[Pharmacy Management System]

By

Mohamed Badawy Sayed
Mostafa Ahmed Hasan El-Gelany

GitHub Repo

Dr.

Bassem Abd El-Atty Eldeeb

Mohamed Ahmed El kholy

Table of Contents

Requirements

Functional requirements

1 Sales:

1.1 Selling Medicine:

1.1.1 The system should allow employees to record sales transactions, including details such as customer information, medicine name, quantity sold, and unit price.

1.1.2 Integration with a barcode scanner or manual entry should be supported for efficient sales processes.

1.2 Inventory Management:

1.2.1 The system should automatically update the available quantity of each medicine after a sale is completed ensuring real-time inventory tracking.

1.2.2 Notifications should be generated when the stock of a particular medicine falls below a predefined threshold.

1.3 Sales Performance Tracking:

1.3.1 The system should provide tools for tracking and analyzing sales performance, including metrics such as total sales, top-selling medicines, and sales trends over time.

2 Purchase:

2.1 Buying Medicine from Distributors:

2.1.1 Employees should be able to create purchase orders for medicines from authorized distributors, specifying details like quantity, and expiration dates.

2.2 Inventory increase:

2.2.1 The system should automatically update the available quantity of each purchased medicine, reflecting the increase in stock.

3 Reports:

3.1 Sales Reports:

3.1.1 The system should generate detailed reports on sales activities, including daily, weekly, and monthly summaries, as well as custom date ranges.

3.1.2 Reports should include information on total sales, and top-selling medicines.

3.2 Purchase Reports:

3.2.1 Detailed reports on purchases should be available, displaying information such as total expenditure, and quantities purchased.

3.3 Medicine Reports:

3.3.1 Comprehensive reports on medicine inventory, including stock levels, expiration dates, and reorder suggestions, should be generated.

3.4 Net Profit Reports:

3.4.1 The system should calculate and provide reports on the net profit, taking into account both sales and purchase transactions.

4 Alerts:

4.1 Expiration Alerts:

4.1.1 The system should generate alerts for employees when the expiration date of a medicine is approaching, allowing timely actions such as marking for discounts or removal from shelves.

5 Employees:

5.1 User Authentication:

5.1.1 The system should have a secure login mechanism with role-based access control, ensuring that employees can only access functions relevant to their roles.

5.2 Activity History:

5.2.1 A detailed activity log should be maintained for each employee, capturing actions such as logins, logouts, sales transactions, purchases, and any system operations.

6 Return:

6.1 Return to Distributors:

6.1.1 Employees should be able to process returns of medicines to distributors within the specified time frame, recording relevant details such as reasons for return and quantities returned.

6.2 Inventory Update:

6.2.1 The system should automatically adjust the available quantity of returned medicines in the inventory.

6.3 Stock Reduction:

6.3.1 The return process should result in a decrease in the number of available medicines, reflecting accurate stock levels.

Non-functional requirements:

1. **Performance**:

1.1 The system should respond to user interactions within two seconds under normal operating conditions.

1.2 The time taken to generate reports, should not exceed five seconds.

2. **Reliability**:

2.1 The system should have a mean time between failures at least 500 hours.

2.2 In the event of a failure, the system should recover within five minutes with minimal data loss.

3. **Security**:

3.1 All sensitive customer and employee data should be encrypted during transmission and storage.

3.2 Role-based access control should be enforced, ensuring that employees can only access functionalities based on their assigned roles.

3.3 Passwords should be stored securely using industry-standard hashing algorithms.

4. Scalability:

4.1 It should support the addition of new features and functionalities without requiring a complete system overhaul.

4.2 Scalability tests should be conducted to ensure optimal performance under increased loads.

5. Usability:

5.1 The user interface should be intuitive and require minimal training for new employees.

5.2 The system should provide clear error messages and guidance to users in case of input errors.

6. Availability:

6.1 The system should be available 100% of the time during regular business hours.

6.2 Scheduled maintenance should be communicated in advance, and efforts should be made to minimize downtime.

7. Backup and Recovery:

7.1 Regular automated backups of the database and system configuration should be performed.

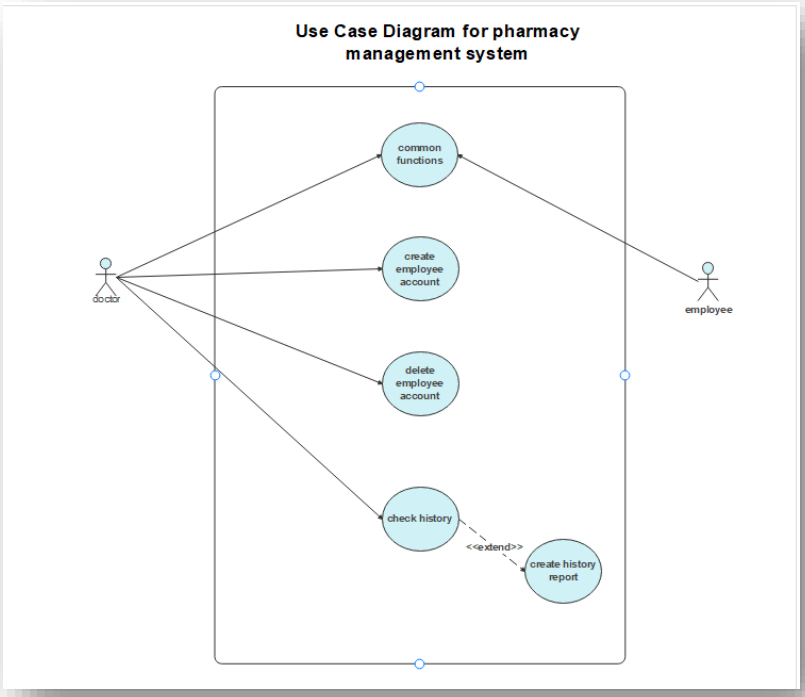
7.2 The backup data should be stored in a geographically separate location to ensure recovery in case of a catastrophic event.

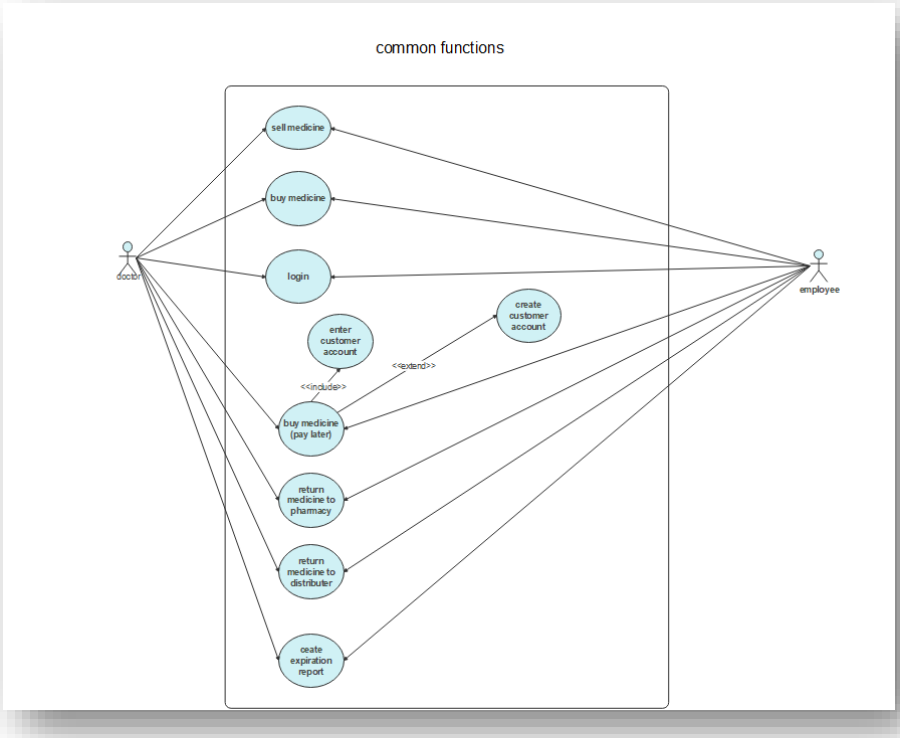
8. Dependability:

8.1 The system should be designed with built-in fault tolerance to minimize the impact of hardware or software failures.

8.2 Dependencies on external services or APIs should be identified.

Use Case Diagram





Use Case Scenario

case name:	Login
case ID:	Op1
Actor(s):	Doctor , employee
Description:	User login to the system
Triggering:	User want to use the system
Preconditions :	User isn't logged in
Main path :	1. Enter system 2. Enter username 3. Enter password Logged in successfully
Alter Path :	Username or password isn't correct 1. System display a message says that invalid user name or password
Postconditions:	User logged in successfully
Priority:	Medium

case name:	Buy medicine
case ID:	Op2
Actor(s):	Doctor , employee
Description:	User buy medicine from distributers
Triggering:	Some types of medicine got finished or about to finish from pharmacy
Preconditions :	Pharmacy needs some types of medicine
Main path :	1. Enter type of medicine 2. Enter amount of medicine 3. Repeat this step for all types of needed medicine 4. Click buy button 5. Show a message that display the amount of needed money
Postconditions:	Medicine bought successfully
Priority:	high

case name:	Sell medicine
case ID:	Op3
Actor(s):	Doctor , employee
Description:	User enter type of medicine and amount of that type to be sold
Triggering:	A customer came to buy some medicine
Preconditions :	Medicine isn't bought yet
Main path :	1. Enter type of medicine 2. Enter amount of medicine to be sold 3. Repeat this step for all types of needed medicine 4. Click sell button 5. Show a message that display the amount of required money
Alter Path :	Some kind of medicine is not in the pharmacy Show a message says that "some kind of medicine is not enough"
Postconditions:	Medicine sold to customer successfully
Priority:	high

case name:	Return medicine to pharmacy
case ID:	Op4
Actor(s):	Doctor , employee
Description:	A customer came to return some medicine after buying it
Triggering:	A customer came to return some medicine
Preconditions :	Medicine isn’t returned yet
Main path :	1. Enter type of medicine 2. Enter amount of medicine to be returned 3. Repeat this step for all types of needed medicine 4. Click return button 5. Show a message that display the amount of money to return to customer
Postconditions:	Medicine returned successfully
Priority:	low

case name:	Return medicine to distributer
case ID:	Op5
Actor(s):	Doctor , employee
Description:	Pharmacy returns medicine to distributer
Triggering:	1. Expiration date of medicine is near 2. There as a big amount of some kind of medicine 3. Pharmacy needs urgent money
Preconditions :	Bought Medicine is still in the pharmacy
Main path :	1. Enter type of medicine 2. Enter amount of medicine to be sold 3. Repeat this step for all types of needed medicine 4. Click return button 5. Show a message that display the amount of required money
Alter Path :	Some kind of medicine is not in the pharmacy Show a message says that “some kind of medicine is not enough”
Postconditions:	Medicine return to distributer successfully
Priority:	low

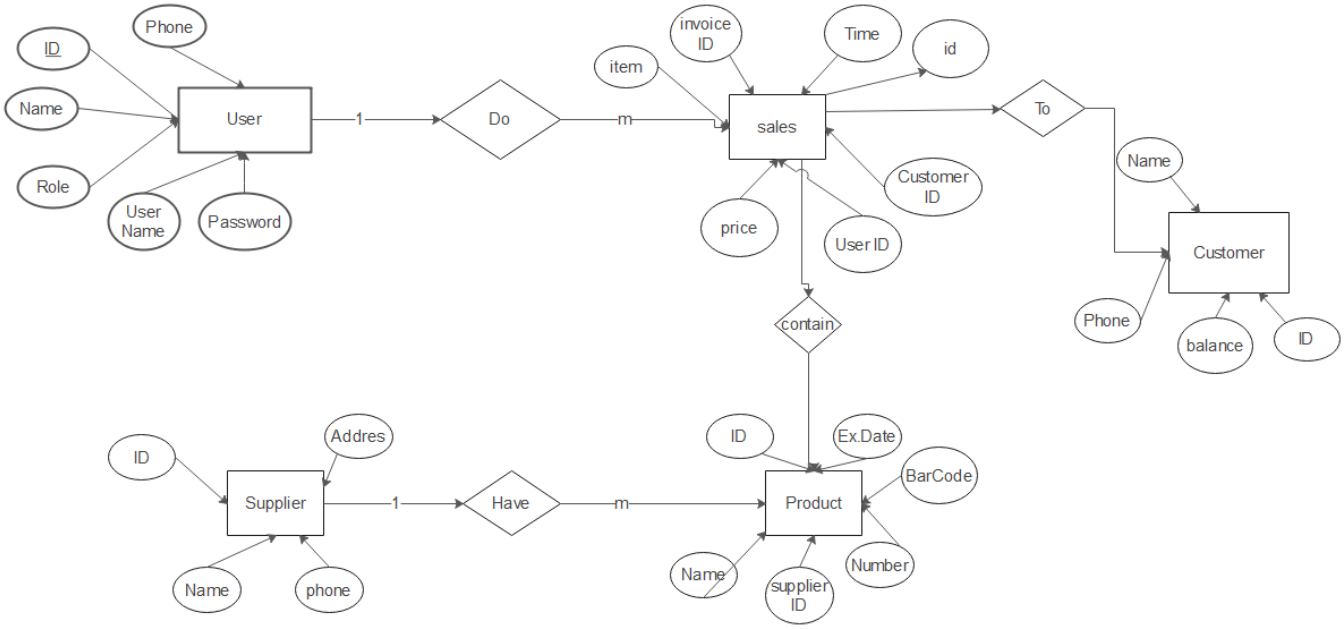
case name:	Create expiration report
case ID:	Op6
Actor(s):	Doctor , employee
Description:	User create report about medicine that is about to get expired
Triggering:	user need to contact with distributer about medicine that need to return
Preconditions :	Report is not created yet
Main path :	1. Enter expiration notification tab 2. Enter button create report 3. Show message says “report created successfully”
Alter Path :	There is no medicine that is about to expire Show message says that “no medicine to get reported”
Postconditions:	Report created successfully
Priority:	low

case name:	Create employee account
case ID:	Op7
Actor(s):	Doctor
Description:	Doctor create new account for new employee to enable him use the system
Triggering:	Doctor hires new employee
Preconditions :	Employee can’t access the system
Main path :	1. Enter username 2. Enter name 3. Enter password 4. Enter password again 5. Enter button create account 6. Show message “account created successfully”
Alter Path :	Case 1: Password entered is not the same in 2 fields Show message “password is not correct” Case 2: username is empty have invalid character Show message “invalid username” Case 3: name is empty Show message “enter name”
Postconditions:	Account created successfully
Priority:	medium

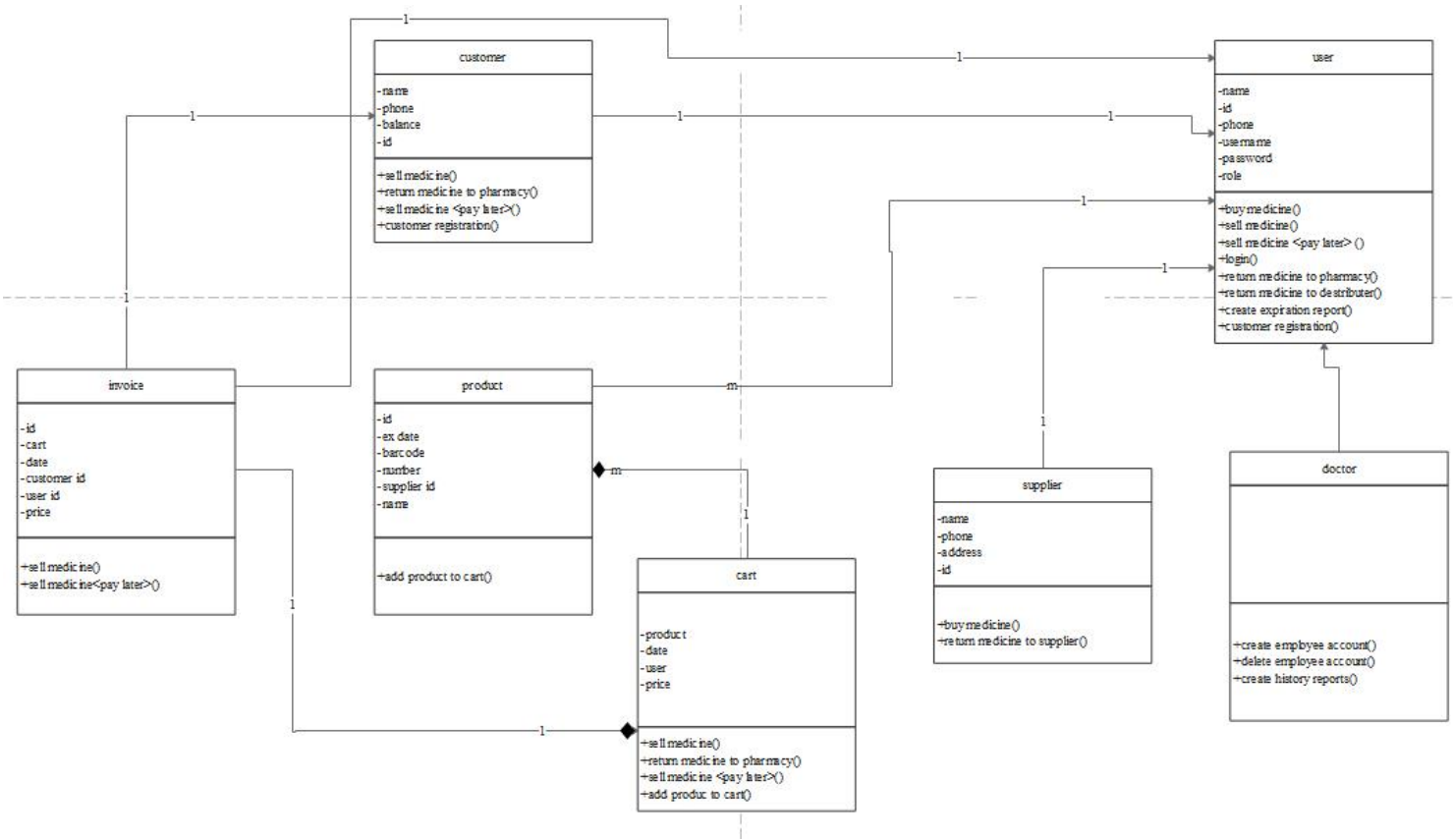
case name:	delete employee account
case ID:	Op8
Actor(s):	Doctor
Description:	Doctor deletes employee's account because this employee is no longer work in the pharmacy
Triggering:	Employee is no longer working in the pharmacy
Preconditions :	Account is still in the system
Main path :	1. Enter username 2. Enter password 3. Enter button delete account 4. Show message "account deleted successfully"
Alter Path :	Account is not in the system Show message "invalid username or password"
Postconditions:	Account deleted successfully
Priority:	low

case name:	Check history
case ID:	Op9
Actor(s):	Doctor
Description:	Doctor check history of all operations that done on the system
Triggering:	Doctor need to check about pharmacy finance
Preconditions :	Doctor need to check history
Main path :	Enter button history (extend) Enter button create report
Postconditions:	Doctor checked history
Priority:	high

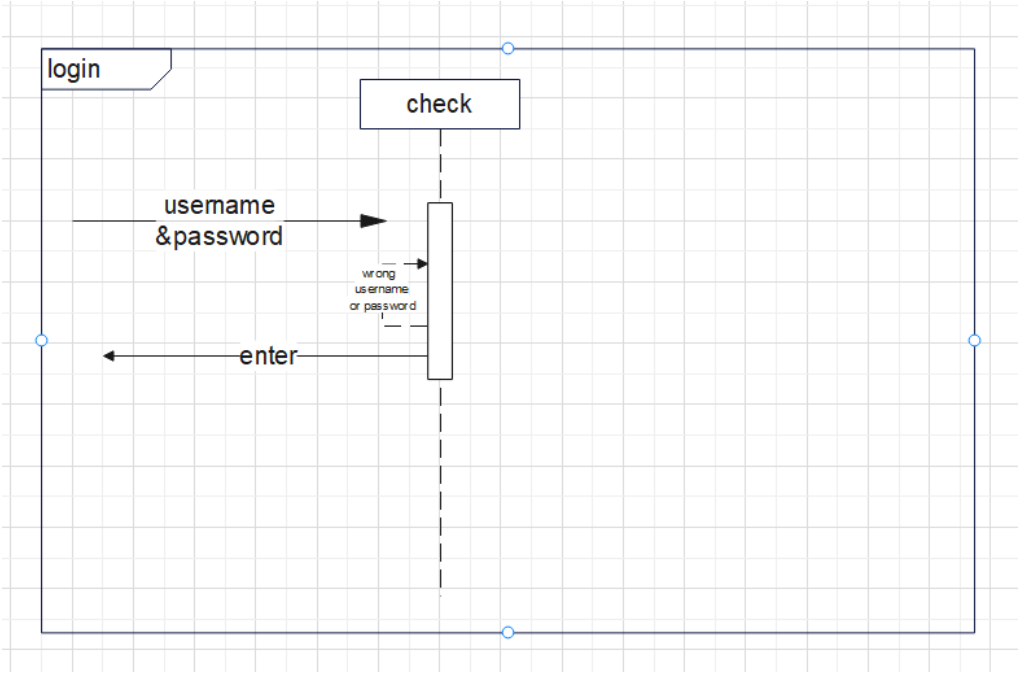
ERD Diagram

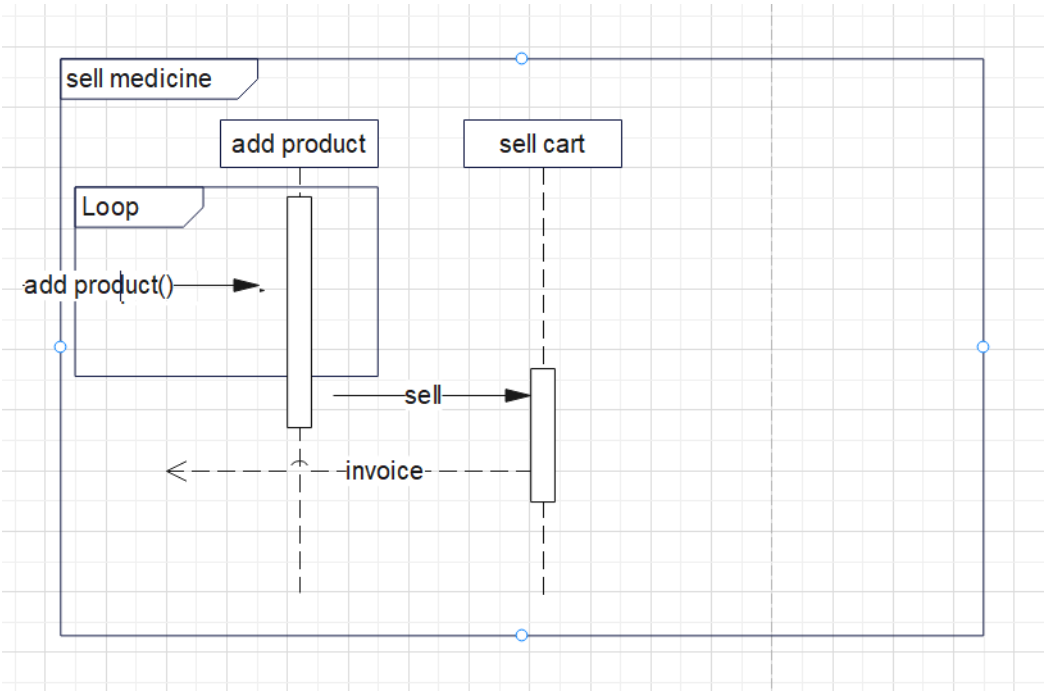

ID
Phone
Name
User
Role
User Name
Password
1
Do
m
item
invoice ID
Time
id
sales
price
User ID
Customer ID
To
Name
Customer
Phone
balance
ID
contain
ID
Addres
Supplier
Name
phone
1
Have
m
ID
Ex.Date
BarCode
Product
Name
supplier ID
Number

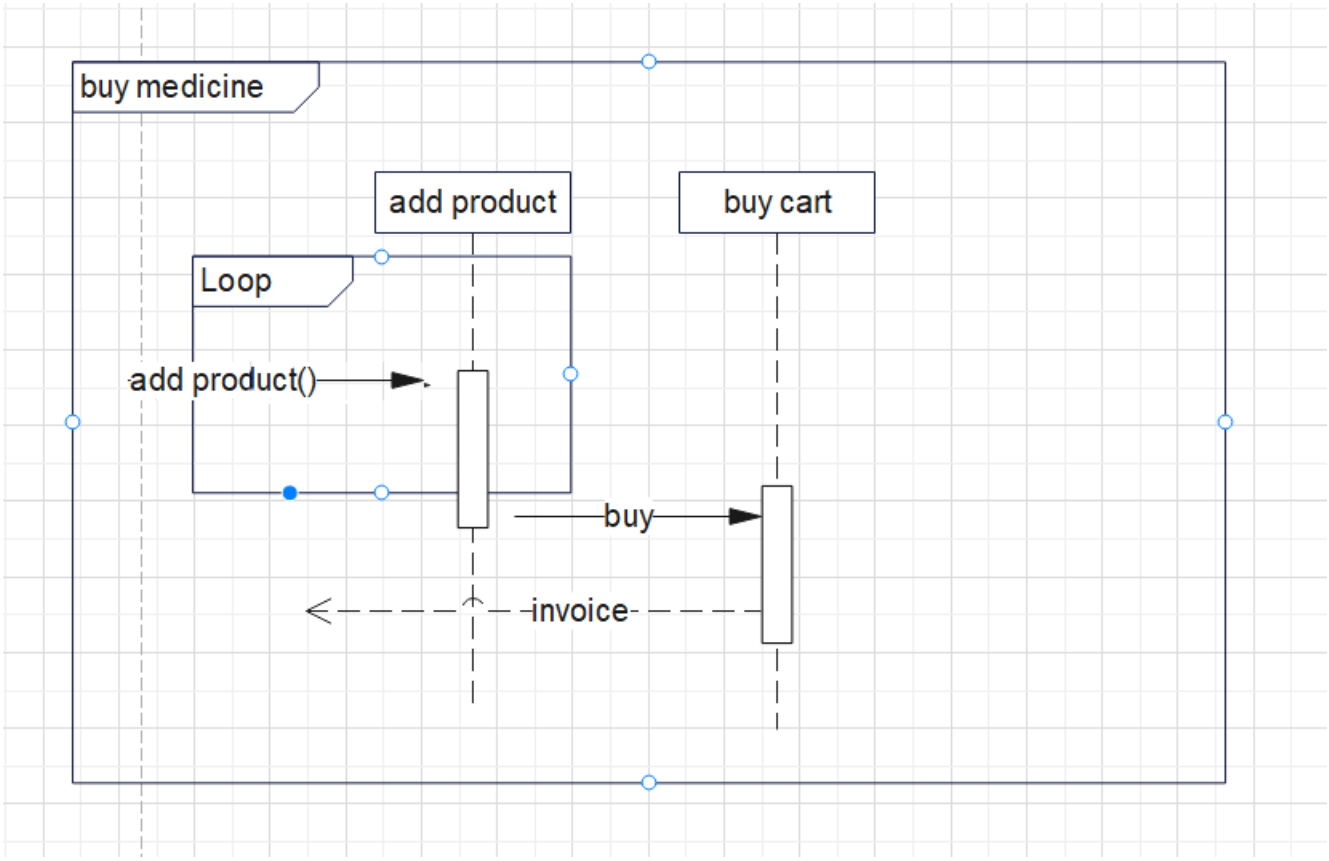
Class Diagram

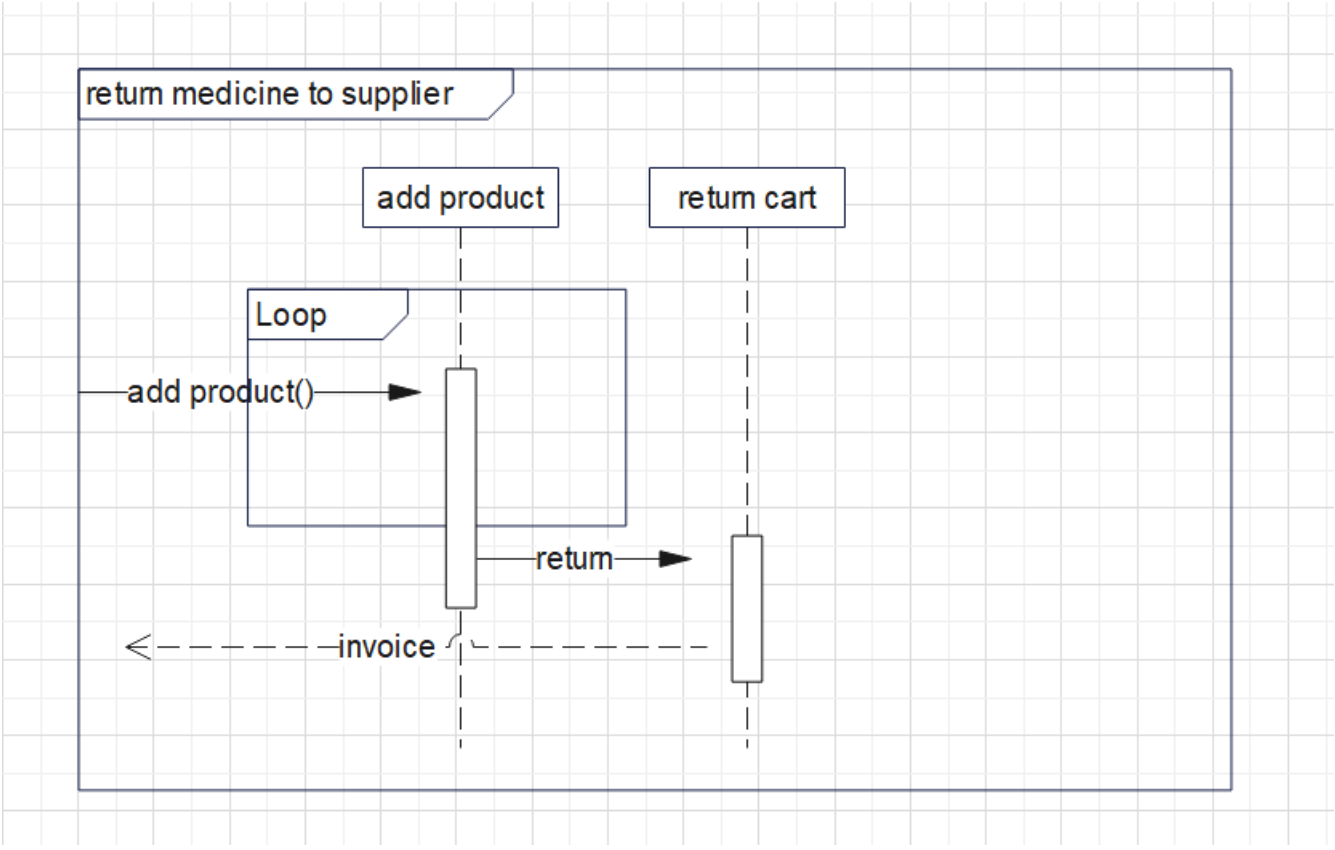


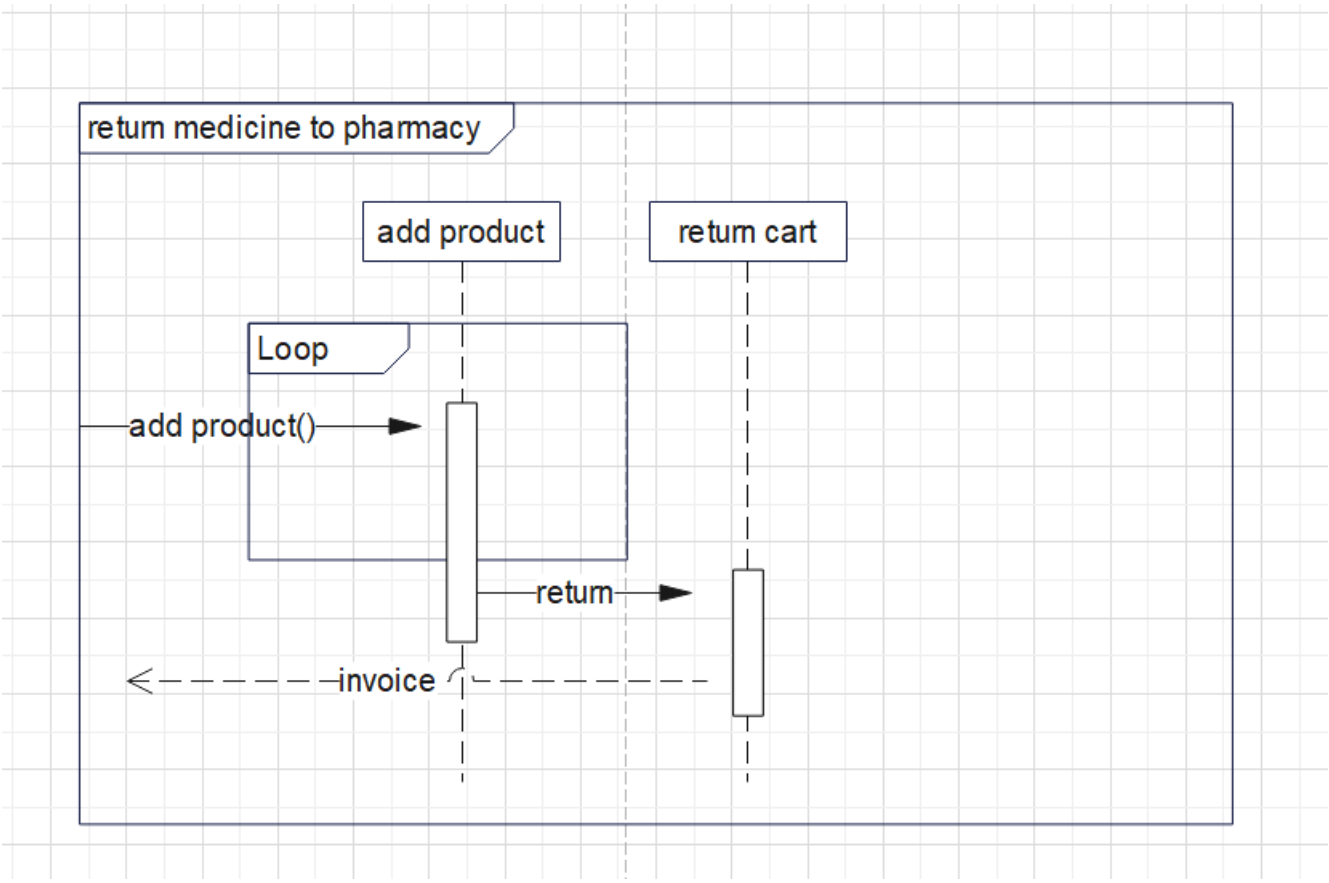
Seqance Diagram

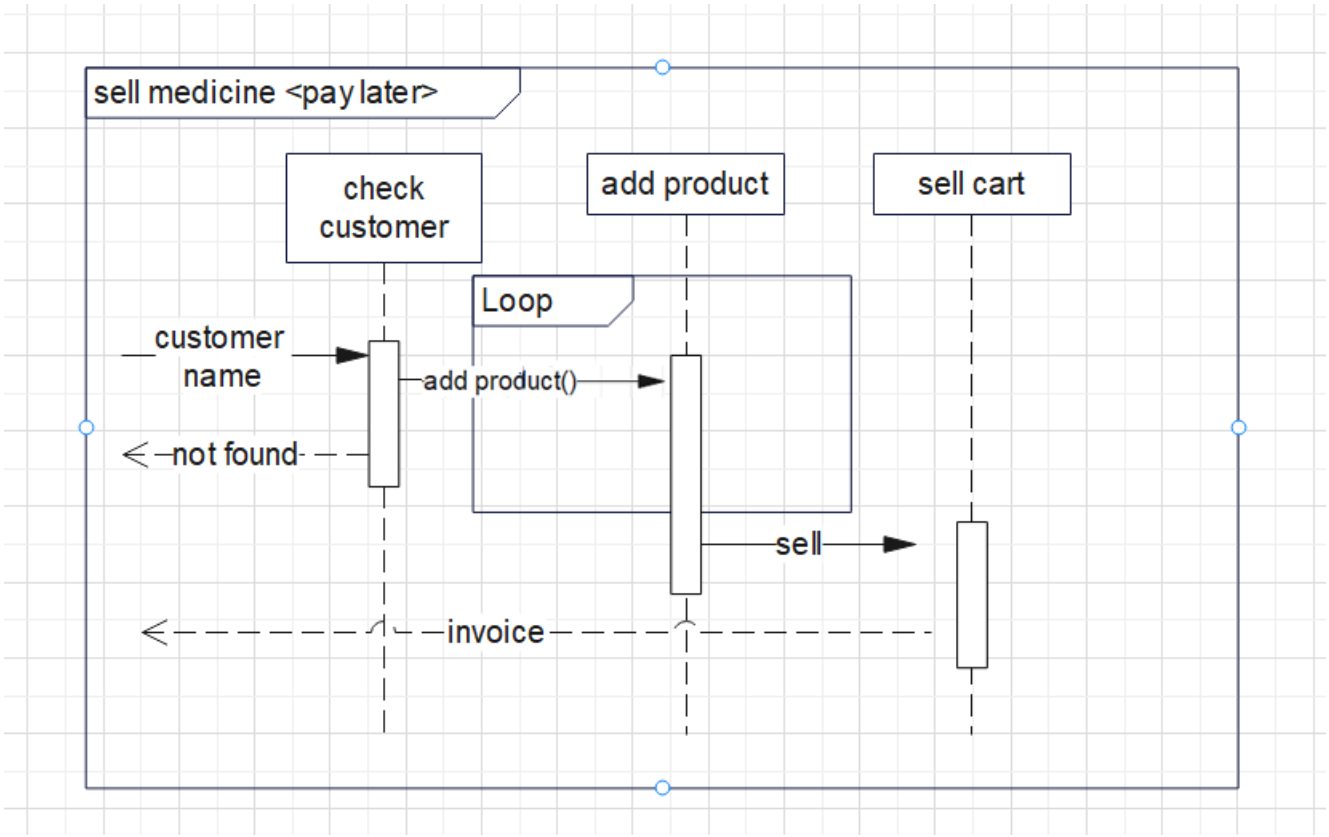





buy medicine
add product
buy cart
Loop
add product()
buy
invoice


return medicine to supplier
add product
return cart
Loop
add product()
return
invoice


return medicine to pharmacy
add product
return cart
Loop
add product()
return
invoice


sell medicine <pay later>
check customer
add product
sell cart
Loop
customer name
add product()
not found
sell
invoice