

## Use Case: Visit system i.2

Actors: System manager

Preconditions: The system is initialized

Postconditions:

Parameters: Payment/ Supply

Actions:

system manager send the parameter that he want to add/change/swap to the adjusted function in the system

the external service add/change/swap the wanted service by adjusted function

## Use Case: Visit system i.5.1

Actors: Shop owner

Preconditions: the owner is logged in and online

Postconditions:

Parameters: notification

Actions:

notification is created by making a purchase from an owner shop

notification is send to the related owner

## Use Case: Visit system i.5.2

Actors: Shop owner

Preconditions: the owner is logged in and online

Postconditions:

Parameters: notification

Actions:

notification is created by open and close of related shop

notification is send to the related owner

## Use Case: Visit system i.5.3

Actors: Shop owner

Preconditions: the owner is logged in and online

Postconditions:

**Parameters:** notification

**Actions:**

notification is created by deleting the owner subscription from an owner shop

notification is send to the related owner

#### **Use Case: Visit system i.5.4**

**Actors:** Subscribed User

**Preconditions:** the Subscribed User is logged in and online

**Postconditions:**

**Parameters:** notification

**Actions:**

notification is created when a message is sent to them

notification is send to the related user

#### **Use Case: Visit system i.6**

**Actors:** Shop owner

**Preconditions:** the Subscribed User is offline

**Postconditions:**

**Parameters:** notification

**Actions:**

notification is created and checks if Subscribed User is online

if not the notification will be saved and will be sent when the use will logged in

#### **Use Case: Visit system ii.1.1**

**Actors:** User

**Preconditions:** The system is initialized

**Postconditions:** The user is now a guest and can shop

**Parameters:**

**Actions:**

User visits the system

User gets the status of Guest and receives a shopping cart

### Use Case: Visit system ii.1.1

Actors: User

Preconditions: The system is initialized

Postconditions: The user is now a guest and can shop

Parameters:

Actions:

User visits the system

User gets the status of Guest and receives a shopping cart

### Use Case: Exit system ii.1.2

Actors: Guest

Preconditions: The system is initialized

Postconditions: The Guest is no longer a user of the system and can't perform any operations in it

Parameters:

Actions:

Guest exits the system

### Use Case: Register ii.1.3

Actors: Guest

Preconditions: The system is initialized and Guest is using it

Postconditions: if the registration succeeded, a new subscribed user is saved in the system

Parameters: user name, password

Actions:

Guest tries to register to the system using the details: user name, password

System saves a subscription with that information

Alternatives:

System detects that there is a user with such a user name and informs the user of the failure of the registration

### Use Case: Login ii.1.4

Actors: Guest

Preconditions: The system is initialized

**Postconditions:** if the login succeeded, the system identifies the **Guest** as a subscribed user

**Parameters:** user name, password

**Actions:**

**Guest** tries to login with the details: user name, password

**Guest** is now identified as a subscribed user

**Alternatives:**

System detects can't find a subscribed user matching the given information and informs the user about the failure of logging in

### **Use Case: search in shopping cart ii.2.4.1**

**Actors:** user

**Preconditions:** The system is initialized

**Parameters:**

**Actions:**

**User** select to inspect the shopping cart

**Guest** is now can see a list of all the **products** that in the **shopping cart** and the quantity and total value of each **product**.

### **Use Case: shopping cart Management ii.2.4.2**

**Actors:** User

**Preconditions:**

**Parameters:** Product

**Actions:**

**User** opens the **Shopping cart**.

System presents 2 options: "**Remove Product**", "**Edit Product quantity**".

If **User** selects "**Remove Product**":

- the **Product** is removing from the shopping cart
- the **Shop** associate with the **product** stop hold it for the customer

If **User** selects "**Edit Product**": Continue in Use Case "**Edit Product quantity**".

- the **User** is enter the new quantity
- if the product in the shop available in this new quantity :
  - the **Shop** associate with the **product** hold it for the customer
  - the Product quantity change to the new quantity.
- alternative the action faild.

## Use Case: **Storage Management ii.4.1**

**Actors:** Owner

**Preconditions:** Owner is logged in and has a Shop.

**Parameters:** Product, Shop

**Actions:**

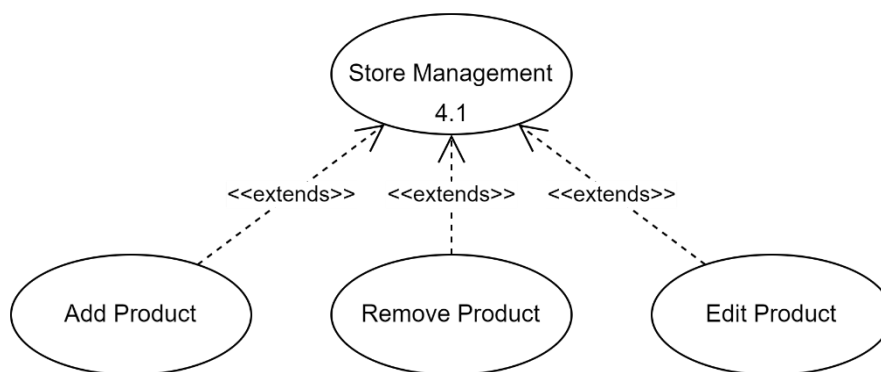
Owner opens Shop's storage.

System presents 3 options: "Add Product", "Remove Product", "Edit Product".

If Owner selects "Add Product": Continue in Use Case "Add Product".

If Owner selects "Remove Product": Continue in Use Case "Remove Product".

If Owner selects "Edit Product": Continue in Use Case "Edit Product".



**Use Case:** Add Product

**Actors:** Owner

**Preconditions:** Owner is logged in and has a Shop with an option to add products.

Shop has a product collection.

Use Case **Storage Management ii.4.1** happened.

**Parameters:** Product, Shop

**Actions:**

System presents an option to enter data of **Product**.

**Owner** enters data of **Product p**.

**Owner** informs the System that he entered all the information.

System adds **Product p** to the **Shop s** product collection.

System informs the **Owner** he added the **Product p**.

**Use Case:** Remove Product

**Actors:** **Owner**

**Preconditions:** **Owner** is logged in and has a **Shop** with an option to remove product.

**Shop** has a product collection.

**Parameters:** **Product**, **Shop**

**Actions:**

System presents a general product collection to search.

**Owner** selects **Product p**.

System removes **Product p** to the **Shop** product collection.

System informs the **Owner** he removed the **Product p**.

**Use Case:** Edit Product

**Actors:** **Owner**

**Preconditions:** **Owner** is logged in and has a **Shop** with an option to edit product.

**Shop** has a product collection.

**Parameters:** **Product**, **Shop**

**Actions:**

System presents a general product collection to search.

**Owner** selects **Product p**.

System shows **Product p** information to the **Owner**.

**Owner** edits information of **Product p**.

**Owner** informs the System he finished editing **Product p**.

System updates **Product p** and informs **Owner** the product has been edited.

## Use Case: Discount Management ii.4.2

**Actors:** Owner, Discounts

**Preconditions:** Owner is logged in and has Discounts in the Shop he owns.

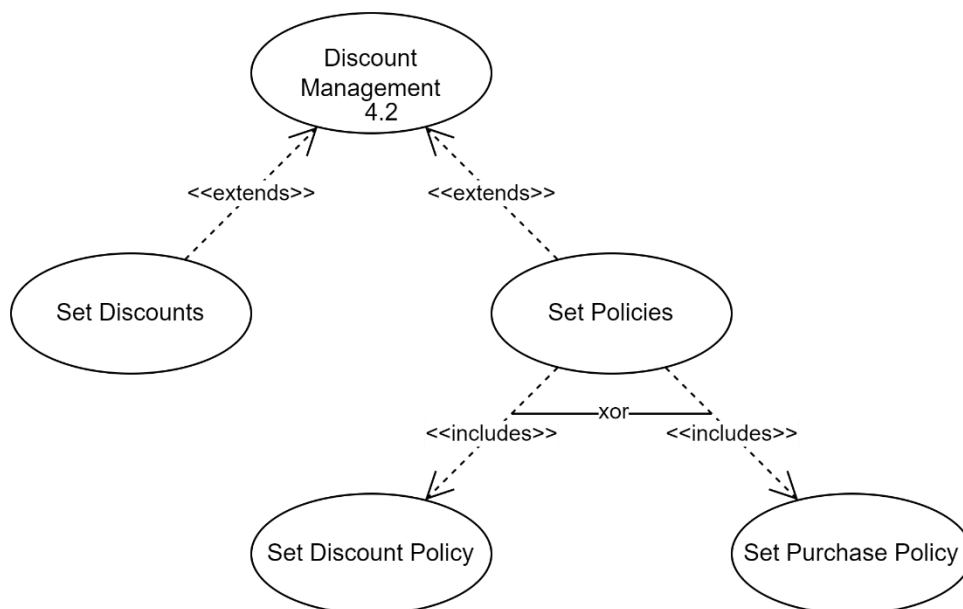
**Parameters:** Discount, Policy

**Actions:**

System presents 2 options: "Set Discounts", "Set Policies".

If Owner selects "Set Discounts": Continue in Use Case "Set Discounts".

If Owner selects "Set Policies": Continue in Use Case "Set Policies".



**Use Case:** Set Discounts

**Actors:** Owner, Purchase Discounts, Discount Management ii.4.2

**Preconditions:** Owner is logged in and has Discounts and an option to set discounts.

Shop has a discount collection.

**Parameters:** Discount

**Actions:**

System presents a general Purchase Discounts collection to search.

Owner selects Discount d.

System shows Discount d information to the Owner.

Owner sets Discount d.

**Owner** informs System he finished setting **Discount d**.

System updates **Discount d** and informs **Owner** the discount has been set.

**Use Case:** Set Policies

**Actors:** **Owner**, **Policy**, **Discount Management ii.4.2**

**Preconditions:** **Owner** is logged in and has **Discount Policy** and an option to set discount policies.

**Parameters:** **Discount**

**Actions:**

System presents 2 options: “**Set Discount Policy**”, “**Set Purchase Policy**”.

If **Owner** selects “**Set Discount Policy**”: Continue in Use Case “**Set Discount Policy**”.

If **Owner** selects “**Set Purchase Policy**”: Continue in Use Case “**Set Purchase Policy**”.

**Use Case:** Set Discount Policy

**Actors:** **Owner**, **Discount Policies**, **Shop Management ii.4.1**

**Preconditions:** **Owner** is logged in and has a **Shop** he manages with an option to set discount policy.

**Shop** has a **Discount Policies** collection.

**Parameters:** **Discount Policy**

**Actions:**

System presents a general **Discount Policies** collection to search.

**Owner** selects **Discount Policy dp**.

System shows **Discount Policy dp** information to the **Owner**.

**Owner** sets information of **Discount Policy dp**.

**Owner** informs the System he finished editing the **Discount Policy dp**.

System updates **Discount Policy dp** and informs **Owner** the discount policy has been edited.

**Use Case:** Set Purchase Policy

**Actors:** **Owner**, **Purchase Policies**, **Shop Management ii.4.1**

**Preconditions:** **Owner** is logged in and has a **Shop** he manages with an option to set purchase policy.

**Shop** has a **Purchase Policies** collection.

**Parameters:** **Purchase Policy**



#### Actions:

System presents a general **Purchase Policies** collection to search.

**Owner** selects **Purchase Policy pp**.

System shows **Purchase Policy pp** information to the **Owner**.

**Owner** sets information of **Purchase Policy pp**.

**Owner** informs the System he finished editing the **Purchase Policy pp**.

System updates **Purchase Policy pp** and informs **Owner** the purchase policy has been edited.

### Use Case: **Appoint Owner ii.4.4**

**Actors:** **Owner**, **Shop**

**Preconditions:** **Owner** is logged in and **Shop** has an option to appoint **Owner**.

**Subscribed User m** is not a **Owner** or **Owner** of this **Shop**.

**Parameters:** **Subscribed User**

#### Actions:

**Owner** opens **Shop** and selects option to appoint **Owner**.

**Owner** enters information of **Subscribed User m** to the System.

System appoints **Owner** to the Shop and gives him his **Owner** permissions.

### Use Case: **Appoint Manager ii.4.6**

**Actors:** **Owner**, **Shop**

**Preconditions:** **Owner** is logged in and **Shop** has an option to appoint **Manager**.

**Subscribed User m** is not a **Manager/Owner** of this **Shop**.

**Parameters:** **Subscribed User**

#### Actions:

**Owner** opens the **Shop** and selects the option to appoint a **Manager**.

**Owner** enters information of **Subscribed User m** to the System.

System appoints **Subscribed User m** as **Manager** to the Shop and gives him his **Manager** permissions.

### Use Case: **Edit Manager Permissions ii.4.7**

**Actors:** Owner, Shop

**Preconditions:** Owner is logged in and Shop has an option to edit manager permissions.

Shop has a Manager collection of the managers he appointed.

**Parameters:** Manager

**Actions:**

Owner opens the Shop and selects the option to edit manager permissions.

Owner selects Manager m from the Manager collection.

System shows Manager m's permissions as a Manager.

Owner edits Manager m's permissions.

System gives the edited permissions to Manager m.

### **Use Case: Close Shop ii.4.9**

**Actors:** Owner, Shop

**Preconditions:** Owner is logged in and is the founder of the Shop.

Shop has an option to close it.

**Parameters:** None

**Actions:**

Owner opens the Shop and selects the option to close it.

System changes the Shop's mode to inactive.

System changes the Shop view permissions to System Manager and Owner only.

System sends notification to Owners and Managers of the Shop, with a message that the Shop closed, using Use Case Real Time Notification 1.1

### **Use Case: Request Role Information ii.4.11**

**Actors:** Owner, Shop

**Preconditions:** Owner is logged in and owns the Shop.

Shop has an option to request roles information, an official collection and a Manager permissions collection.

**Parameters:** Manager

**Actions:**

**Owner** opens the **Shop** and selects the option to request roles.

System shows an official collection.

if Owner selects **Manager m**: System shows **Manager m**'s permissions.

### **Use Case: Receive Purchase History ii.4.13**

**Actors:** **Owner**, **Shop**

**Preconditions:** **Owner** is logged in and owns the **Shop**.

**Shop** has an option to show purchase history and a purchase history collection.

**Parameters:** **Manager**

**Actions:**

**Owner** opens the **Shop** and selects the option to show purchase history.

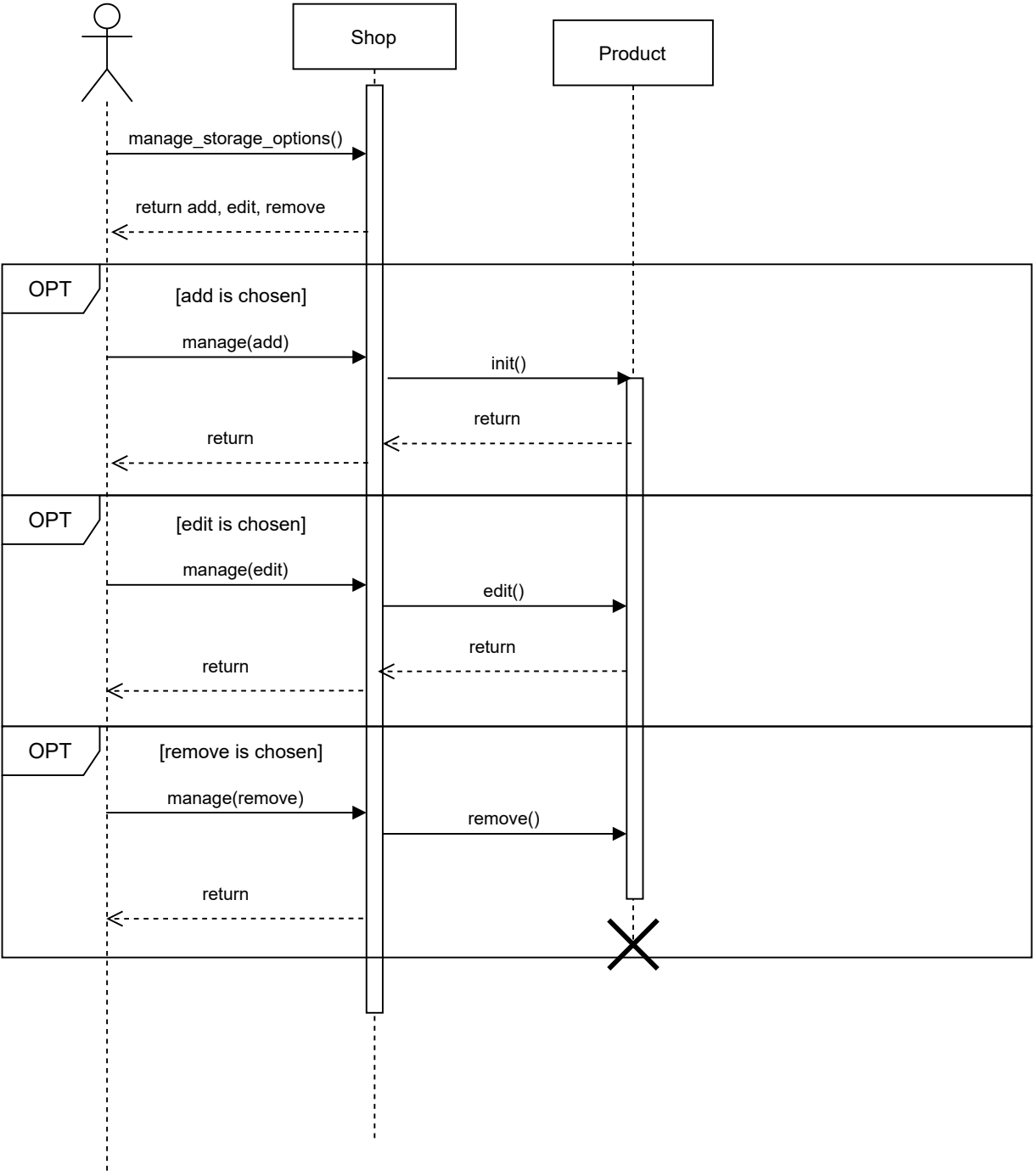
System shows a purchase history collection of all the purchases in the **Shop**.

**Owner can activate the use cases of the Owner if given permission by said Owner.**

Owner(can also be Manager  
if premission allows)

**pre cond:** the system after init and its open, the owner is logged in and has product p

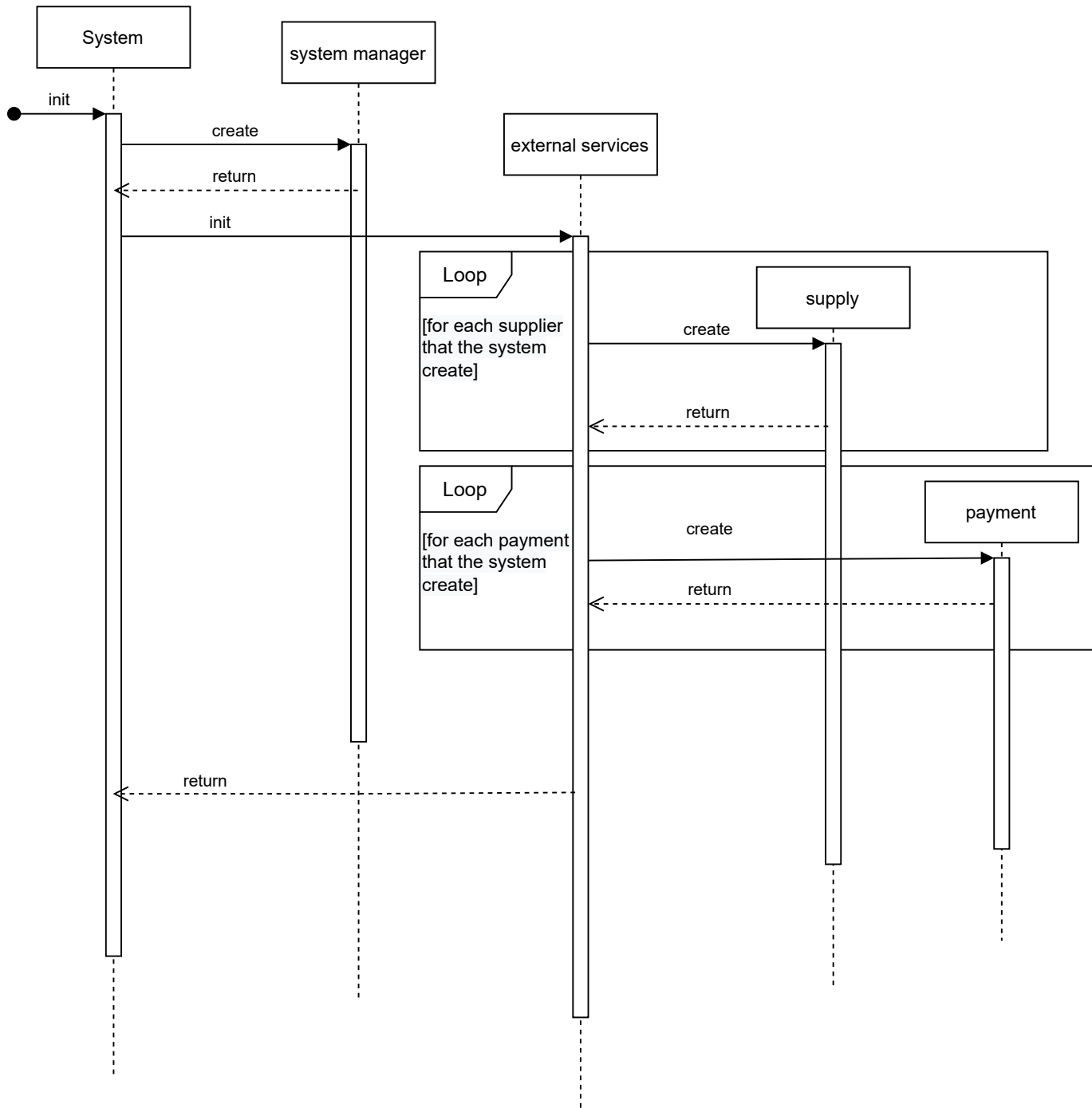
Use Case  
4.1



# Use Case

i.1

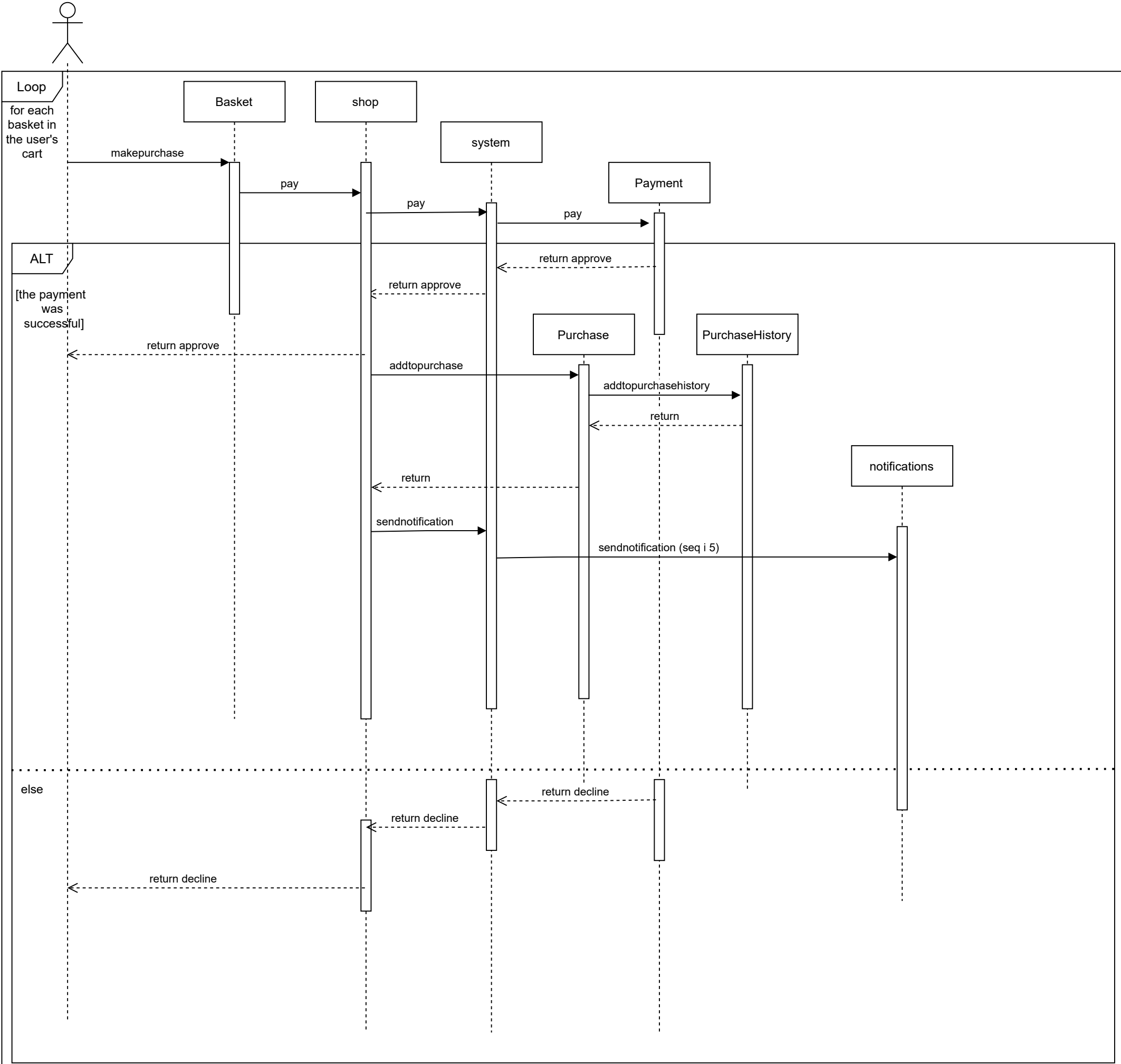
pre cond: the system will have the needed data to create the system manager and all the external services (at least one for each) at the start of the init



precondition: we assume that all the items at the basket are in the supply

Use Case  
i.3

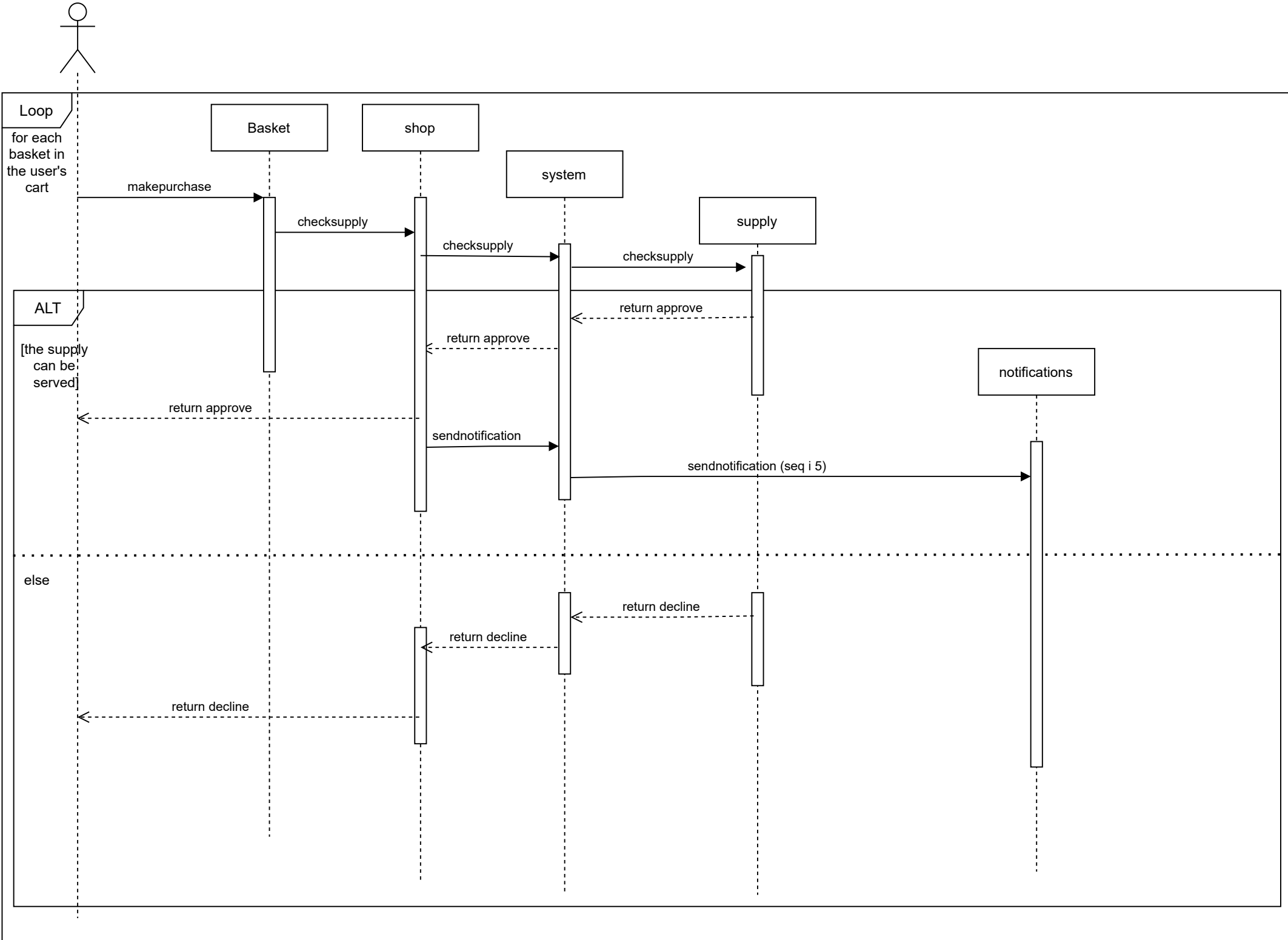
user (guest or member)



precondition: we assume that all the items at the user can afford the payment

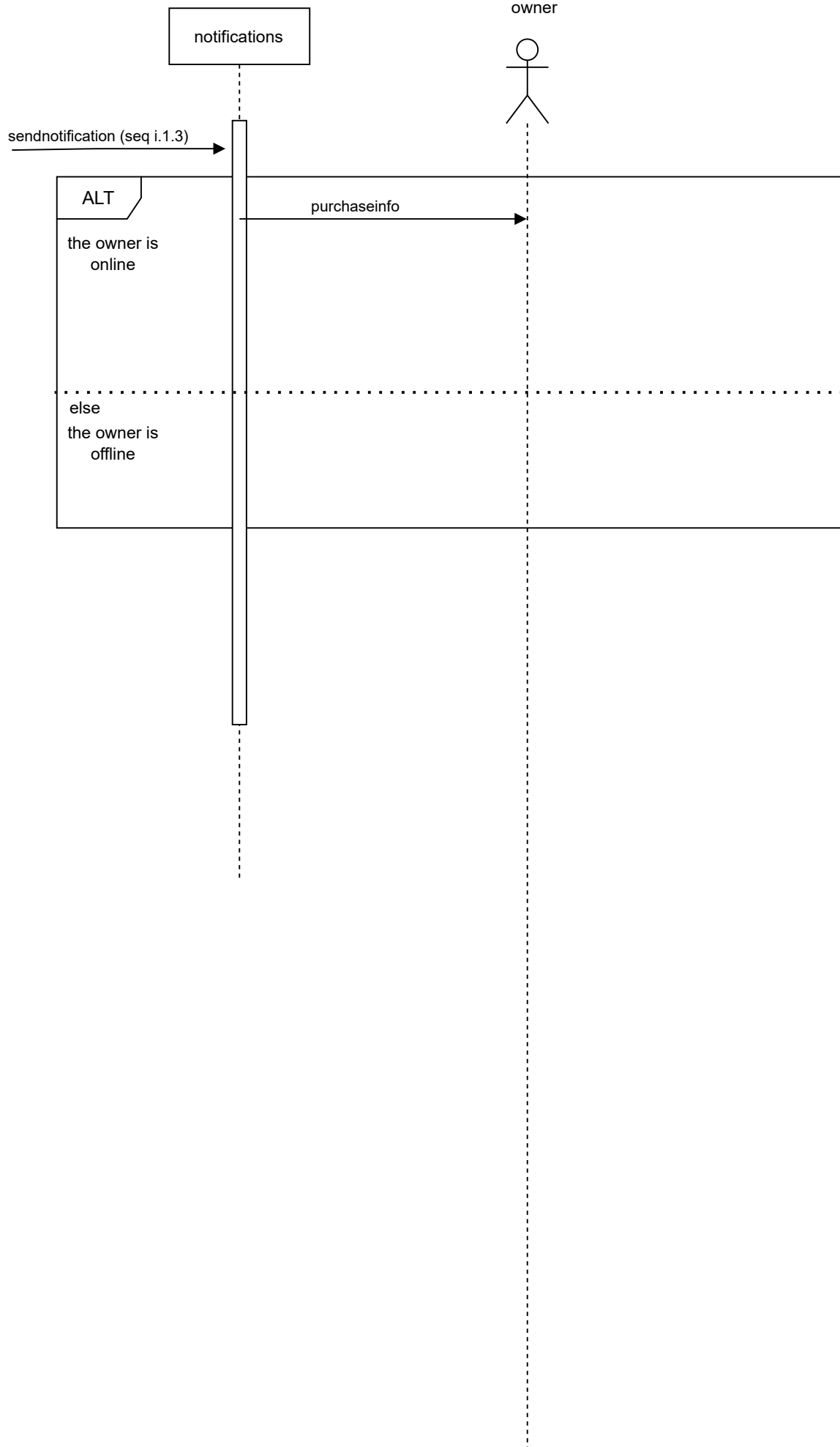
Use Case i.4

user (guest or member)



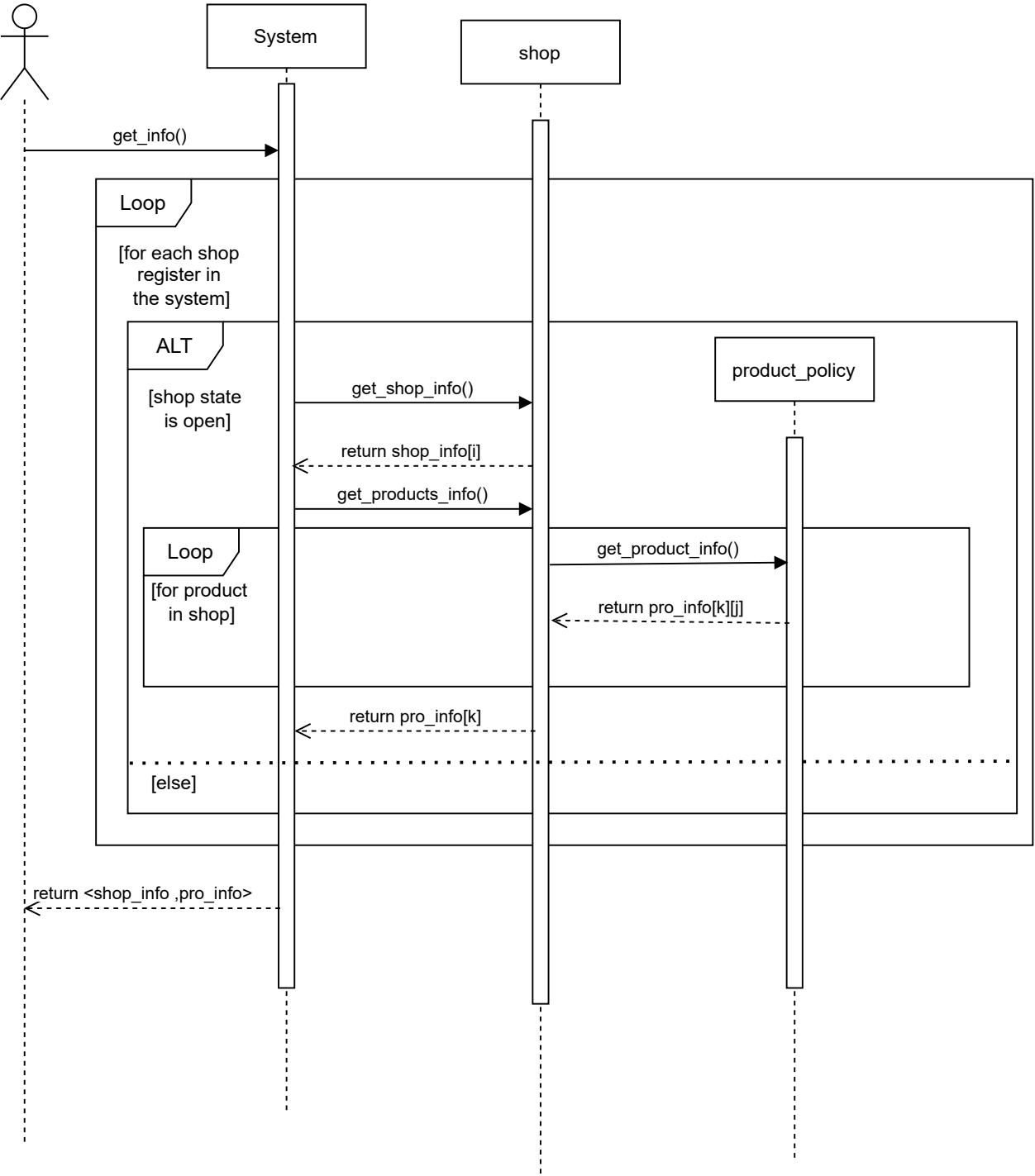
first case: owner get notification when something is bought from his shop

pre cond: the payment was successful and this part is the continue of sequence i.1.3





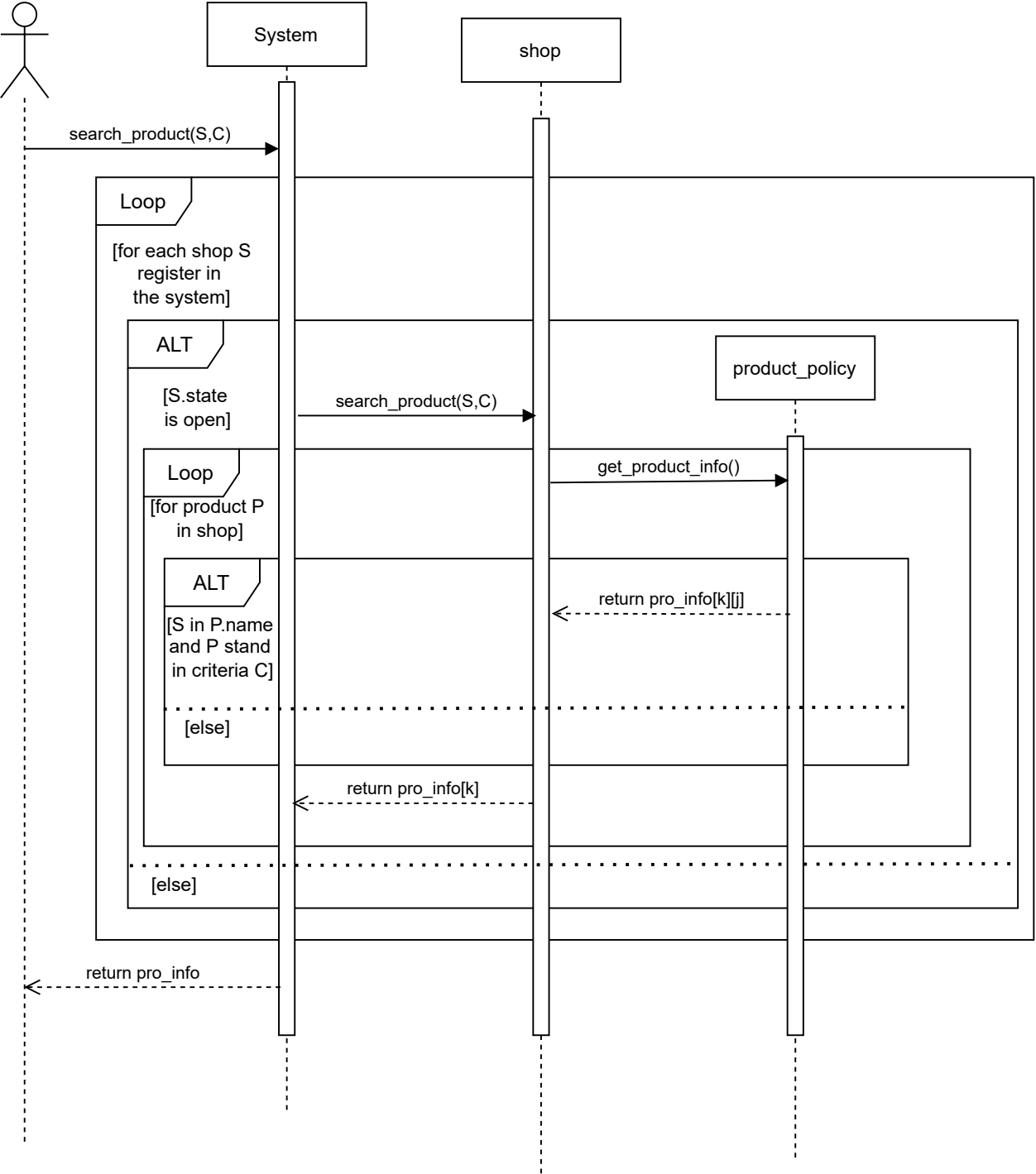
user (guest or member)



pre cond: the system after init and its open, the user has a search word S and a criteria C

user (guest or member)

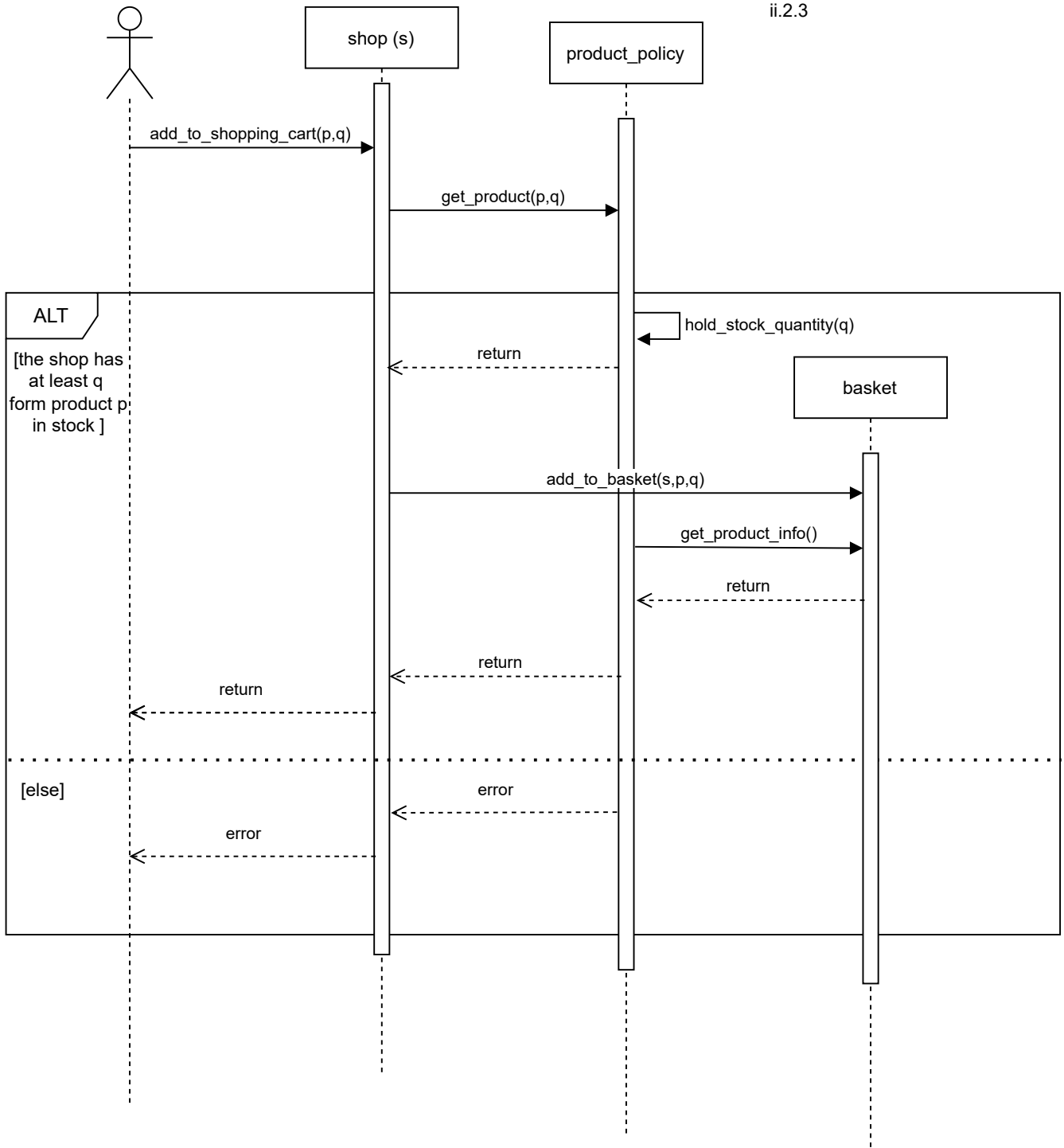
Use Case  
ii.2.2



**pre cond:** the system after init and its open , the user has a **product p** that exist in **shop s** and a **quantity q**

user (guest or member)

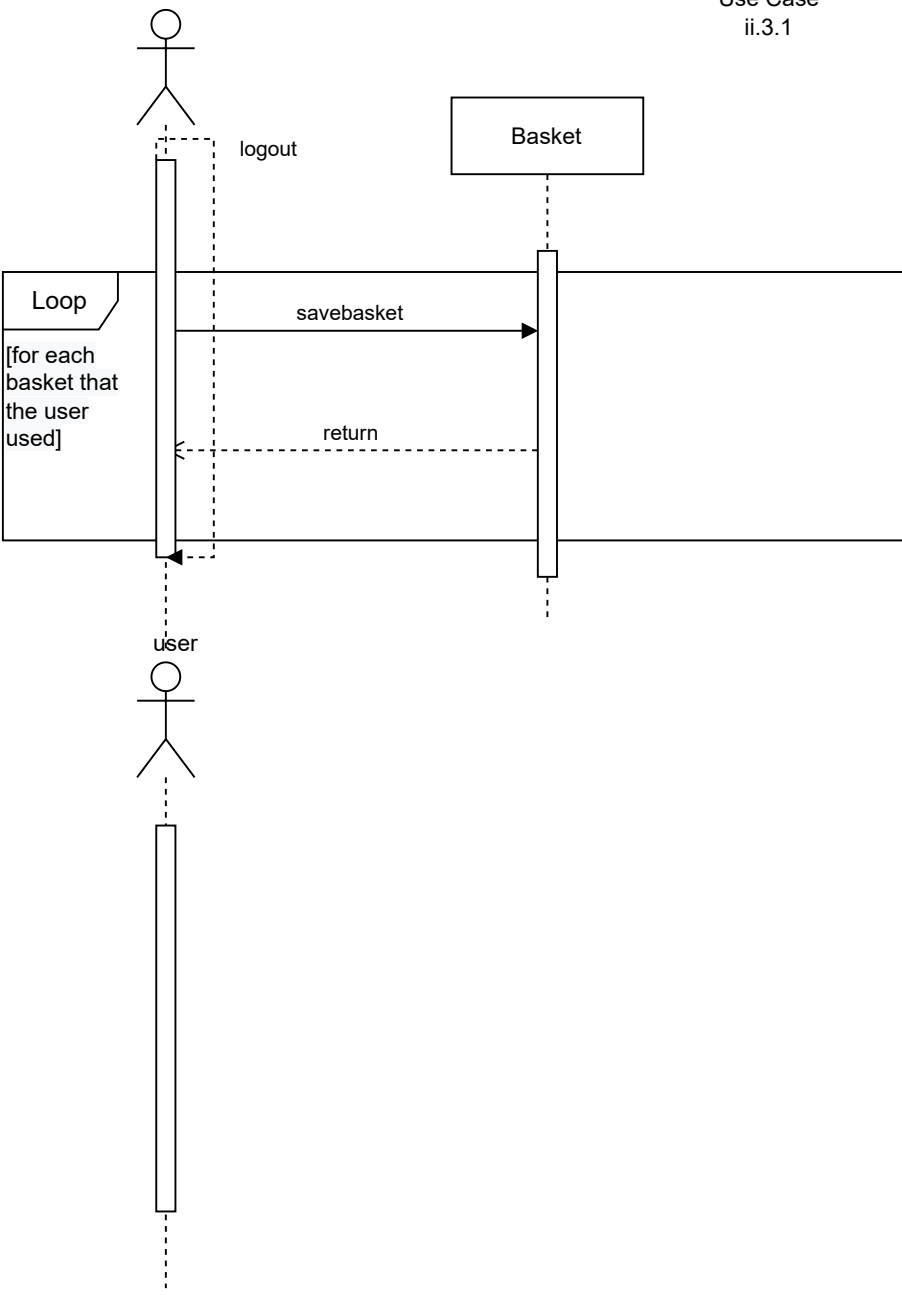
Use Case  
ii.2.3



```
pre condition: the user is successfully logged in to the system as subscribedUser
```

subscribe user

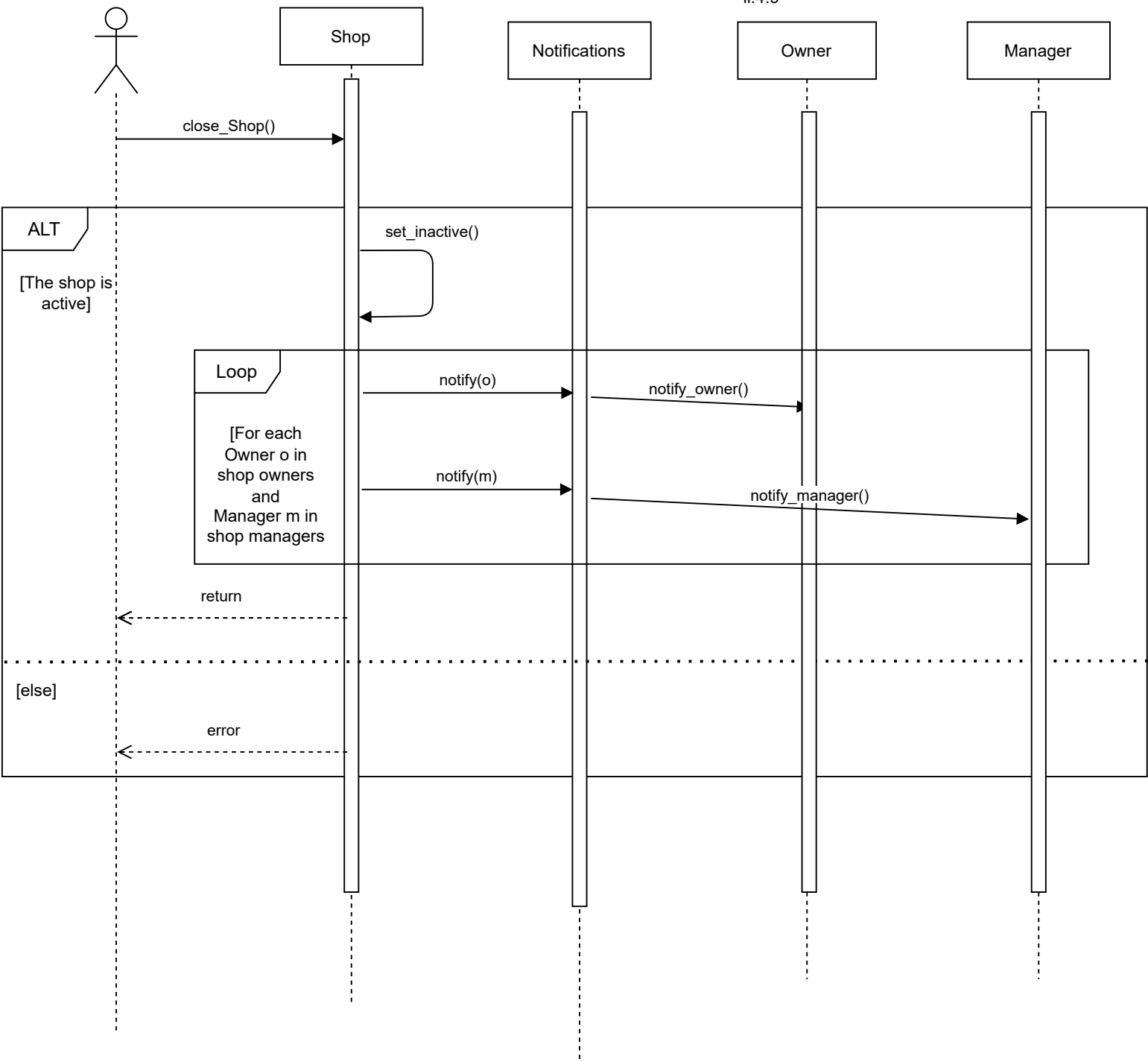
### Use Case ii.3.1



pre cond: the system after init and its open, the owner is logged in

Use Case  
ii.4.9

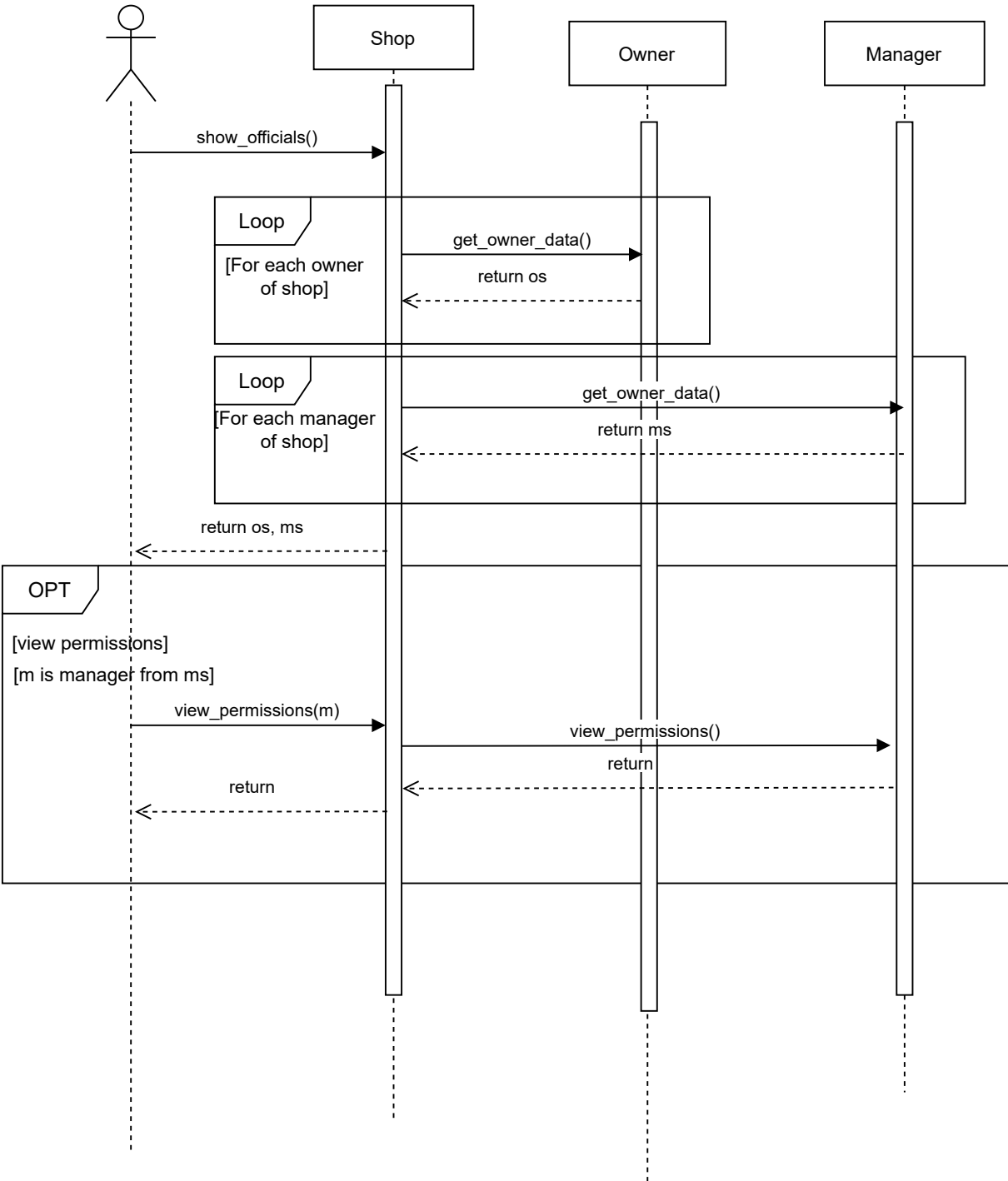
Owner(can also be Manager  
if premission allows)



**pre cond:** the system after init and its open, the owner is logged in

Use Case  
ii.4.11

Owner(can also be Manager  
if premission allows)

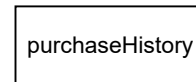
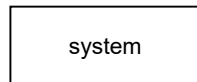
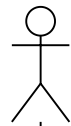


get history of shops

precondition: we assume that all the items at the basket are in the supply

system manager

Use Case  
ii.6.4.1



getallhistory

getshophistory

getshophistory

return

return

return

getspecificshophistory

getshophistory

getshophistory

return

return

return

ALT

get the  
history of  
all the  
shops

Loop

[for each  
shop in  
the system]

Loop

[for each  
purchase in  
the shop]

Loop

[for each  
purchase in  
the shop]

get the  
history of  
a specific  
shop

get history of user

precondition: we assume that all the items at the basket are in the supply

system manager

Use Case  
ii.6.4.2

