WACHEMO UNIVERSITY

COLLEGE OF ENGINEERING AND TECHNOLOGY

DEPARTMENT OF SOFTWARE ENGINEERING

Title: Stock Management System

Group Members **Id**

1. **Minasie Mengesha.....17D6911**
2. **Ruhama Alemu.....172019**
3. **Fetiha Oumer.....171211**
4. **Eyobed Solomon 172510**
5. **Yamilaksira Ashenafi.....17D4021**

ACKNOWLEDGEMENT

❖My appreciation also goes to my instructor, whose support and helped shape this project in the right direction.
❖I would also like to thank Kidemu Mini Market for allowing me to collect real-world information that made this system practical and relevant.
❖Finally, I extend my thanks to everyone who supported and encouraged me during this study.

LIST OF FIGURES

LIST OF TABLES

ACRONYM

❖DB Database
❖ERD Entity Relationship Diagram
❖PK Primary Key
❖FK Foreign Key
❖UI User Interface
❖UML Unified Modeling Language

EXECUTIVE SUMMARY

❖This project focuses on designing a digital stock management system for Kidemu Mini Market, which currently relies on a manual notebook-based inventory process. The existing system makes it difficult to track stock, detect expiry dates, calculate sales, and generate reports accurately.

❖The proposed system replaces the manual method with a computerized solution that allows the owner and the accountant to register products, update stock, record sales, track expiry dates, and generate reports automatically. It ensures accuracy, reduces time spent on manual calculations, and improves decision-making.

❖The system is designed with clear functional and non-functional requirements, supported by UML diagrams such as use case diagrams, class diagrams, sequence diagrams, activity diagrams, and an ER diagram.

❖This project lays the foundation for a reliable, easy-to-use inventory management system tailored to the needs of a small retail business.

TABLE OF CONTENTS

CHAPTER ONE: INTRODUCTION

1.1 Background of the Organization

Kidemu Mini Market

❖Kidemu Mini-market is an owner-operated retail establishment located in Hosana, Ethiopia. Founded and managed by Kidmu Nasru, the business has served the local community for several years as a neighborhood convenience store. The enterprise currently operates with a single full-time staff member, the owner, who oversees all daily operations.

Current Operational Context

❖Kidemu Mini Market currently depends on a fully manual inventory system. The owner records products, sales, expiry dates, and stock levels inside a single notebook. To check availability, he physically checks shelves and estimates what is running out. This manual approach slows down decision-making and easily leads to errors such as missing expiry dates, miscounting stock, or failing to notice low-stock items.

❖As the number of products grows, the notebook-based method becomes unreliable. There is a need for a simple digital system that can track products, update stock automatically after sales, and generate quick reports.

1.2 Statement of the Problem

❖Kidemu Mini Market manual management system causes operational inefficiencies and financial inaccuracies. It leads to:

- Frequent calculation errors in stock and sales records
- No automatic alerts for low stock or expiring product
- Wasted time to check shelves physically
- Difficulties tracking sales, cost, and profit
- Reports are inconsistent and time-consuming to prepare

❖A digital stock management system will eliminate these problems and improve accuracy and efficiency.

1.3 Objective of the project

1.31. General Objective

❖To design and develop a digital stock management system to replace the manual notebook used at Kidemu Mini Market.

1.3.2 Specific Objectives

OWNER-RELATED OBJECTIVES

1. To enable product registration with comprehensive details including expiry dates and minimum stock levels
2. To facilitate real-time inventory adjustments (additions, deductions, and corrections)
3. To implement automated alerts for low-stock items and approaching expiry dates
4. To track and report expired inventory for accurate waste management
5. To replace manual notebook entries with a reliable digital record-keeping system

REPORTING AND ANALYSIS OBJECTIVES

1. To generate automated reports for sales, income, costs, and inventory valuation
2. To provide historical data analysis for business performance tracking
3. To enable easy cross-referencing of item history across purchases and sales

SYSTEM AND SECURITY OBJECTIVES

1. To implement role-based access control (Owner: full access, Accountant: view/report access)
2. To ensure data accuracy and consistency through validation and error-checking mechanisms
3. To minimize overstocking through data-driven inventory recommendations

1.4. Methodologies

1.4.1 Requirement Elicitation Methodology

❖To understand the needs of Kidemu Mini Market, the following methods were used:

- **Observation:** Directly observing how the Owner records stock, checks shelves, and prepares reports.
- **Interview:** Asking the Owner and Accountant about challenges with the current notebook-based system.
- **Document Review:** Examining the existing notebook used for stock records, sales, and expiry information.
- **Problem Analysis:** Identifying issues in accuracy, speed, and reliability based on gathered information.

❖These methods ensured that the system requirements reflect real shop operations.

1.4.2 Requirement Analysis and Modeling

❖After collecting requirements, the system was analyzed and modeled using:

- **Use Case Diagrams:** To identify system actors and their interactions.
- **Sequence Diagrams:** To describe the flow of activities such as adding products, recording sales, and generating reports.
- **Activity Diagrams:** To represent the workflow of major functions.
- **ER Diagram:** To design the database structure and relationships.

❖These models guided the design of a clear and organized system.

1.4.3 System Implementation Methods

❖The system will be implemented using a structured step-by-step approach to ensure that all stock management functions work correctly and match the requirements gathered from the Owner and Accountant.

1. Technology Stack

➢The following tools and technologies will be used during development:

- Programming Language: C++
- Database System: SQL Server
- UML Modeling Tool: Enterprise Architect (for Use Case, Sequence, Activity, Class, and State diagrams)
- Documentation Tool: Microsoft Word
- Interface tool
- Testing Method: Manual testing with sample inputs

❖These tools are chosen because they are reliable, widely used, and suitable for a small desktop-based stock management system.

2. Database Implementation

➢Designing the database structure in Enterprise Architect using an ER diagram Implementing tables in SQL Server for:

- Product
- Stock
- User
- Transaction
- Report

➢Setting primary keys and foreign keys to maintain proper relationships

➢Ensuring data consistency during stock updates, sales recording, and expiry processing

3. User Interface Development (C++)

➢Simple and user-friendly C++ forms will be developed for:

- Login
- View stock
- Add/Update/Delete product
- Add stock
- Record sales
- Record expired items
- Viewing reports
- Viewing alerts

❖The interface will be designed to be easy for the Owner and Accountant to operate with minimal technical knowledge.

4. Business Logic Implementation (C++)

➢C++ functions will implement all core business rules:

•Automatic stock deduction after each sale

•Automatic generation of low-stock and expiry alerts

•Role-based access control:

•Owner → Full access

•Accountant → View-only

❖Input validation for product names, expiry dates, quantities, and prices

Prevention of incorrect or duplicate entries

5. Report Generation

➢The system will automatically generate clear reports, including:

•Sales report •Income and cost report

•Low-stock report •Expiry and wastage report

•Transaction history

❖All reports will be formatted clearly for viewing or printing.

6. Data Migration

➢Since the previous system is manual:

•Existing product data will be manually entered into the system

•Quantities and expiry dates will be transferred from the notebook

•Old financial summaries may remain in the notebook or be entered if needed

❖This ensures a smooth transition from paper to digital.

7. Backup and Recovery

➢Because power interruptions are common:

•Regular database backups will be stored on USB or cloud storage

•SQL Server backup/restore tools will be used

•A UPS (Uninterruptible Power Supply) is recommended to protect the system from sudden shutdown

•Backup procedures will be documented for the Owner

8. Error Handlin

➢The system will include:

•Input validation (numbers, dates, empty fields)

•Warning messages for invalid entries

•Prevention of negative stock values

•Error logs for debugging

•Protection against duplicate product names

9. Testing Strategy

➢Testing will be performed using manual test cases:

- Unit Testing
- Login
- Add product
- Add stock
- Record sale
- Record expired items
- Alerts
- Integration Testing
- C++ interface connected to SQL Server
- Stock update flows
- Report generation
- System Testing
- Running the entire system like a real shop
- Checking correctness of stock, expiry, and reports
- User Acceptance Testing (UAT)
- Owner and Accountant test the complete system

10. Deployment

- Installing the system on the mini market's computer
- Installing SQL Server database
- Creating Owner and Accountant accounts
- Initial data entry (migration)
- Final configuration and testing

11. Maintenance Plan

- Fixing bugs after deployment
- Updating features based on feedback
- Updating the database when product categories change
- Regular database cleanup
- Continuous backups

1.5. Scope and Limitation of the Project

1.5.1 Scope

➢The system covers:

- Product registration
- Stock updates
- Recording sales
- Expiry date alerts
- Low-stock alerts
- Report generation for income, cost, expiry, and sales
- Separate login access for Owner and Accountant

❖The project focuses only on basic stock management for a single mini market location.

1.5.2 Limitations

➢The system does not include:

- Barcode scanning
- Online access or mobile app
- Supplier management
- Multi-branch support
- Automatic pricing updates
- Integration with POS machines

❖It is designed as a simple offline desktop system.

1.6. Significance and Beneficiaries of the project

SIGNIFICANCE

➢The system brings several improvements:

- Reduces stock errors from handwritten notes
- Helps prevent expired items from being sold
- Saves time spent on shelf checking
- It helps to produces accurate financial reports
- Supports better stock planning

BENEFICIARIES

➢***Owner:*** Gains accuracy, faster operations, and better decision-making.

➢***Accountant:*** Receives clear and organized data for income and cost calculations.

➢***Business:*** Reduces financial loss and improves efficiency

1.7 Feasibility Analysis

1.7.1 Operational/Organizational Feasibility

The system fits well with daily operations because the users (Owner and Accountant) already understand the workflows. The system is simple and easy to operate.

1.7.2 Technical Feasibility

The system requires only a basic computer, simple database software, and a lightweight interface. No advanced hardware or internet connectivity is needed.

1.7.3 Economic Feasibility

The cost of development is low because free tools and simple technologies are used. The system reduces losses from expired items and stock errors, making it economically beneficial.

1.7.4 Schedule Feasibility

The project can be completed within the semester timeline because the functions are small and well-defined (product entry, stock updates, reporting, alerts).

1.7.5 Legal Feasibility

The system does not violate any legal requirements. It manages only internal stock data and does not involve customer personal information.

1.7.6 Others (Social Feasibility)

The system improves customer service by ensuring products are available and not expired. It also reduces workload for workers.

1.8 Risk Assessment

POSSIBLE RISKS

1. Data Loss

If the computer fails or restarts unexpectedly without backup, important stock or sales information may be lost.

2. User Errors

Inexperienced users may enter incorrect product information, quantities, or prices, which can affect stock accuracy and reports.

3. Power Interruption (Electricity Problem)

Frequent electricity outages in Ethiopia may interrupt system usage, cause unsaved data to be lost, or delay daily operations.

4. Hardware Failure

Computer damage caused by overheating, physical impact, or power fluctuations could affect system performance or lead to downtime.

5. Limited Technical Skills

The Owner and Accountant may take time to adapt to the digital system, especially if they have limited computer experience.

MITIGATION STRATEGIES

- Regular Data Backups to external storage (USB, cloud, or external drive).
- Use of a power bank to protect the system during power interruptions.
- Simple and user-friendly interface to reduce input errors.
- Basic training for both the Owner and the Accountant.
- Routine maintenance for the computer to prevent hardware issues.

1.9 Development Tools

➢ The following tools may be used:

- ***Database:*** SQL
- ***Design Tools:*** UML (Use case, sequence, activity, class diagram, state chart, ER diagrams)

- ***Interface Tools:***
- ***Documentation:*** Microsoft Word
- ***Testing Tools:*** Manual test cases

1.10 Work Breakdown

1.10.1 Project Plan Activities (Schedule)

- Requirement gathering
- Database design
- Implementation
- Documentation
- System analysis
- Interface design
- Testing
- Final review

1.10.2 Project Organization

- Chapter arrangement
- Documentation writing
- Diagram development
- Review and correction

1.10.3 Team Organization

- All tasks performed by all members of group in parallel way:
- Requirement & Analysis – All Member
- Database Design – All Member
- Documentation & Diagrams – All Member

1.11 Budget Plan

❖Since the project is academic and uses free tools, the cost is minimal:

- ***Software tools***: Free
- ***Laptop/Computer:*** Existing
- ***Printing and binding:*** Minor cost
- ***Electricity:*** Minimal
- ***Total estimated cost:*** Very low / negligible

CHAPTER 2: REQUIREMENT ANALYSIS AND SPECIFICATIONS

2.1. Description of the Current System

❖The mini market operates using a fully manual record-keeping approach. The owner writes new products in a notebook, updates quantities after sales, and occasionally notes expiry dates. To check availability, the owner physically inspects the shelves. This method is slow, error-prone, and relies heavily on memory.

❖The accountant calculates income, cost, and wastage at the end of each month using the same notebook, making the process even more prone to errors.

2.2 Players/Actors in the Existing System

In the current manual system, there are only two people involved in managing and reviewing stock-related information:

OWNER

➢The Owner performs all daily activities, including:

•Recording new products in the notebook
•Updating stock after sales
•Removing unusable or expired items
•Checking shelves to confirm availability
•Identifying low-stock products
•Manually checking expiry dates
•Maintaining the only record book

❖The Owner is the main actor who handles every operation in the existing system.

ACCOUNTANT

➢The Accountant is not involved in daily activities but works periodically. They:

•Review the notebook to calculate monthly income
•Check cost and wastage
•Estimate profit
•Prepare monthly financial summaries

❖The accountant depends fully on the Owner's handwritten entries.

2.3 Use Case Diagram for the Existing System

❖The existing system is completely manual, so all interactions are direct physical tasks by the Owner and Accountant. All connectors are simple Association links.

OWNER USE CASES

- Add New Stock
- Remove Stock
- Identify Low Stock
- Record Expired Items
- Update Stock After Sales
- Check Stock Availability
- Check Expiry Dates

ACCOUNTANT USE CASES

- Review Monthly Records
- Calculate Wastage
- Calculate Income
- Prepare Monthly Summary

❖No system automation, include, or extend relationships exist because the process is fully manual.

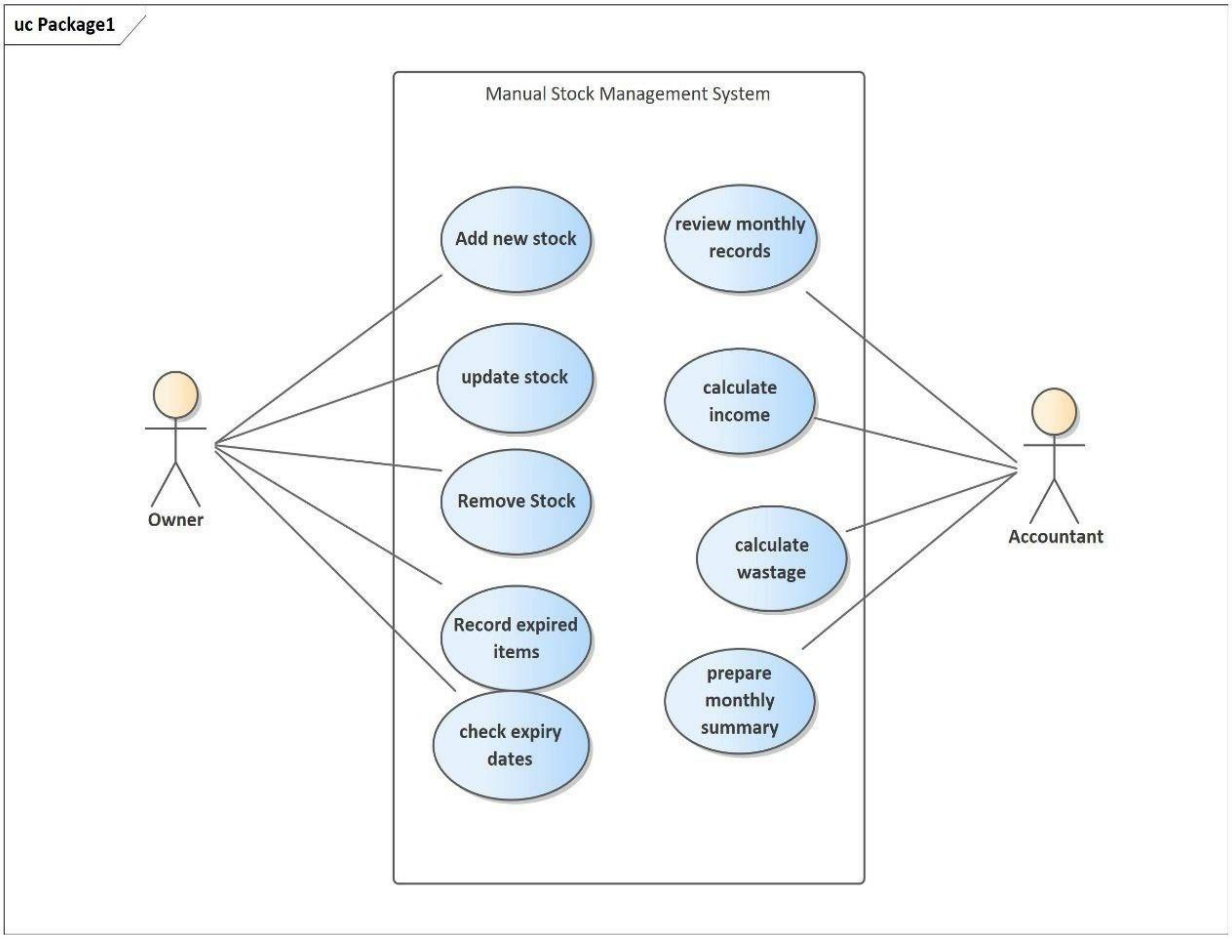


Figure_ 1

2.4 Forms and Documents Used in the Existing System

❖The existing system does not use any digital forms. All records are kept manually using:

- A Single Notebook

This is the main document where:

- New products are written
- Stock updates are recorded
- Sales are written manually
- Expiry information (sometimes) is noted
- Monthly records are checked by the Accountant

- Loose Papers

❖Temporary notes written during busy times before being copied to the notebook.

- Physical Shelf Inspection

❖To check availability, the Owner must walk to the shelves and visually inspect items.

- Accountant Calculations

The Accountant uses the same notebook to prepare financial summaries manually.

❖There are no digital records, templates, or backup documents, making the system disorganized and unreliable.

2.5 Business Rules of the Existing System

❖The existing system follows several informal rules:

- All stock information must be handwritten in the notebook.
- Product name is used as the main identifier because there is no ID or code.
- All items follow one simple format, regardless of type or category.
- Expiry dates are checked manually by reading product labels.

5. Stock decreases only when the Owner writes it down after each sale.
6. Only the Owner manages daily stock activities (add, update, check, remove).
7. The Accountant prepares reports manually based on the handwritten notebook.

These business rules show the high level of dependence on manual work and personal accuracy.

2.6 Proposed System

2.6.1 Overview

❖The proposed system is a simple computer-based stock management system that replaces the handwritten notebook used in the current manual process. It automates stock recording, sales updates, expiry tracking, and report generation. The system allows the Owner to manage all stock-related activities digitally, while the Accountant can view records and generate financial reports without modifying stock data.

❖The new system improves accuracy, reduces human error, provides real-time information, and saves time spent on manual counting and checking shelves. It also introduces automatic alerts for low-stock items and upcoming expiry dates, which are not possible in the current manual system.
The proposed system replaces the notebook with a computer-based stock management system. It allows the owner and accountant to register products, update stock, record sales, and generate expiry or low-stock alerts automatically.

➢Key Improvements:

- Automatic stock updates
- Quick product searches
- Expiry and low-stock notifications
- Accurate and reliable reports

2.6.2 Business Rules of the New System

The proposed system follows the following new business rules:
1. Products must be registered in the system before performing any stock-related operation.
2. Stock can only be increased using the Add Stock function; manual adjustments are not allowed.
3. Expired items must be recorded and included in the wastage report.
4. Product details must include valid attributes such as name, type, cost price, selling price, and expiry date.
5. User access is role-based:
   Owner has full control of all stock and product functions.
   Accountant has view-only access and cannot modify stock data.
6. Reports must be generated automatically and include low-stock items, and expired products.
7. Sensitive operations such as deleting a product or changing a selling price are restricted to the Owner only.

8. The system must maintain data accuracy by validating inputs and preventing duplicate product names.
9. The system must support data backup and recovery to prevent data loss due to power interruption, hardware failure, or other risks.

2.6.3 Functional Requirements

➢Owner Functional Requirements
- Login
- Register new products
- Update product details
- Delete product
- Add stock
- Record sales
- Record expired items
- View stock
- View low-stock alerts
- View expiry alerts
- Generate reports
- View reports

➢Accountant Functional Requirements
- Login
- View stock and sales information
- View expiry and low-stock alerts
- Generate income and cost reports
- View reports

2.6.4 Non-Functional Requirements

- **Usability:** Simple and easy interface
- **Performance:** Fast search and update
- **Reliability:** Accurate calculations
- **Security:** Login-based access
- **Maintainability:** Easy to modify
- **Portability:** Runs on any computer

2.6.5 Actor and Use Case Identification

MAIN ACTORS

1. Owner

- Has full access to the system
- Performs all stock-related activities
- Can add, update, and delete products
- Records sales, expired items, and stock changes
- Views all alerts and generates reports

2. Accountant

- Has view-only access
- Cannot modify stock
- Can view stock information, sales data, and expiry/low-stock alerts
- Generates financial reports such as income, cost, and wastage reports

USE CASES

Owner Use Cases

- Login
- Add Product
- Update Product
- Delete Product
- Add Stock
- Record Sales (auto-reduces stock)
- Record Expired Items
- View Stock
- View Low-Stock Alerts
- View Expiry Alerts
- Generate Reports
- Check Income and Cost

Accountant Use Cases

- Login
- View Stock
- View Sales Data
- View Expiry and Low-Stock Alerts
- Generate Reports
- Check Income and Cost

2.6.6 System Model

❖This section provides a formal specification of the system through a series of architectural and behavioral models. These models serve as the definitive blueprint for the system's construction, ensuring that all functional and non-functional requirements are accurately captured and designed.

2.6.6.1 Use Case Model

❖The use case diagram shows how the Owner and Accountant interact with the stock management system. The Owner performs all stock-related operations such as registering products, updating stock, recording sales, and viewing alerts. The Accountant has read-only access and can view stock, view alerts, and generate financial reports. The login use case is required for both actors.

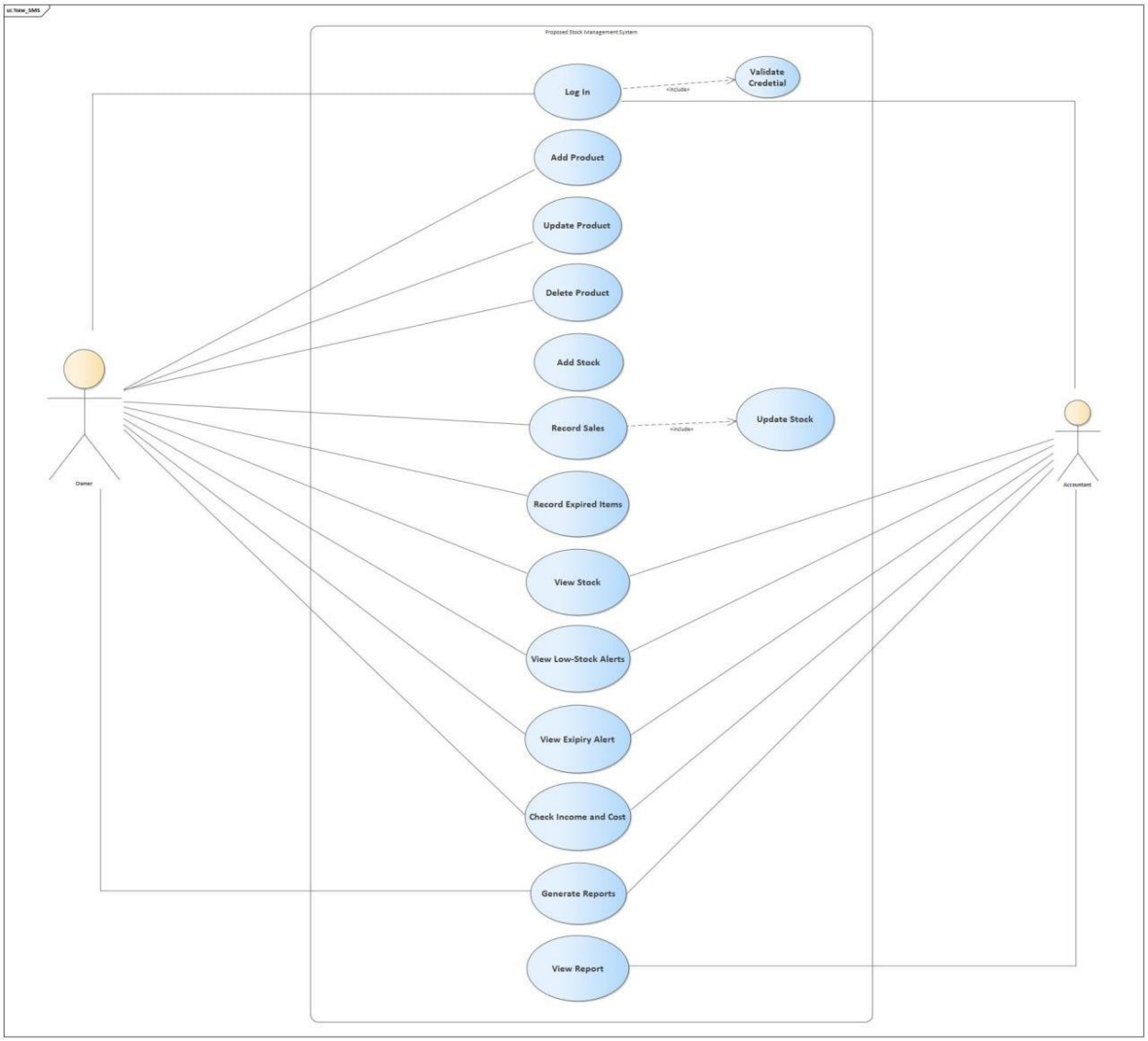


Figure_ 2

2.6.6.1.1 Use Case Descriptions

Use Case: Login

Field	Description
Actor	Owner, Accountant
Description	User enters username and password to access the system.
Precondition	User must have a registered account.
Postcondition	User is logged in with their assigned role.
Main Flow	1. User opens system. 2. Enters username and password. 3. System verifies credentials. 4. User is granted access.

Use Case: Add Product

Field	Description
Actor	Owner
Description	Adds a new product with full details.
Precondition	Owner must be logged in.
Postcondition	Product is saved with a unique ID.
Main Flow	1. Owner selects 'Add Product'. 2. Enters product name, category, cost price, selling price, expiry date, description 3. System saves the product.

Use Case: Update Product

Field	Description
Actor	Owner
Description	Edits product information.
Precondition	Product already exists.
Postcondition	Product details are updated.
Main Flow	1. Owner selects product. 2. Edits fields. 3. System updates the record.

Use Case: Delete Product

Field	Description
Actor	Owner
Description	Removes a product from the system.
Precondition	Product must exist.
Postcondition	Product is deleted.

Main Flow	1. Owner selects product. 2. Confirms deletion. 3. System deletes the product.

Use Case: Add Stock

Field	Description
Actor	Owner
Description	Adds new quantity for an existing product.
Precondition	Product exists.
Postcondition	Stock quantity increases.
Main Flow	1. Owner selects product. 2. Enters new stock amount. 3. System updates quantity.

Use Case: Record Sales

Field	Description
Actor	Owner
Description	Records a sale and automatically reduces stock.
Precondition	Stock must be available.
Postcondition	Stock decreases and transaction is saved.
Main Flow	1. Owner selects product sold. 2. Enters quantity sold. 3. System reduces stock and stores the transaction.

Use Case: Record Expired Items

Field	Description
Actor	Owner
Description	Marks items as expired.
Precondition	Product must exist.
Postcondition	Expired quantity removed from stock.
Main Flow	1. Owner selects product. 2. Enters expired quantity. 3. System deducts the quantity.

Use Case: View Stock

Field	Description
Actor	Owner, Accountant
Description	Shows full stock list and quantity.

Precondition	User must be logged in.
Postcondition	Stock information is displayed.
Main Flow	1. User opens stock page. 2. System displays list.

Use Case: View Low-Stock Alerts

Field	Description
Actor	Owner, Accountant
Description	Displays products below minimum stock level.
Precondition	Stock data must exist.
Postcondition	User sees low-stock list.
Main Flow	1. User opens alerts page. 2. System shows low-stock products.

Use Case: View Expiry Alerts

Field	Description
Actor	Owner, Accountant
Description	Shows items near expiry.
Precondition	Products must have expiry date.
Postcondition	User views expiry alerts.
Main Flow	1. User opens expiry alerts. 2. System lists items near expiry.

Use Case: Generate Reports

Field	Description
Actor	Owner, Accountant
Description	Generates sales, stock, income, or expiry reports.
Precondition	Data must exist.
Postcondition	Report is generated.
Main Flow	1. User selects report type. 2. System gathers data. 3. Displays report.

Use Case: Check Income and Cost

Field	Description
Actor	Accountant
Description	Calculates income, cost, profit.

Precondition	Sales and stock data must exist.
Postcondition	Income/cost summary shown.
Main Flow	1. Accountant selects financial report. 2. System calculates totals. 3. Displays results.

Table_1_ Usecase description

2.6.6.2 Sequences diagram

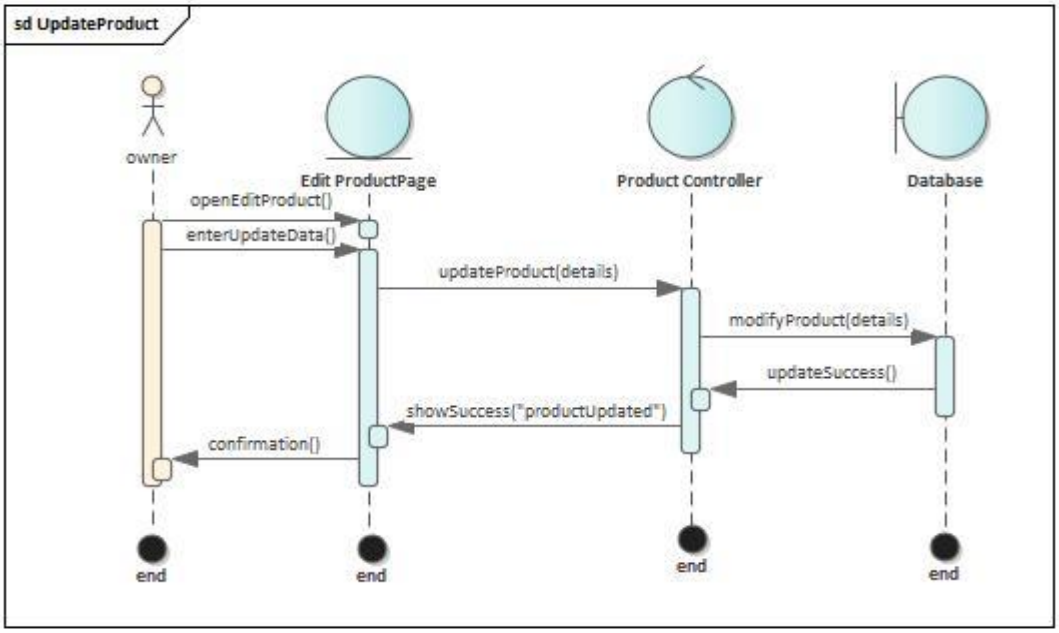


Figure_ 3

Sequences diagram for usecase update product.

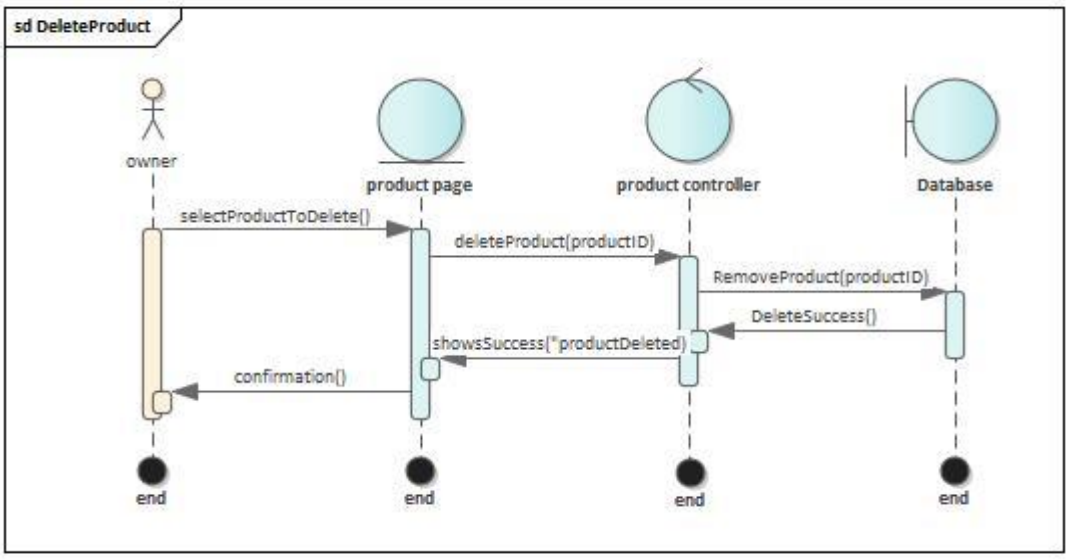


Figure_ 4

Sequences diagram for usecase delete product.

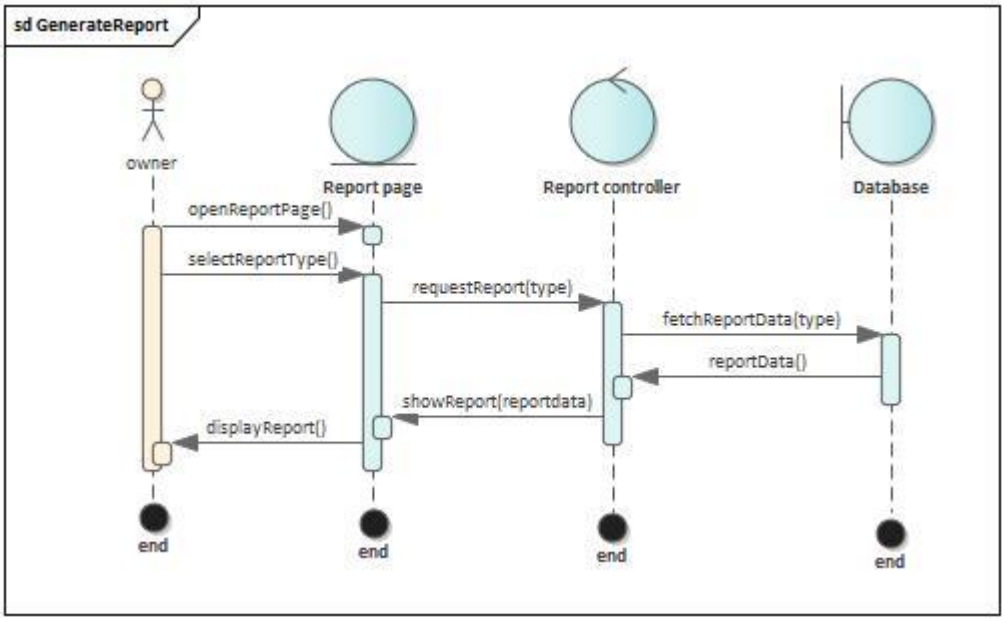


Figure_ 5

Sequences diagram for usecase Generate report.

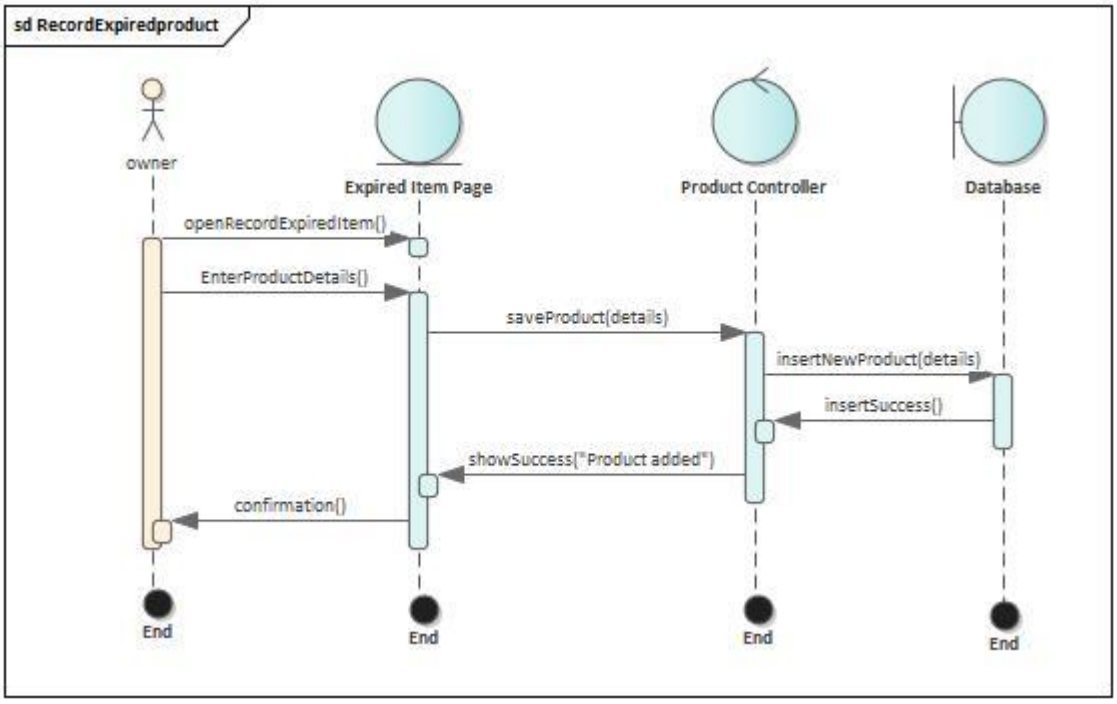


Figure_ 6

Sequences diagram for usecase record expired product.

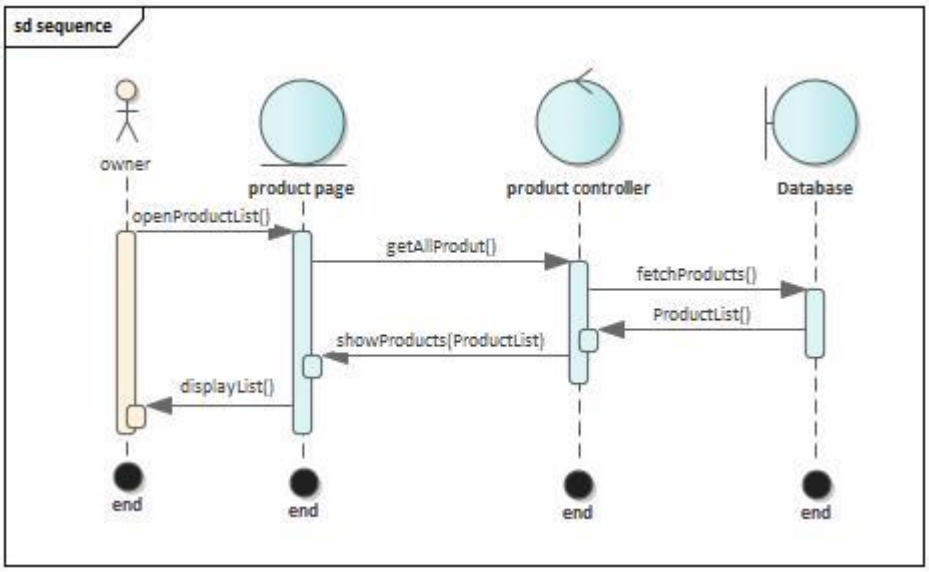


Figure_ 7

Sequences diagram for usecase view product.

2.6.6.3 Activity Diagrams

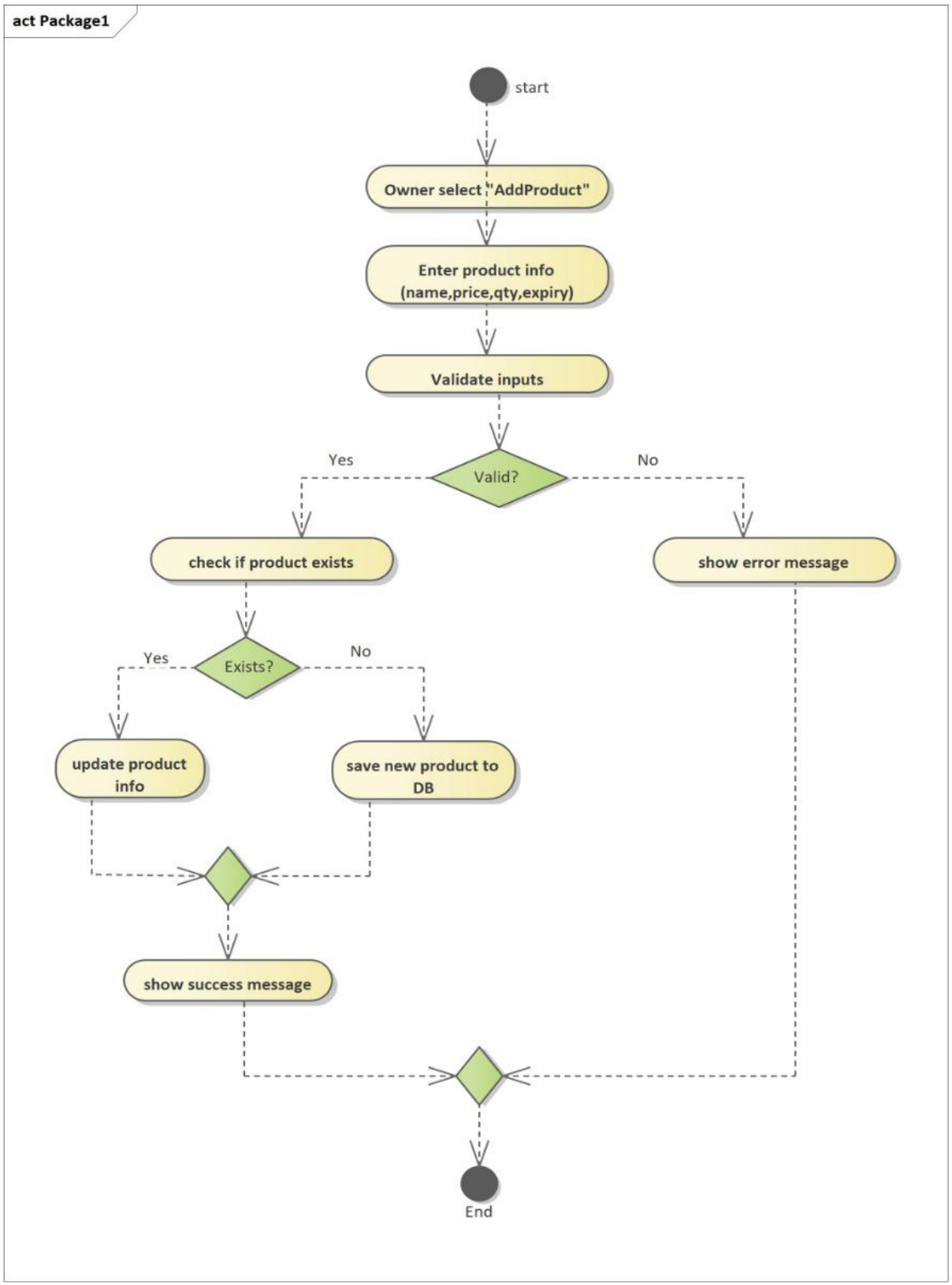


Figure_ 8

Activity diagram for usecase add product

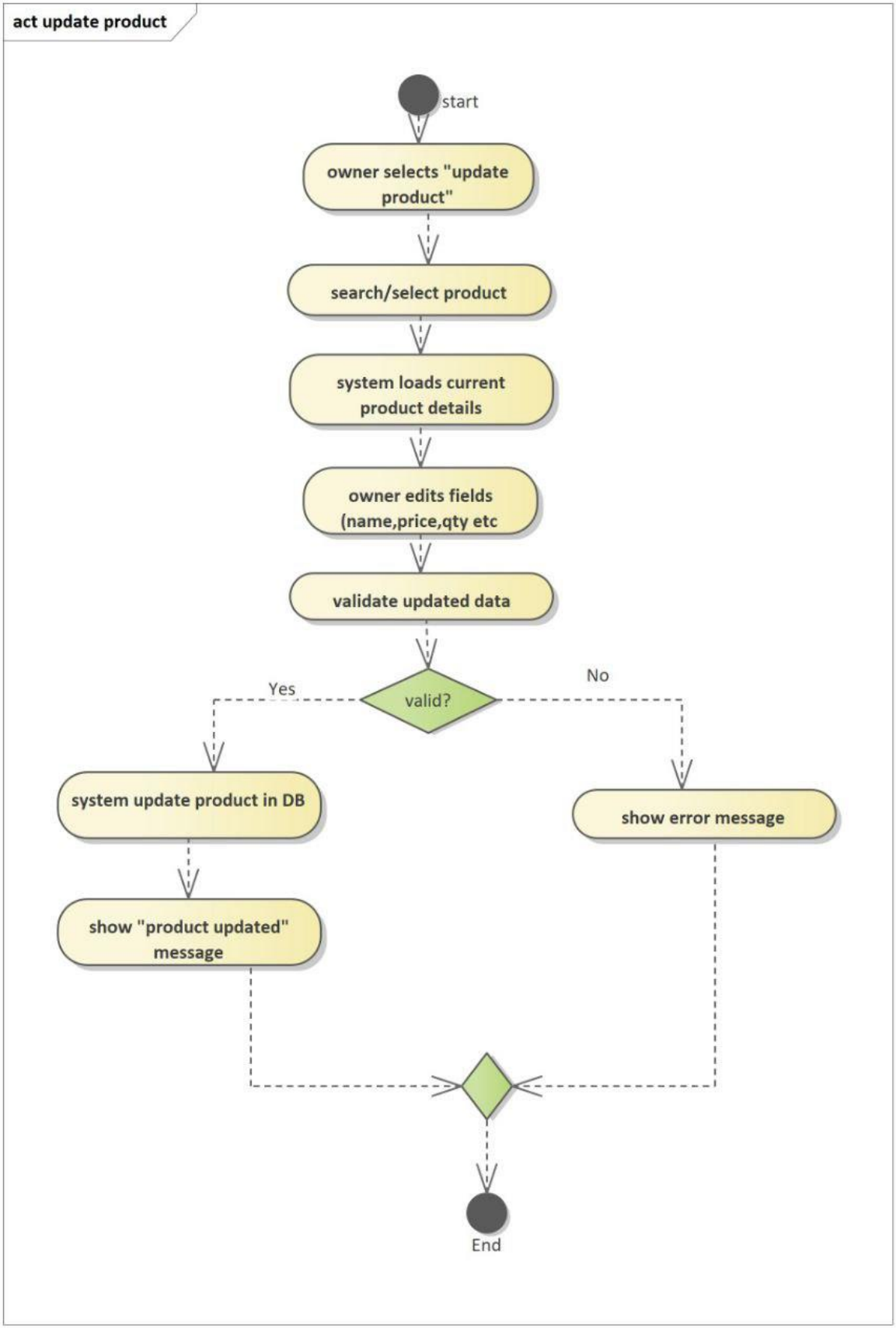


Figure_ 9

Activity diagram for usecase update product.

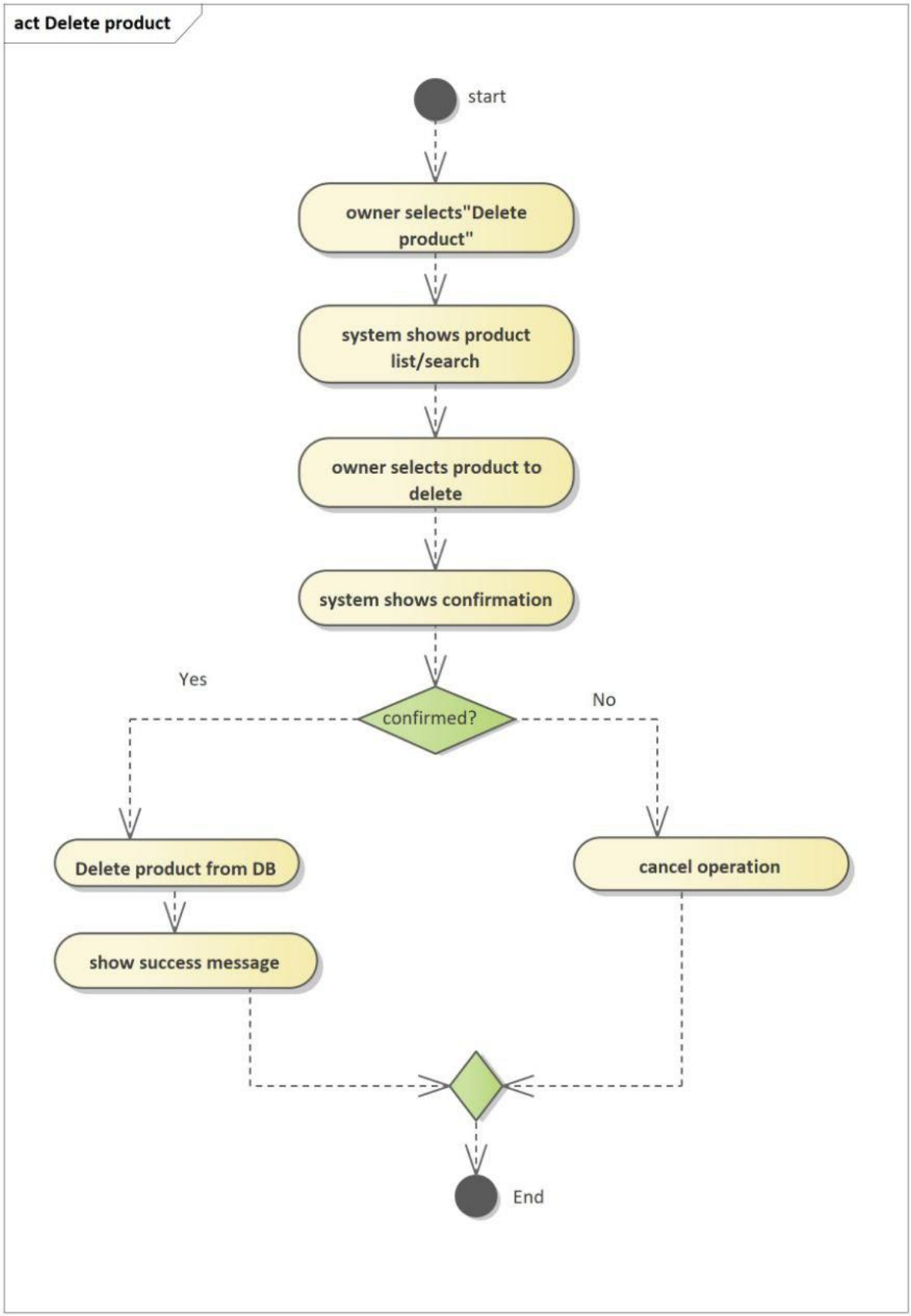


Figure_ 10

Activity diagram for usecase delete product.

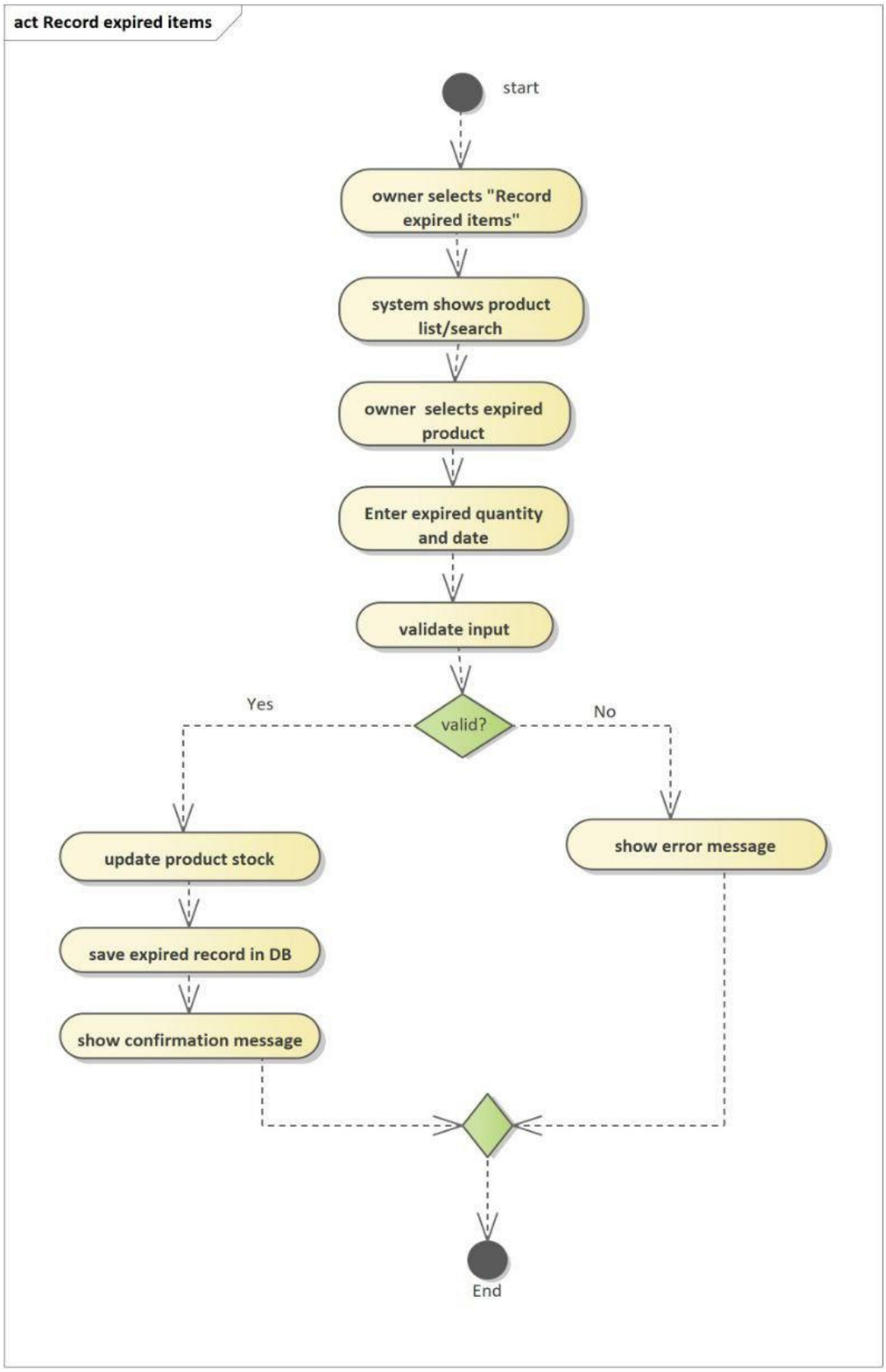


Figure_ 11

Activity diagram for usecase record expired item.

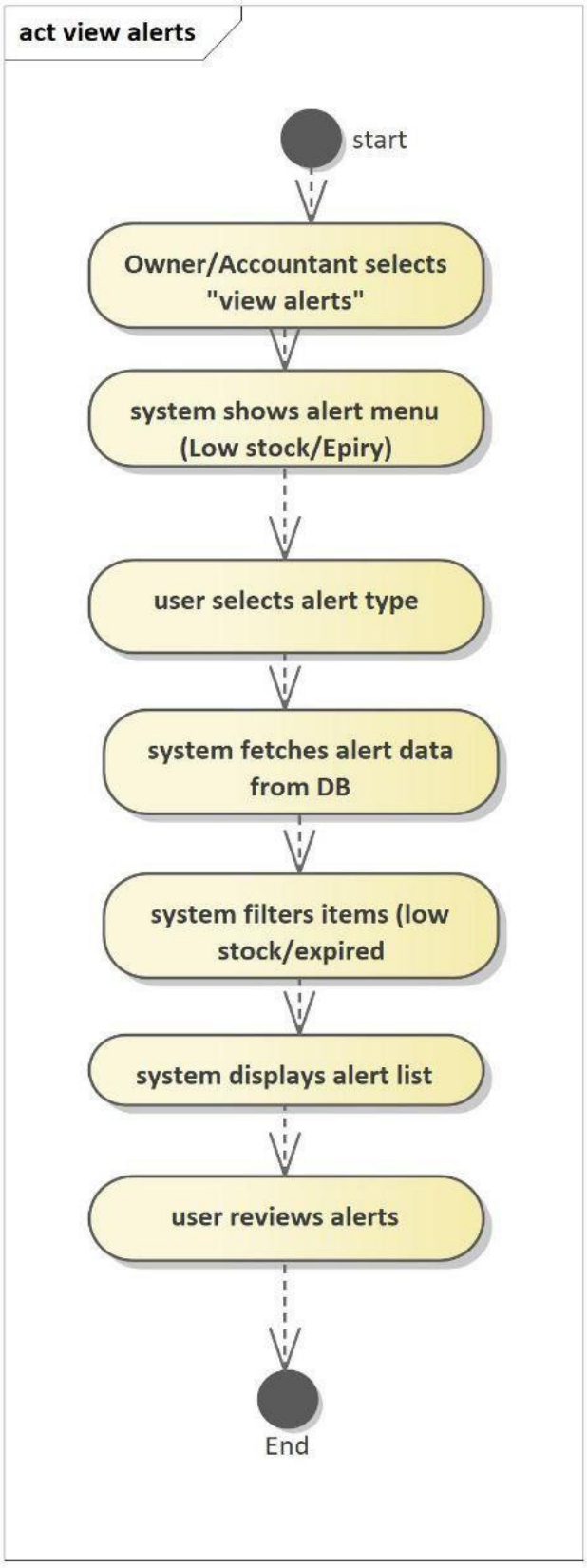
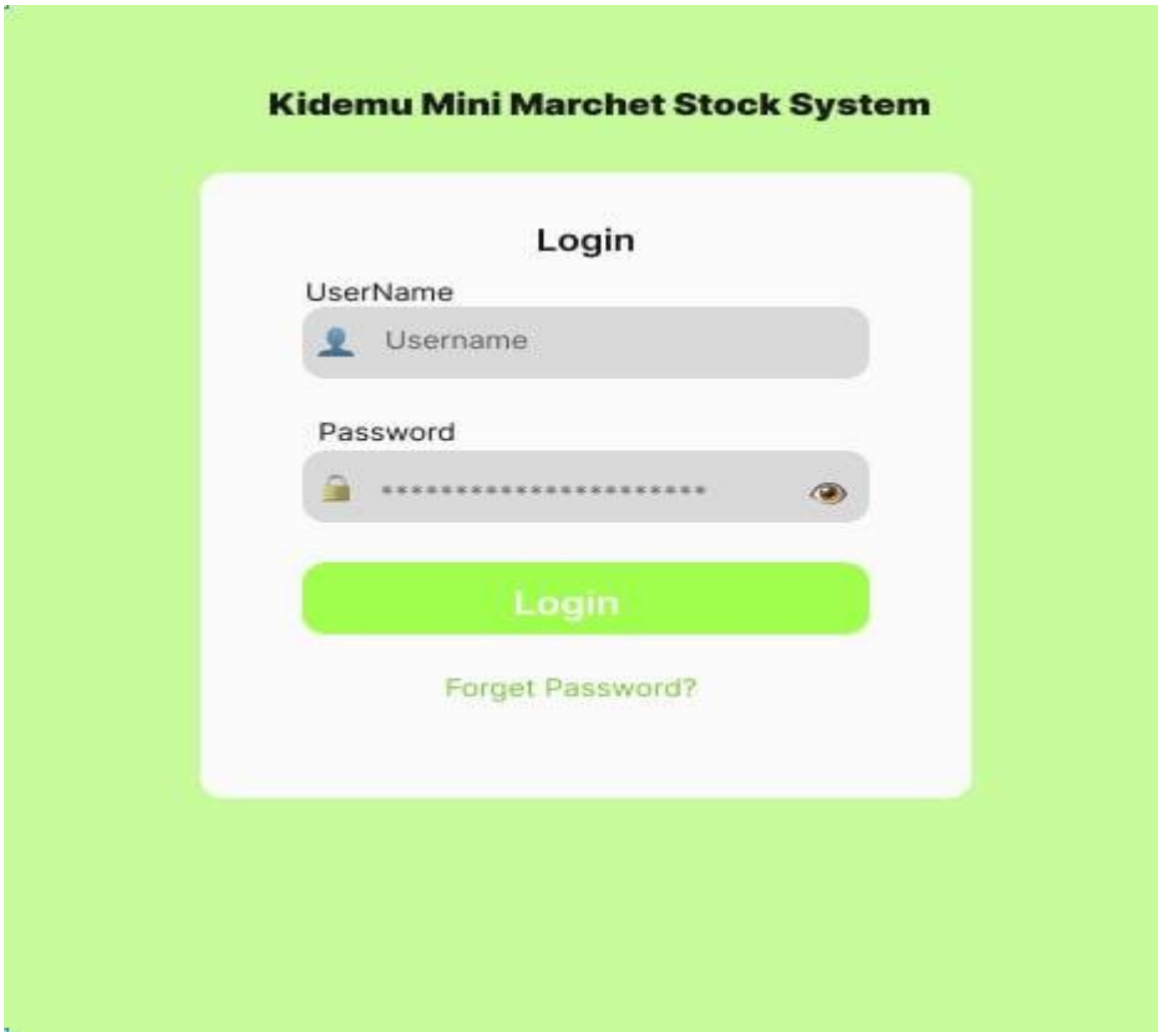


Figure_ 12

Activity diagram for usecase view alerts.

2.6.6.4 State Chart Diagram



2.6.6.5 Class Diagram

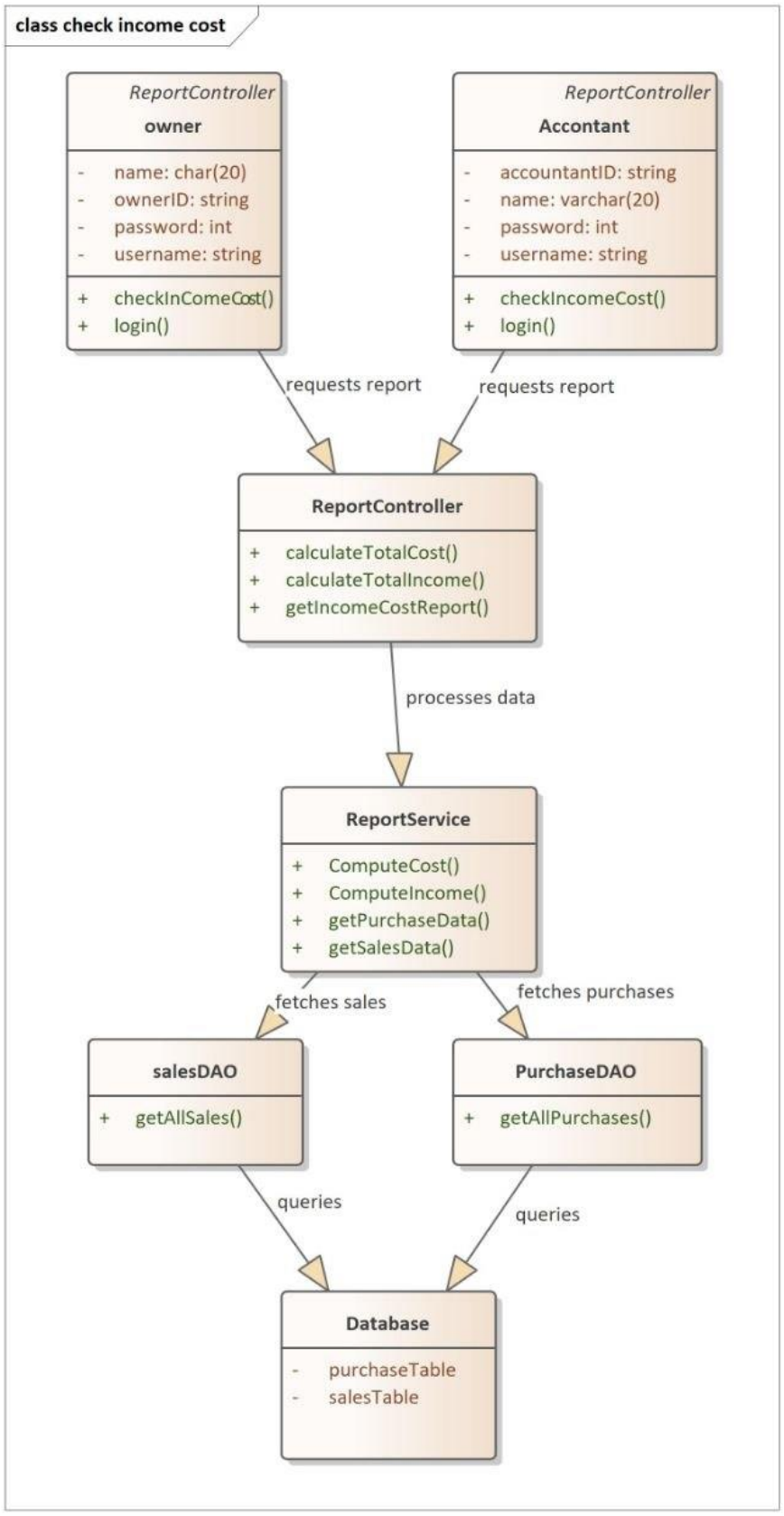


Figure_ 13

Class diagram for usecase check income cost.

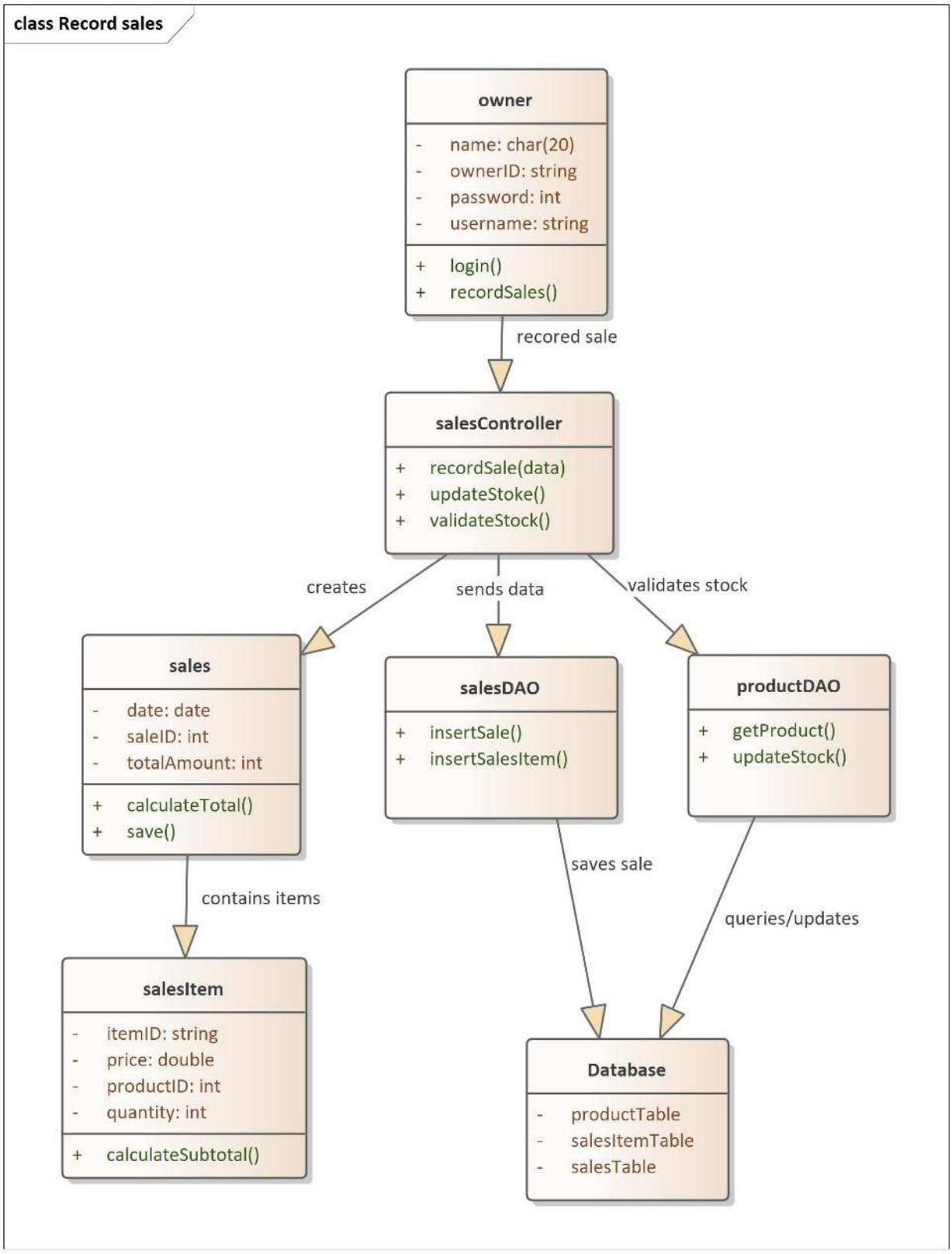


Figure_ 14

Class diagram for usecase record Seles

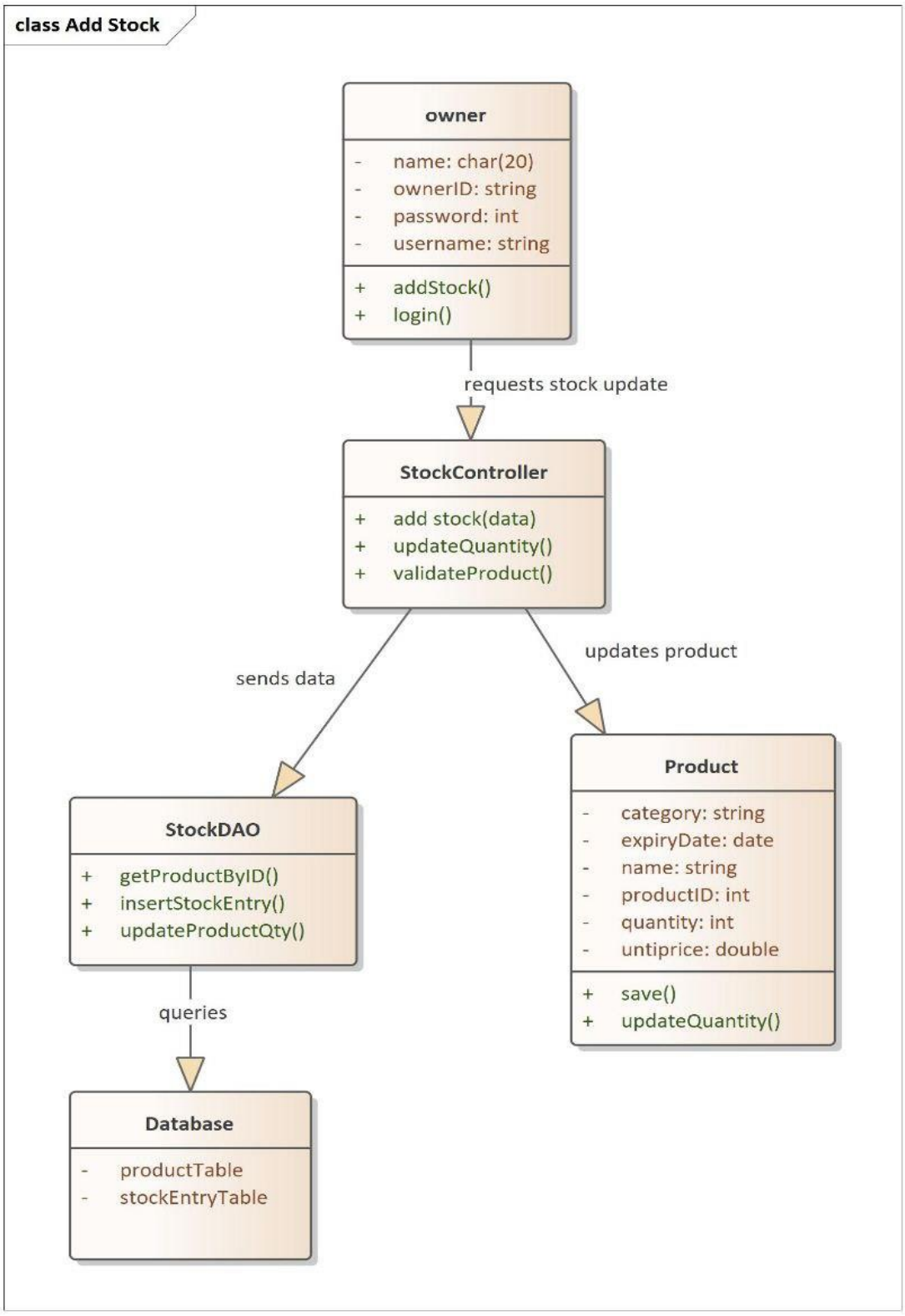


Figure_ 15

Class diagram for usecase add stock.

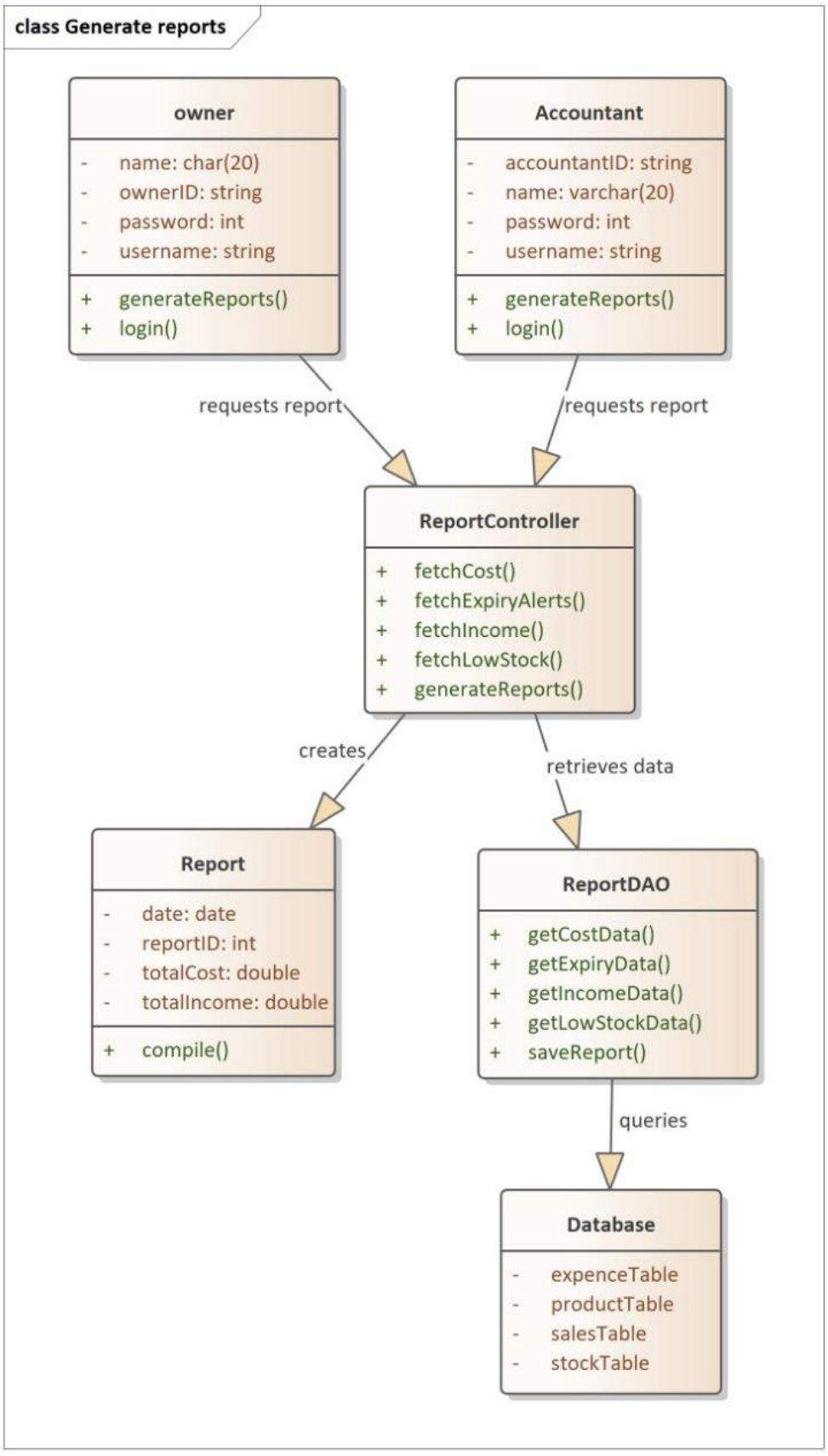


Figure_ 16

Class diagram for usecase generate report.

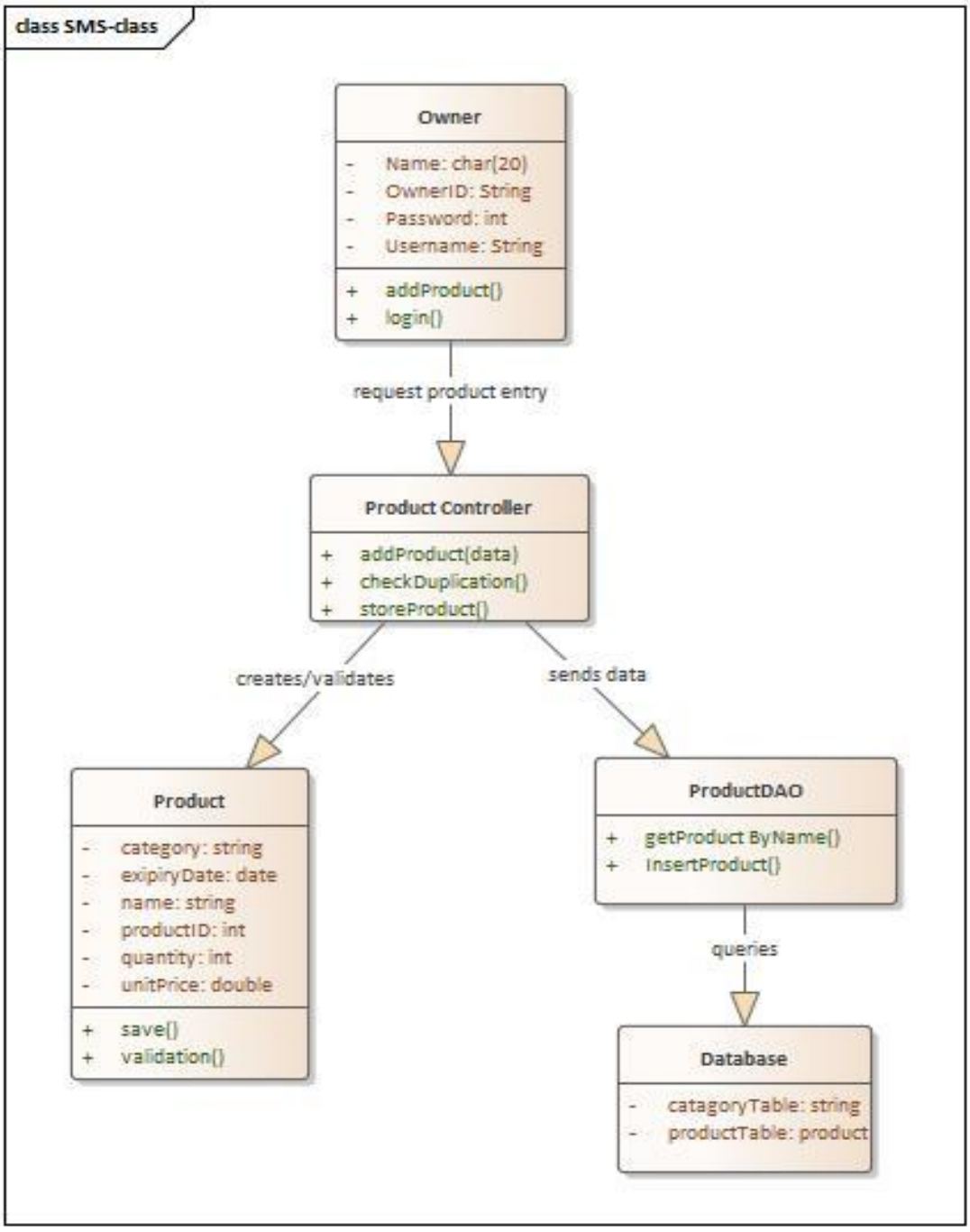


Figure_ 17

Class diagram for usecase add product.

2.6.6.6 User Interface Prototyping

CHAPATER 3 : Database Design

3.1 Conceptual Database Design

❖The conceptual database design defines the main data structure required for the Kidemu Mini Market Stock Management System. At this level, the focus is on identifying the essential data elements the system must store to support product registration, stock updates, sales recording, expiry tracking, and report generation. The conceptual model is independent of SQL Server and provides a clear overview of how information in the mini market will be organized.

3.1.1 Entity Identification and Description

- **Product:** Represents each item sold in the mini market.
- **Stock:** Stores quantity, expiry date, and minimum stock level for each product.
- **User:** Represents Owner and Accountant with login credentials.
- **Transaction:** Records stock-changing activities such as sales or stock additions.
- **Report:** Stores generated reports such as expiry report, low stock report, or income/cost report.

3.1.2 Attribute Identification and Description

1. Product Entity

Attribute	Description
ProductID (PK)	Unique identifier for each product.
ProductName	Name of the product.
Category	Type of the product (cosmetics, packaged food, etc.).
CostPrice	Buying price of the product.
SellingPrice	Price at which the product is sold.
Description	Additional information about the product.

2. Stock Entity

Attribute	Description
StockID (PK)	Unique identifier for each stock record.
ProductID (FK)	Links stock to the product it belongs to.
Quantity	Available number of items in stock.
ExpiryDate	Expiration date of the stock (if applicable).
MinStock	Minimum required quantity before alert.

3. User Entity

Attribute	Description
UserID (PK)	Unique identifier for each user.
UserName	Login username of the Owner or Accountant.
Role	User type (Owner or Accountant).
Password	User login password.

4. Transaction Entity

Attribute	Description
TransactionID (PK)	Unique identifier for each transaction.
ProductID (FK)	Product involved in the transaction.
UserID (FK)	User who performed the transaction.
TransactionType	Type of transaction (Sale, Stock Add, Expired).
Quantity	Number of items affected.
TransactionDate	Date the transaction took place.

5. Report Entity

Attribute	Description
ReportID (PK)	Unique identifier for each report.
ReportType	Type of report (Low Stock, Expiry, Income & Cost, Sales).

GeneratedDate	Date the report was created.
Content	Summary or details of the report.
UserID (FK)	User who generated the report.

Table_2_Attributes identification and description

3.1.3 Relationship Identification and Description

1. PRODUCT – STOCK

Relationship: One PRODUCT → Many STOCK records (1:N)
Description: A single product can appear in multiple stock entries, since each batch may have a different expiry date, quantity, or minimum stock level.

2. PRODUCT – TRANSACTION

Relationship: One PRODUCT → Many TRANSACTIONS (1:N)
Description: Every transaction—whether it's a sale, restock, or expired item record—must be tied to one specific product.

3. USER – TRANSACTION

Relationship: One USER → Many TRANSACTIONS (1:N)
Description: All transactions are recorded under the user who performed them, allowing the system to track who made each action.

4. USER – REPORT

Relationship: One USER → Many REPORTS (1:N)
Description: All system reports are generated by a specific user (Owner or Accountant), and the report history links back to that user.

3.1.4 ER Diagram

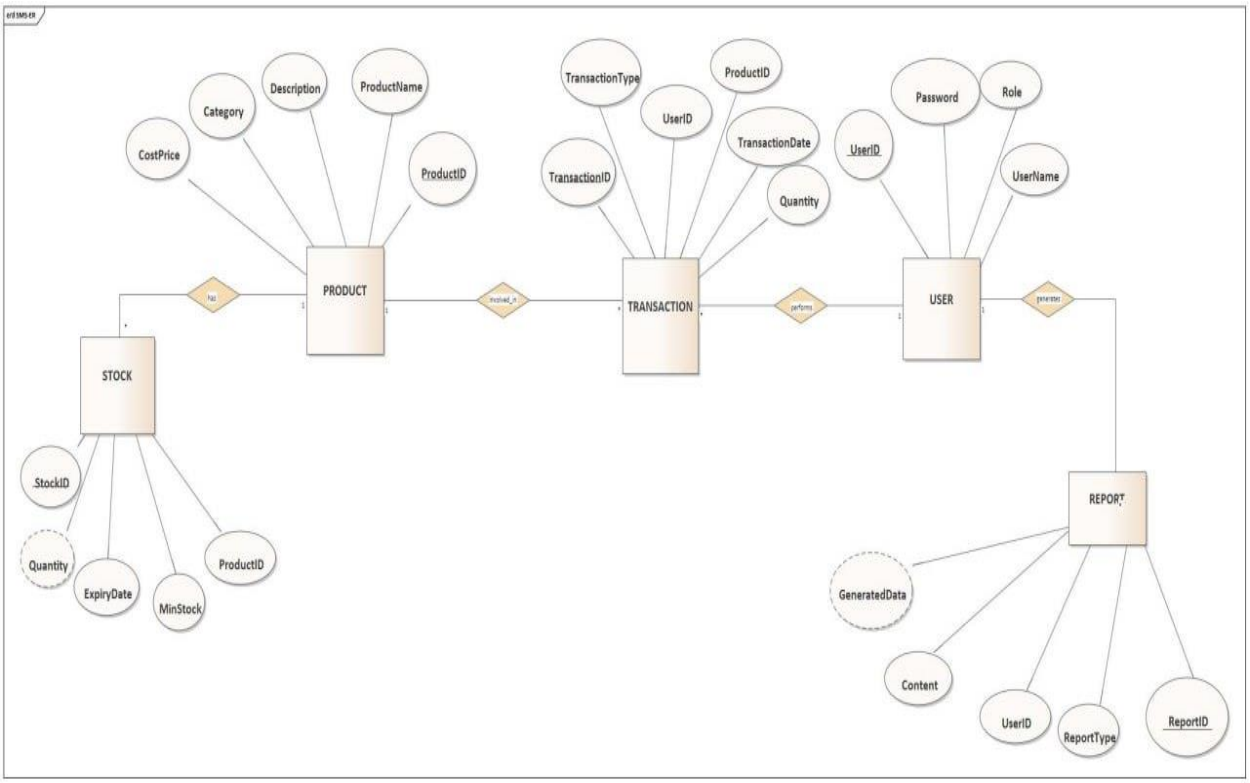


Figure_ 18

ER Diagram Description

- This ER diagram represents a mini-market stock management system. It includes four main entities: Product, Stock, Transaction, User, and Report.
- Product stores details such as name, category, description, and cost price.
- Stock keeps track of product quantities, expiry dates, and minimum stock levels.
- Transaction records product movements, including type, date, quantity, and the user who performed it.
- User represents system users who manage products, stocks, and transactions.
- Report contains generated summaries and analysis created by users.
- The relationships show how products are stored in stock, how users perform transactions on products, and how users generate reports about system activities.