Project Report 2024

NOVEMBER 27

Qubit Technologies

QUBIT

1. Introduction

1.1 Purpose

FarmConnect is a digital platform aimed at addressing the challenges faced by small-scale farmers in Pakistan by bridging the gap between farmers and consumers. The platform enables direct market access, eliminates intermediaries, and ensures fair pricing for agricultural produce. This document outlines the software requirements for the FarmConnect system, focusing on its purpose, objectives, and features.

1.2 Product Scope

FarmConnect provides a user-friendly digital marketplace where small-scale farmers can connect directly with consumers and mills. The platform offers streamlined operations for order and delivery management, ensuring fresh, locally-sourced produce at competitive prices. FarmConnect is specifically designed for small-scale farmers in both rural and urban areas, distinguishing it from large-scale commercial platforms like FarmCrowdy and eFresh.

1.3 Title

FarmConnect: solution designed to empower small-scale farmers by providing equitable market access while simplifying operations for all stakeholders in the agricultural supply chain.

1.4 Objectives

- Enable direct connections between farmers, consumers, and mills.
- Provide affordable, fresh produce to consumers.
- Streamline order and delivery processes through a user-friendly interface.
- Support government monitoring and enforcement of fair pricing regulations.

1.5 Problem Statement

Small-scale farmers in Pakistan face significant challenges due to the exploitation of intermediaries, who often purchase produce at low prices and resell it at higher rates. This practice reduces farmers' profits and hampers their financial growth. The lack of enforcement of government-mandated crop prices has further aggravated the issue. Many farmers, under financial pressure, are compelled to sell their produce at unfair rates.
FarmConnect addresses these challenges by eliminating intermediaries and providing a platform that ensures fair pricing and compliance with government regulations. By leveraging mobile and web technologies, FarmConnect empowers small-scale farmers with equitable access to a transparent and fair marketplace.

2. Overall Description

2.1 Product Perspective

FarmConnect is a new, self-contained product designed specifically for small-scale farmers. Unlike existing platforms targeting commercial farming, FarmConnect tailors its features to meet the unique needs of local farmers. The system comprises a user-friendly frontend (Next.js), a business logic layer enforcing core operations, and a PostgreSQL-based database managed via Supabase.
A simplified architecture diagram illustrates the interaction between the UI, business logic, and database layers.

2.2 Product Functions

- User registration and authentication.
- Management of product catalogs, orders, and deliveries.
- Price setting and compliance enforcement for registered mills.
- Feedback collection and dispute resolution.
- Real-time updates on product availability and order status.

2.3 List of Use Cases

1. Register Farmer
2. Browse Available Products
3. Place an Order
4. Update Crop Availability
5. Set Fair Price for Products
6. Schedule Delivery
7. Manage Consumer Payments
8. Resolve Pricing Disputes
9. Generate Sales Report for Farmers
10. Manage Registered Mills
11. Provide Feedback
12. Review History

2.4 Extended Use Cases

Use Case 1: Register Farmer

- *Created by: Muhammad Tahir*
- **Scope**: Farm Connect
- **Level**: User-goal
- **Primary Actor**: Farmer
- **Stakeholders and Interests**:

- **Farmer**: Wants to register on the platform and access features like uploading products and viewing orders.
- **Admin**: Ensures that only legitimate farmers are registered.
- **Consumers**: Want access to verified farmers for purchasing local products.

- **Preconditions**: Farmer has access to a mobile device or computer with internet connectivity.
- **Postconditions**:
  - **Success**: Farmer account is created, and a welcome message is sent.
  - **Failure**: Farmer is notified of registration errors and corrective actions.
- **Main Success Scenario**:

Step	Actor Action	System Responsibility
1.	Farmer opens the Farm Connect website or app.	
2.	Farmer selects the "Register" option.	
3.	Farmer enters personal details, farm location, and crop types.	
4.		System validates the information.
5.	Farmer submits the registration form.	
6.		System creates a farmer account.
7.		**Farmer receives a confirmation message.**

- **Extensions**:
  - **4a**: If the details entered are invalid (e.g., missing or incorrect fields):
    - System shows an error message, and the farmer is asked to correct the input.
  - **6a**: If the server is down:
    - The system logs the issue and notifies the farmer to try again later.

Use Case 2: Browse Available Product

- *Created by: Muhammad Tahir*
- **Scope**: Farm Connect
- **Level**: User-goal

- **Primary Actor**: Consumer
- **Stakeholders and Interests**:
    - **Consumer**: Wants to access local products quickly, filter products, and make informed choices.
    - **Farmer**: Wants their products to be visible to potential buyers.
    - **Admin**: Ensures platform usability and performance during high traffic.
- **Preconditions**: Consumer has an active account and is logged in.
- **Postconditions**:
    - **Success**: Consumer is able to browse, filter, and view details of available products.
    - **Failure**: Consumer is notified of any issues, like no available products or system errors.
- **Main Success Scenario**:

Step	Actor Action	System Responsibility
1.	**Consumer logs into their account.**	
2.	Consumer navigates to the "Browse Produce" section.	
3.	Consumer filters by location, product type, or price range.	
4.		The system retrieves available products that match the criteria.
5.	Consumer views the product details (price, availability, farmer info).	

- **Extensions**:
    - **3a**: If no products matches the filters:
        - System informs the consumer and suggests broader filters.
    - **5a**: If a product is out of stock:
        - System flags the product as unavailable.

Use Case 3: Place an Order

- *Created by: Muhammad Tahir*

- **Scope**: Farm Connect
- **Level**: User-goal
- **Primary Actor**: Consumer
- **Stakeholders and Interests**:
    - **Consumer**: Wants to purchase products quickly and easily.
    - **Farmer**: Wants to sell their products and confirm orders.
    - **Admin**: Ensures seamless order processing and fulfillment.
    - **Payment Gateway**: Processes the payment securely.
    - **Delivery Service**: Coordinates logistics and ensures timely delivery.
- **Preconditions**: Consumer is logged in and has selected a product.
- **Postconditions**:
    - **Success**: Order is confirmed, and both the farmer and consumer receive a notification.
    - **Failure**: Consumer is notified of the failure, such as stock issues or payment failure.
- **Main Success Scenario**:

Step	Actor Action	System Responsibility
1.	**Consumer selects a product from the "Browse Produce" section.**	
2.	**Consumer adds the product to the cart.**	
3.	**Consumer confirms the order and proceeds to checkout.**	
4.	**Consumer selects delivery and payment options.**	
5.		**Payment gateway processes the payment.**
6.		**System confirms the order and sends a notification to the farmer and the consumer.**

- **Extensions**:
    - **3a**: If the selected quantity exceeds availability:
        - System adjusts the quantity or suggests alternatives.
    - **5a**: If payment fails:
        - System prompts the consumer to retry or select another payment method.

- **6a**: If the system cannot confirm the order due to network issues:
  - System logs the error and informs the consumer to try again later.

Use Case 4: Update Crop Availability

- *Created by: Muhammad Tahir*
- **Scope**: Farm Connect
- **Level**: User-goal
- **Primary Actor**: Farmer
- **Stakeholders and Interests**:
  - **Farmer**: Wants to update the available quantity and types of crops.
  - **Consumers**: Want accurate real-time information on products availability.
  - **Admin**: Ensures platform reliability and that updates occur without issues.
- **Preconditions**: Farmer is logged in and has products listed.
- **Postconditions**:
  - **Success**: Crop availability is updated, and changes are visible to consumers.
  - **Failure**: Farmer is notified of the update failure, and changes are not saved.
- **Main Success Scenario**:

Step	Actor Action	System Responsibility
1.	**Farmer logs into their account.**	
2.	**Farmer navigates to the "Manage Produce" section.**	
3.	**Farmer updates the quantity or type of produce available.**	
4.		**System validates the input.**
5.		**System updates the availability and displays the updated stock on the consumer side.**

- **Extensions**:
  - **4a**: If input is invalid (e.g., negative quantity):

        - System prompts the farmer to correct the information.
    - **5a**: If there is a database error:
        - System logs the issue and notifies the farmer to try again later.

Use Case 5: Set Fair Price for Products

- *Created by: Muhammad Tahir*
- **Scope**: Farm Connect
- **Level**: User-goal
- **Primary Actor**: Farmer
- **Stakeholders and Interests**:
    - **Farmer**: Wants to set a competitive price for their crops, adhering to government guidelines.
    - **Government Regulatory System**: Ensures pricing compliance with legal mandates.
    - **Consumers**: Want fair and transparent pricing.
    - **Admin**: Ensures that prices are compliant and reflect fair market values.
- **Preconditions**: Farmer is registered and has crops listed on the platform.
- **Postconditions**:
    - **Success**: Crop price is set and approved.
    - **Failure**: Price is flagged for non-compliance, and the farmer is notified.
- **Main Success Scenario**:

Step	Actor Action	System Responsibility
1.	**Farmer logs into their account.**	
2.	**Farmer navigates to the "Set Price" section.**	
3.	**Farmer enters the price for their crops.**	
4.		**System checks the price against government-mandated rates.**
5.		**System approves the price and updates it on the platform.**

- **Extensions**:
    - **4a**: If the entered price is above or below government standards:
        - System flags the issue and prompts the farmer to revise the price.

- **5a**: If government rates are not available:
    - System logs the issue and informs the farmer that the rates will be available soon.

Use Case 6: Schedule Delivery

- *Created by: Awais Khan*
- **Scope**: Farm Connect
- **Level**: User-goal
- **Primary Actor**: Farmer
- **Stakeholders and Interests**:
    - **Farmer**: Wants to ensure timely delivery of products to the consumer.
    - **Consumer**: Expects the order to be delivered on time.
    - **Delivery Service**: Provides logistics support for transporting goods.
    - **Admin**: Ensures that delivery scheduling is smooth and error-free.
- **Preconditions**: Farmer has a confirmed order from a consumer.
- **Postconditions**:
    - **Success**: Delivery is scheduled, and notifications are sent to all relevant parties.
    - **Failure**: Farmer is notified of delivery scheduling issues, and the consumer is informed of any delays.
- **Main Success Scenario**:

Step	Actor Action	System Responsibility
1.	**Farmer logs into their account.**	
2.	**Farmer navigates to the "Orders" section and selects a confirmed order.**	
3.	**Farmer selects the "Schedule Delivery" option.**	
4.	**Farmer provides delivery details (e.g., date, time, address).**	
5.		**System contacts the delivery service to schedule the delivery.**
6.		**System confirms the delivery schedule and notifies the farmer and consumer.**

- **Extensions**:
    - **5a**: If the delivery service is unavailable:
        - System informs the farmer and suggests alternative times.
    - **6a**: If there is a conflict in scheduling:
        - System logs the issue and suggests the next available time.

Use Case 7: Manage Consumer Payments

- *Created by: Awais Khan*
- **Scope**: Farm Connect
- **Level**: User-goal
- **Primary Actor**: Admin
- **Stakeholders and Interests**:
    - **Consumer**: Wants a secure and seamless payment experience.
    - **Farmer**: Wants to receive payments without delays or issues.
    - **Payment Gateway**: Ensures secure and correct transaction processing.
    - **Admin**: Oversees and resolves payment issues, ensuring proper funds flow.
- **Preconditions**: A confirmed order is ready for payment processing.
- **Postconditions**:
    - **Success**: Payment is processed, and funds are transferred to the farmer after successful delivery.
    - **Failure**: Admin is alerted of any payment issues, and consumer is notified.
- **Main Success Scenario**:

Step	Actor Action	System Responsibility
1.	**Consumer completes an order and proceeds to checkout.**	
2.		**Admin oversees the integration of the payment gateway.**
3.	**Consumer selects a payment method.**	
4.		**Payment gateway processes the payment.**
5.		**Admin confirms the payment was successful and initiates a receipt to both the farmer and consumer.**

6.		**After delivery, funds are transferred to the farmer's account.**

- **Extensions**:
    - **4a**: If payment fails due to technical issues:
        - System informs the admin and consumer, and suggests retrying or using an alternative payment method.
    - **6a**: If the payment is delayed:
        - System logs the issue, and admin investigates to resolve the problem.

Use Case 8: Resolve Pricing Disputes

- *Created by: Awais Khan*
- **Scope**: Farm Connect
- **Level:** User-goal
- **Primary Actor:** Consumer
- **Stakeholders and Interests:**
    - **Farmer**: Wants fair pricing for their produce.
    - **Consumer**: Expects transparency and adherence to market rates.
    - **Government Regulatory System**: Ensures that government-mandated pricing is enforced.
    - **Admin**: Mediates and resolves disputes efficiently and fairly.
- **Preconditions**: A pricing dispute has been raised by either the farmer or consumer.
- **Postconditions:**
    - **Success**: Dispute is resolved, and a fair price is agreed upon.
    - **Failure**: Admin logs the failure and escalates the issue if it cannot be resolved.
- **Main Success Scenario:**

Step	Actor Action	System Responsibility
1.	**Either the farmer or consumer raises a pricing dispute.**	
2.		**Admin receives the dispute and investigates the details.**
3.		**Admin compares the set price with the government-mandated rates.**
4.	**Admin contacts both parties for clarification and attempts to resolve the issue.**	
5.	**A resolution is reached, and the price is either**	

	adjusted or maintained.	
6.		**Both parties are notified of the outcome, and the transaction proceeds.**

- **Extensions:**
    - **3a**: If the dispute arises due to an external issue (e.g., government rate discrepancies):
        - Admin notifies both parties of the external issue and informs them of the next steps.
    - **5a**: If no resolution can be found:
        - Admin escalates the issue to the government regulatory system for final judgment.

Use Case 9: Generate Sales Report for Farmers

- *Created by: Awais Khan*
- **Scope**: Farm Connect
- **Level**: User-goal
- **Primary Actor**: Farmer
- **Stakeholders and Interests**:
    - **Farmer**: Wants a detailed breakdown of sales to track revenue and performance.
    - **Admin**: Ensures that reporting tools are accurate and functional.
    - **Consumers**: Indirectly benefit from better farmer insights, leading to more reliable products availability.
- **Preconditions**: Farmer has completed at least one sale.
- **Postconditions**:
    - **Success**: Farmer receives a detailed sales report.
    - **Failure**: System informs the farmer of any errors in generating the report.
- **Main Success Scenario**:

Step	Actor Action	System Responsibility
1.	**Farmer logs into their account.**	
2.	**Farmer navigates to the "Reports" section.**	
3.	**Farmer selects a time period for the sales report.**	
4.		**System retrieves sales data and generates the report.**
5.		**Farmer views or downloads the report.**

- **Extensions**:
    - **4a**: If no sales data is available for the selected period:
        - System notifies the farmer that no data is available for that time frame.
    - **5a**: If there is an error in generating the report:
        - System logs the issue and informs the admin to resolve it.

Use Case 10: Manage Registered Mills

- *Created by: Hamza Arshad*
- **Scope**: Farm Connect
- **Level**: User-goal
- **Primary Actor**: Admin
- **Stakeholders and Interests**:
    - **Mills**: Want to register on the platform to purchase crops at fair prices and follow government regulations.
    - **Farmers**: Benefit from knowing which mills are compliant with government pricing rules.
    - **Government Regulatory System**: Requires registered mills for easier enforcement of pricing laws.
    - **Admin**: Ensures mills are registered, verified, and compliant with the rules.
- **Preconditions**: Mill applies for registration on the platform.
- **Postconditions**:
    - **Success**: Mill is registered, verified, and able to purchase crops from farmers.
    - **Failure**: Mill is rejected or flagged for not adhering to government regulations.
- **Main Success Scenario**:

Step	Actor Action	System Responsibility
	Mill submits a registration form on the platform.	
		Admin reviews the details provided by the mill.
		Admin checks if the mill complies with government regulations and pricing policies.

		Admin approves the registration, and the mill is added to the platform.
		Mill can now purchase crops from farmers.

- **Extensions**:
  - **3a**: If the mill's details do not meet regulatory standards:
    - Admin rejects the registration and notifies the mill with a reason for the rejection.
  - **5a**: If the mill is flagged for any legal violations post-registration:
    - Admin temporarily suspends the mill's account until the issue is resolved.

Use Case 11: Provide Feedback

- *Created by: Hamza Arshad*

- **Scope:** Farm Connect
- **Level:** User-goal
- **Primary Actor:** Consumer, Farmer, Mill
- **Stakeholders and Interests:**
  - Consumer: Wants to share their experience with the purchase and delivery of products.
  - Farmer: Desires constructive feedback to improve crop quality, delivery, and overall service.
  - Mill: Wants to provide feedback on the quality of products and the reliability of farmers.
  - Admin: Monitors feedback to ensure fairness, prevent fake reviews, and enhance platform credibility**.**
- **Preconditions:**
  - The user has completed a transaction (order fulfilled, product delivered).
  - The user is logged into their account.
- **Postconditions:**
  - Success: Feedback is successfully submitted and visible to relevant parties (e.g., other consumers, farmers).
  - Failure: User is notified of any errors in submitting feedback and is prompted to try again later.
- **Main Success Scenario:**

Step	Actor Action	System Responsibility
1.	**The user logs into their account after completing a transaction.**	
2.		**The system prompts the user to provide feedback for the transaction.**
3.	**The user navigates to the "Provide Feedback" section for the relevant order.**	
4.	**The user rates their experience and enters comments regarding the transaction (quality of products, timeliness of delivery, service experience).**	
5.		**The system validates the feedback and submits it.**
6.		**The feedback becomes visible on the platform to other users, farmers, or mills, depending on the nature of the transaction.**

- **Extensions:**
  - 4a: If the feedback contains inappropriate language or offensive content:
    - The system flags the feedback for admin review and temporarily withholds it from being posted.
  - 5a: If the system encounters a server error:
    - The user is informed of the issue and prompted to try submitting the feedback later.

Use Case 1: Reviews History

Created by: Hamza Arshad

Scope: Farm Connect
Level: User-goal

Primary Actor:

- Consumer
- Farmer

Stakeholders and Interests:

- **Consumers:** Want to view the past reviews and ratings of farmers to make informed decisions when purchasing produce.
- **Farmers:** Want to review the past reviews and ratings of consumers to decide whether to engage in future transactions.

Preconditions:

- The user (either a farmer or consumer) must be logged into the Farm Connect .
- Past reviews must exist between the user and the respective parties.

Postconditions:

- **Success:** The user (farmer or consumer) can successfully view the review history of the other party.
- **Failure:** No review history is displayed if there are no reviews, or if there's an issue retrieving data.

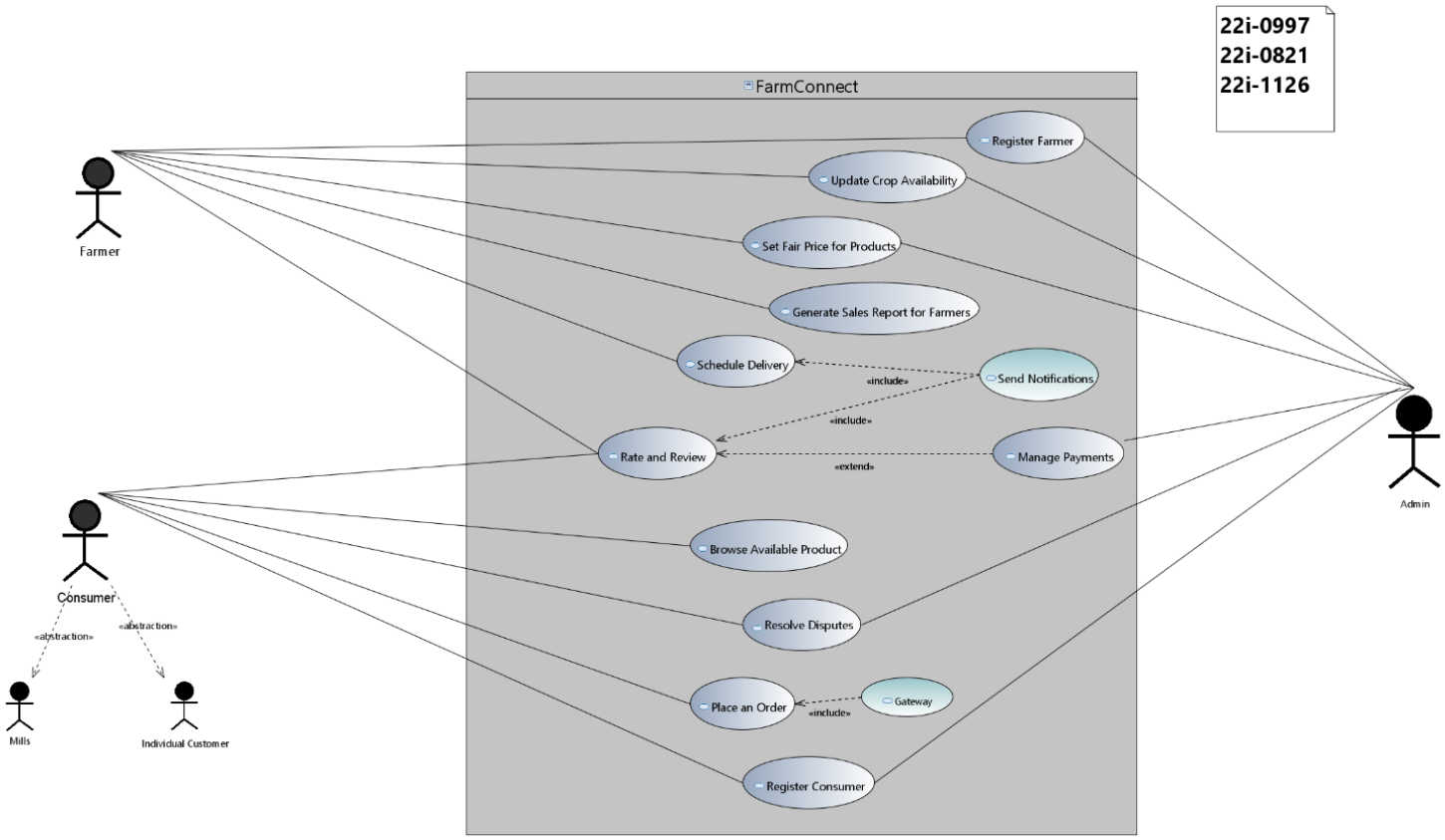
Main Success Scenario:

Step	Actor Action	System Responsibility
1.	**User navigates to the "Reviews History" section in their dashboard.**	
2.	**User selects whether they want to view the review history of farmers (for consumers) or consumers (for farmers).**	
3.		**The system fetches the past review records for the selected parties.**
4.	**The user views the reviews, which may include feedback, ratings, and any other relevant details.**	
5.	**User can filter reviews based on specific criteria, such as date range or rating.**	

Extensions:

- **No Reviews Available:**
  4a. If no reviews exist for the selected party, the system displays a "No reviews available" message.
- **System Error While Retrieving Reviews:**
  4b. If there is a technical issue retrieving reviews, the system prompts the user with an error message and suggests trying again later.

2.5 Use Case Diagram



3. Other Nonfunctional Requirements

3.1 Performance Requirements

FarmConnect must handle up to 500 simultaneous users with response times under 3 seconds for core operations like browsing products or placing orders.

3.2 Safety Requirements

The platform ensures data integrity during transactions and secure handling of sensitive information, including payment details and user credentials.

3.3 Security Requirements

- Encrypted communication between all layers using HTTPS.
- Role-based access control to protect sensitive operations.

3.4 Software Quality Attributes

- **Usability:** A responsive, user-friendly interface accessible on both mobile and desktop.
- **Reliability:** Consistent performance during peak usage.
- **Scalability:** Adaptable to future increases in user base and functionality.
- **Maintainability:** Modular codebase for ease of updates and debugging.

3.5 Business Rules

- Farmers must provide valid identification to register.
- Mills are required to comply with government pricing regulations.
- Disputes must be resolved within 7 business days.

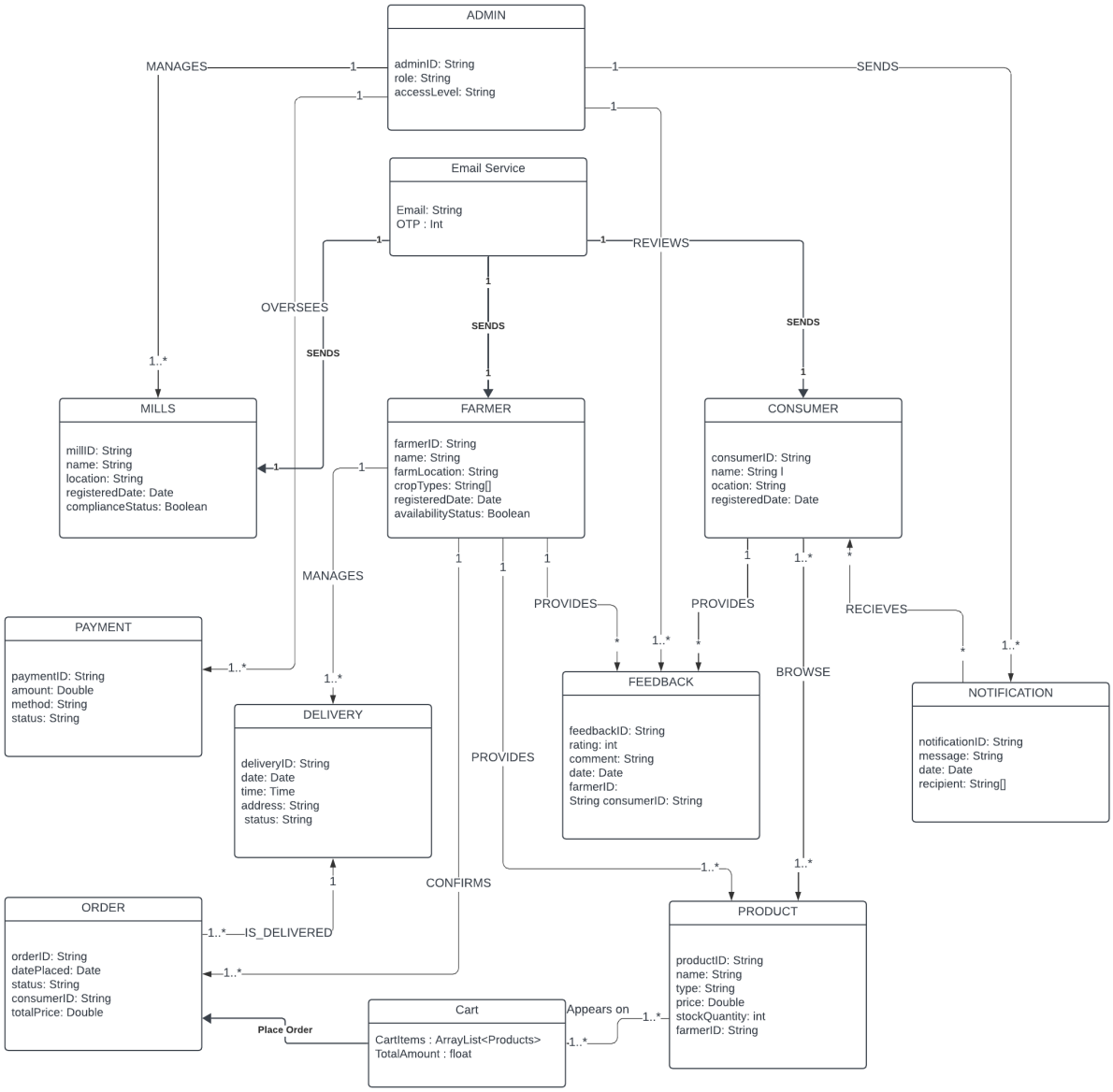
3.6 Operating Environment

The platform operates on a web-based environment, accessible via modern browsers, and integrates with Android and iOS devices.

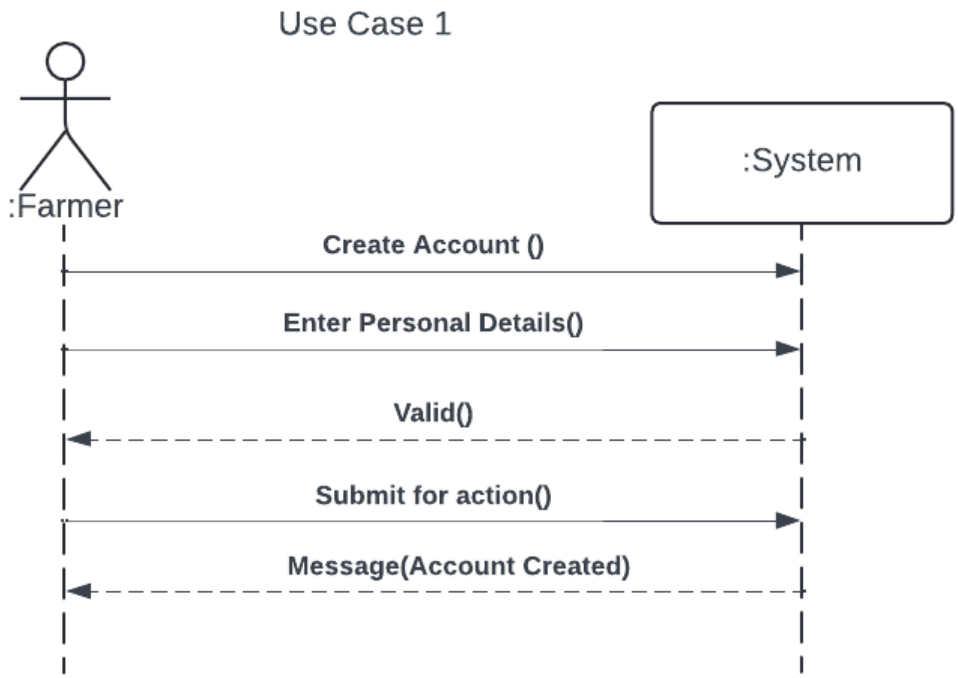
3.7 User Interfaces

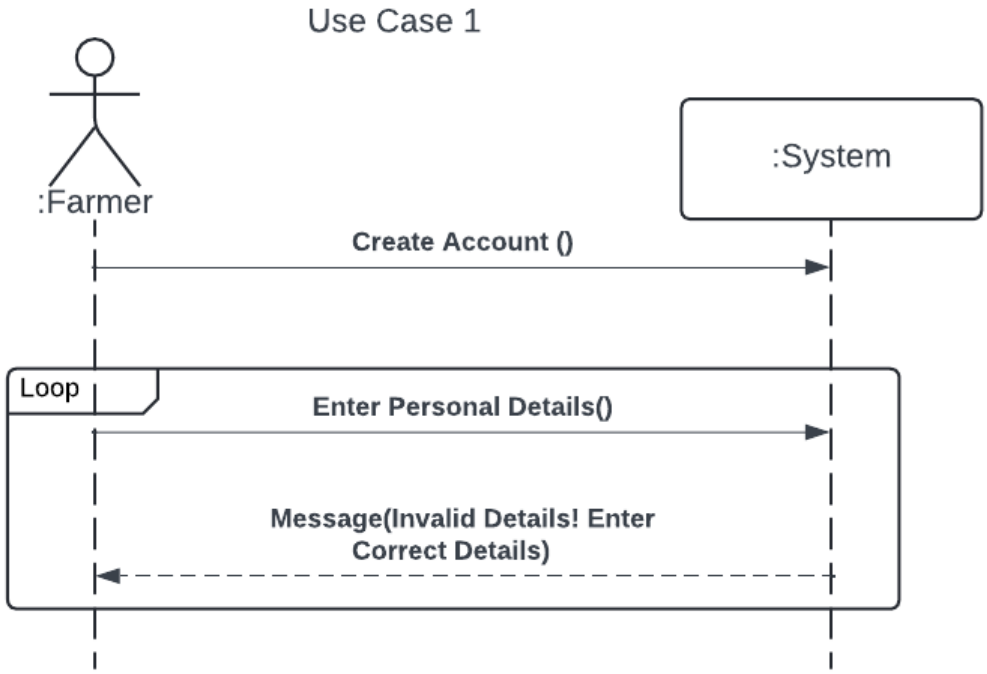
The Next.js frontend offers intuitive navigation, with sections for browsing products, placing orders, and managing user profiles. Sample UI designs and layouts will follow standard accessibility guidelines.

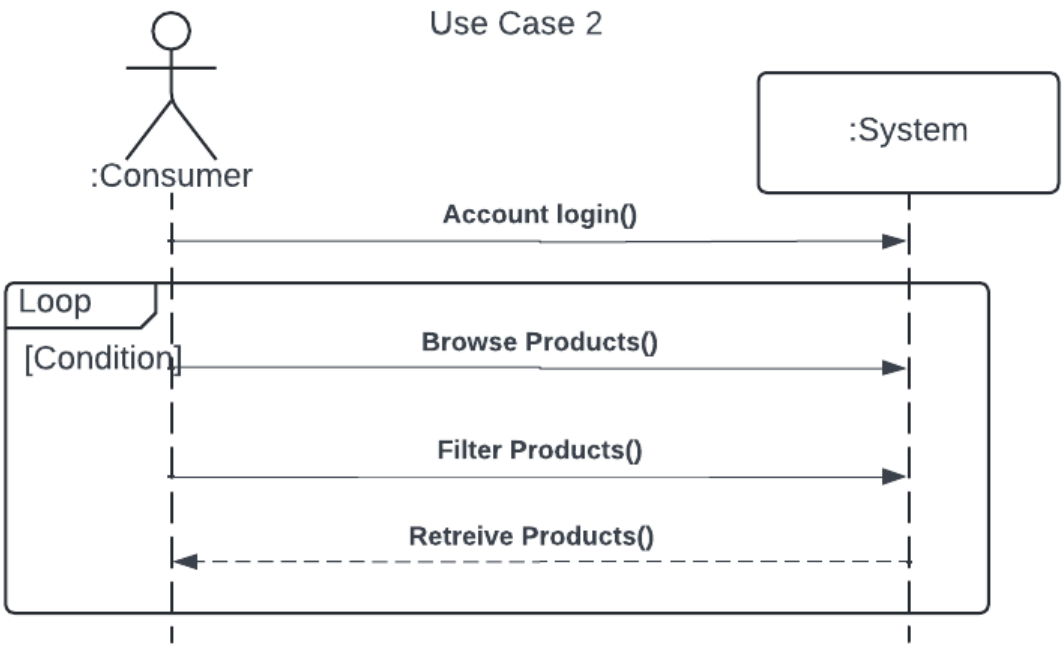
4. Domain Model

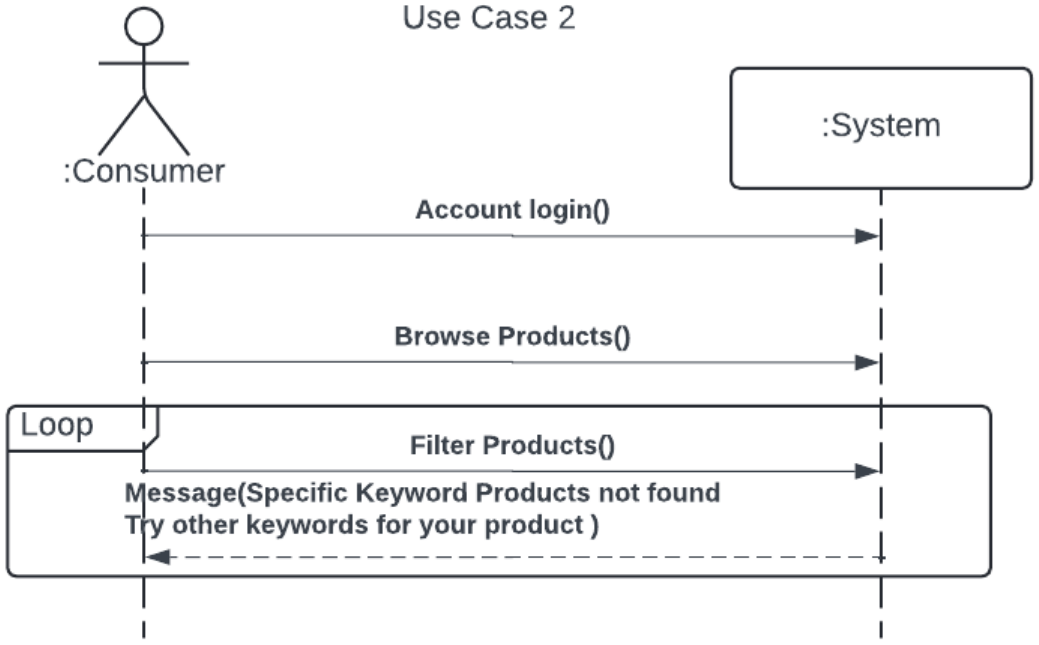


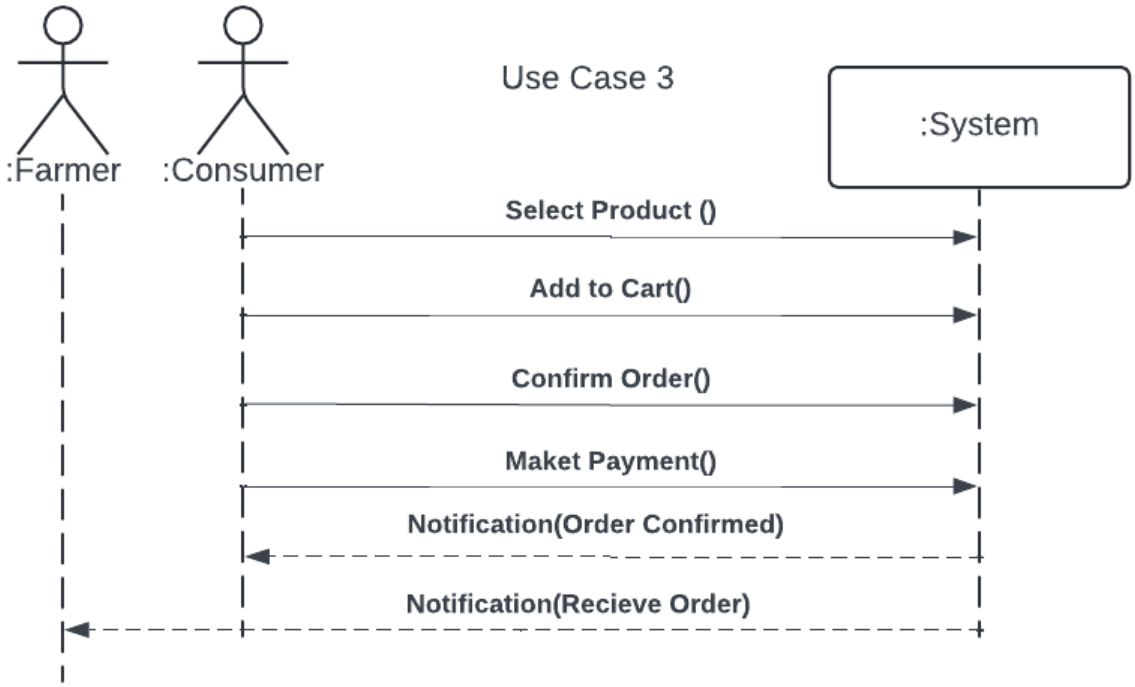
5. System Sequence Diagram

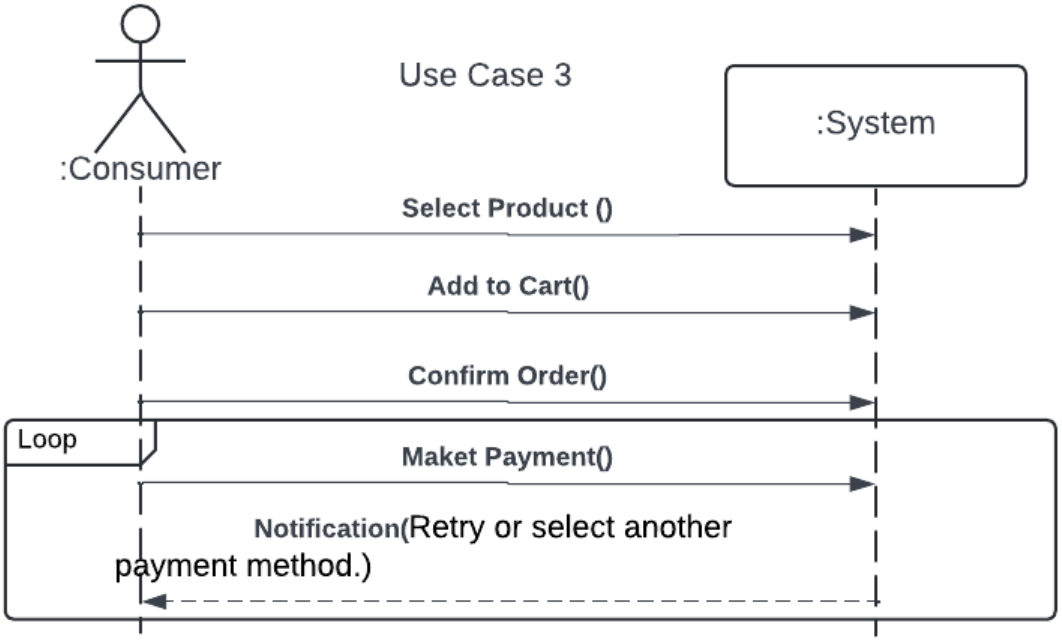


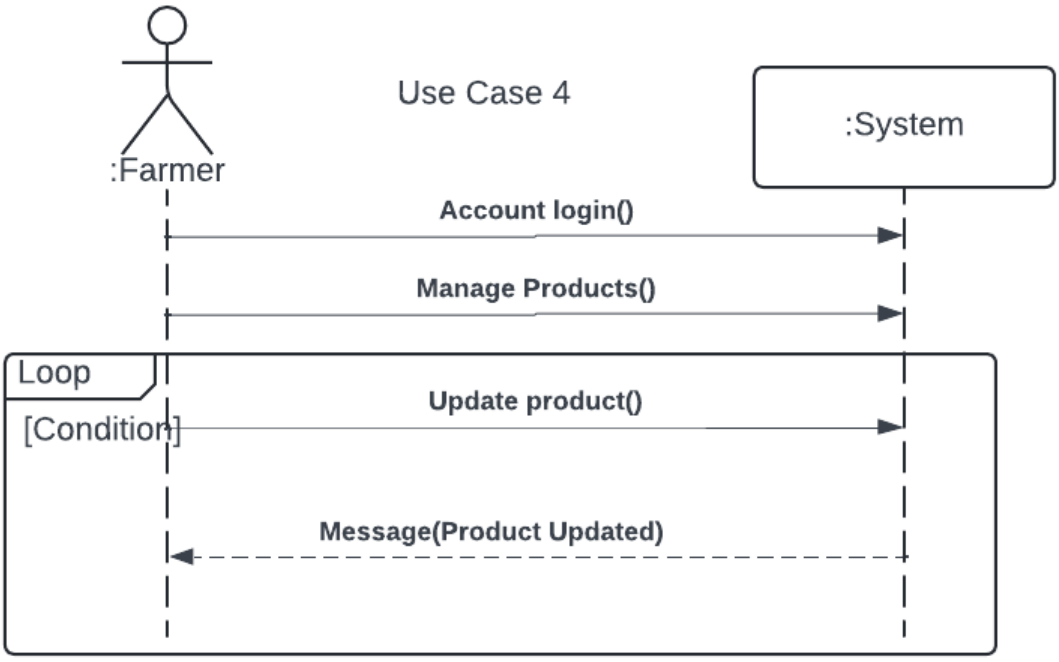


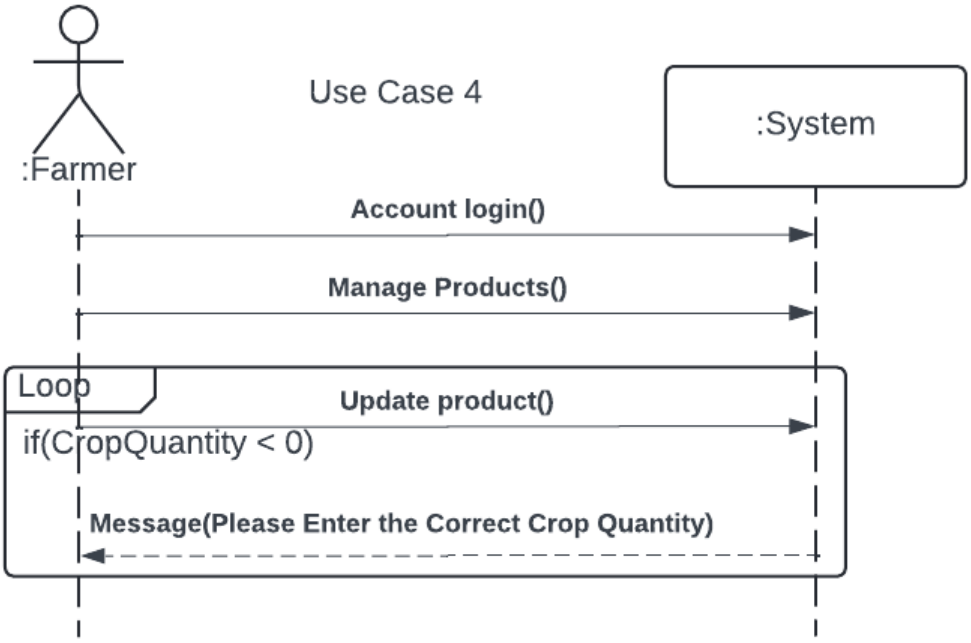

Use Case 2
:Consumer
:System
Account login()
Loop
[Condition]
Browse Products()
Filter Products()
Retreive Products()

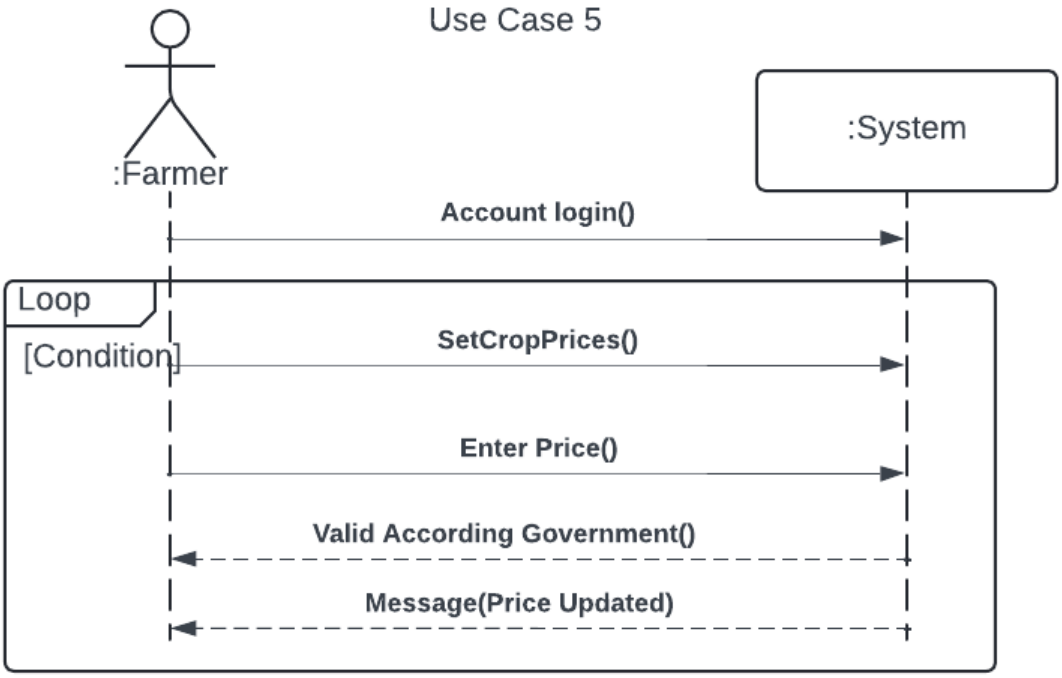

Use Case 2
:Consumer
:System
Account login()
Browse Products()
Loop
Filter Products()
Message(Specific Keyword Products not found
Try other keywords for your product)

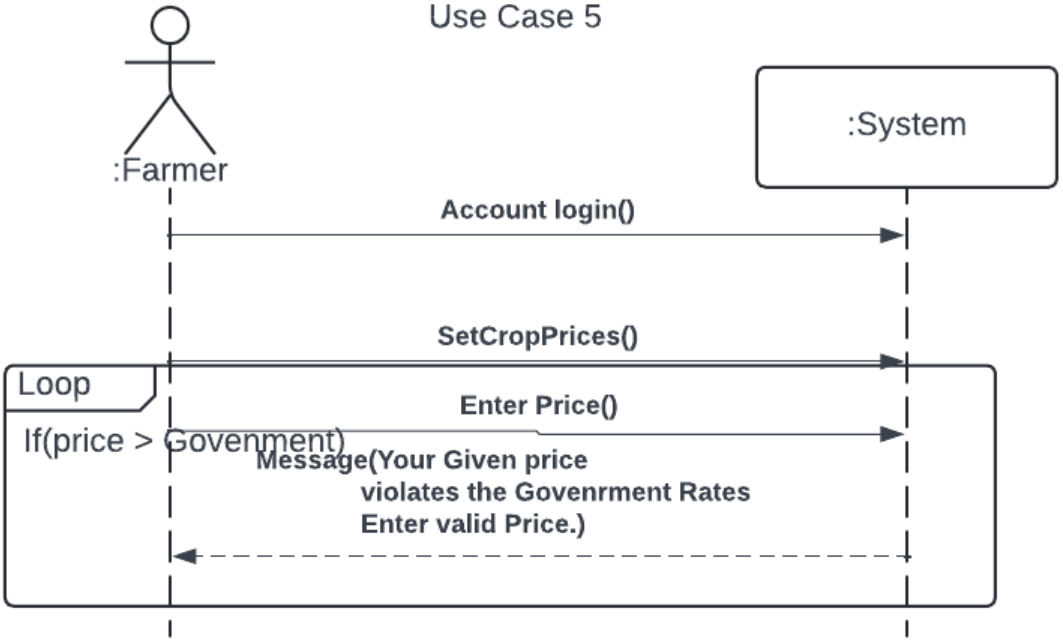

Use Case 3
:Farmer
:Consumer
:System
Select Product ()
Add to Cart()
Confirm Order()
Maket Payment()
Notification(Order Confirmed)
Notification(Recieve Order)

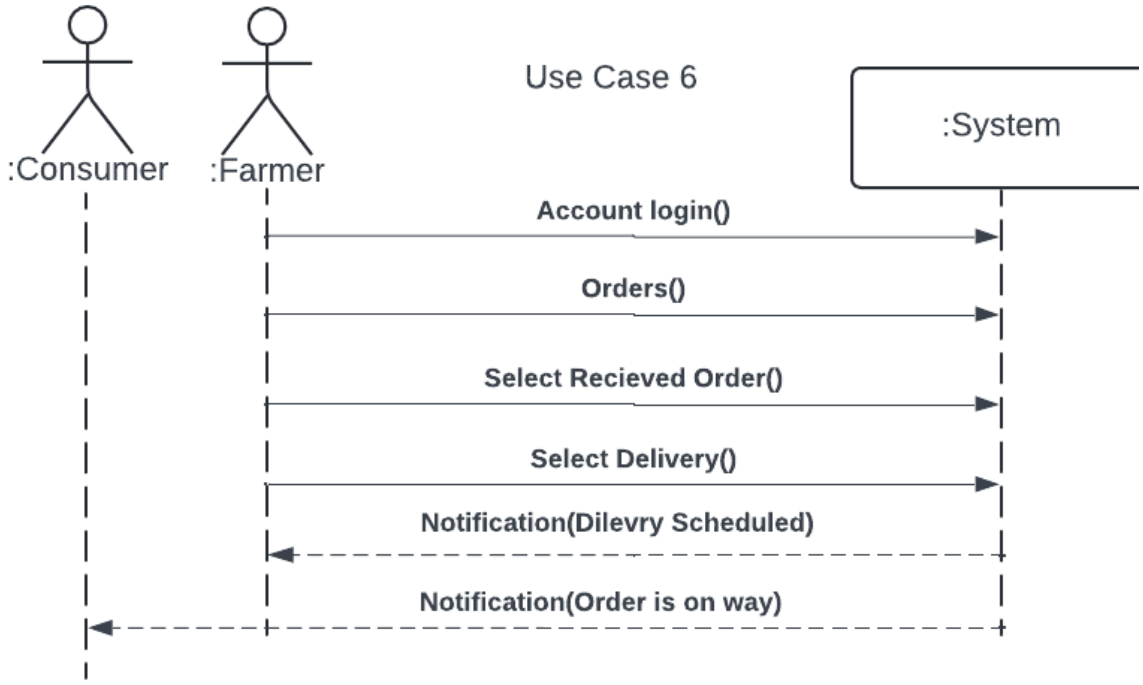

Use Case 3
:Consumer
:System
Select Product ()
Add to Cart()
Confirm Order()
Loop
Maket Payment()
Notification(Retry or select another payment method.)

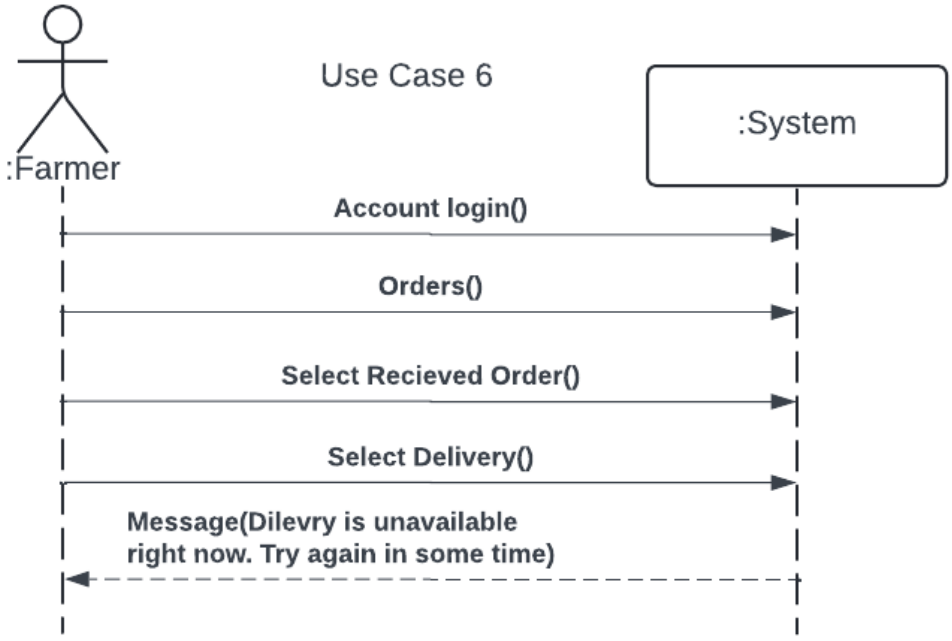

Use Case 4
:Farmer
:System
Account login()
Manage Products()
Loop
[Condition]
Update product()
Message(Product Updated)

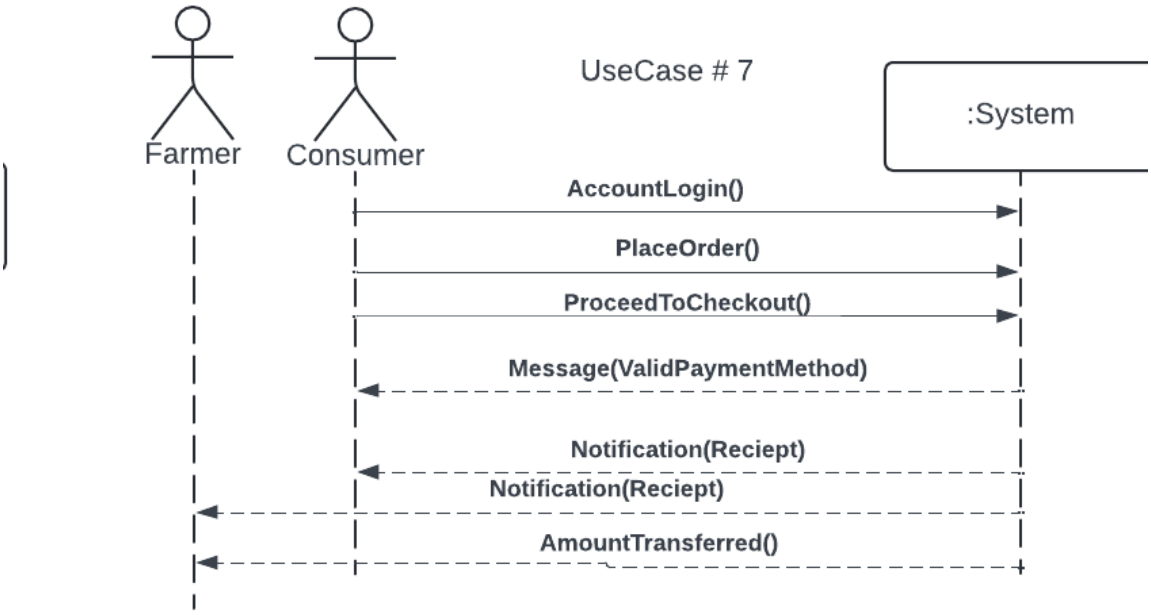

Use Case 4
:Farmer
:System
Account login()
Manage Products()
Loop
if(CropQuantity < 0)
Update product()
Message(Please Enter the Correct Crop Quantity)

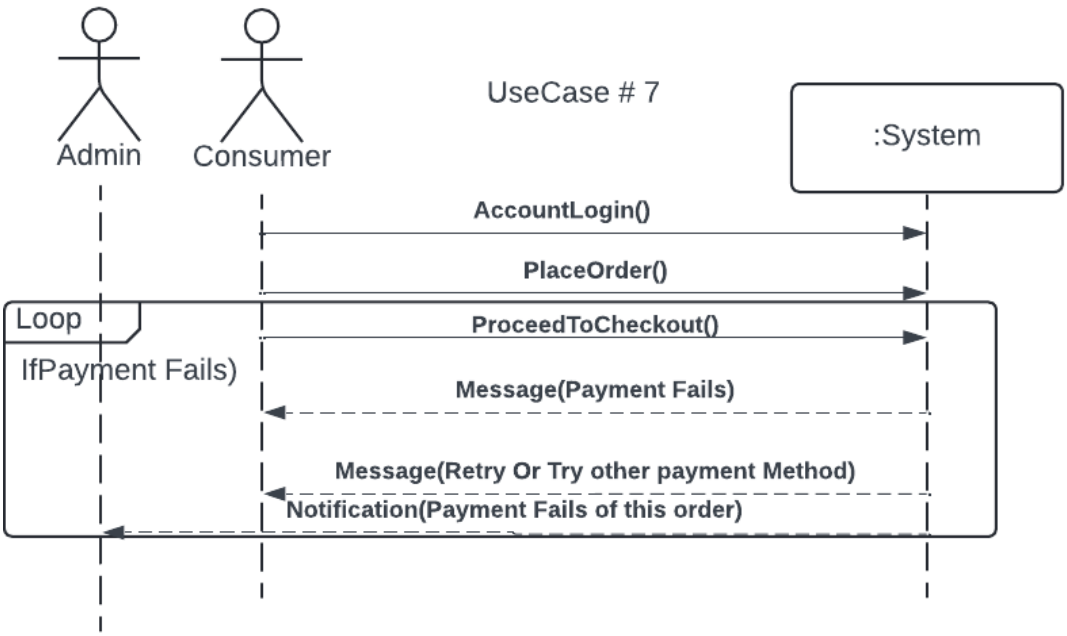

Use Case 5
:Farmer
:System
Account login()
Loop
[Condition]
SetCropPrices()
Enter Price()
Valid According Government()
Message(Price Updated)

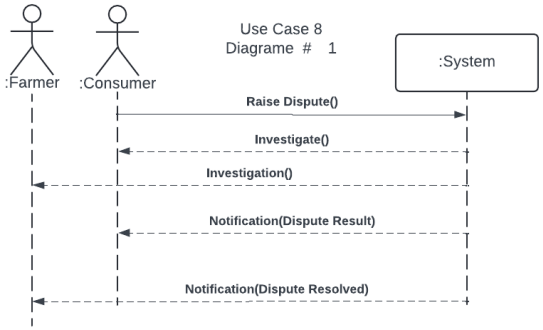

Use Case 5
:Farmer
:System
Account login()
SetCropPrices()
Loop
Enter Price()
If(price > Govenment)
Message(Your Given price
violates the Govenrment Rates
Enter valid Price.)

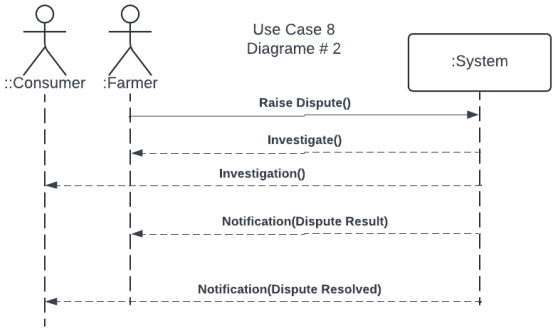

Use Case 6
:Consumer
:Farmer
:System
Account login()
Orders()
Select Recieved Order()
Select Delivery()
Notification(Dilevry Scheduled)
Notification(Order is on way)

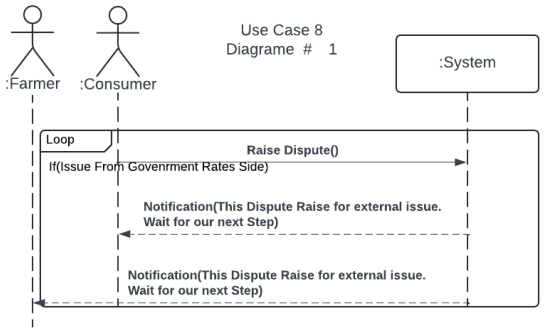

Use Case 6
:Farmer
:System
Account login()
Orders()
Select Recieved Order()
Select Delivery()
Message(Dilevry is unavailable
right now. Try again in some time)

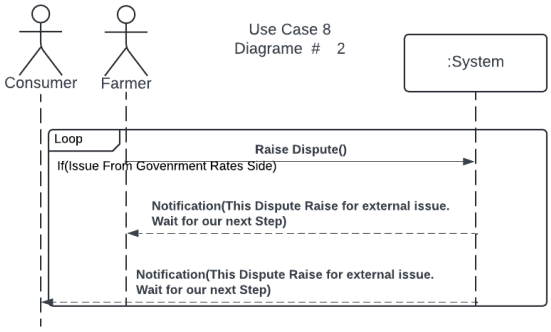

Farmer
Consumer
UseCase # 7
:System
AccountLogin()
PlaceOrder()
ProceedToCheckout()
Message(ValidPaymentMethod)
Notification(Reciept)
Notification(Reciept)
AmountTransferred()

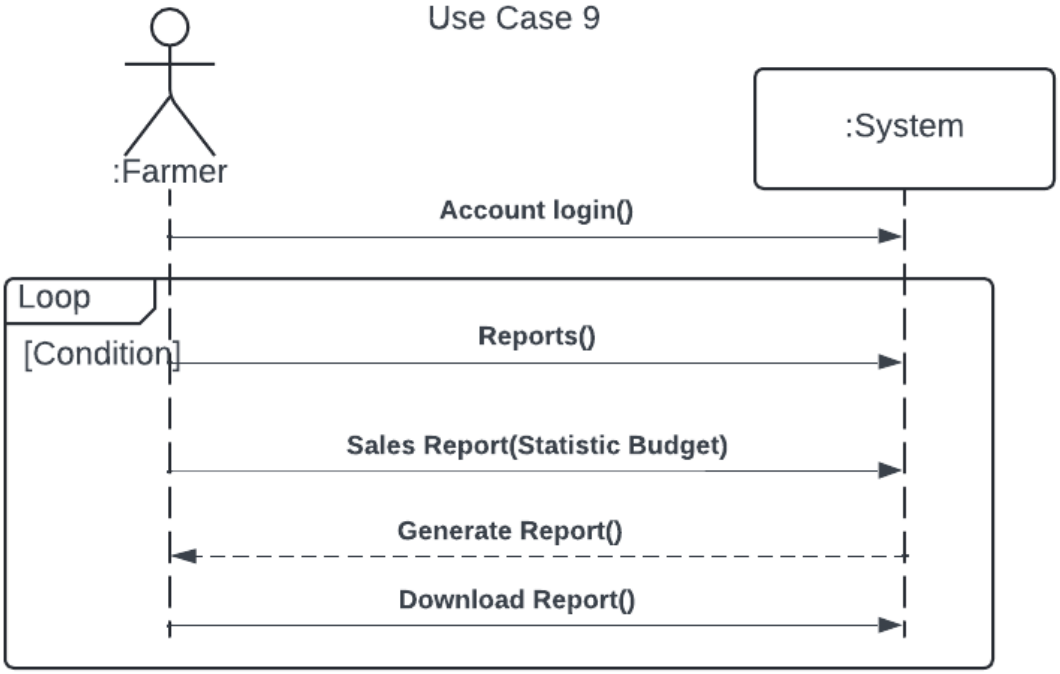

Admin
Consumer
UseCase # 7
:System
AccountLogin()
PlaceOrder()
Loop
IfPayment Fails)
ProceedToCheckout()
Message(Payment Fails)
Message(Retry Or Try other payment Method)
Notification(Payment Fails of this order)

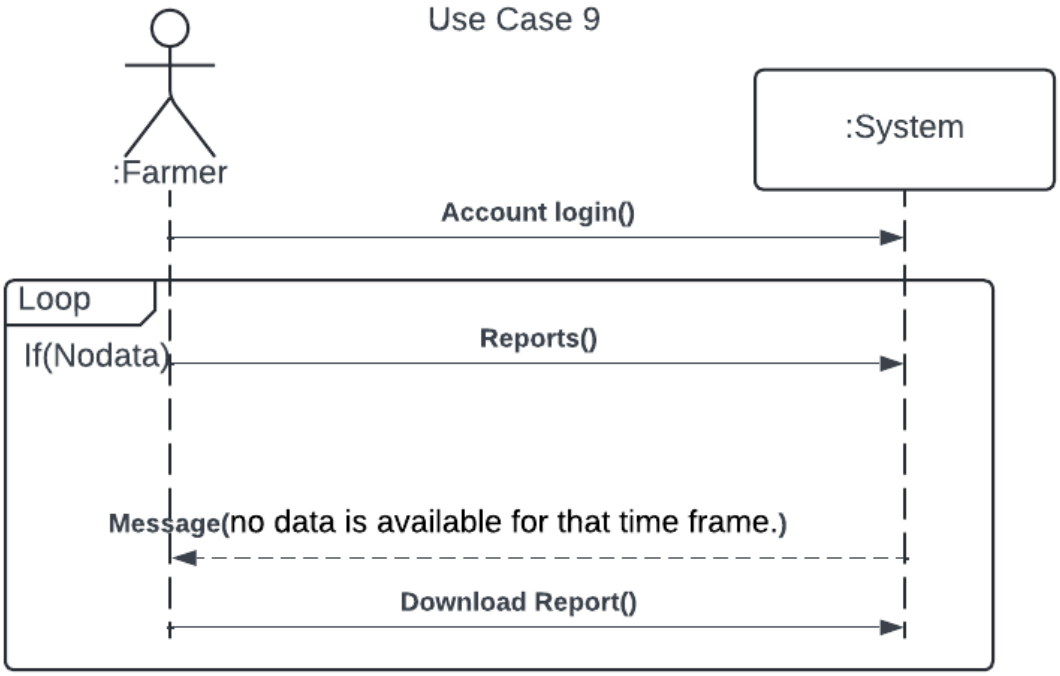

Use Case 8
Diagrame # 1
:Farmer
:Consumer
:System
Raise Dispute()
Investigate()
Investigation()
Notification(Dispute Result)
Notification(Dispute Resolved)

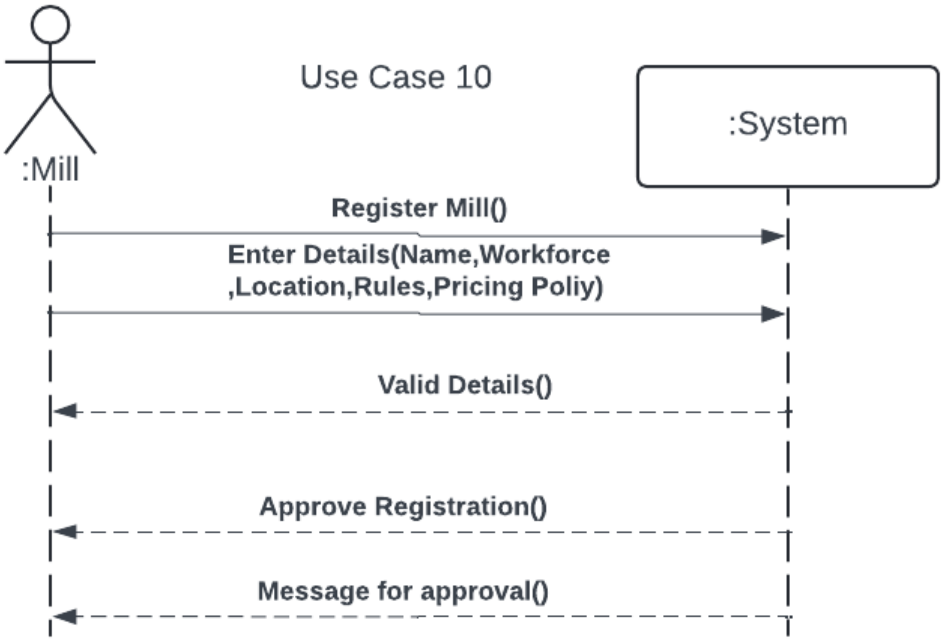

Use Case 8
Diagrame # 2
::Consumer
:Farmer
:System
Raise Dispute()
Investigate()
Investigation()
Notification(Dispute Result)
Notification(Dispute Resolved)


Use Case 8
Diagrame # 1
:Farmer
:Consumer
:System
Loop
Raise Dispute()
If(Issue From Govenrment Rates Side)
Notification(This Dispute Raise for external issue.
Wait for our next Step)
Notification(This Dispute Raise for external issue.
Wait for our next Step)

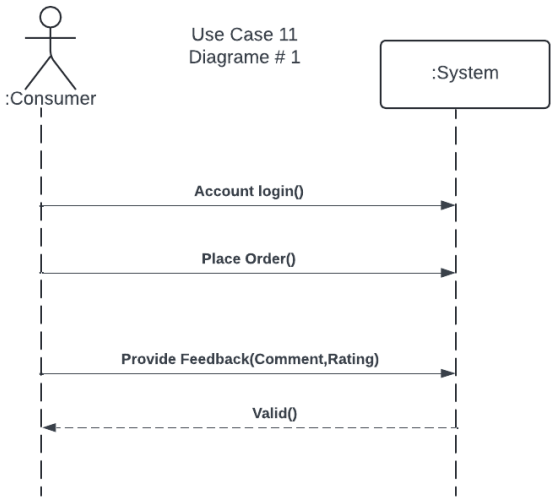

Use Case 8
Diagrame # 2
Consumer
Farmer
:System
Loop
Raise Dispute()
If(Issue From Govenrment Rates Side)
Notification(This Dispute Raise for external issue.
Wait for our next Step)
Notification(This Dispute Raise for external issue.
Wait for our next Step)

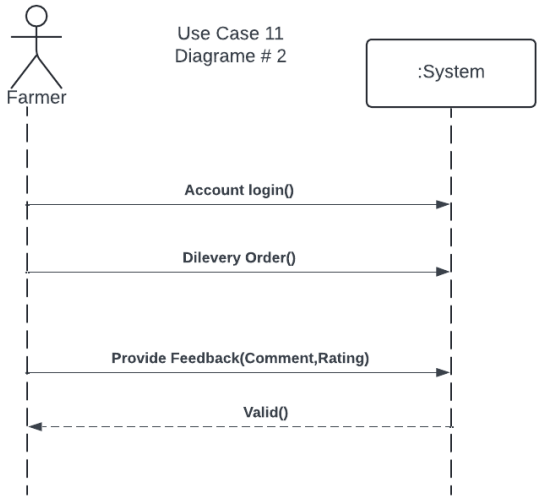

Use Case 9
:Farmer
:System
Account login()
Loop
[Condition]
Reports()
Sales Report(Statistic Budget)
Generate Report()
Download Report()

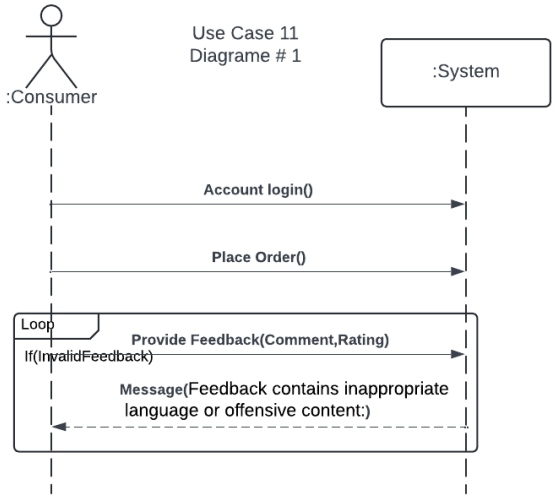

Use Case 9
:Farmer
:System
Account login()
Loop
If(Nodata)
Reports()
Message(no data is available for that time frame.)
Download Report()

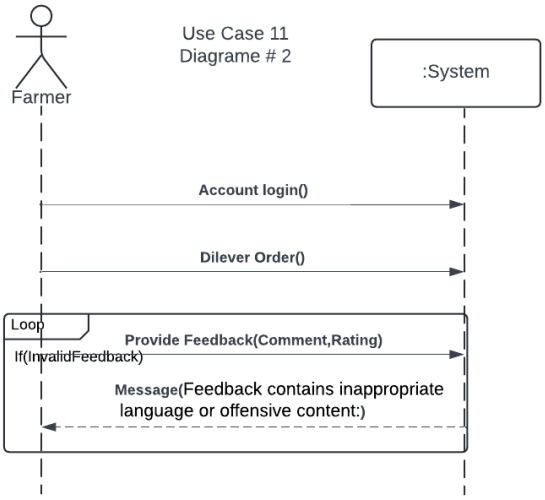

Use Case 10
:Mill
:System
Register Mill()
Enter Details(Name,Workforce
,Location,Rules,Pricing Poliy)
Valid Details()
Approve Registration()
Message for approval()

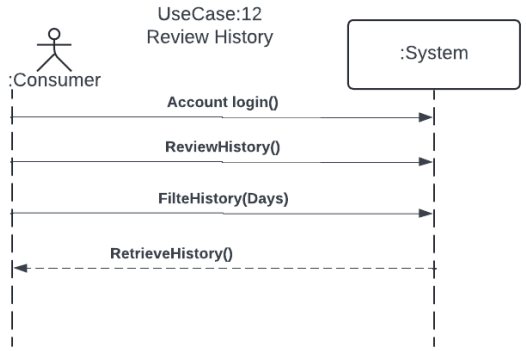

Use Case 10
:Mill
:System
Register Mill()
Loop
If(InvalidDetails)
Enter Details(Name,Workforce
,Location,Rules,Pricing Poliy)
Reject Registration()
Message(The mill's details do not
meet regulatory standards)

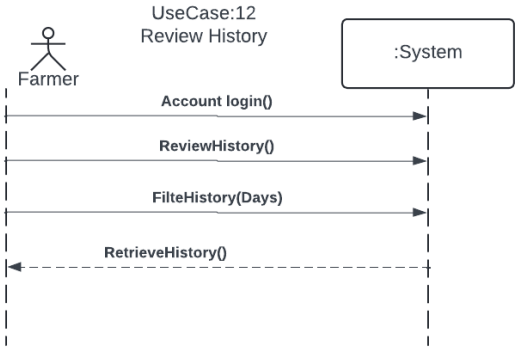

Use Case 11
Diagrame # 1
:Consumer
:System
Account login()
Place Order()
Provide Feedback(Comment,Rating)
Valid()

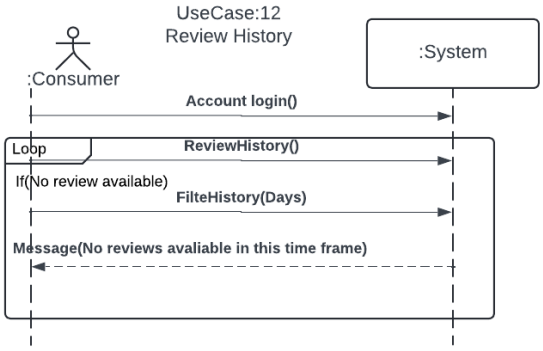

Use Case 11
Diagrame # 2
Farmer
:System
Account login()
Dilevery Order()
Provide Feedback(Comment,Rating)
Valid()

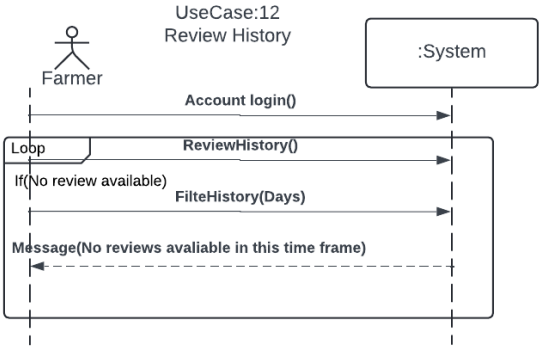

Use Case 11
Diagrame # 1
:Consumer
:System
Account login()
Place Order()
Loop
If(InvalidFeedback)
Provide Feedback(Comment,Rating)
Message(Feedback contains inappropriate language or offensive content:)


Use Case 11
Diagrame # 2
Farmer
:System
Account login()
Dilever Order()
Loop
If(InvalidFeedback)
Provide Feedback(Comment,Rating)
Message(Feedback contains inappropriate language or offensive content:)



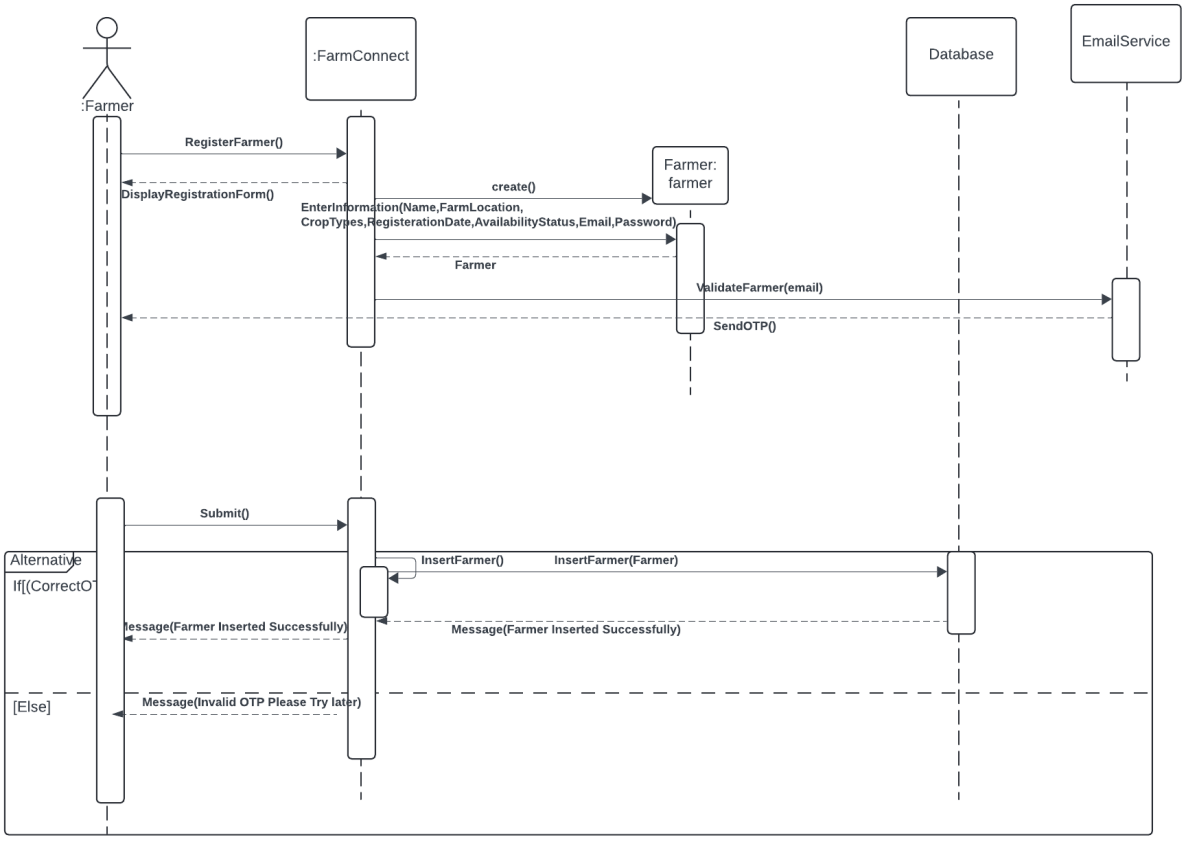




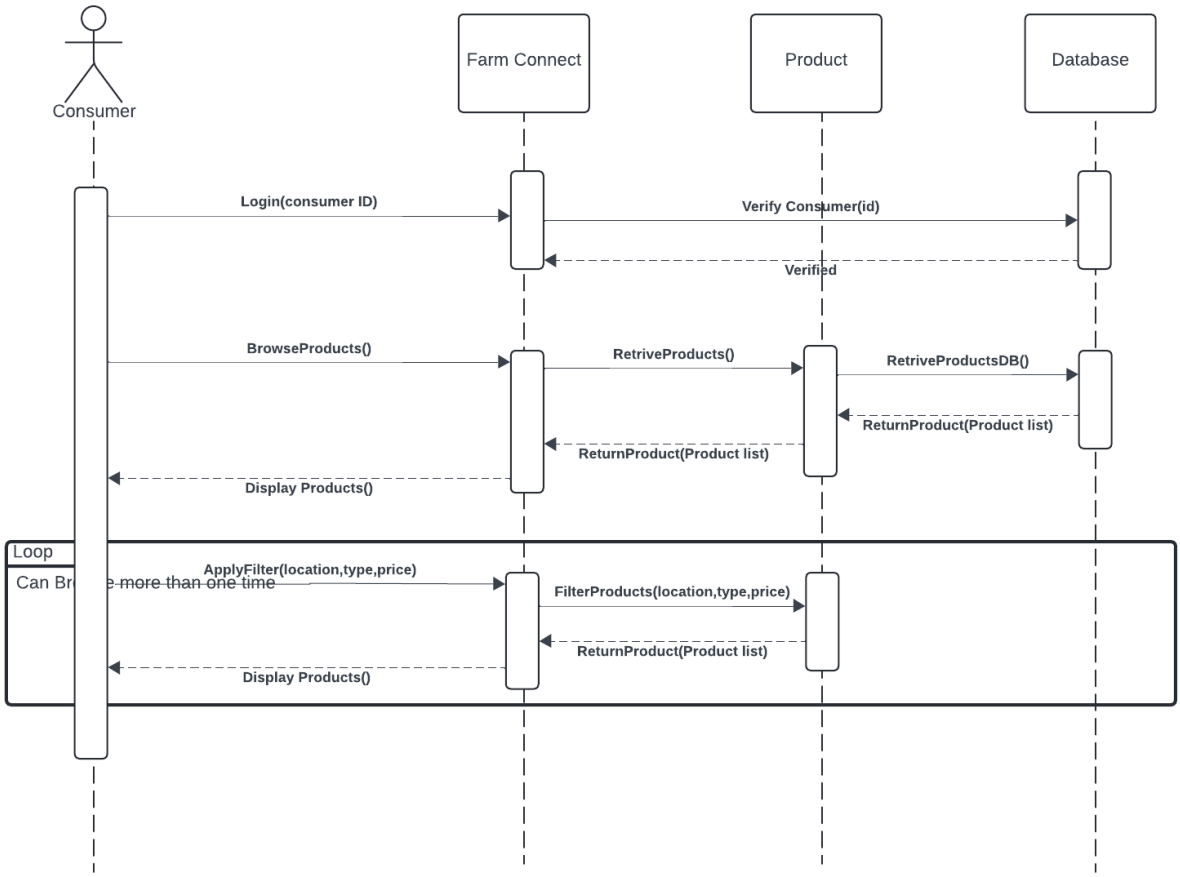


6. Sequence Diagram

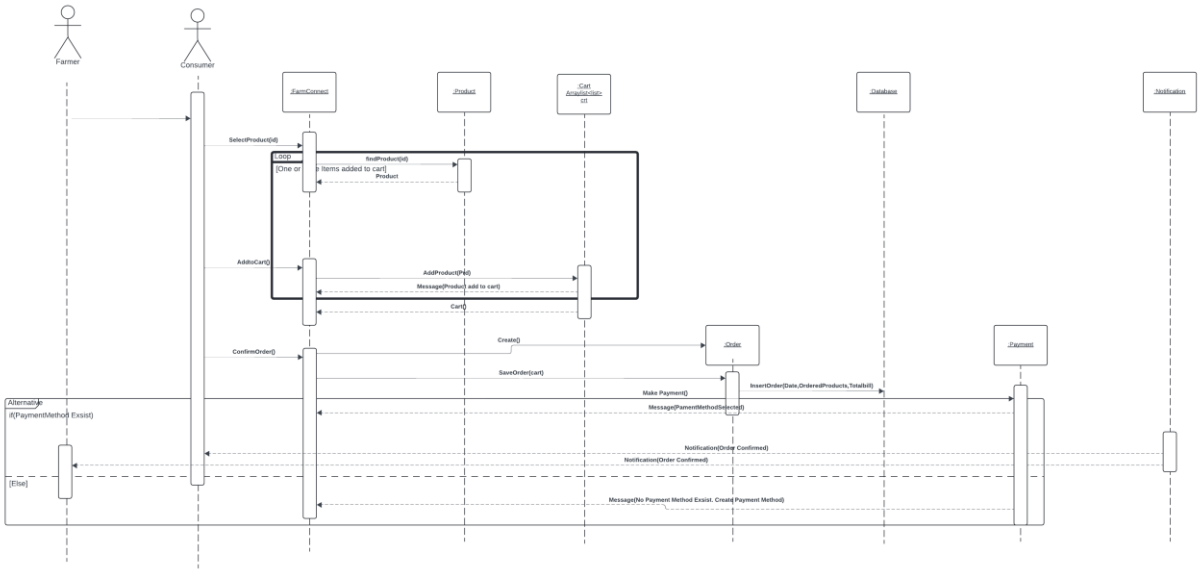
Usecase-1


:Farmer
:FarmConnect
Database
EmailService
Farmer: farmer
RegisterFarmer()
DisplayRegistrationForm()
create()
EnterInformation(Name,FarmLocation,
CropTypes,RegisterationDate,AvailabilityStatus,Email,Password)
Farmer
ValidateFarmer(email)
SendOTP()
Submit()
Alternative
If[(CorrectO
InsertFarmer()
InsertFarmer(Farmer)
Message(Farmer Inserted Successfully)
Message(Farmer Inserted Successfully)
[Else]
Message(Invalid OTP Please Try later)

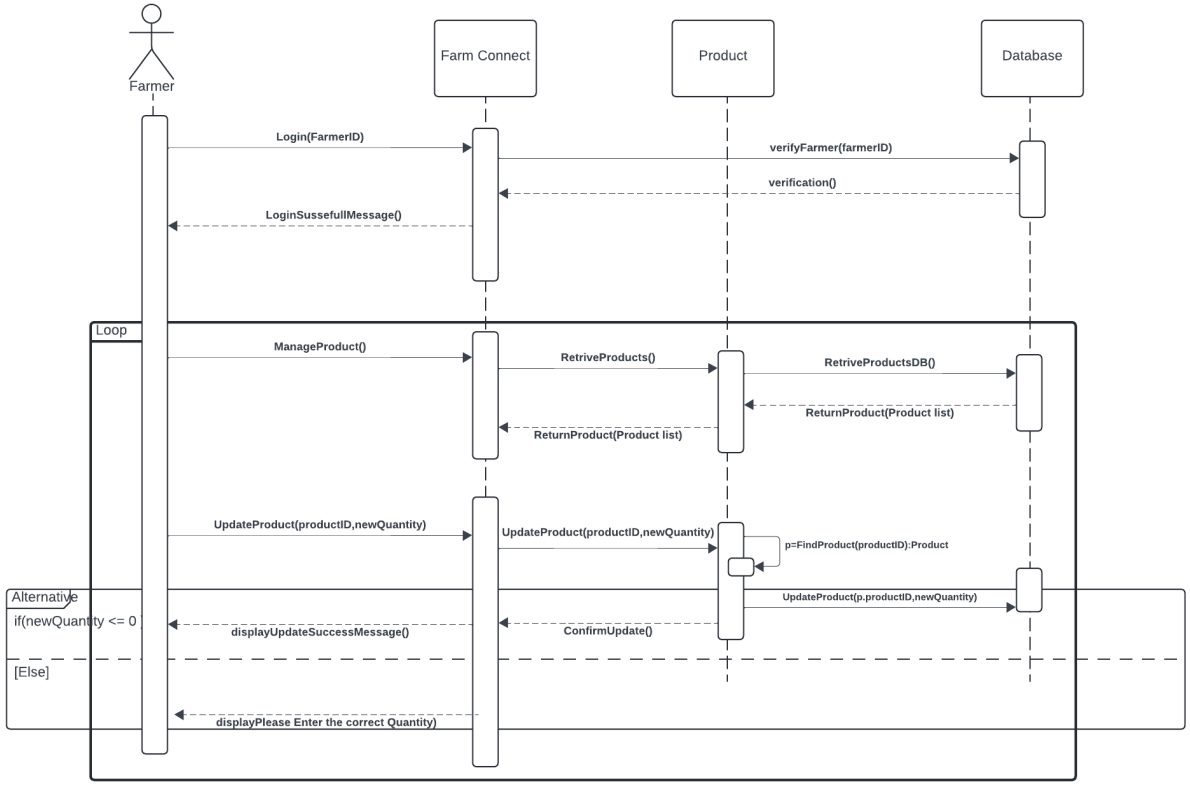
Usecase-2


Consumer
Farm Connect
Product
Database
Login(consumer ID)
Verify Consumer(id)
Verified
BrowseProducts()
RetriveProducts()
RetriveProductsDB()
ReturnProduct(Product list)
ReturnProduct(Product list)
Display Products()
Loop
Can Br e more than one time
ApplyFilter(location,type,price)
FilterProducts(location,type,price)
ReturnProduct(Product list)
Display Products()

Usecase-3

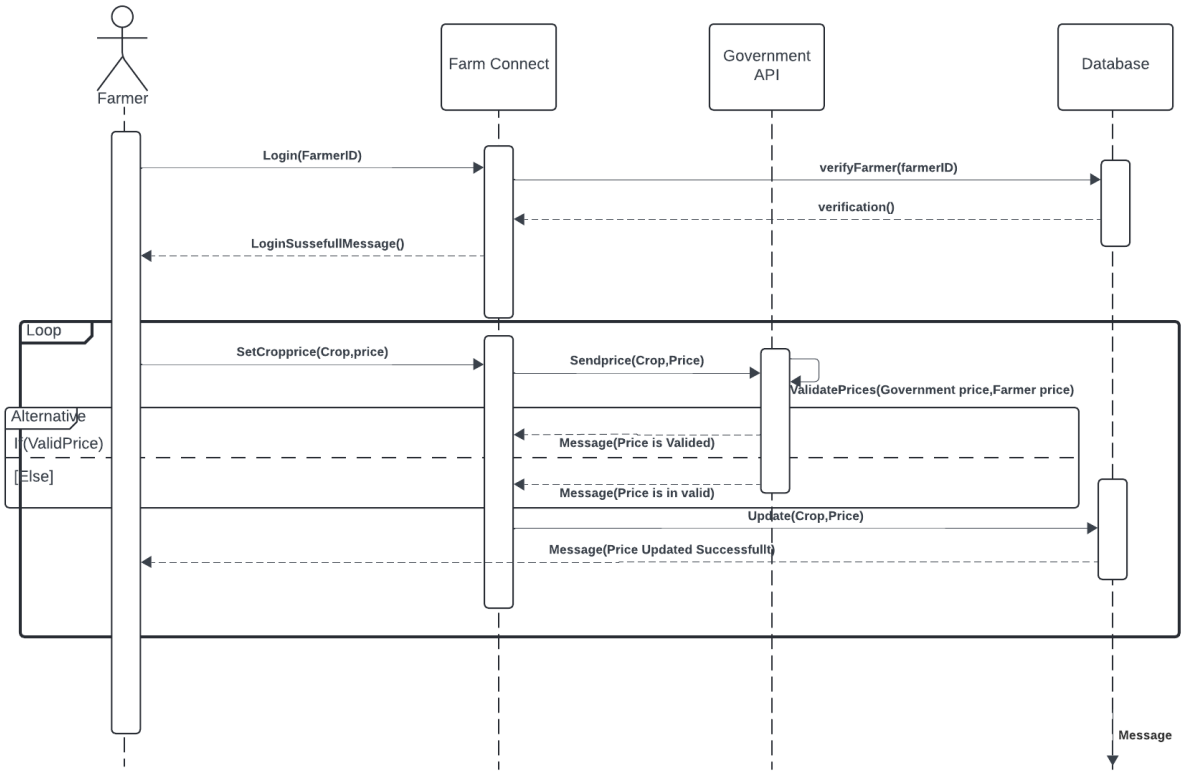


Usecase4

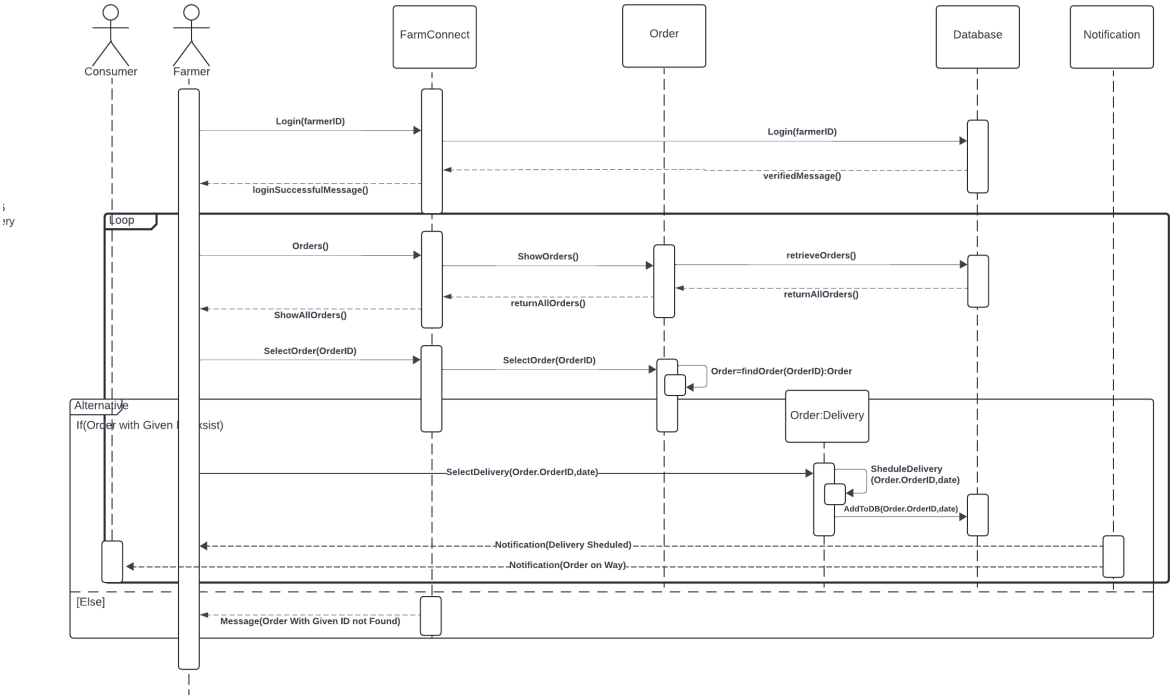


Usecase5

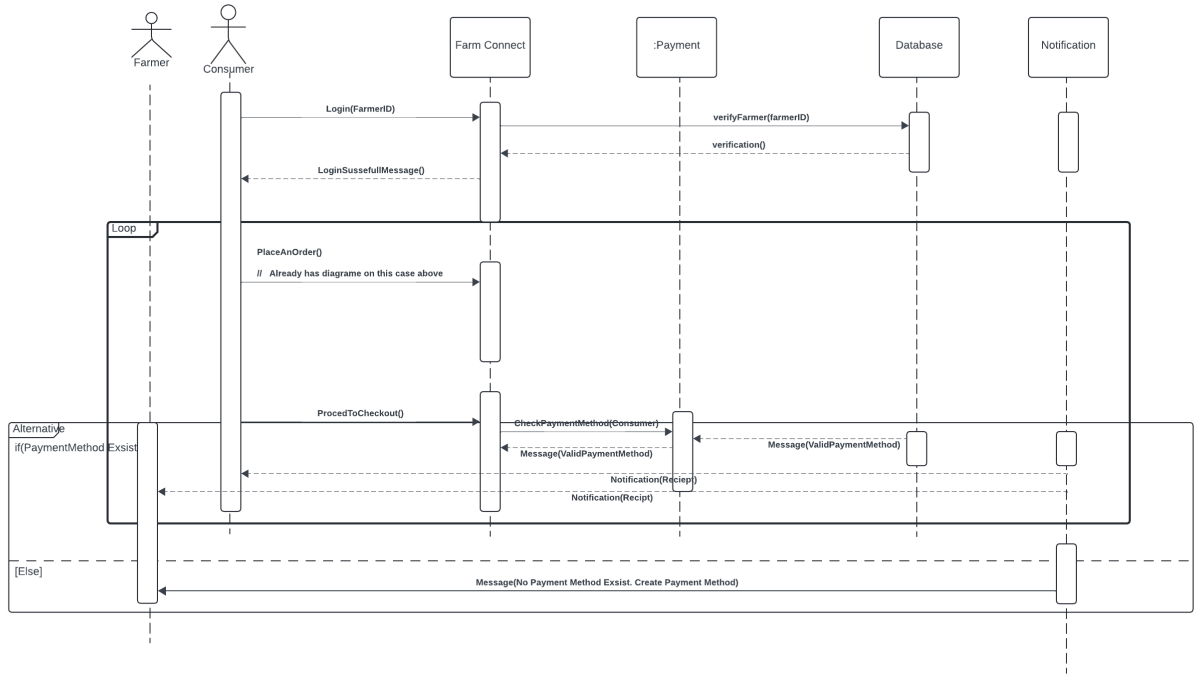
5


Farmer
Farm Connect
Government API
Database
Login(FarmerID)
verifyFarmer(farmerID)
verification()
LoginSussefullMessage()
Loop
SetCropprice(Crop,price)
Sendprice(Crop,Price)
ValidatePrices(Government price,Farmer price)
Alternative
If(ValidPrice)
Message(Price is Valided)
[Else]
Message(Price is in valid)
Update(Crop,Price)
Message(Price Updated Successfullt)
Message

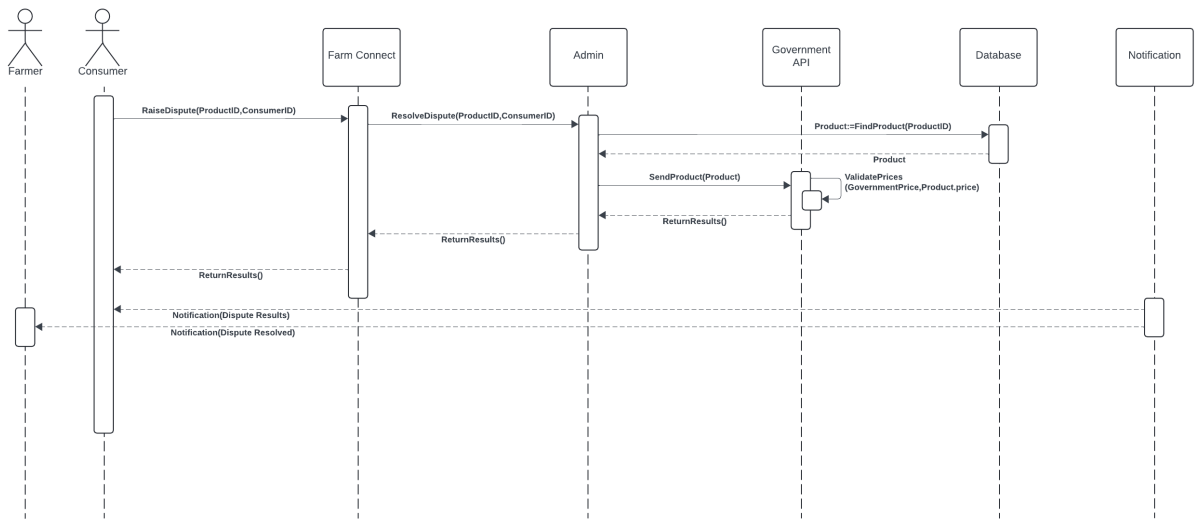
Usecase6



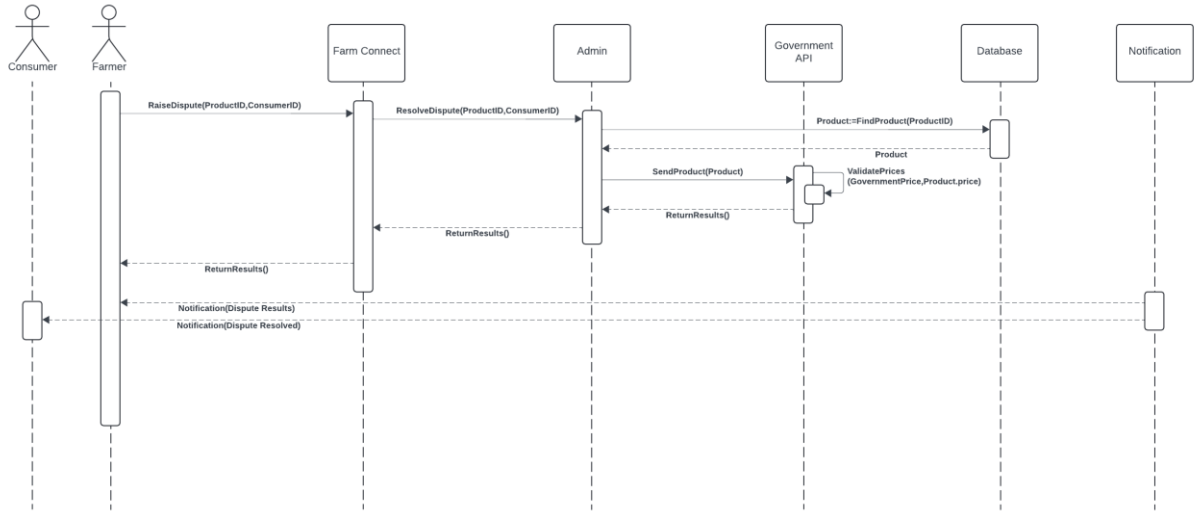
Usecase7



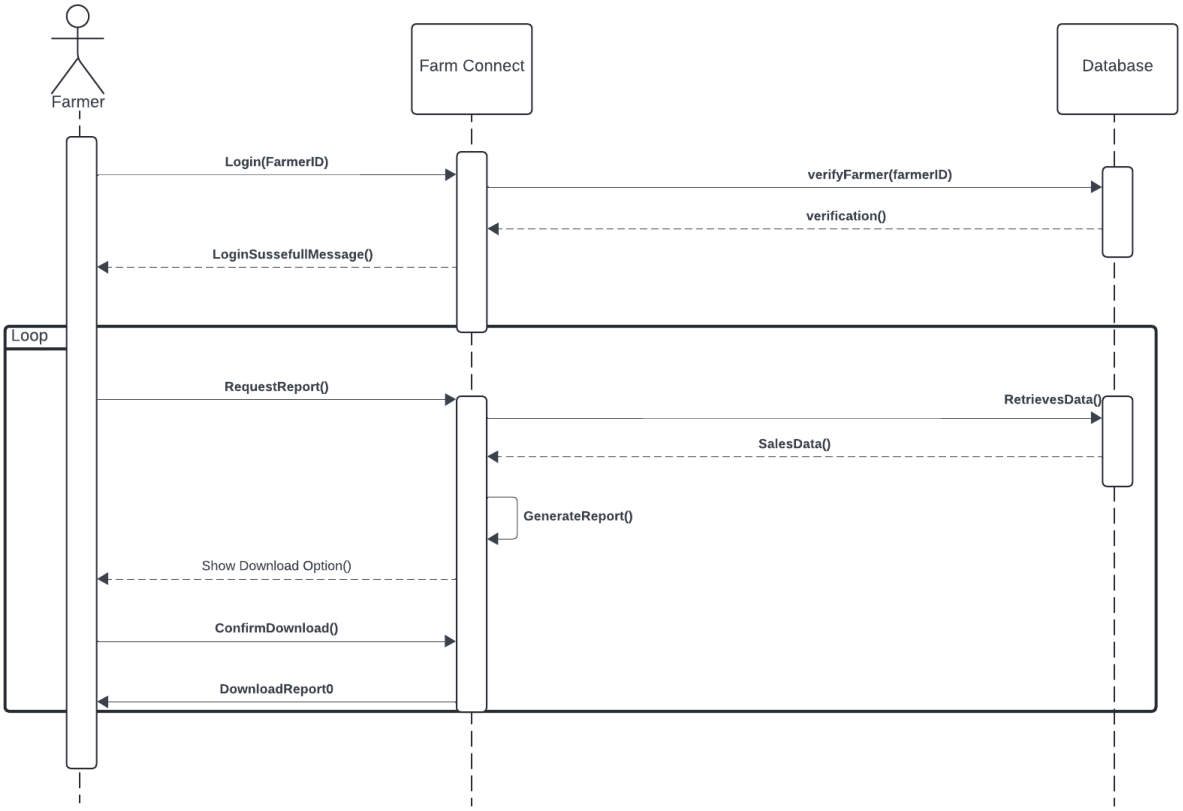
Usecase8(a)



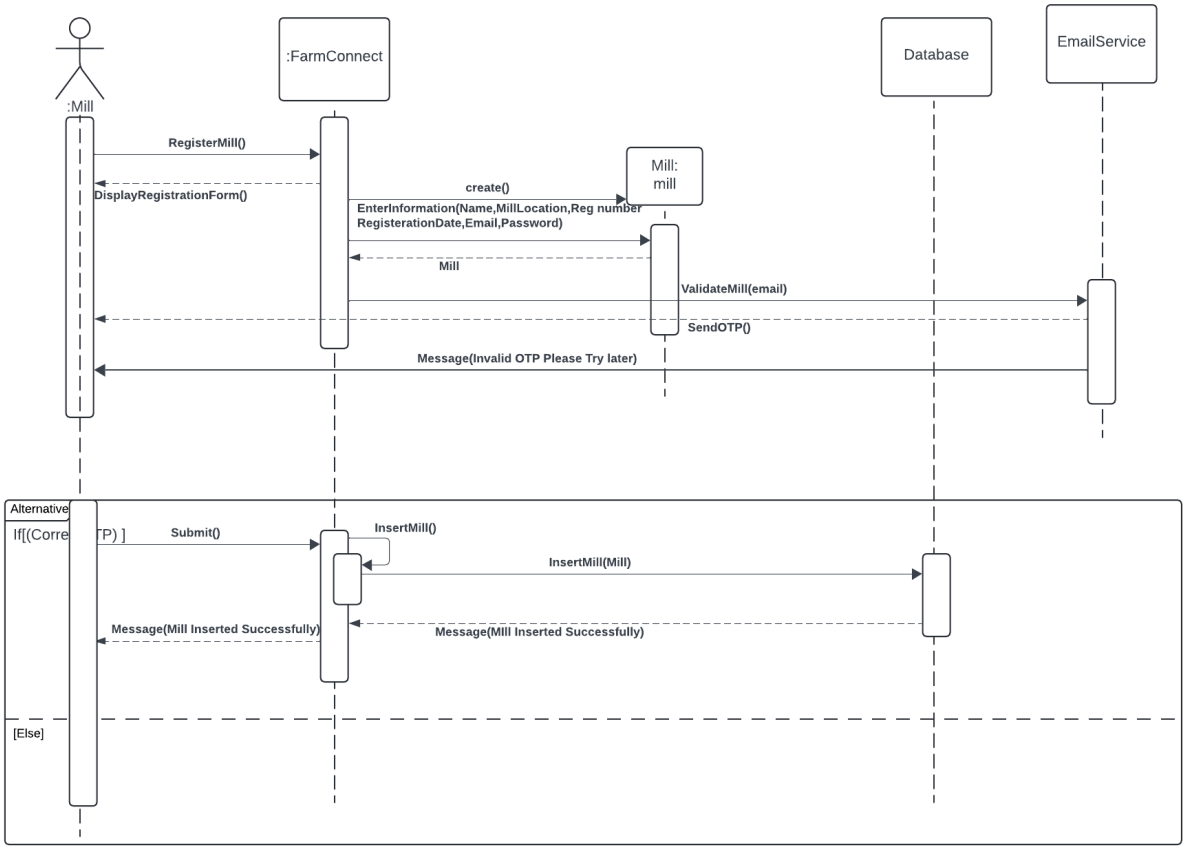
(b)



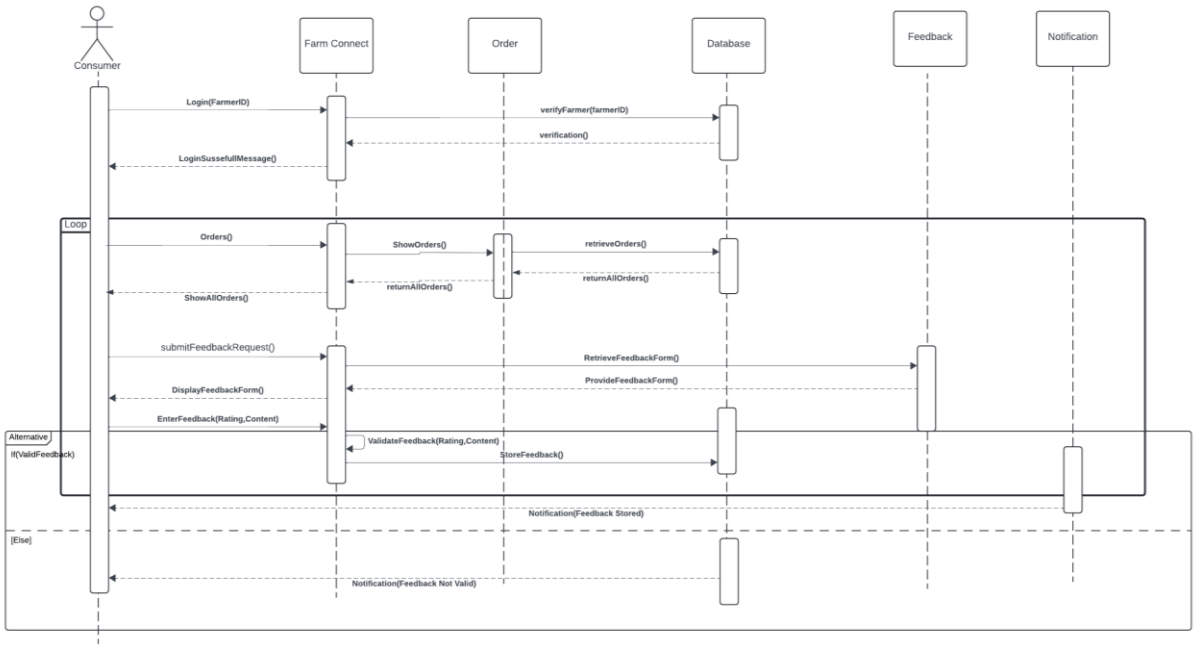
Usecase9


Farmer
Farm Connect
Database
Login(FarmerID)
verifyFarmer(farmerID)
verification()
LoginSussefullMessage()
Loop
RequestReport()
RetrievesData()
SalesData()
GenerateReport()
Show Download Option()
ConfirmDownload()
DownloadReport0

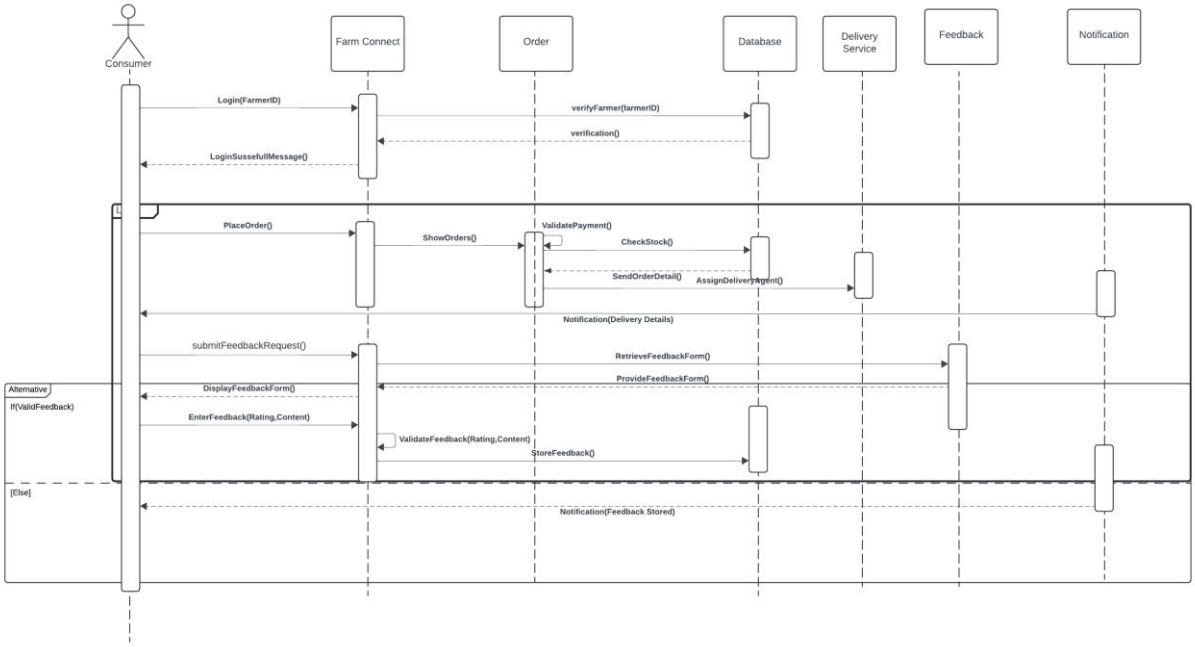
Usecase10


:Mill
:FarmConnect
Database
EmailService
RegisterMill()
DisplayRegistrationForm()
create()
Mill:
mill
EnterInformation(Name,MillLocation,Reg number
RegisterationDate,Email,Password)
Mill
ValidateMill(email)
SendOTP()
Message(Invalid OTP Please Try later)
Alternative
If[(Corre
TP)]
Submit()
InsertMill()
InsertMill(Mill)
Message(Mill Inserted Successfully)
Message(Mill Inserted Successfully)
[Else]

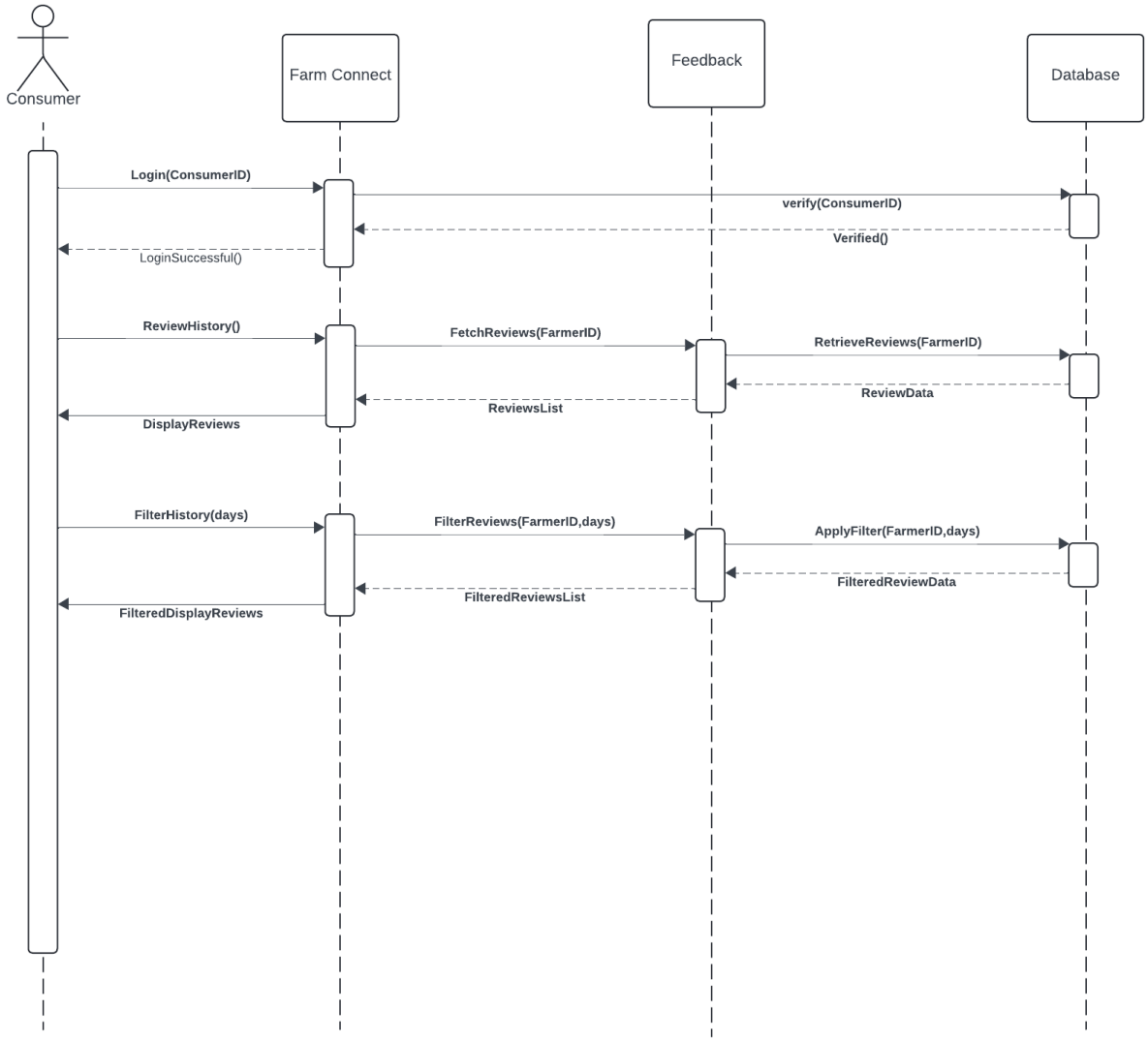
Usecase11(a)


Consumer
Farm Connect
Order
Database
Feedback
Notification
Login(FarmerID)
verifyFarmer(farmerID)
verification()
LoginSussefullMessage()
Loop
Orders()
ShowOrders()
retrieveOrders()
returnAllOrders()
returnAllOrders()
ShowAllOrders()
submitFeedbackRequest()
RetrieveFeedbackForm()
ProvideFeedbackForm()
DisplayFeedbackForm()
EnterFeedback(Rating,Content)
Alternative
If(ValidFeedback)
ValidateFeedback(Rating,Content)
StoreFeedback()
Notification(Feedback Stored)
[Else]
Notification(Feedback Not Valid)

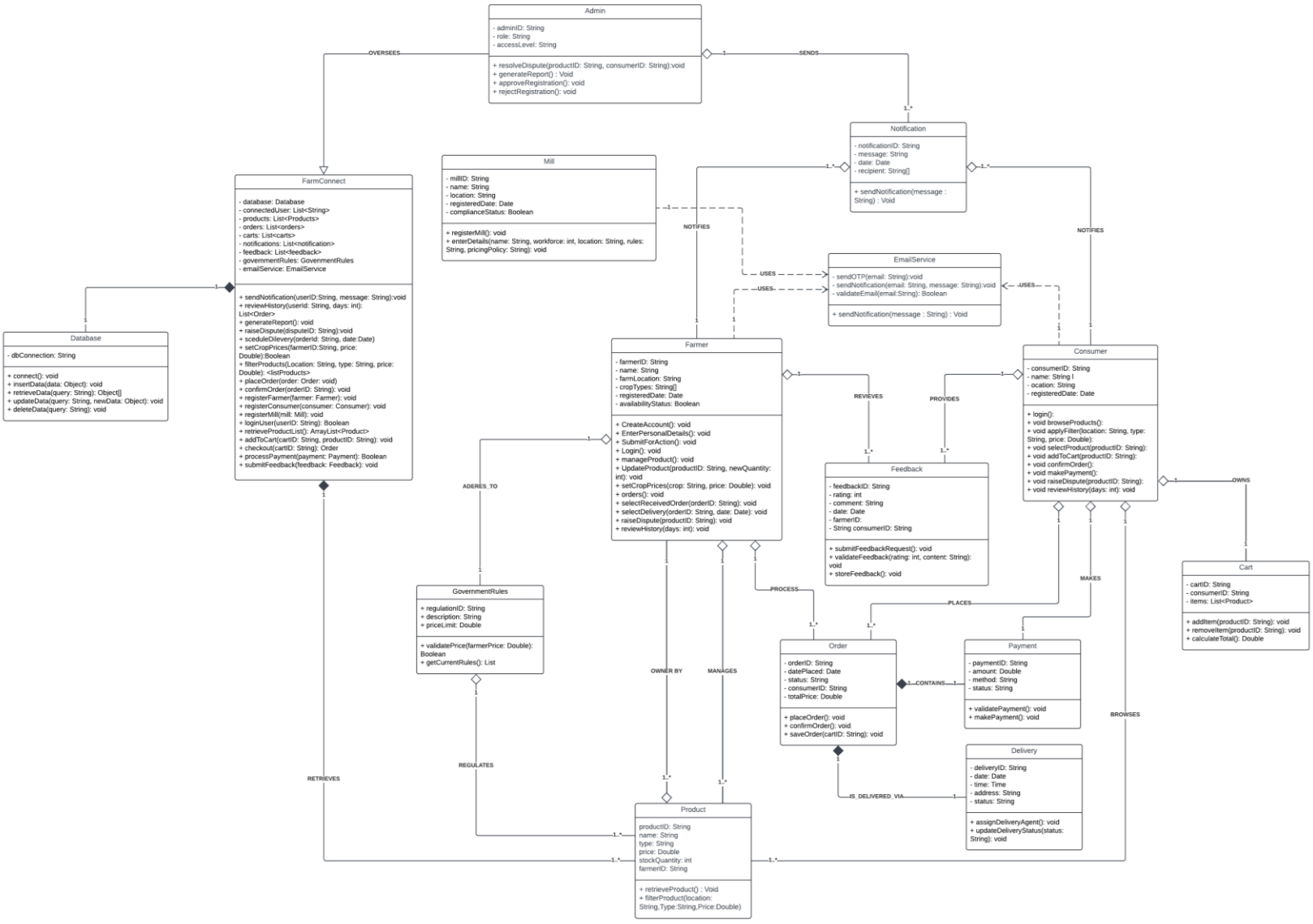
(b)


Consumer
Farm Connect
Order
Database
Delivery Service
Feedback
Notification
Login(FarmerID)
verifyFarmer(farmerID)
verification()
LoginSussefullMessage()
PlaceOrder()
ShowOrders()
ValidatePayment()
CheckStock()
SendOrderDetail()
AssignDeliveryAgent()
Notification(Delivery Details)
submitFeedbackRequest()
RetrieveFeedbackForm()
ProvideFeedbackForm()
Alternative
DisplayFeedbackForm()
If(ValidFeedback)
EnterFeedback(Rating,Content)
ValidateFeedback(Rating,Content)
StoreFeedback()
[Else]
Notification(Feedback Stored)

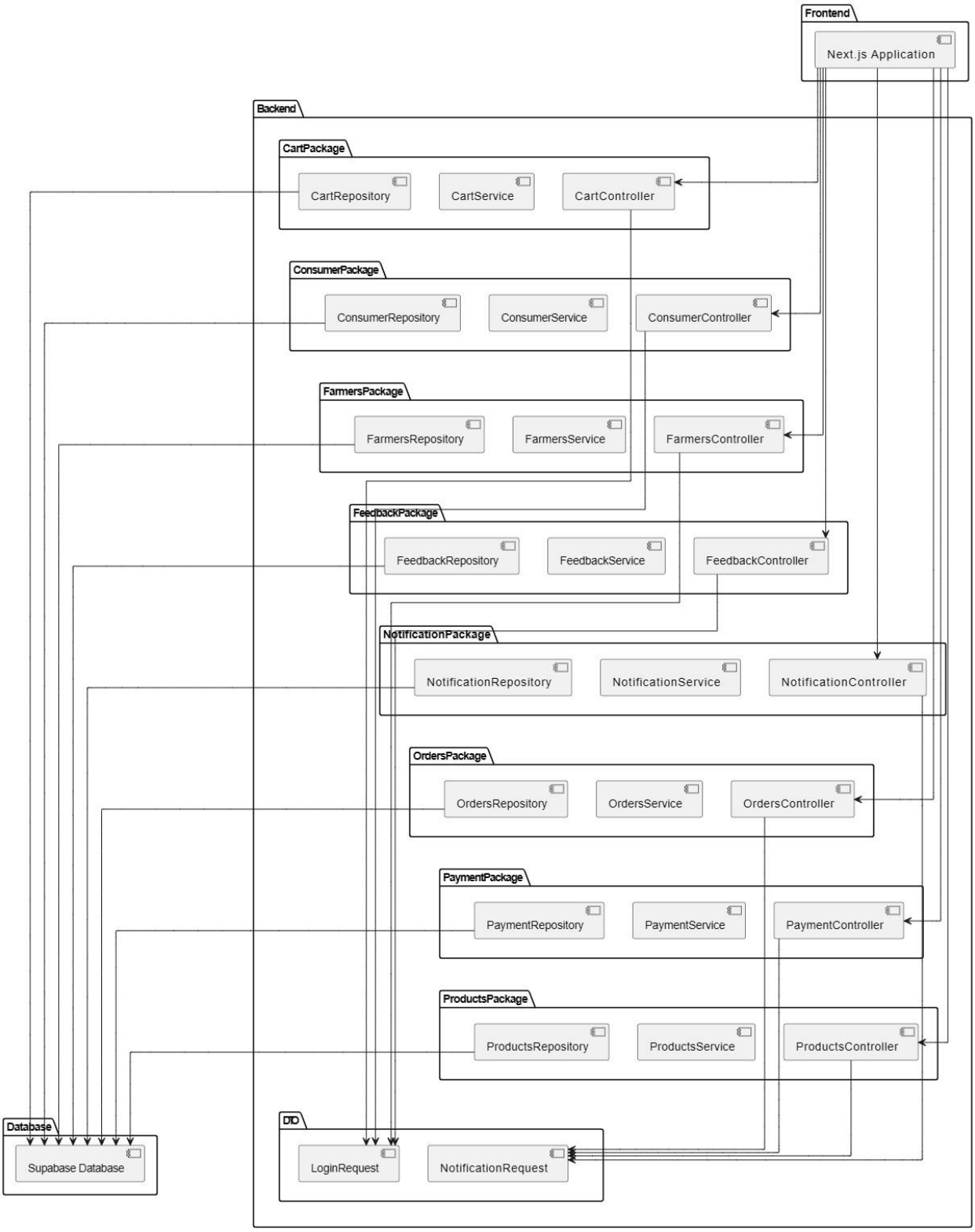
Usecase12


Consumer
Farm Connect
Feedback
Database
Login(ConsumerID)
verify(ConsumerID)
Verified()
LoginSuccessful()
ReviewHistory()
FetchReviews(FarmerID)
RetrieveReviews(FarmerID)
ReviewData
ReviewsList
DisplayReviews
FilterHistory(days)
FilterReviews(FarmerID,days)
ApplyFilter(FarmerID,days)
FilteredReviewData
FilteredReviewsList
FilteredDisplayReviews

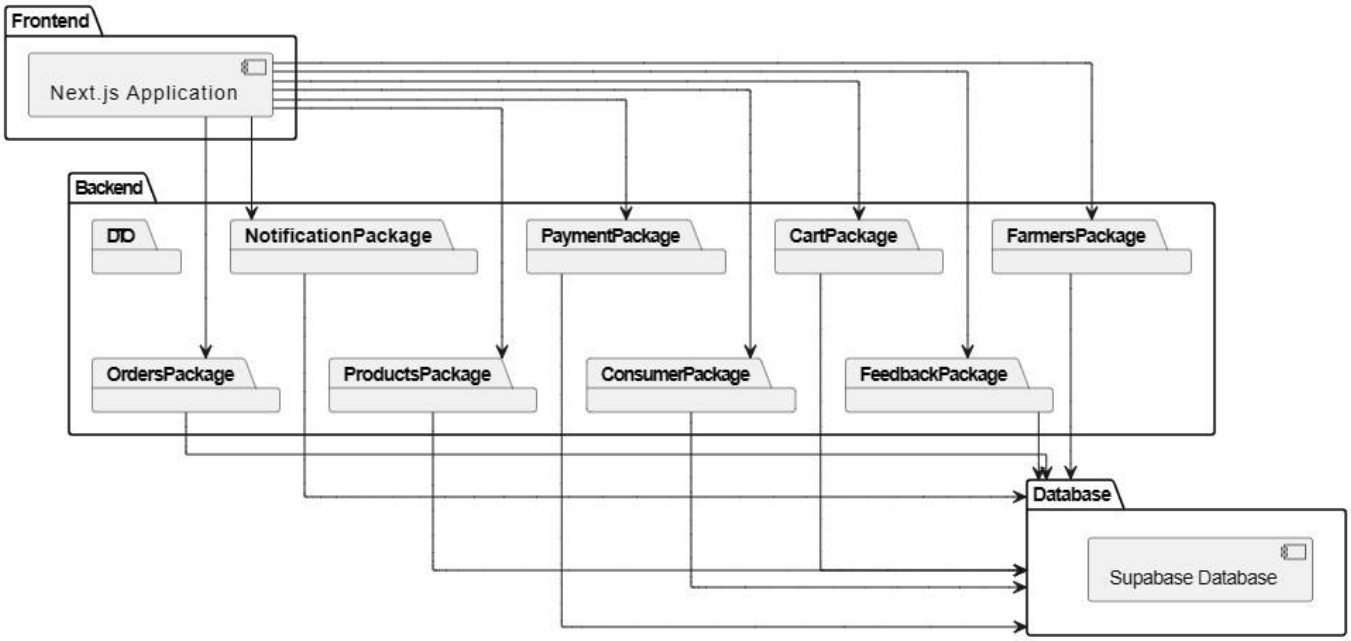
7. Class Diagram



8. Component Diagram



9. Package Diagram



10. Deployment Diagram