System Testing

"Online Bookstore" System

1) Independently Testable Function:

· **Function**: ProcessOrder (a high level functio that handles a customer placing an order for books.

2) Identify Choices:

For the ProcessOrder function, some of choices that can influence its outcome can be:

a) Shopping Cart Content:
- Number of items in the cart
- Availability of books in stock
- Types of items (physical book, e-books)

b) Customer Status:
- New Customer Vs ~~exist~~ ensisting custome
- Valid customer account vs invalid/non ensistent account.

c) Payment Method:
- Credit card
- Gift card
- Debit card
- Invalid paymet details

d) Shipping Address:
- Valid domestic address
- " International "
- Invalid / incomplete address

e) Environmental Factors (Implicit Cho

- Database connection status (up/do
- Payment gateway availability (up/dow
- Inventory system response time.

3) Identify Representative Values (partitions):
Now, lets select representative & boundary values some of the identified ~~values~~ choices:

→ Choice: No of items in the cart
- Representative Values: Empty cart, 1 ite multiple items (e.g., 5 items), very large n items (e.g., 100 items - boundary)

→ Choice: Availability of books in stock
- Representative Values: All items in stock, some items out of stock, all items out of stock.

→ Choice: Customer Status
- Representative Values: New Customer (successful registeration), Existing customer (valid login), Existing customer (invalid login attempt).

Payment Method

· Representative Values: Valid credit card, Invalid credit card(e.g., expired, wrong no), valid gift card, Expired gift card, Insufficient gift card balance, Same for debit card

→ Choice: Shipping Addres.

· RV: Valid domestic address, Valid Internal address (different shipping zones/taxes), Invalid address (address not found, invalid address format)

4) Generate Test Case Specifications (Abstract Combinations): lets combine a few choices to create abstract test case specifications. Obviously we won't be ~~gen~~ generating all posible combinations, but focus on key scenarios:

a) (No of items: 1) + (Availability: All in Stock) + (Customer existing, valid) + (Payment: Valid Credit Card) + (Address: valid domestic)

· Expected Outcome: Order processed successfully, payment debited, order confirmation, shipping initiated.

) (No of items: empty) + (Availability: N/A) + (Cust Existing, valid) + (Payment: N/A) + (Address: N/A)

- Enpected Outcome: Error message: "Cart is empty. cannot proceed with order."

c) (No. of items: Multiple items) + (Availability: ~~All in stock~~ Some out of stoc) + (Customer: Enisting, valid) + (Payment, Valid ~~domestic Address~~ Credit Card) + (Address: Valid domestic)

- Enpected Outcome: System notifies customer about out of stock items, offers options (remove, backorder), and allows partial order or cancellation.

d) (items: 1) + (Aval: All in stock) + (Custmer, New, Success) + (Pay, Invalid credit Card) + (Address; Valid domest

- Enpected Outcome: Payment error message, prompt alternative payment, order not processed until valid payment.

e) (Items, 1) + (Aval, All) + (Cust, Enisting Valid) + (P: valid V
(Add, Invalid format).

- E O: Error Message: "Invalid shipping address f prompt to correct

f) (Items, 1) + (Aval, All) + (Cust: ~~Enist~~ E, V) + (P: Val
(Environmental: Database Connection down)

- E O: System error message, order cannot be plac of connection failure (this might be exception flow use case).