
Software Requirements and Design Document

for

FARMSUITE

Prepared by Salar i200830 Nouman 210416 Sarim i221280

Section: C

FAST UNIVERSITY

25-11-24

Table of Contents

| | |
|--|-----------|
| Table of Contents | ii |
| 1. Introduction..... | 1 |
| 1.1 Purpose | 1 |
| 1.2 Product Scope..... | 1 |
| 1.3 Title | 1 |
| 1.4 Objectives..... | 1 |
| 1.5 Problem Statement..... | 1 |
| 2. Overall Description | 2 |
| 2.1 Product Perspective | 2 |
| 2.2 Product Functions..... | 2 |
| 2.3 List of Use Cases..... | 3 |
| 2.4 Extended Use Cases | 4 |
| 2.5 Use Case Diagram | 5 |
| 3. Other Nonfunctional Requirements | 6 |
| 3.1 Performance Requirements | 6 |
| 3.2 Safety Requirements..... | 6 |
| 3.3 Security Requirements | 6 |
| 3.4 Software Quality Attributes | 6 |
| 3.5 Business Rules..... | 6 |
| 3.6 Operating Environment | 6 |
| 3.7 User Interfaces..... | 7 |
| 4. Domain Model | 7 |
| 5. System Sequence Diagram | 8 |
| 6. Sequence Diagram | 9 |
| 7. Class Diagram | 10 |
| 8. Component Diagram..... | 11 |
| 9. Package Diagram | 12 |
| 10. Deployment Diagram..... | 13 |

1. Introduction

1.1 Purpose

Our application aims to provide a digital interface to the act of managing a dairy farm, it covers communication, records, reports, sales, and more.

1.2 Product Scope

Our application, FarmSuite, will be focused on all the dairy farms scattered all around Pakistan. Our application would not be focused on connecting these farms together, but in fact improving the managerial ability of each individual farm. While some applications related to farming do exist, none exist that are centered around dairy farms.

1.3 Title

FARMSUITE

1.4 Objectives

1. An efficient system for scheduling and managing tasks around a dairy farm.
2. An interface to sell produce at a local shop.
3. An interface to communicate with vets, owners, fodder providers.
4. A record system to show sales reports.
5. A record system to show production reports.
6. Improve time and resource management and reduce labor costs.

1.5 Problem Statement

Across Pakistan, from Sindh to Kashmir, many dairy farms operate with unique regional features. Despite these differences, the essential procedures of dairy farming are remarkably similar. Managing livestock requires urgency, organization, and accurate areas where our application aims to help. Our application streamlines farm processes and facilitates direct communication between farmers, owners, and shopkeepers. It helps farm owners understand their operations better, allowing them to analyze costs, resources, and responsibilities. This enables them to achieve their goals more effectively and maximize their farm's potential.

2. Overall Description

2.1 Product Perspective

FarmSuite is a self-contained software application designed to manage various farm operations, including task assignments, production tracking, sales management, and reporting. It is a new solution for agricultural management and is not a replacement for any existing systems. The software supports different user roles (Owner, Farm Manager, Worker, Shopkeeper, Customer) with role-specific functionalities. It integrates with external systems like veterinary services and payment gateways and includes components for user management, task tracking, sales, health monitoring, and financial reporting.

2.2 Product Functions

The FarmSuite application provides the following major functionalities:

- **User Authentication and Role Management:**
 - Users can log in with different roles (Owner, Farm Manager, Farm Worker, Shopkeeper, Customer).
 - Role-based access ensures that each user has the appropriate permissions for their duties.
- **Task and Production Management:**
 - Assign tasks to farm workers.
 - Update milk production per cow.
 - Manage tasks and their statuses.
- **Sales Management:**
 - Perform sales transactions.
 - Built-in calculator to calculate order bills.
 - Report on sales performance.
- **Health and Activity Monitoring:**
 - Monitor farm activity using a live camera feed (FarmCam).
 - Track livestock health records and schedule veterinary visits.
- **Reporting:**
 - Generate reports on cow performances.
 - Generate reports on sales performances.
 - Report issues to the owner.
- **Inventory and Fodder Management:**
 - Order and manage fodder for the livestock.

- Contact the nursery for waste disposal.
- **Financial Management:**
 - Track farm expenses.
 - Verify credit card details for payment processing.

2.3 List of Use Cases

By Salar Shoaib (20I-0830):

1. Assign tasks
2. Update Milk production per cow
3. Perform sales
4. Get a report on cow performances
5. Get a report on sale performances

By Sarim Rasheed (22I-1280):

6. Built in Calculator to calculate order bill (this is an include use case, so it will be system sub goal in the level field) - by
7. Update tasks and statuses
8. Report issues to owner
9. Send a message to Vet
10. Order and manage fodder

By Nouman Hafeez (21I-0416)

11. Waste disposal contact with nursery
12. Monitor Farm Activity via Live Camera Feed - FarmCam
13. Schedule Veterinary Visits
14. Track Livestock Health Records
15. Manage Farm Expenses
16. Credit card verification

Others:

1. Log in: Owner, Farm Manager, Farm Worker, Shopkeeper, Customer

2.4 Extended Use Cases

| SALAR SHOAIB | | |
|-----------------------------|---|--|
| Use Case Name: | Assign Tasks | |
| Scope: | Farmsuite Application | |
| Level: | User Goal | |
| Primary Actor: | Farm Manager | |
| Stakeholders and Interests: | Farm Owner | |
| Pre-conditions: | Farm Manager or Farm Owner are logged in | |
| Post-conditions: | Tasks have been updated and are available to be fulfilled in the Main Task List | |
| Main Success Scenario: | Actor Action: | System Response: |
| | <ol style="list-style-type: none"> 1. User clicks the "Add Task" Button, and selects the person they are assigning the task to. 2. User writes out the task or selects one of the predefined routine tasks. 3. User submits the task. <p style="text-align: right;">7. User receives a message that their task has been added.</p> | <ol style="list-style-type: none"> 4. System accepts the task and adds it to the Main Task List. 5. System Updates the tasks to the Personal Task Lists of the selected person. 6. System sends a confirmation message to the user. |
| Extensions: | 2.1. User adds specifics to one of the routine tasks. | |

SALAR SHOAIB

| | | |
|-----------------------------|--|---|
| Use Case Name: | Perform Sales | |
| Scope: | Farmsuite Application | |
| Level: | User Goal | |
| Primary Actor: | Customer | |
| Stakeholders and Interests: | Farm Manager, Farm Owner, Shopkeeper | |
| Pre-conditions: | Customers do not need to log in, but the shopkeeper logs in to access the sales system. | |
| Post-conditions: | Purchase quantities have been updated. | |
| Main Success Scenario: | Actor Action: | System Response: |
| | <ol style="list-style-type: none"> 1. Customer selects the item(s) they want. 2. Customer enters the quantities and submits to the system. 4. The customer confirms the order. And enters payment information. 6. User receives a receipt. | <ol style="list-style-type: none"> 3. The system receives values and displays the payment total. 5. The system confirms the payment and updates quantities of products and sends a message to the customer. |
| Extensions: | <p>2.1. Customer enters a value of product that is not available and hence the value is rejected and customer is asked to reenter.</p> <p>4. 1. Customer changes values after seeing the total amount, and then goes to step 3 again.</p> | |

SALAR SHOAIB

| | | |
|-----------------------------|--|--|
| Use Case Name: | Generate Cow Performance Report | |
| Scope: | Farmsuite Application | |
| Level: | User Goal | |
| Primary Actor: | Farm Owner | |
| Stakeholders and Interests: | Farm Manager | |
| Pre-conditions: | Farm Owner is logged in, and at least 30 days have passed in sales and production. | |
| Post-conditions: | A report has been generated. | |
| Main Success Scenario: | Actor Action: | System Response: |
| | <ol style="list-style-type: none"> 1. User clicks the “Generate Report” Button. 2. The User selects a time slot/ month for the report generation. 4. User receives a graph displaying trends and also other useful information. | <ol style="list-style-type: none"> 3. System generates a report and displays it on the user portal. |
| Extensions: | <p>2.1. System warns users that the time slot is incorrect or that the generation threshold (30 days) has not been met.</p> | |

SALAR SHOAIB

| | | |
|-----------------------------|--|---|
| Use Case Name: | Generate Sales Performance Report | |
| Scope: | Farmsuite Application | |
| Level: | User Goal | |
| Primary Actor: | Farm Owner | |
| Stakeholders and Interests: | Shopkeeper | |
| Pre-conditions: | Farm Owner is logged in, and at least 30 days have passed in sales and production. | |
| Post-conditions: | A report has been generated. | |
| Main Success Scenario: | Actor Action: 1. User clicks the “Generate Report” Button. 2. The User selects a time slot/ month for the report generation. 4. User receives a graph displaying trends and also other useful information. | System Response: 3. System generates a report and displays it on the user portal. |
| Extensions: | 2.1. System warns users that the time slot is incorrect or that the generation threshold (30 days) has not been met. | |

SARIM RASHEED

| | | |
|-----------------------------|---|--|
| Use Case Name: | Calculate Order Bill Using Built-in Calculator | |
| Scope: | FarmSuite Application | |
| Level: | Sub-function | |
| Primary Actor: | Farm Owner | |
| Stakeholders and Interests: | <ul style="list-style-type: none"> • Farm Owner: Wants accurate cost calculations for budgeting and financial planning. • Farm Manager: Needs to quickly calculate order costs to make informed purchasing decisions. • Accounting Staff: Interested in ensuring that all costs are tracked accurately for financial records. • Suppliers: Want clear communication regarding the costs associated with their products. | |
| Pre-conditions: | <ul style="list-style-type: none"> • The farm owner is logged into the system. • The order details (items and quantities) are available | |
| Post-conditions: | <ul style="list-style-type: none"> • The total bill is calculated and displayed. • The order is ready for finalization and payment. | |
| Main Success Scenario: | Actor Action: 1. user selects the "Calculate Bill" option for the order. 3. user calculates additional charges (e.g., taxes, shipping) and adds them to the subtotal. 4. user clicks the "Calculate Bill" button. | System Response: 2. System displays the order details, including item quantities, prices, and subtotal. 5. System displays the final total, including any additional charges. |
| Extensions: | 5a. If the calculation is unsuccessful, the system alerts the user to the error and allows them to re-enter the values for a new calculation. | |

SARIM RASHEED

| | | |
|-----------------------------|--|--|
| Use Case Name: | Update Tasks and Statuses | |
| Scope: | FarmSuite Application | |
| Level: | FarmSuite Application | |
| Primary Actor: | Farm Manager | |
| Stakeholders and Interests: | <ul style="list-style-type: none"> ● Farm Manager/Owner: Wants an efficient way to manage daily tasks. ● Employees: Need clear instructions on daily activities. ● Vets/Fodder Providers: Require updates on farm needs. | |
| Pre-conditions: | <ul style="list-style-type: none"> ● The user must be logged into the FarmSuite application. ● The user must have the necessary permissions to update tasks. | |
| Post-conditions: | <ul style="list-style-type: none"> ● The task statuses are updated and saved in the system. ● All relevant stakeholders are notified of the updates. | |
| Main Success Scenario: | Actor Action: | System Response: |
| | <ol style="list-style-type: none"> 1. The user selects 'Update Tasks' from the menu. 3. User selects a task to update. 5. User updates status/details as needed and submits changes. | <ol style="list-style-type: none"> 2. System displays the current task list with statuses. 4. System shows task details with options to change status or modify task information. 6. System confirms the update and notifies relevant stakeholders. |
| Extensions: | <ul style="list-style-type: none"> ● 2a: If a task is overdue, the system alerts the manager and suggests priority reassignment ● 5a: If there is no internet connection, the system stores updates locally until a connection is re-established, then syncs data. | |

SARIM RASHEED

| | | |
|-----------------------------|---|---|
| Use Case Name: | Report Issue to Owner | |
| Scope: | FarmSuite Application | |
| Level: | User Goal | |
| Primary Actor: | Farm Worker | |
| Stakeholders and Interests: | <ul style="list-style-type: none"> • Farm Workers: Want to quickly and efficiently report issues they encounter. • Farm Owners: Need to be informed about issues to take timely action. • Farm Managers: Want to ensure smooth operations by addressing issues promptly. • Vets: May need to be informed about health-related issues. | |
| Pre-conditions: | <ul style="list-style-type: none"> • The user must be logged into the FarmSuite application. • The user must have the necessary permissions to report issues. | |
| Post-conditions: | <ul style="list-style-type: none"> • The reported issue is logged in the system. • The owner is notified of the reported issue. • The issue is tracked until it is resolved. | |
| Main Success Scenario: | Actor Action: | System Response: |
| | <ol style="list-style-type: none"> 1. The user selects the "Report Issue" option.. 3. The user enters issue description, severity level, and any relevant attachments. | <ol style="list-style-type: none"> 2. System presents a form for entering issue details. 4. System validates the input and submits the issue report to the owner. 5. System sends a notification to the owner informing them of the new issue. |
| Extensions: | <ul style="list-style-type: none"> • 3a: If data is incomplete or missing, the system prompts the user to fill in the mandatory fields. • 5a: If the notification isn't received, the system automatically retries using alternative methods such as SMS or email.g | |

SARIM RASHEED

| | | |
|-----------------------------|--|---|
| Use Case Name: | Send a Message to Vet | |
| Scope: | FarmSuite Application | |
| Level: | User Goal level | |
| Primary Actor: | Farm Owner or Farm Worker | |
| Stakeholders and Interests: | <ul style="list-style-type: none"> • Farm Owners/Workers: Need quick, efficient communication with vets about livestock health. • Vets: Require timely, accurate info to provide proper advice/services. • Farm Managers: Ensure prompt action on health issues to maintain livestock well-being. | |
| Pre-conditions: | <ul style="list-style-type: none"> • Farm Manager has logged into the system. • A veterinarian has been added to the contact list. | |
| Post-conditions: | Message is sent to the veterinarian and he is notified. | |
| Main Success Scenario: | Actor Action: 1. The user selects the "Send Message" option. 3. The user selects the veterinarian from the contact list, enters the message content, and attaches any relevant files. 5. The user receives confirmation that the message has been sent. | System Response: 2. System presents a form for entering message details. 4. System validates the input and sends the message to the veterinarian. 6. The system notifies the vet of the new message via email or in-app notification. |
| Extensions: | <ul style="list-style-type: none"> • 3a: If the contact list is empty or unresponsive, the system prompts the user to refresh or add a new contact. • 4a: If message delivery fails, the system retries sending or notifies the user of the failure. | |

SARIM RASHEED

| | | |
|-----------------------------|---|---|
| Use Case Name: | Order and Manage Fodder | |
| Scope: | FarmSuite Application | |
| Level: | User Goal Level | |
| Primary Actor: | Farm Manager | |
| Stakeholders and Interests: | <ul style="list-style-type: none"> • farm manager: Needs to ensure a steady supply of fodder for livestock. • Fodder suppliers: Want to receive orders and manage deliveries efficiently. • Farm workers: Need to be informed about fodder availability and storage. | |
| Pre-conditions: | <ul style="list-style-type: none"> • Farm Manager has logged into the system. • A fodder provider has been added to the contact list. | |
| Post-conditions: | <ul style="list-style-type: none"> • Fodder is ordered and delivery is scheduled. • Inventory is updated with the new fodder order. | |
| Main Success Scenario: | Actor Action: | System Response: |
| | <ol style="list-style-type: none"> 1. The user selects the "Order Fodder" option. 3. The user selects the fodder type and quantity, chooses the provider, and specifies the delivery date. 5. The user reviews and confirms the order. | <ol style="list-style-type: none"> 2. System presents a form for entering order details. 4. System calculates the total cost and generates an order. 6. System sends the order to the provider and updates the fodder inventory. |
| Extensions: | <ul style="list-style-type: none"> • 3a: If the selected fodder type is unavailable, the system prompts the user to choose from alternative options or suggests restock dates. • 4a: If the calculation fails, the system alerts the user and provides a manual input option for the cost. • 6a: If the order fails to send, the system retries the transmission or allows the user to contact the provider directly for confirmation. | |

SARIM RASHEED

| | | |
|-----------------------------|---|--|
| Use Case Name: | Order and Manage Fodder | |
| Scope: | FarmSuite Application | |
| Level: | User Goal Level | |
| Primary Actor: | Farm Manager | |
| Stakeholders and Interests: | <ul style="list-style-type: none"> • farm manager: Needs to ensure a steady supply of fodder for livestock. • Fodder suppliers: Want to receive orders and manage deliveries efficiently. • Farm workers: Need to be informed about fodder availability and storage. | |
| Pre-conditions: | <ul style="list-style-type: none"> • Farm Manager has logged into the system. • A fodder provider has been added to the contact list. | |
| Post-conditions: | <ul style="list-style-type: none"> • Fodder is ordered and delivery is scheduled. • Inventory is updated with the new fodder order. | |
| Main Success Scenario: | Actor Action: <ol style="list-style-type: none"> 1. The user selects the "Order Fodder" option. 3. The user selects the fodder type and quantity, chooses the provider, and specifies the delivery date. 5. The user reviews and confirms the order. | System Response: <ol style="list-style-type: none"> 2. System presents a form for entering order details. 4. System calculates the total cost and generates an order. 6. System sends the order to the provider and updates the fodder inventory. |
| Extensions: | <ul style="list-style-type: none"> • 3a: If the selected fodder type is unavailable, the system prompts the user to choose from alternative options or suggests restock dates. • 4a: If the calculation fails, the system alerts the user and provides a manual input option for the cost. • 6a: If the order fails to send, the system retries the transmission or allows the user to contact the provider directly for confirmation. | |

NOUMAN HAFEEZ

| | | |
|-----------------------------|---|--|
| Use Case Name: | Waste Disposal Contact with Nursery | |
| Scope: | Farmsuite Application | |
| Level: | User Goal | |
| Primary Actor: | Farm Manager | |
| Stakeholders and Interests: | Farm Owner, Nursery, Environment Authorities | |
| Pre-conditions: | <ul style="list-style-type: none"> • Farm Manager is logged in. • Nursery contact information is available in the system. | |
| Post-conditions: | <ul style="list-style-type: none"> • Waste disposal request has been sent to the nursery. • Confirmation of request received. | |
| Main Success Scenario: | Actor Action: 1. User selects the "Request Waste Disposal" option. 3. User enters waste type, quantity, and preferred pickup time. 5. User submits the request. 7. User receives a confirmation message regarding the request. | System Response: 2. System presents a form for entering waste details. 4. System validates the input and prepares the request. 6. System sends the request to the nursery and confirms it. |
| Extensions: | 3a. If the input is incomplete, the system prompts the user to complete the form. 5a. If the request fails to send, the system retries or prompts the user to check the nursery contact. | |

NOUMAN HAFEEZ

| | | |
|-----------------------------|--|--|
| Use Case Name: | Monitor Farm Activity via Live Camera Feed - FarmCam | |
| Scope: | Farmsuite Application | |
| Level: | User Goal | |
| Primary Actor: | Farm Manager | |
| Stakeholders and Interests: | Farm Owner, Nursery, Security Authorities | |
| Pre-conditions: | <ul style="list-style-type: none"> • Farm Manager is logged in. • Cameras are installed and functioning. | |
| Post-conditions: | <ul style="list-style-type: none"> • Live camera feed is displayed. • User can monitor activities in real-time. | |
| Main Success Scenario: | Actor Action: 1. User selects the "Live Camera Feed" option. 3. User selects a specific camera to view. 5. User can toggle between different camera feeds. | System Response: 2. System connects to the camera and retrieves the feed. 4. System displays the live feed from the selected camera. 6. System updates the display with the selected feed. |
| Extensions: | 2a. If the camera feed fails to load, the system displays an error message and retries. 5a. If a camera is offline, the system notifies the user and suggests checking the camera's status. | |

NOUMAN HAFEEZ

| | | |
|-----------------------------|--|--|
| Use Case Name: | Schedule Veterinary Visits | |
| Scope: | Farmsuite Application | |
| Level: | User Goal | |
| Primary Actor: | Farm Manager | |
| Stakeholders and Interests: | Farm Owner, Veterinarian, Farm Workers | |
| Pre-conditions: | <ul style="list-style-type: none"> • Farm Manager is logged in. • Veterinarian contact information is available. | |
| Post-conditions: | <ul style="list-style-type: none"> • Veterinary appointment is scheduled. • Notifications sent to relevant stakeholders. | |
| Main Success Scenario: | Actor Action: 1. User selects the "Schedule Veterinary Visit" option. 3. User selects a date and time for the visit. 5. User confirms the appointment. 7. User receives a confirmation message about the visit. | System Response: 2. System presents a calendar for selecting dates. 4. System checks veterinarian availability for the selected time. 6. System schedules the visit and sends notifications to the vet and farm workers. |
| Extensions: | 3a. If the selected date/time is unavailable, the system suggests alternative slots. 5a. If the appointment cannot be confirmed, the system notifies the user of the issue. | |

NOUMAN HAFEEZ

| | | |
|-----------------------------|--|---|
| Use Case Name: | Track Livestock Health Records | |
| Scope: | Farmsuite Application | |
| Level: | User Goal | |
| Primary Actor: | Farm Manager | |
| Stakeholders and Interests: | Farm Owner, Veterinarians, Farm Workers | |
| Pre-conditions: | <ul style="list-style-type: none"> • Farm Manager is logged in. • Health records are available in the system. | |
| Post-conditions: | <ul style="list-style-type: none"> • Health records are retrieved and displayed. | |
| Main Success Scenario: | Actor Action: 1. User selects the "View Livestock Health Records" option. 3. User selects a specific animal to view details. 5. User can add or update health records as needed. | System Response: 2. System displays a list of livestock with health records. 4. System displays detailed health records for the selected animal. 6. System saves the changes and confirms the update. |
| Extensions: | 2a. If no records exist for the selected animal, the system notifies the user. 5a. If the update fails, the system alerts the user and allows retrying. | |

NOUMAN HAFEEZ

| | | |
|-----------------------------|---|---|
| Use Case Name: | Track Livestock Health Records | |
| Scope: | Farmsuite Application | |
| Level: | User Goal | |
| Primary Actor: | Farm Owner | |
| Stakeholders and Interests: | Farm Manager, Accounting Staff, Farm Workers | |
| Pre-conditions: | <ul style="list-style-type: none"> • Farm Owner is logged in. • Expense categories are set up in the system. | |
| Post-conditions: | <ul style="list-style-type: none"> • Expenses are updated and saved in the system. • Reports generated for analysis. | |
| Main Success Scenario: | Actor Action: | System Response: |
| | 1. User selects the "Manage Expenses" option. 3. User enters the expense category, amount, and date. 5. User submits the expense. 7. User can generate a report of expenses if needed. | 2. System presents a form for entering expense details. 4. System validates the input and prepares to save. 6. System updates the expense record and confirms the entry. 8. System displays the report based on the recorded expenses. |
| Extensions: | 3a. If the input is invalid (e.g., negative amount), the system prompts for correction. 7a. If report generation fails, the system alerts the user and suggests retrying. | |

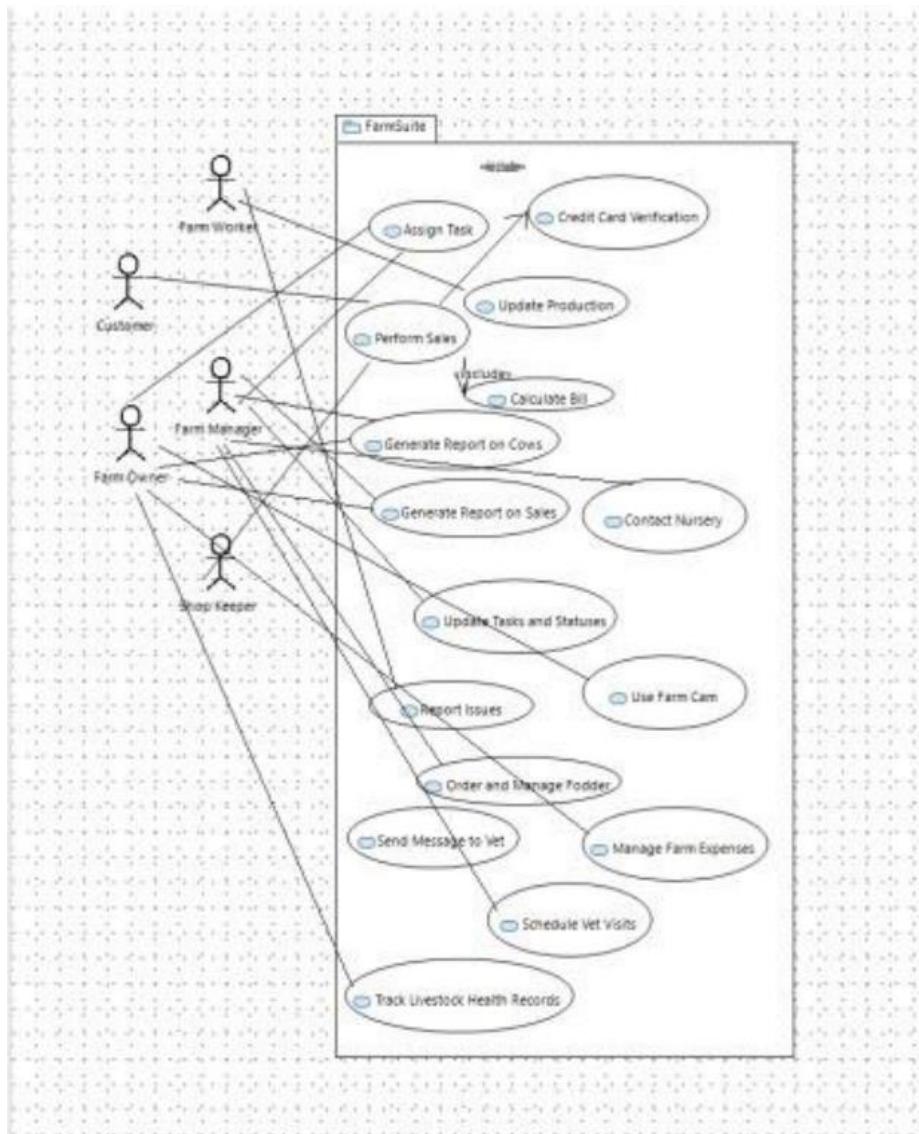
NOUMAN HAFEEZ

| | | |
|-----------------------------|---|---|
| Use Case Name: | Credit Card Verification | |
| Scope: | Farmsuite Application | |
| Level: | Sub-function | |
| Primary Actor: | Farm Owner | |
| Stakeholders and Interests: | Farm Manager, Accounting Staff, Customers | |
| Pre-conditions: | <ul style="list-style-type: none"> • Farm Owner is logged in. • Credit card details are entered for verification. | |
| Post-conditions: | <ul style="list-style-type: none"> • Credit card is verified or rejected. • Appropriate actions are taken based on verification results. | |
| Main Success Scenario: | <p>Actor Action:</p> <p>1. User selects the "Verify Credit Card" option.</p> <p>3. User enters credit card number, expiration date, and CVV.</p> <p>5. User receives a confirmation of verification status.</p> | <p>System Response:</p> <p>2. System prompts for credit card details.</p> <p>4. System processes the verification request.</p> <p>6. System updates the transaction status accordingly.</p> |
| Extensions: | <p>3a. If the credit card details are invalid, the system alerts the user to re-enter information.</p> <p>5a. If the verification fails, the system provides reasons for the failure and suggests corrective actions.</p> | |

SALAR SHOAIB

| | | |
|-----------------------------|--|---|
| Use Case Name: | Update Daily Production(Per Cow) | |
| Scope: | Farmsuite Application | |
| Level: | User Goal | |
| Primary Actor: | Farm Worker | |
| Stakeholders and Interests: | Farm Manager, Farm Owner | |
| Pre-conditions: | Farm Worker is logged in. New day has begun, so all cow status has been reset. | |
| Post-conditions: | All cow's statuses have been updated. | |
| Main Success Scenario: | Actor Action: 1. Farm worker selects a cow. 2. Farm worker selects the option to milk the cow and puts in the amount of milk produced into the system and submits. 3. Worker receives a confirmation message, and the cow's status is automatically updated. | System Response: 4. System accepts the value and checks if it's in the normal range. 5. System updates shop with the milk amount produced. 6. System sends a success message to the worker. |
| Extensions: | 4.1. Milk amount entered is in the abnormal range. <ul style="list-style-type: none"> . 4.1.1. If it is extremely below the avg, the vet is called. 4.1.2 If it is extremely above the avg, the amount is rejected from being entered into the system. | |

2.5 Use Case Diagram



3. Other Nonfunctional Requirements

3.1 Performance Requirements

<If there are performance requirements for the product under various circumstances, state them here and explain their rationale, to help the developers understand the intent and make suitable design choices. Specify the timing relationships for real time systems. Make such requirements as specific as possible. You may need to state performance requirements for individual functional requirements or features.>

3.2 Safety Requirements

<Specify those requirements that are concerned with possible loss, damage, or harm that could result from the use of the product. Define any safeguards or actions that must be taken, as well as actions that must be prevented. Refer to any external policies or regulations that state safety issues that affect the product's design or use. Define any safety certifications that must be satisfied.>

3.3 Security Requirements

<Specify any requirements regarding security or privacy issues surrounding use of the product or protection of the data used or created by the product. Define any user identity authentication requirements. Refer to any external policies or regulations containing security issues that affect the product. Define any security or privacy certifications that must be satisfied.>

3.4 Software Quality Attributes

<Specify any additional quality characteristics for the product that will be important to either the customers or the developers. Some to consider are: adaptability, availability, correctness, flexibility, interoperability, maintainability, portability, reliability, reusability, robustness, testability, and usability. Write these to be specific, quantitative, and verifiable when possible. At the least, clarify the relative preferences for various attributes, such as ease of use over ease of learning.>

3.5 Business Rules

<List any operating principles about the product, such as which individuals or roles can perform which functions under specific circumstances. These are not functional requirements in themselves, but they may imply certain functional requirements to enforce the rules.>

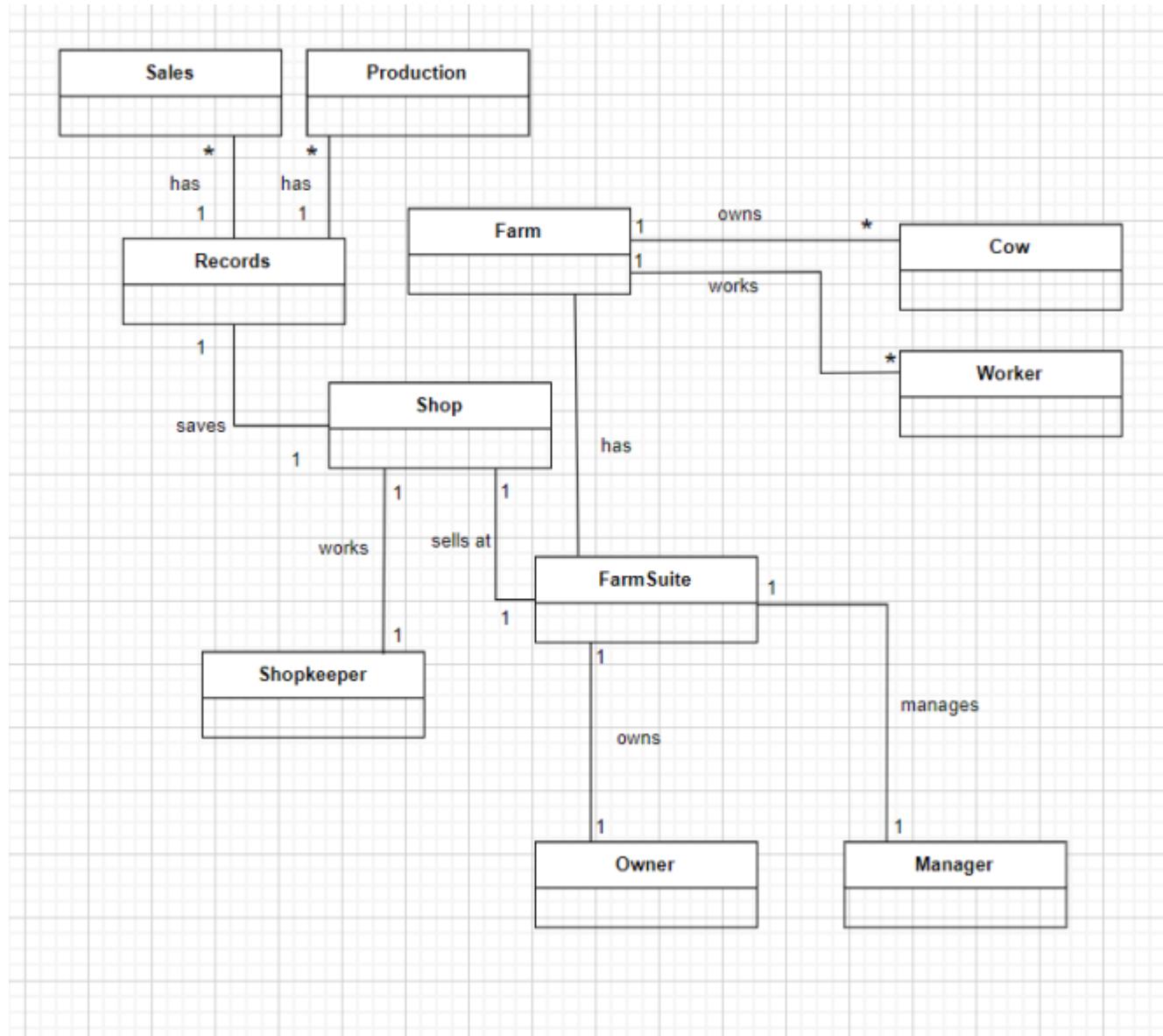
3.6 Operating Environment

<Describe the environment in which the software will operate, including the hardware platform, operating system and versions, and any other software components or applications with which it must peacefully coexist.>

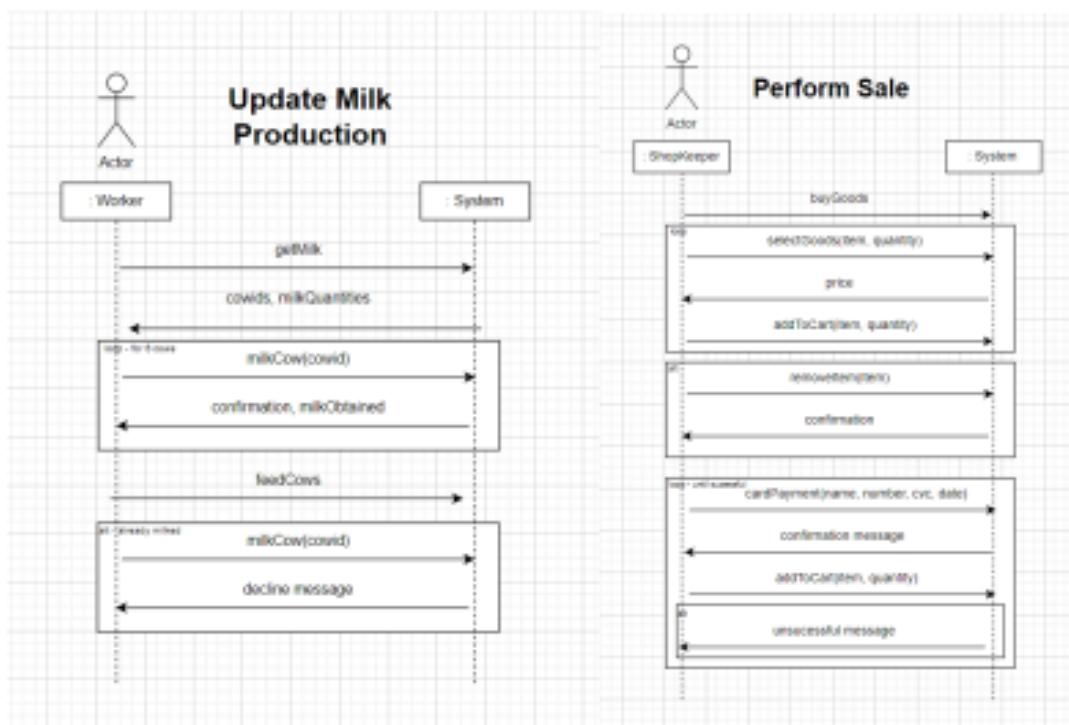
3.7 User Interfaces

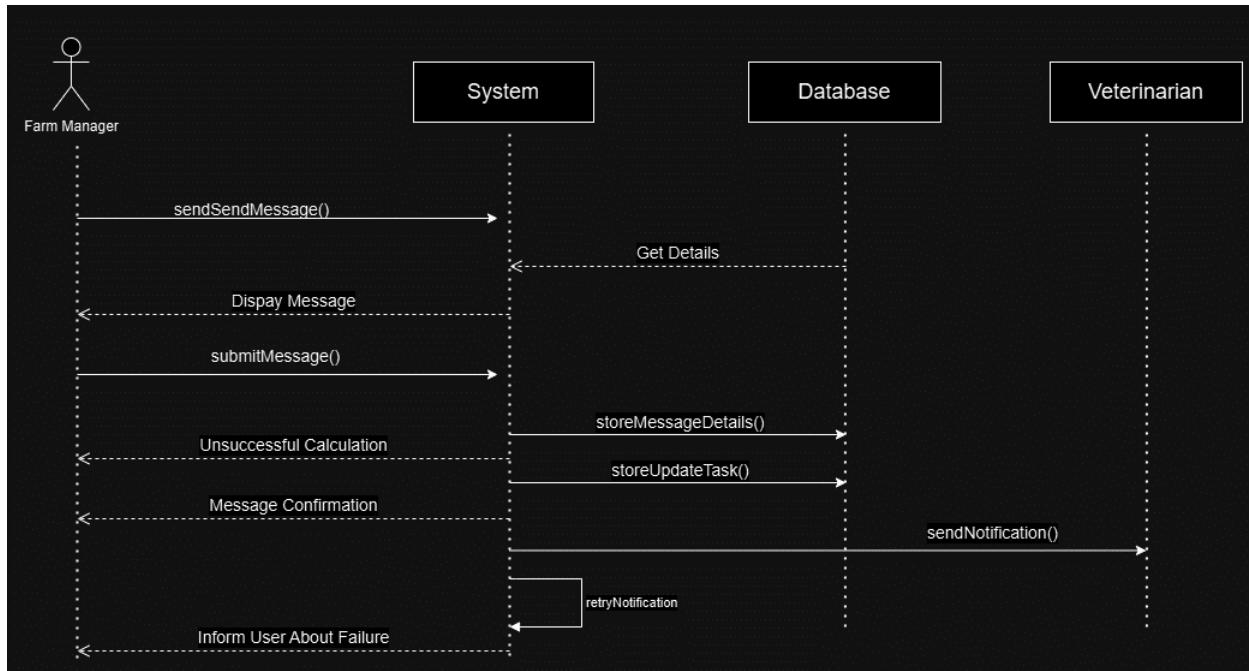
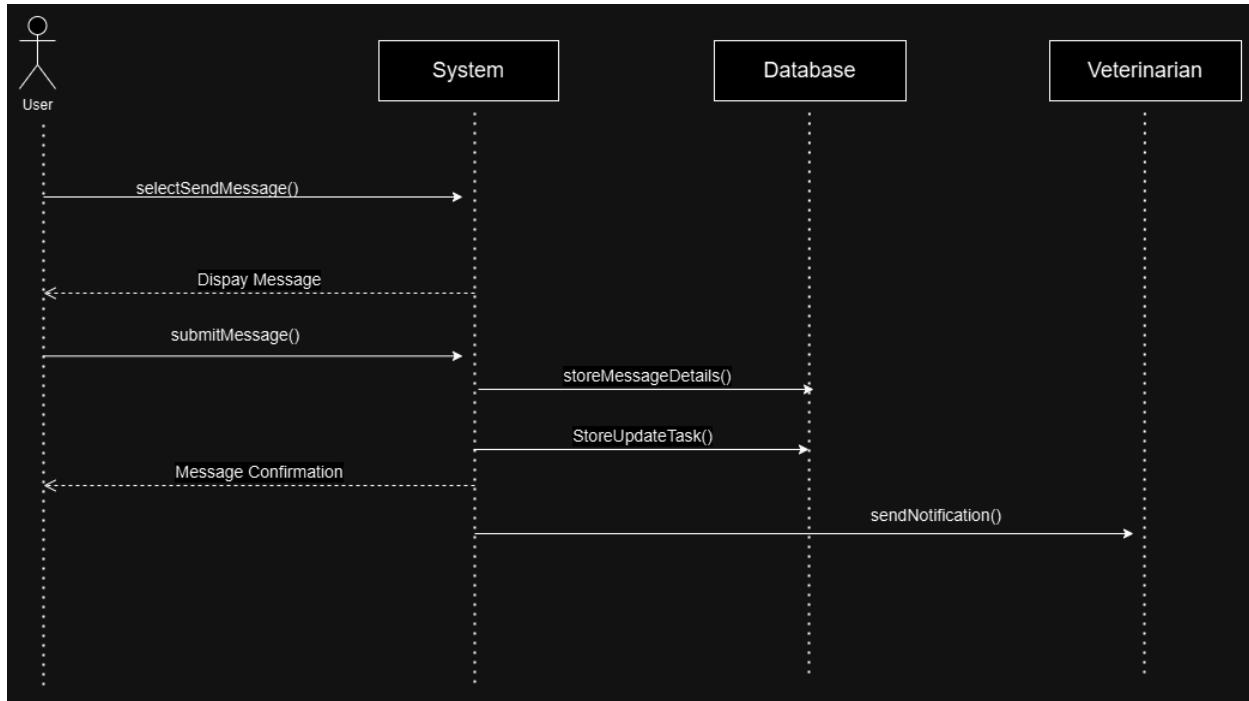
<Describe the logical characteristics of each interface between the software product and the users. This may include sample screen images, any GUI standards or product family style guides that are to be followed, screen layout constraints, standard buttons and functions (e.g., help) that will appear on every screen, keyboard shortcuts, error message display standards, and so on. Define the software components for which a user interface is needed. Details of the user interface design should be documented in a separate user interface specification.>

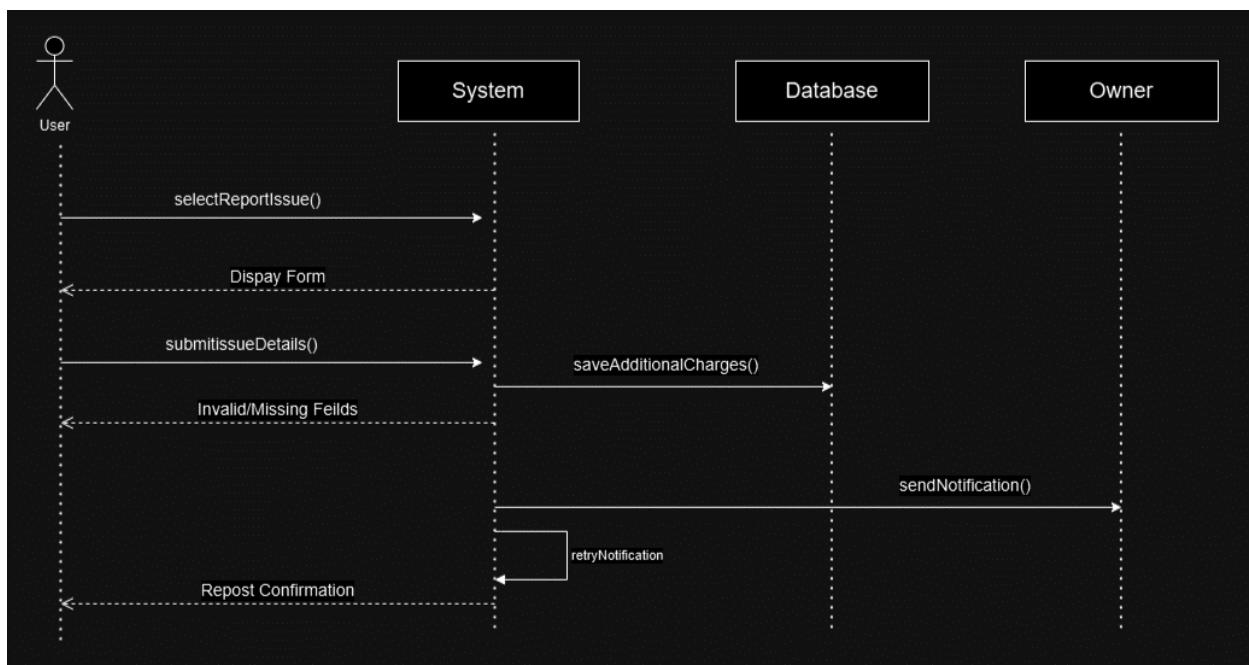
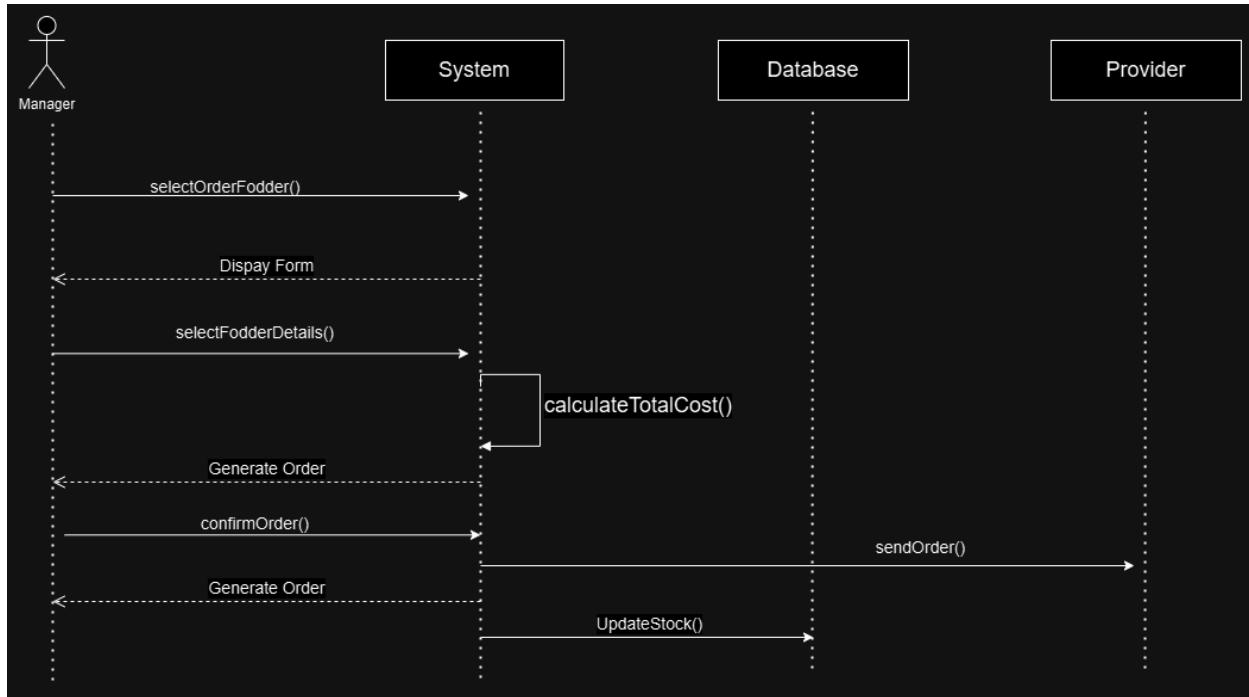
4. Domain Model



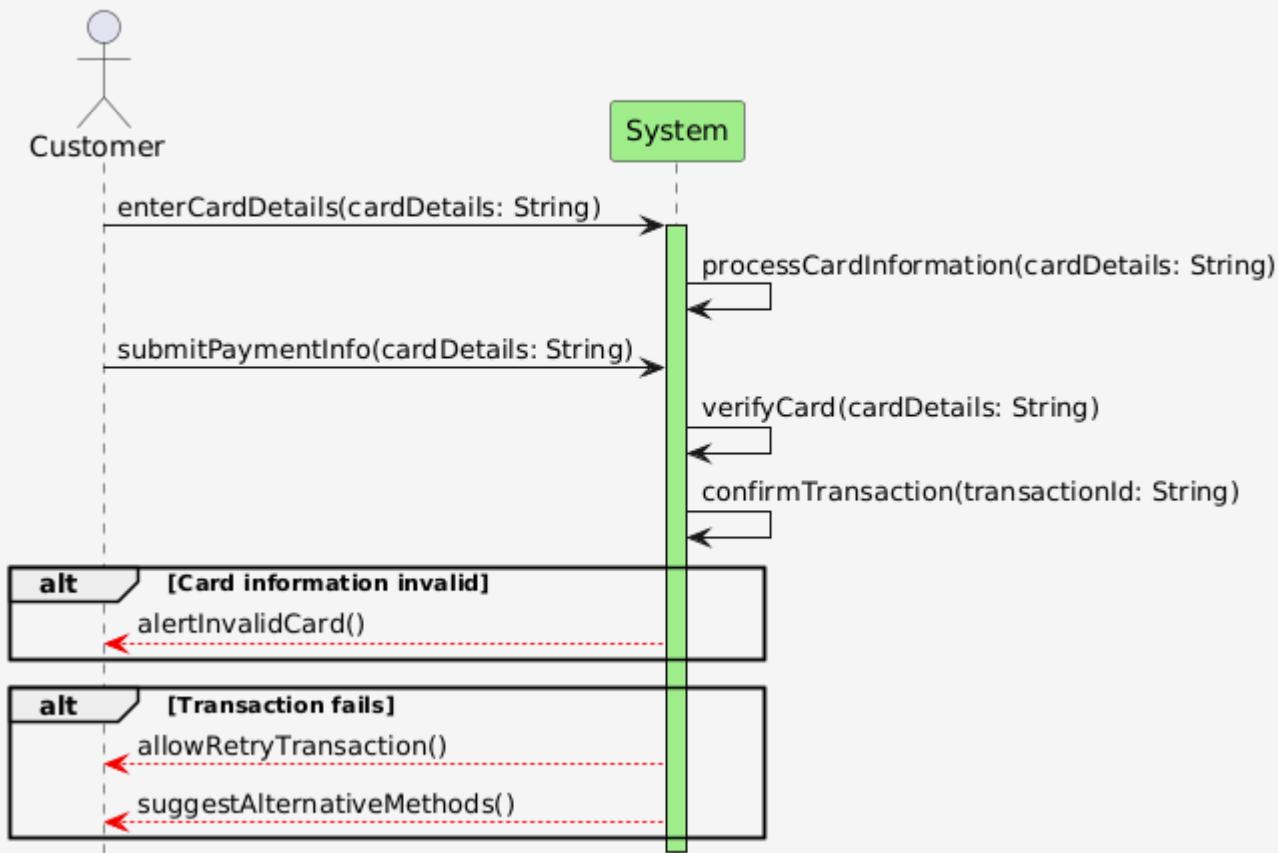
5. System Sequence Diagram



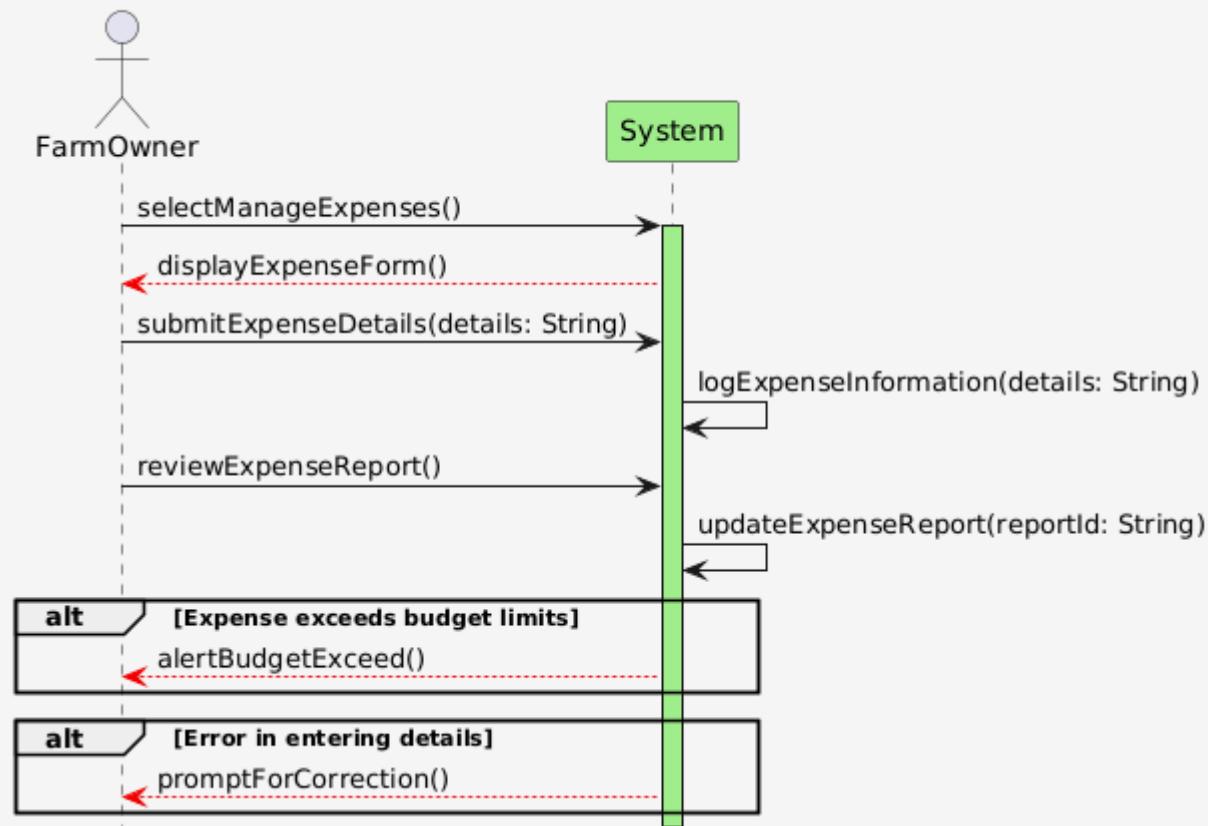




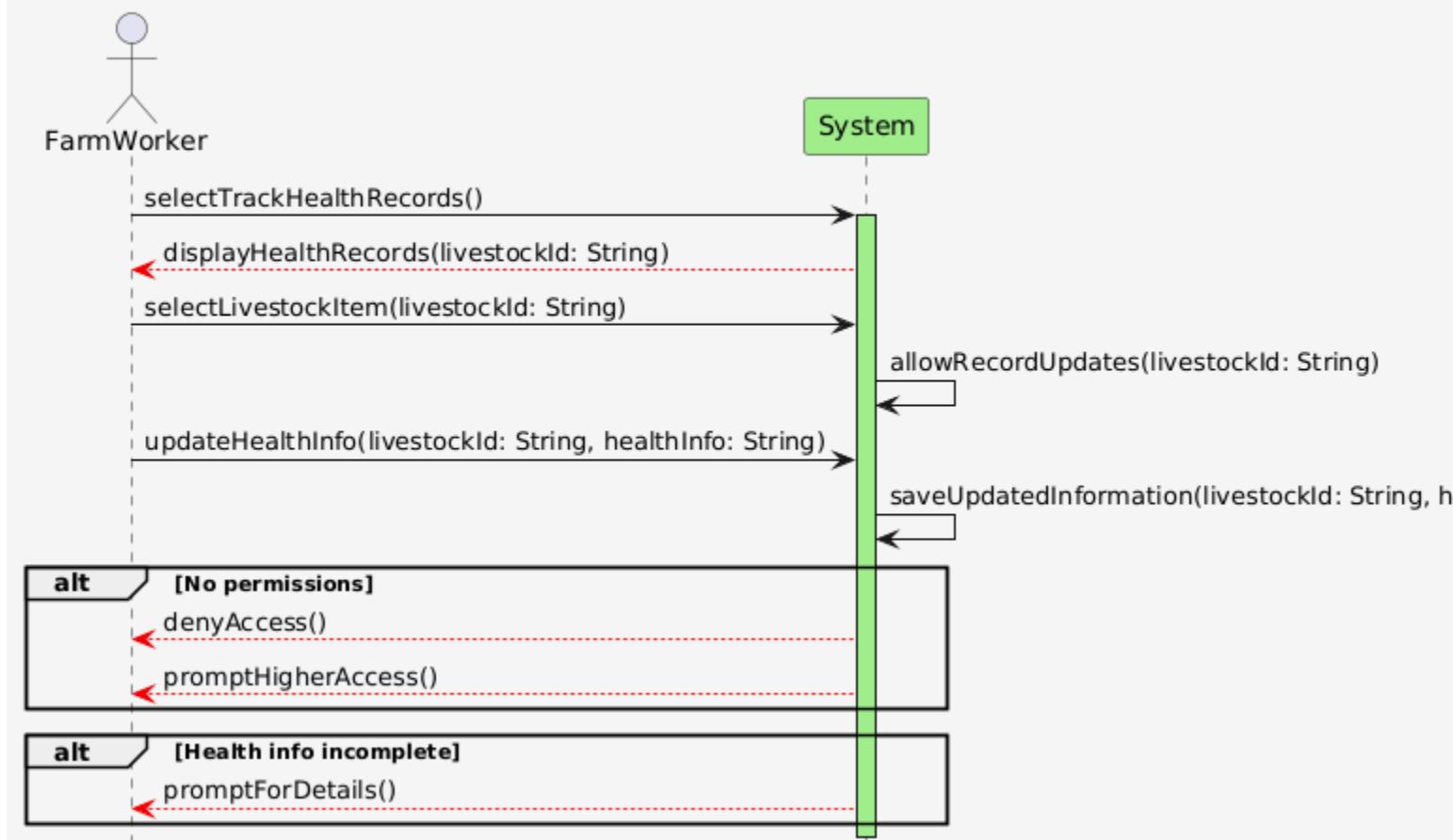
SSD for Credit Card Verification



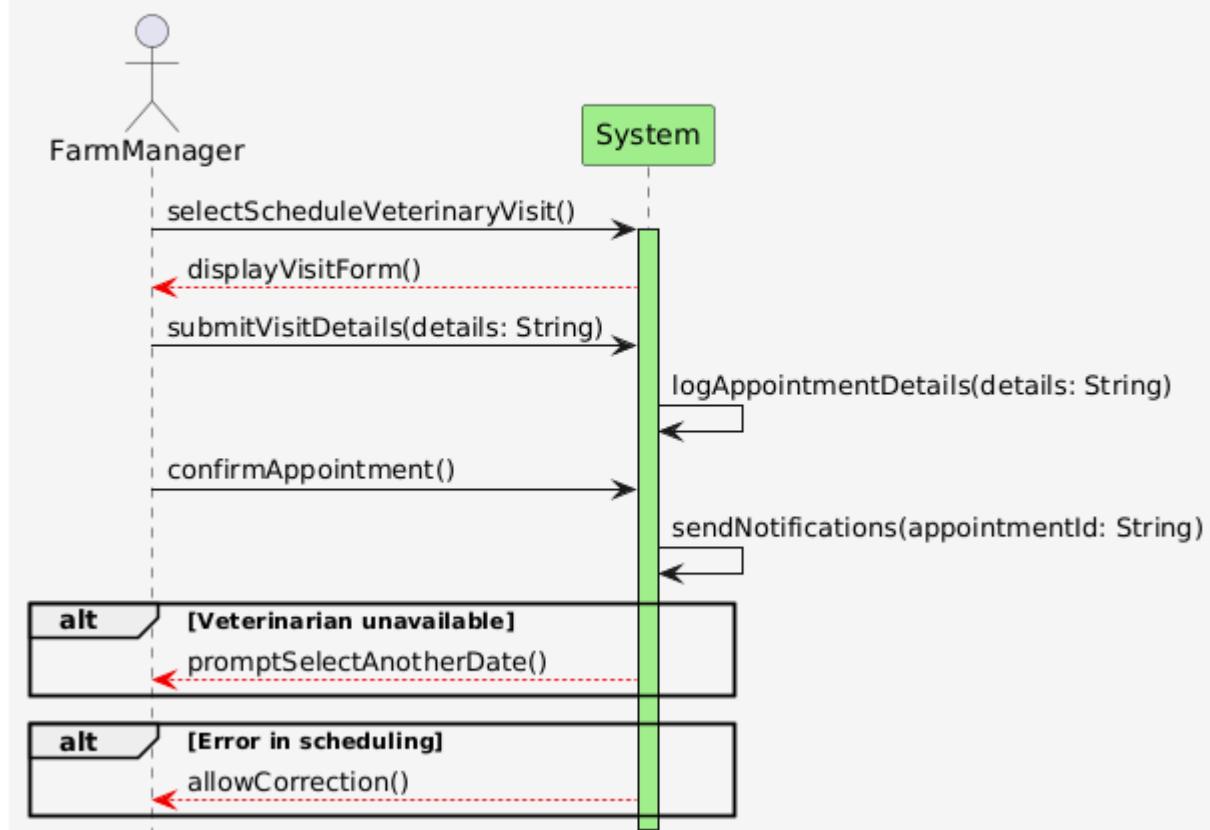
SSD for Manage Farm Expenses



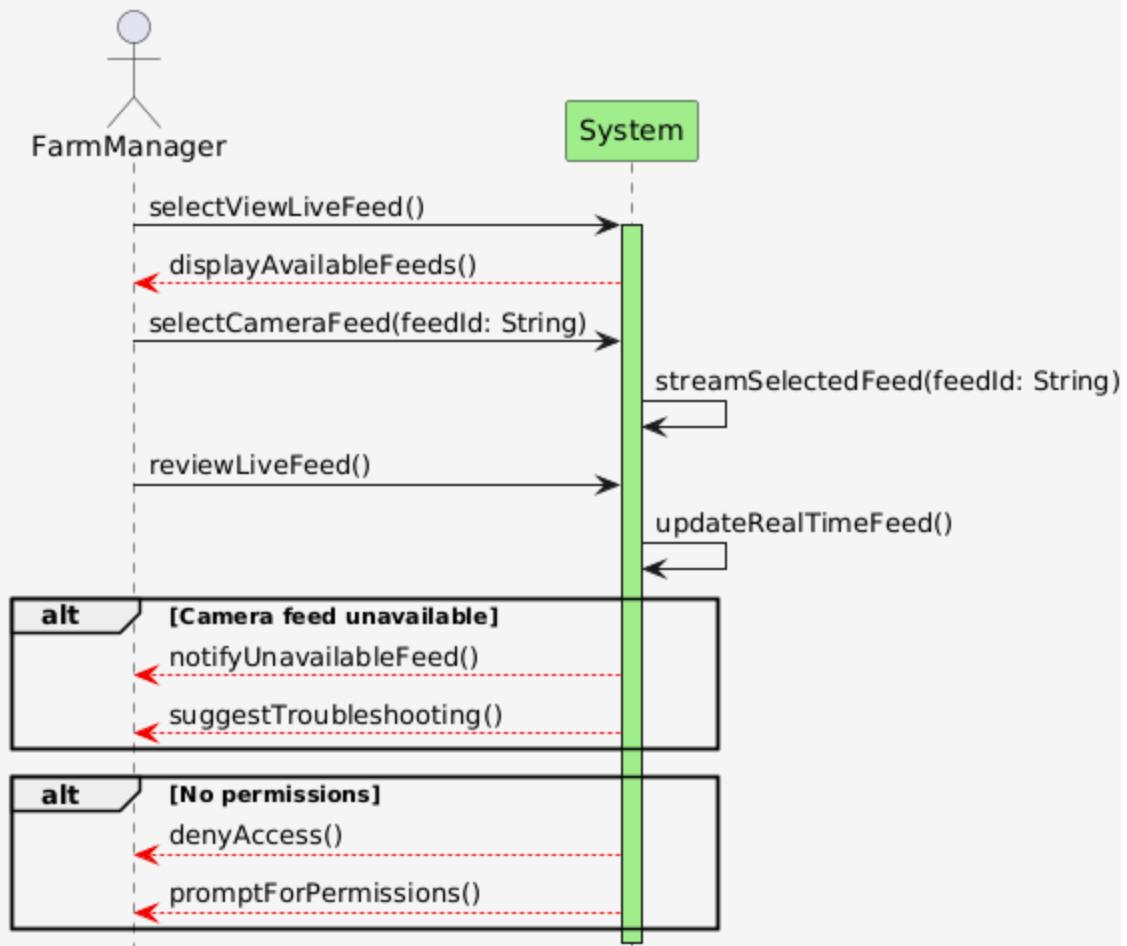
SSD for Track Livestock Health Records



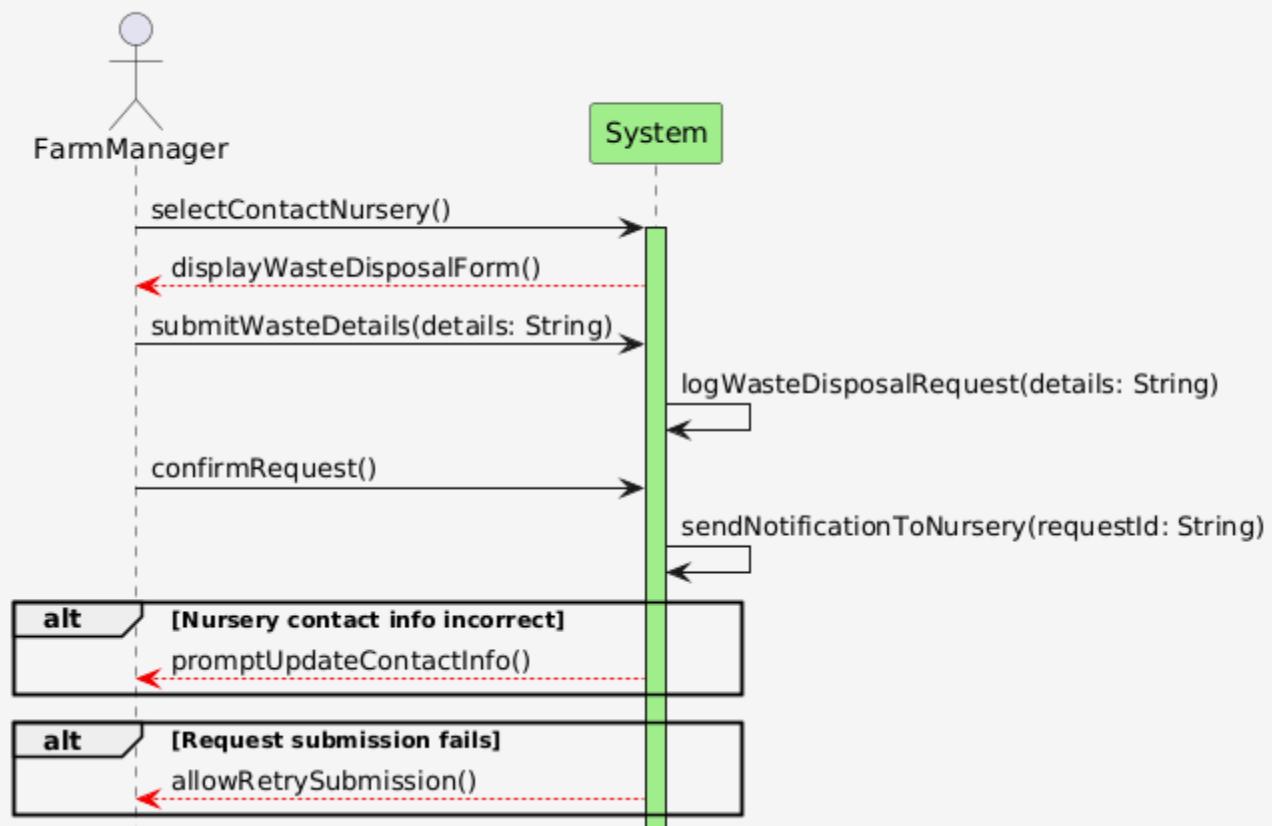
SSD for Schedule Veterinary Visits



SSD for Monitor Farm Activity via Live Camera Feed - FarmCam

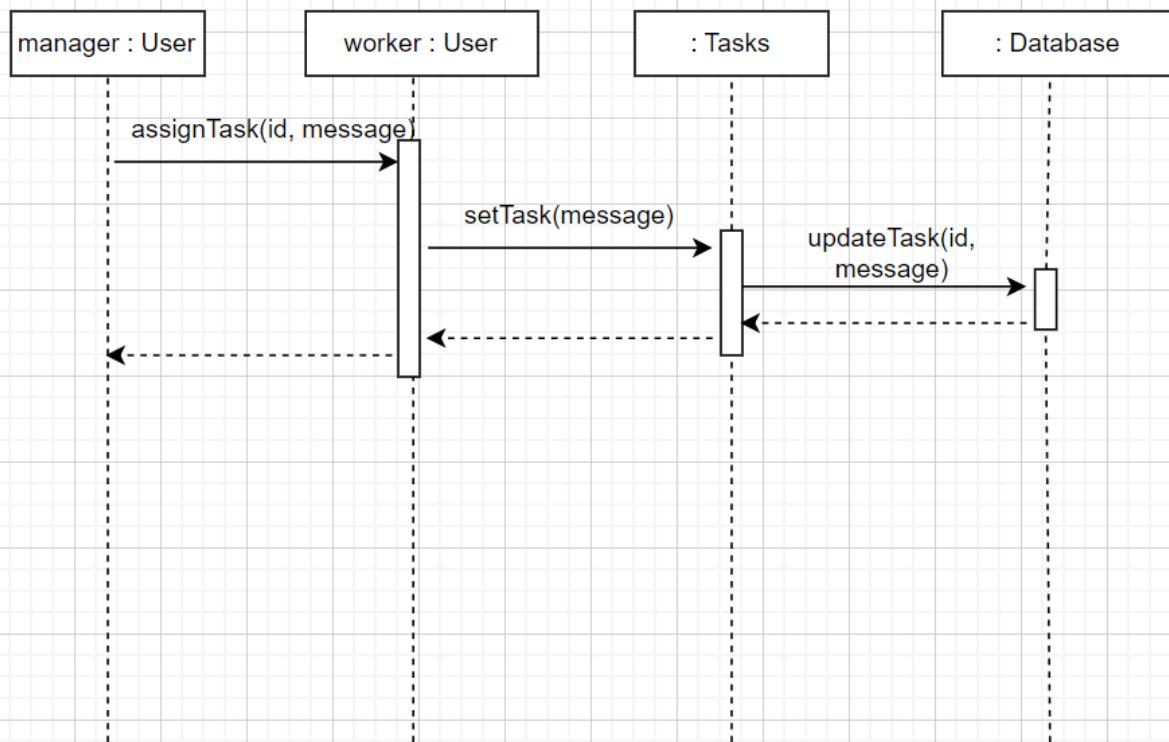


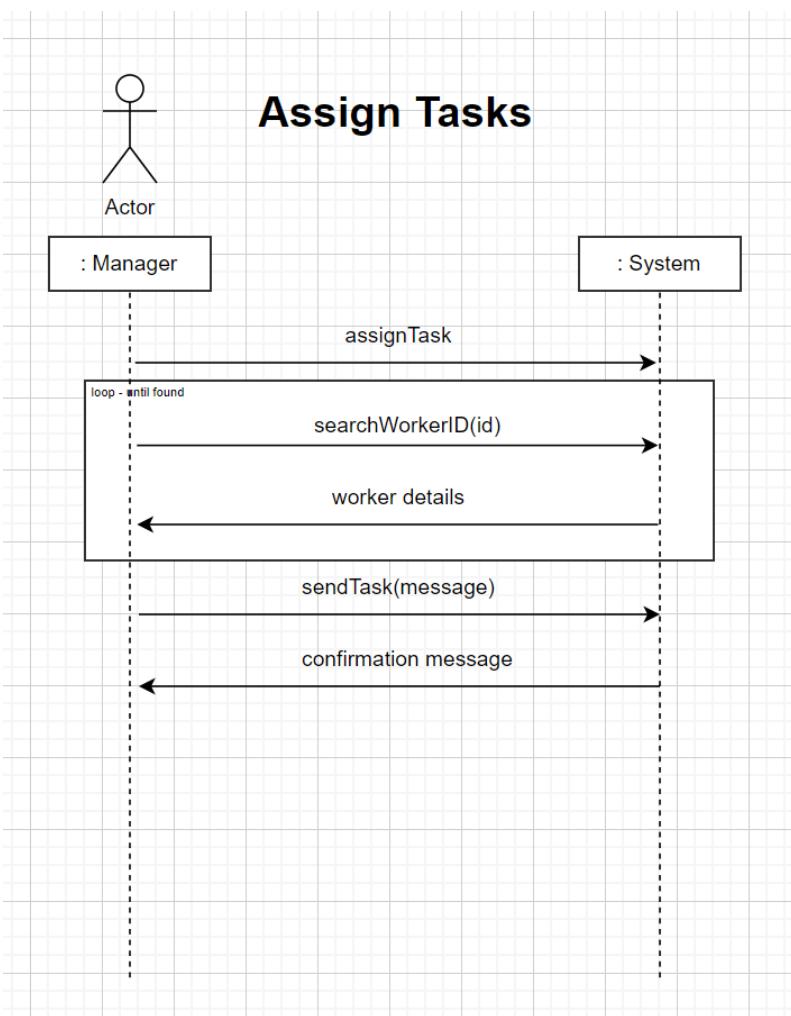
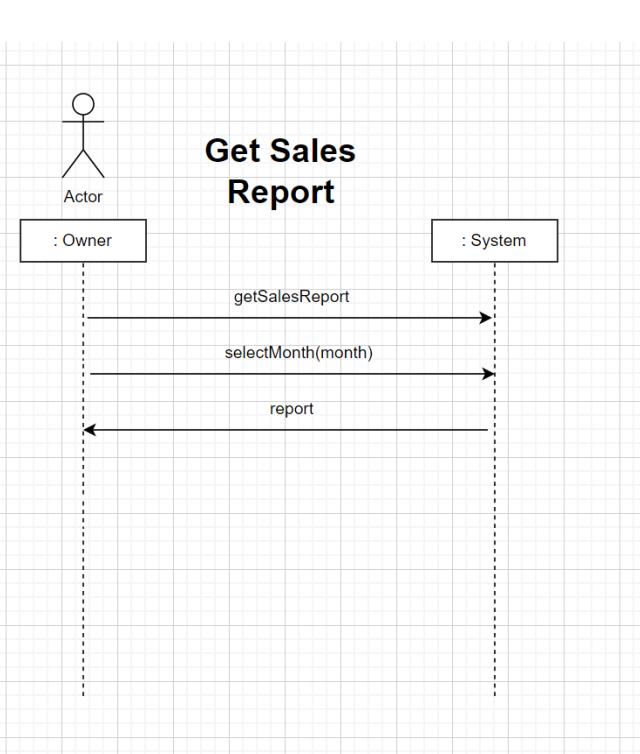
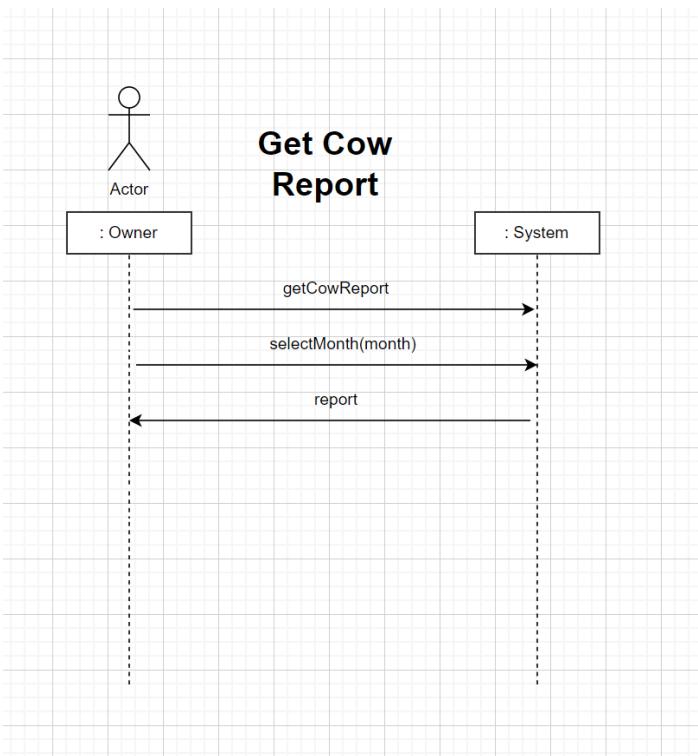
SSD for Waste Disposal Contact with Nursery

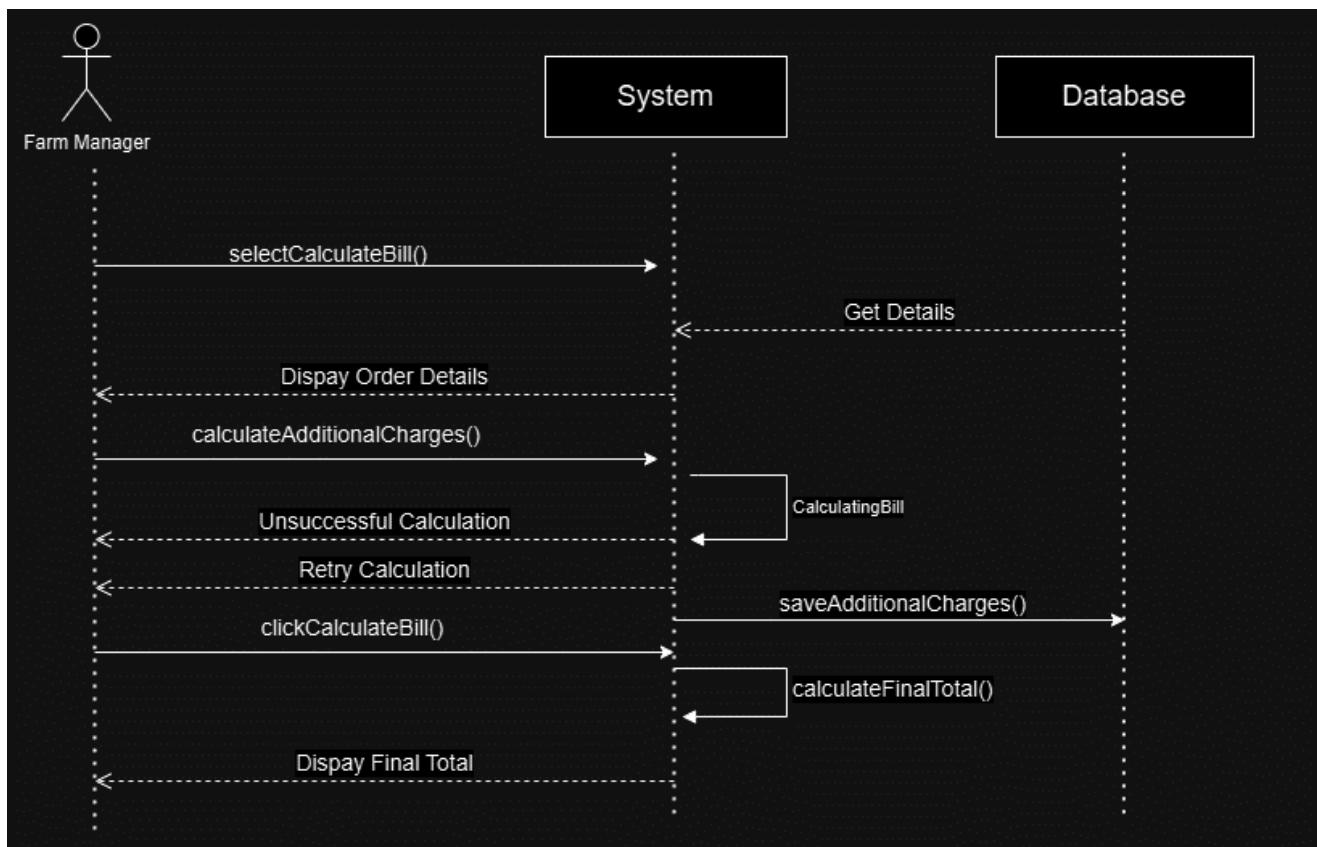
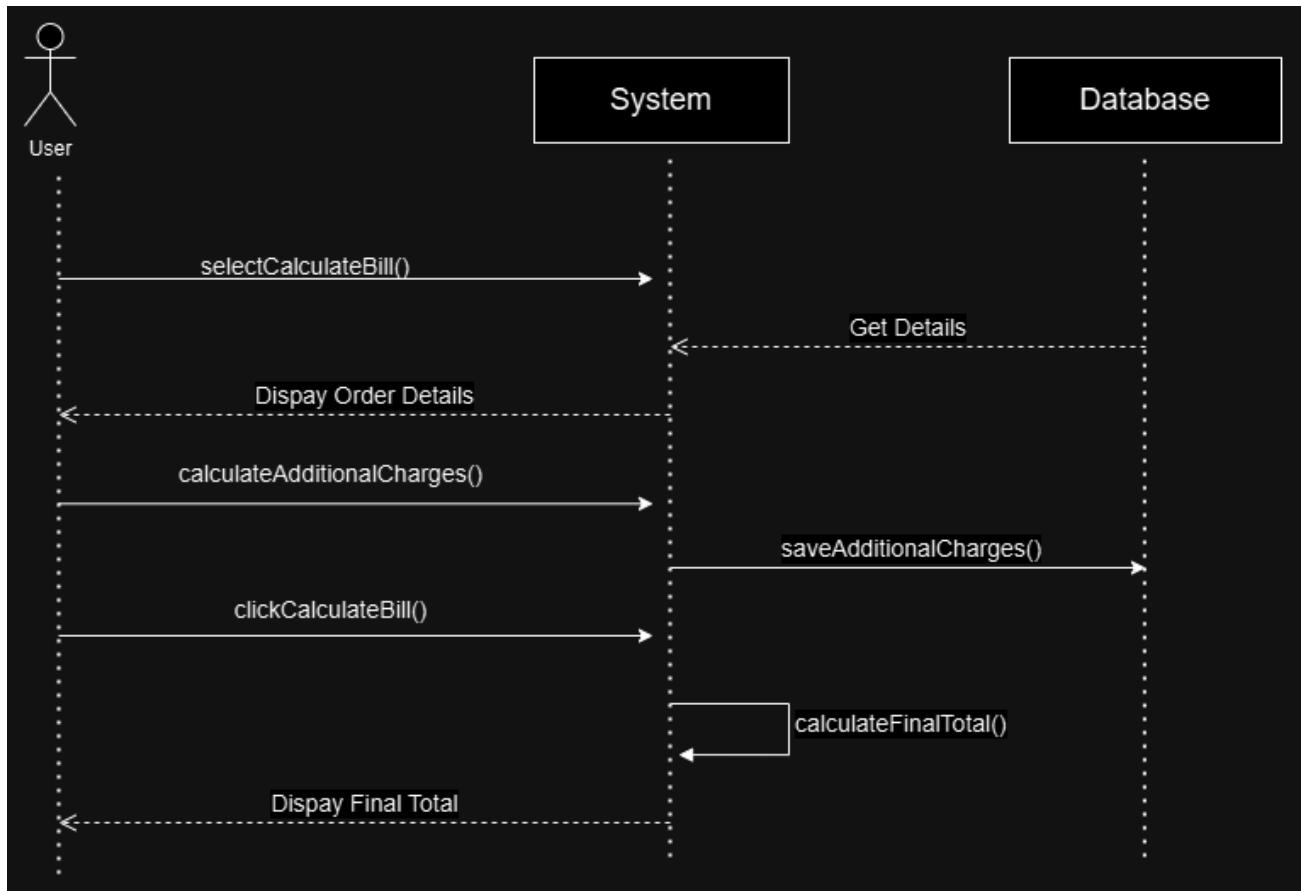


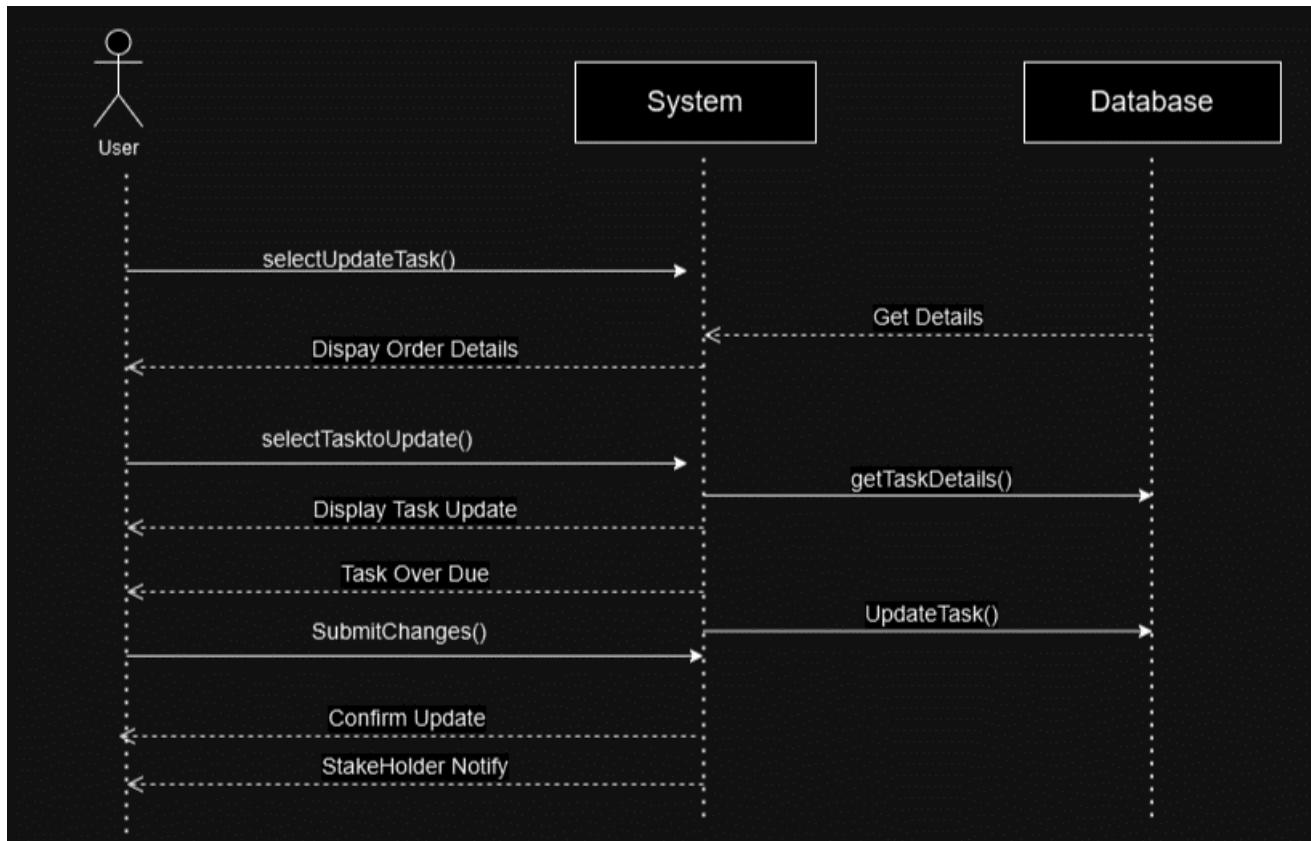
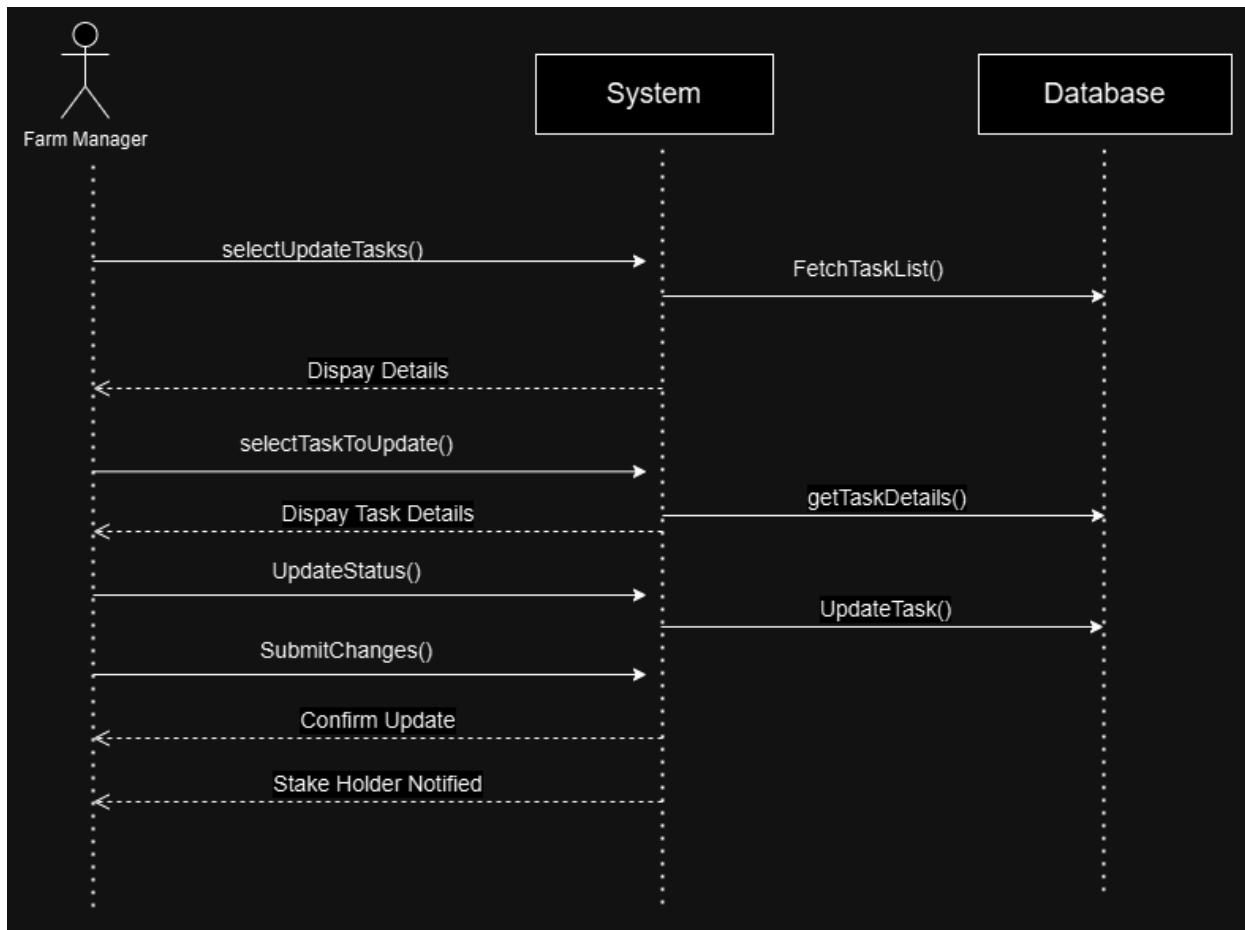
SD

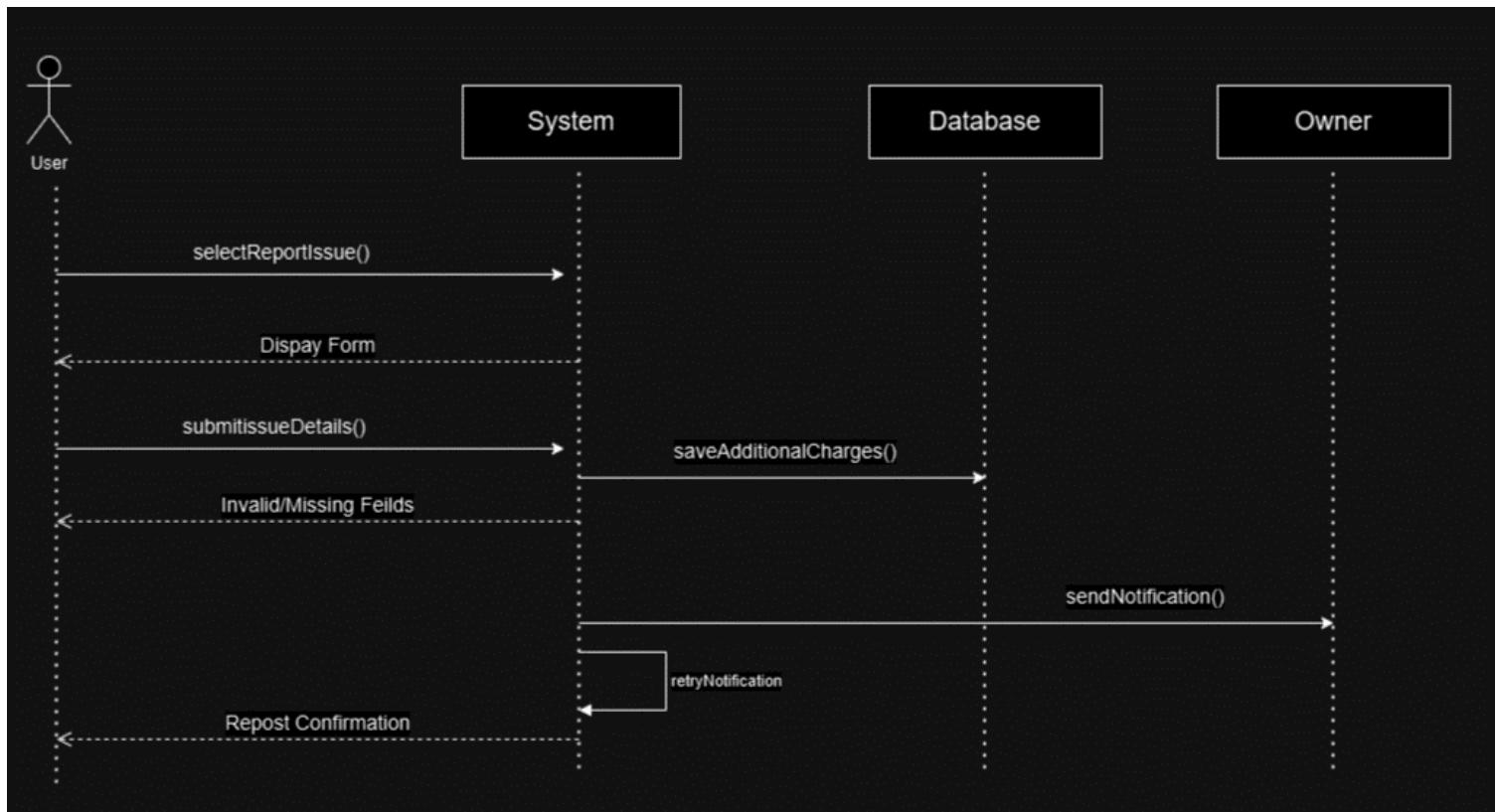
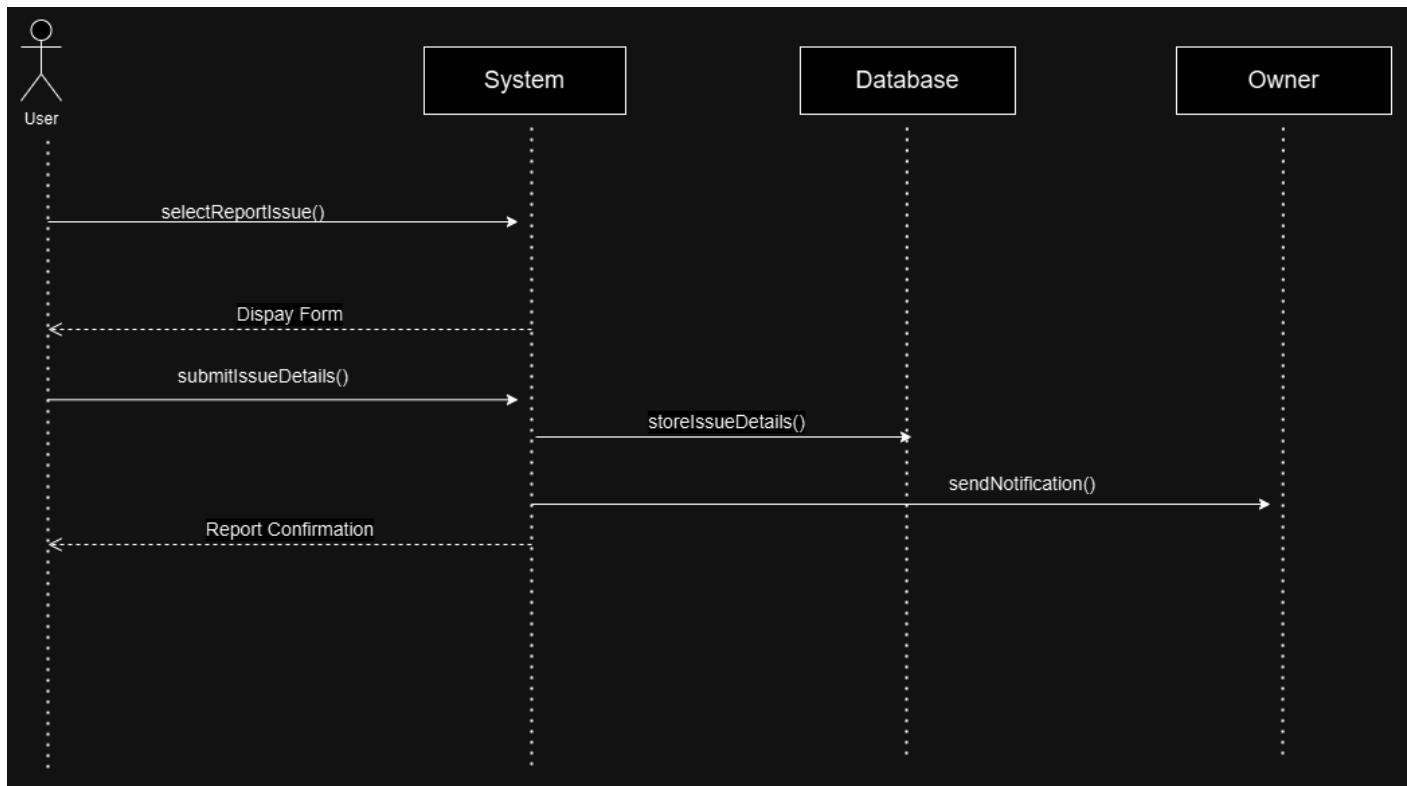
Assign Task



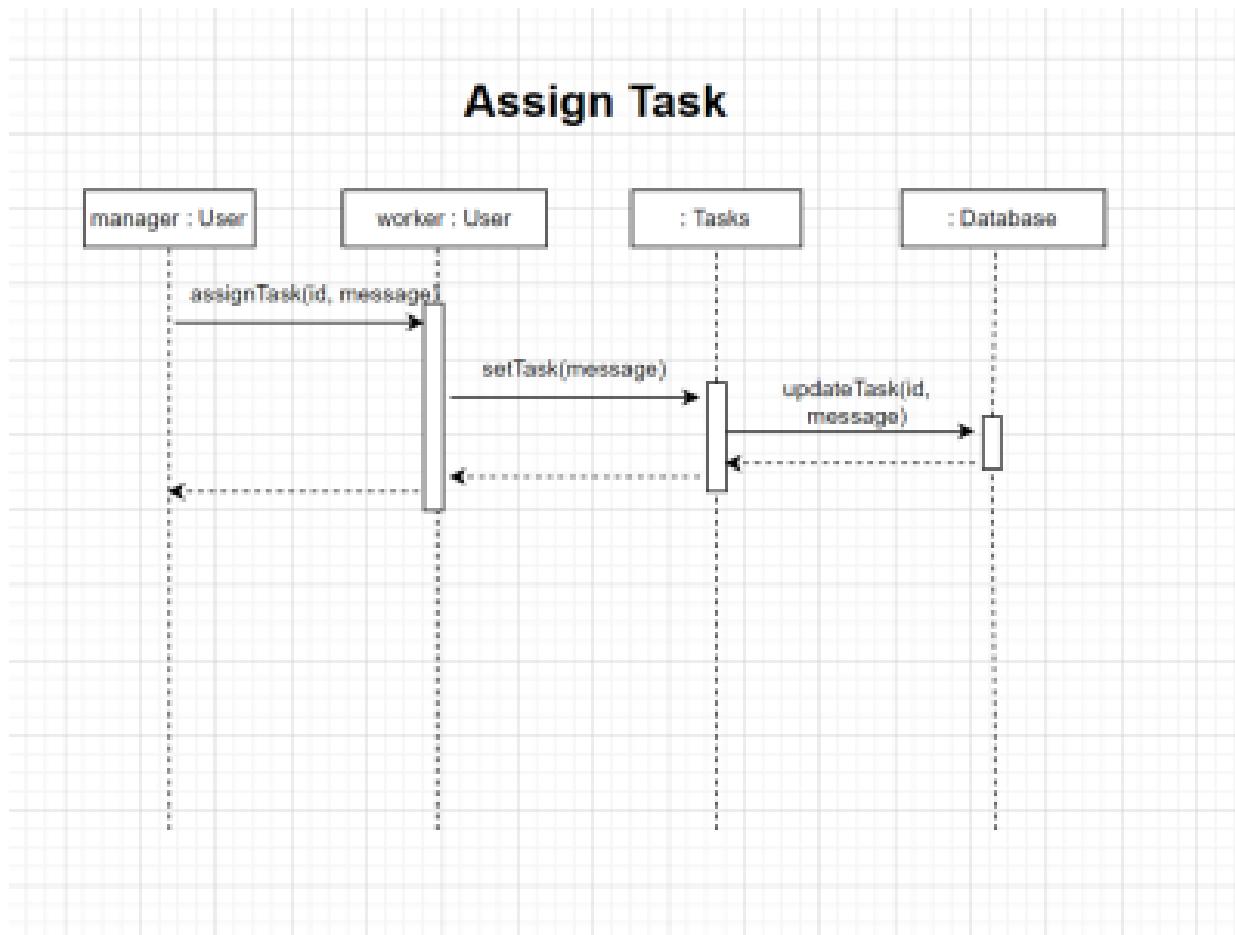




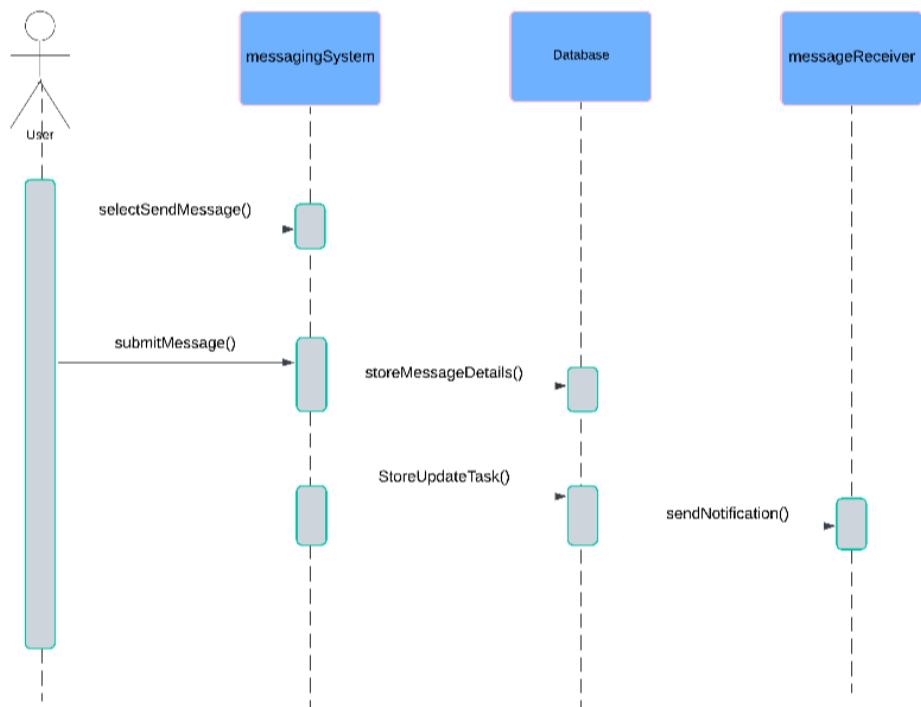




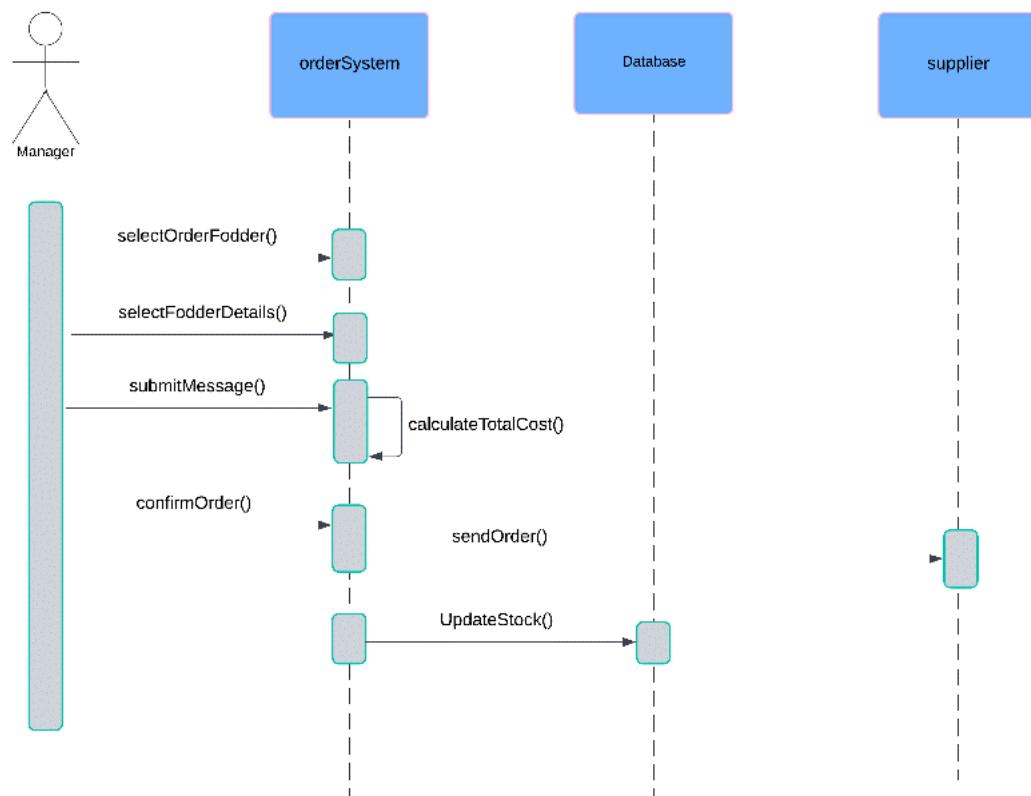
6. Sequence Diagram

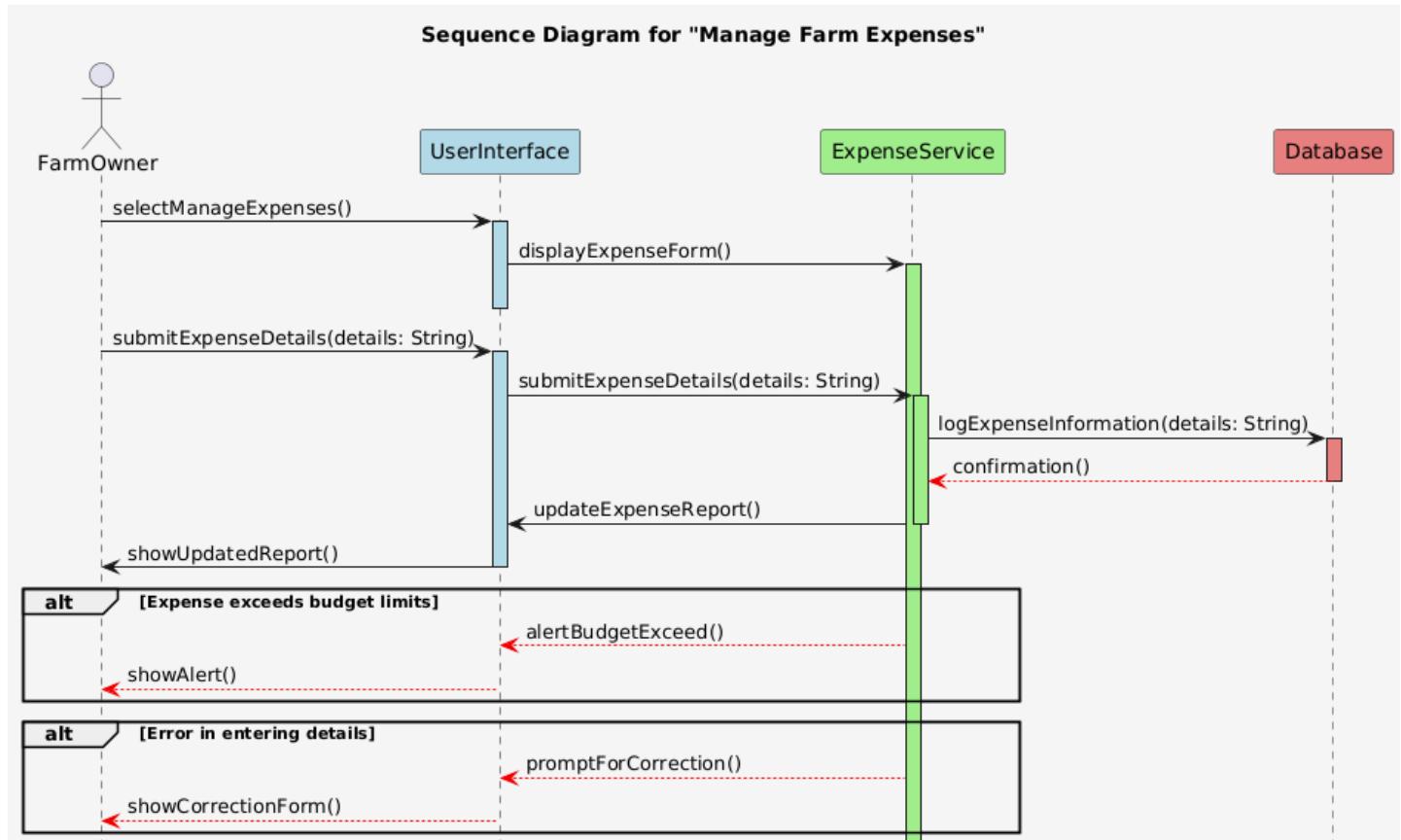
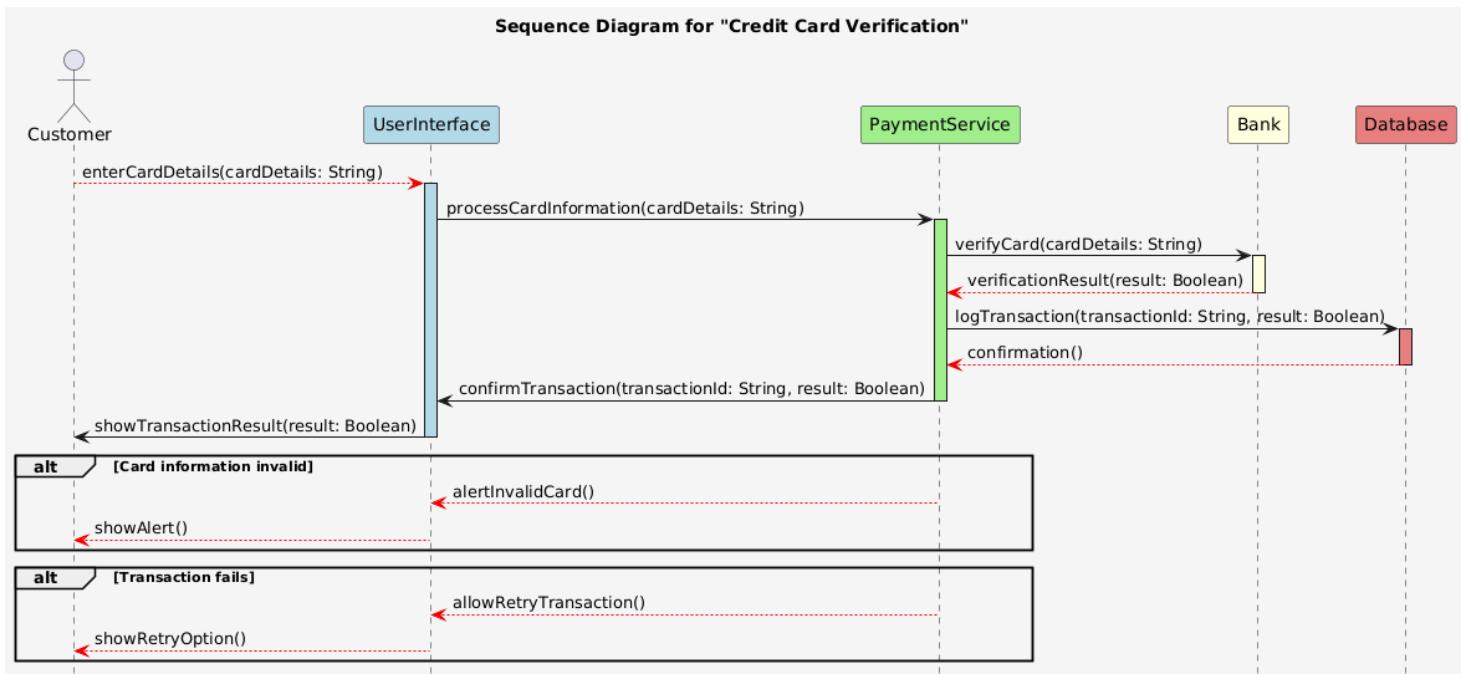


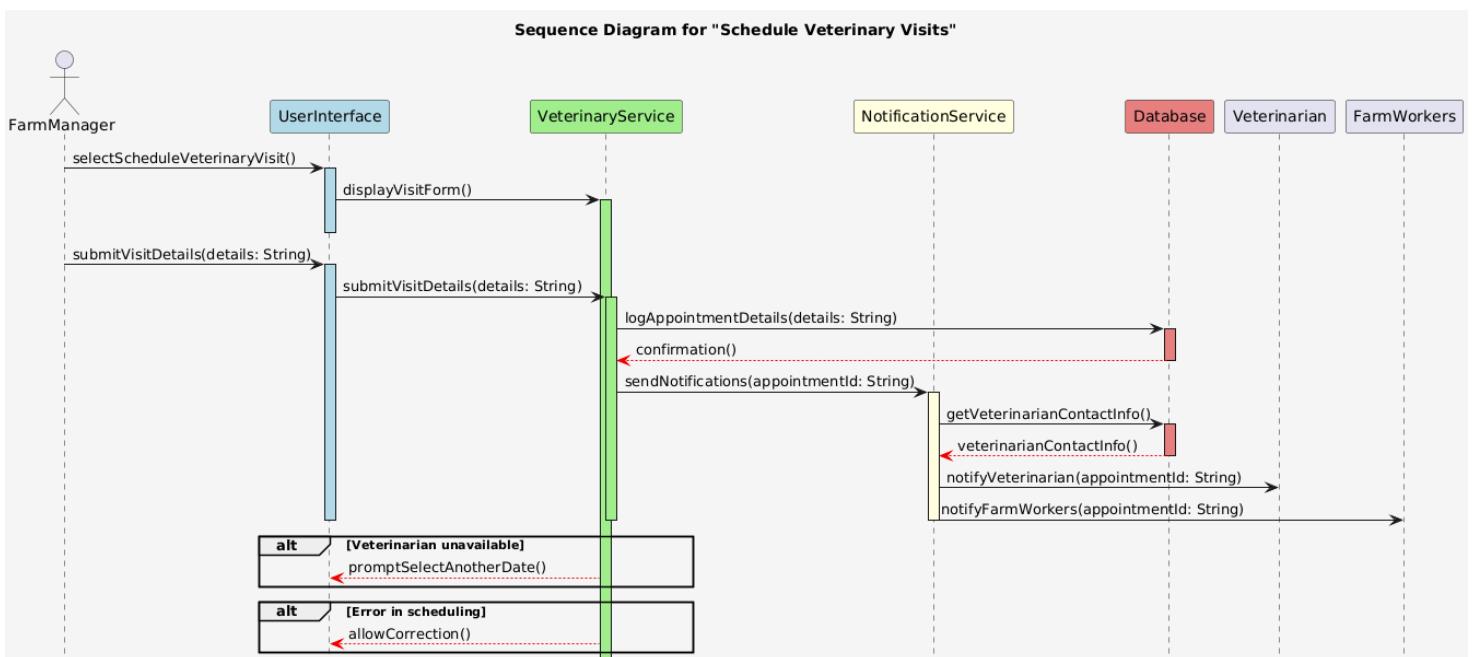
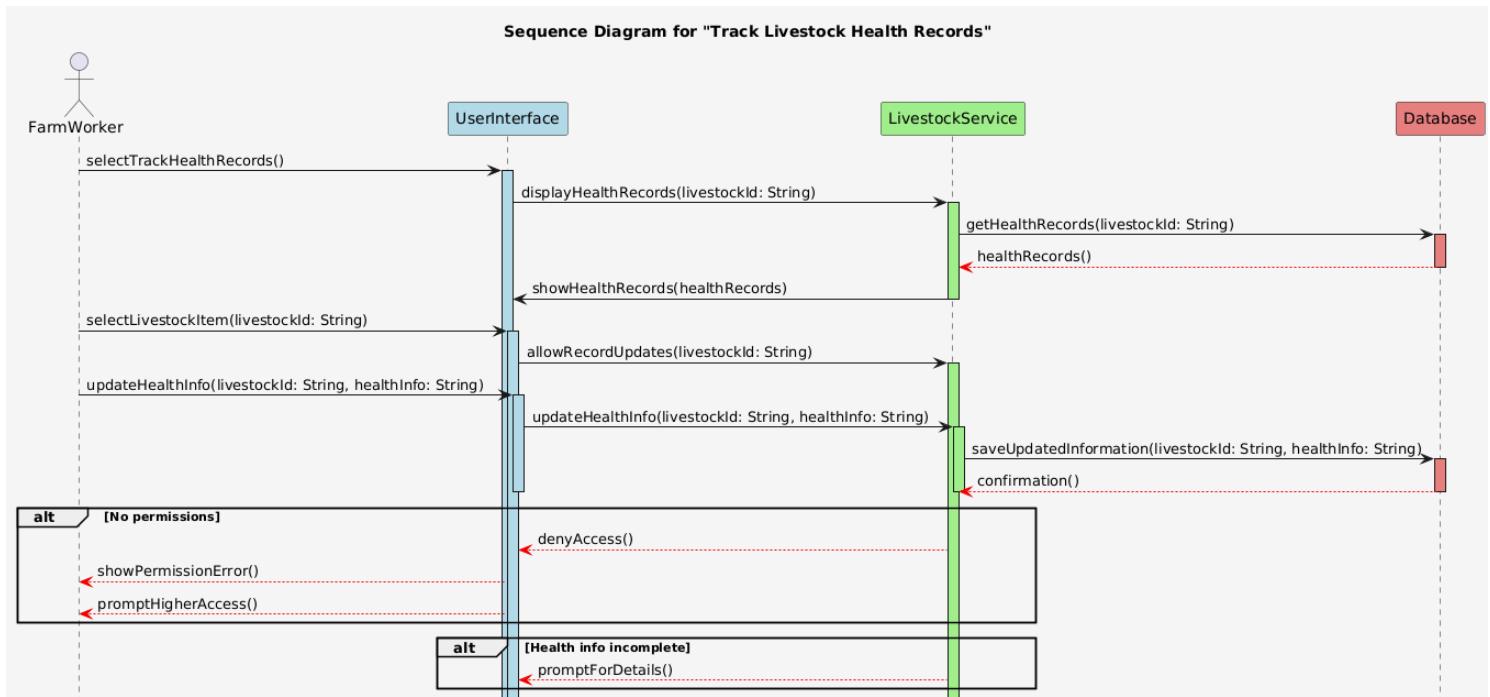
Q4:



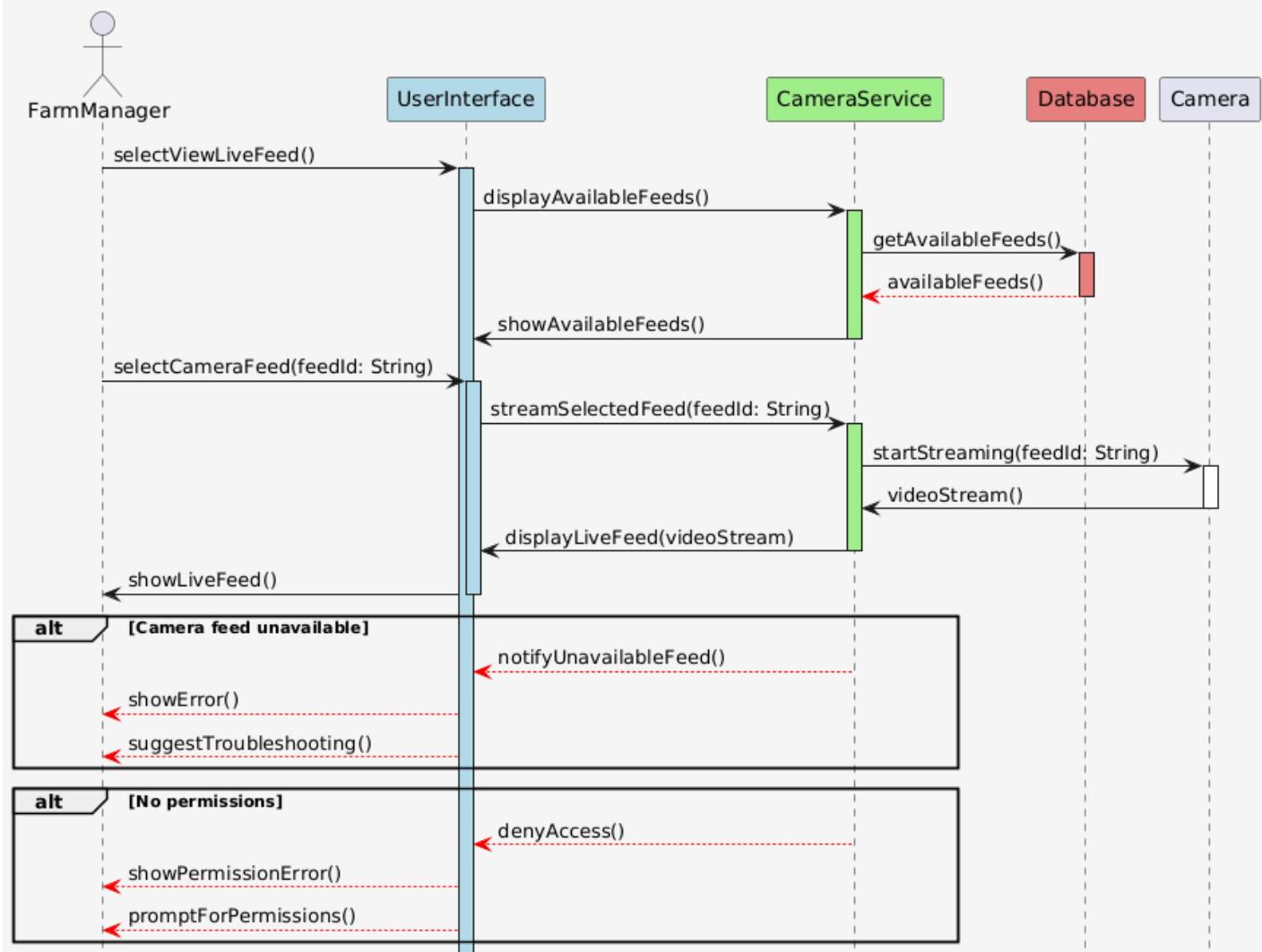
Q5:



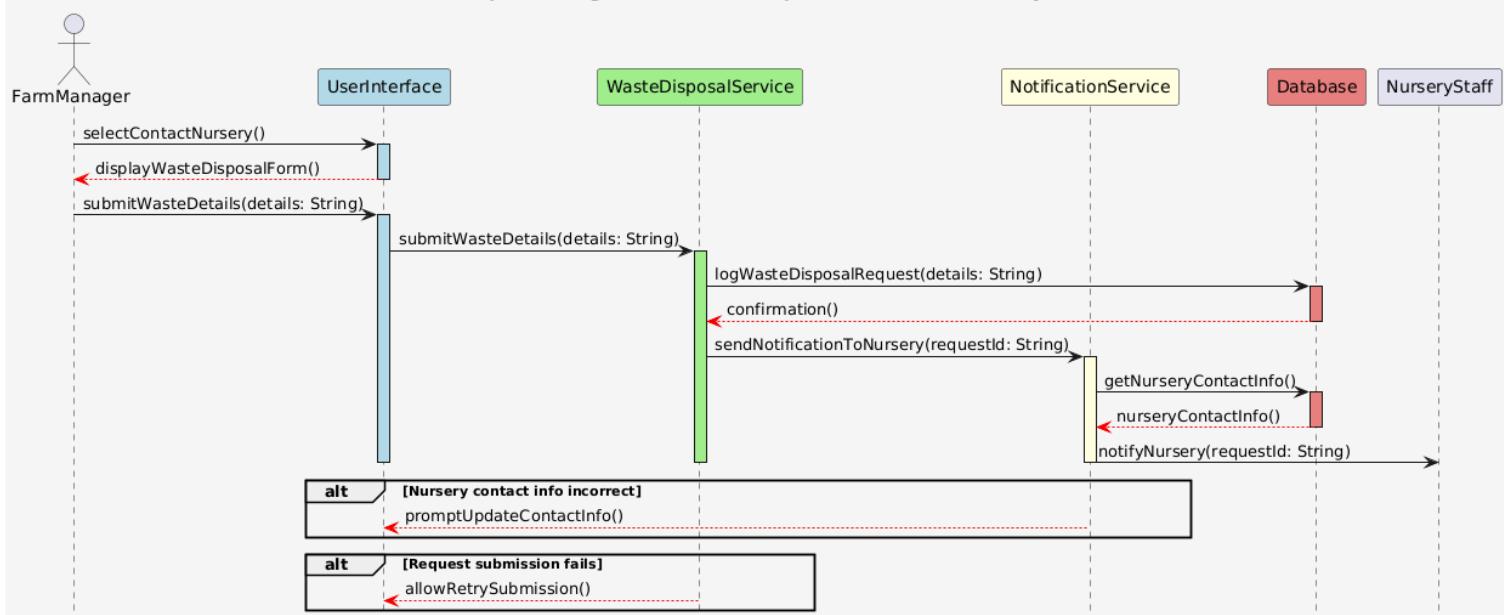




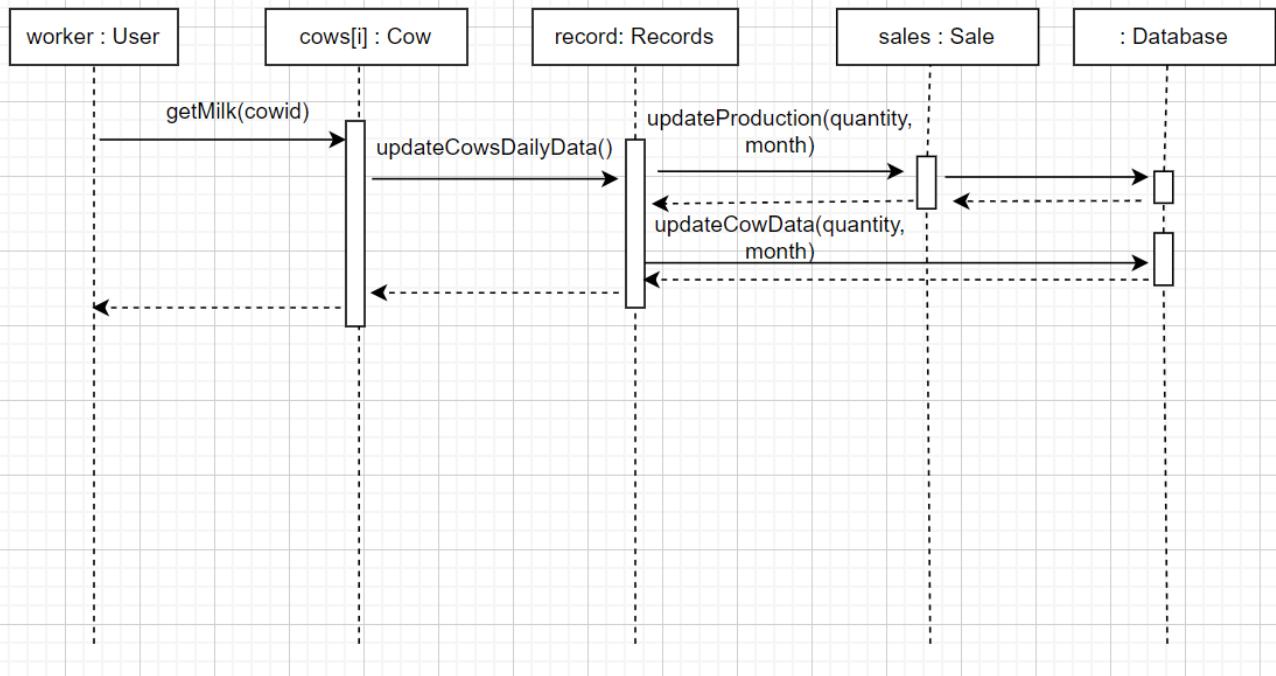
Sequence Diagram for "Monitor Farm Activity via Live Camera Feed - FarmCam"



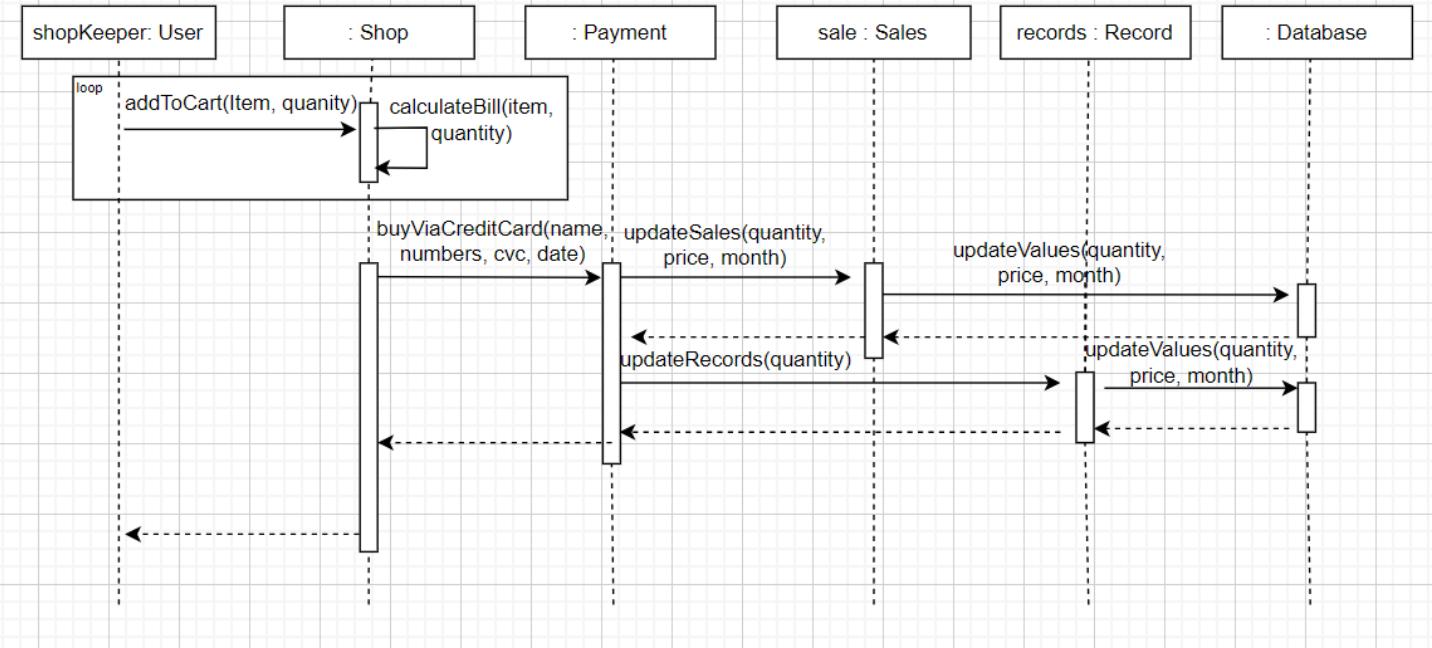
Sequence Diagram for "Waste Disposal Contact with Nursery"



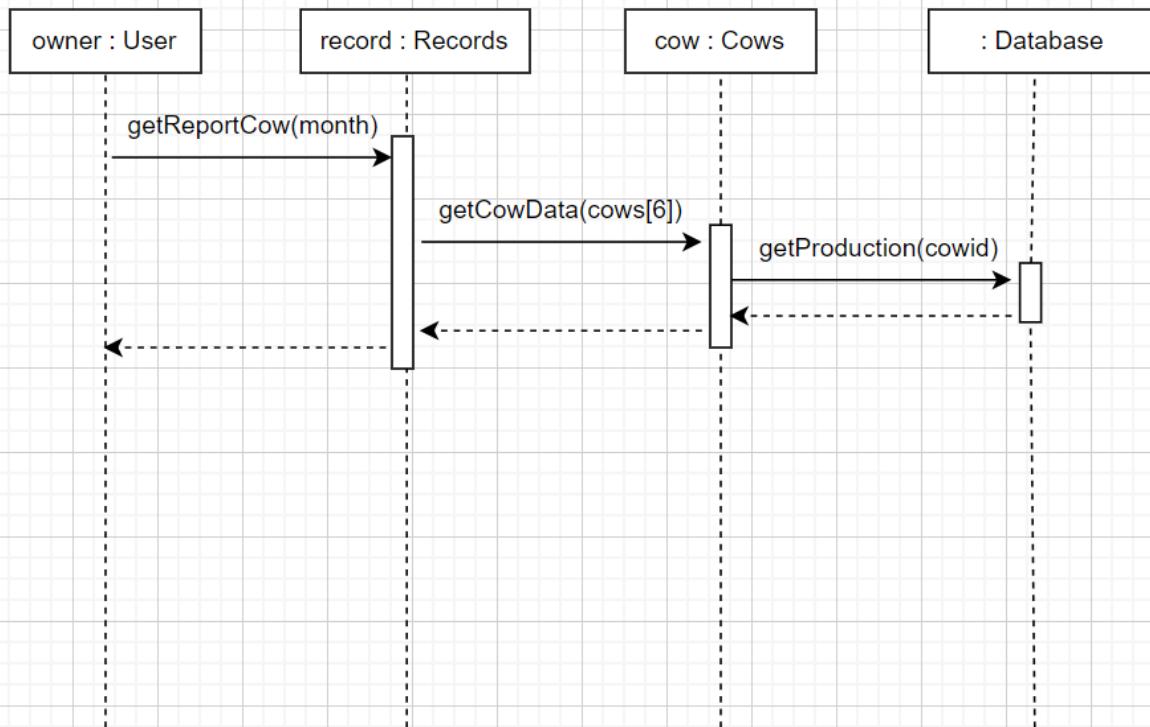
Update Milk Production



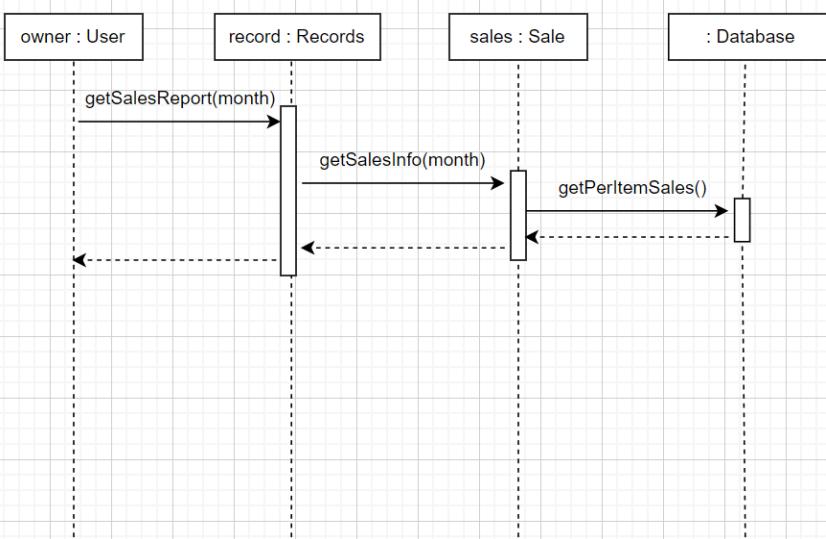
Perform Sale



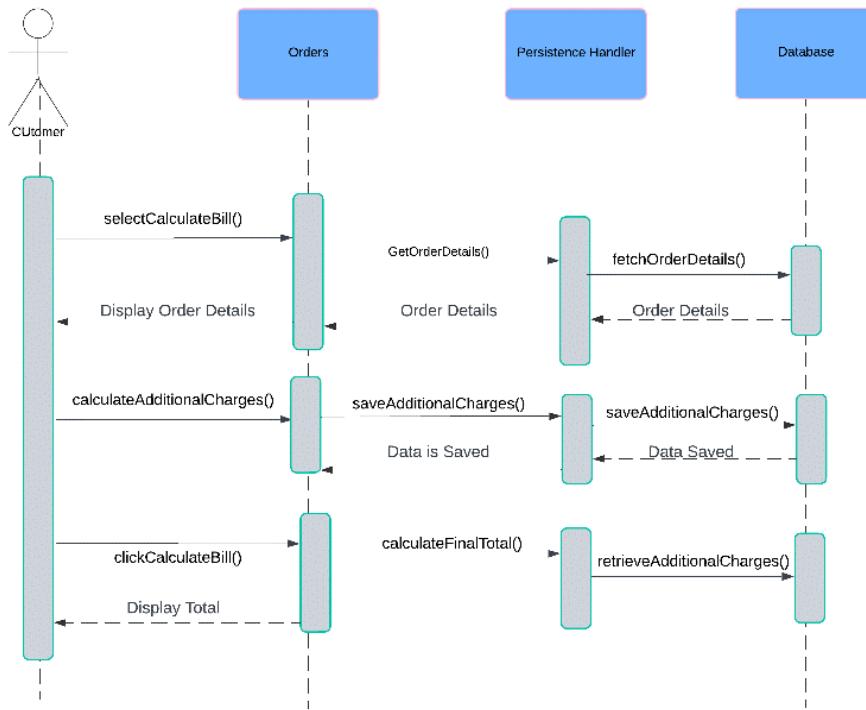
Get Cow Production Report



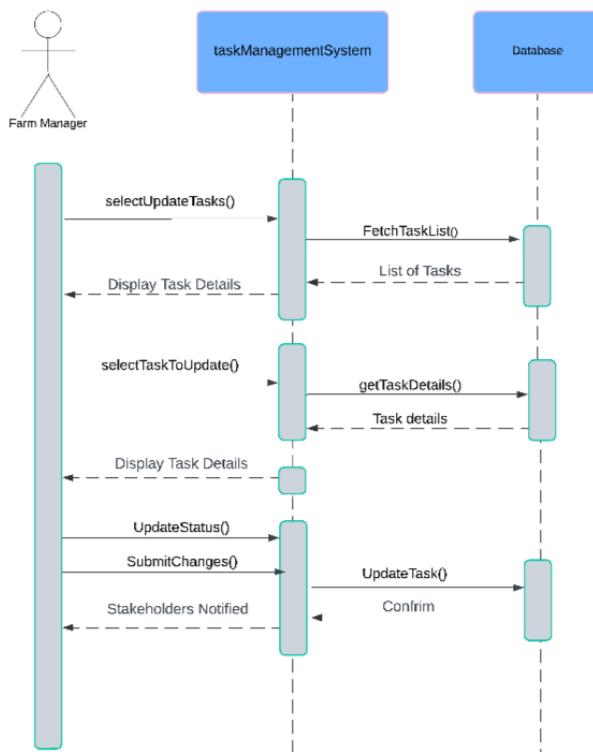
Get Sales Report



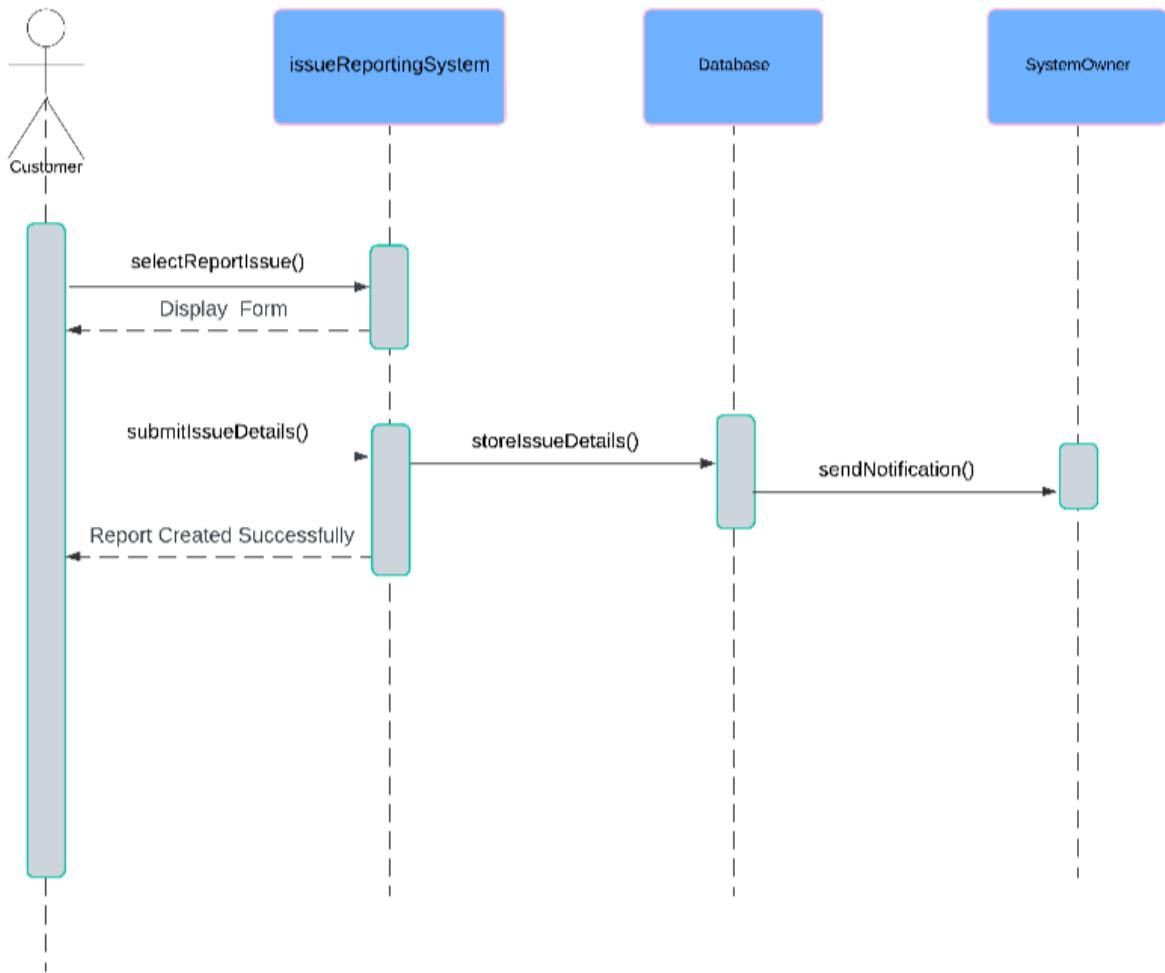
Q1:



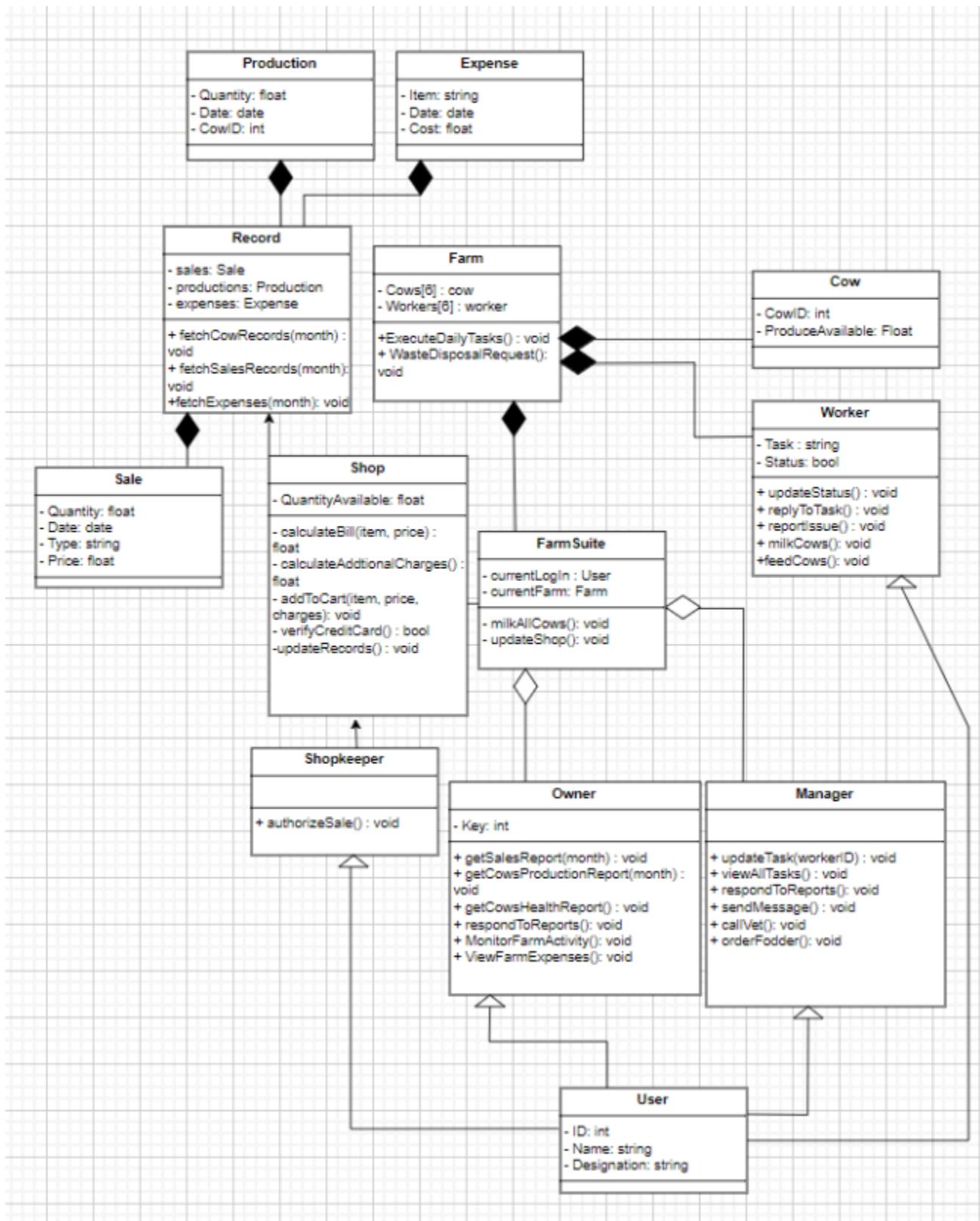
Q2:



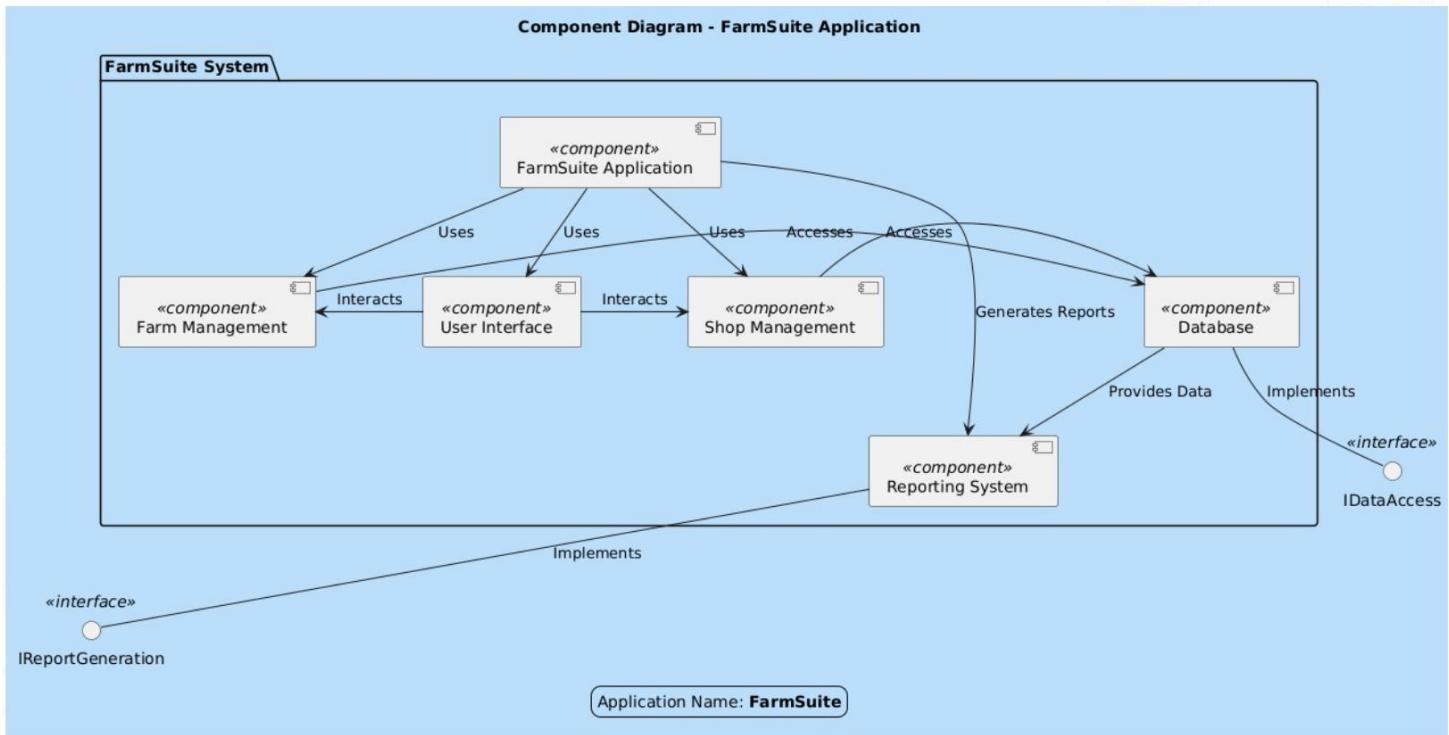
Q:3



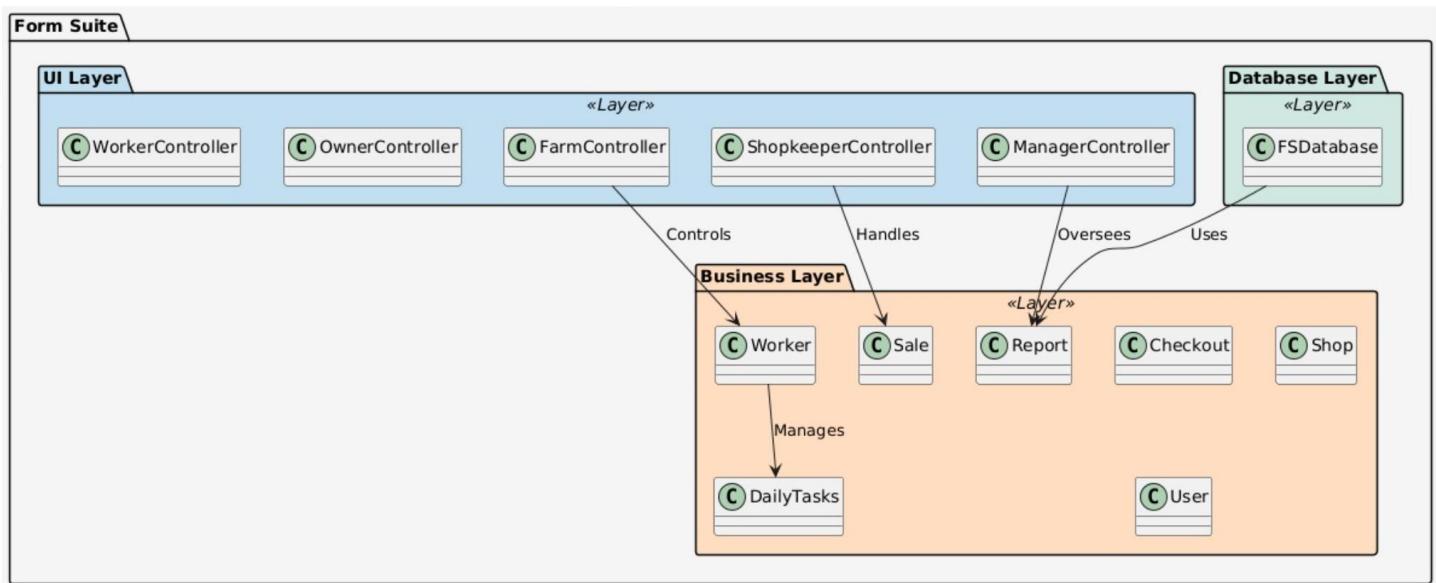
7. Class Diagram



8. Component Diagram



9. Package Diagram



10. Deployment Diagram

