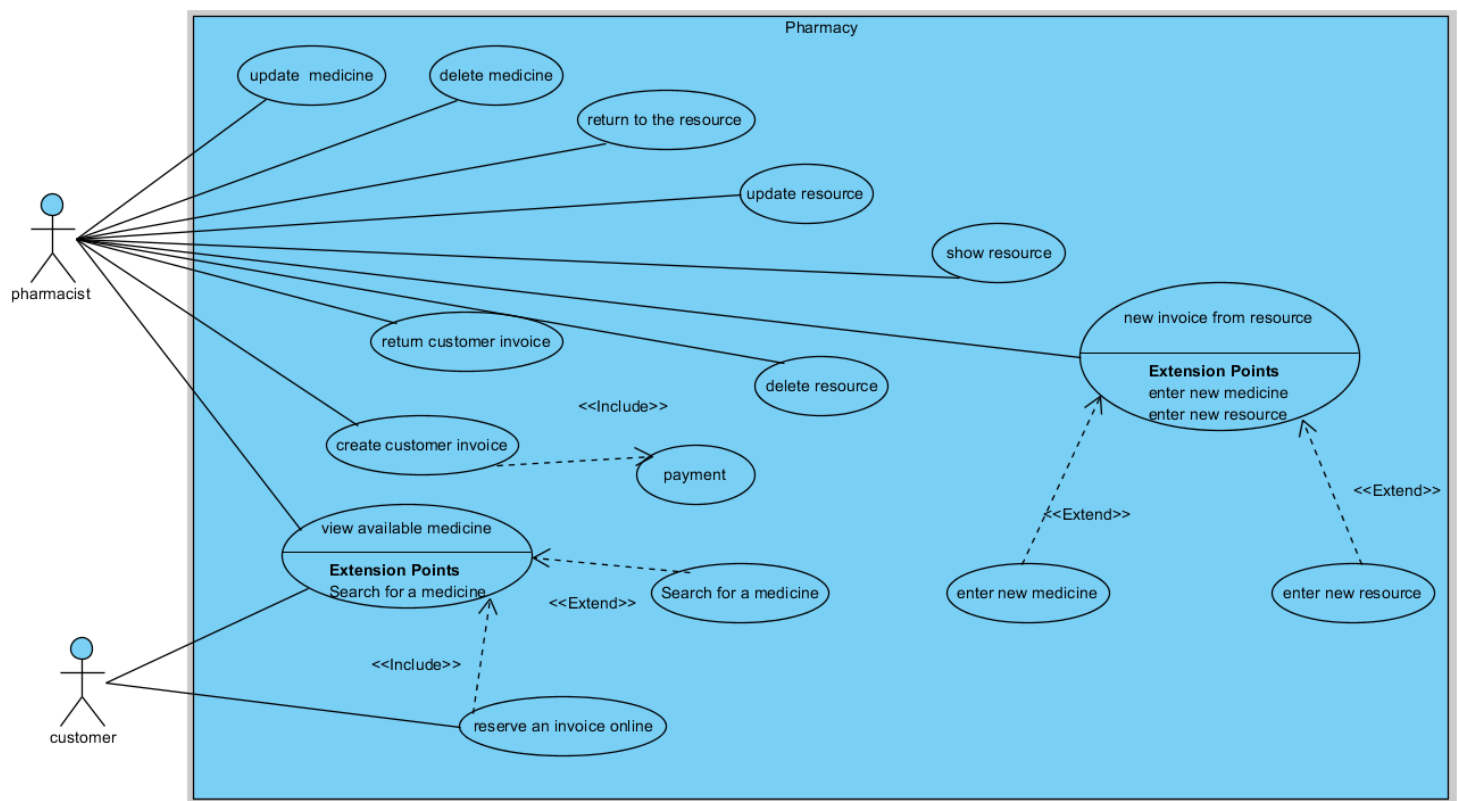1/9/2020

Software Engineering

GRPharmicy
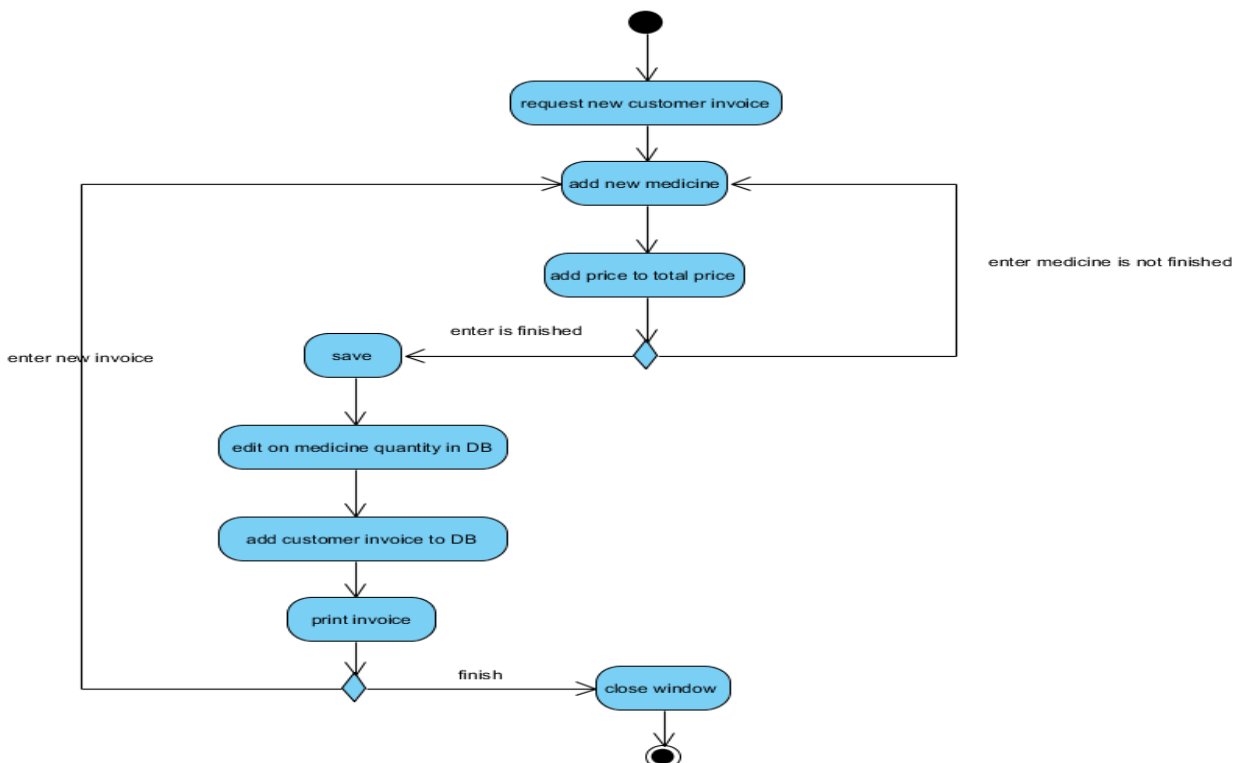
Engineer: Kinan Abd Albaki
Students: Rafah Almounajed, Ghoufran Albakri
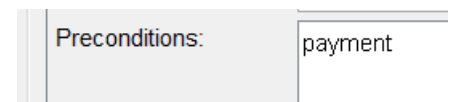
Use case:

Pharmacy

- update medicine
- delete medicine
- return to the resource
- update resource
- show resource
- new invoice from resource
  - **Extension Points**
    enter new medicine
    enter new resource
- return customer invoice
- delete resource
- create customer invoice
- payment
- view available medicine
  - **Extension Points**
    Search for a medicine
- Search for a medicine
- enter new medicine
- enter new resource
- reserve an invoice online

<<Include>>
<<Extend>>
<<Extend>>
<<Extend>>
<<Include>>

pharmacist

customer

New customer invoice: activity diagram

- request new customer invoice
- add new medicine
- add price to total price
- enter medicine is not finished
- enter is finished
- save
- enter new invoice
- edit on medicine quantity in DB
- add customer invoice to DB
- print invoice
- finish
- close window

New customer invoice: sequence diagram

New customer invoice: description

create customer invoice Details

Name: create customer invoice

| Info | Flow of Events | Details | Requirements | Diagrams | Test Plan | References | Description |

Rank: <Unspecified>

Justification:

Primary Actors: pharmacist

Supporting Actors:

Documentation:

HTML B I U ≡ ≡ ≡ ≡ := := F Fꜰ

```
to achieve the required we will do this steps:
1-enter the date of invoice
2-then will start to enter items with there quantity and price
3-when finished .. save this invoice in DB with do an edit in DB on saved items.
```

create customer invoice Details

Name: create customer invoice

| Info | Flow of Events | Details | Requirements | Diagrams | Test Plan | Reference |

Flow of Events

1. enter the date of invoice
2. then will start to enter items with there quantity and price
3. when finished .. save this invoice in DB with do an edit in DB on saved items.

Preconditions: payment

Payment activity diagram:
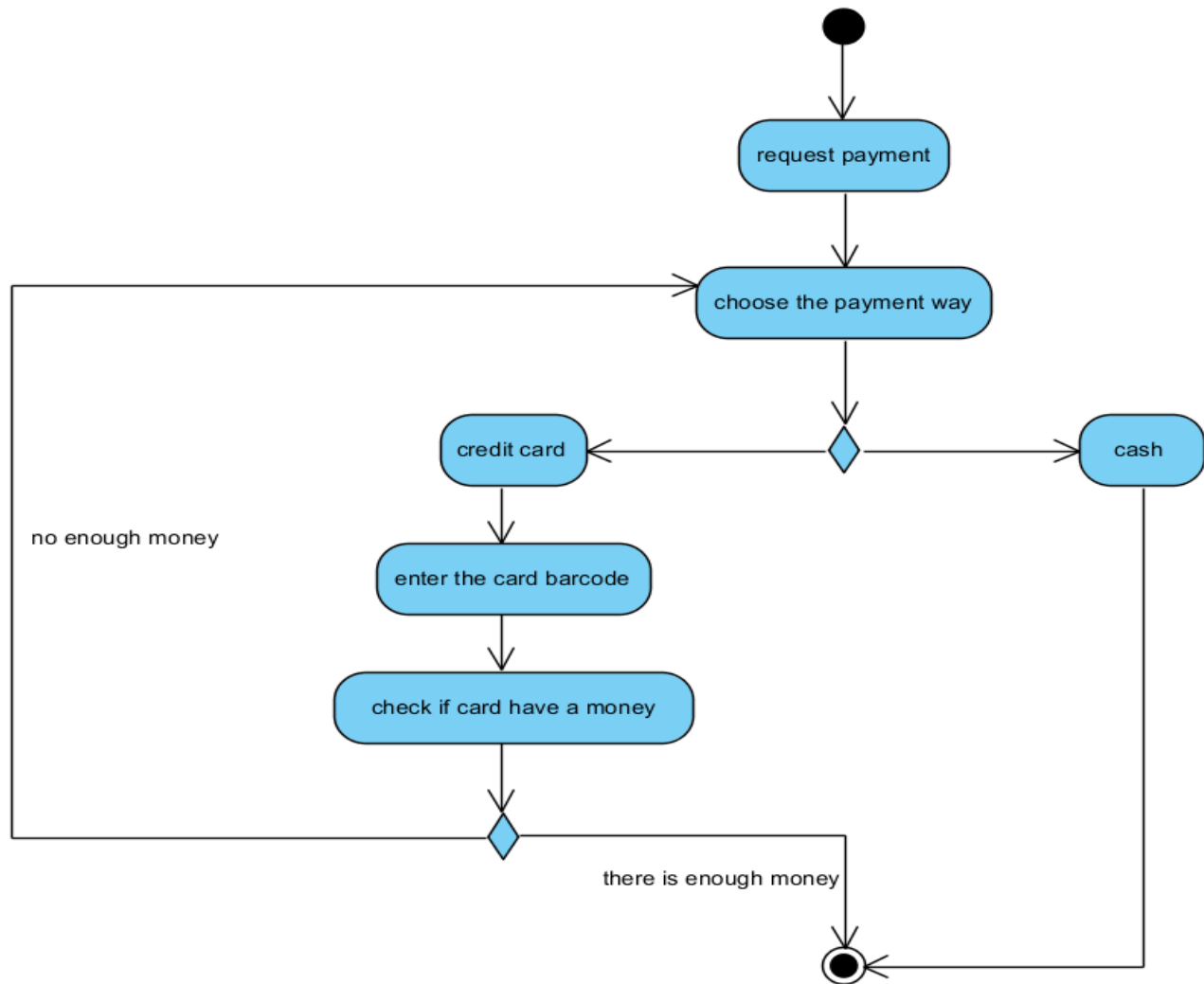
Payment: description

payment Details

Name: payment

Info	Flow of Events	Details	Requirements	Diagrams	Te

Rank: <Unspecified>

Justification:

Primary Actors:

Supporting Actors:

Documentation:

HTML   B I U ≡ ≡ ≡ ≡ ≡ F Fғ

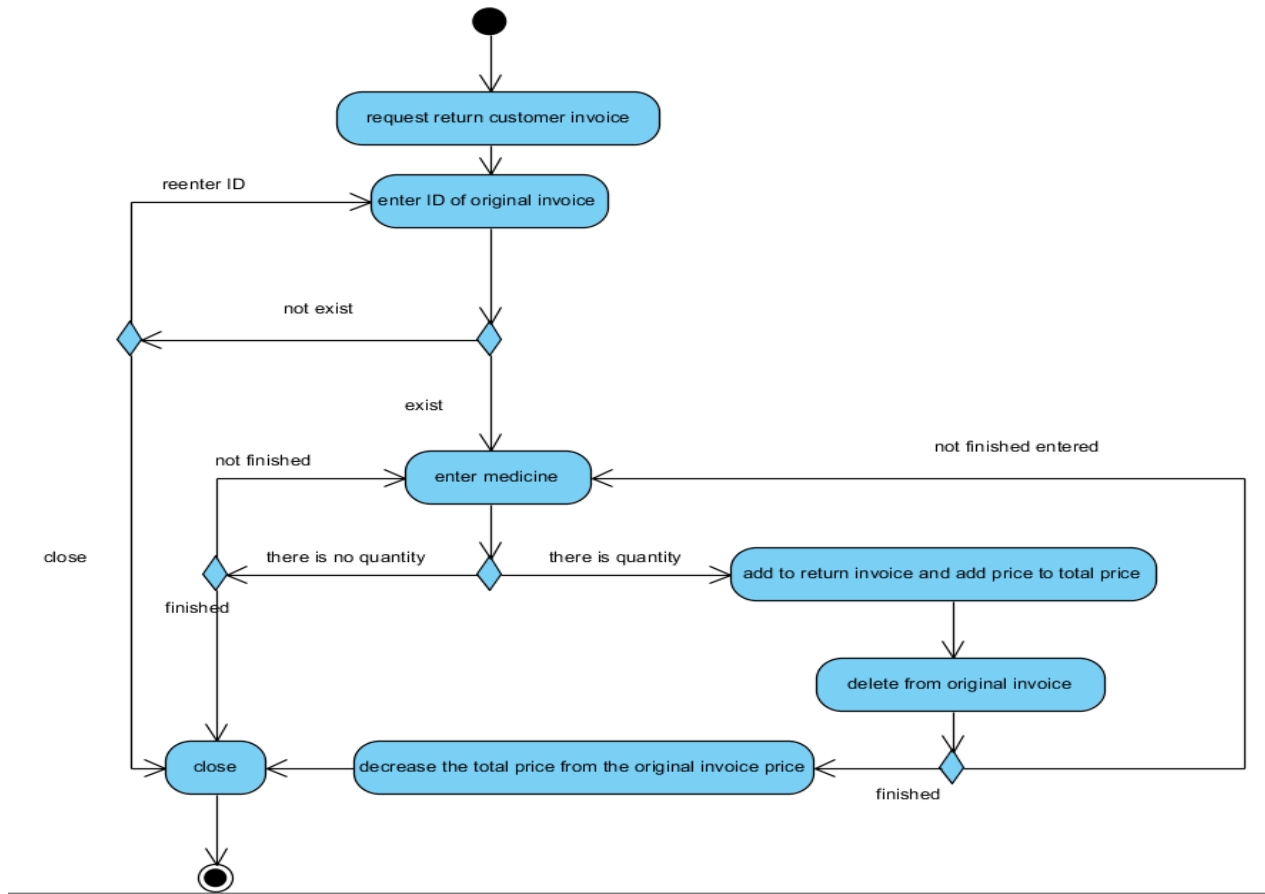choose the way of payment then pay and save

payment Details

Name: payment

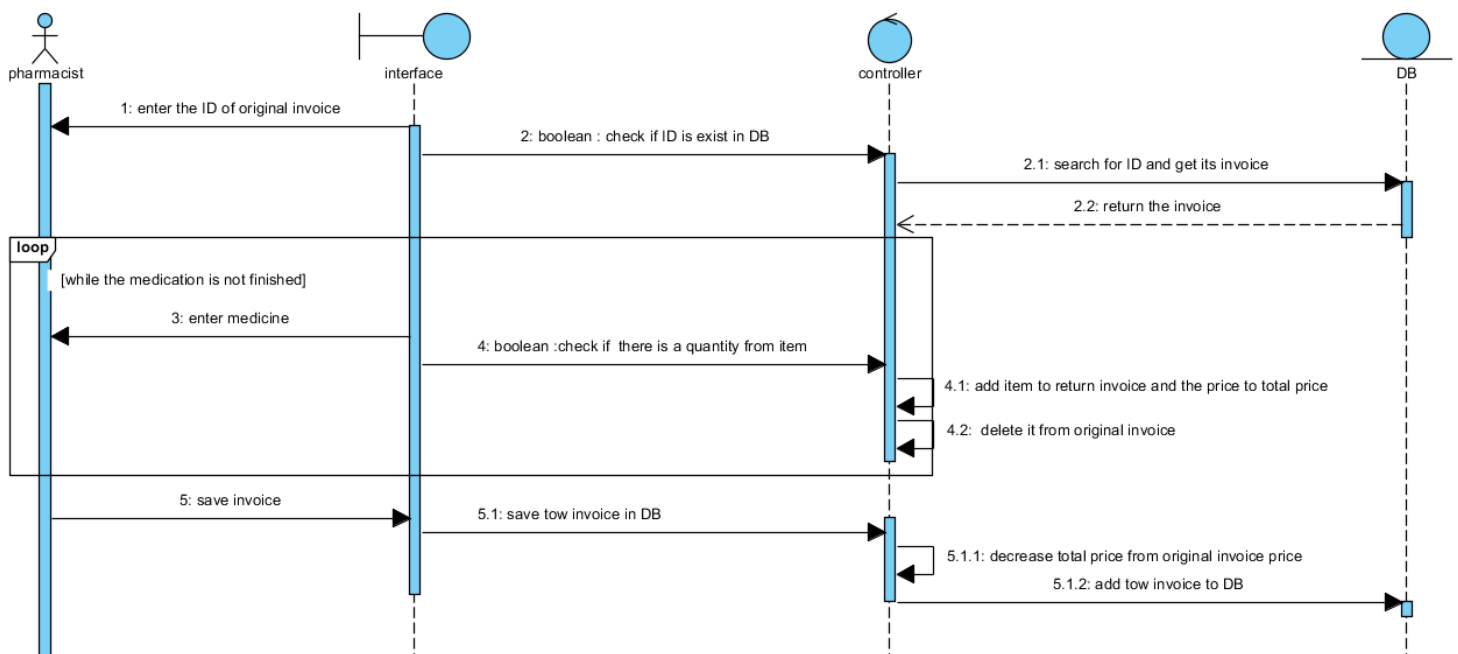| Info	Flow of Events	Details	Requirements	Diagrams	Test Plan	References

Flow of Events

1. enter the way of payment
2. if cash save invoice
3. if credit card then check if it have enough money or not then pay and save invoice

Post-conditions: create customer invoice

Return customer invoice: activity diagram

```
                              ●
                              │
                              ▼
              ┌────────────────────────────────┐
              │  request return customer invoice │
              └────────────────────────────────┘
                              │
   reenter ID                 ▼
              ┌────────────────────────────────┐
              │   enter ID of original invoice   │
              └────────────────────────────────┘
                              │
              not exist        ▼
         ◇◄─────────────────◇
                              │
                           exist
                              ▼
 not finished        ┌──────────────────┐         not finished entered
              ┌─────►│   enter medicine  │◄──────────────────────────┐
              │      └──────────────────┘                            │
              │                │                                     │
   close      │  there is no quantity    there is quantity           │
              ◇◄────────────◇──────────►┌──────────────────────────────────┐
              │                         │ add to return invoice and add      │
           finished                     │ price to total price               │
              │                         └──────────────────────────────────┘
              │                                        │
              │                                        ▼
              │                         ┌──────────────────────────────┐
              │                         │  delete from original invoice  │
              │                         └──────────────────────────────┘
              ▼                                        │
          ┌───────┐   ┌────────────────────────────────────┐    ◇
          │ close │◄──│ decrease the total price from the    │◄───┘
          └───────┘   │ original invoice price               │  finished
              │       └────────────────────────────────────┘
              ▼
              ◉
```

Return customer invoice: sequence diagram

Return customer invoice:

return customer invoice Details

Name: return customer invoice

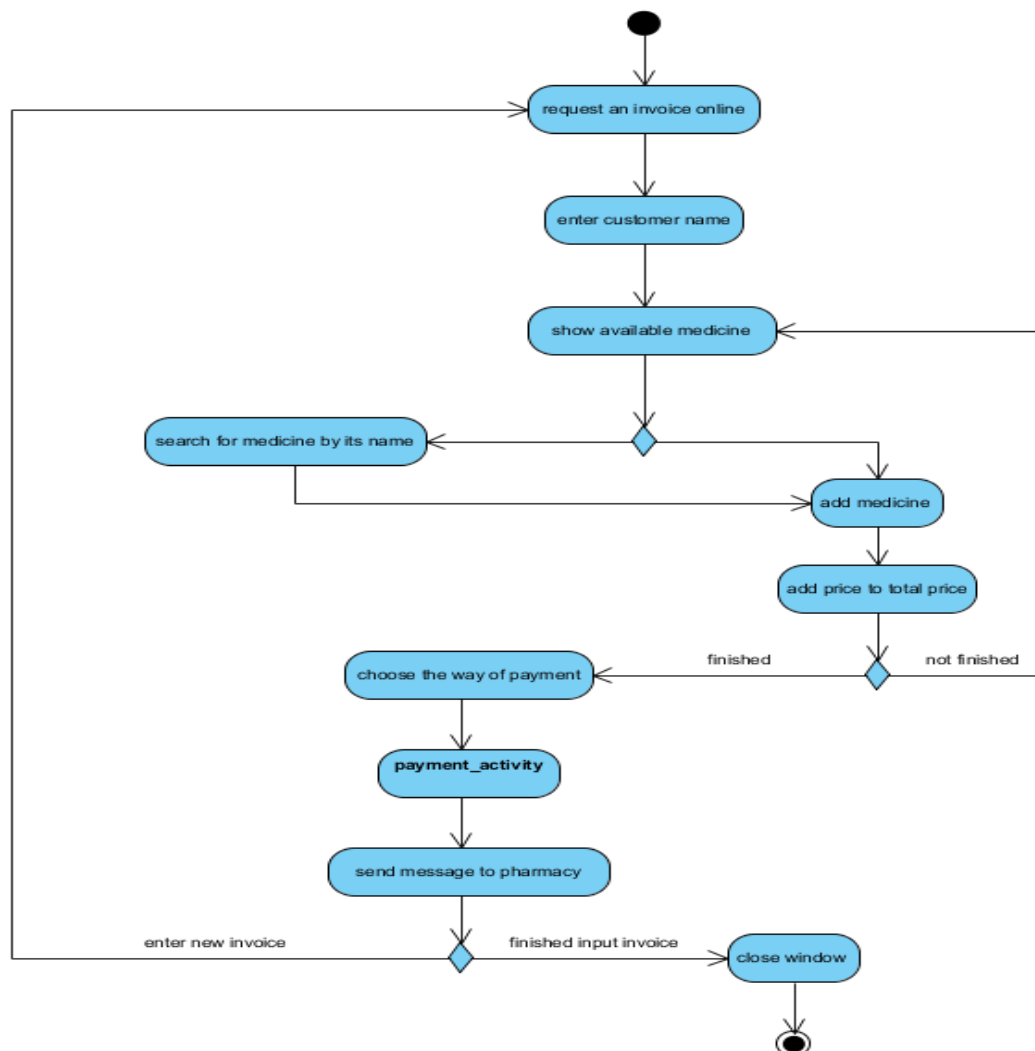| Info | Flow of Events | Details | Requirements | Diagrams | Test Plan | References | Description |

Rank: <Unspecified>

Justification:

Primary Actors: pharmacist

Supporting Actors:

Documentation:

HTML

```
to achieve the required we will do this steps:
You will enter the return date and customer name
and make sure that the customer exist in DB
if yes, you can start enter items with their quantity
when you have finished then save this return invoice in DB
and do an edit on quantity items and the mean price in DB
```

return customer invoice Details

Name: return customer invoice

| Info | Flow of Events | Details | Requirements | Diagrams | Test Plan | References |

Flow of Events

1. create new return customer invoice and enter the ID of original customer invoice
2. enter the return date
3. enter the return items with there quantity
4. decrease total price from original invoice price

Preconditions:
1-pharmacist should have an account in this website
2-enter customer invoice before this return invoice

Reserve an invoice online: activity diagram

Reserve an invoice online: sequence diagram
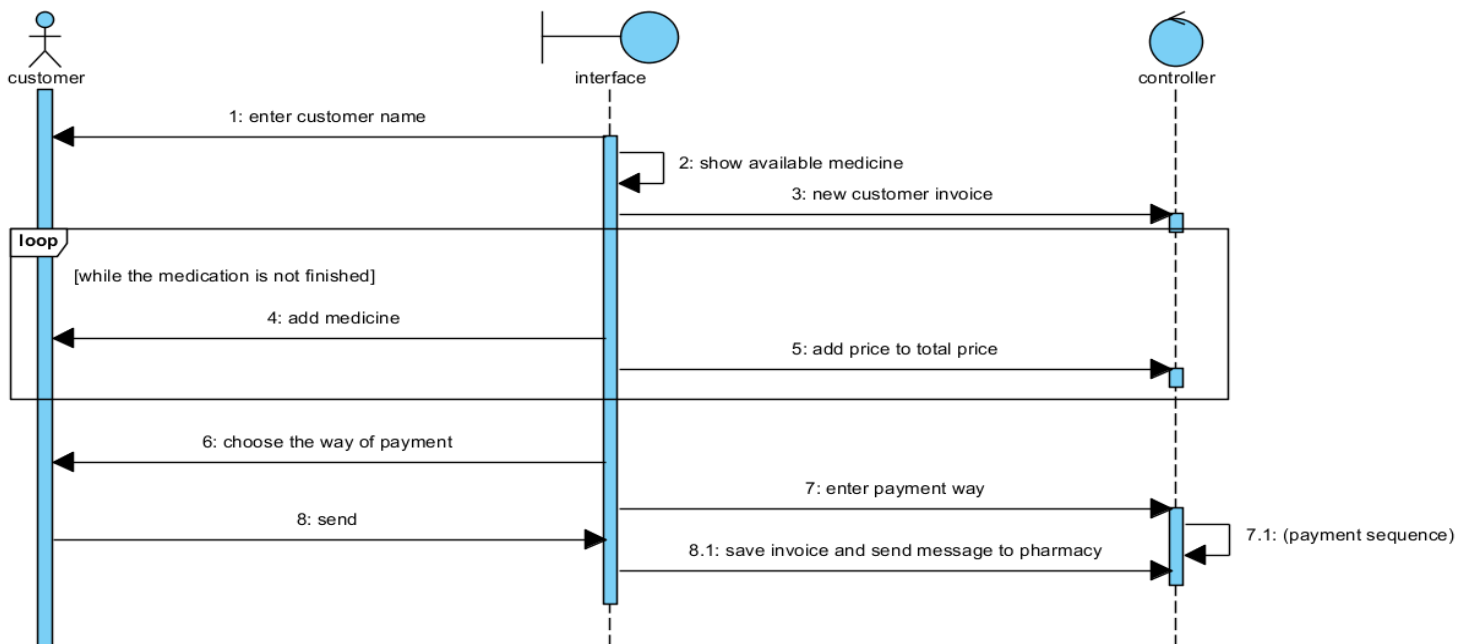
Reserve an invoice online: description

reserve an invoice online Details

Name: reserve an invoice online

Info | Flow of Events | Details | Requirements | Diagrams | Test Plan | References

Flow of Events

1. view available medicine
2. choose medicine and add its price to total price
3. choose the way of payment
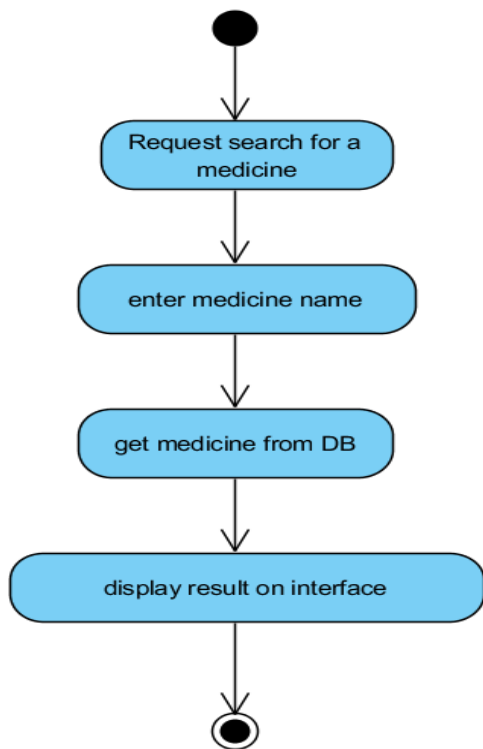4. send invoice such a message to pharmacy

Name: reserve an invoice online

Info | Flow of Events | Details | Requirer

Rank: <Unspecified>

Justification:

Primary Actors: customer
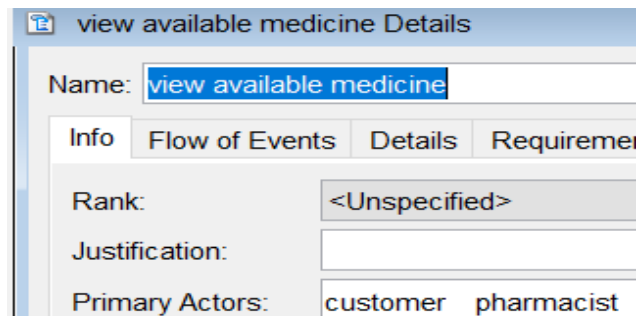
Search for a medicine:

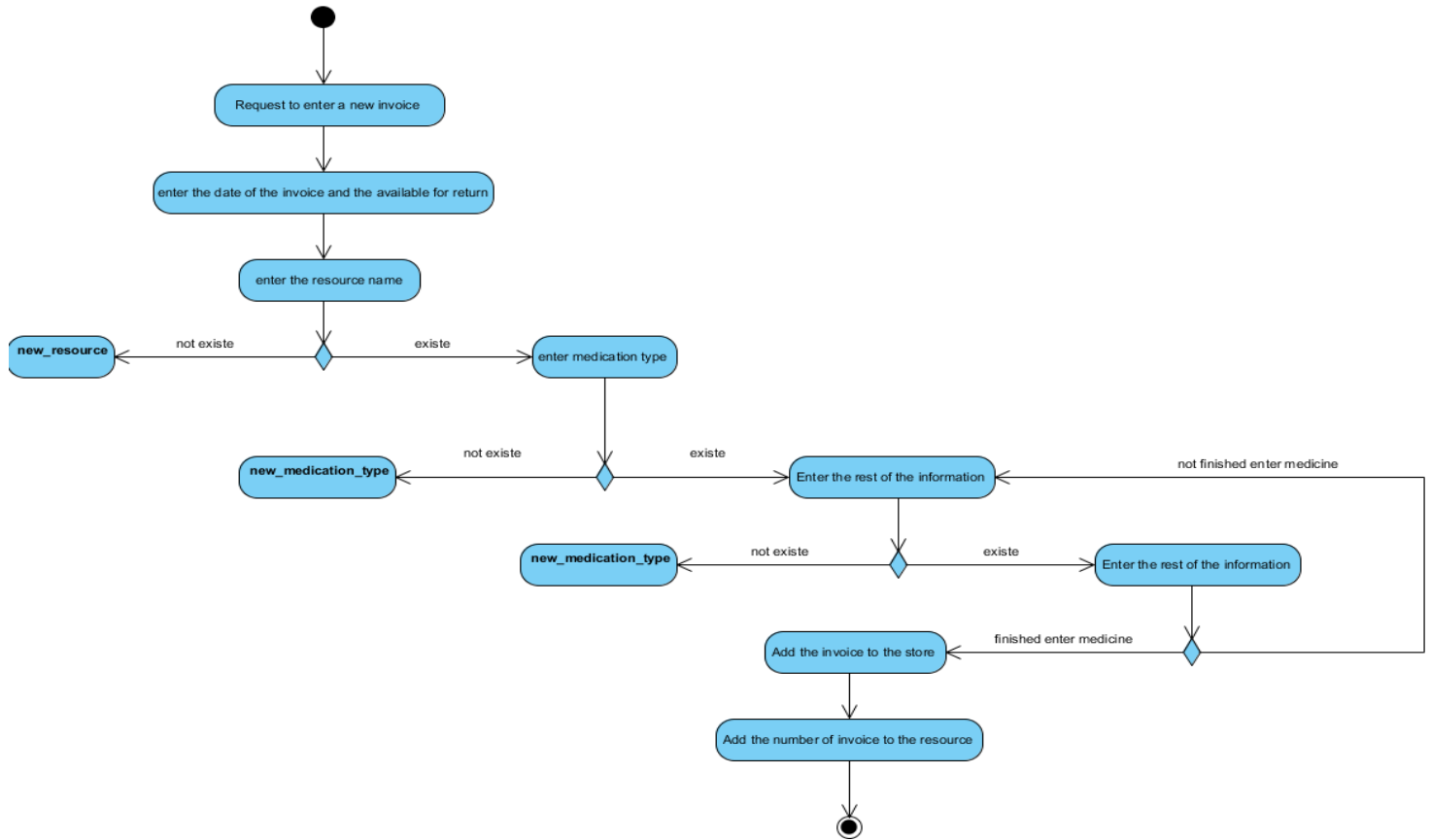## activity diagram
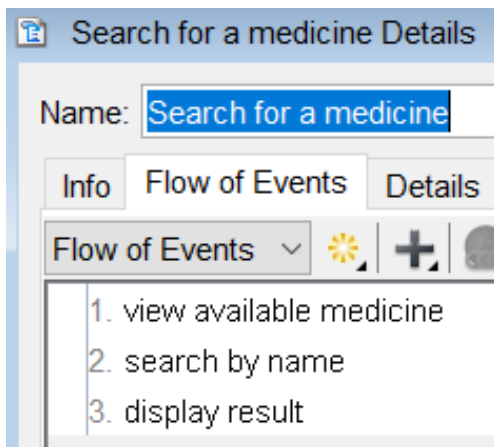
description

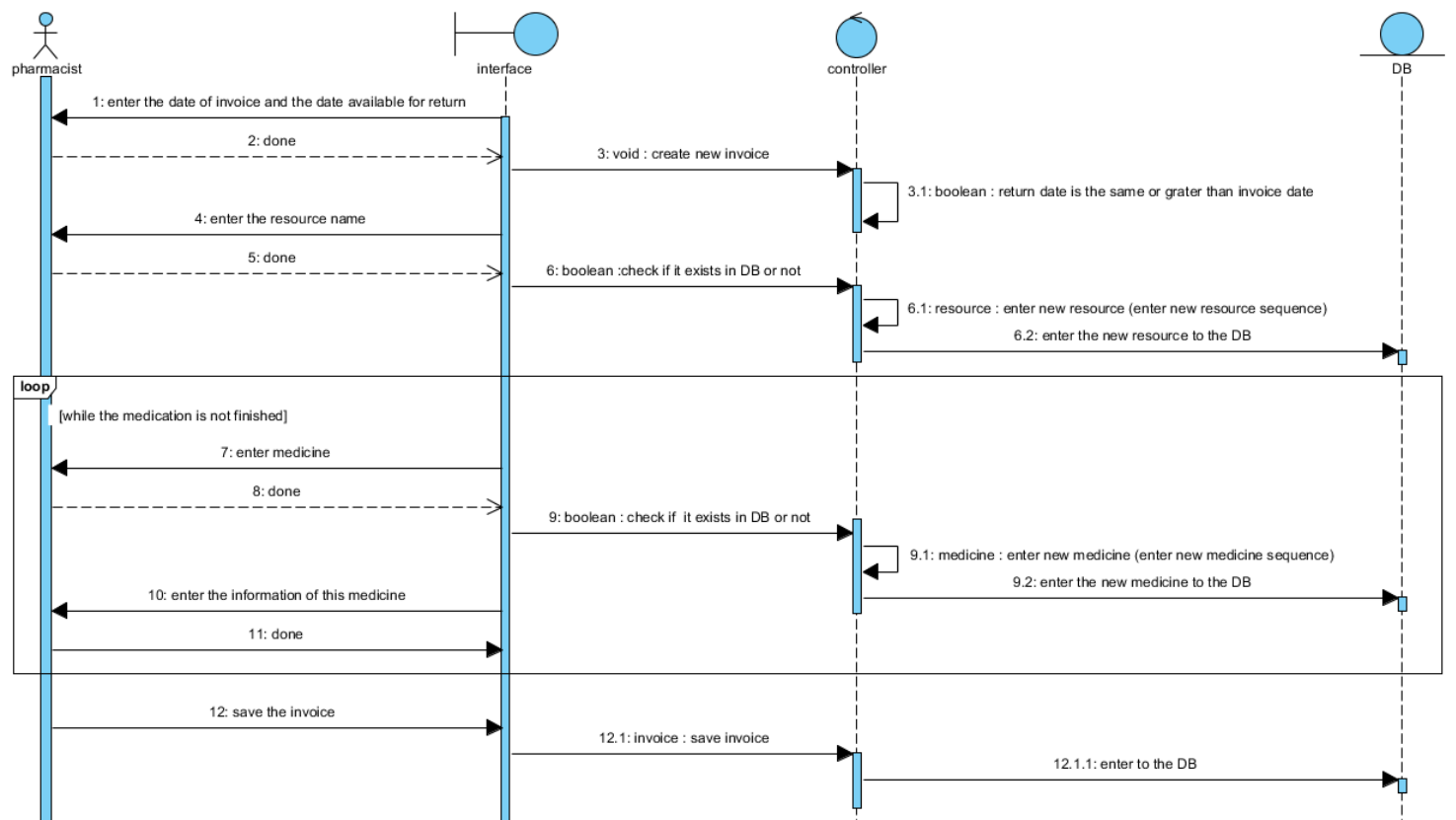## View available medicine:

activity diagram:
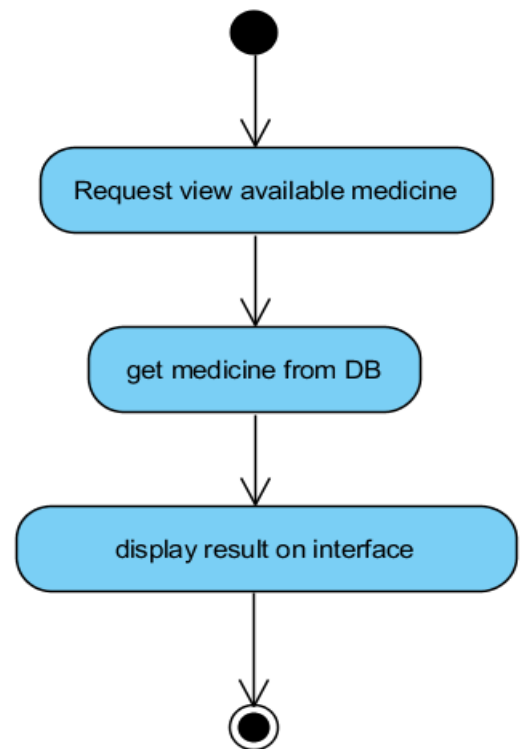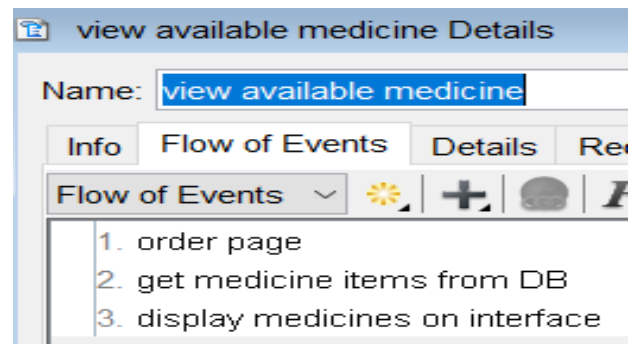
description

Enter new invoice: activity diagram

Enter new invoice: sequence diagram

Description:

new invoice from resource Details

Name: new invoice from resource

Info | Flow of Events | Details | Requirements | Diagrams | Test Plan | References | Description

Rank: <Unspecified>

Justification: add a new invoice that pay from resource

Primary Actors: pharmacist

Supporting Actors:

Documentation:

HTML   B *I* U ≡ ≡ ≡ ≡ ≡ F Fʀ

```
to achieve the required we will do this steps:
1- enter the date of invoice and available date for return items from this invoice
or all of this invoice and the available date for return must be the same or grater than of invoice date
2- then will enter the resource name and make sure that the resource is exist in DB earlier
and if not you must order to enter a new resource in DB
3- then will start to enter items with (start date ,end date ,quantity and price)
4- when finished .. save this invoice in DB with do an edit in DB on saved items.
```

new invoice from resource Details

Name: new invoice from resource

Info | Flow of Events | Details | Requir

Flow of Events ∨

1. select resource name
2. insert invoice date
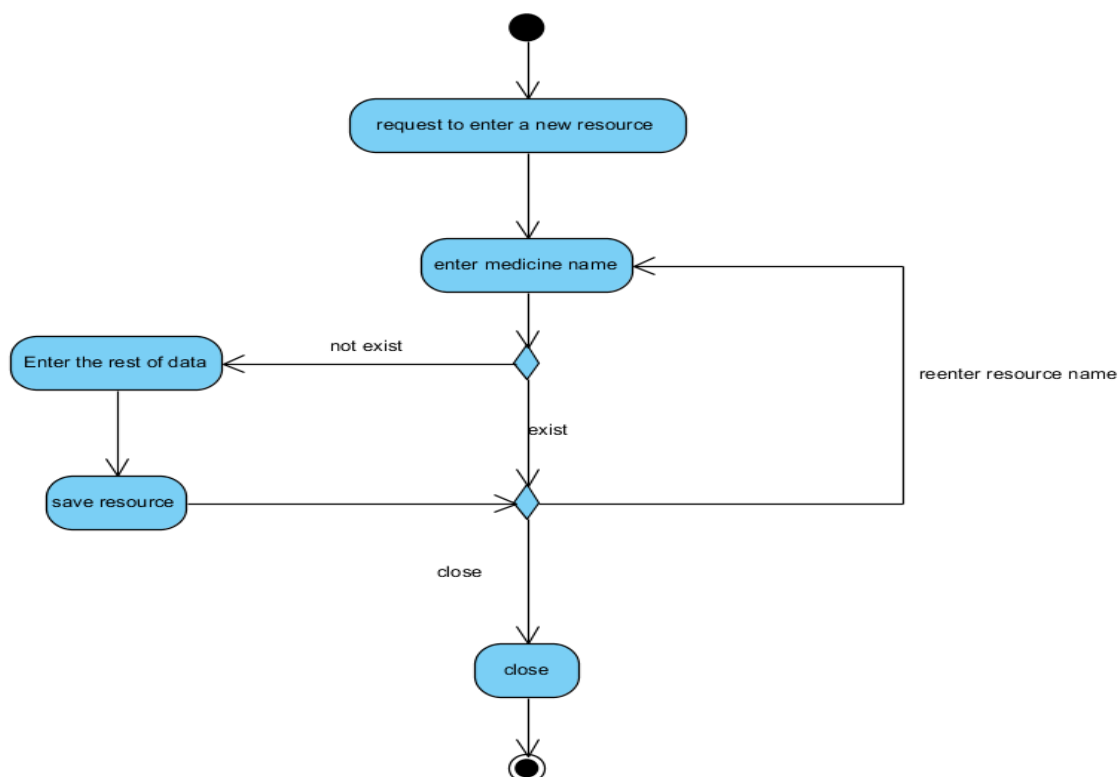3. add medicines and there info
4. insert available return invoice

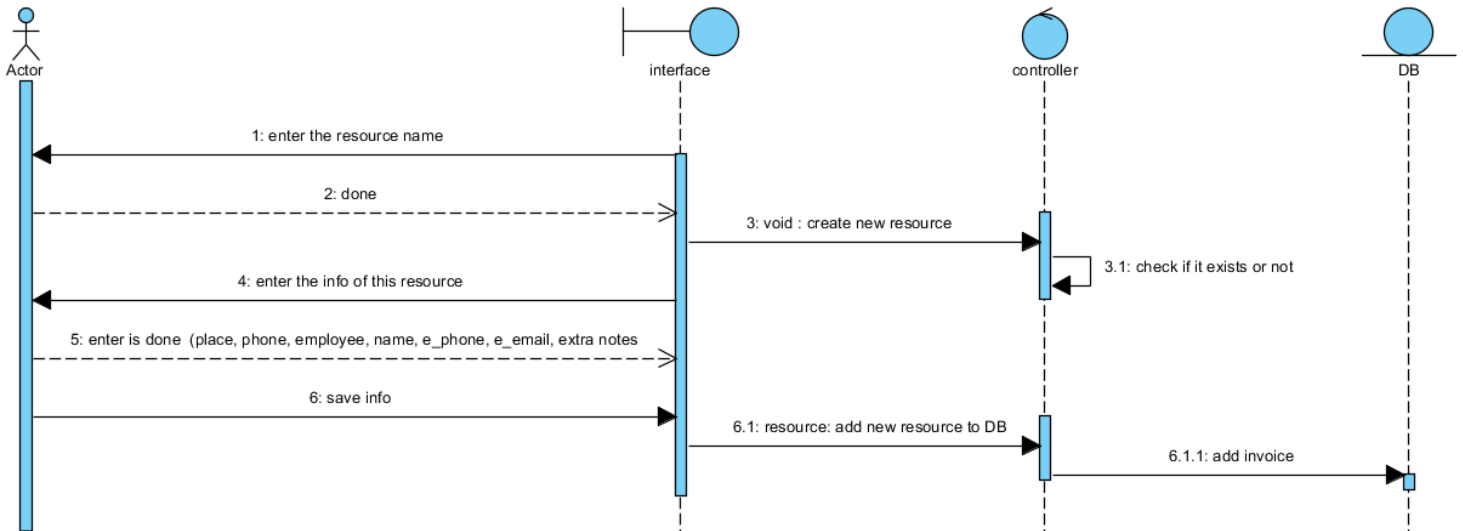Preconditions: pharmacist should have an account in this website

Post-conditions: return resource invoice

Enter new resource: activity diagram

Enter new resource: sequence diagram

Description:

enter new resource Details

Name: enter new resource

| Info | Flow of Events | Details | Requirements | Diagrams | Test Plan | References | Description |

Rank: <Unspecified>

Justification: add new not existed resource that pharmacy deals with

Primary Actors: pharmacist

Supporting Actors:

Documentation:

HTML   B I U ≡ ≡ ≡ ≣ ≣ F Fᵣ

```
to achieve the required we will do this steps:
will enter the resource name and check if the resource existed earlier or not ,
if not so you will enter the resource information (place, phone, employee, employee name,
 employee email, employee phone and extra note if you want)
after finished .. save the resource in DB
```

enter new resource Details

Name: enter new resource

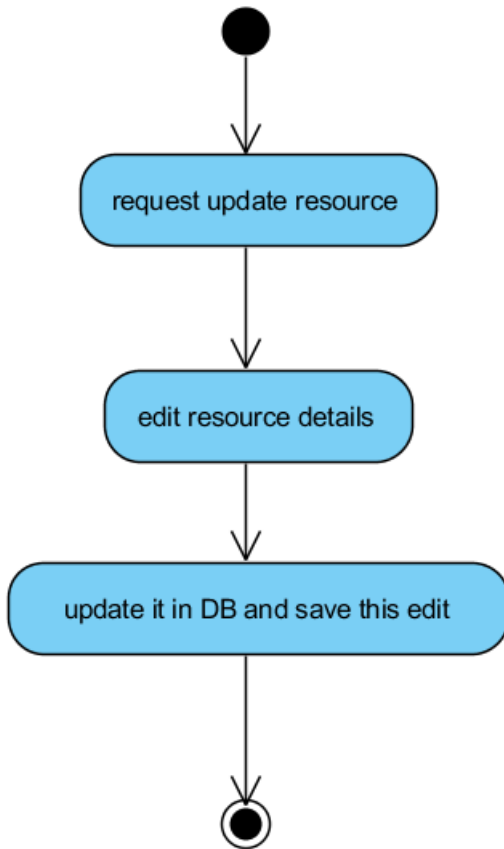| Info | Flow of Events | Details | Requirements | Diagrams | Test Plan | References | Description |

Flow of Events

1. enter the info of resource (place, name, phone, employee, employee email, employee phone, extra note)
2. save in DB

Preconditions:	1-pharmacist should have an account in this website 2-enter new resource invoice
Post-conditions:	1- update resource 2-delete resource

Update resource:

Activity diagram

```
        ●
        │
        ▼
┌──────────────────────┐
│ request update resource │
└──────────────────────┘
        │
        ▼
┌──────────────────┐
│ edit resource details │
└──────────────────┘
        │
        ▼
┌──────────────────────────────┐
│ update it in DB and save this edit │
└──────────────────────────────┘
        │
        ▼
        ◉
```

Delete resource:

Activity diagram

```
        ●
        │
        ▼
┌──────────────────────┐
│ request delete resource │
└──────────────────────┘
        │
        ▼
┌──────────────────────────┐
│ choose resource to delete it │
└──────────────────────────┘
        │
        ▼
┌──────────────────────────────┐
│ delete it in DB and save this edit │
└──────────────────────────────┘
        │
        ▼
        ◉
```
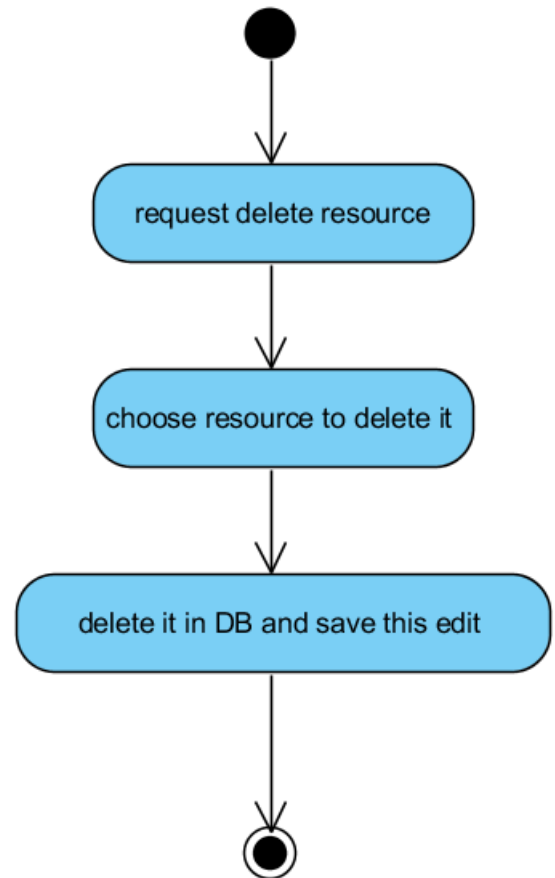
Description:

Name: update resource

| Info | Flow of Events | Details |

Flow of Events

1. open resource page
2. do edit in resource details
3. save edit in DB

Preconditions: insert resource

Name: delete resource

| Info | Flow of Events | De |

Flow of Events

1. choose resource
2. delete it from DB

Preconditions: insert resource

Enter new medicine: activity diagram
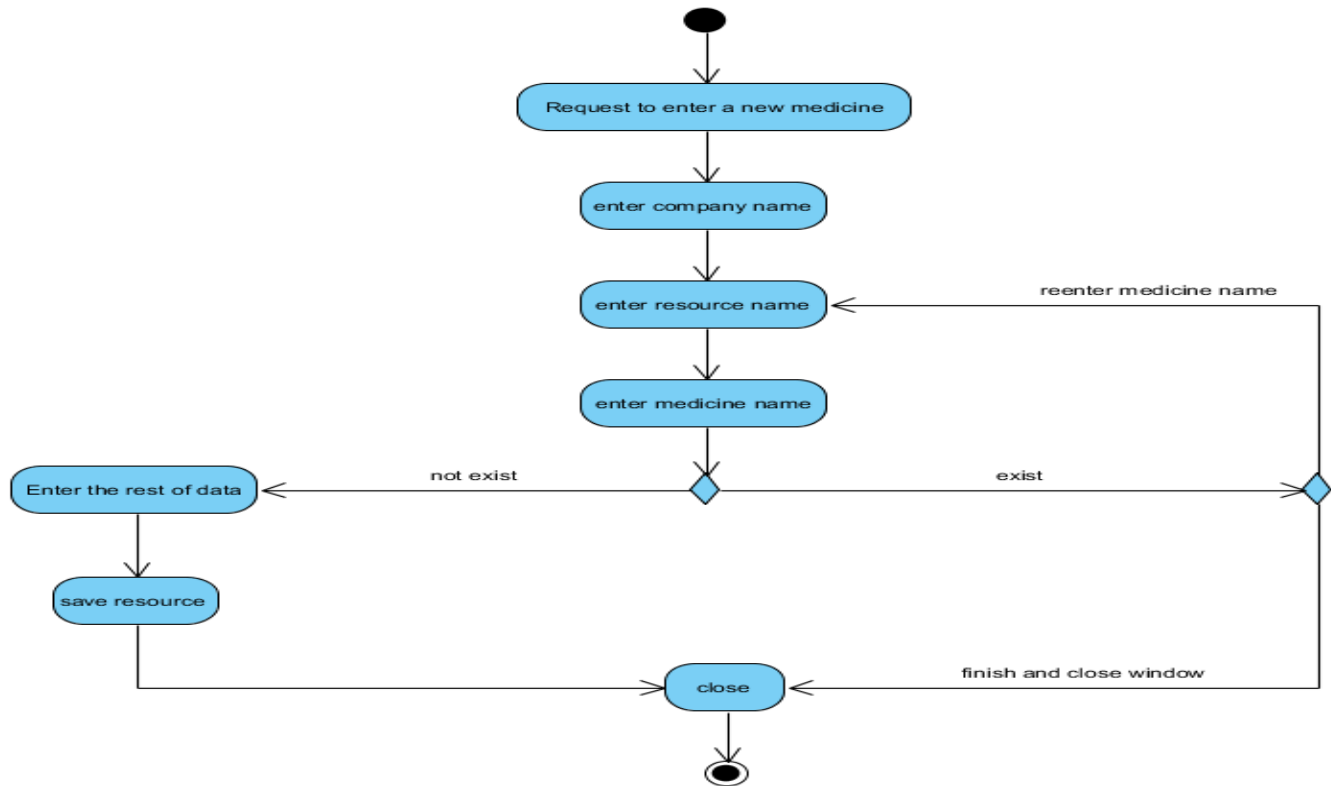
Enter new medicine: sequence diagram
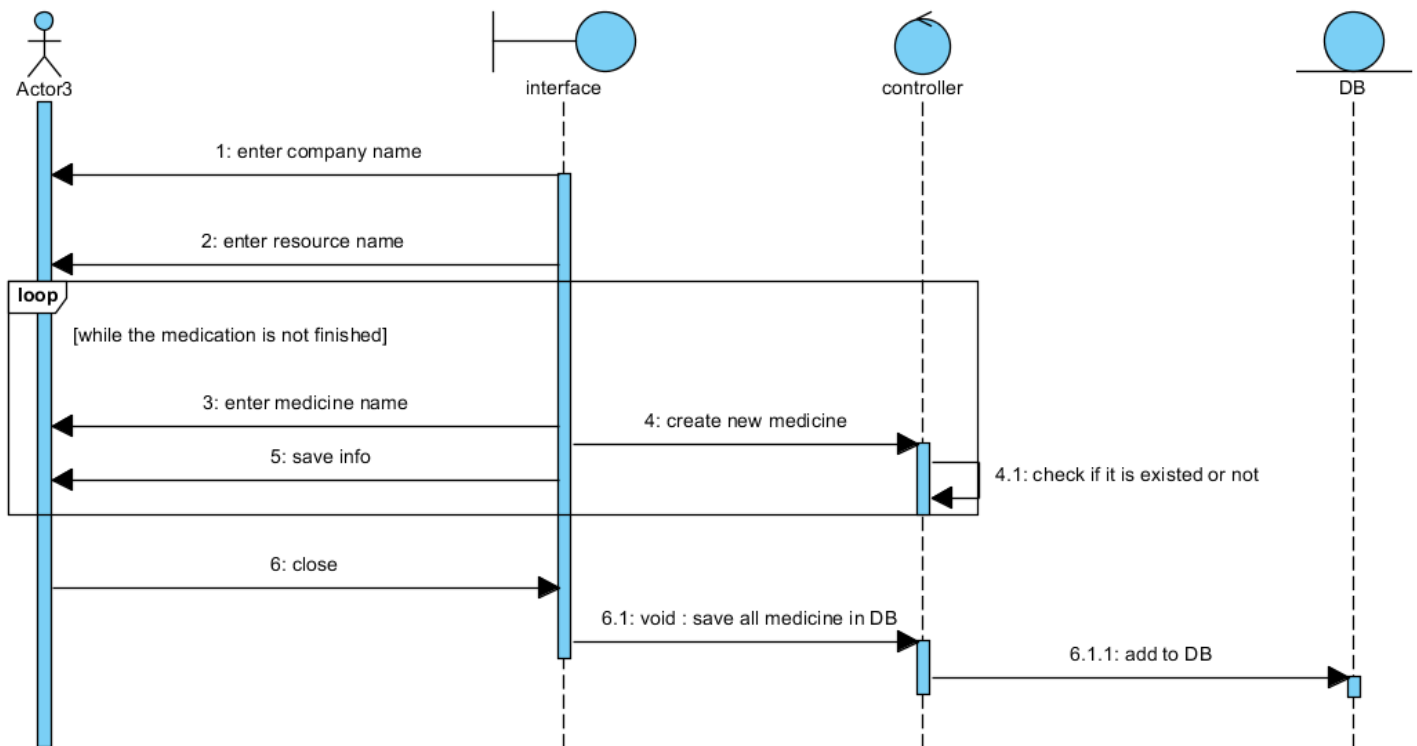
Description:

enter new medicine Details

Name: enter new medicine

Info	Flow of Events	Details	Requirements	Diagrams	Test Plan	References	Description

Rank: <Unspecified>

Justification: add new item to DB

Primary Actors:

Supporting Actors:

Documentation:

HTML   B *I* U ≡ ≡ ≡ ≡ ≡ ≡ F Fғ

```
to achieve the required we will do this steps:
you will enter medicine name and make sure if not exist early in DB,
if no then enter the company name and the resource name and save info in DB
```

---

enter new medicine Details

Name: enter new medicine

Info	Flow of Events	Details	Requiremen

Flow of Events ∨

1. enter medicine name
2. select if it must be pay with OTC or not
3. medicine form
4. company name

| Preconditions:	1-pharmacist should have an account in this website 2-insert new invoice
Post-conditions:	1-delete medicine 2-update medicine

Delete medicine: Update medicine:

Activity diagram   Activity diagram
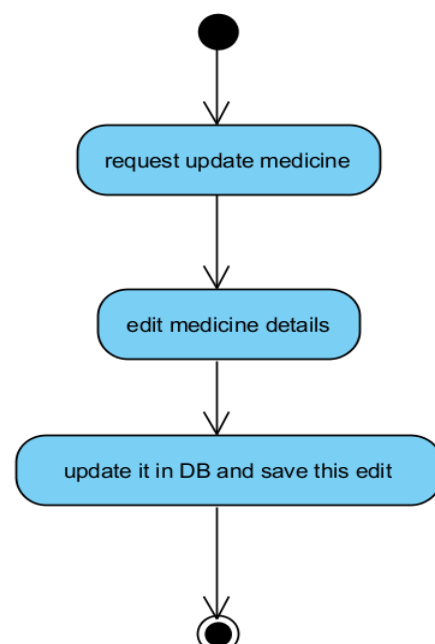
Description:

delete medicine Details

Name: delete medicine

Info | Flow of Events | Details

Flow of Events

1. choose medicine
2. delete from DB with its info

Preconditions: insert medicine

update medicine Details

Name: update medicine

Info | Flow of Events | Details

Flow of Events

1. open medicine page
2. do edit in medicine details
3. save edit in DB

Preconditions: insert medicine

Return to the resource: activity diagram
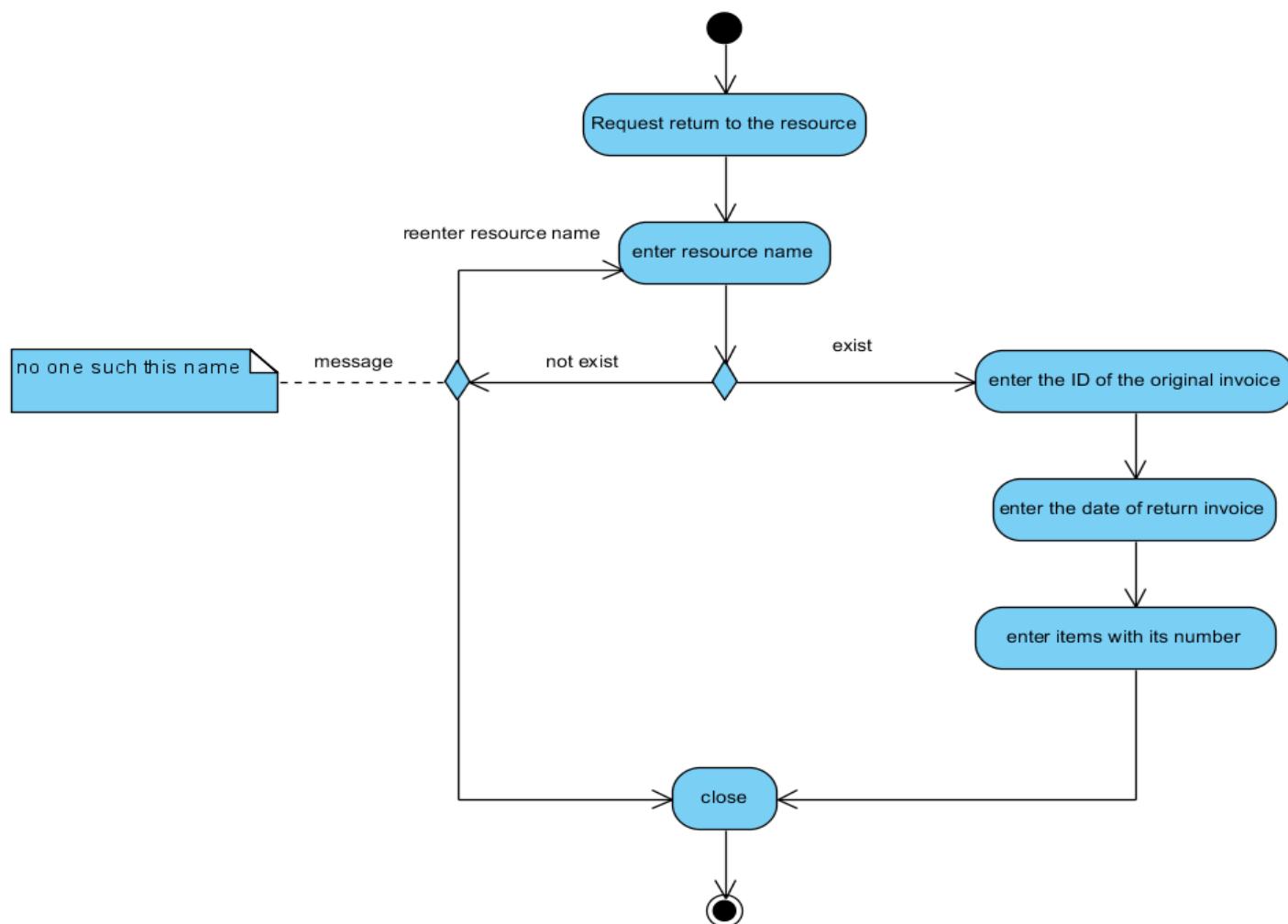
Return to the resource: sequence diagram
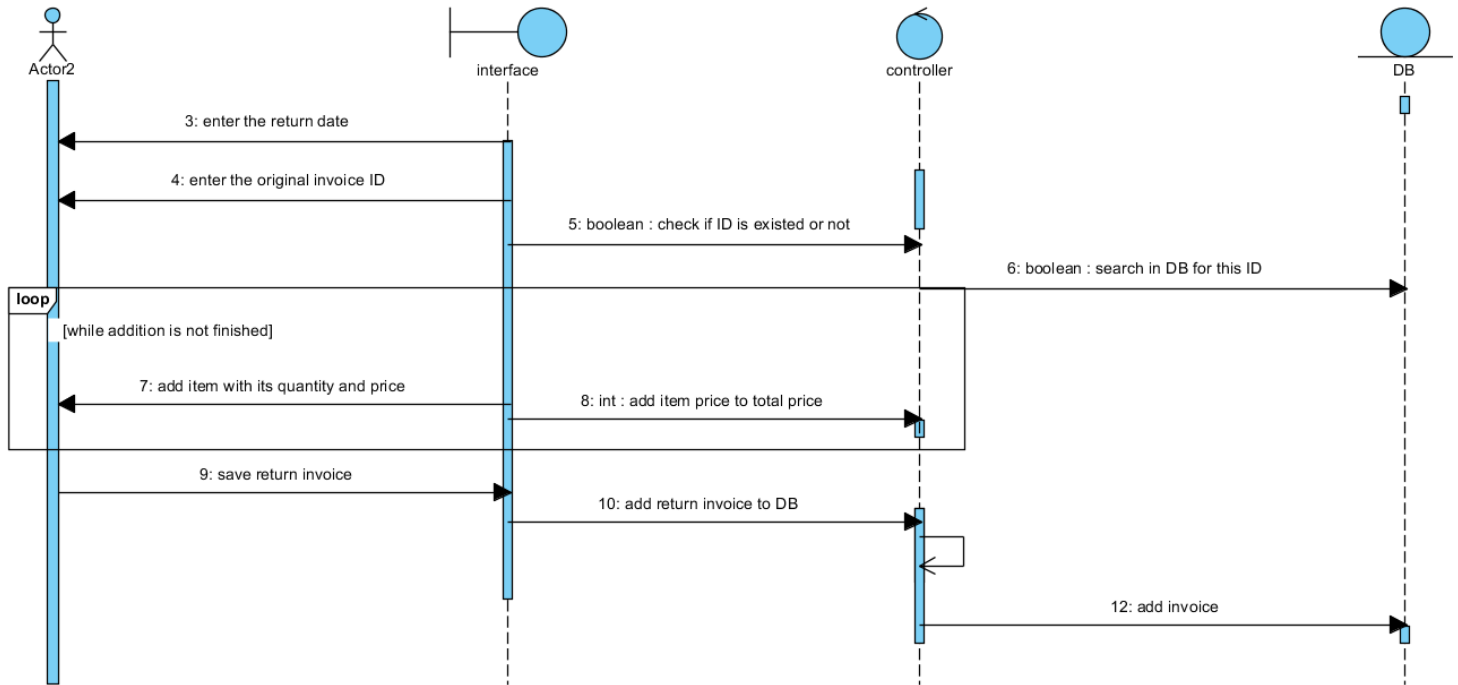
Description:

return to the resource Details

Name: return to the resource

Info | Flow of Events | Details | Requirements | Diagrams | Test Plan | References | Description

Rank: <Unspecified>

Justification: return invoice that took from resource

Primary Actors: pharmacist

Supporting Actors:

Documentation:

HTML | B I U ...

to achieve the required we will do this steps:
You will enter the return date and resource name and make sure that the resource exist in DB
if yes, you can start enter items with their quantity
when you have finished then save this return invoice in DB and do an edit on quantity items and the mean price in DB

return to the resource Details

Name: return to the resource

Info | Flow of Events | Details | Requirements | Diagrams | Test Plan | Refe

Flow of Events

1. create new return resource invoice connecting with original invoice by ID
2. select the return date and return items with there quantity
3. decrease total price from from original price

Preconditions:
1-pharmacist should have an account in this website
2-insert invoice before this invoice

View resource: available diagram

```
        ●
        │
        ▼
┌─────────────────────┐
│ Request view resource│
└─────────────────────┘
        │
        ▼
┌─────────────────────┐
│  get resource from DB│
└─────────────────────┘
        │
        ▼
┌─────────────────────┐
│ sort resource by alphabet│
└─────────────────────┘
        │
        ▼
┌─────────────────────┐
│ display result on interface│
└─────────────────────┘
        │
        ▼
        ◉
```
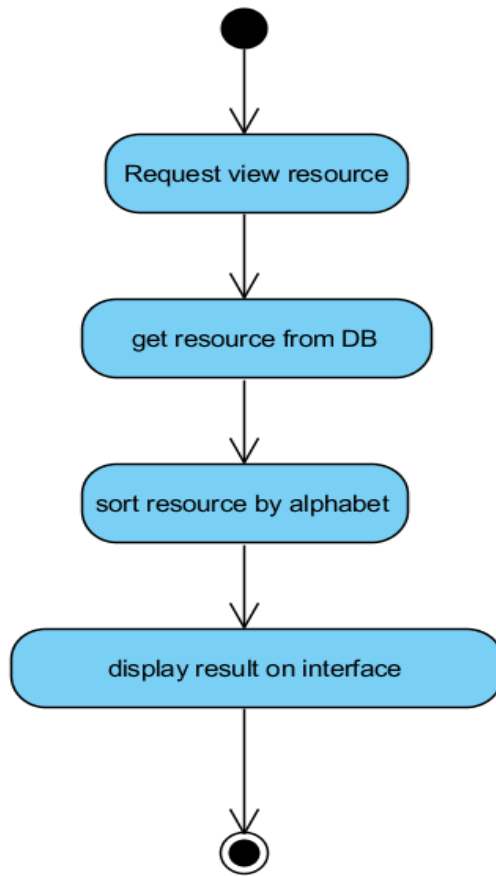
Description:

show resource Details

Name: show resource

| Info | Flow of Events | Details | Requirements |

Flow of Events

1. order to show the resource
2. then the project will get it from the DB
3. sort it by alphabet
4. display in interface

Name: show resource

| Info | Flow of Events | Details | Requirements | Diagrams | Test Plan | References | Description |

Rank: <Unspecified>

Justification: display the resource that dealing with pharmacy

Primary Actors: pharmacist

Supporting Actors:

Documentation:

HTML   B I U ≡ ≡ ≡ ≡ ≡ ≡ F FF

In this use case the pharmacist will order to show the resource .. then the project will get it from the DB and sort it by alphabet then display in interface

Class Diagram:

user
- -username : string
- -email : string
- -password : string
- -phonne : int

CustomerReturnInvoice
- -medicine : Medicine
- -count : int
- -returnedPrice

User
- -userName : string
- -email : string
- -password : string
- -phoneNumber : int

- +login()
- +registering()
- -cheackUser()
- +getUserName() : string
- +newSellingInvoice() : SellingInvoice
- +newCustomerReturnInvoice() : CustomerReturnInvoice
- +newResource()

Resource
- -name : string
- -location : string
- -phone : int

SellingInvoice
- -medicine : Medicine
- -count : int
- -price : float
- -date : Date

- +insert(dataBase : string, table : String)
- +deleate()

ResourceReturnInvoice
- -date : Date
- -medicine : Medicine
- -count : int
- -Price : float
- -user : User
- --

Medicine
- -name : string
- -OTC : boolean
- -company : string
- -farmacyform

- +insert(database : string, tableName : string)
- +updatename(name : string)
- +deleat()

Resource
- -name
- -location
- -phone : int
- -entryInvoices : EntryInvoice<ArrayList>
- -resourceReturnInvoices : ResourceReturnInvoice<ArrayList>

- +addEntryInv(ev : EntryInvoice)
- +addReturnInvoic(rv : ResourceReturnInvoice)

EntryInvoice
- -user : User
- -medicine : Medicine
- -pharmacyPrice : float
- -price : float
- -count : int
- -productionDate : Date
- -expirationDate : Date

ERD:

resource

id	integer(11)
name	varchar(255)
phone	integer(11)
location	varchar(255)

entry invoice

id	integer(11)
recourse_id	integer(11)
medicine_id	integer(11)
user_id	integer(11)
couunt	integer(10)
Purchasing price	float(10)
selling price	float(10)
Consumer price	integer(10)
Production Date	date
Expiration date	date
usersid	integer(11)
resourceid	integer(11)
midicineid	integer(11)

midicine

id	integer(11)
name	varchar(255)
company	varchar(255)
OTC	integer(2)
pharmacy_form	varchar(255)

resource return invoice

id	integer(11)
midicineid	integer(11)
resourceid	integer(11)
usersid	integer(11)
count	integer(11)
return_date	date
return_price	float(10)
resourceid2	integer(11)
usersid2	integer(11)
midicineid2	integer(11)

customer return invoice

id	integer(11)
midicineid	integer(11)
usersid	integer(11)
count	integer(11)
price	float(10)
midicineid2	integer(11)
usersid2	integer(11)

selling invoice

id	integer(11)
midicineid	integer(11)
usersid	integer(11)
count	integer(10)
price	float(10)
usersid2	integer(11)
midicineid2	integer(11)

users

id	integer(11)
full_name	varchar(255)
password	varchar(255)
email	varchar(255)
phone	integer(11)

gitHup: https://github.com/rafahmnjd/GRPharmacy