

FACULTY OF COMPUTING AND INFORMATION MANAGEMENT

BACHELOR OF INFORMATION TECHNOLOGY

**LIVESTOCARE HUB SYSTEM**

**Name**: ***CLIFORD KARIMI MURIUKI***

**ADM.N.O: 20/04226**

**Course: BACHELOR OF SCIENCE (information Technology)**

**UNIT:** **BIT 3105 PROJECT PROGRAMMING**

**SUPERVISOR: JUSTUS MUTIA**

# SOFWARE DESIGN SPECIFICATION

Contents

[SOFWARE DESIGN SPECIFICATION 1](#_Toc161830145)

[Business rules 3](#_Toc161830146)

[Process modeling 4](#_Toc161830147)

[Identifying Symbols and Their Meaning 4](#_Toc161830148)

[Identifying Processes 4](#_Toc161830149)

[Context Diagram for LMS 4](#_Toc161830150)

[System (Single Process): 4](#_Toc161830151)

[External Entities: 4](#_Toc161830152)

[Data Flows: 5](#_Toc161830153)

[System Boundary: 5](#_Toc161830154)

[Interactions: 5](#_Toc161830155)

[Constructing DFDs 6](#_Toc161830156)

[Level 0 DFD: 6](#_Toc161830157)

[Level 1 DFDs: 6](#_Toc161830158)

[Level 2 DFDs: 6](#_Toc161830159)

[Level 3 DFDs: 6](#_Toc161830160)

[Key Considerations 6](#_Toc161830161)

[Data flow diagram 8](#_Toc161830162)

[Use Cases 8](#_Toc161830163)

[Record Livestock Data: 8](#_Toc161830164)

[Generate Reports: 8](#_Toc161830165)

[Manage Inventory: 8](#_Toc161830166)

[Authenticate Users: 8](#_Toc161830167)

[DFD Fragments 8](#_Toc161830168)

[Flowcharts of the use cases 10](#_Toc161830169)

[Data modeling (For database systems) 11](#_Toc161830170)

[Major Entities and Attributes 11](#_Toc161830171)

[Data Dictionary 11](#_Toc161830172)

[Entity Relationship Diagram (ERD) 12](#_Toc161830173)

[Pseudocodes 13](#_Toc161830174)

[Pseudocode for Adding New Health Observation 13](#_Toc161830175)

[Flowchart for Adding New Health Observation 14](#_Toc161830176)

[Web site structure (For web based systems) 15](#_Toc161830177)

# Business rules

In the context of the Livestock Management System (LMS) being developed, the organization's operations revolve around efficient and effective management of livestock-related activities. This includes tasks such as monitoring the health and well-being of livestock, tracking inventory of feed and supplies, analyzing performance metrics, and generating reports for decision-making. Farmers, livestock managers, and veterinarians are actively involved in recording livestock data, conducting health assessments, administering treatments, and maintaining inventory levels. The system boundary encompasses all aspects of livestock management within the organization, from data input and processing to reporting and analysis. By automating these operations and providing a centralized platform for information management, the LMS aims to streamline workflows, improve data accuracy, and enhance overall productivity in the livestock management processes.

# Process modeling

## Identifying Symbols and Their Meaning

* **Rectangle (Process):** Represents a process or action that transforms inputs into outputs.
* **Arrow (Flow):** Represents the flow of data or information between processes, data stores, and external entities.
* **Circle (Terminator):** Represents external entities interacting with the system, such as users or external systems.
* **Double Rectangle (Data Store):** Represents a data storage location, such as a database or file system.

## Identifying Processes

* **Record Livestock Data:** Process of capturing and inputting data related to livestock, including species, breed, health observations, and inventory updates.
* **Generate Reports:** Process of generating reports and analytics based on stored data, providing insights into livestock health, performance, and inventory status.
* **Manage Inventory:** Process of tracking and managing inventory items such as feed, medication, and supplies, ensuring adequate stock levels and usage tracking.
* **Authenticate Users:** Process of verifying user credentials and granting access based on assigned roles and permissions within the system.

# **Context Diagram for LMS**

The Context Diagram serves as an overview of the LMS's interaction with external entities, showcasing the system's boundary and primary interactions. Here are the key components highlighted in the Context Diagram:

### System (Single Process):

* + The System represents the Livestock Management System as a whole, encapsulating all internal processes, data flows, and interactions within the system boundary.

### External Entities:

* + **Users (Farmers, Livestock Managers, Veterinarians):**
    - These external entities interact with the LMS to perform various tasks such as recording livestock data, managing inventory, generating reports, and accessing system functionalities based on their roles and permissions.
  + **External Data Sources (Sensors, Weather APIs):**
    - These external entities provide real-time data inputs to the LMS, such as environmental conditions, livestock health indicators, and market data. The LMS processes and integrates this data for analysis and decision-making.

### Data Flows:

* + **Inputs from Users:**
    - Data flows from external entities (users) into the system, including livestock details, health observations, inventory updates, and report generation requests.
  + **Inputs from External Data Sources:**
    - Real-time data flows from external data sources (sensors, weather APIs) into the system, providing environmental data, livestock health metrics, and other relevant information.
  + **Outputs to Users:**
    - Processed data and generated reports flow from the system back to external entities (users), presenting insights, analytics, and actionable information based on user requests and system operations.

### System Boundary:

* + The System Boundary delineates the boundary between the Livestock Management System and its external environment, illustrating the scope of system interactions and data exchanges.
  + It highlights what is contained within the system (internal processes, data stores) and what interfaces with the system (external entities, data sources).

### Interactions:

* + The Context Diagram shows the primary interactions and data flows between the system (LMS) and its external entities (Users, External Data Sources), emphasizing the flow of information into and out of the system.

# Constructing DFDs

## Level 0 DFD:

* The Level 0 DFD provides an overview of the main processes and interactions within the LMS system boundary.
* It includes the primary processes such as Record Livestock Data, Generate Reports, Manage Inventory, and Authenticate Users.
* External entities (Users, External Data Sources) interact with the system through data flows, and data stores are utilized to store and retrieve information as needed.

## Level 1 DFDs:

* Level 1 DFDs break down each main process from Level 0 into more detailed sub processes and data flows.
* For example, the Record Livestock Data process may be decomposed into sub processes such as Input Livestock Details, Capture Health Observations, Update Inventory, and Store Data.
* Similarly, Generate Reports may include sub processes like Data Retrieval, Analytics Processing, Report Formatting, and Presentation.

## Level 2 DFDs:

* Level 2 DFDs further decompose the sub processes from Level 1 into specific tasks, data transformations, and interactions.
* For instance, Input Livestock Details may involve tasks such as Species Selection, Breed Selection, Age Input, Gender Input, and Weight Input.
* Data Retrieval for Reports may include tasks such as Querying Database, Aggregating Data, Applying Filters, and Sorting Data.

## Level 3 DFDs:

* Level 3 DFDs provide a highly detailed view of individual tasks, data elements, and data flows within each subprocess from Level 2.
* For example, the task of Querying Database in Data Retrieval for Reports may involve specific SQL queries, data extraction methods, and filtering criteria.
* Each data element and data flow is clearly labeled and defined, showcasing the granular level of system functionality and data processing.

### Key Considerations

* **Data Stores:** Data stores such as databases, files, and memory locations are depicted in DFDs to show where data is stored and accessed within the system.
* **Data Flows:** Data flows indicate the movement of data between processes, data stores, and external entities, highlighting the flow of information within the system.
* **Processes:** Processes represent actions or transformations that occur within the system, including data validation, calculations, and data manipulation tasks.
* **External Entities:** External entities (Users, External Data Sources) interact with the system through input/output data flows, influencing system behavior and operations.

# Data flow diagram

## Use Cases

### Record Livestock Data:

* + Farmer inputs details of a new livestock entry.
  + Livestock manager updates health observations.
  + Inventory manager adds or removes inventory items related to livestock.

### Generate Reports:

* + Farmer generates a health report for individual livestock.
  + Livestock manager generates productivity reports for a specific period.
  + Veterinarian generates treatment history reports for diagnosis.

### Manage Inventory:

* + Inventory manager adds new inventory items to the system.
  + Inventory manager updates stock levels based on usage.
  + System automatically alerts when inventory levels are low.

### Authenticate Users:

* + Users log in to the system using their credentials.
  + Admin manages user roles and permissions.
  + System logs user activities for audit purposes.

## DFD Fragments

1. **Record Livestock Data (Level 0 DFD):**
   * Input Livestock Details
   * Update Health Observations
   * Manage Inventory
2. **Generate Reports (Level 0 DFD):**
   * Generate Health Reports
   * Generate Productivity Reports
   * Generate Treatment History Reports
3. **Manage Inventory (Level 0 DFD):**
   * Add New Inventory Items
   * Update Stock Levels
   * Low Inventory Alerts
4. **Authenticate Users (Level 0 DFD):**
   * User Login
   * Manage User Roles
   * User Activity Logging

**Decomposition to Level 1 DFDs:**

1. **Record Livestock Data (Level 1 DFD):**
   * Input Livestock Details (Level 2 DFD)
   * Update Health Observations (Level 2 DFD)
   * Manage Inventory (Level 2 DFD)
2. **Generate Reports (Level 1 DFD):**
   * Generate Health Reports (Level 2 DFD)
   * Generate Productivity Reports (Level 2 DFD)
   * Generate Treatment History Reports (Level 2 DFD)
3. **Manage Inventory (Level 1 DFD):**
   * Add New Inventory Items (Level 2 DFD)
   * Update Stock Levels (Level 2 DFD)
   * Low Inventory Alerts (Level 2 DFD)
4. **Authenticate Users (Level 1 DFD):**
   * User Login (Level 2 DFD)
   * Manage User Roles (Level 2 DFD)
   * User Activity Logging (Level 2 DFD)

**Decomposition to Level 2 and Level 3 DFDs:**

* For each Level 1 DFD component, further decompose into specific tasks, data flows, and interactions as needed. Label each process, data store, data flow, and external entity clearly to ensure understanding and readability.

This structured approach helps in systematically capturing the system's functionalities, interactions, and data flows across different levels of detail, making it easier for stakeholders to comprehend and analyze the Livestock Management System's architecture and operations.

## Flowcharts of the use cases

Here's a structured flowchart for the "Record Livestock Data" use case, decomposed from Level 0 to Level 2:

# Data modeling (For database systems)

## Major Entities and Attributes

a. **Entities:**

* Livestock (e.g., cows, sheep, pigs)
* Health Observations
* Inventory Items
* Users (Farmers, Managers, Veterinarians)

b. **Attributes:**

* Livestock: ID, Species, Breed, Age, Gender, Weight
* Health Observations: ID, Livestock ID (foreign key), Temperature, Heart Rate, Condition
* Inventory Items: ID, Name, Quantity, Unit Price
* Users: ID, Name, Role, Username, Password

## Data Dictionary

A data dictionary is a centralized repository that contains detailed information about the data elements, data structures, data types, and data relationships within a database or information system. It serves as a reference guide for database administrators, developers, and users to understand the structure and meaning of the data stored in the system. Here's my data dictionary for the Livestock Management System:

**1. Livestock Table:**

* **ID:** Primary key uniquely identifying each livestock record.
* **Species:** Type of livestock species (e.g., cow, sheep, pig).
* **Breed:** Specific breed of the livestock.
* **Age:** Age of the livestock.
* **Gender:** Gender of the livestock (e.g., male, female).
* **Weight:** Weight of the livestock.

**2. Health Observations Table:**

* **ID:** Primary key uniquely identifying each health observation record.
* **Livestock\_ID:** Foreign key referencing the ID of the corresponding livestock.
* **Observation\_Date:** Date when the health observation was recorded.
* **Temperature:** Temperature of the livestock during the observation.
* **Heart\_Rate:** Heart rate of the livestock during the observation.
* **Condition:** Health condition or observations noted during the observation.

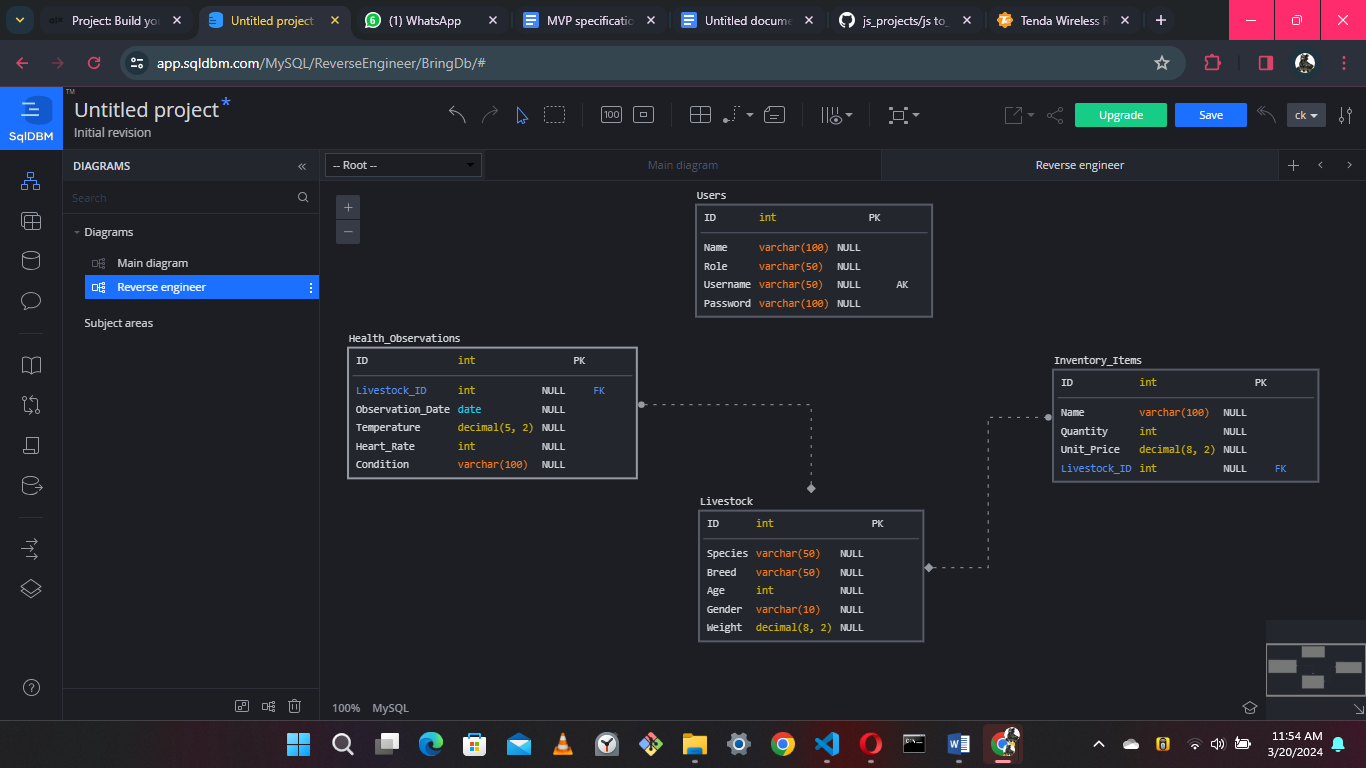
**3. Inventory Items Table:**

* **ID:** Primary key uniquely identifying each inventory item.
* **Name:** Name or description of the inventory item.
* **Quantity:** Quantity of the inventory item available.
* **Unit\_Price:** Price of the inventory item per unit.
* **Livestock\_ID:** Foreign key referencing the ID of the corresponding livestock (if applicable).

**4. Users Table:**

* **ID:** Primary key uniquely identifying each user.
* **Name:** Full name of the user.
* **Role:** Role or position of the user in the system (e.g., Farmer, Manager, Veterinarian).
* **Username:** Unique username used for authentication.
* **Password:** Encrypted password for user authentication.

## Entity Relationship Diagram (ERD)



## Pseudocodes

### Pseudocode for Adding New Health Observation

**Input:**

* Livestock ID (livestock\_id)
* Observation Date (observation\_date)
* Temperature (temperature)
* Heart Rate (heart\_rate) Condition (condition)

FUNCTION AddHealthObservation(livestock\_id, observation\_date, temperature, heart\_rate, condition)

// Validate input data

IF livestock\_id is not valid THEN

Display error message: "Invalid Livestock ID."

RETURN

END IF

IF observation\_date is not a valid date THEN

Display error message: "Invalid Observation Date."

RETURN

END IF

IF temperature is not a valid number THEN

Display error message: "Invalid Temperature."

RETURN

END IF

IF heart\_rate is not a valid number THEN

Display error message: "Invalid Heart Rate."

RETURN

END IF

// Insert new health observation into database

INSERT INTO Health\_Observations (Livestock\_ID, Observation\_Date, Temperature, Heart\_Rate, Condition)

VALUES (livestock\_id, observation\_date, temperature, heart\_rate, condition)

// Display success message

Display message: "Health Observation added successfully for Livestock ID: <livestock\_id>."

END FUNCTION

**// Example usage of AddHealthObservation function**

livestock\_id = 123

observation\_date = "2024-02-10"

Temperature = 38.5

heart\_rate = 80

Condition = "Stable"

AddHealthObservation(livestock\_id, observation\_date, temperature, heart\_rate, condition)

### Flowchart for Adding New Health Observation

# Web site structure (For web based systems)