

# **Software Requirements Specification**

**Version 0.1**

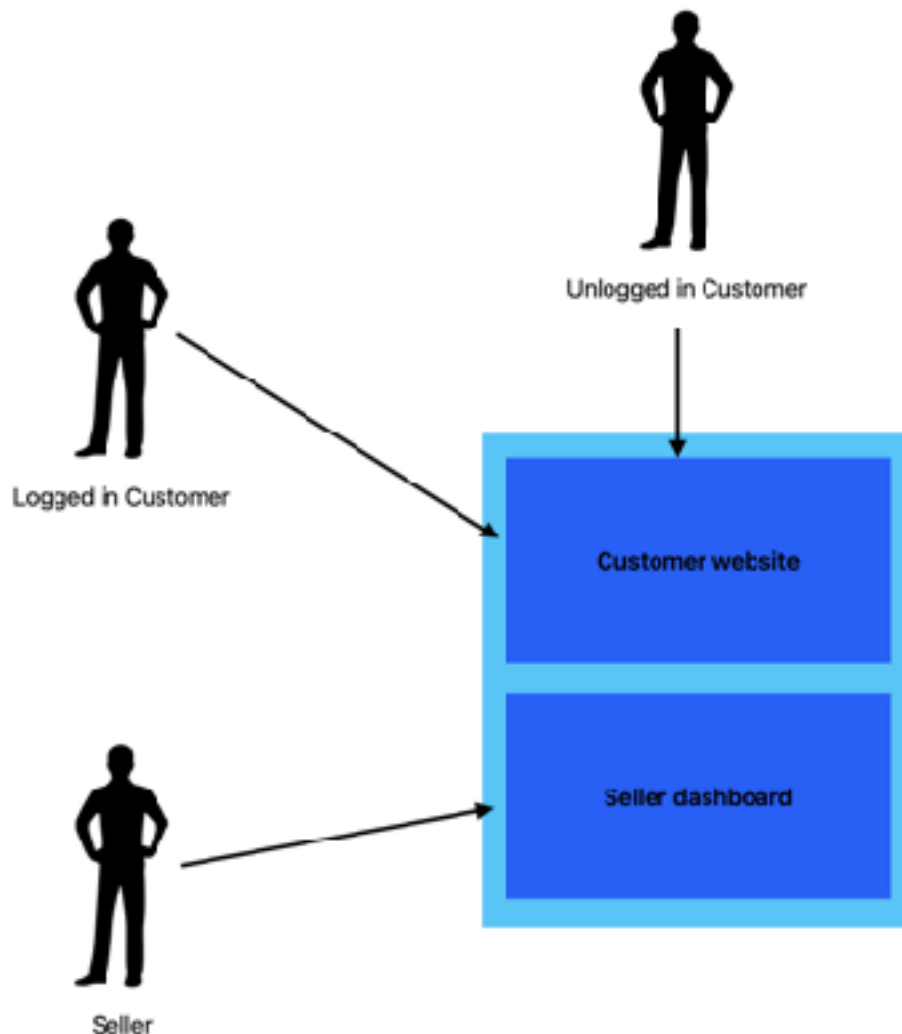
**myMag, an eMag clone**

**Author Isarescu Mihai**

# Scope of the project

Clone a popular marketplace that has complex functionality in order to have a strong grasp of full stack development. In a project of this scale the need to make configurable accounts for the sellers and customers is necessary. On myMag, like in the original website it clones, multiple sellers can sell on the platform, so that a customer has the choice if they want to buy from one vendor or another, you can buy from multiple vendors and see all the fees from all of them at checkout. Users can leave reviews or return the items. The client also has warranty tracker for the products ordered via the platform.

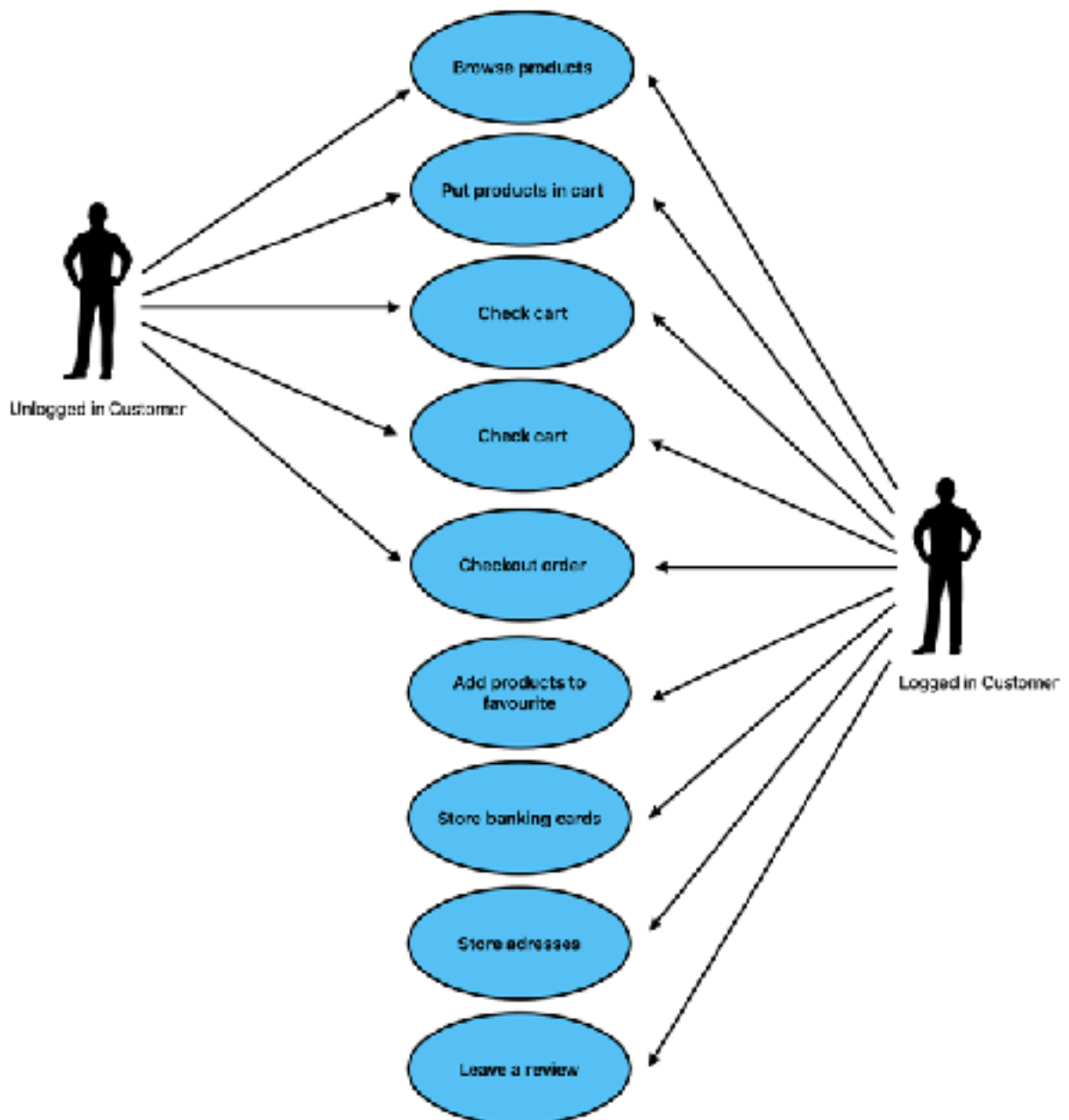
## Overall system description

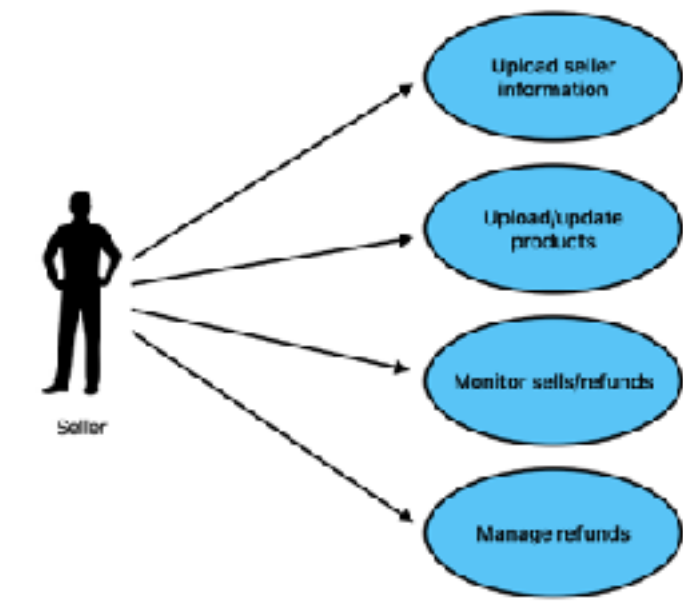


System Environment

The marketplace has 2 parts, the sellers and the customers. In this design the sellers has a separate dashboard where they can upload the products they sell and update that product information. The customers can browse the store, add items to cart, proceed to checkout, leave reviews, return ordered items, all the usual e-commerce stuff.

## General use case diagrams





# Requirements

## Functional Requirements

Present actors in functional requirements:

- Unlogged customer;
- Logged customer;
- Seller;

<b>Use Case Name</b>	Browse Products
<b>Ref</b>	Func_Browse 1.0
<b>Participants</b>	Unlogged customer, Logged customer
<b>Trigger</b>	Customer visits the website
<b>Default Path</b>	Main page presents the product category menu. It also presents a list of products based on most popular categories for unlogged customer and based on last visited categories for logged users
<b>Alternate Paths</b>	The customer can use the product category menu to search products by categories
<b>Exception Paths</b>	Customer leaves the site
<b>Other</b>	

<b>Use Case Name</b>	Add products to cart
<b>Ref</b>	Func_AddToCart 1.0
<b>Participants</b>	Unlogged customer, Logged customer
<b>Trigger</b>	Customer clicks 'add to cart' button
<b>Default Path</b>	Customer clicks the button from the product page
<b>Alternate Paths</b>	Customer clicks the button from the product card in the browse panel
<b>Exception Paths</b>	Customer removes the item by doing Func_UpdateCartItem
<b>Other</b>	

<b>Use Case Name</b>	Update products from cart
<b>Ref</b>	Func_UpdateCartItem 1.0
<b>Participants</b>	Unlogged customer, Logged customer
<b>Trigger</b>	Customer checks the cart by doing Func_CheckCart; Then Customer clicks either +/- buttons in the cart page to edit quantity; Or Customer clicks the x button in the cart page/dropdown to remove the item;
<b>Default Path</b>	Customer clicks the buttons from the cart preview page
<b>Alternate Paths</b>	Customer clicks the buttons from the cart preview dropdown
<b>Exception Paths</b>	Customer finished checked out the order by doing Func_Checkout
<b>Other</b>	+/- is not available in the cart dropdown, only in the preview cart page

<b>Use Case Name</b>	Check cart items
<b>Ref</b>	Func_CheckCart 1.0
<b>Participants</b>	Unlogged customer, Logged customer
<b>Trigger</b>	Customer hovers over 'my cart' label to expose the cart dropdown; Or Customer clicks the 'go to my cart' button to go to the cart preview page;
<b>Default Path</b>	Customer visits cart preview page
<b>Alternate Paths</b>	Customer hovers over the modal
<b>Exception Paths</b>	Customer finished checked out the order by doing Func_Checkout
<b>Other</b>	

<b>Use Case Name</b>	Checkout order
<b>Ref</b>	Func_Checkout 1.0
<b>Participants</b>	Unlogged customer, Logged customer
<b>Trigger</b>	Customer click 'proceed to checkout' button
<b>Default Path</b>	Customer clicks the button from the cart preview page; Customer is redirected to the checkout flow;
<b>Alternate Paths</b>	
<b>Exception Paths</b>	
<b>Other</b>	

<b>Use Case Name</b>	Add products to favourite
<b>Ref</b>	Func_AddFavourite 1.0
<b>Participants</b>	Logged customer
<b>Trigger</b>	Customer click 'add to favourites' button
<b>Default Path</b>	Customer clicks the button from the product page
<b>Alternate Paths</b>	Customer clicks the button from the product card in the browse panel
<b>Exception Paths</b>	Customer removes the item by doing Func_RemoveFavourite
<b>Other</b>	

<b>Use Case Name</b>	Check favourite items
<b>Ref</b>	Func_CheckFavourite 1.0
<b>Participants</b>	Logged customer
<b>Trigger</b>	Customer hovers over 'my favourites' label to expose the cart dropdown; Or Customer clicks the 'go to my favourites' button to go to the cart preview page;
<b>Default Path</b>	Customer visits favourite preview page
<b>Alternate Paths</b>	Customer hovers favourite the modal
<b>Exception Paths</b>	Customer removes the item by doing Func_RemoveFavourite
<b>Other</b>	

<b>Use Case Name</b>	Remove favourite items
<b>Ref</b>	Func_RemoveFavourite 1.0
<b>Participants</b>	Logged customer
<b>Trigger</b>	Customer checks the cart by doing Func_CheckFavourite; Then Customer clicks the 'remove favourite' button;
<b>Default Path</b>	Customer clicks the button from the favourite preview page
<b>Alternate Paths</b>	Customer clicks the button from the favourite preview dropdown
<b>Exception Paths</b>	
<b>Other</b>	

<b>Use Case Name</b>	Store banking cards
<b>Ref</b>	Func_StoreCard 1.0
<b>Participants</b>	Logged customer
<b>Trigger</b>	Doing Func_Checkout customer clicks 'save card' button
<b>Default Path</b>	Customer clicks the button during the checkout flow
<b>Alternate Paths</b>	
<b>Exception Paths</b>	
<b>Other</b>	

<b>Use Case Name</b>	Remove banking cards
<b>Ref</b>	Func_RemoveCard 1.0
<b>Participants</b>	Logged customer
<b>Trigger</b>	Customer clicks 'remove card' button
<b>Default Path</b>	Customer clicks the button in the view cards page
<b>Alternate Paths</b>	
<b>Exception Paths</b>	
<b>Other</b>	

<b>Use Case Name</b>	Store address
<b>Ref</b>	Func_StoreAddress 1.0
<b>Participants</b>	Logged customer
<b>Trigger</b>	Doing Func_Checkout customer clicks 'save address' button
<b>Default Path</b>	Customer clicks the button during the checkout flow
<b>Alternate Paths</b>	
<b>Exception Paths</b>	
<b>Other</b>	



<b>Use Case Name</b>	Remove address
<b>Ref</b>	Func_RemoveAddress 1.0
<b>Participants</b>	Logged customer
<b>Trigger</b>	Customer clicks 'remove address' button
<b>Default Path</b>	Customer clicks the button in the view addresses page
<b>Alternate Paths</b>	
<b>Exception Paths</b>	
<b>Other</b>	

<b>Use Case Name</b>	Add Review
<b>Ref</b>	Func_AddReview 1.0
<b>Participants</b>	Logged customer
<b>Trigger</b>	Customer clicks 'write review' button
<b>Default Path</b>	Customer clicks the button in the view order page
<b>Alternate Paths</b>	Customer clicks the button in the product page
<b>Exception Paths</b>	
<b>Other</b>	Reviews could be implemented to be a suggestion 1 week after the order has been delivered

<b>Use Case Name</b>	Remove Review
<b>Ref</b>	Func_RemoveReview 1.0
<b>Participants</b>	Logged customer
<b>Trigger</b>	Customer clicks 'remove review' button
<b>Default Path</b>	Customer clicks the button in the my reviews page
<b>Alternate Paths</b>	Customer clicks the button in the product page
<b>Exception Paths</b>	
<b>Other</b>	

<b>Use Case Name</b>	Add seller information
<b>Ref</b>	Func_AddSellerInfo 1.0
<b>Participants</b>	Seller
<b>Trigger</b>	Seller creates an account
<b>Default Path</b>	In the create account flow the seller introduces info
<b>Alternate Paths</b>	
<b>Exception Paths</b>	The seller cancels the process
<b>Other</b>	

<b>Use Case Name</b>	Update seller information
<b>Ref</b>	Func_UpdateSellerInfo 1.0
<b>Participants</b>	Seller
<b>Trigger</b>	Seller clicks 'update info' button
<b>Default Path</b>	Seller clicks the button in the my profile page
<b>Alternate Paths</b>	
<b>Exception Paths</b>	The seller cancels the changes
<b>Other</b>	

<b>Use Case Name</b>	Add product
<b>Ref</b>	Func_AddProduct 1.0
<b>Participants</b>	Seller
<b>Trigger</b>	Seller clicks add product button
<b>Default Path</b>	Seller clicks the button from the main dashboard; Then Seller completes the product information;
<b>Alternate Paths</b>	
<b>Exception Paths</b>	
<b>Other</b>	

<b>Use Case Name</b>	Update product
<b>Ref</b>	Func_UpdateProduct 1.0
<b>Participants</b>	Seller
<b>Trigger</b>	Seller clicks update product button
<b>Default Path</b>	Seller clicks the button from product card in the main dashboard; Then Seller completes the product information;
<b>Alternate Paths</b>	Seller clicks the button from the product page;
<b>Exception Paths</b>	Seller removes the product by doing Func_RemovProduct
<b>Other</b>	

<b>Use Case Name</b>	Remove product
<b>Ref</b>	Func_RemovProduct 1.0
<b>Participants</b>	Seller
<b>Trigger</b>	Seller clicks remove product button
<b>Default Path</b>	Seller clicks the button from product card in the main dashboard; Then Seller confirms deletion modal
<b>Alternate Paths</b>	Seller clicks the button from the product page;
<b>Exception Paths</b>	Seller does not confirm the deletion
<b>Other</b>	

TODO: think about tracking the sells and refunds.  
 TODO: think about implementing the refunds.