

CS341 Shopping Cart Functional Requirements

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

September 26, 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, name, 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.

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.