

CS341 Shopping Cart Functional Requirements

Group 4: Sal Skare, Jack Englund, David Wissink, and John Collins

September 27, 2018

1 About this document

This document describes the functional requirements for the software for an online e-commerce shopping cart software product.

2 Functional Requirements

2.1 Users

<i>Index:</i>	U.1
<i>Name:</i>	CreateUser
<i>Purpose:</i>	To create a new user in the database.
<i>Input parameters:</i>	Email, Password, Guest user ID, Type of user [<i>customer, admin</i>].
<i>Action:</i>	Creates a new entry in the database for the user, hashing the password. If the guest user ID is given, the guest user cart will be imported to the new user.
<i>Output parameters:</i>	Success or failure message.
<i>Exceptions:</i>	The user already exists, not all required fields were given, there was a database error, the user does not have permission to create a new admin.
<i>Remarks:</i>	It will be up to the front-end design to also log in the newly created user if that functionality is desired. Also, to create an admin user requires an existing admin to make the request, including a valid Access Token
<i>Cross references:</i>	AuthenticateUser

<i>Index:</i>	U.2
<i>Name:</i>	ModifyUser
<i>Purpose:</i>	To change or information about a user.
<i>Input parameters:</i>	Email, Access Token, User fields to change (<i>e.g. Name, Password, etc.</i>).
<i>Action:</i>	Updates the given fields for the user identified by their email address.
<i>Output parameters:</i>	Success or failure message.
<i>Exceptions:</i>	Email was not given, field to set is of an invalid type, there was a database error.
<i>Remarks:</i>	None.
<i>Cross references:</i>	AuthenticateUser
<hr/>	
<i>Index:</i>	U.3
<i>Name:</i>	AuthenticateUser
<i>Purpose:</i>	Authenticates the user to the server for a set period of time.
<i>Input parameters:</i>	Email, Password
<i>Action:</i>	Generates an Access Token for the given user.
<i>Output parameters:</i>	A temporary Access Token to be included in all requests that require elevated permission.
<i>Exceptions:</i>	The user does not exist, the password did not match the user, there was a database error.
<i>Remarks:</i>	The server will log the time the token was created and mark it as expired after a set length of time, requiring the user to log in again.
<i>Cross references:</i>	None.
<hr/>	
<i>Index:</i>	U.4
<i>Name:</i>	GetUser
<i>Purpose:</i>	Returns information about the user
<i>Input parameters:</i>	Email, Access Token
<i>Action:</i>	Queries the database for the user, then returns all available fields.
<i>Output parameters:</i>	An <i>User object</i> , containing, <i>email</i> , <i>name</i> , <i>type of user</i> .
<i>Exceptions:</i>	The user is not authenticated, there was a database error.
<i>Remarks:</i>	None.
<i>Cross references:</i>	None.

<i>Index:</i>	U.5
<i>Name:</i>	AddItemToCart
<i>Purpose:</i>	Adds the item to the user's cart.
<i>Input parameters:</i>	Email, Access Token, Item ID, count
<i>Action:</i>	Adds the number of items given by the count to the specified user's cart.
<i>Output parameters:</i>	A success or failure message.
<i>Exceptions:</i>	The user is not authenticated, the item does not exist, there was a database error.
<i>Remarks:</i>	None.
<i>Cross references:</i>	None.
<hr/>	
<i>Index:</i>	U.6
<i>Name:</i>	RemoveItemFromCart
<i>Purpose:</i>	Removes the item from the user's cart.
<i>Input parameters:</i>	Email, Access Token, Item ID, count
<i>Action:</i>	Removes the number of items given by the count from the specified user's cart.
<i>Output parameters:</i>	A success or failure message.
<i>Exceptions:</i>	The user is not authenticated, the item does not exist, there was a database error.
<i>Remarks:</i>	None.
<i>Cross references:</i>	None.
<hr/>	
<i>Index:</i>	U.7
<i>Name:</i>	GetCart
<i>Purpose:</i>	Returns a list of all items in a user's cart.
<i>Input parameters:</i>	Email, Access Token
<i>Action:</i>	Creates a list of item IDs and count in an authenticated user's cart.
<i>Output parameters:</i>	A list of Item IDs and the count for each item.
<i>Exceptions:</i>	The user is not authenticated, there was a database error.
<i>Remarks:</i>	None.
<i>Cross references:</i>	None.

2.2 Guests

<i>Index:</i>	G.1
<i>Name:</i>	AddItemToGuestCart
<i>Purpose:</i>	Adds the item to the guest user's cart.
<i>Input parameters:</i>	Guest user's id (A randomly generated GUID), Item ID, item count.
<i>Action:</i>	Adds the number of items given by the count to the specified guest user's cart. If this is the first time calling this function, an entry for the guest user will first be created in the database.
<i>Output parameters:</i>	A success or failure message.
<i>Exceptions:</i>	The item does not exist, there was a database error.
<i>Remarks:</i>	The front-end should store the guest user ID in a browser cookie so that subsequent requests can be made.
<i>Cross references:</i>	None.

<i>Index:</i>	G.2
<i>Name:</i>	RemoveItemFromGuestCart
<i>Purpose:</i>	Removes the item from the guest user's cart.
<i>Input parameters:</i>	Guest user's id, Item ID, count
<i>Action:</i>	Removes the number of items given by the count from the specified guest user's cart.
<i>Output parameters:</i>	A success or failure message.
<i>Exceptions:</i>	The guest does not exist, the item does not exist, there was a database error.
<i>Remarks:</i>	None.
<i>Cross references:</i>	None.

<i>Index:</i>	G.3
<i>Name:</i>	GetGuestCart
<i>Purpose:</i>	Returns a list of all items in a guest user's cart.
<i>Input parameters:</i>	Guest user's ID
<i>Action:</i>	Creates a list of item IDs and count in a guest user's cart.
<i>Output parameters:</i>	A list of Item IDs and the count for each item.
<i>Exceptions:</i>	The guest user does not exist, there was a database error.
<i>Remarks:</i>	None
<i>Cross references:</i>	None.

2.3 Promotions

<i>Index:</i>	P.1
<i>Name:</i>	CreatePromotion
<i>Purpose:</i>	To create a promotion for users to apply to items
<i>Input parameters:</i>	Name, Discount Code, Discount Type [<i>bogo, percent</i>], <i>Percent, End date</i>
<i>Action:</i>	Creates a new promotion in the database.
<i>Output parameters:</i>	Success or Failure message.
<i>Exceptions:</i>	The discount code already exists, invalid end date
<i>Remarks:</i>	The promotion should only be valid until the end date, the code can only be used once per user.
<i>Cross references:</i>	None.

<i>Index:</i>	P.2
<i>Name:</i>	EndPromotion
<i>Purpose:</i>	To forcefully end a promotion
<i>Input parameters:</i>	Discount Code [<i>Identifier</i>].
<i>Action:</i>	Ends given promotion from database
<i>Output parameters:</i>	Success or Failure message.
<i>Exceptions:</i>	The discount code doesn't exist, the promotion is already ended
<i>Remarks:</i>	Edit a field in the database, 'ongoing', to mark it as done. Otherwise could just change the end date to the current datetime.
<i>Cross references:</i>	None.

<i>Index:</i>	P.3
<i>Name:</i>	GetPromotion
<i>Purpose:</i>	Returns the type percentage of the promotion.
<i>Input parameters:</i>	Discount Code [<i>Identifier</i>].
<i>Action:</i>	Gets a promotion based on the discount code given.
<i>Output parameters:</i>	Success or Failure message.
<i>Exceptions:</i>	The discount code doesn't exist, the promotion is over.
<i>Remarks:</i>	This should be used during ApplyPromotion
<i>Cross references:</i>	None.

<i>Index:</i>	P.4
<i>Name:</i>	ApplyPromotion
<i>Purpose:</i>	Applies the given promotion to a user's cart
<i>Input parameters:</i>	Discount Code [<i>Identifier</i>].
<i>Action:</i>	Applies the discount percentage and type to the user's existing cart, modifies their total.
<i>Output parameters:</i>	Success or Failure message.
<i>Exceptions:</i>	The discount code doesn't exist, the promotion is over.
<i>Remarks:</i>	This should be used when a user enters a discount code at checkout.
<i>Cross references:</i>	None.

2.4 Product Page

<i>Index:</i>	PP.1
<i>Name:</i>	LoadProduct
<i>Purpose:</i>	Loads product information
<i>Input parameters:</i>	Unique Product Id
<i>Action:</i>	Returns product information like name, price, quantity, image, etc.
<i>Output parameters:</i>	Success or Failure message.
<i>Exceptions:</i>	Invalid PID.
<i>Remarks:</i>	None.
<i>Cross references:</i>	None.

<i>Index:</i>	PP.2
<i>Name:</i>	NewProduct
<i>Purpose:</i>	Creates a new product entry in the db.
<i>Input parameters:</i>	Name, Category, Price, Seller, Initial Quantity, Max Quantity, Reviews, Images
<i>Action:</i>	Creates a new entry in the product database.
<i>Output parameters:</i>	Success or Failure message.
<i>Exceptions:</i>	Max Quantity is less than 1, Image files are too big.
<i>Remarks:</i>	None.
<i>Cross references:</i>	None.
<hr/>	
<i>Index:</i>	PP.3
<i>Name:</i>	EditProduct
<i>Purpose:</i>	Edits an already existing product
<i>Input parameters:</i>	PID, Name, Category, Price, Seller, Initial Quantity, Max Quantity, Reviews, Images
<i>Action:</i>	Edits the row of PID in the database.
<i>Output parameters:</i>	Success or Failure message.
<i>Exceptions:</i>	PID is not valid, then same exceptions as NewProduct
<i>Remarks:</i>	None.
<i>Cross references:</i>	None.
<hr/>	
<i>Index:</i>	PP.4
<i>Name:</i>	DeleteProduct
<i>Purpose:</i>	Removes a product from the database
<i>Input parameters:</i>	PID
<i>Action:</i>	Removes the row of the given PID from the database.
<i>Output parameters:</i>	Success or Failure message.
<i>Exceptions:</i>	PID does not exist.
<i>Remarks:</i>	None.
<i>Cross references:</i>	None.
<hr/>	

2.5 Searching

<i>Index:</i>	S.1
<i>Name:</i>	SearchByName
<i>Purpose:</i>	Searches for any products containing the name they entered.
<i>Input parameters:</i>	Name or partial name.
<i>Action:</i>	Searches the database for any product that contains the input parameter
<i>Output parameters:</i>	Success or Failure message.
<i>Exceptions:</i>	No products meet the parameters.
<i>Remarks:</i>	None.
<i>Cross references:</i>	None.

<i>Index:</i>	S.2
<i>Name:</i>	SearchByCategory
<i>Purpose:</i>	Searches for any products in the given category
<i>Input parameters:</i>	Category
<i>Action:</i>	Searches the database for any products in the category
<i>Output parameters:</i>	Success or Failure message.
<i>Exceptions:</i>	No products meet the parameters.
<i>Remarks:</i>	None.
<i>Cross references:</i>	None.

<i>Index:</i>	S.3
<i>Name:</i>	FilterSearchResult
<i>Purpose:</i>	Filter a result set based on the parameters given.
<i>Input parameters:</i>	Price, Seller, Reviews/Rating, Category.
<i>Action:</i>	Returns all products that follow the parameters given.
<i>Output parameters:</i>	Success or Failure message.
<i>Exceptions:</i>	No products meet the parameters.
<i>Remarks:</i>	None.
<i>Cross references:</i>	None.

2.6 Functional Requirements for the Checkout Process

<i>Index:</i>	CO.1
<i>Name:</i>	SelectPaymentOption
<i>Purpose:</i>	To allow a guest or registered user to select a payment option to apply to the cart balance.
<i>Input parameters: none Action:</i>	Setups a payment method to be applied to the shopping cart balance for registered user and guest. Selects an existing payment method for registered user. Payment will be handled through Paypal user ID is given, the guest user cart will be imported to the new user.
<i>Output parameters:</i>	Success or failure message.
<i>Exceptions:</i>	Cart has a balance of 0.00.
<i>Remarks:</i>	This method is used only during the checkout routine. Adjusting payment for a registered user can be done with the corresponding user setting functions. Should be incorporated into user friendly GUI.
<i>Cross references:</i>	CO.4

<i>Index:</i>	CO.2
<i>Name:</i>	SelectShippingOption
<i>Purpose:</i>	To allow a guest or registered user to select a shipping address for their order.
<i>Input parameters:</i>	First Name, Last Name, Street Address, Apt. Number, City, State, Zip Code, Phone Number, Shipping Class
<i>Action:</i>	Initialize a new shipping method to be applied. Select an existing shipping method for registered guests. Validates the user's data.
<i>Output parameters:</i>	Success/Fail Notification
<i>Exceptions:</i>	The supplied data is invalid.
<i>Remarks:</i>	This method is used only during the checkout routine. Adjusting payment for a registered user can be done with the corresponding user setting functions. User friendly form recommended.
<i>Cross references:</i>	CO.4

<i>Index:</i>	CO.3
<i>Name:</i>	ApplyPromocode
<i>Purpose:</i>	To allow both users to apply promocodes to their shopping cart balance.
<i>Input parameters:</i>	Promocode
<i>Action:</i>	Validates promocode. Applies promocode to shopping cart to alter the balance of the shopping cart (for both registered and guest users)
<i>Output parameters:</i>	Amount Reduced
<i>Exceptions:</i>	Promocode is invalid
<i>Remarks:</i>	This method is used only during the checkout routine. Should be incorporated into user friendly GUI.
<i>Cross references:</i>	CO.4
<i>Index:</i>	CO.4
<i>Name:</i>	SubmitCheckout
<i>Purpose:</i>	To submit the cart, along with payment, for processing by the admin
<i>Input parameters:</i>	Cart, Payment Method, Shipping Method
<i>Action:</i>	Submits the order for processing by the admin. Reduces the quantity in stock of each item. Generates user notification. Generates admin notification.
<i>Output parameters:</i>	Success/Fail Notification
<i>Exceptions:</i>	Cart is empty. Payment or shipping is invalid.
<i>Remarks:</i>	This method is used to conclude the checkout process. Should be incorporated into user friendly GUI.
<i>Cross references:</i>	CO.1, CO.2, CO.3, CO.5
<i>Index:</i>	CO.5
<i>Name:</i>	GenerateAdminNotification
<i>Purpose:</i>	To create an entry in the administration's log so the order may be processed
<i>Input parameters:</i>	User Name, User Email, Items, Shipping Method
<i>Action:</i>	Notifies appropriate admin. Creates an entry in the admin outgoing processing log
<i>Output parameters:</i>	none
<i>Exceptions:</i>	none
<i>Remarks:</i>	This method is for internal use only. No User will have access to this function.
<i>Cross references:</i>	CO.4

2.7 Functional Requirements for Out Going Product Management

<i>Index:</i>	PM.1
<i>Name:</i>	DisplayOrders
<i>Purpose:</i>	To display all of the out going orders.
<i>Input parameters:</i>	none
<i>Action:</i>	Displays the orders in chronological order. Displays the status of the order. Displays the items purchased and their quantities. Displays additional data relevant to the user and the order.
<i>Output parameters:</i>	Admin GUI
<i>Exceptions:</i>	none
<i>Remarks:</i>	This method is for admin use, specifically those in charge of getting orders in the mail.

Cross references:

<i>Index:</i>	PM.2
<i>Name:</i>	GenerateShippingLabel
<i>Purpose:</i>	To fetch the shipping address of an outgoing order
<i>Input parameters:</i>	order ID
<i>Action:</i>	Creates a shipping label based on user data from the checkout process.
<i>Output parameters:</i>	A properly formatted shipping label
<i>Exceptions:</i>	none
<i>Remarks:</i>	This method is used by the admin only.
<i>Cross references:</i>	CO.2

<i>Index:</i>	PM.3
<i>Name:</i>	LocateItems
<i>Purpose:</i>	To provide locations of all items in an order.
<i>Input parameters:</i>	Order ID
<i>Action:</i>	Generates a document, which will contain each item from a given order and it's location on site
<i>Output parameters:</i>	Location of items document
<i>Exceptions:</i>	none
<i>Remarks:</i>	this method is for admin use only
<i>Cross references:</i>	none
<hr/>	
<i>Index:</i>	PM.4
<i>Name:</i>	UpdateOrderStatus
<i>Purpose:</i>	To indicate an items progress from checkout to delivery
<i>Input parameters:</i>	none
<i>Action:</i>	Advances an items status. Updates admin log
<i>Output parameters:</i>	Success/Fail notification
<i>Exceptions:</i>	none
<i>Remarks:</i>	This method is for admin use only.
<i>Cross references:</i>	none
<hr/>	
<i>Index:</i>	PM.5
<i>Name:</i>	CancelOrder
<i>Purpose:</i>	Allow admin to cancel a specific order
<i>Input parameters:</i>	Order ID, Reason Code
<i>Action:</i>	Cancels the order. Removes the order from the outgoing log. Generates appropriate response based on reason code(back order, refund, damaged merchandise, etc.).
<i>Output parameters:</i>	user notification
<i>Exceptions:</i>	Invalid reason code
<i>Remarks:</i>	this method is for admin use only. Reason codes should perform appropriate actions such as refunding orders.
<i>Cross references:</i>	none
<hr/>	

3 Assumptions:

1. Users are identified by their email addresses.
2. If a user wants to change their email address, they will need to create a new account.
3. All payment info and address data will be handled through PayPal.
4. The administrative user duties will include: maintenance of the inventory/users and control of outgoing orders.
5. The customer who uses this product will create a custom front-end for their business, using the documentation and example front end as a guide.