**XML File Key Words:**

Order

Order Retrieved

Get Order

Get ISBN

ISBN Retrieved

V

Check Credit Card

Credit Card

Mail Address

ISBN

Customer

Verify Mail Address

Validate Credit

Card

Validate Email

Address

Credit Card Checked

Mail Address Verified

V

XOR

Place Order

Order Placed

Confirm Order

Order Confirmed

XOR

Order

completed

Credit Card Valid = invalid

Mail Address valid = False

Get ISBN

Mail

Address

Mail Address

Valid

Credit Card Valid = Valid

Mail Address Valid = trye

Customer

Analyse Order

Order Accepted

Order Rejected

Order Received

Check Stock

Product Available

Product

Not Available

Purchase New

Material

Plan Production

Raw Material

Available

Production Plan

Ready

Make Products

Products Available

Ship Products

Products Shipped

Send Invoice

Invoice Sent

* I have got all of these keywords from my terminal please based on these keywords give me a business process scenario.
* Create a concise BPMN diagram representation in markdown in sequence with activities, events, gateways, lanes, and pools using the provided keywords. Use all notations, name each notation, and organize the diagram for clarity.
* You have not notified all the keywords which I have mention above. Kindly include all the keywords and specify pools, lanes , events , gateways, activities

**# Business Process Scenario BPMN Diagram**

**## Pools**

- Customer

- System

**## Lanes**

- Customer

- System

**## Activities**

1. Customer creates an order

2. System retrieves the order

3. System gets the ISBN

4. Customer provides credit card information

5. System checks the credit card

6. Customer verifies the mailing address

7. System checks the mailing address

8. System places the order

9. System confirms the order

10. System checks the stock

11. System checks product availability

12. System purchases new material

13. System plans production

14. System checks raw material availability

15. System produces the products

16. System ships the products

17. System sends the invoice

18. System analyzes the order

19. System accepts or rejects the order

20. System completes the order

**## Events**

- Order received

- Order accepted

- Order rejected

- Product available

- Product not available

- Purchase new material

- Raw material available

- Ready

- Products available

- Products shipped

- Invoice sent

- Credit card valid

- Mail address valid

**## Gateways**

- Exclusive Gateway (XOR)

**## BPMN Diagram**

graph LR

subgraph Pool\_Customer

C[Customer]

end

subgraph Pool\_System

S[System]

end

subgraph Lane\_Customer

C -->|Create Order| S

C -->|Provide Credit Card Information| S

C -->|Verify Mailing Address| S

end

subgraph Lane\_System

S -->|Retrieve Order| S

S -->|Get ISBN| S

S -->|Check Credit Card| S

S -->|Check Mailing Address| S

S -->|Place Order| S

S -->|Confirm Order| S

S -->|Check Stock| S

S -->|Check Product Availability| S

S -->|Purchase New Material| S

S -->|Plan Production| S

S -->|Check Raw Material Availability| S

S -->|Produce Products| S

S -->|Ship Products| S

S -->|Send Invoice| S

S -->|Analyze Order| S

S -->|Accept or Reject Order| S

S -->|Complete Order| S

end

C -->|Order received| S

S -->|Order accepted| S

S -->|Order rejected| S

S -->|Product available| S

S -->|Product not available| S

S -->|Purchase new material| S

S -->|Raw material available| S

S -->|Ready| S

S -->|Products available| S

S -->|Products shipped| S

S -->|Invoice sent| S

S -->|Credit card valid| S

S -->|Mail address valid| S

**Conclusion:**

When I first parse the XML file to get keywords it does not provide me with all the keywords in business scenario but after giving multiple prompts it gives me accurate keywords with accurate workflow.

**Class:**

import openai

class ChatGPTAPI:

"""A singleton class that makes requests to the OpenAI ChatGPT API."""

\_instance = None

def \_init\_(self):

if ChatGPTAPI.\_instance is not None:

raise Exception("API is a singleton class.")

openai.api\_key = "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"

def requestGPT(self, prompt):

"""Makes a request to the OpenAI ChatGPT API and returns the response."""

completion = openai.ChatCompletion.create(

model="gpt-3.5-turbo",

messages=[

{"role": "system", "content": "You are a helpful assistant."},

{"role": "user", "content": f"{prompt}"}

]

)

return completion.choices[0].message

@classmethod

def getInstance(cls):

"""Gets the instance of the API class."""

if cls.\_instance is None:

cls.\_instance = ChatGPTAPI()

return cls.\_instance