

**Use Case** uc-101

User wants to order a book

**Success Criteria:**

User receives confirmation on book order.  
Book order information sent to shipping department's queue.

**Main Success Scenario:**

1. Login to application
2. Search for book
3. Add book to shopping cart
4. Go to checkout
5. Enter shipping address
6. Enter credit card information
7. Submit purchase request

**Extensions:**

- 1a. User id does not exist
- 1b. Invalid password
- 1c. Password is case sensitive (switch upper/lowercase)
- 1d. User cancels out
  
- 2a. Title (as entered) does not exist
- 2b. Author name misspelled
- 2c. Author name left blank
- 2d. User exits search
- 2e. User cancels out of application
  
- 3a. User wants to select a different book
- 3b. User cancels out of application
- 3c. User proceeds to checkout (without adding book to cart)
  
- 5a. Address left blank
- 5b. Address invalid format
  
- 6a. User wants to pay by check
- 6b. User wants to pay by money order
- 6c. User wants to pay by gift card
- 6d. Credit card information rejected
- 6e. User cancels out of application
  
- 7a. User cancels purchase order
- 7b. User cancels out of application

## Test Log for Book Purchase use case

Use Case uc-101

Date:

Build Number:

Test Extension	Priority	Pass/Fail	Input values	Bugs	Observations
main					
1a					
1b					
1c					
1d					
2a					
2b					
2c					
2d					
2e					
3a					
3b					
3c					
5a					
5b					
6a					
6b					
6c					
6d					
6e					
7a					
7b					

**Use Case** uc-102

Send data from local to remote machine

**Success Criteria:**

Remote machine receives all converted data.  
Remote machine sends an acknowledgement to local machine.  
On local machine, data marked as successfully sent.  
Local machine receives acknowledgement from remote machine.

**Main Success Scenario:**

1. Programmatically Login to local database
2. Pull unsent data from the database
3. Convert that data to a pre-determined XML format and validate against a given schema definition
4. Connect to remote message queue
5. Send converted XML data to the remote queue
6. Receive acknowledgement from remote machine
7. Mark converted data as successfully sent to remote queue
8. Look for more unsent data to send to remote queue (Back to step 2)

**Extensions:**

- 1a. Unable to connect to database for whatever reason
- 1b. Invalid username/password
- 1c. Unable to connect to database port
  
- 2a. Expected data not found
- 2b. Inconsistency in database
- 2c. Connection fails during database pull
  
- 3a. Schema validation fails
  
- 4a. Connection to remote queue fails
  
- 5a. Connection fails during sending of data.
  
- 6a. No acknowledgement received by local machine
  
- 7a. Failure to mark data as sent in local database