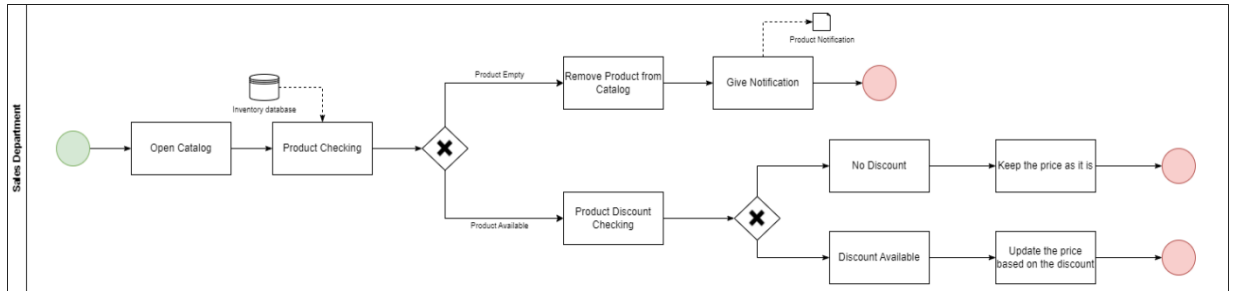


## Sales Management

### 1. Product Management

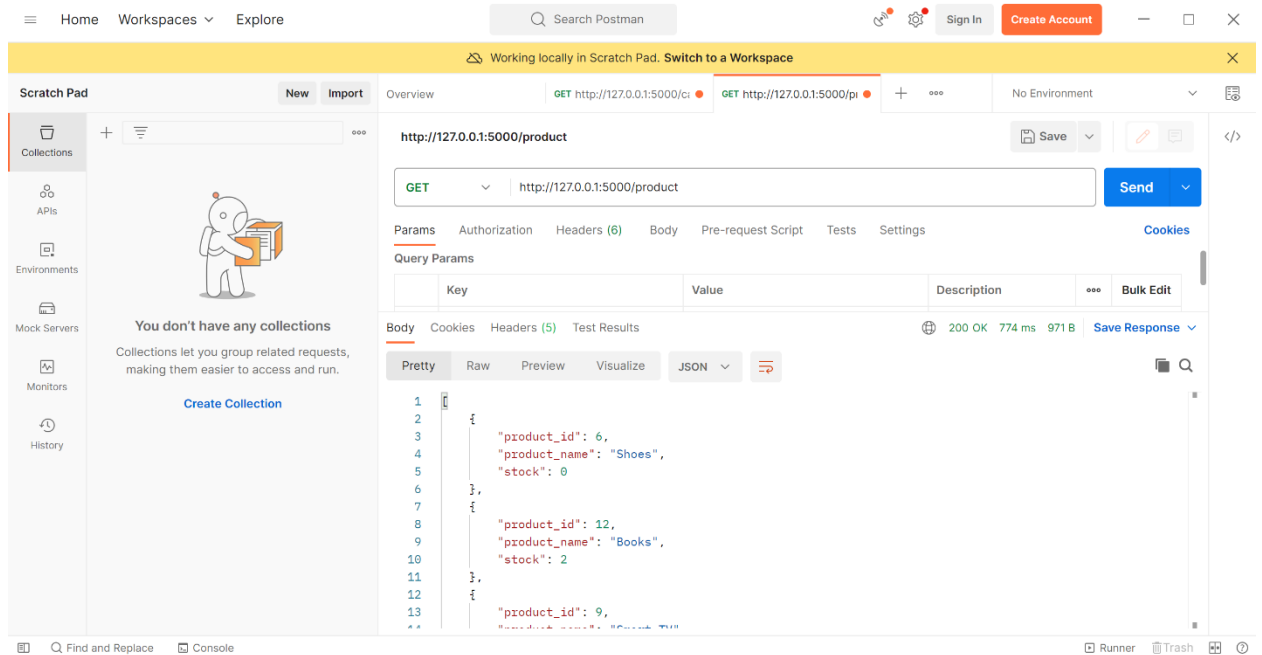


- Open catalog:

```
Welcome to ecommerce
1. Sales
2. Inventory
3. Employee
Enter your choice: 1
Display:
    1. Catalog
    2. Discount
    3. Booking List
    4. Notice
    0. Exit
Welcome to Sales: 1
    1. Display Catalog
    2. Add Catalog
    3. Update Catalog
    4. Delete Catalog
    0. Exit
Welcome to Catalog: █
```

```
1. Display Catalog
2. Add Catalog
3. Update Catalog
4. Delete Catalog
0. Exit
Welcome to Catalog: 1
product_id  product_name  price detail
0           1      Shoes    150000.0    -
1           2      Mobile Phone 3000000.0    -
2           3      Laptop    8000000.0    -
3           4      Headphones 500000.0     -
4           5      Watch    1000000.0    -
5           6      Shoes    750000.0     -
6           7      Handbag   900000.0     -
7           8      Camera   2000000.0    -
8           9      Smart TV  5000000.0    -
9          10      Gaming Console 4000000.0    -
10          11      Sunglasses 350000.0     -
11          12      Books    100000.0     -
12          13      Jewelry  1500000.0    -
13          14      Perfume   250000.0     -
14          15      Cosmetics 500000.0     -
```

- Product checking using API:



Shoes are empty

- Remove product from catalog:

```

1. Display Catalog
2. Add Catalog
3. Update Catalog
4. Delete Catalog

0. Exit
Welcome to Catalog: 4
Product ID: 6
Product deleted successfully

```

```

1. Display Catalog
2. Add Catalog
3. Update Catalog
4. Delete Catalog

0. Exit
Welcome to Catalog: 1

```

product_id	product_name	price	detail
0	1 Shoes	150000.0	-
1	2 Mobile Phone	3000000.0	-
2	3 Laptop	8000000.0	-
3	4 Headphones	500000.0	-
4	5 Watch	1000000.0	-
5	7 Handbag	900000.0	-
6	8 Camera	2000000.0	-
7	9 Smart TV	5000000.0	-
8	10 Gaming Console	4000000.0	-
9	11 Sunglasses	350000.0	-
10	12 Books	100000.0	-
11	13 Jewelry	1500000.0	-
12	14 Perfume	250000.0	-
13	15 Cosmetics	500000.0	-

- Give notification:

```
Display:
    1. Catalog
    2. Discount
    3. Booking List
    4. Notice

    0. Exit
Welcome to Sales: 4
Please input the product id: 6
What's the problem? Product empty
Notification added successfully
```

- Product discount checking:

```
Display:
    1. Catalog
    2. Discount
    3. Booking List
    4. Notice

    0. Exit
Welcome to Sales: 2

    1. Display Dicount
    2. Add Dicount
    3. Apply Discount

    0. Exit
Welcome to Discount: 1
  product_id  discount
0           1      0.30
1           7      0.20
2          12      0.35
```

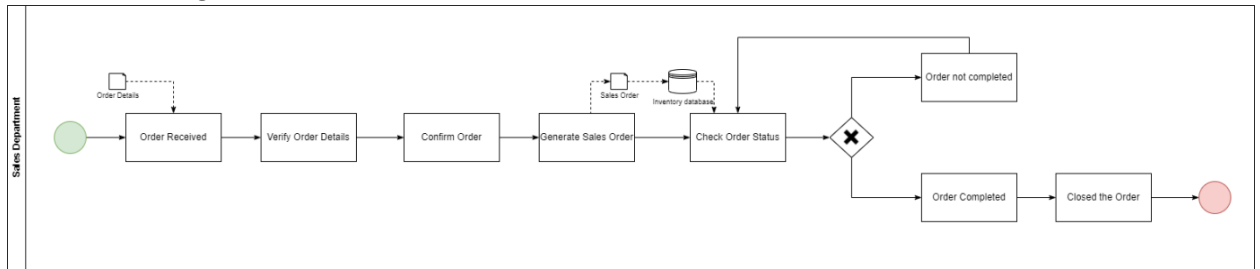
- Update the price based on the discount:

```
    1. Display Dicount
    2. Add Dicount
    3. Apply Discount

    0. Exit
Welcome to Discount: 3
Input product ID: 7
Discount applied successfully. The new price is: 720000.00
```

	product_id	product_name	price	detail
0	1	Shoes	150000.0	-
1	2	Mobile Phone	3000000.0	-
2	3	Laptop	8000000.0	-
3	4	Headphones	500000.0	-
4	5	Watch	1000000.0	-
5	7	Handbag	720000.0	discount applied
6	8	Camera	2000000.0	-
7	9	Smart TV	5000000.0	-
8	10	Gaming Console	4000000.0	-
9	11	Sunglasses	350000.0	-
10	12	Books	100000.0	-
11	13	Jewelry	1500000.0	-
12	14	Perfume	250000.0	-
13	15	Cosmetics	500000.0	-

## 2. Order Processing



- Check order in bookinglist table:

```
1. Display Booking List
2. Confirmed Customer Order
3. Update Booking Status

0. Exit
Welcome to Booking List: 1
 booking_id customer_id customer_name products total_price customer_address customer_order booking_status
0          1           1         Anon Books (1)    750000.0      Bandung      On Progress  On Progress
```

- Confirm order:

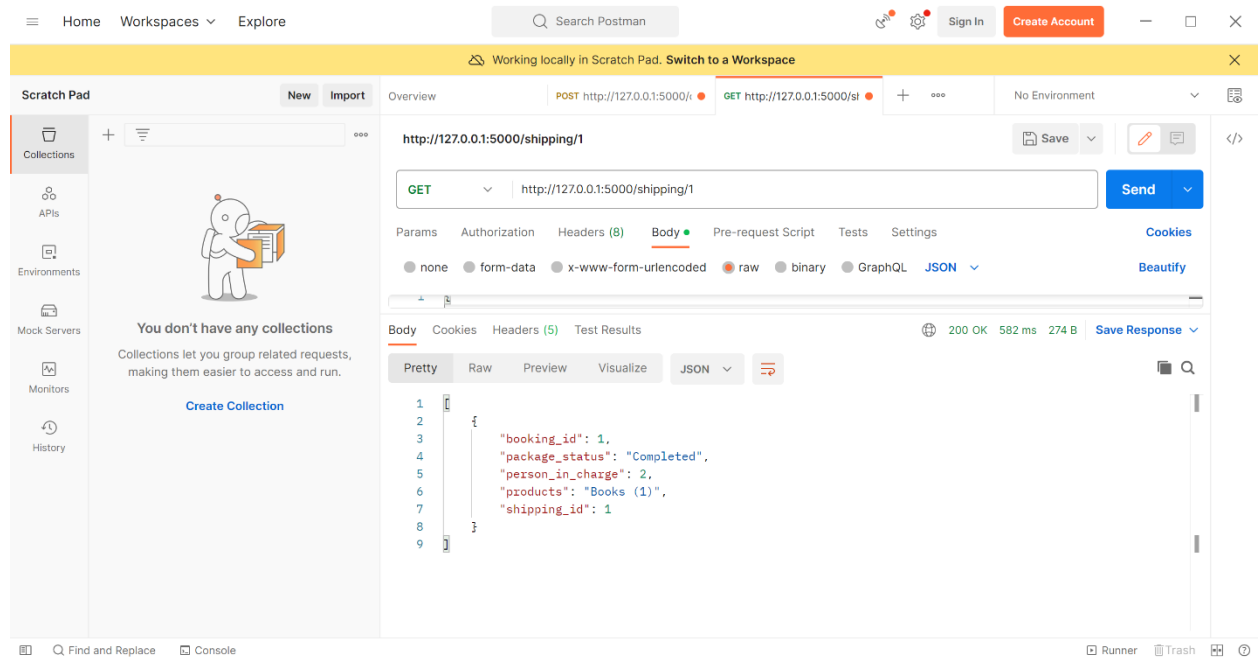
```
1. Display Booking List
2. Confirmed Customer Order
3. Update Booking Status

0. Exit
Welcome to Booking List: 2
Booking ID: 1
Do you want to confirm the booking? (y/n)y
Customer order updated successfully

1. Display Booking List
2. Confirmed Customer Order
3. Update Booking Status

0. Exit
Welcome to Booking List: 1
 booking_id customer_id customer_name products total_price customer_address customer_order booking_status
0          1           1         Anon Books (1)    750000.0      Bandung      Confirmed  On Progress
```

- Shipping status checking through API based on the booking\_id



The shipping is completed

- Completed booking:

```

1. Display Booking List
2. Confirmed Customer Order
3. Update Booking Status

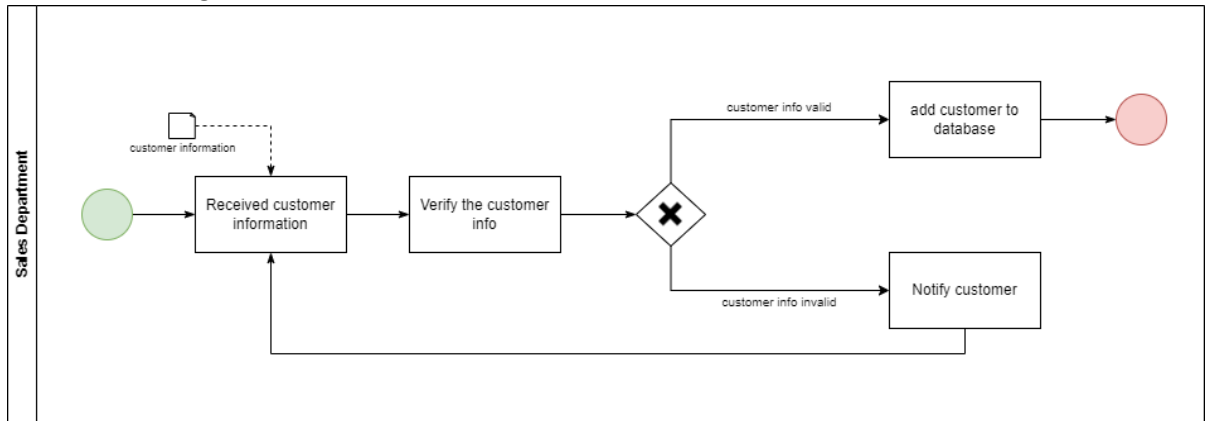
0. Exit
Welcome to Booking List: 3
Booking ID: 1
Booking status updated successfully

1. Display Booking List
2. Confirmed Customer Order
3. Update Booking Status

0. Exit
Welcome to Booking List: 1
  booking_id  customer_id  customer_name  products  total_price  customer_address  customer_order  booking_status
0           1           1           Anon  Books (1)    750000.0         Bandung        Confirmed    Completed

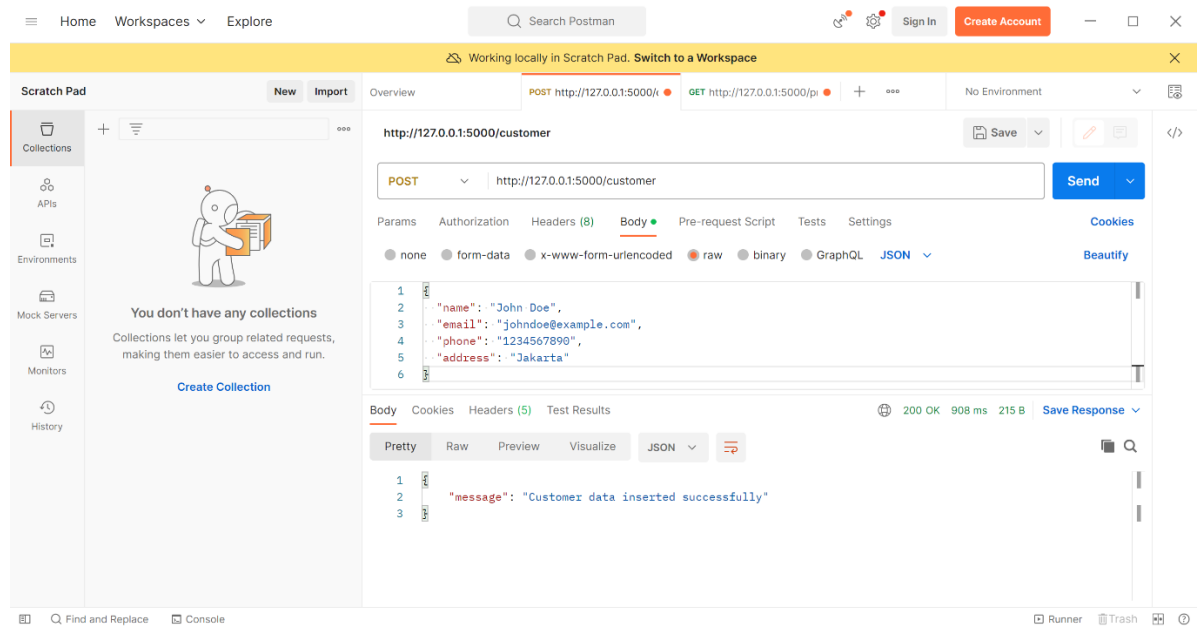
```

### 3. Customer Management

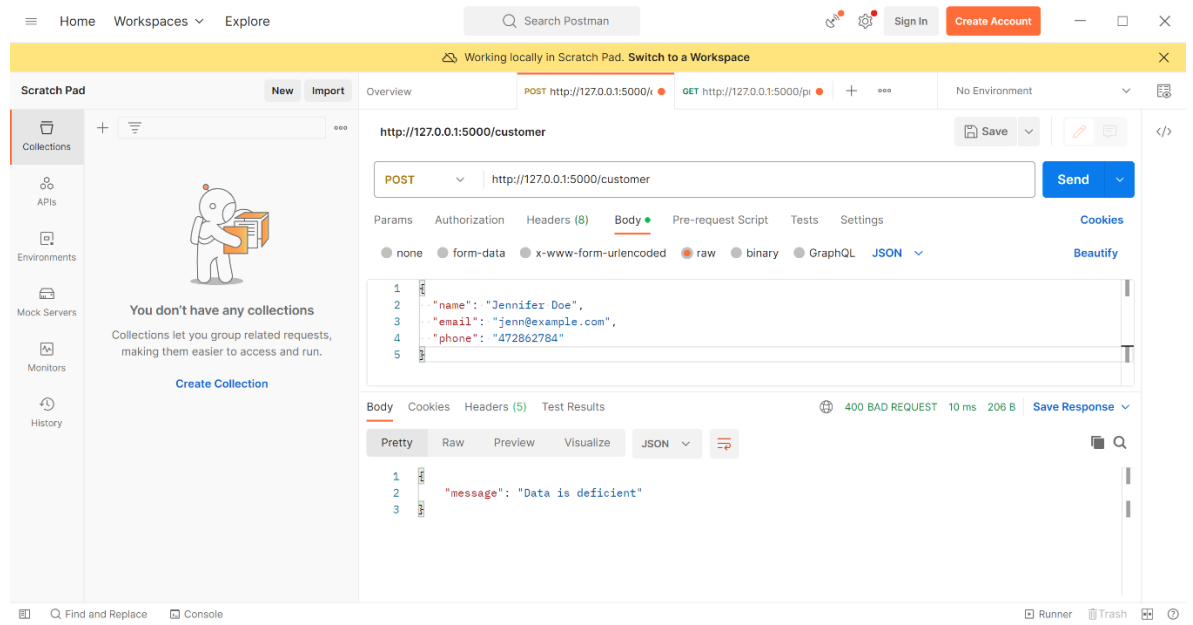


Customer sign in through API Post:

- If the data required filled in

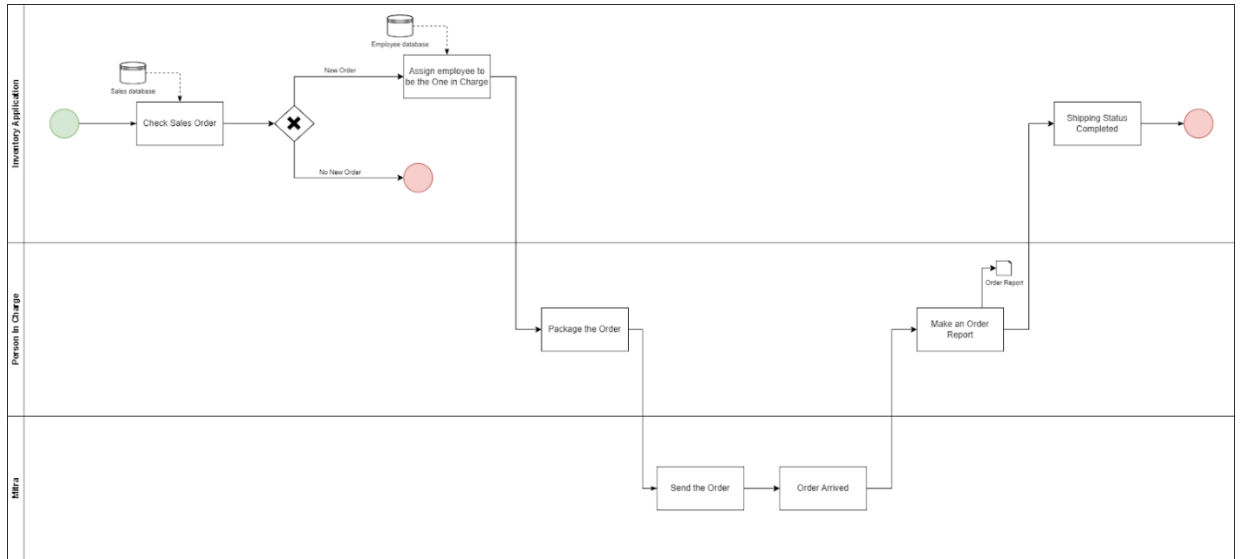


- If the data required not all filled in



# Inventory Management

## 1. Shipping Process



- Check booking that not yet completed through API:

Postman interface showing a GET request to `http://127.0.0.1:5000/bookinglist` with a successful response (200 OK) containing booking details.

**Request:** GET `http://127.0.0.1:5000/bookinglist`

**Response Body:**

```
1 {
2   "booking_id": 1,
3   "booking_status": "On Progress",
4   "customer_address": "Bandung",
5   "customer_id": 1,
6   "customer_name": "Anon",
7   "customer_order": "Confirmed",
8   "products": "Books (1)",
9   "total_price": "750000"
10 }
11
12
```



- Add new shipping order:

```
Display:
  1. Products
  2. Orders
  3. Receipt
  4. Shipping
  5. Request Task

  0. Exit
Welcome to Inventory: 4

  1. Shipping history
  2. Add new shipping order
  3. Assign person in charge for the package
  4. Change shipping status

  0. Exit
Please input your choice: 2
Please input the booking's ID: 1
How many item? 1
Input the product name: Books
Input the quantity: 1
Books (1) added to the shipping table.
```

```
  1. Shipping history
  2. Add new shipping order
  3. Assign person in charge for the package
  4. Change shipping status

  0. Exit
Please input your choice: 1
  shipping_id booking_id products person_in_charge request_id package_status
0             1           1 Books (1)           None       None       Preparing
```

- Checking inventory employee through API:

The screenshot shows the Postman interface with a GET request to `http://127.0.0.1:5000/employee_list`. The response is a JSON array of two employees:

```
[
  {
    "employee_id": 2,
    "name": "Jane Smith",
    "position": "Inventory",
    "status": "Active"
  },
  {
    "employee_id": 5,
    "name": "Rizquillah Maziya Isnaeni",
    "position": "Inventory",
    "status": "Active"
  }
]
```

The status bar indicates a 200 OK response with a 767 ms response time and 341 B of data.

- Request the employee to be the person in charge:

```

Display:
    1. Products
    2. Orders
    3. Receipt
    4. Shipping
    5. Request Task

    0. Exit
Welcome to Inventory: 5

    1. Request list
    2. Add new request

    0. Exit
Please input your choice2
Please input the employee's ID: 2
Detail example:
    Shipping ID 7
    Receipt ID 3
Please input the detail: Shipping ID 1
New request added to the request task table.

```

```

    1. Request list
    2. Add new request

    0. Exit
Please input your choice1
  request_id  employee_id      detail  status
0           1           2 Shipping ID 1  Waiting

```

- Check if the request already accepted or not:

```

    1. Request list
    2. Add new request

    0. Exit
Please input your choice1
  request_id  employee_id      detail  status
0           1           2 Shipping ID 1  Confirmed
1           2           5 Receipt ID 2  Confirmed

```

- Assign employee to handle the shipping:

```

    1. Shipping history
    2. Add new shipping order
    3. Assign person in charge for the package
    4. Change shipping status

    0. Exit
Please input your choice: 3
Please input the order's ID: 1
Please input the request's ID: 1
Please input the person in charge's employee id: 2
Employee 2 assigned to receipt 1.

```

- After the package sent and arrived, change the shipping status to completed:

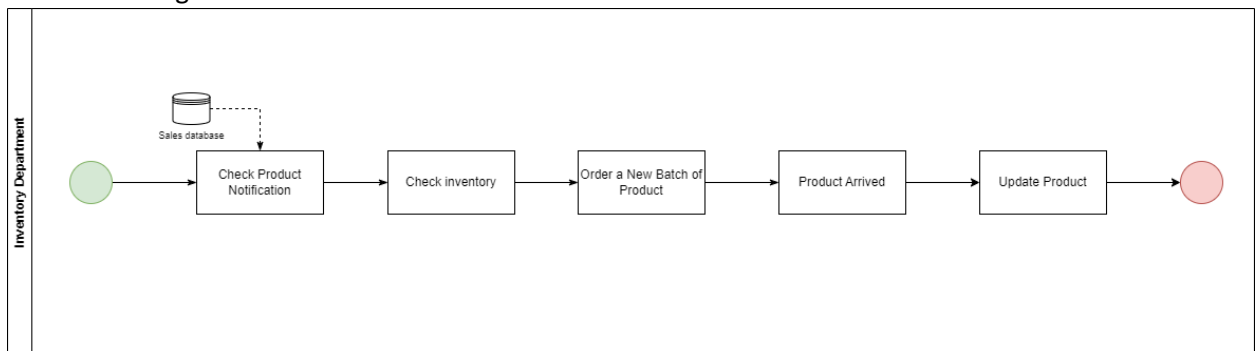
```

1. Shipping history
2. Add new shipping order
3. Assign person in charge for the package
4. Change shipping status

0. Exit
Please input your choice: 4
Please input the order's ID: 1
Change the package status to: Completed
Package status updated to Completed for booking ID 1.

```

## 2. Product Management



- Check notify from sales through API:

The screenshot shows the Postman application interface. The top bar includes navigation links (Home, Workspaces, Explore), a search bar, and user options (Sign In, Create Account). The main workspace is titled 'Working locally in Scratch Pad. Switch to a Workspace'.

On the left, the 'Scratch Pad' sidebar shows a 'Collections' section with a message: 'You don't have any collections. Collections let you group related requests, making them easier to access and run. Create Collection'.

The main area displays an API request configuration for the endpoint `http://127.0.0.1:5000/notify`. The request method is set to 'GET'. The 'Query Params' table is empty.

The 'Body' tab is selected, showing a JSON response in 'Pretty' format:

```

{
  "detail": "On Progress",
  "id": 3,
  "message": "Product empty",
  "product_id": 6
}

```

At the bottom, the status bar indicates the request was successful: '200 OK 839 ms 240 B'. The bottom of the interface includes a 'Find and Replace' bar and a 'Console' area.

- Check the notified product:

```

1. Products List Detail
2. Add Product
3. Update Product
4. Delete Product

0. Exit
Please input your choice: 1
Do you want to seek specific product? (y/n) y
Please input the product ID: 6
  product_id product_name  stock
0           6      Shoes      0

```

- Add new batch of product:

```

Display:
1. Products
2. Orders
3. Receipt
4. Shipping
5. Request Task

0. Exit
Welcome to Inventory: 2

1. Order history
2. Add new batch of product
3. Change order status

0. Exit
Please input your choice: 2
Please input the product's ID: 6
Please input the desired quantity: 10
New order added to the orders table.

```

```

1. Order history
2. Add new batch of product
3. Change order status

0. Exit
Please input your choice: 1
  order_id product_id quantity  status
0         1           6       10 On Progress

```

- After product arrived, check the receipt's status of the order

```

1. Receipts list
2. Add new receipt
3. Assign person in charge for the receipt
4. Change receipt status

0. Exit
Please input your choice: 1
  receipt_id order_id received_quantity person_in_charge request_id status
0           2         1              10             5         2 Completed

```

- Completed the order

```

1. Order history
2. Add new batch of product
3. Change order status

0. Exit
Please input your choice: 3
Please input the order's ID: 1
Change order status to Completed? (y/n) y
Order status updated successfully.

```

- The stock will automatically added based on how much is ordered in the order table

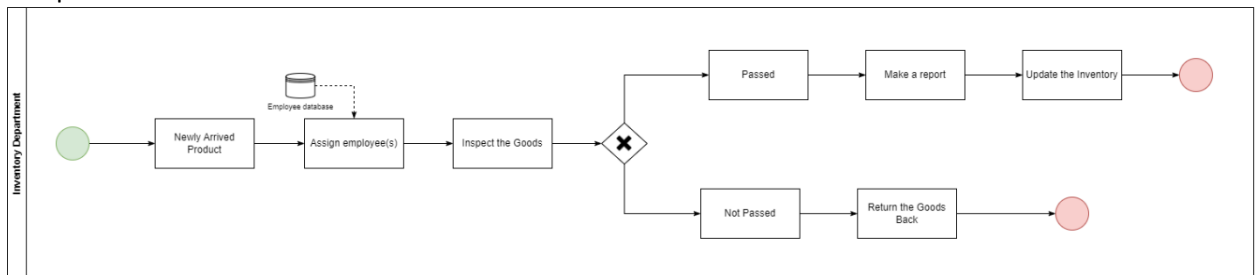
```

1. Products List Detail
2. Add Product
3. Update Product
4. Delete Product

0. Exit
Please input your choice: 1
Do you want to seek specific product? (y/n) y
Please input the product ID: 6
product_id product_name stock
0          6          Shoes    10

```

### 3. Recipient of Goods



- After the order arrived, add new receipt:

```

1. Receipts list
2. Add new receipt
3. Assign person in charge for the receipt
4. Change receipt status

0. Exit
Please input your choice: 1
receipt_id order_id received_quantity person_in_charge request_id status
0          2          1                10          None      None  Received

```

- Request for a person in charge to inspect the goods:

```

1. Request list
2. Add new request

0. Exit
Please input your choice2
Please input the employee's ID: 5
Detail example:
        Shipping ID 7
        Receipt ID 3
Please input the detail: Receipt ID 2
New request added to the request task table.

```

- Check if the request already accepted or not

```

1. Request list
2. Add new request

0. Exit
Please input your choice: 1
  request_id  employee_id      detail      status
0           1           2  Shipping ID 1  Confirmed
1           2           5  Receipt ID 2  Confirmed

```

- Assign the employee to inspect the goods

```

1. Receipts list
2. Add new receipt
3. Assign person in charge for the receipt
4. Change receipt status

0. Exit
Please input your choice: 3
Please input the receipt's ID: 2
Please input the request's ID: 2
Please input the person in charge's employee id: 5
Employee 5 assigned to receipt 2.

```

- After inspect the goods, change the status of it

```

1. Receipts list
2. Add new receipt
3. Assign person in charge for the receipt
4. Change receipt status

0. Exit
Please input your choice: 4
Please input the receipt's ID: 2

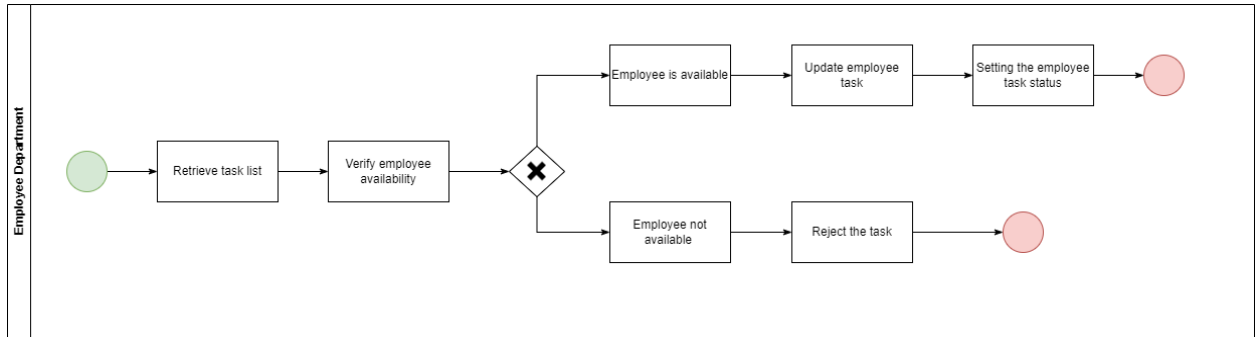
Received    = goods arrived but not yet to be inspected
Discrepancy = goods quantity not match with the order
Returned    = goods are not appropriated (damaged)
Warning     = goods not appropriated and wrong quantity
Completed   = goods appropriated and ready to sell

Change status to: Completed
Receipt 2 status updated to Completed. Request 2 status also updated to 'Completed'.

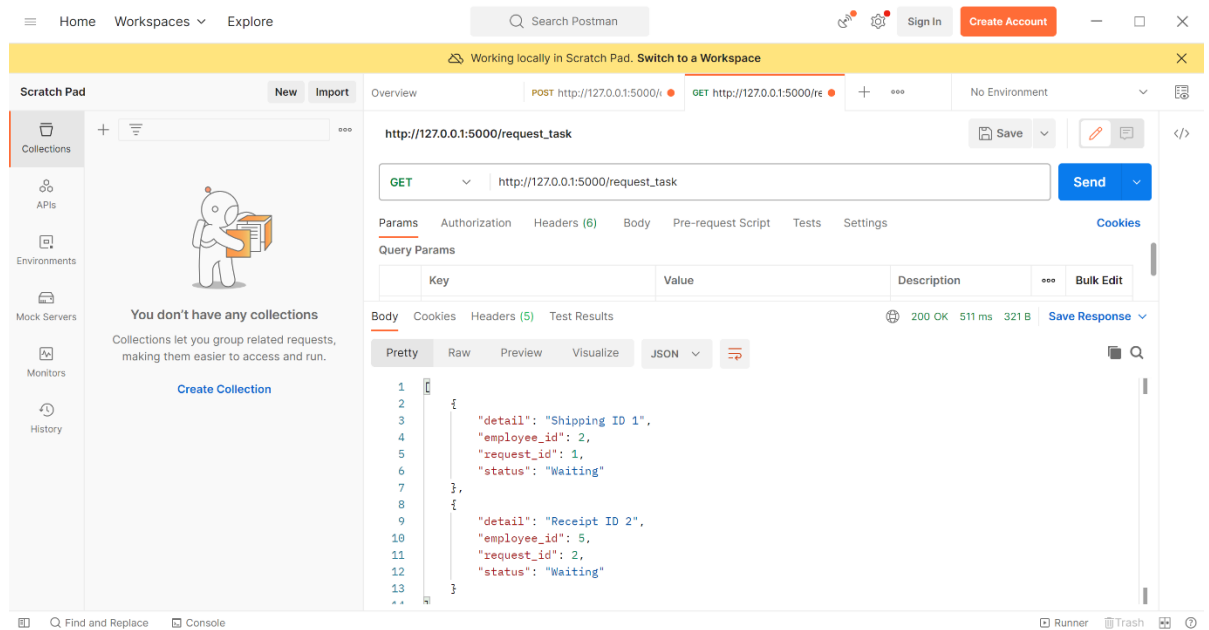
```

## Employee Management

### 1. Task Assignment and Responsibilities



- Check requested task through API:



Check the requested employee's current task:

```
Display:
1. Employee List
2. Absence
3. Task
4. Payment

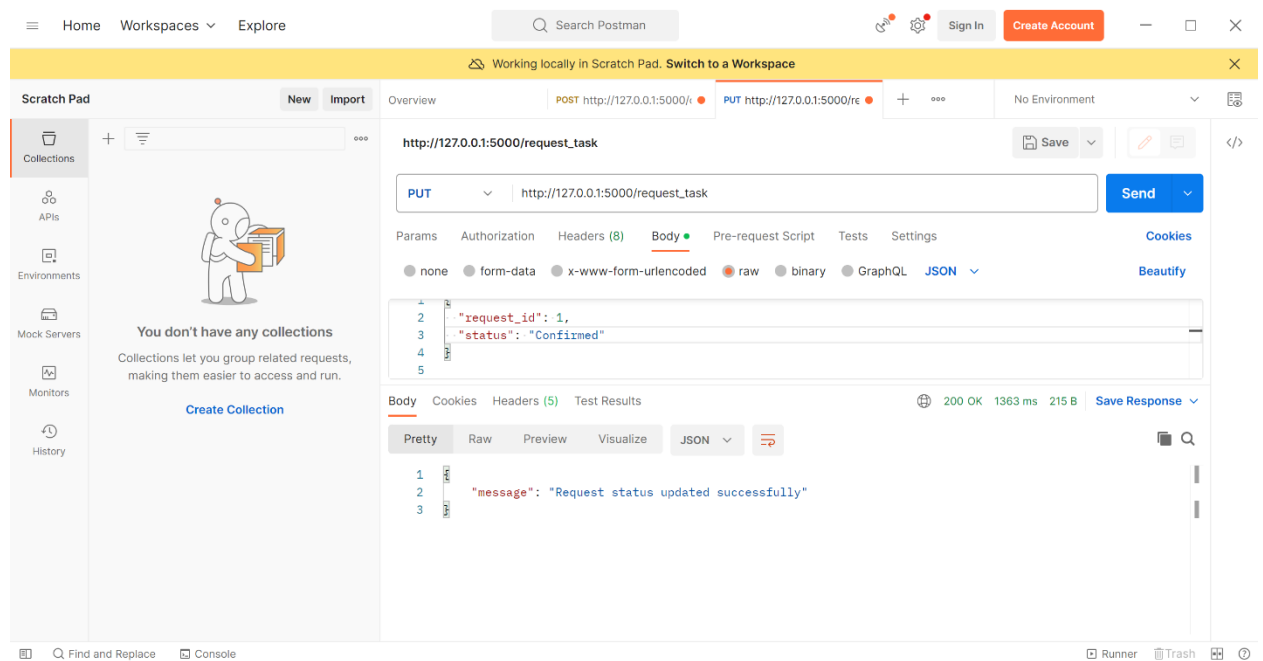
0. Exit
Welcome to Employee: 3
Display:
1. Employee Task
2. Add Task
3. Update Task

0. Exit
What do you seek in employee task? 1
do you want to seek specific employee? (y/n) y
Please input the employee ID: 2
Task ID Employee ID Name Task Task Status
0 2 2 Jane Smith Prepare storage report Completed

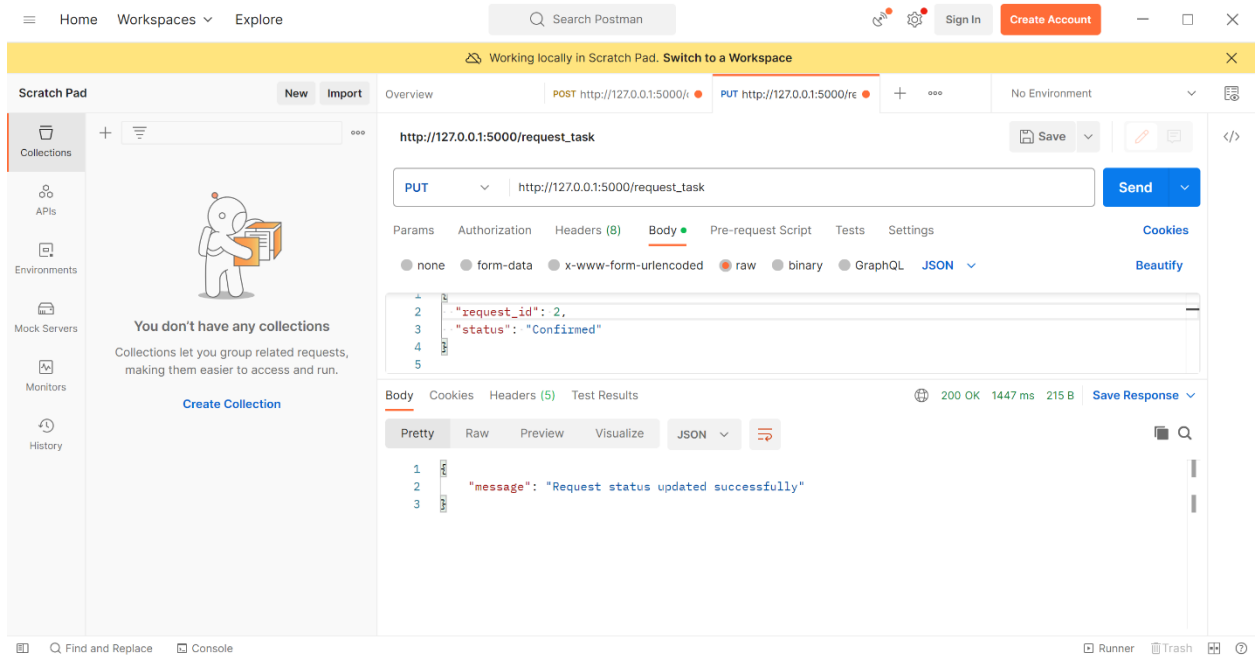
Display:
1. Employee Task
2. Add Task
3. Update Task

0. Exit
What do you seek in employee task? 1
do you want to seek specific employee? (y/n) y
Please input the employee ID: 5
Task ID Employee ID Name Task Task Status
0 5 5 Rizqullah Maziyah Isnaeni Prepare package report On Progress
```

- Accept the request:







- Add the task to the table:

```

Display:
  1. Employee List
  2. Absence
  3. Task
  4. Payment

  0. Exit
Welcome to Employee: 3
Display:
  1. Employee Task
  2. Add Task
  3. Update Task

  0. Exit
Insert employee ID: 2
Add new task: Prepare shipping package ID 1
Task added successfully!
  
```

```

Display:
  1. Employee Task
  2. Add Task
  3. Update Task

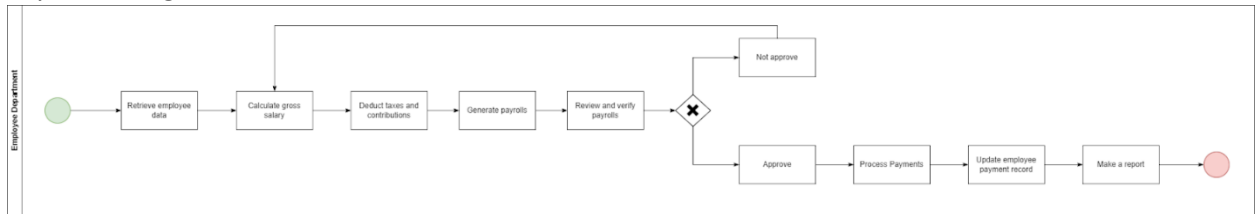
  0. Exit
What do you seek in employee task? 2
Insert employee ID: 5
Add new task: Inspect goods receipt ID 2
Task added successfully!
  
```

```

Display:
  1. Employee Task
  2. Add Task
  3. Update Task

  0. Exit
What do you seek in employee task? 1
do you want to seek specific employee? (y/n) n
Combined Task and Employee List:
  task_id  employee_id          name                task                task_status
0         3             3          Mike Johnaon      Follow up on sales leads  On Progress
1         4             4      Prima Febrianto  Conduct employee training  On Progress
2         5             5 Rizqullah Maziyah Isaeni  Prepare package report    On Progress
3         6             2           Jane Smith  Prepare shipping package ID 1  On Progress
4         8             5 Rizqullah Maziyah Isaeni  Inspect goods receipt ID 2  On Progress
5         1             1           John Doe    Prepare quarterly report    Completed
6         2             2           Jane Smith  Prepare storage report      Completed
  
```

## 2. Payroll Management



- Add Payroll:

```
Display:
1. Employee List
2. Absence
3. Task
4. Payment

0. Exit
Welcome to Employee: 4
"Display:
1. Payment List
2. Add Payroll
3. Payment
What do you seek in Payment? 2
Please input the employee's ID: 1
Please input the gross salary: 17000000
Please input the date: 2023-05-31
New payroll added successfully.
```

```
"Display:
1. Payment List
2. Add Payroll
3. Payment
What do you seek in Payment? 1
id employee_id gross_salary taxes net_salary payroll_date payroll_status payment_date payment_status
0 1 1 17000000.00 0.15 14450000.00 2023-05-31 Waiting None None
```

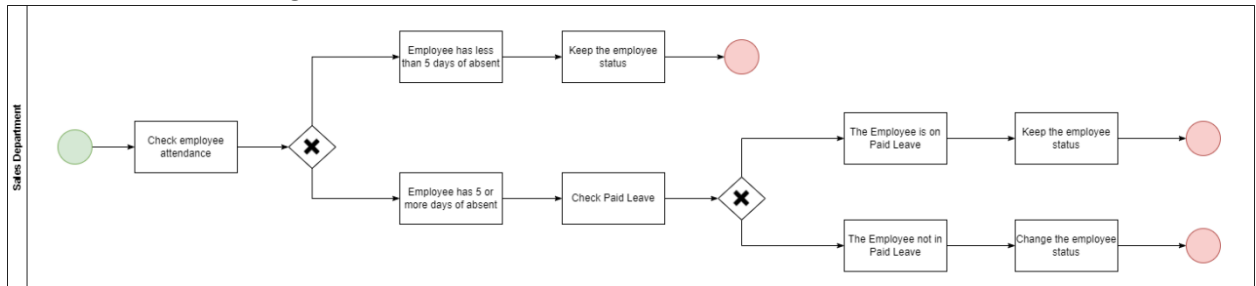
- Process the payment

```
"Display:
1. Payment List
2. Add Payroll
3. Payment
What do you seek in Payment? 3
Please input the payment's ID: 1
Please input the date: 2023-06-05
Payment processed successfully.
```

- Payment completed

```
"Display:
1. Payment List
2. Add Payroll
3. Payment
What do you seek in Payment? 1
id employee_id gross_salary taxes net_salary payroll_date payroll_status payment_date payment_status
0 1 1 17000000.00 0.15 14450000.00 2023-05-31 Confirmed 2023-06-05 Completed
```

### 3. Attendance and Warning Process



- Check attendance from absence table:

```
Display:
  1. Employee List
  2. Absence
  3. Task
  4. Payment

  0. Exit
Welcome to Employee: 2
Display:
  1. Absence list
  2. Add Absence

  0. Exit
What do you seek in absence? 1
Combined Absence and Employee List:
  employee_id  name month  absence_days  paid_leave_days
0           1   John Doe  May           6             2.0
1           2   Jane Smith May            3             NaN
2           3   Mike Johnaon May            7             NaN
3           4  Prima Febrianto May            1             NaN
```

There is an employee who absent more than 5 in May and no paid leave

- Change employee status to warning:

```

Display:
  1. Employee List
  2. Absence
  3. Task
  4. Payment

  0. Exit
Welcome to Employee: 1
Display:
      1. Employee List
      2. Change Employee Status

      0. Exit
What do you seek in employee list? 2
Insert employee id: 3
Change employee status to: Warning
Employee status updated successfully!

```

```

Display:
      1. Employee List
      2. Change Employee Status

      0. Exit
What do you seek in employee list? 1

```

employee_id	name	position	status	
0	1	John Doe	Manager	Active
1	2	Jane Smith	Inventory	Active
2	3	Mike Johnaon	Sales	Warning
3	4	Prima Febrianto	HR Specialist	Active
4	5	Rizqullah Maziyah Isnaeni	Inventory	Active
5	6	Brillian Adhiyaksa Kuswandi	Sales	Active

## External Party – Customer

- Get all the catalog list

The screenshot shows the Postman application interface. On the left, the 'Scratch Pad' sidebar is visible with options like Collections, APIs, Environments, Mock Servers, Monitors, and History. The main workspace displays a POST request to `http://127.0.0.1:5000/bookinglist`. The request body is a JSON object: 

```
{  "name": "John Doe",  "products": [    {      "name": "Watch",      "quantity": 1    },    {      "name": "Jewelry",      "quantity": 5    }  ]}
```

. The response is shown in the 'Body' tab, indicating a 200 OK status with a response time of 2.23s and a body size of 208 B. The response body is: 

```
{  "message": "Booking created successfully"}
```

- Add booking

This screenshot shows the same Postman interface as above, but with a more detailed response. The POST request to `http://127.0.0.1:5000/bookinglist` is shown with the same JSON body. The response is a 200 OK with a response time of 1822 ms and a body size of 230 B. The response body is: 

```
{  "message": "Booking added successfully",  "total_price": "8500000"}
```