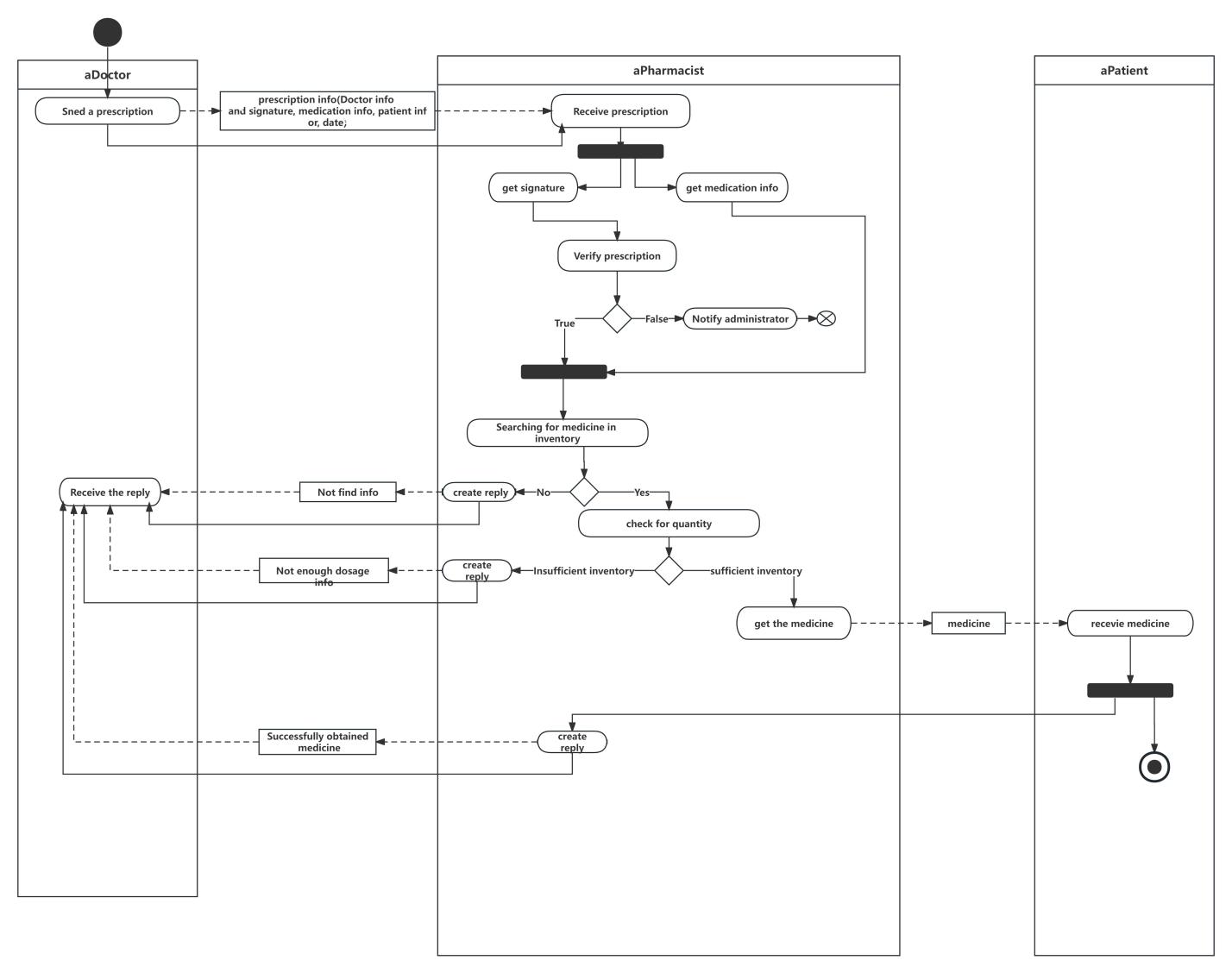


Software Engineering Coursework

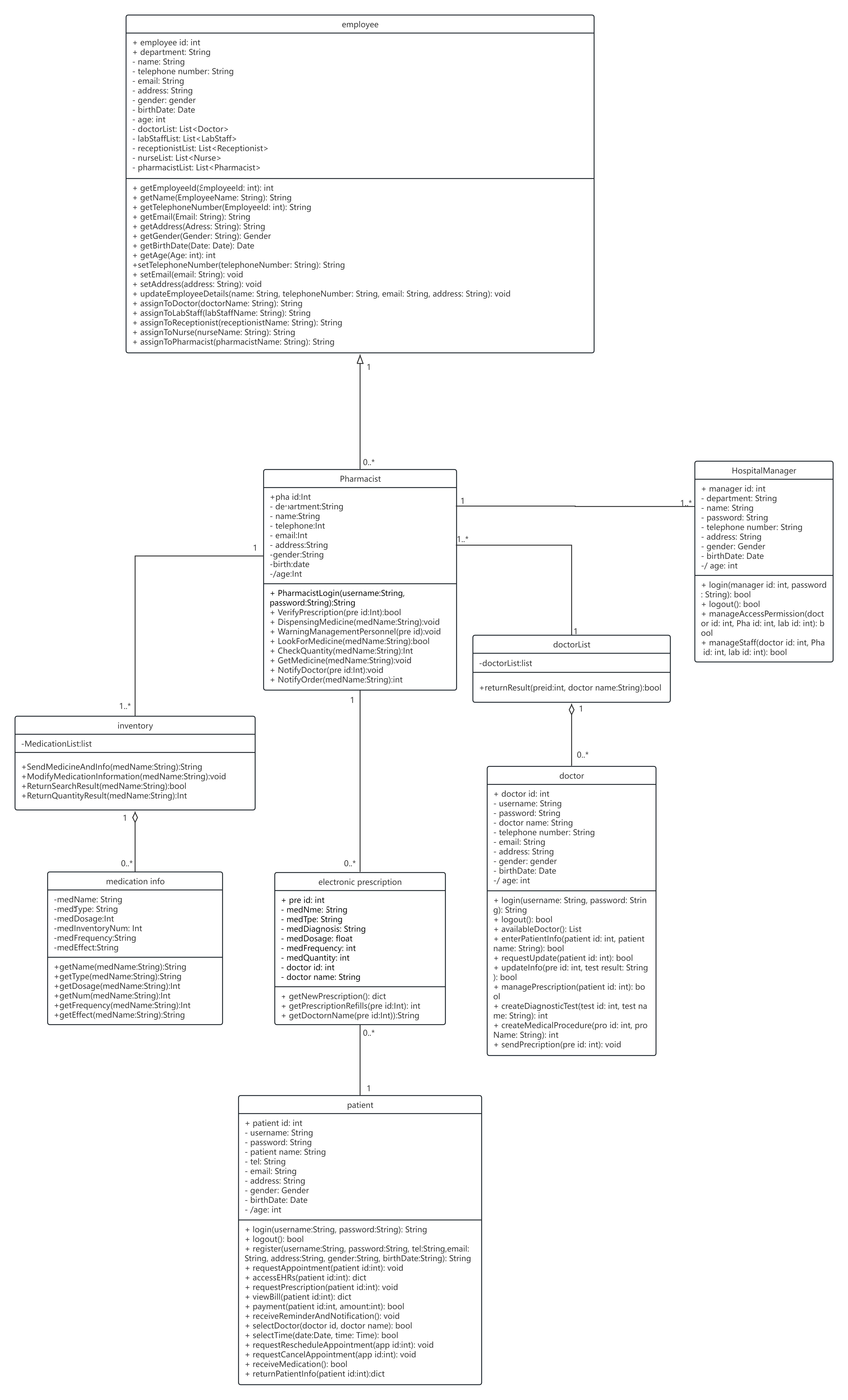
|  |  |
| --- | --- |
| **English name:** | **Young** |
| **Chinese name:** | **Yang Chengchen** |
| **Student Number:** | **202018010115** |
| **Module Code:** | **CHC6173** |
| **Module Name:** | **Software Engineering** |

**1.Task1(b)-Software Modelling and Specification**

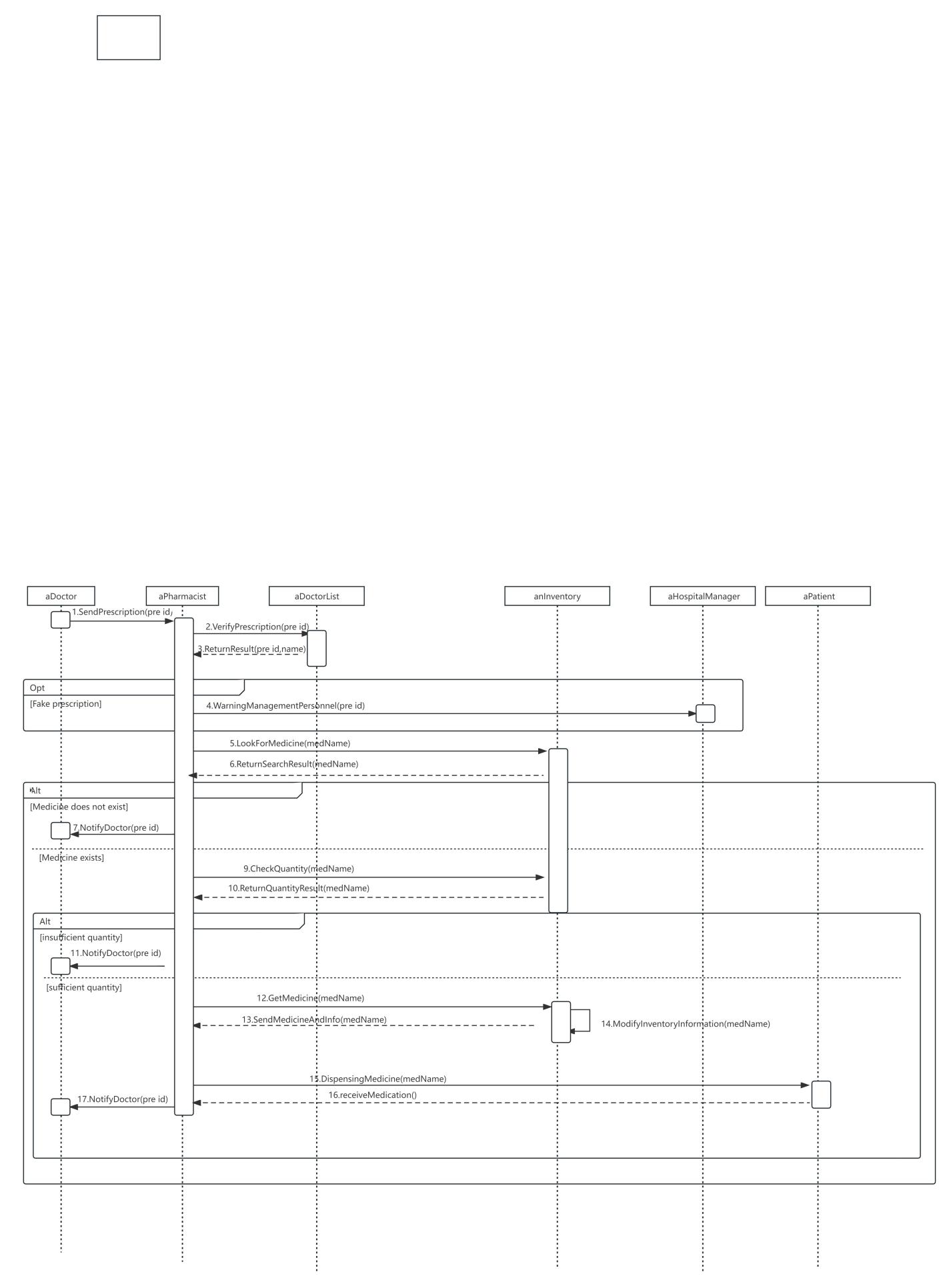
1.1Activity Model

****

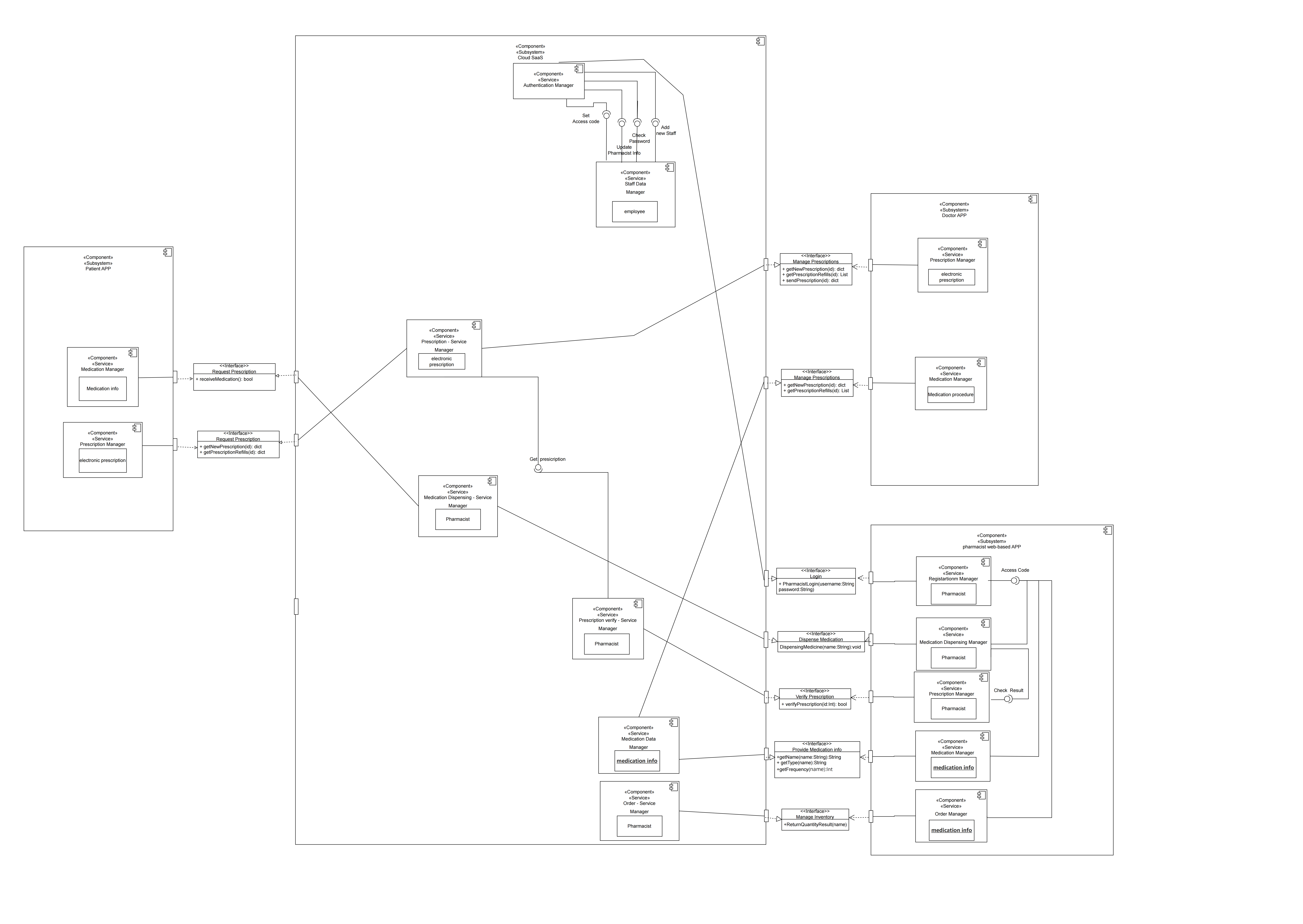
1.2Structural Model



1.3Behaviour Model

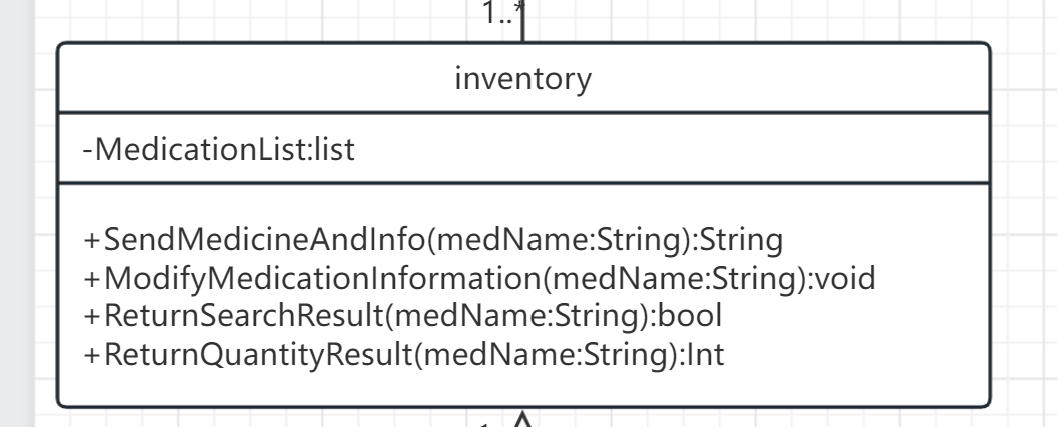


1. **Task2(b)-Software Architectural Design**

****

1. **Task 3- Software Testing** 
   1. Unit test plan

To test the microservice that provide the inventory class, I design test cases for the inventory class.



Class to test:Inventory

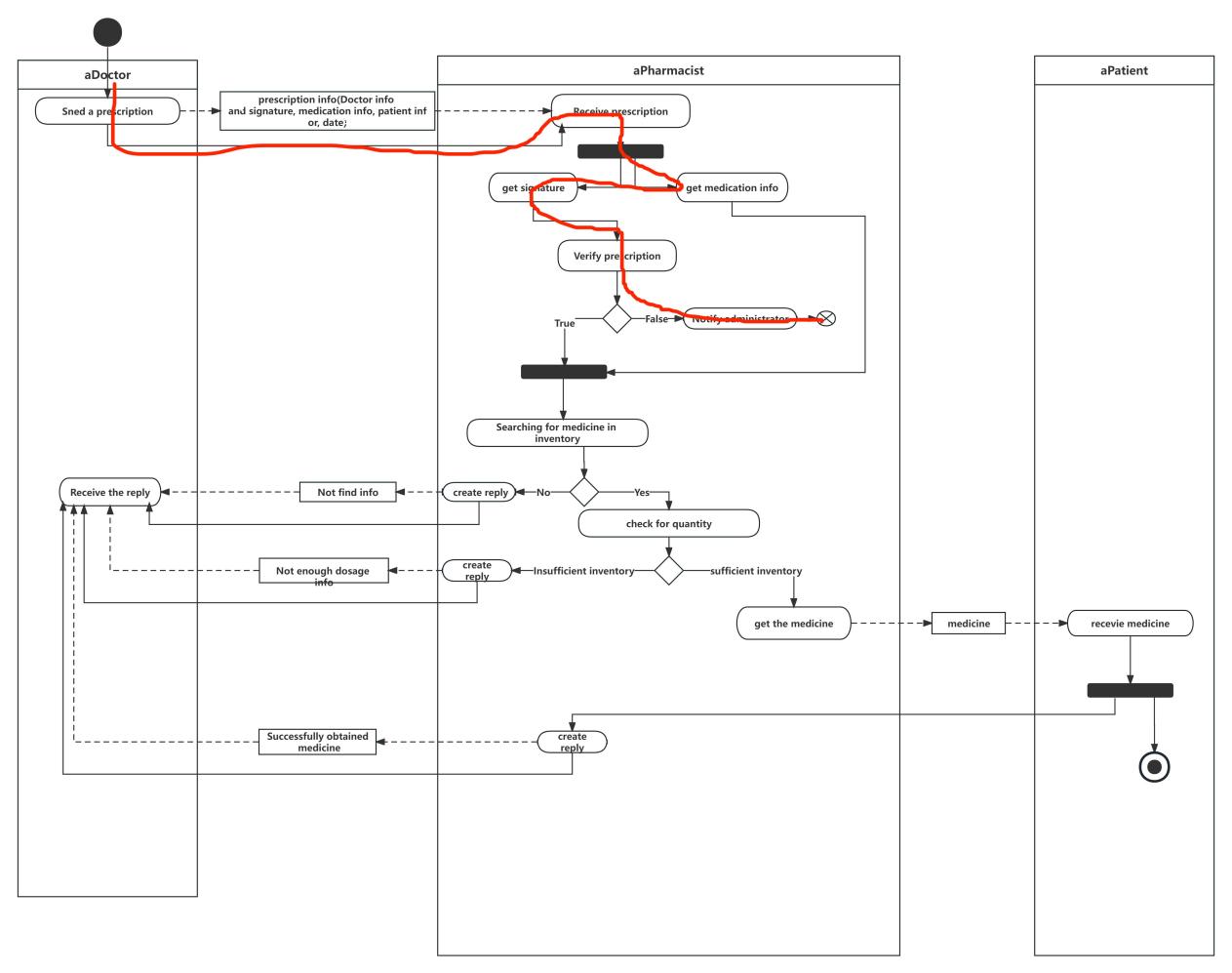
|  |  |  |
| --- | --- | --- |
| Test Case | Method and Parameters | Expected output |
| Successfully sent medication and its information to patients | SendMedicineAndInfo(Amoxicillin) | Successfully sent message |
| Medication and its information failed to sent to patients | SendMedicineAndInfo(Amoxicillin) | Error message |
| Reduce the quantity of medication in inventory | ModifyMedicationInformation(Amoxicillin) | nothing |
| Add the quantity of medication in inventory | ModifyMedicationInformation(Amoxicillin) | nothing |
| Successfully found the required medication in the inventory | ReturnSearchResult(Amoxicillin) | True |
| Not found the required medication in the inventory | ReturnSearchResult(Amoxicillin) | False |
| Find the quantity of required medication | ReturnQuantityResult(Amoxicillin) | Number of medication |

* 1. System test plan

Use Case: medication dispensing

3.2.1 Scenarios1

Scenario: Problem with prescription verification



|  |  |
| --- | --- |
| Doctor | pharmacist |
| 1.send a prescription | 2.receive prescription |
|  | 3.get signature |
|  | 4.get medication info |
|  | 5.verify prescription |
|  | 6.Notify administrator |

Derive Test Case from Scenarios:

Test Data

* Input:

prescription info:doctor signature

* Stored data:

a.On mobile devise: no data stored

b.On cloud: prescription info

* Output: Exit process

Test Process

1.Set up test context

a.Mobile device: not to contain prescription info;

b.Prescription database: (a) to contain doctor signature (b) to contain the required medication information

2.send prescription

a.Expected output: doctor send prescription to pharmacist

3.receive prescription

a.Expected output: pharmacist receive prescription

4.get signature

a.Expected output: the doctor signature

b.Check: if output match the expected

5.get medication info

a.Expected output: the medication info

b.Check: if output match the expected

6.input signature

7.verify prescription

a.Expected output: “False” of boolean value

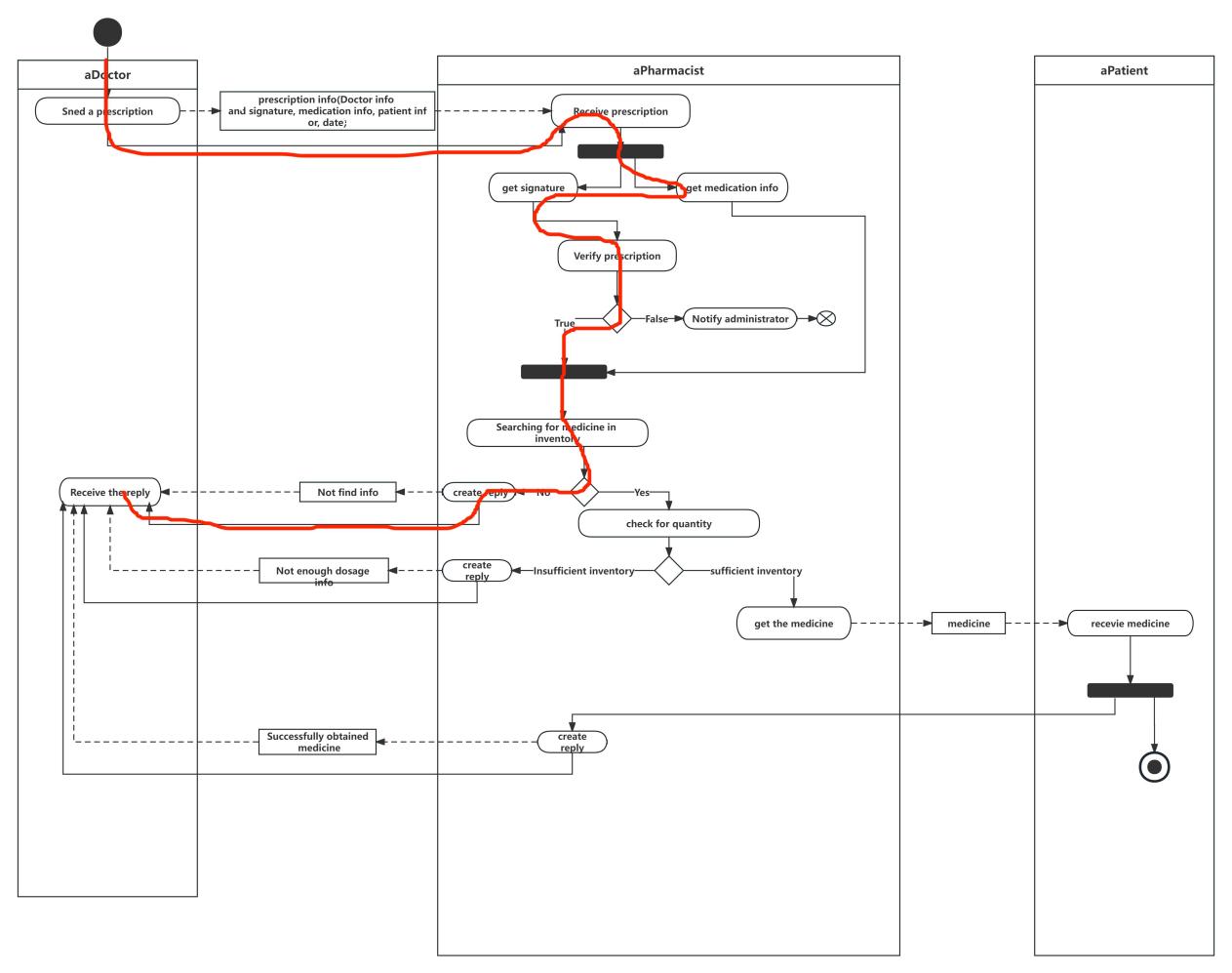
b.Check: if output match the expected

8.Notify administrator

a.Expected output: Notification sent to administrator

3,2,2 Scenarios2

Scenario: inventory don’t have the medication



|  |  |
| --- | --- |
| Doctor | pharmacist |
| 1.send a prescription | 2.receive prescription |
|  | 3.get signature |
|  | 4.get medication info |
|  | 5.verify prescription |
|  | 6.Searching for medicine in inventory |
|  | 7.create reply |
| 8.receive the reply |  |

Derive Test Case from Scenarios:

Test Data

* Input:

prescription info: the doctor signature, the required medication information

Inventory:the information about current inventory of the medication

* Stored data:

a.On mobile devise: no data stored

b.On cloud: prescription info, inventory list

* Output: a reply to doctor

Test Process

1.Set up test context

a.Mobile device: not to contain prescription info;

b.Prescription database: (a) to contain doctor signature (b) to contain the medication information

C.inventory database: to contain the medication information in the current inventory

2.send prescription

a.Expected output:doctor send prescription to pharmacist

3.receive prescription

a.Expected output: pharmacist receive prescription

4.get signature

a.Expected output: the doctor signature

b.Check: if output match the expected

5.get medication info

a.Expected output: the medication info

b.Check: if output match the expected

6.input signature and medication info

7.verify prescription

a.Expected output: “True” of boolean value

b.Check: if output match the expected

8.Look for Medicine

a.Expected output: “False”of boolean value

b.Check: if output match the expected.

9.create reply

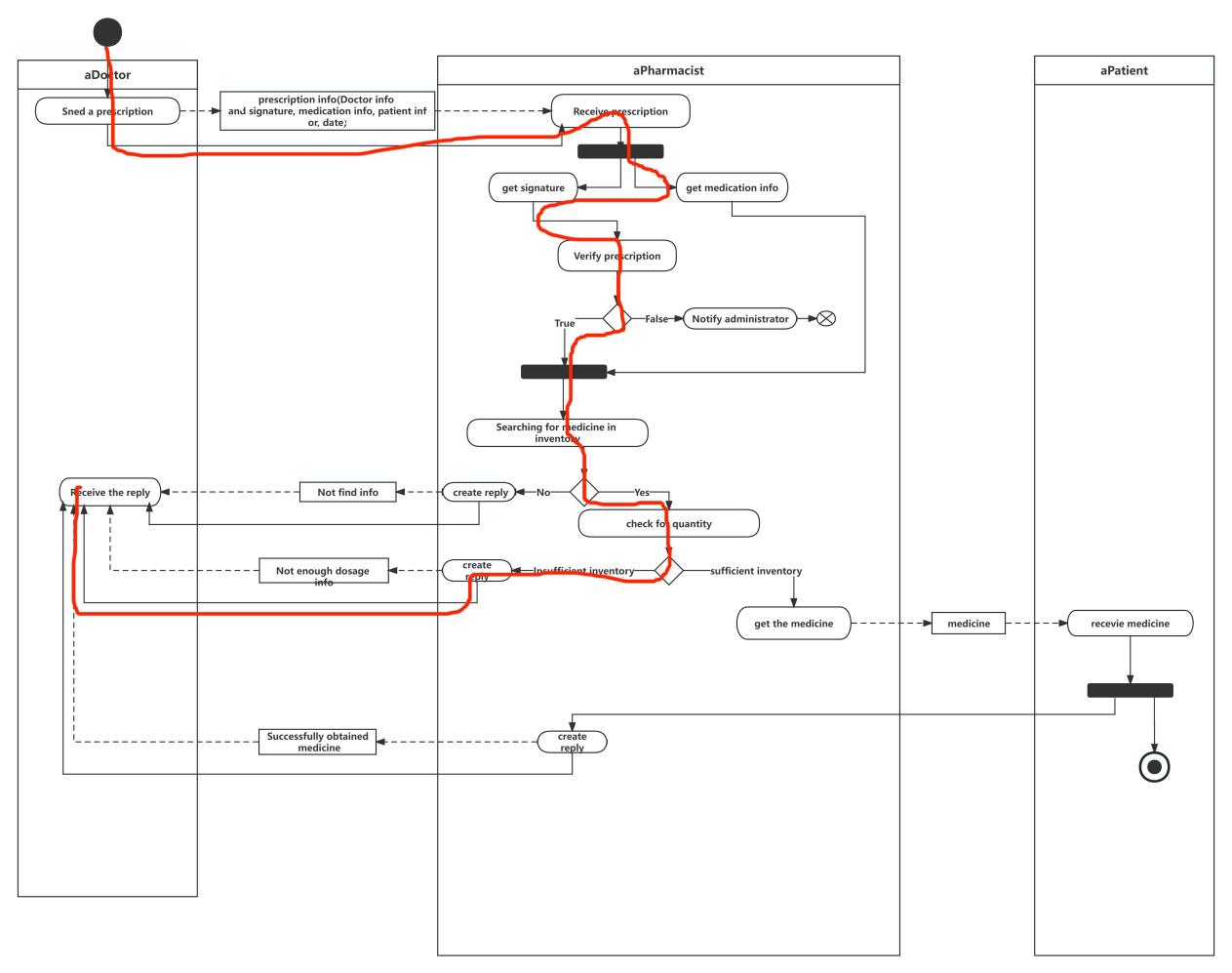
a.Expected output: a reply to create

10.receive the reply

a.Expected output:doctor receive a reply

3.2.3Scenarios3

Scenario: inventory don’t have enough dosage of the medication.



|  |  |
| --- | --- |
| Doctor | pharmacist |
| 1.send a prescription | 2.receive prescription |
|  | 3.get signature |
|  | 4.get medication info |
|  | 5.verify prescription |
|  | 6.Searching for medicine in inventory |
|  | 8.check for quantity |
|  | 9.create reply |
| 10.receive the reply |  |

Derive Test Case from Scenarios:

Test Data

* Input:

prescription info: the doctor signature, the required medication information

Inventory:the information about current inventory of the medication

* Stored data:

a.On mobile devise: no data stored

b.On cloud: prescription info, inventory list

* Output: an reply to doctor

Test Process

1.Set up test context

a.Mobile device: not to contain prescription info;

b.Prescription database: (a) to contain doctor signature (b) to contain the medication information

C.inventory database: to contain the medication info in the current inventory

2.send prescription

a.Expected output: doctor send prescription to pharmacist

3.receive prescription

a.Expected output: pharmacist receive prescription

4.get signature

a.Expected output: the doctor signature

b.Check: if output match the expected

5.get medication info

a.Expected output: the medication info

b.Check: if output match the expected

6.input signature and medication info

7.verify prescription

a.Expected output: “True” of boolean value

b.Check: if output match the expected

8.Look for Medicine

a.Expected output: “True” of boolean value

b.Check: if output match the expected.

9.check for quantity

a.Expected output: “False” of boolean value

b.Check: if output match the expected.

10.create reply

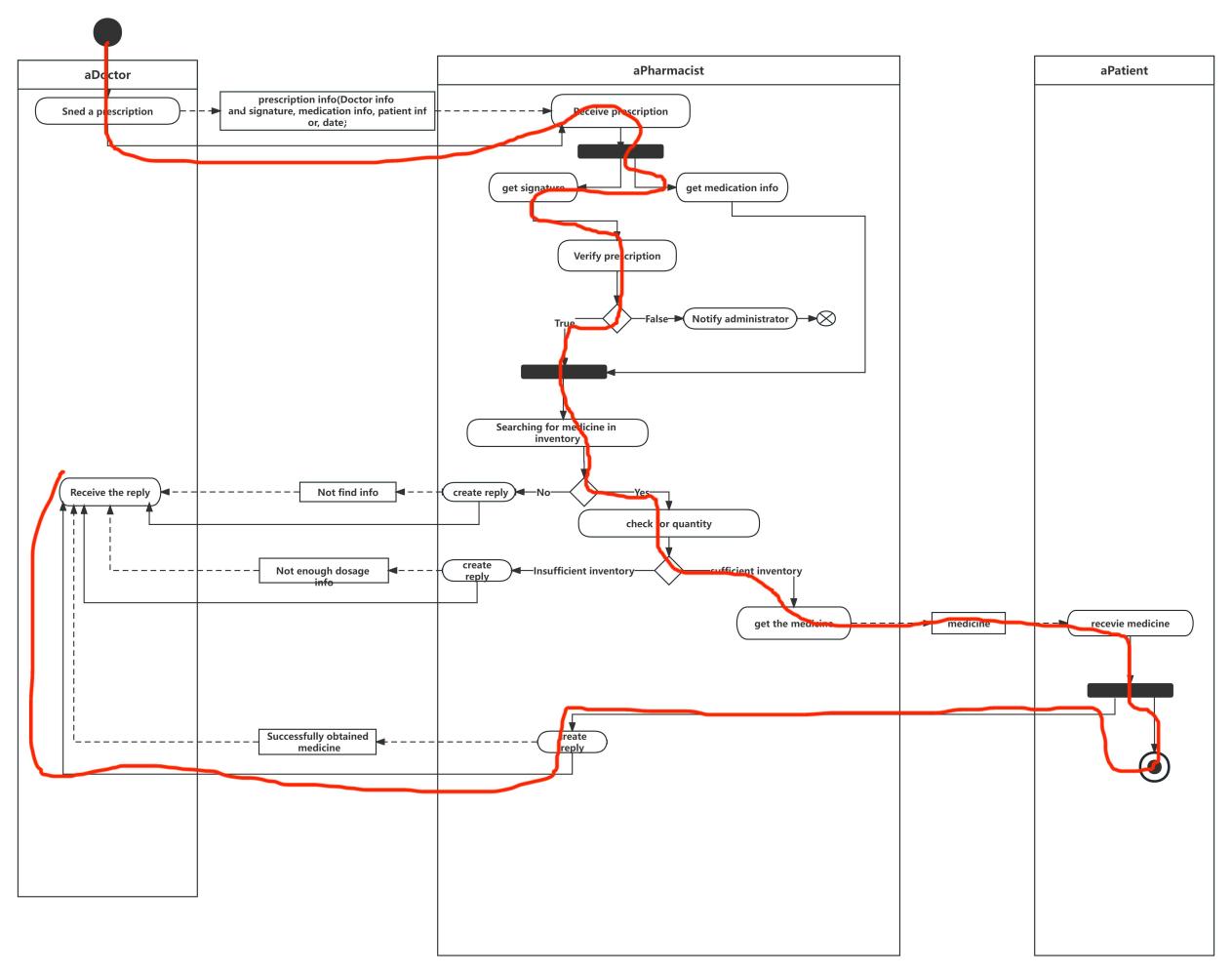
a.Expected output: a reply to create

11.receive the reply

a.Expected output: doctor receive a reply

3.2.4 Scenarios4

Scenario:Successful medication dispensing



|  |  |  |
| --- | --- | --- |
| Doctor | pharmacist | patient |
| 1.send a prescription | 2.receive prescription |  |
|  | 3.get signature |  |
|  | 4.get medication info |  |
|  | 5.verify prescription |  |
|  | 6.Searching for medicine in inventory |  |
|  | 8.check for quantity |  |
|  | 9.get the medicine |  |
| 11.receive the reply |  | 10.receive the medicine |

Derive Test Case from Scenarios:

Test Data

* Input:

prescription info: the doctor signature, the required medication information

Inventory:the information about current inventory of the medication

* Stored data:

a.On mobile devise: no data stored

b.On cloud: prescription info, inventory list

* Output: the patient receive the medicine

Test Process

1.Set up test context

a.Mobile device: not to contain prescription info;

b.Prescription database: (a) to contain doctor signature (b) to contain the medication information

c.inventory database: to contain the medication info in the current inventory

2.send prescription

a.Expected output: doctor send prescription to pharmacist

3.receive prescription

a.Expected output: pharmacist receive prescription

4.get signature

a.Expected output: the doctor signature

b.Check: if output match the expected

5.get medication info

a.Expected output: the medication info

b.Check: if output match the expected

6.input signature and medication info

7.verify prescription

a.Expected output: “True” of boolean value

b.Check: if output match the expected

8.Look for medicine

a.Expected output: “True” of boolean value

b.Check: if output match the expected.

9.check for quantity

a.Expected output: “True” of boolean value

b.Check: if output match the expected.

10.get the medication

a.Expected output: pharmacist get the medication

11.receive the medicine

a.Expected output: the patient get the medication

12.create reply

a.Expected output: a reply to create

13.receive the reply

a.Expected output:doctor receive a reply