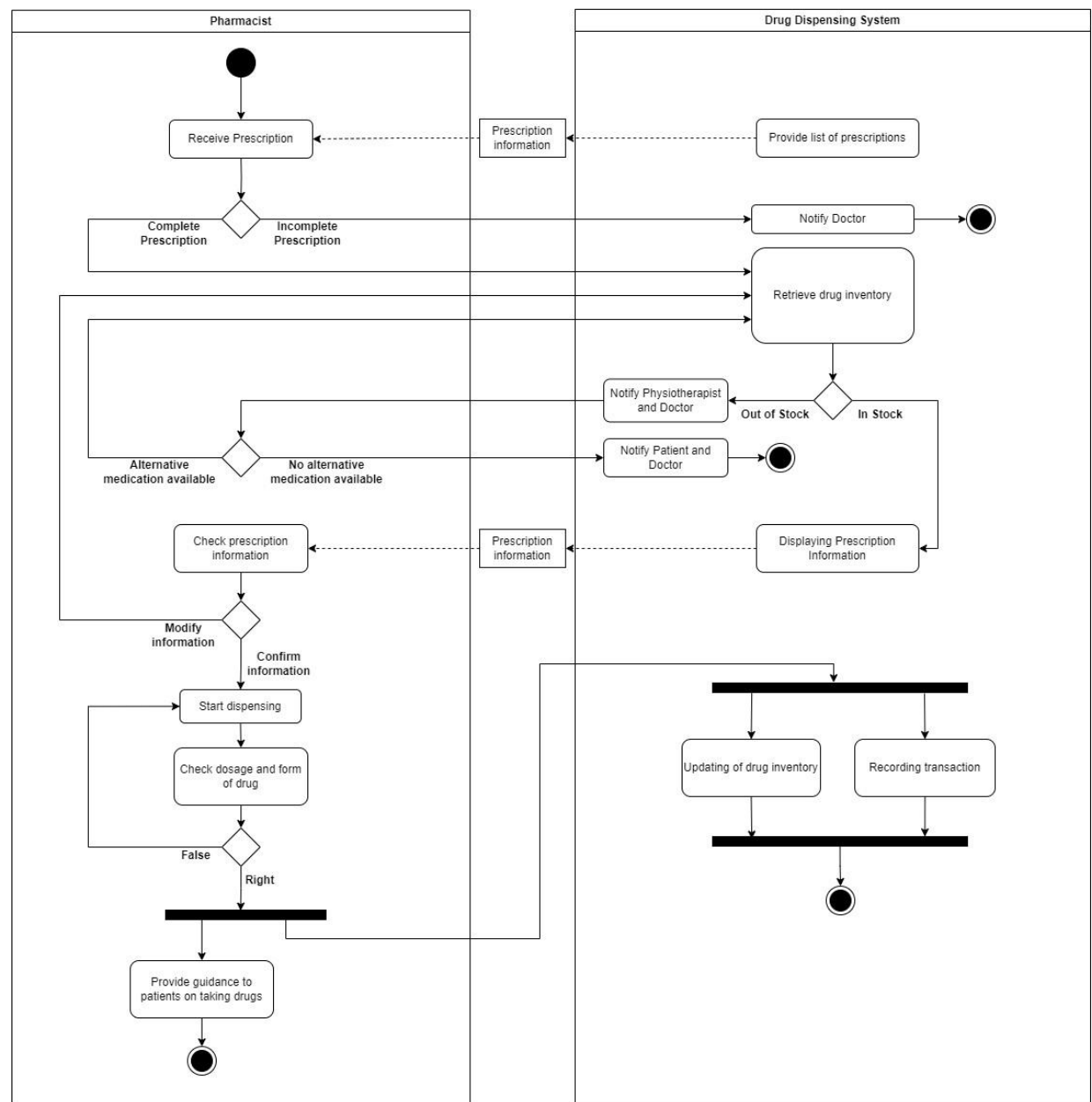
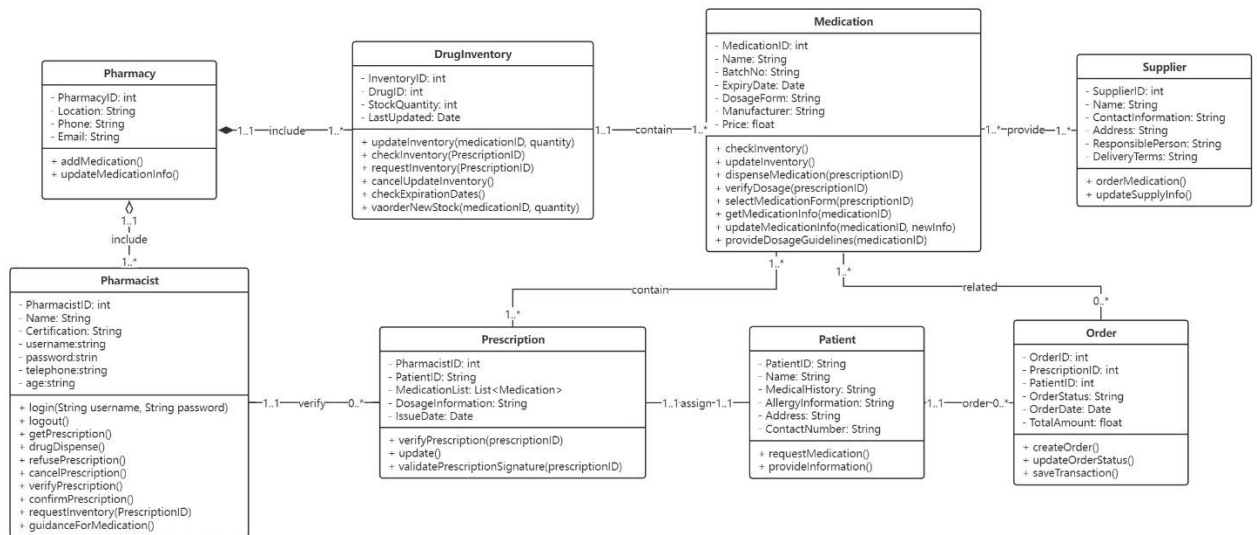


Task 1:

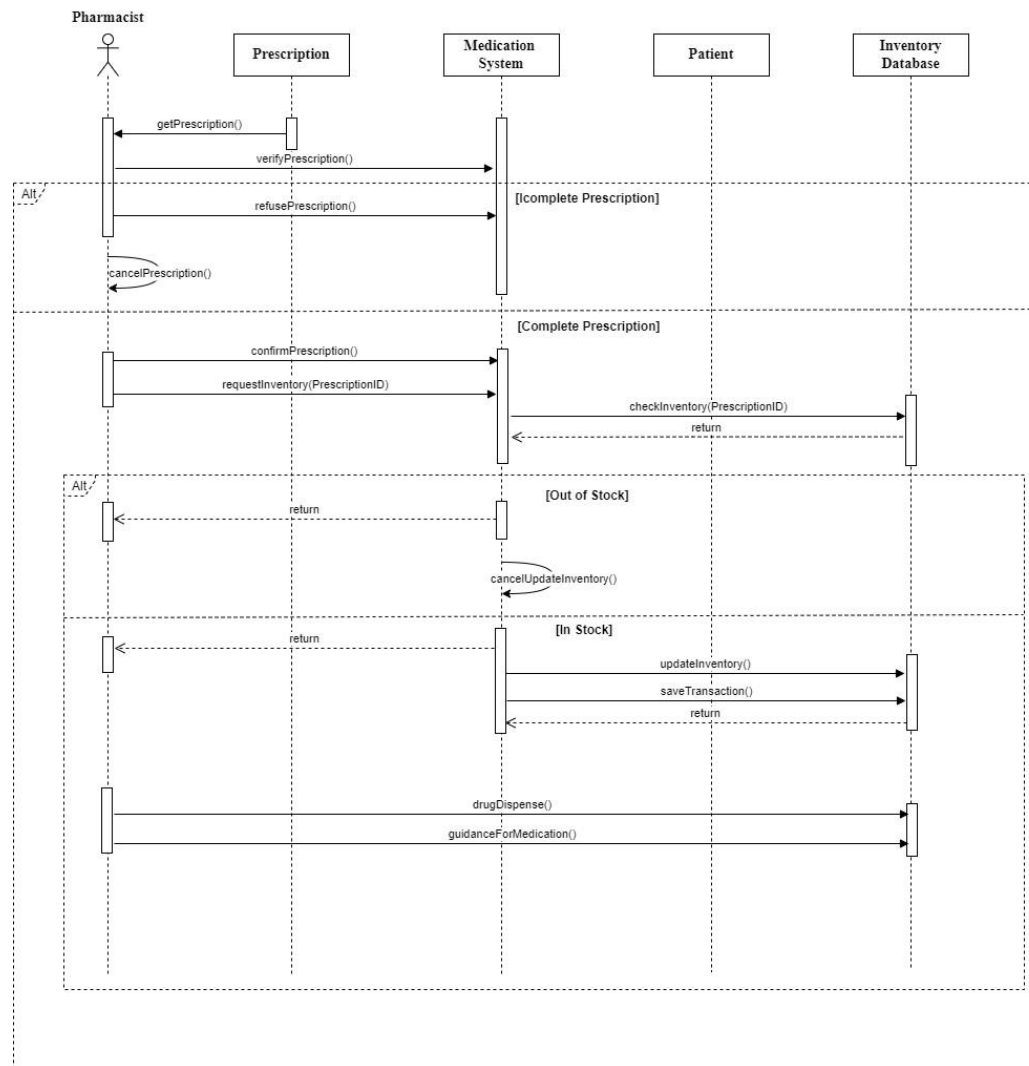
a) Activity diagram



b) Class diagram

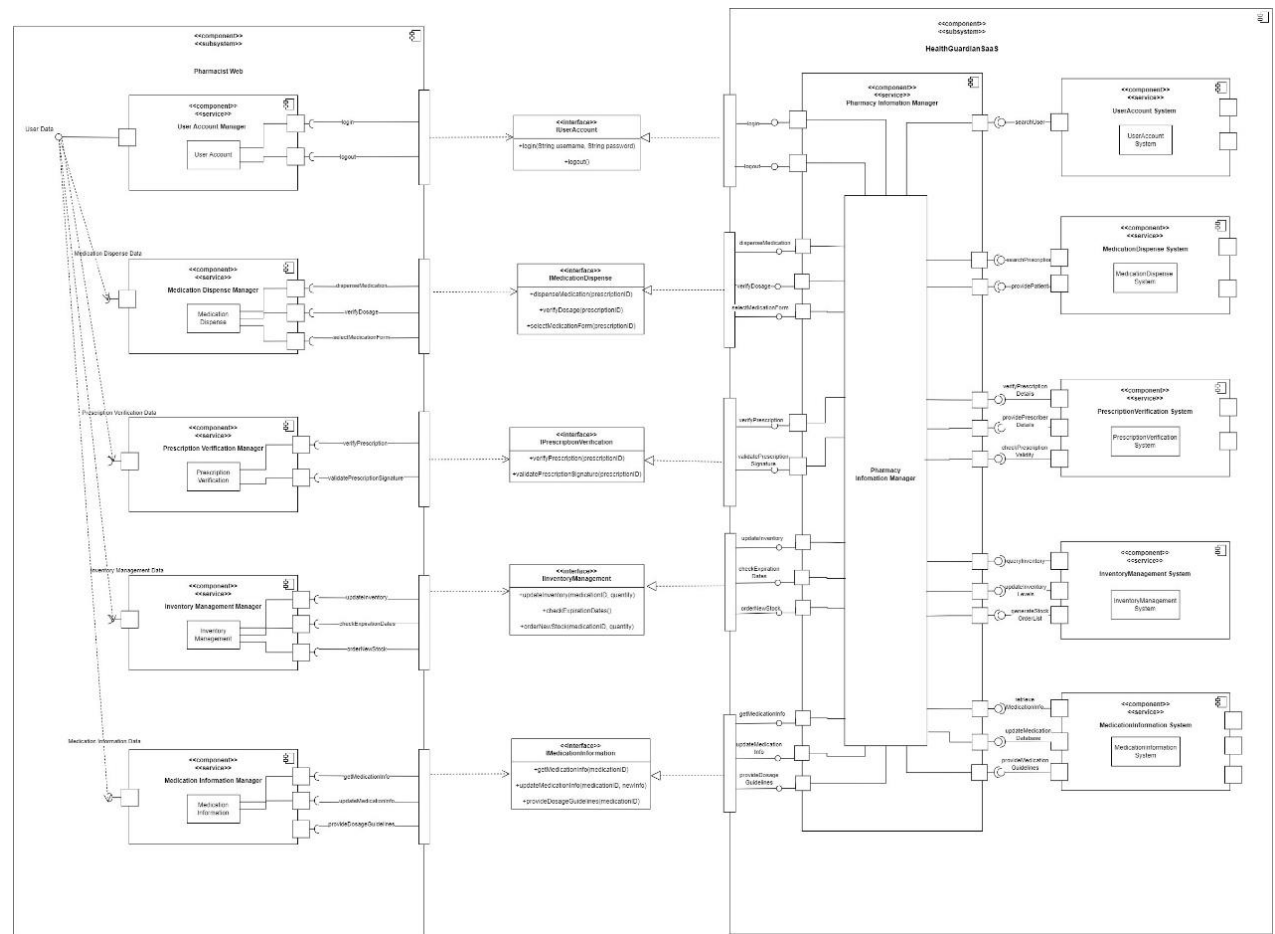


c) Behavior diagram



Task 2:

a) Component diagram



b) Table

Component table:

Component	User Account Manager
Description	This is a service operating in the cloud, managing and maintaining user account information for pharmacists and patients, including authentication and authorization control.
Type	Service
Required Interfaces:	User Authentication
Provided Interfaces:	User Registration, User Login, Permissions Update, Account Information Maintenance

Component	Medication Dispense Manager
Description	This cloud-based service processes medication dispensing requests, ensuring medications are dispensed correctly and safely to patients.
Type	Service
Required Interfaces:	IMedicationDispense
Provided Interfaces:	Dispensing Records, Dispensing Status Update, Dispensing Scheduling

Component	Prescription Verification Manager
Description	This service is tasked with verifying the authenticity and accuracy of prescriptions issued by doctors, preventing errors or misuse, in the cloud environment.
Type	Service
Required Interfaces:	IPrescriptionVerification
Provided Interfaces:	Prescription Review, Verification Records, Doctor Information Confirmation

Component	Inventory Management Manager
Description	Operating in the cloud, this service is responsible for monitoring and managing medication inventory levels, ensuring the continuity and compliance of drug supply.
Type	Service
Required Interfaces:	IInventoryManagement
Provided Interfaces:	Inventory Level Monitoring, Inventory Updates, Inventory Report Generation

Component	Medication Information Manager
Description	This is a cloud service for storing, retrieving, and updating the medication information database, providing accurate drug knowledge and data.
Type	Service
Required Interfaces:	IMedicationInformation
Provided Interfaces:	Medication Information Query, Medication Information Update, Drug Interaction Check

Component	Pharmacy Information Manager
-----------	------------------------------

Description	This central service orchestrates the flow of information between different systems within the pharmacy management architecture. It ensures data coherence and provides a single point of integration for user accounts, medication dispensing, prescription verification, inventory management, and medication information systems.
Type	Service
Required Interfaces:	IUserAccount, IMedicationDispense, IPrescriptionVerification, IInventoryManagement, IMedicationInformation
Provided Interfaces:	Central Data Coordination, System Integration, Data Analytics, Reporting

Component	User Account System
Description	This system manages all aspects of user account creation, authentication, and authorization. It ensures that user data is securely handled and that appropriate access levels are maintained for both pharmacists and patients.
Type	Service
Required Interfaces:	IUserAccount
Provided Interfaces:	Account Creation, Authentication, Authorization, User Data Management

Component	Medication Dispense System
Description	This system manages the dispensing of medications. It tracks prescriptions, dispensing activities, and integrates with inventory systems to ensure that medications are available and dispensed accurately.
Type	Service
Required Interfaces:	IMedicationDispense
Provided Interfaces:	Dispense Processing, Medication Tracking, Dispense Verification, Interaction Alerts

Component	Prescription Verification System
-----------	----------------------------------

Description	This system validates prescriptions against medical records and medication information to ensure that prescriptions are accurate and appropriate for patients.
Type	Service
Required Interfaces:	IPrescriptionVerification
Provided Interfaces:	Prescription Validation, Verification Logging, Prescriber Verification, Compliance Checking

Component	Inventory Management System
Description	This system is responsible for managing the stock levels of medications. It tracks inventory, forecasts demand, and signals when restocking is necessary to prevent shortages.
Type	Service
Required Interfaces:	IInventoryManagement
Provided Interfaces:	Inventory Tracking, Demand Forecasting, Stock Replenishment, Inventory Auditing

Component	Medication Information System
Description	This system maintains a comprehensive database of medication information, including dosages, contraindications, side effects, and interactions. It supports other systems by providing detailed medication data for dispensing and verification processes.
Type	Service
Required Interfaces:	IMedicationInformation
Provided Interfaces:	Medication Data Retrieval, Information Updates, Medication Interaction Checking, Drug Information Distribution

Interface tables:

Interface Name	IUserAccount
Provider	User Account System
	<i>Operation:</i> userLogin
	<i>Request Parameters:</i> userName, password

	<p><i>Response parameters:</i> userID, accessToken, errorMessage</p> <p><i>Function:</i> Validates user credentials and, if successful, returns a user ID and access token.</p>
--	---

Interface Name	IMedicationDispense
Provider	Medication Dispense System
	<p><i>Operation:</i> dispenseMedication</p> <p><i>Request Parameters:</i> prescriptionID, quantity</p> <p><i>Response parameters:</i> dispenseConfirmation, errorMessage</p> <p><i>Function:</i> Dispenses medication based on a valid prescription and, if successful, returns a confirmation of dispensing.</p>
	<p><i>Operation:</i> cancelDispense</p> <p><i>Request Parameters:</i> dispenseID</p> <p><i>Response parameters:</i> cancelConfirmation, errorMessage</p> <p><i>Function:</i> Cancels a medication dispensing and, if successful, returns a confirmation of the cancellation.</p>

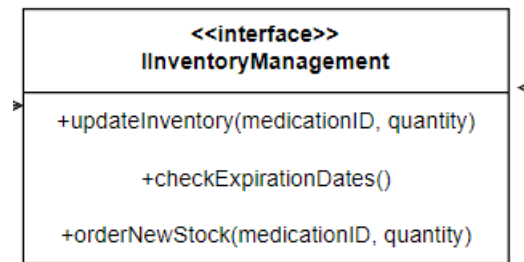
Interface Name	IPrescriptionVerification
Provider	Prescription Verification System
	<p><i>Operation:</i> verifyPrescription</p> <p><i>Request Parameters:</i> prescriptionID</p> <p><i>Response parameters:</i> verificationStatus, errorMessage</p> <p><i>Function:</i> Verifies the legitimacy of a prescription and, if successful, returns the status of verification.</p>
	<p><i>Operation:</i> invalidatePrescription</p> <p><i>Request Parameters:</i> prescriptionID</p> <p><i>Response parameters:</i> invalidationStatus, errorMessage</p> <p><i>Function:</i> Invalidates a prescription and, if successful, returns the status of invalidation.</p>

Interface Name	IInventoryManagement
Provider	Inventory Management System
	<i>Operation:</i> updateStock
	<i>Request Parameters:</i> medicationID, adjustmentQuantity <i>Response parameters:</i> updateStatus, errorMessage <i>Function:</i> Updates the stock quantity of a specific medication and, if successful, returns the status of the update.
	<i>Operation:</i> checkInventory
	<i>Request Parameters:</i> medicationID <i>Response parameters:</i> inventoryLevel, errorMessage <i>Function:</i> Checks the inventory level of a specific medication and returns the current stock level.

Interface Name	IMedicationInformation
Provider	Medication Information System
	<i>Operation:</i> retrieveMedicationInfo
	<i>Request Parameters:</i> medicationID <i>Response parameters:</i> medicationInfo, errorMessage <i>Function:</i> Retrieves information for a specific medication and, if successful, returns the medication information.
	<i>Operation:</i> updateMedicationInfo
	<i>Request Parameters:</i> medicationID, newInformation <i>Response parameters:</i> updateConfirmation, errorMessage <i>Function:</i> Updates information in the medication database and, if successful, returns a confirmation of the update.

Task 3: Software Testing

1. Unit Test Plan for InventoryManagement Interface:



Test Cases for updateInventory Method:

Test Case	Method and Parameters	Expected Output
Successful Inventory Update	updateInventory("MED123", 50)	true
Update with Invalid Medication ID	updateInventory("INVALID", 50)	false
Update with Negative Quantity	updateInventory("MED123", -10)	false

Test Cases for checkExpirationDates Method:

Test Case	Method and Parameters	Expected Output
Check for Expired Medications	checkExpirationDates()	Method call with no return value (side effects expected)

Test Cases for orderNewStock Method:

Test Case	Method and Parameters	Expected Output
Successful Stock Order	orderNewStock("MED123", 100)	true
Order with Invalid Medication ID	orderNewStock("INVALID", 100)	false
Order with Negative Quantity	orderNewStock("MED123", -50)	false

2. System Test Plan for Use Case: Inventory ManagementScenario: Pharmacist

Secure Login and Access

Scenario: Managing Medication Inventory

Test Progress for Updating Inventory:

Actor (Pharmacy Technician)	System (Pharmacy Management Software)
Logs into the system and navigates to the inventory section.	Authenticates the user and provides access to inventory management.
Enters medication ID and updates the quantity on hand.	Validates the medication ID and updates the inventory count.
Awaits confirmation of the inventory update.	Confirms the update and logs the inventory change.

Test Data:

- Input: medicationID = "MED123", quantity = 50
- Expected Output: Confirmation of inventory update

Test Progress for Reordering Stock:

Actor (Pharmacy Technician)	System (Pharmacy Management Software)
Initiates a check for medication expiration dates.	Scans the inventory for expired or soon-to-expire medications.
Reviews the report of medications nearing expiration.	Provides a report with expiration details and highlights any actions needed.

Test Data:

- Input: Expiration date check initiated by the user
- Expected Output: Report of medications with upcoming expiration dates

Test Progress for Reordering Stock:

Actor (Pharmacy Technician)	System (Pharmacy Management
-----------------------------	-----------------------------

	Software)
Identifies a medication with low stock and decides to order more.	Retrieves current stock levels for verification.
Enters medication ID and the quantity to reorder.	Processes the reorder request and places an order with the supplier.
Awaits confirmation of the stock reorder.	Confirms the reorder and updates the pending stock levels.

Test Data:

- Input: medicationID = "MED123", quantity = 100
- Expected Output: Confirmation of new stock order