

# Test Cases

<b>Test ID</b>	TC002
<b>REST Request</b>	GET/Search?name=item1
<b>Category</b>	Evaluates if existing items in catalogue can be retrieved from Catalogue table in Database, after item search is initiated.
<b>Requirements Coverage</b>	<i>UC2.1-Succesful-BrowseCatalogueofAuctionedItems-ItemSearch</i>
<b>Initial Condition</b>	User is signed in.
<b>Procedure</b>	1. The user types in the name of the item they are looking for. 2. The user selects the search button.
<b>Expected Outcome</b>	A list of items matching the keyword the user enters into the search bar; “item1”, is displayed to the user.
<b>Notes</b>	The name of the item that the user provides in the search bar must contain letters, may or may not contain numbers, and must not contain any special characters.

<b>Test ID</b>	TC003
<b>REST Request</b>	GET/Search?name=item2
<b>Category</b>	Evaluates if non-existing items are not retrieved from the Catalogue table in Database, after item search is initiated.
<b>Requirements Coverage</b>	<i>UC2.1-Succesful-BrowseCatalogueofAuctionedItems-ItemSearch</i>
<b>Initial Condition</b>	User is signed in.
<b>Procedure</b>	1. The user types in the name of the item they are looking for. 2. The user selects the search button.
<b>Expected Outcome</b>	A message specifying that the item does not exist; “There are no item2 items.”
<b>Notes</b>	The name of the item that the user provides in the search bar must contain letters, may or may not contain numbers, and must not contain any special characters.

<b>Test ID</b>	TC004
<b>REST Request</b>	GET/Search?name=
<b>Category</b>	Evaluates if searching for an unnamed item from the Catalogue table in Database cannot occur.
<b>Requirements Coverage</b>	<i>UC2.1-Fail-BrowseCatalogueofAuctionedItems-ItemSearch</i>
<b>Initial Condition</b>	User is signed in.
<b>Procedure</b>	1. The user leaves the search bar empty. 2. The user selects the search button.
<b>Expected Outcome</b>	An error message specifying that the user must enter an item name to search for.
<b>Notes</b>	The name of the item that the user provides in the search bar must contain letters, may or may not contain numbers, and must not contain any special characters.

<b>Test ID</b>	TC005
<b>REST Request</b>	POST/Sell?name=item1&description=furniture&auctionType=Forward&time=15&startBidPrice=125

<b>Category</b>	Evaluates if the catalogue service is able to store a valid item for sale in the Catalogue table of the Database, given that the user supplies correct inputs.
<b>Requirements Coverage</b>	<i>UC7-Successful-SellItem</i>
<b>Initial Condition</b>	User is signed in.
<b>Procedure</b>	<ol style="list-style-type: none"> <li>1. The user inputs the name of the item.</li> <li>2. The user inputs a description of the item.</li> <li>3. The user inputs the type of auction the item will be entered in.</li> <li>4. The user inputs the duration of the auction.</li> <li>5 The user inputs the starting bid price of the item on the auction.</li> <li>6. The user selects the Sell Item button.</li> </ol>
<b>Expected Outcome</b>	A confirmation message is displayed to the user informing them the item has been added to the catalogue.
<b>Notes</b>	The name must contain letters, may or may not contain numbers, and no special characters. The description and auctionType must only contain letters.

<b>Test ID</b>	TC006
<b>REST Request</b>	POST/Sell?name=item1!&description=furniture&auctionType=Forward&time=15&startBidPrice=125
<b>Category</b>	Evaluates if the catalogue service is able to properly store an item for sale in the Catalogue table of the Database, given the correct user inputs.
<b>Requirements Coverage</b>	<i>UC7-Fail-SellItem</i>
<b>Initial Condition</b>	User is signed in.
<b>Procedure</b>	<ol style="list-style-type: none"> <li>1. The user inputs the name of the item.</li> <li>2. The user inputs a description of the item.</li> <li>3. The user inputs the type of auction the item will be entered in.</li> <li>4. The user inputs the duration of the auction.</li> <li>5. The user inputs the starting bid price of the item on the auction.</li> <li>6. The user selects the Sell Item button.</li> </ol>
<b>Expected Outcome</b>	An error message is displayed informing the user of the specific incorrect inputs; “item1! is an invalid name”.
<b>Notes</b>	The name must contain letters, may or may not contain numbers, and no special characters. The description and auctionType must only contain letters.

<b>Test ID</b>	TC007
<b>Category</b>	Checks if the paymentservice is able to print out a valid receipt for the purchase given that the user has inputted correct inputs.
<b>Requirements Coverage</b>	<i>UC6-Successful-GetReciept</i>
<b>Initial Condition</b>	User is signed in and User has made a purchase
<b>Procedure</b>	<ol style="list-style-type: none"> <li>1. The user inputs name</li> <li>2. The user provides shipping address</li> <li>3. The user wins bid for item</li> <li>4. The user pays for item</li> </ol>
<b>Expected Outcome</b>	The expected outcome of the test case is that the user gets a receipt containing all their information regarding the purchase

Notes	
-------	--

Test ID	TC008
REST Request	PUT/Pay?bidprice=
Category	Makes payment for item that they have won the bid for.
Requirements Coverage	<i>UC5-Successful-MakePayment</i>
Initial Condition	User is signed in nd has won the bid for the selected item.
Procedure	<ol style="list-style-type: none"> <li>1. The user has bid for item.</li> <li>2. The user has won bid.</li> </ol>
Expected Outcome	A successful payment message
Notes	The name of the item that the user provides in the search bar must contain letters, may or may not contain numbers, and must not contain any special characters.

Test ID	TC009
REST Request	PUT/Pay?bidprice=
Category	Evaluates if the payment wasnt sufficient enough
Requirements Coverage	<i>UC5-Failed-MakePayment</i>
Initial Condition	User is signed in nd has won the bid for the selected item.
Procedure	<ol style="list-style-type: none"> <li>1. The user inputs name</li> <li>2. The user provides shipping address</li> <li>3. The user wins bid for item</li> <li>4. The user pays for item</li> </ol>
Expected Outcome	An error message specifying that the user has provided an insufficient amount of the winning bid price
Notes	

Test ID	TC010
REST Request	POST/sign_in?new_user
Category	Creating a new user successfully
Requirements Coverage	<i>UC1.1-Succssful-SignUp</i>
Initial Condition	Username, password and user information do not exist yet
Procedure	<ol style="list-style-type: none"> <li>1. The user enters website and enters sign up page</li> <li>2. User enters username, password and personal information</li> <li>3. The user proceeds to ecommerce login webpage</li> </ol>
Expected Outcome	Login page to use newly created account information, username and password to enter website
Notes	User must remember their username and password

Test ID	TC011
REST Request	POST/sign_in?new_user

<b>Category</b>	Failing to create a new user due to redundant information or incorrect user information
<b>Requirements Coverage</b>	<i>UC1.1-Failed-SignUp</i>
<b>Initial Condition</b>	Username, password and user information do exist already
<b>Procedure</b>	<ol style="list-style-type: none"> <li>1. The user enters website and enters sign up page</li> <li>2. User enters username, password and personal information</li> <li>3. The user stays on sign-up page and is presented with error pop-up window</li> </ol>
<b>Expected Outcome</b>	Error failed to sign up pop-up window shows up
<b>Notes</b>	User has limited tries to make profile, 5 attempts in 24 hours, to prevent spam, bots and hacks

<b>Test ID</b>	TC012
<b>REST Request</b>	POST/sign_in?username&password
<b>Category</b>	Authenticating username and password entered are correct
<b>Requirements Coverage</b>	<i>UC1.2-Succssful-SignIn</i>
<b>Initial Condition</b>	Username and password are correct
<b>Procedure</b>	<ol style="list-style-type: none"> <li>1. The user enters username &amp; password</li> <li>2. Username and password are correct</li> <li>3. The user proceeds to ecommerce webpage</li> </ol>
<b>Expected Outcome</b>	Logged-in ecommerce web page is displayed
<b>Notes</b>	User is who they say they are, and therefore knows username and password

<b>Test ID</b>	TC013
<b>REST Request</b>	POST/sign_in?username&password
<b>Category</b>	Authenticating username and password entered are incorrect
<b>Requirements Coverage</b>	<i>UC1.2-Failed-SignIn</i>
<b>Initial Condition</b>	Username and password are incorrect
<b>Procedure</b>	<ol style="list-style-type: none"> <li>1. The user enters username &amp; password</li> <li>2. Username and password are incorrect</li> <li>3. The user stays on sign-in page and is presented with error pop-up window</li> </ol>
<b>Expected Outcome</b>	Error failed to login pop-up window shows up
<b>Notes</b>	User is not who they say they are, and therefore does not know username and password

<b>Test ID</b>	TC014
<b>REST Request</b>	GET/bid?itemId=01
<b>Category</b>	Retrieve an Item's bid page with all the correct information
<b>Requirements Coverage</b>	UC2.3 Successful-Item-Selection
<b>Initial Condition</b>	These is an item with the given itemId
<b>Procedure</b>	<ol style="list-style-type: none"> <li>1. User searches an item</li> <li>2. User selects radio button</li> <li>3. User presses select</li> </ol>
<b>Expected Outcome</b>	User is directed to the bid page for the item showing highest bid, highest bidder and options to bid
<b>Notes</b>	User should be signed in, price should be a number

<b>Test ID</b>	TC001
<b>REST Request</b>	PUT/bid?price=100
<b>Category</b>	ensuring only a higher bid is accepted
<b>Requirements Coverage</b>	UC3.1-Failed-ForwardAuction-Bid
<b>Initial Condition</b>	A bid has already been made and recorded and it's higher than 100
<b>Procedure</b>	1. The user selects a forward Bid item 2. The user enters a lower price than the highest bid displayed 3. The user presses bid
<b>Expected Outcome</b>	User is redirected to the bid page for the item and a failed bid message is shown.
<b>Notes</b>	User should be signed in, price should be a number

<b>Test ID</b>	TC015
<b>REST Request</b>	PUT/bid?price=100
<b>Category</b>	Facilitate forward auction bidding
<b>Requirements Coverage</b>	UC3.1-Successful-ForwardAuction-Bid
<b>Initial Condition</b>	The entered bid is the highest so far
<b>Procedure</b>	1. The user selects a forward Bid item 2. The user enters a higher bid than displayed 3. The user presses bid
<b>Expected Outcome</b>	The page is refreshed displaying the user's new bid
<b>Notes</b>	User should be signed in, price should be a number

<b>Test ID</b>	TC016
<b>REST Request</b>	GET/End?itemId=01&userId=11
<b>Category</b>	End Auction and notify users
<b>Requirements Coverage</b>	UC4-Auction-Ended
<b>Initial Condition</b>	No one else has pressed buy
<b>Procedure</b>	1. User opens an item bid page 2. User presses buy on a Dutch auction
<b>Expected Outcome</b>	An auction ended page should be displayed for all participating users
<b>Notes</b>	User should be signed in

<b>Test ID</b>	TC017
<b>REST Request</b>	POST/AuctionInfo?itemId=10&starttime=20-10-2023&endtime=30-10-2023
<b>Category</b>	Register New item to sell
<b>Requirements Coverage</b>	UC7-Sell-update-database
<b>Initial Condition</b>	
<b>Procedure</b>	1. User opens sell item page

	2.User enters description, type of auction and start price 3.User clicks sell
<b>Expected Outcome</b>	Returns true for status of row entry
<b>Notes</b>	User should be signed in, itemId is an integer

<b>Test ID</b>	TC018
<b>REST Request</b>	GET/AuctionInfo?itemId=10
<b>Category</b>	Retrieve auction info
<b>Requirements Coverage</b>	UC2.3 Successful-Item-Selection
<b>Initial Condition</b>	The item with the itemId exists
<b>Procedure</b>	1. User selects an item from search result / item list
<b>Expected Outcome</b>	Return an item object for the row with the given itemId
<b>Notes</b>	itemId is an integer

<b>Test ID</b>	TC019
<b>REST Request</b>	DELETE/CatalogInfo?itemId=10
<b>Category</b>	Remove item from catalogue
<b>Requirements Coverage</b>	UC5-Successful-Payment
<b>Initial Condition</b>	The item with the itemId exists
<b>Procedure</b>	1. User wins an item 2. User pays for item successfully
<b>Expected Outcome</b>	Item is deleted from catalog table and returns true
<b>Notes</b>	User should be signed in, itemId is integer

<b>Test ID</b>	TC020
<b>REST Request</b>	PUT/AuctionInfo?itemId=10&highestBid=1000&highestBidder=11
<b>Category</b>	Update highest bidder
<b>Requirements Coverage</b>	UC3.1-Successful-ForwardAuction-Bid
<b>Initial Condition</b>	The entered bid is the highest so far
<b>Procedure</b>	1.Update new bid, bidder 2.User opens forward auction item bid page 3.User enters price a higher price than listed 4.User presses bid
<b>Expected Outcome</b>	The new bid and bidder is stored in the auction table, and returns true
<b>Notes</b>	User should be signed in, price should be a number