

TEST Suite

for

Online Sales Portal

Prepared by:

Kanchi Mohan Krishna - 20CS10030
Lav Jharwal - 20CS30031
Chenna Keshava Reddy - 20CS10014

March 23, 2022

Contents

NOTE.....	1
CLASSES.....	2
1. Address Class	
2. UClass	
3. MangerClass	
4. CustomerClass	
5. Buyer Class	
6. SellerClass	
7. BuyRequest Class	
8. ItemClass	
9. Category Class	
INTERFACES	8
1. Login Interface	
2. SignupInterface	
3. AuditInterface	

NOTE:

As mentioned in the Test Plan , For the purpose of testing a sample database (dummy users and items) with the following data can be used when required.

Managers:

- admin (uid: 0001)

Buyers:

- buyer1 (uid: 1001, city: city1)
- buyer2 (uid: 1002, city: city1)
- buyer3 (uid: 1003, city: city2)

Sellers:

- seller1 (uid: 2001)
- seller2 (uid: 2002)
- seller3 (uid: 2003)

Items:

- item1 (uid: 3001, city: city1, category: cat1, seller: seller1)
- item2 (uid: 3002, city: city2, category: cat2, seller: seller2)

Category:

- cat1(uid: 4001)
- cat2(uid: 4002)

BuyRequests:

- req1(item: item1, buyer: buyer1, status: pending)
- req2(item: item2, buyer: buyer2, status: rejected)
- req3(item: item2, buyer: buyer3, status: approved)

1. Address class

	'House Number'	'Street'	'Locality'	'City'	'State'	'Pincode'
CASE-1:	“ “	“ “	“ “	“ “	“ “	“ “
Output - Exception raised because of null values						
CASE-2:	“ 2/72 B“	“C.Nagar “	“IIT “	“ Kharagpur“	“WestBengal “	“12ab34 “
Output - Exception raised because of non integral value of pincode						
CASE-3:	“ 2/72 B“	“C.Nagar “	“ “	“ Kharagpur“	“WestBengal “	“12ab34 “
Output - Exception raised because some values are null.						
CASE-4:	“ 2/72 B“	“T.Nagar “	“IIT “	“ Kharagpur“	“WestBengal “	“125634 “
Output - Data Saved						

2. User Class

	'Name'	'Telephone Number'	'Email address'
CASE-1:	“ “	“987654321”	“kesav@”
Output - Exception raised for null value of name, invalid of telephone number, invalid Email.			
CASE-2:	“chenna”	“9876543210”	“ <u>ckesav63@gmail.com</u> ”
Output - No exceptions raised.			

3. Manager Class

Pass correct parameters for parent class and address (see correct parameters in address and user class), incorrect parameters for gender and date of birth.

‘Gender’

‘Date of Birth’

CASE-1: “ “ “12/13/2002”

Output - [Exception raised for null values of gender and invalid date of birth.](#)

CASE-2: “FEMALE” “12/11/2002”

Output - [No exceptions raised.](#)

4. Customer Class

Pass correct parameters for parent class (see correct parameters in user class), incorrect parameters for city.

‘City’

CASE-1: “ “

Output - [Exceptions raised for empty city field.](#)

CASE-2: “Kharagpur”

Output - [No exceptions raised.](#)

5. Buyer Class

Constructor:

Constructor uses parent's constructor (No new parameters to be passed) Test using the test cases for customer class.

Methods:

generateBuyRequest()

	'Buyer'	'Item'	'Price'	'output'
CASE-1:	Buyer2	item1	2000	successful
CASE-2:	Buyer1	item1	1000	unsuccessful
CASE-3:	Buyer2	item4	-500	unsuccessful

6. Seller Class

	'Buyer'	'Item'	'Price'	'Status'
CASE-1:	Buyer1	item1	100	pending
CASE-2:	Buyer2	item4	6000	pending
CASE-3:	Buyer4	item2	400	pending

7. BuyRequest Class

‘BuyRequest’ ‘Buyer’ ‘Seller’ ‘Item’ ‘Buyer.city == Seller.city’ ‘Price’ ‘Output’

Req1	Buyer1	Seller1	Item1	True	2000	Created
Req2	Buyer2	Seller2	Item4	True	1500	Created
Req3	Buyer4	Seller1	Item2	False	1000	Not Created

8. Item Class

‘Name’ ‘Category’ ‘Price’ ‘City’ ‘Info’ ‘Heavy’

CASE-1: “ “ “ “ “ “ “ “ “ “ “ “

Output - Exception raised because of null values

CASE-2: “Asus Mobile“ “Mobile “ “15000 “ “New Delhi“ “Nice Mobile“ “False“

Output - No exceptions raised.

CASE-3: “ HP mouse“ “Electronics “ “1000 “ “ Agra“ “Efficient“ “False “

Output - No exceptions raised.

9. Category Class

1. Pass incorrect parameters

name - ""

Output- Exceptions raised for empty name field.

2. Pass all correct parameters

name - "category1"

Output - No exceptions raised.

3. name - "category2"

Output - No exceptions raised.

Check for unique id: category1.id! = category2.id

Output - True

INTERFACES

1. Login Interface

Testing Manager

<Email Id>	<password>	<Type>	<Output>
mohank@gmail.com	jharwall	Manager	True
ckesav63@gmail.com	abc	Manager	Error
mohank@gmail.com	abc	Manager	Error
mohank@gmail.com	jharwall	Buyer	Error
ckesav63@gmail.com	jharwall	Manager	Error

Testing Buyer

<EmailID>	<password>	<Type>	<Output>
mohank@gmail.com	jharwall	Buyer	True
ckesav63@gmail.com	abc	Buyer	Error
mohank@gmail.com	abc	Buyer	Error
mohank@gmail.com	jharwall	Manager	Error
ckesav63@gmail.com	jharwall	Buyer	Error

Testing Seller

<Email Id>	<password>	<Type>	<Output>
lav12@gmail.com	abcd@123	Seller	True
ckesav63@gmail.com	abc	Seller	Error
lav12@gmail.com	abc	Seller	Error
lav12@gmail.com	abcd@123	Buyer	Error
ckesav63@gmail.com	abcd@123	Seller	Error

2. Signup Interface

Signup of the following users with all correct fields and email address is tested on an initially empty database:

Email ID	Type	Output
ckesav63@gmail.com	Seller	Signup successful
mohan3@gmail.com	Buyer	Signup successful
lav4@gmail.com	Manager	Sign Up successful
ckesav63@gmail.com	Seller	Seller with given email already exists
ckesav634@gmail.com	Seller	Signup successful
mohan3@gmail.com	Buyer	Buyer with a given email already exists.
lav44@gmail.com	Manager	Signup successful
ckesav63@gmail.com	Seller	Seller with given email already exists

3. Audit Interface

Let the database contain

Items:

- item1 (uid: 2011, isheavy: true, city: city1, category: cat1, seller: seller1)
- item2 (uid: 2012, isheavy: false, city: city2, category: cat2, seller: seller2)
- item3 (uid: 2013, isheavy: true, city: city4, category: cat1, seller: seller1)

Categories:

- cat1
- cat2

Pending Buy Requests:

- item1 (by buyer1, pending)
- item1 (by buyer2, pending)
- item3 (by buyer3, pending)

If cat1 is deleted

'Item'	'Item ID'	'Category'
item1	2011	unknown
item2	2012	cat2
item3	2013	unknown

If item1 is deleted

'Item'	'Item ID'	'Category'
item2	2012	cat2
item3	2013	unknown