RAJALAKSHMI ENGINEERING COLLEGE

RAJALAKSHMI NAGAR, THANDALAM – 602 105

RAJALAKSHMI
ENGINEERING COLLEGE

CB23332
SOFTWARE ENGINEERING LAB

Laboratory Record Note Book

Name : . . . . . .

Year / Branch / Section : . . . . . .

Register No. : . . . . . .

Semester : . . . . . .

Academic Year : . . . . . .

RAJALAKSHMI ENGINEERING COLLEGE (AUTONOMOUS)

RAJALAKSHMI NAGAR, THANDALAM – 602-105

BONAFIDE CERTIFICATE

NAME:_____**REGISTER NO.:** _____

ACADEMIC YEAR: 2024-25 **SEMESTER:** III **BRANCH:**_____B.E/B.Tech

This Certification is the bonafide record of work done by the above student in the

CB23332-SOFTWARE ENGINEERING - Laboratory during the year 2024 – 2025.

Signature of Faculty -in – Charge

Submitted for the Practical Examination held on _____

Internal Examiner External Examiner

INDEX

S. No.	Name of the Experiment	Expt. Date	Faculty Sign
1.	Preparing Problem Statement		
2.	Software Requirement Specification (SRS)		
3.	Entity-Relational Diagram		
4.	Data Flow Diagram		
5.	Use Case Diagram		
6.	Activity Diagram		
7.	State Chart Diagram		
8.	Sequence Diagram		
9.	Collaboration Diagramt		
10.	Class Diagram		

EX NO:1	WRITE THE COMPLETE PROBLEM STATEMENT
DATE:	

AIM:

To prepare PROBLEM STATEMENT for FAKE MEDICINE MANAGEMENT SYSTEM.

ALGORITHM:

1. **Identify the Core Problem**: The problem statement should clearly identify the issue of fake medicine in the healthcare system and the impact it has on public health.
2. **Keep it High-Level**: Describe the problem without specifying a solution. This helps to keep the discussion open to different possible solutions.
3. **Audience Clarity**: Write the problem statement in non-technical language to ensure it's accessible to both technical and non-technical stakeholders.
4. **Specify the Impact and Urgency**: Indicate how the issue of fake medicines affects stakeholders and why it needs attention.
5. **Provide a Baseline**: Outline any existing measures or regulations in place to manage fake medicines, along with their limitations.
6. **Define Success Metrics**: Specify how success will be measured, focusing on the reduction of fake medicines in the supply chain.

INPUT:

- The input to the requirements engineering process is the problem statement provided by stakeholders, including healthcare authorities, pharmacies, and patients.
- It includes an overview of the current state of the medicine supply chain, existing detection methods, and desired improvements.
- The requirements elicitation phase begins with gathering details on the challenges stakeholders face with fake medicines and the necessary data sources for system implementation.
- This process involves interactions with healthcare professionals, regulators, and pharmaceutical suppliers.

Problem:

The prevalence of fake medicines has become a severe public health concern, affecting patient safety and trust in the healthcare system. A recent survey across various healthcare providers revealed that around 15% of medicines in the supply chain are counterfeit, leading to compromised patient outcomes and increased healthcare costs. This issue is exacerbated by limited tracking and verification methods, which enable counterfeit medicines to circulate in the market undetected. To address this, a reliable and transparent medicine verification system is essential.

Background:

The issue of counterfeit medicine is growing rapidly, with patients unknowingly purchasing ineffective or harmful drugs. These fake medicines are often difficult to trace due to gaps in the supply chain and a lack of robust tracking systems. Current approaches, such as manual inspections and minimal barcode scanning, are insufficient for addressing the scale of the problem. This project aims to design a system that ensures the authenticity of medicines by leveraging modern tracking and verification technologies.

Relevance:

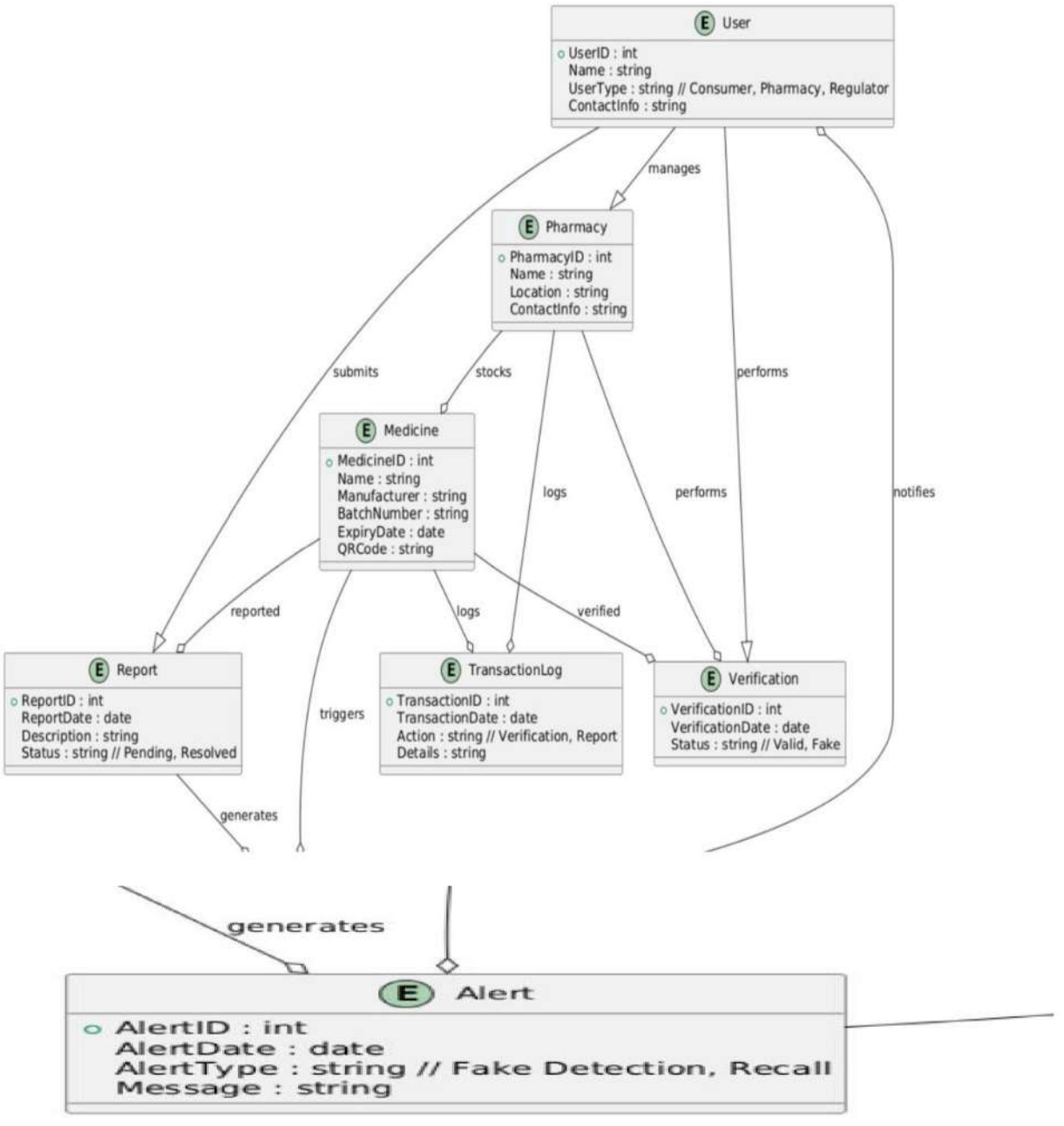
Counterfeit medicines undermine the healthcare system's effectiveness, as patients receiving these drugs experience delayed treatment or worsening conditions. Authenticating medicines at each stage of the supply chain is vital to restore trust in healthcare providers and ensure patient safety. Addressing this issue not only improves patient outcomes but also reduces the financial and reputational costs incurred by healthcare systems dealing with the consequences of fake medicines.

Objectives:

The primary objective of this project is to implement a digital medicine verification and tracking system that significantly reduces the percentage of fake medicines in circulation. Specific objectives include:

- Conducting a comprehensive analysis of the supply chain to identify vulnerability points where fake medicines enter.
- Developing a robust system using technologies like QR codes, RFID, or blockchain to track and authenticate medicines from manufacturers to end consumers.
- Enforcing secure checkpoints at multiple stages of the supply chain, ensuring each medicine's origin and quality are verified.
- Implementing a reporting mechanism for pharmacies and patients to easily verify the authenticity of medicines.
- Training healthcare providers, pharmacists, and supply chain personnel on using the new verification system effectively.
- Regularly monitoring and updating the system to adapt to evolving counterfeit medicine tactics.
- Evaluating system effectiveness through key performance indicators (KPIs) such as the percentage reduction of fake medicines in circulation and response time for issue resolution.

Result:


User
UserID : int
Name : string
UserType : string // Consumer, Pharmacy, Regulator
ContactInfo : string
manages
Pharmacy
PharmacyID : int
Name : string
Location : string
ContactInfo : string
submits
stocks
performs
Medicine
MedicineID : int
Name : string
Manufacturer : string
BatchNumber : string
ExpiryDate : date
QRCode : string
logs
performs
notifies
reported
logs
verified
Report
ReportID : int
ReportDate : date
Description : string
Status : string // Pending, Resolved
triggers
TransactionLog
TransactionID : int
TransactionDate : date
Action : string // Verification, Report
Details : string
Verification
VerificationID : int
VerificationDate : date
Status : string // Valid, Fake
generates
generates
Alert
AlertID : int
AlertDate : date
AlertType : string // Fake Detection, Recall
Message : string

EX NO:2	WRITE THE SOFTWARE REQUIREMENT SPECIFICATION DOCUMENT
DATE:	

AIM:

To do requirement analysis and develop Software Requirement Specification Sheet(SRS) for Fake Medicine Management System

ALGORITHM:

An SRS for the Fake Medicine Management System should address the following areas:

a) **Functionality**: What tasks will the system perform? b) **External Interfaces**: How does the system interact with users, databases, other software, and hardware? c) **Performance**: Expected speed, response time, and availability for the system. d) **Attributes**: Portability, security, and maintainability of the system. e) **Design Constraints**: Standards, technology policies, operating environment, and resource limitations.

---

1. INTRODUCTION

1.1 PURPOSE

The purpose of this document is to outline the requirements for developing an online Fake Medicine Management System. The system aims to detect, report, and track counterfeit medicines, providing a safe and reliable platform for healthcare providers, pharmacies, and patients.

1.2 DOCUMENT CONVENTIONS

This document uses the following conventions:

- **FMMS**: Fake Medicine Management System
- **UI**: User Interface
- **DB**: Database
- **API**: Application Programming Interface

1.3 INTENDED AUDIENCE AND READING SUGGESTIONS

This project is designed for healthcare administrators, regulatory authorities, pharmacists, and end-users concerned with counterfeit medicines. It provides functionality for monitoring, verifying, and reporting fake medicine in the supply chain.

1.4 PROJECT SCOPE

The FMMS will help reduce the spread of counterfeit medicines by offering an efficient and user-friendly platform to verify medicine authenticity. The system will integrate with pharmaceutical databases and support real-time tracking and reporting of suspicious medicines.

1.5 REFERENCES

- Fundamentals of database systems by Ramez Elmasri and Shamkant B. Navathe
- WHO guidelines on counterfeit drug detection and tracking

OVERALL DESCRIPTION

2.1 PRODUCT PERSPECTIVE

The FMMS will utilize a distributed database to store information about medicines, including their origin, manufacturing details, and distribution. The main components include:

- **Medicine Database**: Contains detailed information on registered medicines, such as manufacturer details, batch numbers, and expiration dates.
- **Pharmacy Records**: Tracks pharmacies and retailers that carry the medicines, including inventory details.
- **Transaction Logs**: Logs all transactions related to medicine verification, including reports of fake medicines.

2.2 PRODUCT FEATURES

Key features of the FMMS include:

- **Medicine Verification**: Users can scan a QR code or enter a unique ID to verify the authenticity of a medicine.
- **Suspicious Medicine Reporting**: Allows pharmacies and consumers to report fake or suspicious medicines.
- **Medicine Tracking**: Tracks medicine from the point of manufacture to the end consumer, ensuring traceability.

Alert Notifications: Sends alerts when fake medicine is detected in the supply chain.

2.3 USER CLASSES AND CHARACTERISTICS

The system supports different types of users with specific access rights:

- **Consumers**: Can verify medicines and report suspicious cases.
- **Pharmacies**: Manage inventory and verify incoming shipments.
- **Regulators**: Monitor and respond to reports, enforce compliance, and initiate investigations.

2.4 OPERATING ENVIRONMENT

- **Operating System**: Windows, Linux
- **Database**: SQL+ or distributed databases
- **Platform**: Web-based, compatible with modern browsers and mobile devices

2.5 DESIGN AND IMPLEMENTATION CONSTRAINTS

1. Data storage must comply with industry standards for data protection and privacy.
2. The system should support data fragmentation and geographic distribution to ensure efficient data access and reliability.

2.6 ASSUMPTION AND DEPENDENCIES

The FMMS assumes a robust internet connection for real-time verification and updates. It depends on accurate data from pharmaceutical manufacturers and regulatory authorities.

SYSTEM FEATURES

3.1 DESCRIPTION AND PRIORITY

This high-priority system is critical in preventing the spread of counterfeit medicines, which is essential for public safety.

3.2 STIMULUS/RESPONSE SEQUENCES

- **Medicine Verification**: User inputs or scans a medicine ID, and the system displays authenticity details.
- **Reporting**: User reports a suspicious medicine; the system logs it and notifies the regulator.

3.3 FUNCTIONAL REQUIREMENTS

1. **Database Integration**: Maintain a distributed database to store and verify details of medicines and transactions.
2. **Client/Server System**: The server stores data, while the client handles medicine verification and reporting.

---

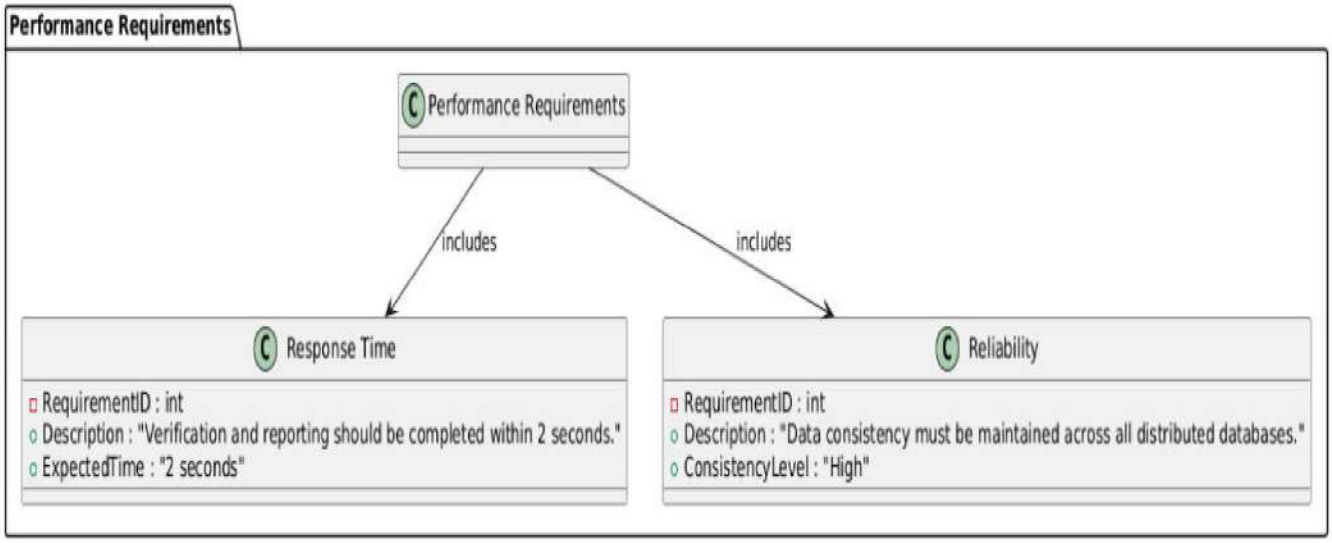
4. EXTERNAL INTERFACE REQUIREMENTS

4.1 USER INTERFACES

- **Front-end Software**: Web interface built with HTML, CSS, and JavaScript.
- **Back-end Software**: API layer built using Python or Node.js to interact with the database.

4.2 HARDWARE INTERFACES

Device Requirements: Compatible with computers, tablets, and smartphones with internet access



4.3 SOFTWARE INTERFACES

- **Database**: SQL+ database for storage and retrieval of data.
- **Programming Language**: Python, JavaScript, or PHP for implementation.

4.4 COMMUNICATION INTERFACES

The FMMS supports standard web communication protocols, including HTTPS for secure data transmission.

---

5. NONFUNCTIONAL REQUIREMENTS

5.1 PERFORMANCE REQUIREMENTS

- **Response Time**: Verification and reporting should be completed within 2 seconds.
- **Reliability**: Data consistency must be maintained across all distributed databases.

5.2 SAFETY REQUIREMENTS

Backups and recovery methods should be implemented to prevent data loss due to technical failures.

5.3 SECURITY REQUIREMENTS

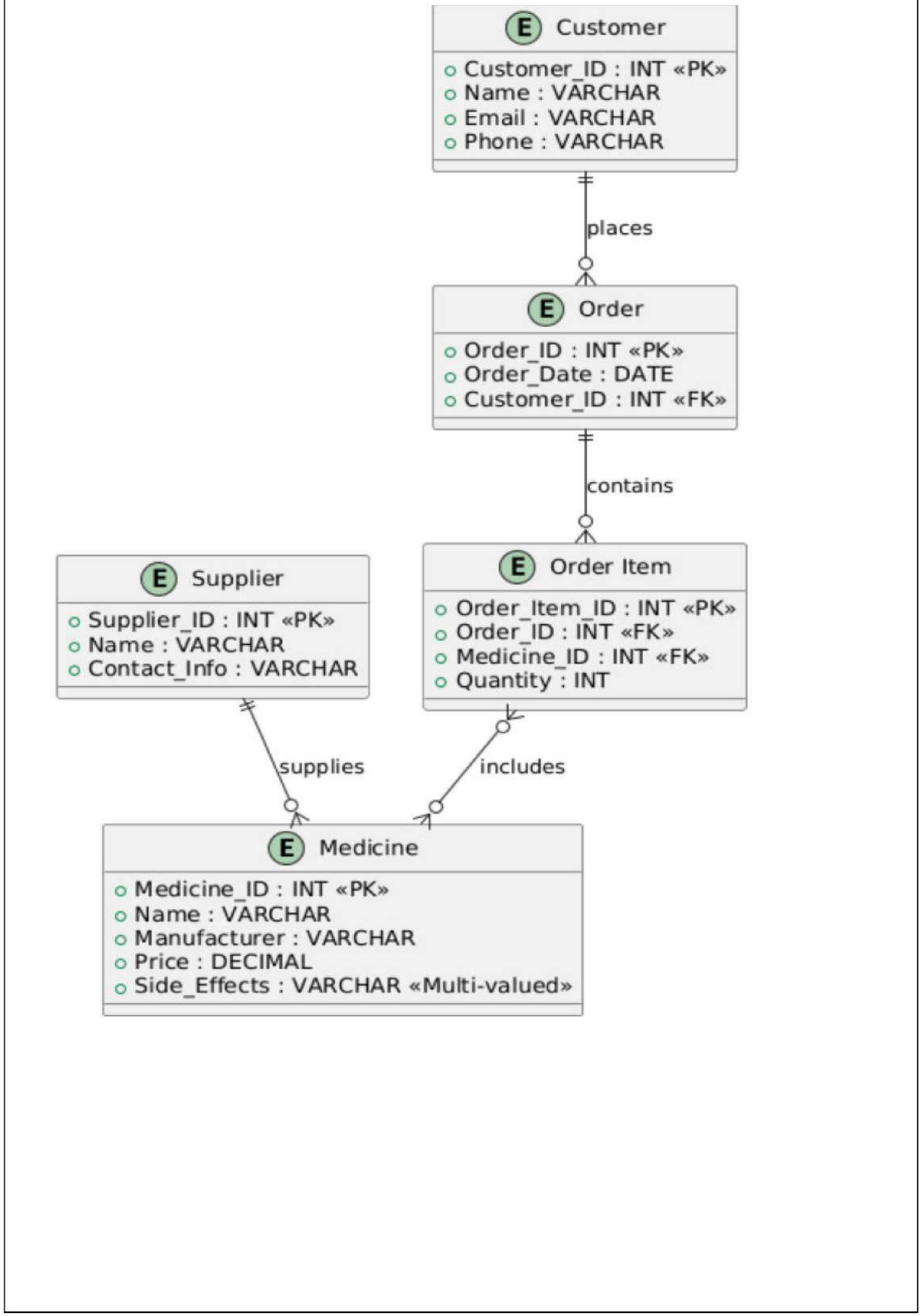
Data access must be controlled with user authentication and authorization, ensuring that only verified personnel can view or edit data.

5.4 SOFTWARE QUALITY ATTRIBUTES

- **Availability**: The system should be available 24/7 to accommodate different time zones.
- **Maintainability**: Regular updates and maintenance are required to ensure system accuracy.

- **Usability**: The system should be intuitive for users with basic computer knowledge.

Result:


Customer
Customer_ID : INT «PK»
Name : VARCHAR
Email : VARCHAR
Phone : VARCHAR
places
Order
Order_ID : INT «PK»
Order_Date : DATE
Customer_ID : INT «FK»
contains
Supplier
Supplier_ID : INT «PK»
Name : VARCHAR
Contact_Info : VARCHAR
Order Item
Order_Item_ID : INT «PK»
Order_ID : INT «FK»
Medicine_ID : INT «FK»
Quantity : INT
supplies
includes
Medicine
Medicine_ID : INT «PK»
Name : VARCHAR
Manufacturer : VARCHAR
Price : DECIMAL
Side_Effects : VARCHAR «Multi-valued»

EX NO:3	DRAW THE ENTITY RELATIONSHIP DIAGRAM
DATE:	

AIM:

To Draw the Entity Relationship Diagram For Fake Medicine Management System

ALGORITHM:

Step 1: Mapping of Regular Entity Types

Step 2: Mapping of Weak Entity Types

Step 3: Mapping of Binary 1:1 Relation Types

Step 4: Mapping of Binary 1:N Relationship Types.

Step 5: Mapping of Binary M:N Relationship Types.

Step 6: Mapping of Multivalued attributes.

INPUT:

Entities

Entity Relationship Matrix

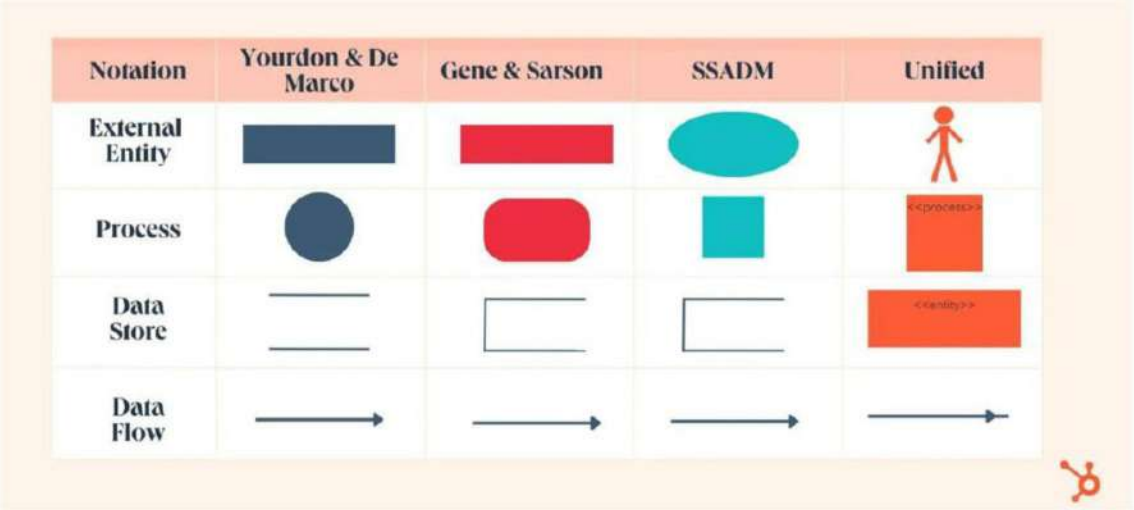
Primary Keys

Attributes

Mapping of Attributes with Entities

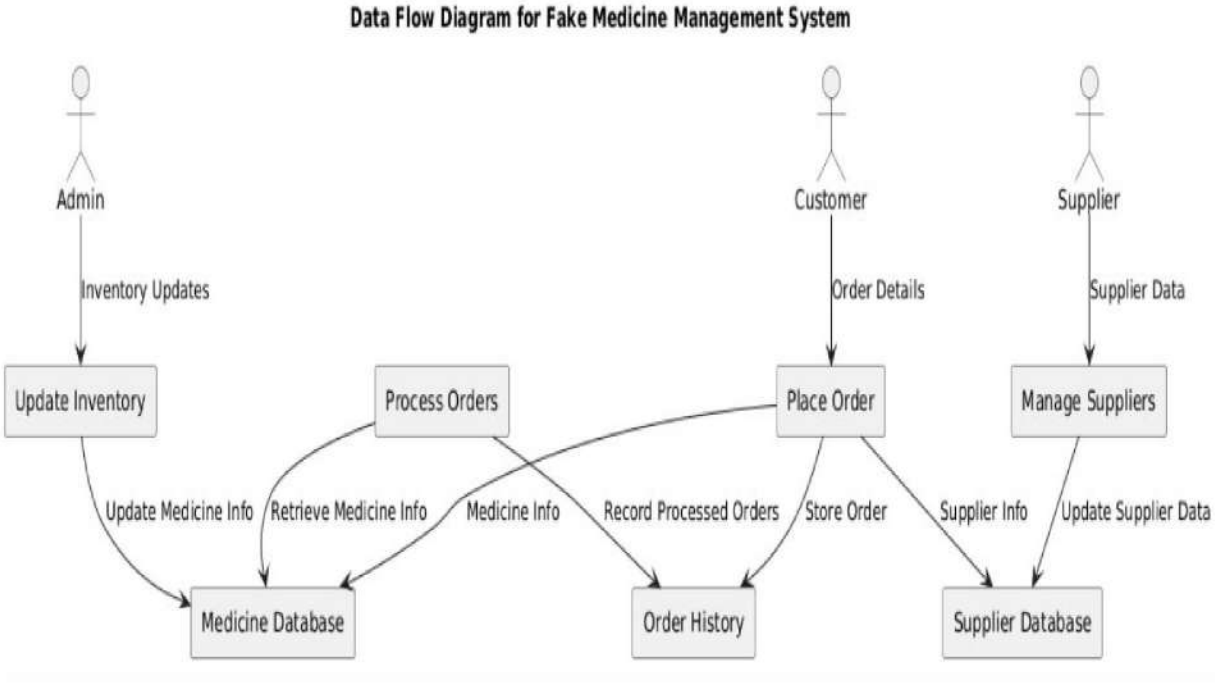
Result:

INPUT SYMBOLS:-



Notation	Yourdon & De Marco	Gene & Sarson	SSADM	Unified
External Entity				
Process				
Data Store				
Data Flow				

SAMPLE OUTPUT :



EX NO:4	DRAW THE DATA FLOW DIAGRAMS AT LEVEL 0 AND LEVEL 1
DATE:	

AIM:

To Draw the Data Flow Diagram For Fake Medicine Management System

ALGORITHM:

1. Open the Visual Paradigm to draw DFD (Ex.Lucidchart)
2. Select a data flow diagram template
3. Name the data flow diagram
4. Add an external entity that starts the process
5. Add a Process to the DFD
6. Add a data store to the diagram
7. Continue to add items to the DFD
8. Add data flow to the DFD
9. Name the data flow
10. Customize the DFD with colours and fonts
11. Add a title and share your data flow diagram

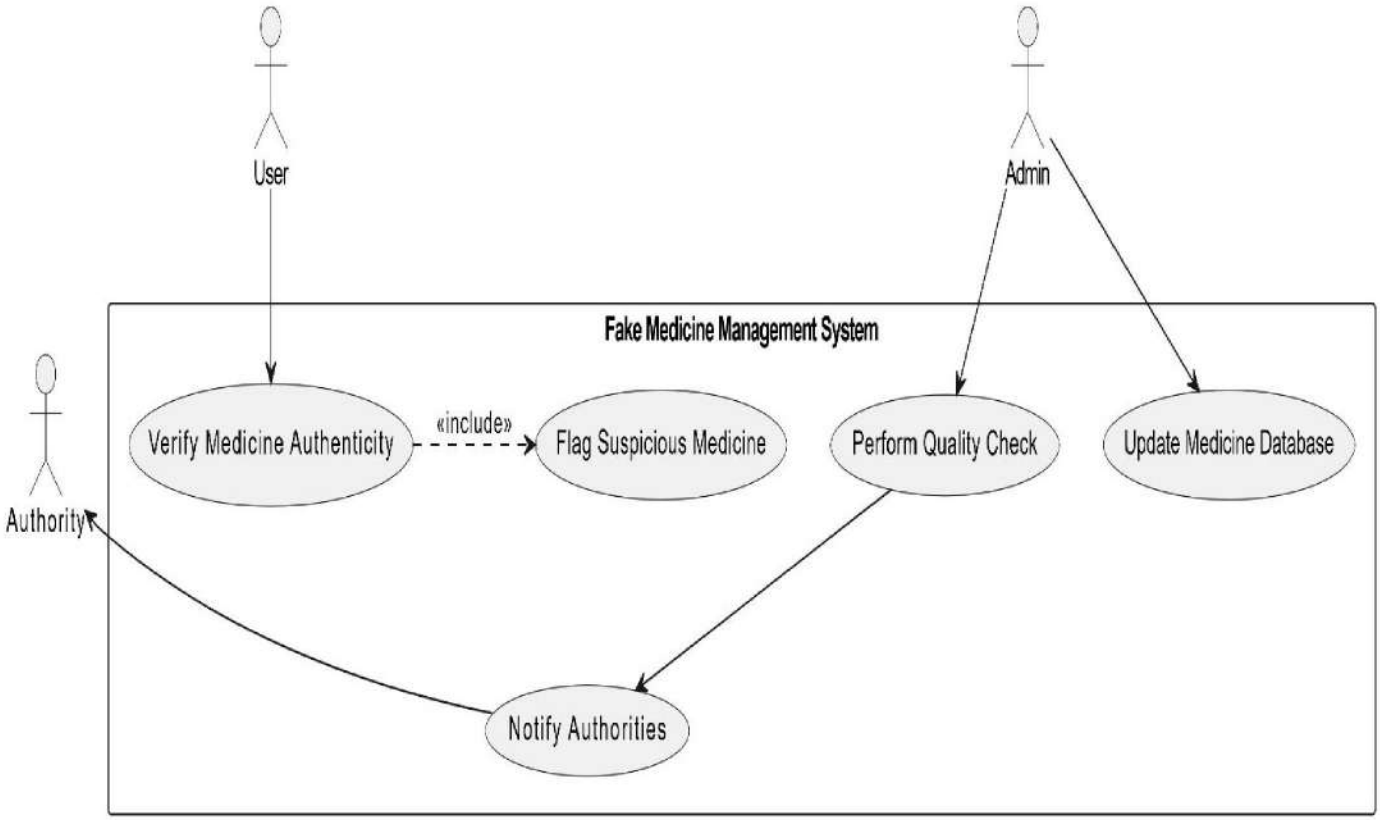
INPUT:

Processes

Datastores

External Entities

Result:


User
Admin
Fake Medicine Management System
Authority
Verify Medicine Authenticity
«include»
Flag Suspicious Medicine
Perform Quality Check
Update Medicine Database
Notify Authorities

EX NO:5	DRAW USE CASE DIAGRAM
DATE:	

AIM:

To Draw the Use Case Diagram for Fake Medicine Management System

ALGORITHM:

Step 1: Identify Actors

Step 2: Identify Use Cases

Step 3: Connect Actors and Use Cases

Step 4: Add System Boundary

Step 5: Define Relationships

Step 6: Review and Refine

Step 7: Validate

INPUTS:

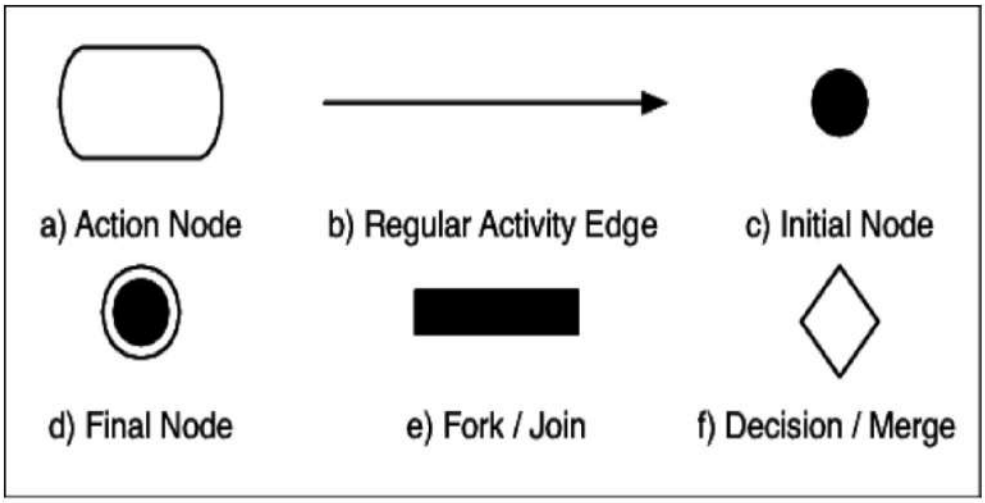
Actors

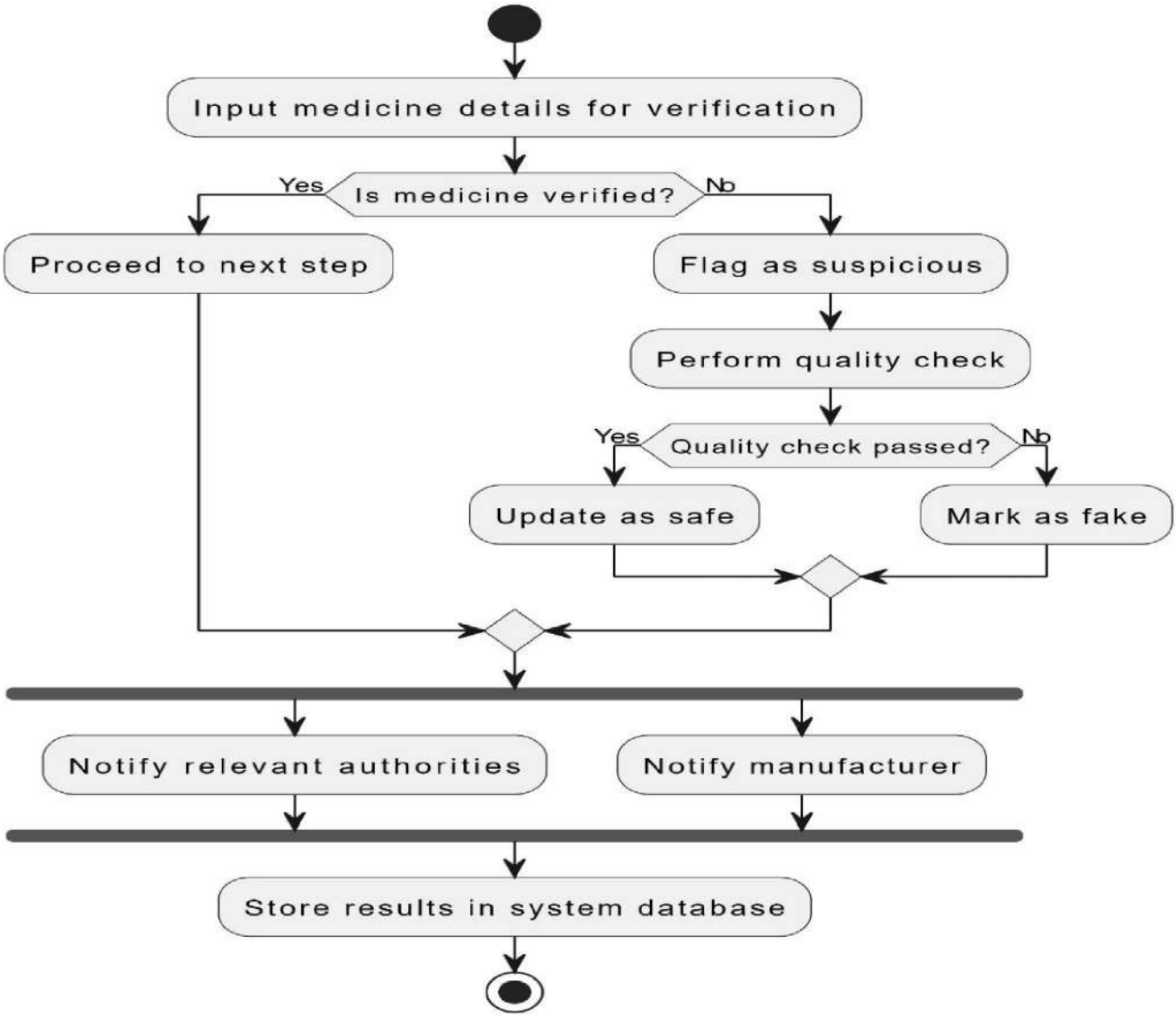
Use Cases

Relations

Result:

SYMBOL:


a) Action Node
b) Regular Activity Edge
c) Initial Node
d) Final Node
e) Fork / Join
f) Decision / Merge


Input medicine details for verification
Is medicine verified?
Yes
No
Proceed to next step
Flag as suspicious
Perform quality check
Quality check passed?
Yes
No
Update as safe
Mark as fake
Notify relevant authorities
Notify manufacturer
Store results in system database

EX NO:6	DRAW ACTIVITY DIAGRAM OF ALL USE CASES.
DATE:	

AIM:

To Draw the activity Diagram for Fake Medicine Management System

ALGORITHM:

Step 1: Identify the Initial State and Final States

Step 2: Identify the Intermediate Activities Needed

Step 3: Identify the Conditions or Constraints

Step 4: Draw the Diagram with Appropriate Notations

INPUTS:

Activities

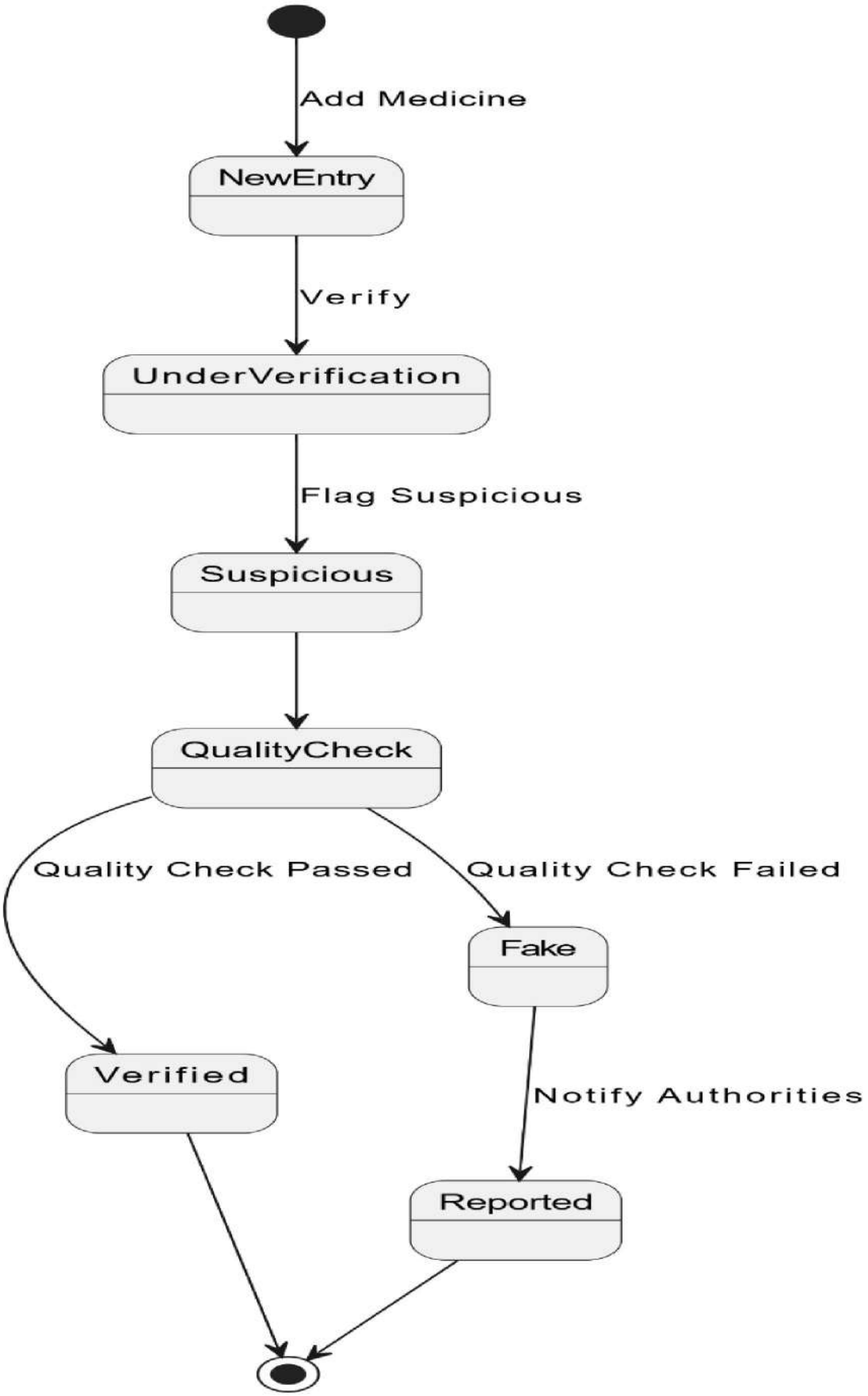
Decision Points

Guards

Parallel Activities

Conditions

Result:


Add Medicine
NewEntry
Verify
UnderVerification
Flag Suspicious
Suspicious
QualityCheck
Quality Check Passed
Quality Check Failed
Fake
Verified
Notify Authorities
Reported

EX NO:7	DRAW STATE CHART DIAGRAM OF ALL USE CASES.
DATE:	

AIM:

To Draw the State Chart Diagram for Fake Medicine Management System

ALGORITHM:

STEP-1: Identify the important objects to be analysed.

STEP-2: Identify the states.

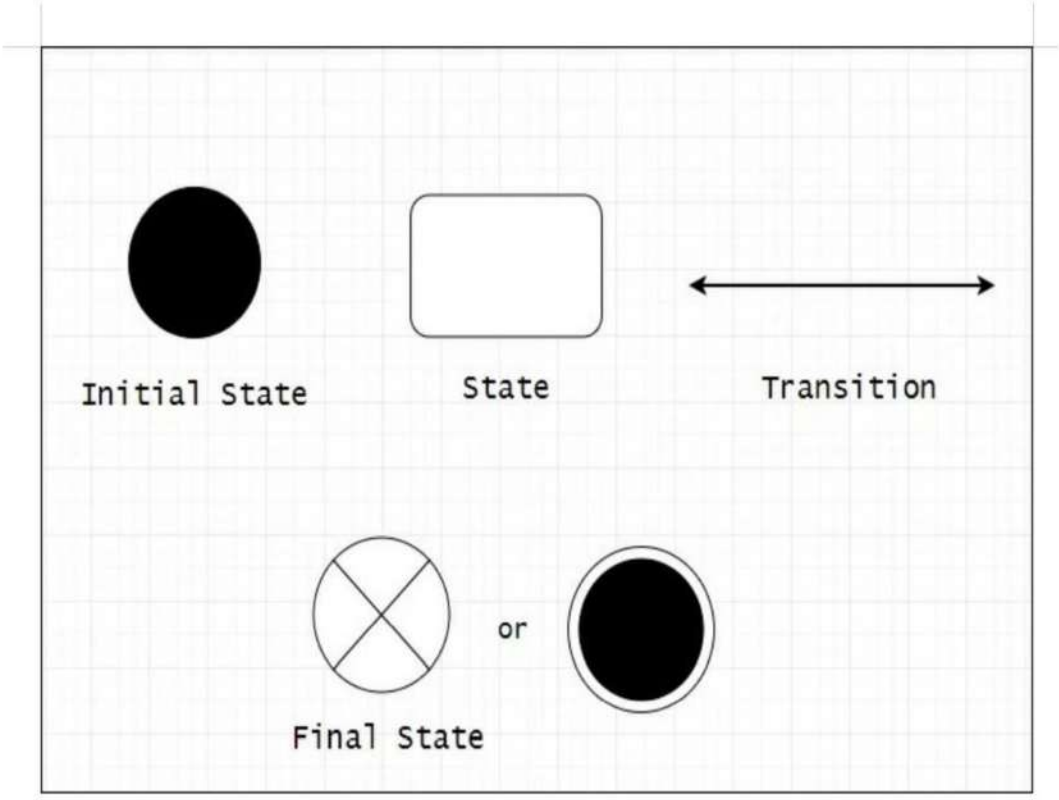
STEP-3: Identify the events.

INPUTS:

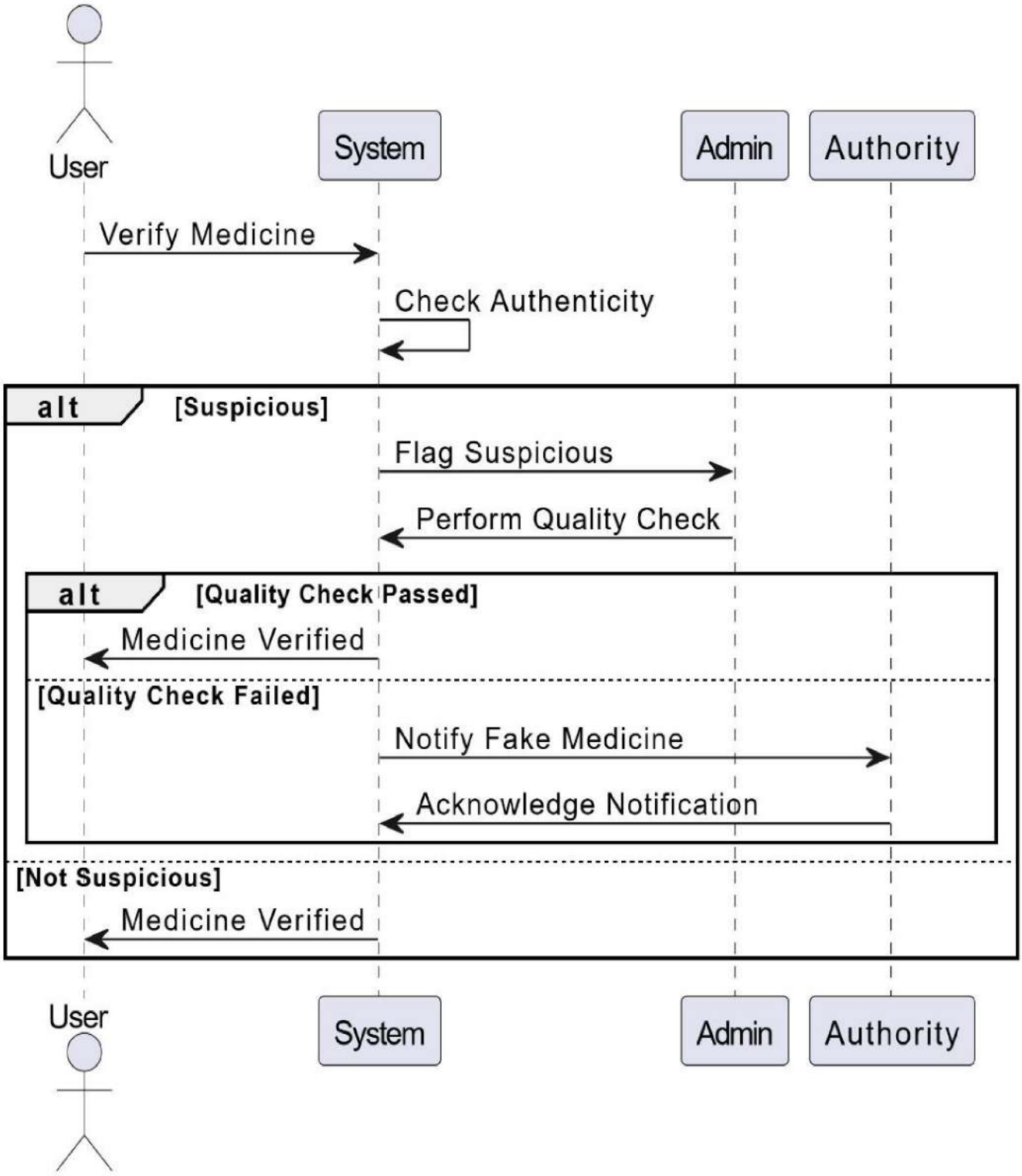
Objects

States

Events



RESULT:


User
System
Admin
Authority
Verify Medicine
Check Authenticity
alt
[Suspicious]
Flag Suspicious
Perform Quality Check
alt
[Quality Check Passed]
Medicine Verified
[Quality Check Failed]
Notify Fake Medicine
Acknowledge Notification
[Not Suspicious]
Medicine Verified
User
System
Admin
Authority

EX NO:8	DRAW SEQUENCE DIAGRAM OF ALL USE CASES.
DATE:	

AIM: To Draw the Sequence Diagram for Fake Medicine Management System

ALGORITHM:

1. Identify the Scenario
2. List the Participants
3. Define Lifelines
4. Arrange Lifelines
5. Add Activation Bars
6. Draw Messages
7. Include Return Messages
8. Indicate Timing and Order
9. Include Conditions and Loops
10. Consider Parallel Execution
11. Review and Refine
12. Add Annotations and Comments
13. Document Assumptions and Constraints
14. Use a Tool to create a neat sequence diagram

INPUTS:

Objects taking part in the interaction.

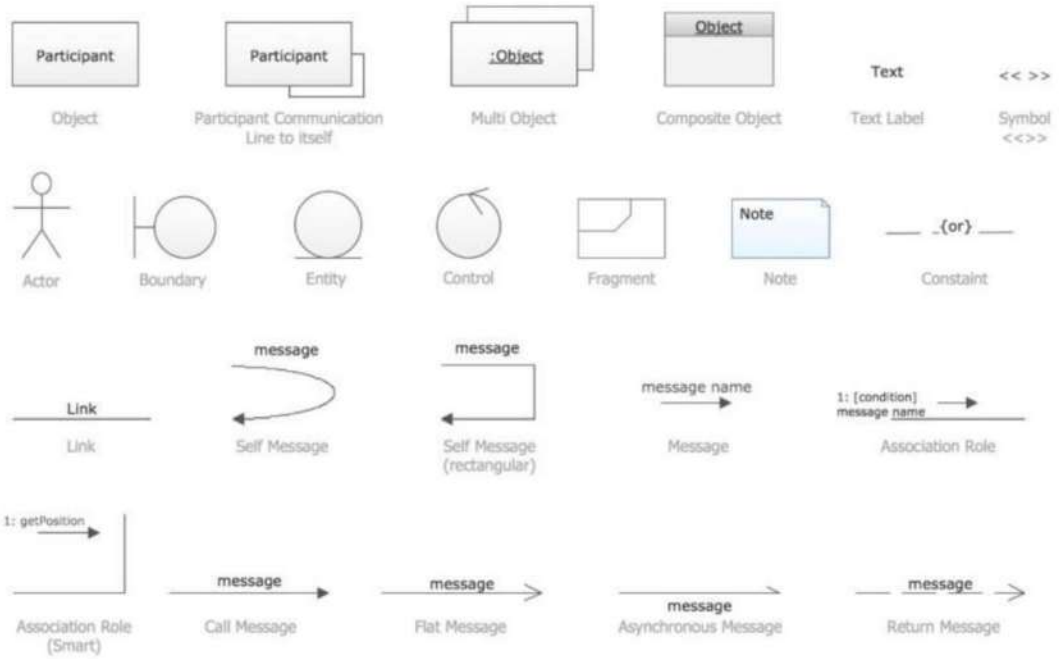
Message flows among the objects.

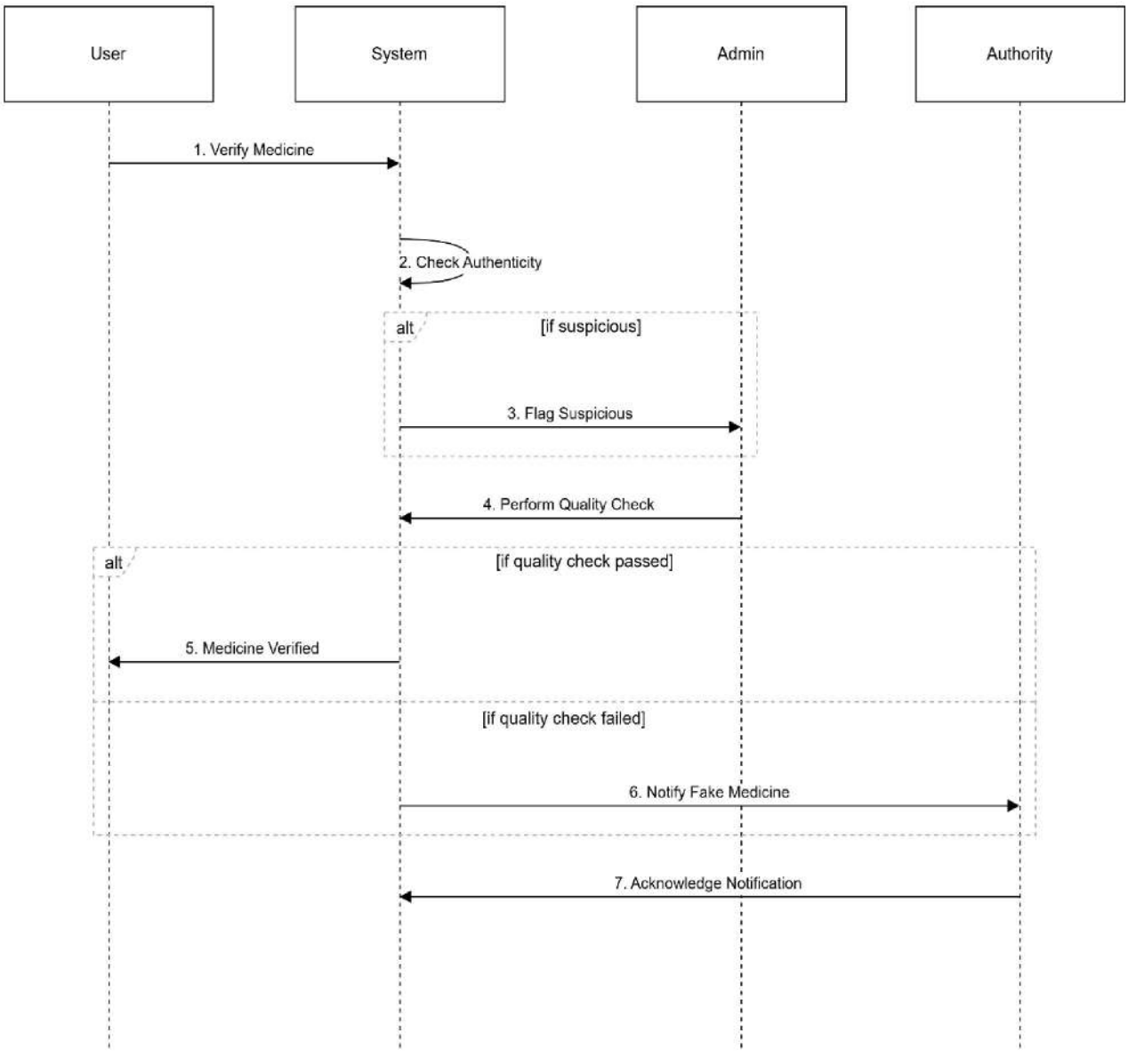
The sequence in which the messages are flowing.

Object organization.

Result:

SYMBOL:


Participant
Object
Participant
Participant Communication Line to itself
:Object
Multi Object
Object
Composite Object
Text
Text Label
<< >>
Symbol <<>>
Actor
Boundary
Entity
Control
Fragment
Note
Note
{or}
Constaint
Link
Link
message
Self Message
message
Self Message (rectangular)
message name
Message
1: [condition] message name
Association Role
1: getPosition
Association Role (Smart)
message
Call Message
message
Flat Message
message
Asynchronous Message
message
Return Message


User
System
Admin
Authority
1. Verify Medicine
2. Check Authenticity
alt
[if suspicious]
3. Flag Suspicious
4. Perform Quality Check
alt
[if quality check passed]
5. Medicine Verified
[if quality check failed]
6. Notify Fake Medicine
7. Acknowledge Notification

EX NO:9	DRAW COLLABORATION DIAGRAM OF ALL USE CASES
DATE:	

AIM:

To Draw the Collaboration Diagram for Fake medicine Management System

ALGORITHM:

Step 1: Identify Objects/Participants

Step 2: Define Interactions

Step 3: Add Messages

Step 4: Consider Relationships

Step 5: Document the collaboration diagram along with any relevant explanations or annotations.

INPUTS:

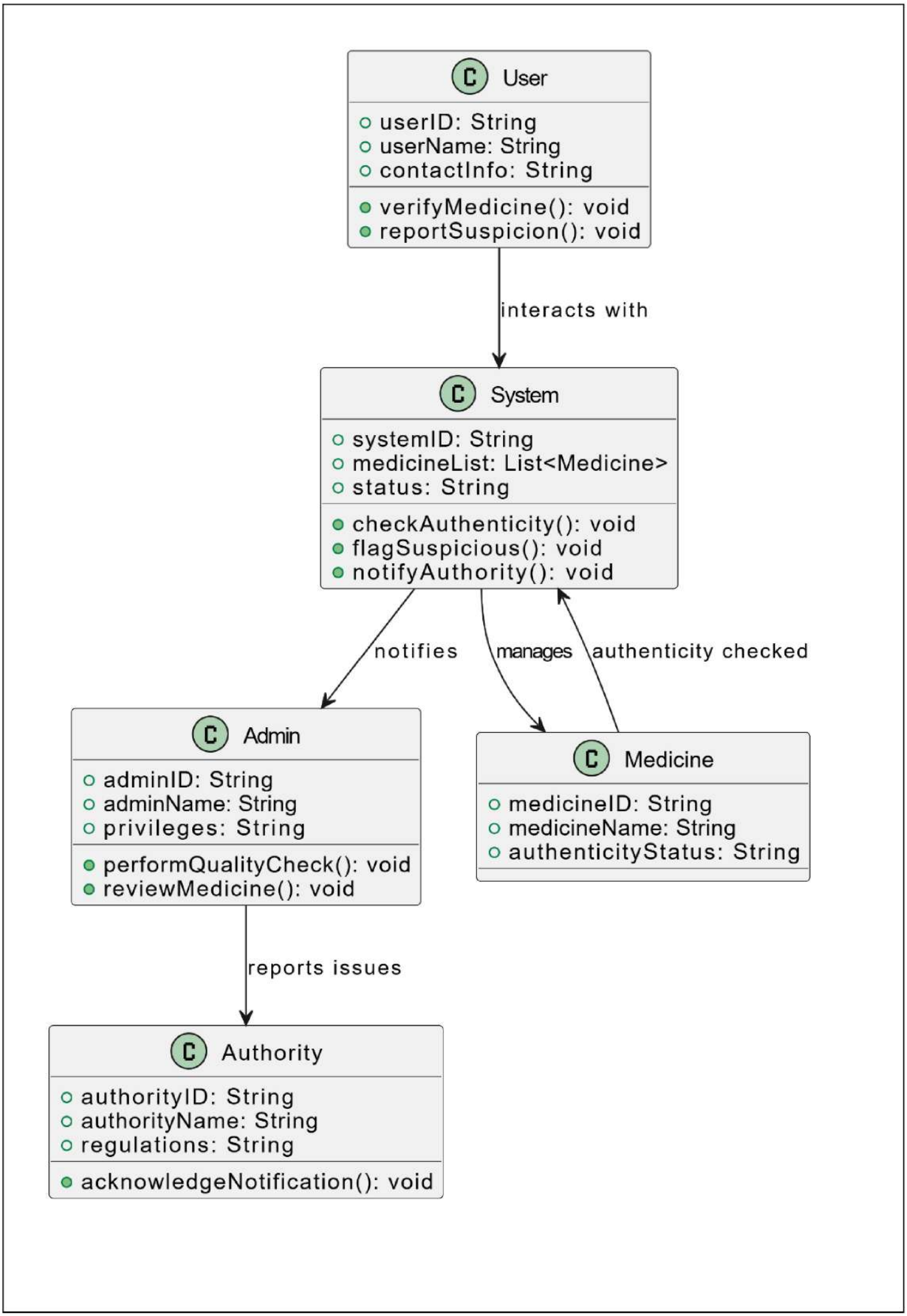
Objects taking part in the interaction.

Message flows among the objects.

The sequence in which the messages are flowing.

Object organization.

Result:


User
userID: String
userName: String
contactInfo: String
verifyMedicine(): void
reportSuspicion(): void
interacts with
System
systemID: String
medicineList: List<Medicine>
status: String
checkAuthenticity(): void
flagSuspicious(): void
notifyAuthority(): void
notifies
manages
authenticity checked
Admin
adminID: String
adminName: String
privileges: String
performQualityCheck(): void
reviewMedicine(): void
Medicine
medicineID: String
medicineName: String
authenticityStatus: String
reports issues
Authority
authorityID: String
authorityName: String
regulations: String
acknowledgeNotification(): void

EX NO:10	ASSIGN OBJECTS IN SEQUENCE DIAGRAM TO CLASSES AND MAKE CLASS DIAGRAM.
DATE:	

AIM:

To Draw the Class Diagram for Fake Medicine Management System

ALGORITHM:

1. Identify Classes
2. List Attributes and Methods
3. Identify Relationships
4. Create Class Boxes
5. Add Attributes and Methods
6. Draw Relationships
7. Label Relationships
8. Review and Refine
9. Use Tools for Digital Drawing

INPUTS:

1. Class Name
2. Attributes
3. Methods
4. Visibility Notation

RESULT:

CODE FOR MINI PROJECT

IMPLEMENTATION USING PYTHON :

CODE:

```
import hashlib
import random

class FakeMedicineManagementSystem:
    def __init__(self):
        self.medicine_database = {}
        self.reported_fake_medicines = []

    def generate_unique_id(self, medicine_name):
        random_salt = random.randint(1000, 9999)
        return hashlib.sha256(f"{medicine_name}{random_salt}".encode()).hexdigest()[:8]

    def add_medicine(self):
        name = input("Enter medicine name: ")
        manufacturer = input("Enter manufacturer name: ")
        expiry_date = input("Enter expiry date (YYYY-MM-DD): ")
        unique_id = self.generate_unique_id(name)
        self.medicine_database[unique_id] = {
            "name": name,
            "manufacturer": manufacturer,
            "expiry_date": expiry_date,
        }
        print(f"Medicine added with ID: {unique_id}")

    def verify_medicine(self):
        unique_id = input("Enter medicine ID to verify: ")
        if unique_id in self.medicine_database:
            medicine = self.medicine_database[unique_id]
            print(f"Medicine Verified: {medicine['name']} by {medicine['manufacturer']}")
        else:
```

```
            print("Fake medicine detected or invalid ID.")

    def report_fake_medicine(self):
        unique_id = input("Enter medicine ID to report: ")
        if unique_id not in self.medicine_database:
            self.reported_fake_medicines.append(unique_id)
            print("Medicine reported as fake.")
        else:
            print("Medicine found in the database. No action taken.")

    def display_all_medicines(self):
        if not self.medicine_database:
            print("No medicines in the database.")
        else:
            print("Registered Medicines:")
            for unique_id, details in self.medicine_database.items():
                print(f"ID: {unique_id}, Name: {details['name']}, Manufacturer: {details['manufacturer']},
Expiry: {details['expiry_date']}")

    def display_fake_reports(self):
        if not self.reported_fake_medicines:
            print("No fake medicines reported.")
        else:
            print("Reported Fake Medicines IDs:")
            for unique_id in self.reported_fake_medicines:
                print(f"- {unique_id}")



def main():
    system = FakeMedicineManagementSystem()

    while True:
        print("\nFake Medicine Management System")
        print("1. Add Medicine")
```

```
        print("2. Verify Medicine")
        print("3. Report Fake Medicine")
        print("4. Display All Medicines")
        print("5. Display Fake Reports")
        print("6. Exit")

        choice = input("Enter your choice: ")

        if choice == "1":
            system.add_medicine()
        elif choice == "2":
            system.verify_medicine()
        elif choice == "3":
            system.report_fake_medicine()
        elif choice == "4":
            system.display_all_medicines()
        elif choice == "5":
            system.display_fake_reports()
        elif choice == "6":
            print("Exiting the system. Goodbye!")
            break
        else:
            print("Invalid choice. Please try again.")

if __name__ == "__main__":
    main()
```

Input:

```
Fake Medicine Management System
1. Add Medicine
2. Verify Medicine
3. Report Fake Medicine
4. Display All Medicines
5. Display Fake Reports
```

6. Exit

Enter your choice: 1

Enter medicine name: Paracetamol

Enter manufacturer name: ABC Pharma

Enter expiry date (YYYY-MM-DD): 2025-12-31

Output:

Medicine added with ID: a1b2c3d4

Scenario 2: Verifying a Valid Medicine

Input:

Fake Medicine Management System

1. Add Medicine

2. Verify Medicine

3. Report Fake Medicine

4. Display All Medicines

5. Display Fake Reports

6. Exit

Enter your choice: 2

Enter medicine ID to verify: a1b2c3d4

Output:

Medicine Verified: Paracetamol by ABC Pharma

Scenario 3: Verifying an Invalid Medicine

Input:

Fake Medicine Management System

1. Add Medicine

2. Verify Medicine

3. Report Fake Medicine

4. Display All Medicines

5. Display Fake Reports

6. Exit

Enter your choice: 2

Enter medicine ID to verify: invalid123

Output:

Fake medicine detected or invalid ID.

Scenario 4: Reporting a Fake Medicine

Input:

Fake Medicine Management System

1. Add Medicine

2. Verify Medicine

3. Report Fake Medicine

4. Display All Medicines

5. Display Fake Reports

6. Exit

Enter your choice: 3

Enter medicine ID to report: invalid123

Output:

Medicine reported as fake.

Scenario 5: Displaying All Medicines

Input:

markdown

Fake Medicine Management System

1. Add Medicine

2. Verify Medicine

3. Report Fake Medicine

4. Display All Medicines

5. Display Fake Reports

6. Exit

Enter your choice: 4

Output:

Registered Medicines:

ID: a1b2c3d4, Name: Paracetamol, Manufacturer: ABC Pharma, Expiry: 2025-12-31

Scenario 6: Displaying Fake Reports

Input:

markdown

Fake Medicine Management System

1. Add Medicine

2. Verify Medicine

3. Report Fake Medicine

4. Display All Medicines

5. Display Fake Reports

6. Exit

Enter your choice: 5

Output:

Reported Fake Medicines IDs:

- invalid123
