#For readers:

Originally done as a school project, this is the documentation for my Ecommerce project which had missed some of the scope of this documentation. Happy viewing!

Problem:

Sales are quick, if you take too much time people will often turn elsewhere. So you need software that is just as quick. Focusing more on smaller businesses, if you lack a platform, other tech giants have you covered, namely Facebook, eBay, Amazon, and some of the newer names like Mercari. But, humans can only manage so many listings at a time, and surely if you're listing variations of the same item, it's only logical there could be a way to ease the burden and remove 'busy work'. Ensuring your items can be listed, responses kept track of, and listings updated, there are many elements to using these platforms that can be monitored in one simple place. Relisting items can also be a hassle, so having an integrated system that stores one copy but can duplicate it to multiple platforms may be a blessing, creating so much more traffic with half the cost of work.
Offering a place where all your listings are viewable as condensed site versions, ordered by either listing date or interaction date, you can easily see the state of your listings. Integrating both a 'quick-view' and already adapted APIs of sites such as eBay, our platform will offer methods of requesting and sending information regarding your listing to the desired location.

Requirements

Functional

1. Seller must be able to edit listings freely among various platforms
2. Seller may need to be informed of shipping information and inquiries
3. The seller must confirm listings to prevent ‘accidental listings’
4. Seller must be able to have a quick view of all items in an order of their choosing
5. The seller can write notes about customers to view later
6. Expected profits can be easily viewed
7. Merchants can delete listings as needed
8. Customer Accounts can be created to have a collection of shipping information
9. Shipping information is accessible on the platform (IF possible)

Non-Functional

1. The platform must not exceed a loading time of 30 seconds
2. Notifications notify on a 24-hour cycle or important updates
3. The listing creation screen must be one click away from the main page
4. Confirmation of listings will have 3 confirmations
5. Listings must be saved frequently to prevent data loss on crash
6. Must be able to access recently updated items on the main page
7. Page deletion may be done automatically if specified by the seller, otherwise must be within two clicks
8. The performance will not be largely impacted at 100+ listings
9. Expected profits are very visible on the home screen

Constraints

1. Will make use of the Java language
2. Will use the Salesforce Apex
3. Incorporates various APIs from e-commerce platforms (eBay API)
4. Listings will be a custom object

Competitors:

Crosslist, FBMFox, Zdrop, Inkfrog, Acclerlist, Helium 10, A2X, Webgility

Scenarios:

1) John has just acquired many goods that must be listed by the same day.
   John writes down his listing information on his phone.
   The software quickly adds new information and distributes it to its various platforms.

2) Antonio wants to inquire about a specific item.
   Anotonio lives in a very different time zone from Sally, the seller.

   Sally gets a notification on her device saying there is an action on her listing and can inform Antonio of the situation.

3) The item Antonio purchased will be ready to ship the next day.
   Sanjay has used eBay for years, listing every item he has sold manually.
   He installs the software and extracts all of the listings he has created to monitor and update them in bulk.

4) Carl has worked a long day and is very tired but knows he needs to list some items so they're on the market and garnering interactions.
   Carl, unfortunately, mixed his items up and needs to quickly revert his listings and or update the current listings to accurately reflect the item.

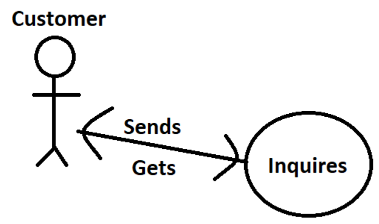
5) Hugh has started his small sales business, listing a handful of items on his first day.
   Hugh has unbeknownst to him listed items with PID, personal Identifying data.
   The app prevented him from typing in certain keywords and gave confirmation steps to ensure all the information was secure.

Use Cases:

A. A seller has a large inventory that must be listed in a timely fashion across their listing platforms
B. Customer wants to receive quick responses and information regarding their shipments and orders
C. A seller is notified when a customer makes inquiries across any of his used platforms
D. The seller tries to create a listing for the incorrect item / puts incorrect information (including pictures) and will be required to check again to confirm his listing

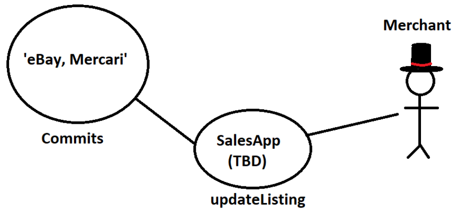
E. The seller attempts to relist an already sold and possibly shipped object, will be informed of such a case and prevent the listing

Case B:

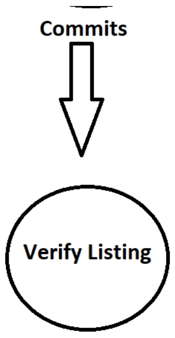


See next page

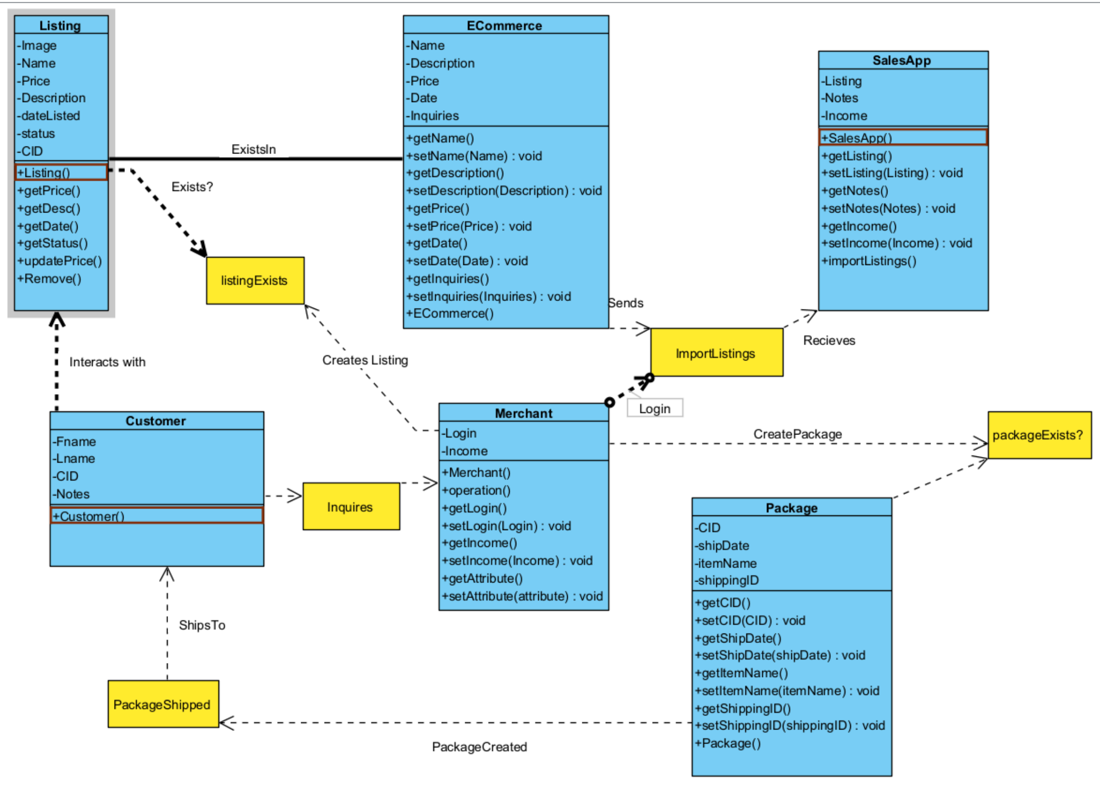
Case A:



Case D:



UML:



OCL Constraints:

Listing Exists:

```
    if(!ListingExists)
    { createListing()}
```

Inquiries:

```
    if(filterMessage(CustMesg))
    { NotifyMerch();}
    //Default response or Merchant Response
    RespondTo(CustMesg);
```

PackageShipped:

```
    if(isNull(ShippingID))
    {newShippingID(CID, PackageNO)}
    NotifyCust();

//Ensure the merchant knows
    if(shippingID == null || shippingDate < currentDate - DaySpec)
    {
        NotifyMerchant();
        CreateShippingID();
    }
    /*DaySpec is the merchants chosen date for when they need to ship a package*/
```

Package Exists:

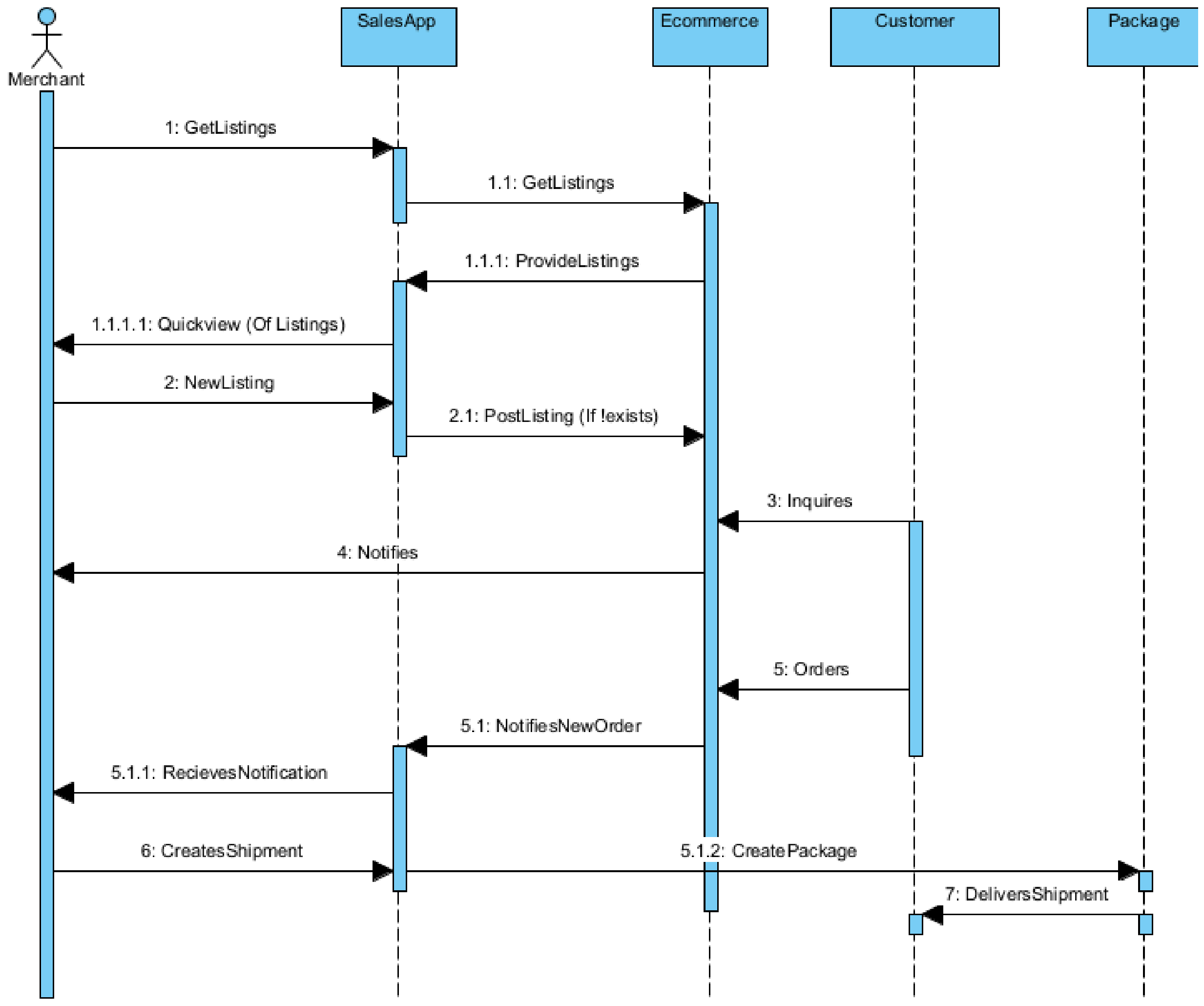
```
if(isNull(PackageNO))
{CreatePackage()}
```

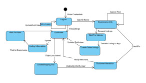
ImportListings:

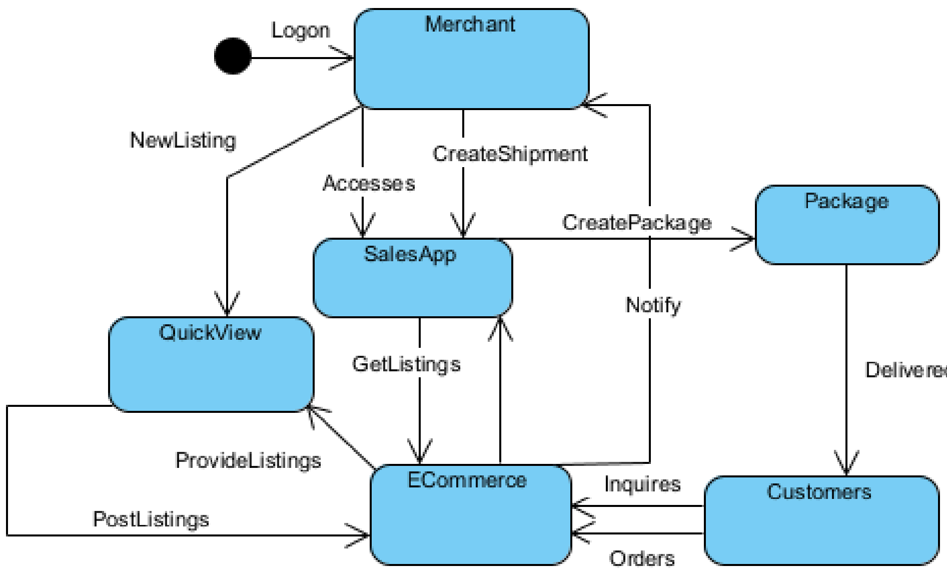
```
while(!listingExists)
{ addListing() } //To home page
```

<u>Sequence Diagram:</u>

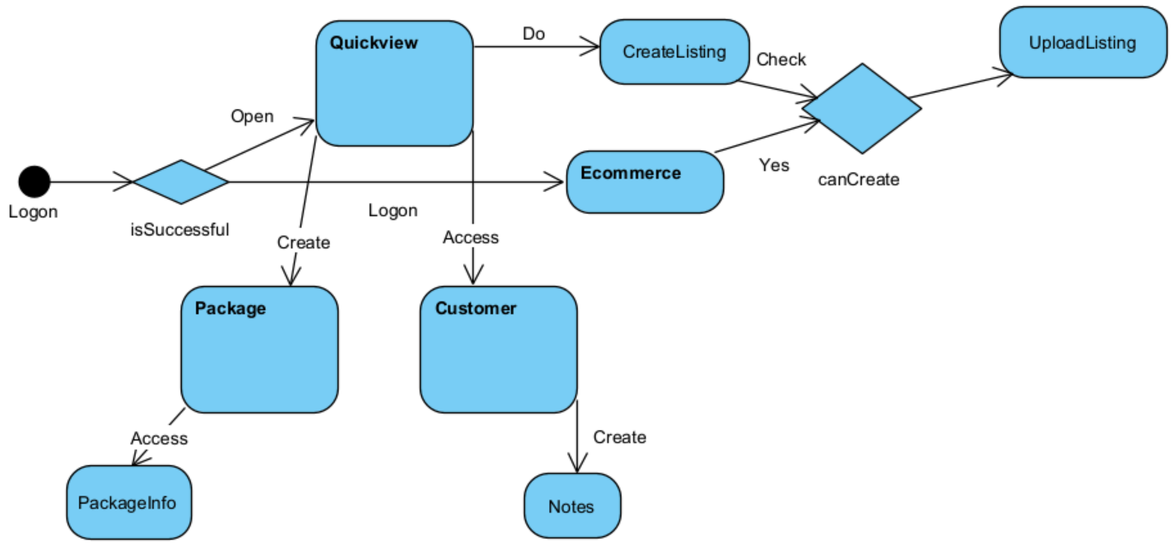
//Old


Merchant
SalesApp
Ecommerce
Customer
Package
1: GetListings
1.1: GetListings
1.1.1: ProvideListings
1.1.1.1: Quickview (Of Listings)
2: NewListing
2.1: PostListing (If !exists)
3: Inquires
4: Notifies
5: Orders
5.1: NotifiesNewOrder
5.1.1: RecievesNotification
6: CreatesShipment
5.1.2: CreatePackage
7: DeliversShipment

State Diagram:  //OLD



Activity Diagram:



Implementation Specs:

-Will make use of Various Ecommerce APIs
-Class Based infrastructure
-Secure 2-Step Login

-Verification of steps
-Easily Accessible elements