Test plan for Sprint 3.

**1.Implement Forgotten Password**  
  
When click on the Forgotten Password link, just go to a simple page with an input field to enter the user's email, then generate a random password and send to the user's email, like this one:  
<http://www.trademe.co.nz/Members/PasswordReminder.aspx>

Go to login page, and check the presence of forgot password button.

Click the button, and check the returned forgot password page.

Perform literature check on the forgot password page.

Run a sanity check on the input fields.

Provide valid information and a confirmation page or popup would appear.

Check the reset password email sent.

Follow the reset password link to the validation page.

Perform sanity check again on the input fields, on the repeating new passwords, etc, against reset rule.

Provide valid password, and check the confirmation message.

Make sure you are logged in automatically. (optional)

Log off, and re-log in, and check if the new password is in place.

**2. Allow payment by credit card and PayPal, and allow credit card to be saved. Credit Card gateway will be using DPS.**  
  
All transaction on this platform can be paid by Visa card, Master card and PayPal. I.e. buyer can use this mechanism to pay seller, and seller can use this mechanism to pay the website administrator too.

Recommend manage your cards session, in addition to saved last used card.

Run sanity check on card maintenance page as usual web input form test, as test plan 1.

Test plan 2 is to check savinf card information facility.

Recommended is to add a check box for user to save it, after the card information is filled and confirming payment will also get the current card saved.

Based on DPS payment, check the third party payment interface is smoothly connect through shopmax payment portal.  
  
 **3. As a user should be able to login using social sites as a buyer or seller.**  
  
User is able to register as buyer or seller, a seller can also be a buyer. People can also login with their Facebook or Twitter or Gmail account, and then they immediately become a buyer. For a buyer to apply to be a seller they just need to fill more information.

Check the login page has the multiple option of social account.

Ideally, the different social account need to be selected prior to the input, as this way, we can change login facility at the background, as they may use different portal.

Perform sanity check with the input fields.

Make sure all provided social site are available to be logged in.

Check the logged in user is active as a buyer, or a email notification or activation is sent.

Check the activation link is desiable.

Test plan 2 is to convert the logged in buyer to seller.

This would simply adopt the normal registered user/seller conversion test plan.  
  
  
**4. Import Product from crawler output ( JIRA SD-5 )**  
  
Create a scheduled job to import product, runs daily automatically, but can also be run manually if needed.  
  
The product import format and sample file etc has been explained in this Wiki page: <https://sites.google.com/site/shopmaxwiki/import-product-file-format>  
  
Note: this task is only for importing product from crawler output file only, not include file format for seller.

Preparation:

Form a valid csv Import file according to the standard. ([http://supercsv.sourceforge.net/csv\_specification.html](http://www.google.com/url?q=http%3A%2F%2Fsupercsv.sourceforge.net%2Fcsv_specification.html&sa=D&sntz=1&usg=AFrqEzcU6z0gt9Dq_oN9P83ckWJ8Q8HGLA) )

File format for seller:

"Product Name","Product Description","Category Path","List Price","Discount Price","Discount Price Start Date","Discount Price End Date","Stock","Shipping Size","Product Image Filename","SKU"

Check the picture of each product in the according field. Validation check is requested.

Check if the system will only take the first 6, minimum of one image is required, empty string not allowed.

If "Product Image Filename" is empty, it will automatically use sku\_1.jpg, sku\_2.jpg.... sku\_6.jpg as product image, if both "Product Image Filename" and "SKU" field is empty on the same row, then an error should be thrown. However not necessarily all products will have 6 images up to sku\_6.jpg, it is possible to have only sku\_1.jpg and sku\_2.jpg in the file system, and this is valid.

In the "Category Path" fields, each category is separated by /

E.g.  art/print/other

All optional fields are:

"Discount Price","Discount Price Start Date","Discount Price End Date","Stock","Shipping Size","SKU"

If "Stock" not specified, it means unlimited stock.

If "Shipping Size" not specified, it means pick up in store only.

Check the file format for crawler

The file format for crawler is almost the same as for seller, except one extra field "Product Thumbnail URL" at the end, i.e.

"ProductName","ProductDescription","CategoryPath","ListPrice","DiscountPrice","DiscountPriceStartDate","DiscountPriceEndDate","Stock","ShippingSize","ProductImageFilename",

"SKU","ProductThumbnailURL","SellerSubdomainName"

In crawler exported product import files, the value for "DiscountPriceStartDate","DiscountPriceEndDate","Stock","Shipping Size","SKU" may be empty.

And the value for "ProductImageFilename","ProductThumbnailURL"  are the full URL to the image, if there are more than one, it will also be delimited by ;

The images in "ProductImageFilename" are the large images, whereas the one in "ProductThumbnailURL" are small thumbnail images.

The values in "ProductDescription" will be HTML decoded, i.e. <br/> becomes line break, &amp; becomes & etc.

Line break is acceptable in CSV standard.

All sellers for product import will have been pre-created. The "SellerSubdomainName" is the subdomain name of the seller on our website, e.g.[dicksmith.shopmax.co.nz](http://dicksmith.shopmax.co.nz/" \t "_blank), this is used to identify the seller. Attached crawl\_list\_with\_seller\_mapping.csv shows a list of seller subdomain names.

The import will update the product if it exists. So the way it identifies a product is via "SellerSubdomainName" and "SKU" or "ProductName"

If SKU is present in import file, then it will use SKU to find the product in that seller.

If SKU is EMPTY, then it will use the "ProductName" to identify the product, i.e. oldName.equalsIgnoreCase( newName );

Make sure that all products imported from crawler will be marked as in store purchase only, i.e. can't be purchased online.

Check the upload facility, according the future design, common literature check applies.

Check for any error occurred during the process.

End to end test to check imported product are in the desired place with any additional request for automatic crawler import.  
  
  
**5. On upload product page, allow upload image before the product is created ( JIRA SD-4)**  
  
On upload product page, it allows the user to upload up to 4 product images.  
( e.g. <https://shopmaxseller.ofbizdemo.com/shop/control/uploadproduct> )  
  
Allow user to upload and preview the image before the product is created.  
  
It should delete any temp file / content if user cancels.  
  
Just FYI, there is a jQuery library that allows preview image after upload, check out the Basic Plus UI demo at <http://blueimp.github.io/jQuery-File-Upload/>  
  
I think the current latest version is 8.2.1: <https://github.com/blueimp/jQuery-File-Upload/tree/8.2.1>  
  
You can use this if you want.

Log in as buyer, and make sure the facility is hidden.

Log in as seller, and perform literature check for the image uploading web fields.

For dynamix html, please check predefined effects according to the Jquery library definition.

Check the uploaded image size, quality and other aspects.

Make sure the image can be uploaded without having the product created.

Check the preview/deletion actions.

Check all error messages during the sanity check.

Check the stored file folder structure if needed for optimal performance.  
  
  
  
**6. Import category ( JIRA SD-3 )**  
  
Import category from CSV file. The file format is attached.  
  
The second column is the category ID, the / represents the category hierarchy relationship. E.g. apparel has not other category ID in front of it, so it's a level 1 category.  
  
apparel/baby means the Baby category is a child category of apparel. "apparel/baby" is the full category ID of the Baby category.  
Each / is one level down.  
  
ShowOnHomePage is an attribute on the category that will be used later.  
  
The categories should be created under the SHOPMAX\_BROWSE\_ROOT catalog at the moment I think, in order for it to show. But not sure if this is correct or should be changed. It's depending on another requirement about Category.  
  
Every time the category is imported it re-creates the whole category structure, not update.

Prepare the csv file as instructed in the design document.

Check the upload facility with literature check.

Make sure the end to end test passes for importing process.

Note the identical category and make sure it appears in the system.

Note down any error occurred during the importing process.

Make sure a performance indicator is designed and recorded, using big size file.

Minimal sanity check on csv with bad format, as for internal process, a file format is almost healthy.

Yet to be determined that if any sub category can be imported at the same. It will be confirmed once implemented.

**7. Jira SD-6, Complete My Profile functions for both buyer and seller**

Off manage branch page, check the formation of the page, along with all link forward and backward.

Check all of the My Profile functions, make sure a logic literature check is done.

Test need to include buyer, compact version; and seller, with additional sections.

Check combination of logic connection between fields.

Make sure the saving/cancel/quite mechanism works as designed.

**8. Jira SD-7, Allow advanced upload product option, screen might need to be modified**

Sanity-check the variant advanced product options.

Upload product variant is according to category type.

Cross category for identical user, need to be verified for their logical connection. When necessary a matrix can be introduced as test case.

Find out and test the allowed and limited product that can be uploaded for identical category.

Check the preset of the category in affecting the advance options.

Check the search and filter functions from frontend to ensure that the advanced option is adopted in those functions.

Check the Advance Options UI and literature.

A variant can be reflected to the buyer’s front end, check with identical products with variants.

Checkout process, cart, order, confirming order and email notification should all have these options selected in it.

An additional window may be introduced at frontend, that buyer can operate on, during purchasing.

Note all errors occurred during the test.

**9. Jira SD-8, Use Solr 4 for search and filtering on the website, allow fuzzy search**

Refer to Solr documents, as it is used for product search and search for shop on the website.

Prepare identical data according to the search types.

Perform test on the following 3 search types:

1. Search product across the whole website

2. Search for shop

3. Search product within a shop

Check 1 and 3, for narrowing the search result by price, category and suburb (location), also check undo their filtering.

Check the corresponding dropdown value for these 3 functions:

1. All Products

2. Search Shop

3. This Shop

Test searching shops, by remove the filter by price. The search result can also be sorted by product/shop name, price, user rating and store vicinity. The sorting AJAX might have been done. Need to double check.

Check the sort by options:

1. Price high to low

2. Price low to high

3. Avg. customer review

4. Distance

Check the Distance option that will only be available when the current user's position is known, either the user enters it or via HTML5.

Check switching between grid view and list view in search result page, (in AJAX), the page won't refresh. All static HTML in action at <http://hongchen.me/shopmax/Search_list_view.html>   ( for other pages, just change the HTML file name )

Location needs to be check for Google maps connection.  In detail of distance shop location and products need to be checked. Check the refresh will not lose the current search location.

If the shop has more than 1 branch, check if it shows the closest location.