk Distributor Office. This software helps to register all the details and records of farmers, stocks, bills, etc. To manage all these, they require a full-fledged software which will take care of all these.

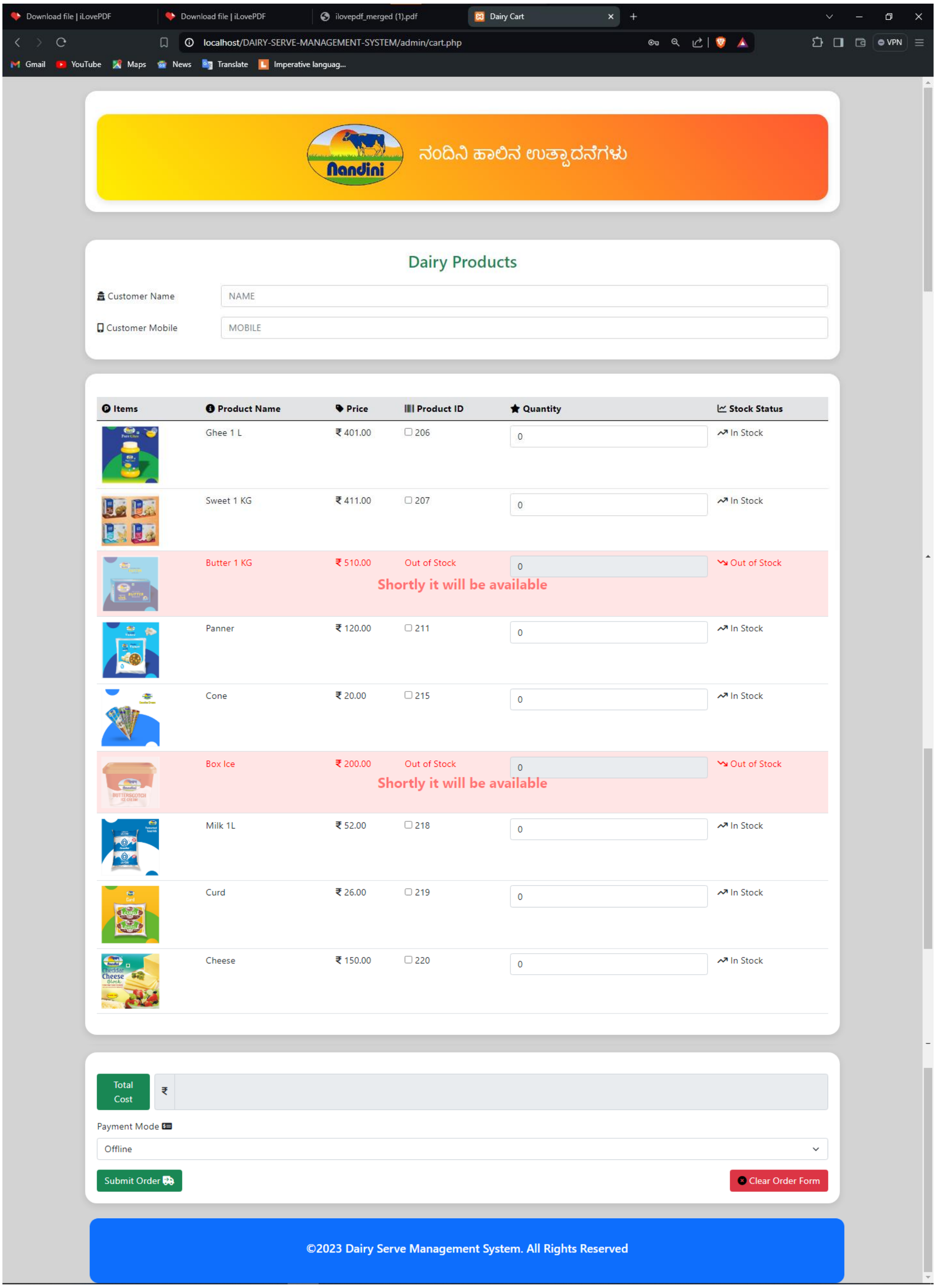
Our Aim

To build a strong, healthy, and nutritious country by providing employment to all small-scale farmers. We can improve data quality so that we can view the records on a daily basis. There will be no error in generating the bill. Reduce manual work. Prevent and reduce human error. Easy access to all records.

Working

The milk will be collected from farmers in milk centers from different villages. Each village has a milk center. An employee is assigned to a milk center for daily milk collection. The route officer will collect milk from the milk center to the head office, where it will be purified and converted into various products.

DAIRY PRODUCTS CART



ನಂದಿನಿ ಹಾಲಿನ ಉತ್ಪಾದನೆಗಳು

Dairy Products

Customer Name: NAME

Customer Mobile: MOBILE

Items	Product Name	Price	Product ID	Quantity	Stock Status
	Ghee 1 L	₹ 401.00	206	0	In Stock
	Sweet 1 KG	₹ 411.00	207	0	In Stock
	Butter 1 KG	₹ 510.00	Out of Stock	0	Out of Stock
	Shortly it will be available				
	Panner	₹ 120.00	211	0	In Stock
	Cone	₹ 20.00	215	0	In Stock
	Box Ice	₹ 200.00	Out of Stock	0	Out of Stock
	Shortly it will be available				
	Milk 1L	₹ 52.00	218	0	In Stock
	Curd	₹ 26.00	219	0	In Stock
	Cheese	₹ 150.00	220	0	In Stock

Total Cost ₹

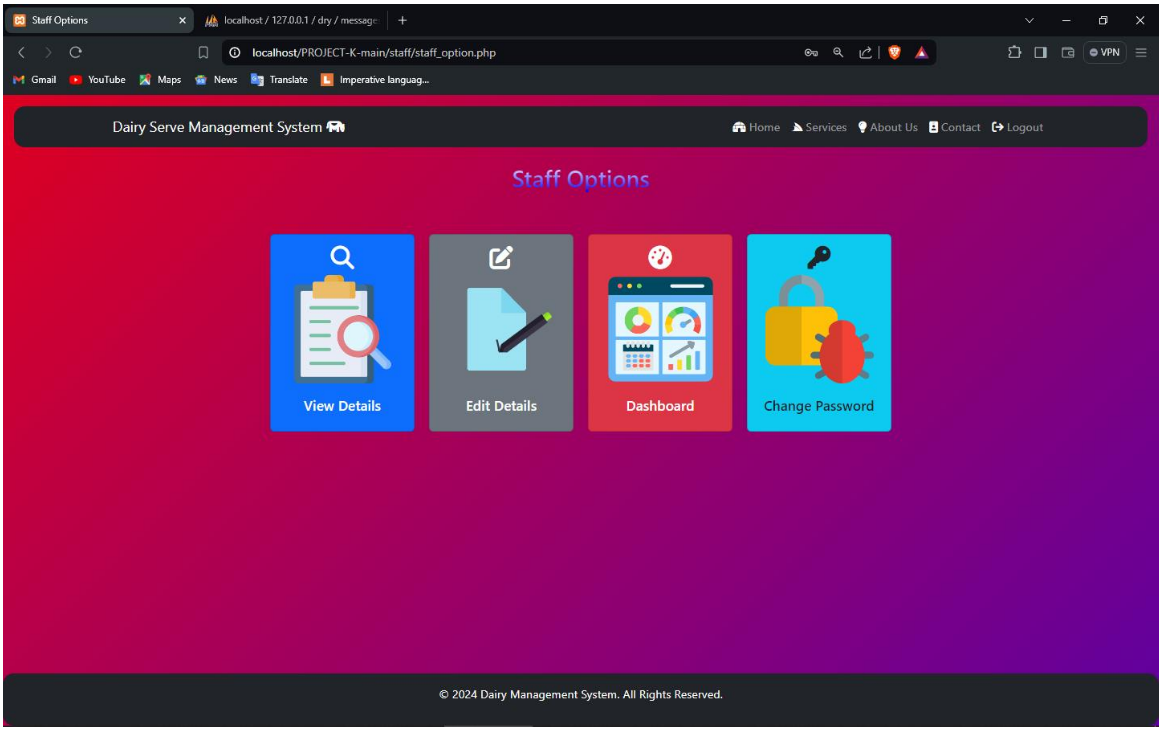
Payment Mode: Offline

Submit Order

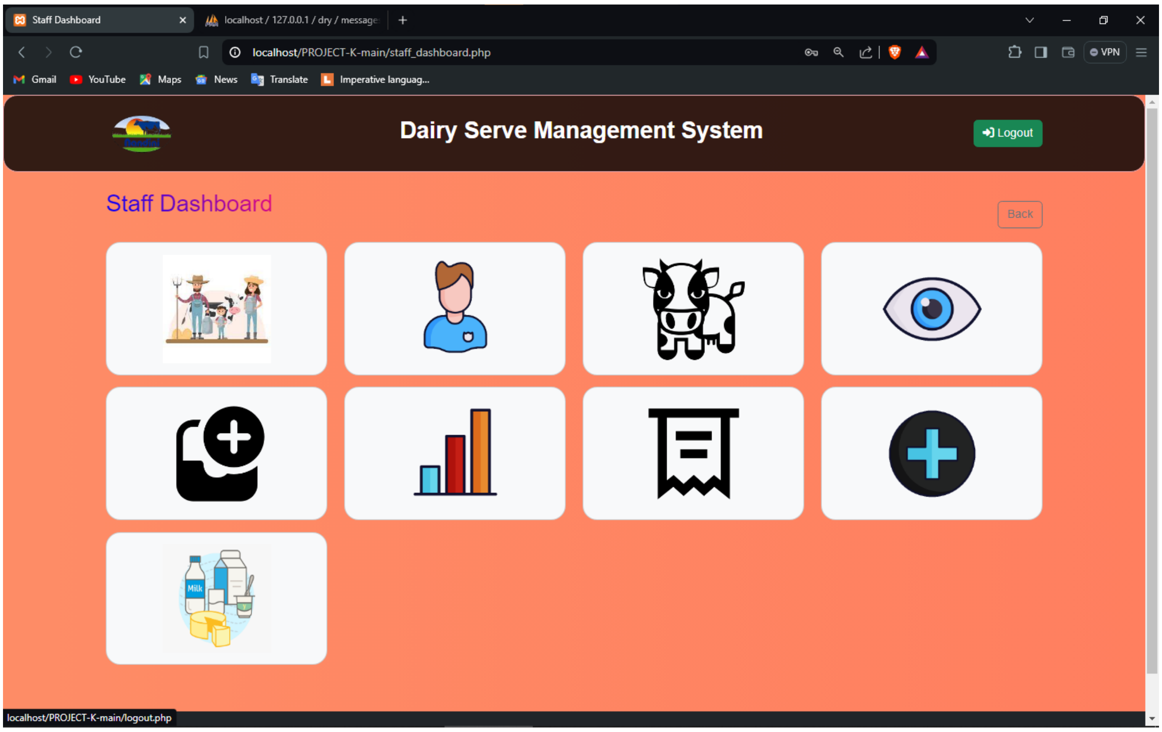
Clear Order Form

©2023 Dairy Serve Management System. All Rights Reserved

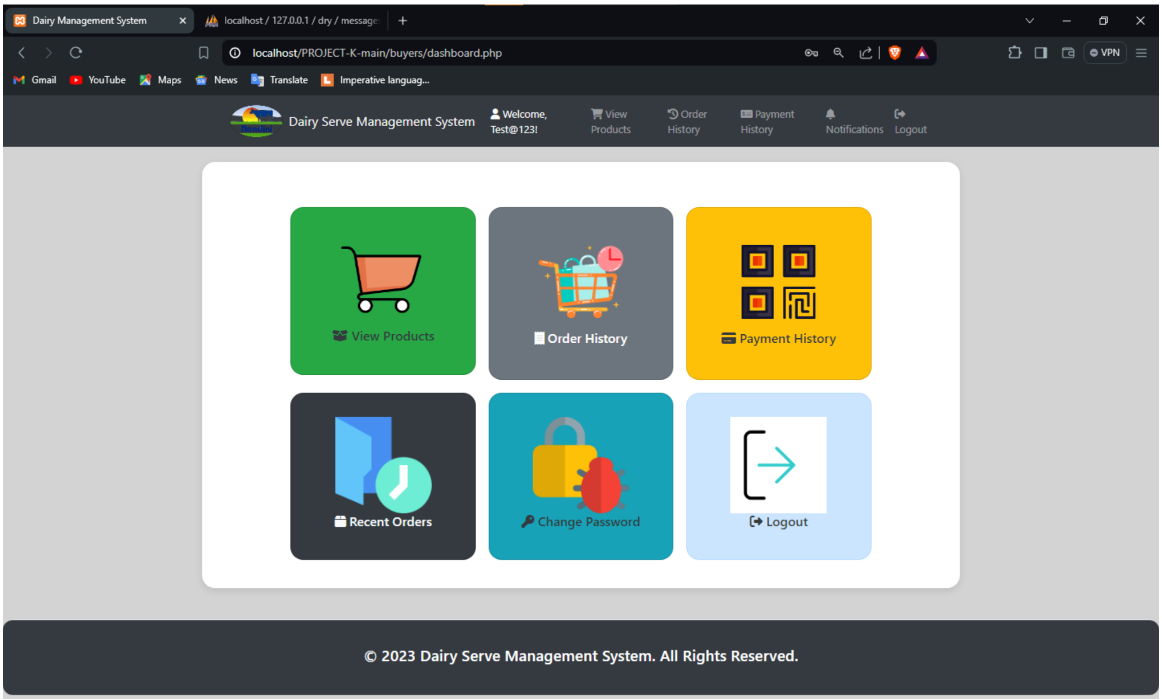
STAFF PAGE


Staff Options
localhost / 127.0.0.1 / dry / message
localhost/PROJECT-K-main/staff/staff_option.php
Gmail
YouTube
Maps
News
Translate
Imperative languag...
Dairy Serve Management System
Home
Services
About Us
Contact
Logout
Staff Options
View Details
Edit Details
Dashboard
Change Password
© 2024 Dairy Management System. All Rights Reserved.

STAFF DASHBOARD

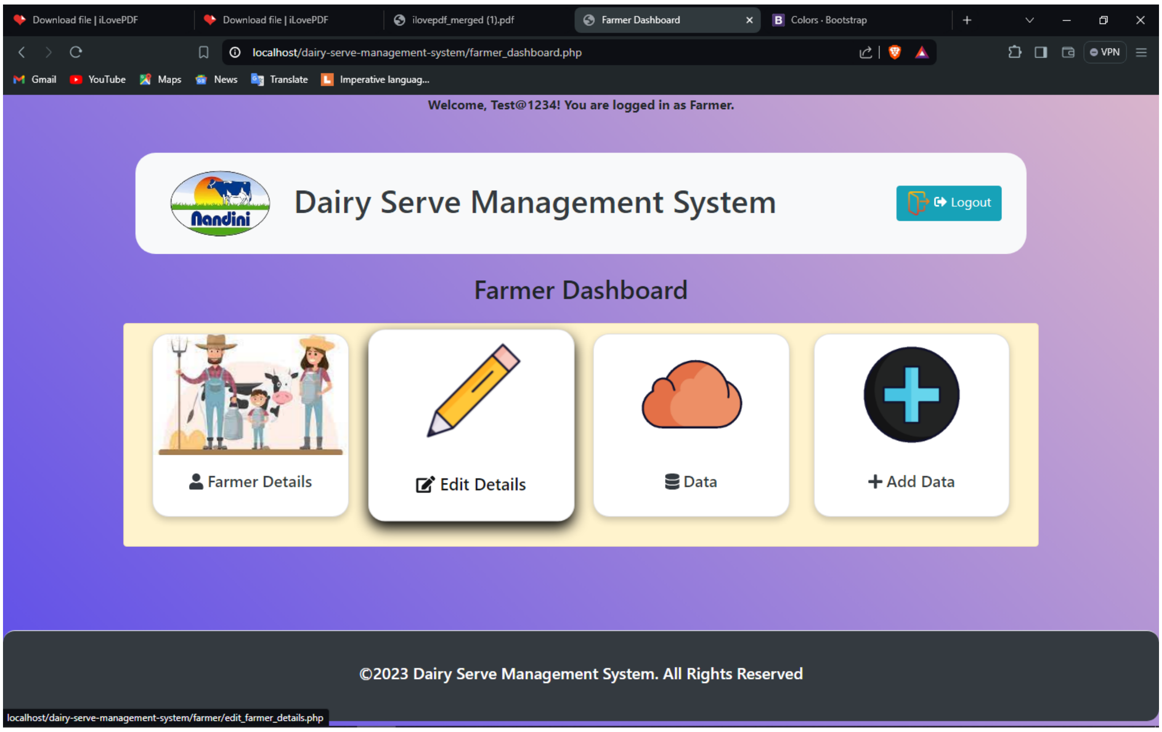

Staff Dashboard
localhost / 127.0.0.1 / dry / message
localhost/PROJECT-K-main/staff_dashboard.php
Gmail
YouTube
Maps
News
Translate
Imperative languag...
Dairy Serve Management System
Logout
Staff Dashboard
Back
Milk
localhost/PROJECT-K-main/logout.php

BUYER DASHBOARD



© 2023 Dairy Serve Management System. All Rights Reserved.

FARMER DASHBOARD



Welcome, Test@1234! You are logged in as Farmer.

©2023 Dairy Serve Management System. All Rights Reserved

9. CONCLUSION

The Dairy Serve Management System (DSMS) is a comprehensive and user-friendly application tailored to meet the diverse needs of administrators, staff, and buyers. Through meticulous development and rigorous testing, the system has proven to be a reliable and efficient solution for managing dairy-related activities. DSMS prioritizes the user experience with intuitive navigation and clear interfaces, offering administrators the ability to manage products, oversee transactions, and generate detailed reports, while staff can manage their profiles and daily data entries, and buyers can browse products, place orders, and review their histories.

The application supports simultaneous access and logins from multiple systems, ensuring robust performance and reliability. Its cross-browser compatibility guarantees a consistent user experience regardless of the preferred browser. Extensive testing and debugging have resulted in a stable system, minimizing operational risks. Adhering to specified restrictions, DSMS delivers predictable and efficient performance. Designed with scalability in mind, it can accommodate future enhancements and new features, making it a valuable tool for the evolving needs of the dairy industry.

In conclusion, DSMS is a robust, reliable, and user-friendly solution that effectively addresses the needs of all its users, ensuring efficient dairy operations management.

This document outlines the development and features of the Dairy Serve Management System, a comprehensive solution designed to manage various aspects of a dairy business. The complete source code for this project is available on GitHub: [GitHub Repository]

(https://github.com/Ravi191203/DAIRY-SERVE-MANAGEMENT-SYSTEM)

10. FUTURE ENHANCEMENTS

Every application has its own merits and demerits, and the Dairy Serve Management System (DSMS) is no exception. While the project has successfully covered almost all the current requirements, there is always room for further development and improvement. The structured and modular nature of the coding allows for easy modifications and the addition of new modules.

Future enhancements to DSMS could focus on making the application even more attractive and useful. This can include:

1. **Enhanced User Interface**: Improving the visual design and user experience to make the application more intuitive and appealing.
2. **Advanced Analytics**: Implementing advanced analytics and reporting features to provide deeper insights into data, benefiting both administrators and users.
3. **Mobile Compatibility**: Developing a mobile-friendly version of the application to ensure seamless access on smartphones and tablets.
4. **Expanded Functionality**: Adding new modules, such as advanced inventory management, customer relationship management (CRM), or expanded payment gateway options.
5. **Enhanced Security**: Continuously updating and enhancing security features to protect sensitive data and ensure compliance with industry standards.

These enhancements will make DSMS more interactive, efficient, and beneficial for all users, ensuring it remains a valuable tool for dairy management.

11. BIBLIOGRAPHY

The development of the Dairy Serve Management System (DSMS) has been supported by various references and resources. Here is a list of the materials and sources consulted during the project:

11.1 URL REFERENTIAL

11.1.1 Websites:

- W3Schools. (2024). PHP Tutorial. Retrieved from https://www.w3schools.com/php/
- Apache Friends. (2024). XAMPP Installation Guide. Retrieved from https://www.apachefriends.org/index.html
- Bootstrap. (2024). Introduction to Bootstrap. Retrieved from https://getbootstrap.com/docs/5.0/getting-started/introduction/
- Font Awesome. (2024). Get Started with Font Awesome. Retrieved from https://fontawesome.com/get-started
- WampServer. (2024). WampServer Installation Guide. Retrieved from http://www.wampserver.com/en/

11.1.2 Online Tutorials and Documentation:

- PHP Documentation. (2024). PHP Manual. Retrieved from https://www.php.net/docs.php
- FPDF Manual. (2024). FPDF: Free PDF Library. Retrieved from http://www.fpdf.org/

11.2 AI Assistance:

- OpenAI. (2024). ChatGPT: Your virtual assistant for project development and more. OpenAI. Retrieved from https://www.openai.com/chatgpt
- Gemini. (2024). Gemini: Enhance your project development experience with AI. Gemini Retrieved from https://www.gemini.google.com

This comprehensive list of resources has been instrumental in the successful development and implementation of the DSMS, providing valuable information, guidelines, and support throughout the project lifecycle.