Account Related:

Forgot username/password:

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: (

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:

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:

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:

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.

$User->getUserProfile($userName)

$User->addUser($info)

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

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

Preview Store Profile:

Shows all the info the system has on that store

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

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.

$Company->getCompanyByName($companyName)

$Collections->getCollectionsByCompanyId($companyId)

Add/Post Shop to favorites:

Shop is added to user’s favorites list.

$User->addShopToFavorites($companyId)

Share post/shop (need discussion):

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

Message Shop:

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

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

$Message->addNewMessage($msginfo)

Take Action:

Shop WholesaleCatalog:

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

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

Request Sample:

Message is sent to shop-admin requesting sample.

$Message->addNewMessage($msginfo)

Buy One Sample/Buy Product(s):

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

$Orders->startNewOrder($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.

$Product->getAllSellableProductsByCategory($category)

$Product->getCompanyProducts($companyId)

$Product->getSimilarProducts($productId)

Adjust color/type:

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

$Options->getDefaultOptionsForProducttype($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.

$Orders->placeOrder($orderId)

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

$Orders->selectShippingOption($orderId)

Add note to seller:

Seller is sent a message referencing the order.

$Message->addNewMessage($msginfo)

Save Order In Progress:

Order status set to in-progress.

$Orders->startNewOrder($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

$Orders->setStatusCanceled($orderId)

Return To Saved Order:

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

$Orders->getLineItems($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)

Find/Launch Seller Catalogs:

Get catalogs that are open for sale of a specific company

$Collections->getCollectionsByCompanyId($companyId)

Place digital order at show:

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

$Company->findCompanyFromCameraBarcode()

$Product->findProductFromCameraBarcode()

$Orders->addProductFromCameraBarcode()

$Orders->placeOrder($orderId)

Save shop to favorites:

Add shop to favorites list.

$Company->saveCompanyToFavorites($companyId)

$Orders->duplicateOrder($orderId)

Manage Orders:

View existing orders with specified status.

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.

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

Order->addToFavorite(productId, $quantity)

Get value of possible purchase list

Total up value of possible purchases.

Order->getValueOfFavorites($company)

Open Orders – message seller:

Send message to seller of open order.

$Message->addNewMessage($msginfo)

Open Order – cancel an open order:

Cancel an open order and message the 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.

$Orders->editOrder($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)

Confirmed Orders – re-order:

Shipped Orders – re-order:

Duplicate order offering opportunity to edit quantity, options etc.

$Orders->getOrderWithLineItems($orderId)

$Orders->duplicateOrder($orderId)

$Orders->placeOrder($orderId)

Declined Orders – edit order:

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

$Orders->editOrder($orderId)

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

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

Declined Orders – clear order:

Remove a declined order from system entirely.

$Orders->removeOrder($orderId)

Saved Orders – cancel order:

Set status of in-progress order to canceled.

$orders->setStatusCanceled($orderId)

$Orders->removeStartingOrder($orderId)

Saved Orders – resume order:

Place an in-progress order

$Orders->getOrderWithLineItems($orderId)

$Orders->placeOrder($orderId)

Search:

Free Form Search:

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

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

Product Category Search:

A step-wise exploration of category tree.

$NestedCategory->findChildNodes($parentNodeName)

Shop Location Search:

A search or shops based on their location determined by zipcode

$Company->findCompaniesNearZip($zipcode)

Shop LocationGroup Search:

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

$Company->findCompanyByLocationCode($locationCode)

Messageing: (needs more discussion)

Initiate nessage with shop:

Send message to shop-admin

$Message->addNewMessage($msginfo)

Access message center:

See all your current messages.

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

See previous message:

Page to previous message.

$Message->getPreviousMessage($messageId)

See next message:

Page to next message.

$Message->getNextMessage($messageId)

Get All Message Headers in thread

See all message headers in current thread.

$Message->getMessagesInThread($messageId)

Add message to thread:

Add a reply message to current thread.

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

Add picture to message:

Upload or add camera picture to current message

$Message->uploadPicture()

$Message->addPicture($messageId)

Mark thread:

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

$Message->markThread(readUnreadStatus)

Delete Message:

Remove a message from the system entirely.

$Message->removeMessage($messageId)

Specials:

View Current Specials:

View products that are contained within collections of type specials.

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

$product->getProductsInSpecialCollections($categoryFilter)

Get Participating Stores:

View a list of stores with products in specials collections.

$Collections->getCompaniesHavingCollection($collectionId)

Add Product to Special:

Add a product to a collection types as a specials collection

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

Set end date for special:

Set an expire date for a specials collection;

$Collections->setEndDate($collectionId)

Remove product from special:

Remove a product from a specials collection.

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

Get sales figures on special:

Get sales data from products in a specials collection

Collections->getSalesFromCollection($collectionId, $filter)

Menus/Other:

View News and Updates:

View messages typed as news or promotional annopuncements

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)

User->viewPrivacyPolicy($locationId)

Access Help:

Context-sensitive help.

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

Catalogs:

Preview Catalog:

Show preview of catalog

$Collections-> getCollectionsByCompanyId($companyId)

Create Catalog:

Set up a new (empty) catalog

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

Edit Catalog:

Edit existing catalog, showing, adding and deleting products

$Collections-> editCollection($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.

$Collections->setCollectionStatus($status)

Add Product to Catalog:

Add a product to an existing catalog

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

Remove Product From Catalog:

Remove the association of aproduct with an existing catalog.

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

Products:

Add New Product:

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

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)

Associate Default Options with Product:

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

Options->linkDefaultOptionsToProduct($productType, $productId)

Remove Product:

Remove a product entirely

Product-> removeProduct($productId)