Account Related:

Forgot username/password: (PC and mobile views)

Send email to user containing one-time key that allows access

To change password screens

1. User is at the login screen
2. User clicks on “forgot password” screen
3. User is shown a screen asking for email address and enters it
4. Email address is checked to see if it is in database. If not, User is shown a ‘not on file screen”
5. Email is sent to address provided containing one-time key
6. User clicks on the one-time key or enters it into browser
7. User is shown a new password screen
8. New password is checked to see if it is secure
9. Password is saved
10. User is shown a screen prompting login with new password

$User->getUserProfile($userName)

$User->editUserProfile($user, $editInfo)

$User->checkPasswordSecure($password)

$Message->sendEmail($userEmail)

$User->updatePassword($userId, $newPassword)

Sign Up: (PC and mobile views)

Register user and company, capturing all appropriate profile info. This

Information can be captured from Google or Facebook using an oath-based

Register using Facebook/Google info. Subsequent screens capture shop profile.

Prospective users will be asked to identify company affiliation by selecting

from a list of already-registered companies, or entering a new company profile.

1. User is at Login Screen and clicks on the register link. If the user clicks on register via facebook or google, they will go through the Oauth login for the appropriate service. What info that is available will be captured, and they will be ushered to the profile entry screens with that info pre-entered in the appropriate fields.
2. User goes to profile entry screen. This screen has entry’s for all the usual profile info as well as a buyer/seller/customer choice, a company profile link, and password entry fields
3. If the user chooses customer, the a pro-forma company record is entered and permissions are set that prevent access to the wholesale stuff
4. If the user chooses buyer or seller, the next screen they see will be a identify company screen in which the user can either select a company with which they are affiliated or go to a new company profile screen. If the user selects a company, a message is sent to the store-admin asking for approval before the user has order permissions. This message also has a pulldown specifying the user’s company role.
5. The company profile screen includes billing info and shipping info links if the user is a seller. These must be entered if the user is given order processing permissions.

$User->addUser($info)

$User->addUserToCompany($user, $company,$companyRole)

$User->oathFacebookRegInfo($facebookId)

$User->oathGoogRegInfo($googleId)

$ShipInfo->addShipInfo($companyId, $newInfo)

$BillInfo->addBillInfo($companyId, $newInfo)

Get approval/rejection: (PC and mobile views)

Registration requests come to admin as a message of a particular type.

1. Admin is in message center screen and clicks on message identified as a registration request.
2. The message contains a summary of the profile info and a ‘show details’ button as well as an approve/decline button. The message also has a ‘remove user’ button which terminates the process.
3. If the admin clicks on the show details button, the admin is shown the profile screen in view-only mode
4. The message also has an entry area for a reply
5. If the admin approves, an email and a message are both sent to user. The email has a confirm key the user has to click on the confirm registration. When confirm link is activated, te user’s permissions are changed.
6. If the admin declines, email goes to user with a link to access further messageing. The email can contain info the admin types in the the message entry field contained in the request message.

$User->changeUserRole($user, $role)

$Message->getMessagesByType($userId, $filter)

$Message->sentEmailToUser($emailAddress)

$Message-addNewMessage($msginfo)

$User->removeUser($userId)

Ask questions about registration: (PC and mobile views)

Questions come as messages of a particular type.

1. Messages from applying users appear in the admin’s message center as a part of a thread.
2. Admin has option to also send emails
3. Admin’s reply screen has click boxes for frequently asked questions and requests for further information.

$Message->addNewMessage($msginfo)

$Message->sentEmailToUser($emailAddress)

Create/edit/store profile: (PC and mobile views)

User enters profile information for store. Creator of store profile becomes

store-admin until that is changed. Store-admin ok’s registration requests

from prospective users identifying affiliation with that store, and sets

company role for these people.

(see description of initial profile entry above Sign up for initial profile entry)

1. In PC interface, user clicks on “My Profile” tab at top of screen
2. In phone interface, User clicks on profile icon at the bottom.
3. Profile Screen is displayed with the available info filled in the respective fields.
4. User touches or clicks on save button to update information

$User->getUserProfile($userName)

$User->addUser($info)

$User->editUserProfile($user, $editInfo)

Preview Store Profile:

Shows all the info the system has on that store. (PC view only)

Company->getCompanyByName($companyName) - change: return obj not id

User is an admin:

1. User clicks on the admin tab and has the tabs expanded to display the shops tab. User clicks on that.
2. User is shown a scrolling list of all stores with the usual find box.
3. User finds the appropriate company and clicks on it
4. User is shown store profile in read-only mode
5. At the bottom of the screen are buttons for Users, Orders, Catalogs ??
6. If the user clicks on users, is shown a scrollable list of users associated with that shop.
7. If the use drills into users, is ahown a profile page for that user, At the bottom is a button that will drill into the user’s order activity.
8. If the user clicks on Orders is shown a scrollable list of order activity.

Add/Edit payment Info:

Sets a company’s payment pref’s and identifies payment processor.

BillInfo->addBillInfo($companyId, $newInfo)

BillInfo->editBillInfo($billInfoId, $editInfo)

BillInfo->removeBillInfo($billInfoRecordId)

BillInfo->getBillInfoById($billInfoId)

BillInfo->getBillInfoForCompany($companyId)

(BillInfo may need chaging when cc processing is determined)

Browse:

View Shop Profile:

View all info on a shop including all their catalogs w drill-down

To show catalog contents.

(See description above)

$Company->getCompanyByName($companyName)

$Collections->getCollectionsByCompanyId($companyId)

Add/Post Shop to favorites:

Shop is added to user’s favorites list.

1. User is at any screen displaying a shop profile
2. User clicks on ‘add to favorite shops’ button

$User->addShopToFavorites($companyId)

Share post/shop (need discussion):

1. User is at order screen, shop, or specials screen
2. User clicks on ‘share with others’ button
3. Dialog appears – giving options – share with ‘favorites, share with particular shops (option to select shops pulldown with multi-selects)
4. Message entry box is included
5. User clicks on share button
6. Message is sent to intended recepients.

$Message->sendMessageToGroup($msgInfo, $groupId)

Message Shop:

Message is created that is sent to shop’s admin role person.

1. User (buyer usually) is at product screen
2. Clicks on ‘message this shop’ button
3. Dialog appears with message entry box
4. User clicks on send button
5. Message is dispatched to shop’s shop-admin role.

$User->getUserInCompanyRole($user, $copmpanyRole)

$Message->addNewMessage($msginfo)

Take Action:

Shop WholesaleCatalog:

Catalogs identified as wholesale catalog type are shown for a company.

Mobile:

1. User touches the find icon at the bottom
2. Prompt – Shop or Product
3. User selects Shop
4. User is prompted “Shop Name, Product Category, Location, Favorites, Specials or Marketplace
5. If user selects Shop Name, an entry field appears, user types in shop name, and shop is searched for. If not found, dialog box – shop not found, and entry field to try again, or button to go back.
6. If user selects Product category, he/she is ushered through the category tree and when they arrive at a leaf, a list of shops offering the desired product categories is displayed. If there are no shops selling that category, a dialog box indicating that is displayed and a button to go back. At each level, if there are shops selling products in the wider category, a button is displayed to “see shops”
7. If the user selects location, an entry field for zipcode is presented along with a pulldown specifying how close. User enters a zipcodee, and a list of shops is presented.
8. If the user selects marketplace, a scrollable list of marketplaces is presented.
9. If the user selects favorites, a selection list of shops designed as his/her facvorites is displayed.
10. After getting to the shop, the user is shown a list of products available from that shop’s current catalog.
11. At this screen, there will be an ‘add to favorites’ button.
12. If the shop has more than one catalog (rare) there will be a pulldown to select the desired catalog.

$Collection->getCollectionByCompanyAndType($company, $collectionType)

Request Sample:

Message is sent to shop-admin requesting sample.

1. Buyer is at product screen
2. Clicks on ‘message this shop’ button
3. Dialog appears with message entry box
4. User clicks on send button
5. Message is dispatched to shop’s shop-admin role with a r equest for sample header.

$Message->addNewMessage($msginfo)

Buy One Sample/Buy Product(s):

Order is placed for sample product(s). Same as any order except it is q1.

1. Buyer is at product screen
2. Buyer clicks on buy one sample button
3. Order screen is displayed with all the order info filled in
4. Buyer clicks on order button
5. If buyer doesn’t have shipinfo filled in a dialog appears to capture the ship info.
6. Buyer is asked for shipping type
7. Buyer is ushered to the payment processing service
8. Order is placed with message to seller created.

$Orders->startNewOrder($company, $user)

$Orders->addProductToOrder($orderId, $productToAdd, $optionsSelected, $thisShipTypeId, $quantity)

$Orders->getIncompleteOrders($companyId)

$Orders->placeOrder($orderId)

$Orders->selectPayment($orderId) - Need to finalize payment processing

$Orders->selectShippingOption($orderId)

Browse/Select Product:

Products are displayed with option to add product to order. ~~Option to see~~

~~similar(competing) products from same categories.~~

1. User touches the find icon at the bottom
2. Prompt – Shop or Product
3. User selects product
4. User goes down the category tree. At every juncture, user can click on ‘see products’ button. A number is provided within the button indicating how many products are in that category.
5. At the l;eaf level, user is ushered to a scrolled list of products. Mini icons appear next to the product name. Upon mounse-over, the mini-icons blow up to bigger pictures
6. User clicks on the picture and is ushered into the product details screen.

$Product->getAllSellableProductsByCategory($category)

$Product->getCompanyProducts($companyId)

$Product->getSimilarProducts($productId)

Adjust color/type:

During the adding a product to an order – can set options (color size etc)

1. User is at a line-item screen in the order process
2. Pulldowns will appear for each set of the options that have been selected from the default options when the product was originally entered.
3. If user fails to select an option prior to clicking the ‘add’ button, then a dialog screen will appear to get that information.

$Options->getDefaultOptionsForProducttype($productType)

$Orders->addProductToOrder($orderId, $productToAdd, $optionsSelected, $thisShipTypeId, $quantity)

Confirm Order: (what does “add optional charge mean?)

Confirm the placement of the order, and forward it to seller.

1. User has entered line items and selected shipping information and clicks on ‘place order button.
2. If the user has never entered their shipping profile, they are prompted to do so
3. Order summary screen is displayed. This screen includes a message to seller text entry.
4. User clicks on ‘confirm order information.
5. Status of order is changed to ‘order placed’. If a message to seller was entered in the confirmation screen, that message is sent to the seller.

$Orders->placeOrder($orderId)

$Orders->selectPayment($orderId) - Need to finalize payment processing

$Orders->selectShippingOption($orderId)

Add note to seller:

Seller is sent a message referencing the order.

(see above)

$Message->addNewMessage($msginfo)

Save Order In Progress:

Order status set to in-progress.

1. New order has been initiated. As a result, order’s status is ‘in-progress’
2. The order will appear on the buyer’s ‘my orders’ list with an ‘in-progress’ status indicated.

$Orders->startNewOrder($company, $user)

$Orders->addProductToOrder($orderId, $productToAdd, $optionsSelected, $thisShipTypeId, $quantity)

$Orders->getIncompleteOrders($companyId)

Cancel Order:

Order status set to canceled. Order not deleted and can be revived later

1. User is at an order entry screen
2. User click on the ‘cancel order’ button
3. User is shown an order summary screen with an ‘are you sure’ dialog.
4. User clicks on ‘ok’ button
5. Order’s status is set to ‘canceled’ and no longer appears on the user’s ‘My Orders’ list.

$Orders->setStatusCanceled($orderId)

Return To Saved Order:

List to canceled orders is displayed. Order can be set to status in-progress

1. User is at ‘My Orders’ screen
2. User clicks on ‘show canceled orders’
3. User selects order he/she is interested in
4. User is shown order summary screen
5. User clicks on ‘re-activate this order’ button
6. Order’s status is changed to ‘in-progress’
7. Order now appears on the regular ‘my orders’ screen.

$Orders->getLineItems($orderId)

$Orders->getOrderAll($orderId)

Orders-getCanceledOrders($companyId)

Find Joonley Sellers:

List of companies either by location or product types shown.

$Company->findCompaniesByLocation($location)

$Company->findCompaniesByProductTypes($productTypes)

1. User touches the find icon at the bottom
2. Prompt – Shop or Product
3. User selects Shop
4. User is prompted “Shop Name, Product Category, Location, Favorites, or Marketplace
5. If user selects Shop Name, an entry field appears, user types in shop name, and shop is searched for. If not found, dialog box – shop not found, and entry field to try again, or button to go back.
6. If user selects Product category, he/she is ushered through the category tree and when they arrive at a leaf, a list of shops offering the desired product categories is displayed. If there are no shops selling that category, a dialog box indicating that is displayed and a button to go back. At each level, if there are shops selling products in the wider category, a button is displayed to “see shops”
7. If the user selects location, an entry field for zipcode is presented along with a pulldown specifying how close. User enters a zipcodee, and a list of shops is presented.
8. If the user selects marketplace, a scrollable list of marketplaces is presented.
9. If the user selects favorites, a selection list of shops designed as his/her facvorites is displayed.

Find/Launch Seller Catalogs:

Get catalogs that are open for sale of a specific company

(see above ‘Shop WholesaleCatalog’)

$Collections->getCollectionsByCompanyId($companyId)

Place digital order at show:

Capture barcode from camera. Include in favorites list or order.

1. User is at a show looking over products at a Joonley seller’s both. The user notices a product he/she is interested in. There is a label on the product with a barcode.
2. User clicks on photo icon, pointing camera at product’s barcode
3. If user is not at order screen with an in-progress order, a dialog appears asking ‘add product to order’ or ‘add product to favorites’.
4. If the response is to add product to favorites, the product is added to favorites and the user goes back to whatever he/she was doing previously.
5. If the user is at an order screen, the camera captures the barcode, looks up the product, and adds it as a line item to the order.
6. User completes the options and quantity info
7. User clicks on ‘add item to order’ button.

$Company->findCompanyFromCameraBarcode()

$Product->findProductFromCameraBarcode()

$Orders->addProductFromCameraBarcode()

$Orders->placeOrder($orderId)

Save shop to favorites:

Add shop to favorites list.

1. User is at any screen showing shop profiles or shop catalog
2. User click on ‘add shop to favorites’ button
3. Shop is added to favorites list

$Company->saveCompanyToFavorites($companyId)

$Orders->duplicateOrder($orderId)

Manage Orders:

View existing orders with specified status. The ‘My Orders’ screen is a scrollable list of existing orders. At the top, there is a pulldown indicating filter by order status. At the bottom is a total value field reflecting the totals associated with a particular filter choice. The value field is only shown for open, confirmed, or saved orders. Individual orders can be clicked upon to view the details of a particular order

View Open Orders:

View Confirmed Orders:

View Shipped Orders:

View Declined Orders:

View Saved Orders:

$Orders->getOrdersByStatus($companyId, $status)

$Orders->getCompanyOrdersByStatus($company, $status)

$Orders->getUserOrdersByStatus($user, $status)

$Orders->getLineItems($orderId)

See Total Value of Orders:

Total up the value of all pending orders.

Orders->getValueOfSavedOrders($companyId)

See Total Saved:

See all the orders that have been saved but not yet finalized.

$Orders->getOrderTotals($company, $status)

Add product to possibly purchase.

1. User is at a product screen
2. Clicks on ‘add to favorites button’
3. Pulldown appears asking which favorites list
4. When the list if selected, the product is added to the appropriate list.

Add product to a list of products buyer is thinking or odering.

1. User is looking at a product screen
2. User clicks on favorites button
3. Dialog appears to select which favorites list
4. User selects a list
5. Product is added to that list

Order->addToFavorite(productId, $quantity)

Get value of possible purchase list

Total up value of possible purchases.

1. User is looking at a favorites list
2. User presses totals button
3. Dialog appears requesting a default quantity
4. User can go into each item and enter a quantity
5. User presses ‘see totals’ button
6. A totals page is shown

Order->getValueOfFavorites($company)

Open Orders – message seller:

Send message to seller of open order.

1. User is on a product page
2. User clicks on message icon
3. Message field appears
4. User types in message and clicks send button

$Message->addNewMessage($msginfo)

Open Order – cancel an open order:

Cancel an open order and message the seller.

1. User is at the Order screen
2. User clicks on the cancel order button
3. Dialog opens with an ‘are you sure ?’ prompt and a message entry box
4. User enters message and clicks on ‘cancel this order’button
5. Order status is changed to canceled, and message is sent to seller.

$orders->setStatusCanceled($orderId)

$Orders->removeStartingOrder($orderId)

$Message->addNewMessage($msginfo)

Open Order – edit open order:

Edit the details of an order before it’s placed – options,items, terms etc.

1. User is at Order screen
2. User clicks on edit button
3. Order-level properties – shipping info are displayed in editable fields
4. User can click on any one of the line items, bringing up an editable screen where any of the details of the line item can be changed
5. User can click on the ‘remove this item’ to remove the item from the oder. There will be an ‘are you sure dialog’
6. Back at rthe order screen, there will be a ‘save changes’ button that will cause the edited order to be saved.
7. There will also be an ‘add line item’ button that allows the user to add items to the order.

$Orders->editOrder($orderId)

$Orders->removeLineItem($OrderId, $lineItemId)

$Orders->addProductToOrder($orderId, $productToAdd, $optionsSelected, $thisShipTypeId, $quantity)

Confirmed Orders – message seller:

Shipped Orders – message seller:

Declined Orders- message seller:

Message the seller.

$Message->addNewMessage($msginfo)

1) User is at an Order screen

2) User clicks on the mail icon at the bottom of the screen

3) Message box appears. User types in message. Hits send. Details of the order are attached to the message

Confirmed Orders – re-order:

Shipped Orders – re-order:

Duplicate order offering opportunity to edit quantity, options etc.

1. User is at the order screen
2. User clicks on the ‘re-order’ button
3. The order is duplicated in its entirety and displayed as an open editable order (see edit order above)
4. User can edit order as if it were any open order
5. User can place the order by clicking on the ‘place order’ button.

$Orders->getOrderWithLineItems($orderId)

$Orders->duplicateOrder($orderId)

$Orders->placeOrder($orderId)

Declined Orders – edit order:

Edit a declined order, set items, terms, options etc. Can re-activate it.

1. User is at orders list screen
2. User uses pulldown to select ‘declined orders’
3. List of declined orders is displayed – user clicks on one of them.
4. Order is displayed
5. User clicks on ‘re-active this order’ button
6. Order status is changed to ‘open – in progress’
7. User can edit the order (see edit order above)
8. User can re-submit the order, and include a message to seller.

$Orders->editOrder($orderId)

$Orders->removeLineItem($OrderId, $lineItemId)

$Orders->addProductToOrder($orderId, $productToAdd, $optionsSelected, $thisShipTypeId, $quantity)

Declined Orders – clear order:

Remove a declined order from system entirely.

1. User is at orders list screen
2. User uses pulldown to select ‘declined orders’
3. List of declined orders is displayed – user clicks on one of them.
4. Order is displayed
5. User clicks on ‘clear this order’ button
6. Order is removed from the system.

$Orders->removeOrder($orderId)

Saved Orders – cancel order:

Set status of in-progress order to canceled.

1. User is at the orders screen
2. User employs pulldown to select select ‘open in-progress’ orders
3. List of open orders is displayed
4. User selects the appropriate order
5. User clicks on the ‘cancel this order’ button at the bottom
6. Order’s status is set to canceled.

$orders->setStatusCanceled($orderId)

$Orders->removeStartingOrder($orderId)

Saved Orders – resume order:

Place an in-progress order

1. User is at the orders screen
2. User employs pulldown to select select ‘open in-progress’ orders
3. List of open orders is displayed
4. User selects the appropriate order
5. User can then edit or place the order as described above.

$Orders->getOrderWithLineItems($orderId)

$Orders->placeOrder($orderId)

Search:

This describes a search for

Free Form Search:

A key-word search based on product descriptions with type filtering.

1. User clicks on search icon
2. Dialog appears asking for type of search
3. User selects key-word search
4. Dialog asking whether it is a product search or a shop search.
5. Text box appears into which the user types the search term
6. List of products or shops carrying products satisfying the search is displayed.

$Product->searchProductDescriptions($targetText, $filters)

Product/shop Category Search:

A step-wise exploration of category tree.

1. User clicks on search icon
2. Dialog appears asking for type of search
3. User selects category search
4. Dialog asking whether it is a product search or a shop search.
5. User clicks his/her way through the tree. Can click on ‘see products’ button at any level, and will display a list of products included in that sub-category
6. If the user reaches the leaf of the tree, a list of products falling uder it will be displayed

$NestedCategory->findChildNodes($parentNodeName)

Shop Location Search:

A search or shops based on their location determined by zipcode

1. User clicks on search icon
2. User selects shops
3. User selects zipcode location search
4. User types in zip code
5. User selects how close
6. List of shops is displayed
7. User selects shop
8. Catalog is displayed.

$Company->findCompaniesNearZip($zipcode)

Shop Marketplace Search:

A search based on self-assigned location codes ex: east coast, New York etc.

1. User clicks on search icon
2. User selects marketyplace search
3. List of marketplaces appears
4. User selects marketplace from the list
5. List of shops appears
6. User selects a shop
7. Shop’s catalog is displayed.

$Company->findCompanyByLocationCode($locationCode)

Messageing: (needs more discussion)

Initiate nessage with shop:

Send message to shop-admin

1. User is at Product screen or Shop screen
2. User clicks on ‘message the seller’ button
3. Message box appears and user enters the message
4. User clicks on send button. A new thread is created.

$Message->addNewMessage($msginfo)

Access message center:

See all your current messages.

1. User clicks on mail icon at the bottom
2. A scrollable list of all unread thread headers is displayed
3. User can click on an individual message to read it.

$Message->getMessagesToUser($user) – need to add filter

See previous message:

Page to previous message.

1. User is at an individual message screen
2. User clicks on up arrow icon
3. Previous thread is displayed

$Message->getPreviousMessage($messageId)

See next message:

Page to next message.

1. User is at an individual message screen
2. User clicks on down arrow icon
3. Next thread is displayed

$Message->getNextMessage($messageId)

Get All Message Headers in thread

See all message headers in current thread.

1. User is at the message screen
2. Thread’s initial message is displayed at top
3. Headers are below the message

$Message->getMessagesInThread($messageId)

Add message to thread:

Add a reply message to current thread.

1. User is at the message screen
2. User clicks on ‘plus’ icon
3. Message box appears
4. User enters text
5. User clicks on save button
6. Message is added to the thread.

$Message->addMessageToThread($messageId, $msginfo)

Add picture to message:

Upload or add camera picture to current message

1. User is at a message entry box
2. User clicks on upload icon – file browser appears, file uploaded and attached to the current message
3. Use clicks on the camera icon
4. Camera takes picture
5. Picture is uploaded and attached to the message.

$Message->uploadPicture()

$Message->addPicture($messageId)

Mark thread:

Mark a thread read or unread. Will revert to read if new reply added.

1. User is at the individual message screen
2. User clicks on ‘mark this thread as read
3. Thread will no longer appear in list of messages unless user clicks on see all messages button in the message center.

$Message->markThread(readUnreadStatus)

Delete Message:

Remove a message from the system entirely.

1. User is at individual message screen
2. User clicks on delete this thread button
3. Thread is removed from the system perminantly

$Message->removeMessage($messageId)

Specials:

Specials are handled in two ways by the system – products are that part of a special are contained in a collection that is typed as a specials collection. Specials prices, terms, and expiration dates are to be embedded in the ContainedAs record. Specials will be promoted as messages that are also appropriately typed.

View Current Specials:

View products that are contained within collections of type specials.

1. User clicks on the specials icon
2. Scrollable list of products contained in special collections is displayed
3. User can click on an individual product
4. User can click on ‘order this’ button – transition to order process.

$Product->getProductsInCollection($collectionId, $categoryFilter)

$product->getProductsInSpecialCollections($categoryFilter)

Get Participating Stores:

View a list of stores with products in specials collections.

1. User touches the find icon at the bottom
2. Prompt – Shop or Product
3. User selects Shop
4. User is prompted “Shop Name, Product Category, Location, Favorites, Specials or Marketplace
5. User selects Specials
6. Scrollable list of shops having products in special collections is displayed

$Collections->getCompaniesHavingCollection($collectionId)

Add Product to Special:

Add a product to a collection types as a specials collection

1. User is a shop-admin and is at a product screen
2. User clicks on ‘add to specials’
3. Dialog appears asking for discount and term information
4. User clicks on ‘add’ button
5. Product is added to special collections. ContainedAs record is updated.

$Collections->addProductToCollection($productId, $collectionId, $containedAsId)

Set end date for special:

Set an expire date for a specials collection;

(see above – this is set when the product is added)

$Collections->setEndDate($collectionId)

Remove product from special:

Remove a product from a specials collection.

1. User is store-admin
2. User goes to product that is in a special collection
3. User clicks on ‘remove from specials’ button

$Collections->removeProduct($productId, $collectionId)

Get sales figures on special:

Get sales data from products in a specials collection

1. User is store-admin
2. User goes to product in specials
3. User clicks on ‘see sales data’
4. Report is printed

(we probably should include a report that covers all the items a store has under specials)

Collections->getSalesFromCollection($collectionId, $filter)

Menus/Other:

View News and Updates:

View messages typed as news or promotional annopuncements

1. User is looking at messages
2. News, updates, and promotional announcements are displayed in a special color.

Messages->getFilteredMessagesToUser($userId, $filter)

User Settings Menu:

Set user-controllable settings (needs discussion)

User->editUserProfile($user, $editInfo)

Admin - System Settings (product categories):

Settings that can be made by admin – categories etc.

NestedCategory->addCategory($name, $description, $parentNodeName)

NesterCategory->deleteCategory($nodeName)

Super- Admin – Event Types:

Event types – only super-admin (developer) will be interested in this.

Events->addEventType($eventType)

Events->removeEventType($eventType)

Super-Admin – Permissions

Set permissions for various roles.

Roles->addRole($roleName, $permissions)

View Privacy Policy:

Display privacy policies (may vary by country)

1)User is on profile page

2) Use clicks on privacy policy link

3) policy is displayed

User->viewPrivacyPolicy($locationId)

Access Help:

Context-sensitive help.

1. User can be on any page
2. User clicks on Question-mark icon
3. Help for that page is displayed.

$User->help($pageLocation, $locationId)

Catalogs:

Store can have a number of catalogs. There is one catalog that contains every product a store has listed, but that catalog is not published. Store can have catalogs that are in the process of preparation but not published. Stores can have one or more than one catalog published.

Preview Catalog:

Show preview of catalog

1. User is store-admin
2. User is on products page
3. User clicks view catalogs link
4. List of catalogs for that shop is displayed
5. Use can click on an individual catalog to get a list of products contained

$Collections-> getCollectionsByCompanyId($companyId)

Create Catalog:

Set up a new (empty) catalog

1. User is store-admin
2. User is on products page
3. User clicks view catalogs link
4. From the catalogs page, user clicks on ‘add catalog’ link
5. Entry page for new catalog is shown
6. User clicks on ‘save’ button. Catalog will now appear in the list of catalogs

$Collections-> addCollection($collectionName, $collectionSlug,$collectionDescription,$collectionType,$collectionStatus, $collectionCompany)

Edit Catalog:

Edit existing catalog, showing, adding and deleting products

1. User is store-admin
2. User is on products page
3. User clicks view catalogs link
4. User selects a catalog
5. Catalog is displayed
6. User clicks on add product link – list of products from all his/her shop’s catalogs is displayed
7. User can select a product
8. Product is added
9. User can click on remove product from catalog link to delete product from a catalog.

$Collections-> editCollection($collectionId,$collectionName, $collectionSlug,$collectionDescription,$collectionType,$collectionStatus, $collectionCompanys)

$Collections->removeProductFromCollection(($productId, $collectionId)

$Collections-> addProductToCollection($productId, $collectionId, $containedAsId)

Publish Catalog:

Set status of catalog to make it visible to prospective buyers.

1. User is store-admin
2. User is on products page
3. User clicks view catalogs link
4. User selects a catalog
5. Catalog is displayed
6. User selects status from pulldown
7. User saves to update the catalog’s status

$Collections->setCollectionStatus($status)

Add Product to Catalog:

Add a product to an existing catalog

(see above)

$Collections-> addProductToCollection($productId, $collectionId, $containedAsId)

Remove Product From Catalog:

Remove the association of aproduct with an existing catalog.

(see above)

$Collections->removeProductFromCollection(($productId, $collectionId)

Products:

All of a store’s products are contained within an unpublished ‘store catalog’ and are added to it when the product is entered. Every product has a category, and this category determines a set of default options that the product can have, such as sizes, colors etc. When the product is added, selections from this list of defaults can be made to publish which options for a particular product are available for sale. Default terms are linked to the store itself. Again, thwn the product is added, selections can be made from the list of default terms to publish which terms are actually available for a particular product.

Add New Product:

Enter a new product using defaults (options, terms)associated with a product type.

1. User is shop-admin
2. User is at products page (this is a page listing everything in the store catalog)
3. User clicks on the ‘add product’ button
4. Entry page opens up allowing entry of product info, upload of picture etc.
5. User can select catalogs to publish the product into from a checkbox list
6. User can select options from a checkbox list
7. User can select terms from a checkbox list
8. User clicks on save and product is added

Product->addProductUsingDefaults($productInfo)

Options->linkDefaultOptionsToProduct($productType, $productId)

Terms-> addDefaultTermsToProduct($companyId, $productId)

Set Default Options for Product Type:

Establish the default set of options and terms for a product type.

Options->addDefaultOption($option, $productType)

1. User is admin
2. User goes down category tree and adds a new leaf
3. User fills in name of new category
4. User can add options to a list of default options for this product type.
5. User clicks on save, and new category along with type info is added

Associate Default Options with Product:

Associate a set of default options with a product (color, size etc).

(see above)

Options->linkDefaultOptionsToProduct($productType, $productId)

Remove Product:

Remove a product entirely

1. User is shop-admin
2. User is at product page
3. User selects a product
4. User clicks on remove product button
5. There is an ‘are you surte dialog’

Product-> removeProduct($productId)

View Favorites list:

Shows the the existing favories listrs. User can select one to look at and products on the list are displayed.

1. User is shop-admin
2. User is at product page
3. User clicks on favorites button
4. A list of the user’s favorites lists is displayed
5. User can select a particular favorites list
6. Products on the list are displayed.

Add new favorites list

User can add a new favorites list

1. User is shop-admin
2. User is at product page
3. User clicks on favorites button
4. User clicks on ‘add a list’ button
5. Info on list is entered
6. User click on save button.

Remove a favorites list

1. User is shop-admin
2. User is at product page
3. User clicks on favorites button
4. User clicks on ‘remove this list’ button
5. There is an ‘are you sure’ dialog.

Edit a favorites list

1. User is shop-admin
2. User is at product page
3. User clicks on favorites button
4. A list of the user’s favorites lists is displayed
5. User can select a particular favorites list
6. Products on the list are displayed.
7. User can click on add a product
8. User finds the product from the store catalog
9. User clicks on add this product button
10. User can also click on remove product button