/// scratch paper – 5/22/17

/// mini conversions – sub project of the R3 of sub inventory

/// we have a number of sources that have some info. They are:

* Code and database stuff from Calvin Chipman
  + We need a high level template (cookie cutter level)
  + They build as many templates as needed, they can just use them as needed – the secret is what is the parent unit of measure and what is the base unit of measure on the template.
  + On the conversion details table and the conversion activity tables. They will need to be corp specific.
  + We need to show a message back to the user based on the usage and conversion activity.
  + When removing from the cart, set the quantity to 0 and set the status to 0 on the conversion activity table.
  + When clearing the cart, set all quantities to 0 and set the status to 0 on the conversion activity table.
  + We may want to generate a cart number based on the payee and some kind of numeric value.
  + What about an auto clean-up rule for purging items on hold (claimed). Maybe even have run auto clean-up or manual clean-up processes based off of the numeric date/time stamp.
  + If we are doing restrictions… what about items that don’t go as deep as the mini conversions? What about sub inventory? What about parent inventory?
  + The pricing engine will need to use some of the mini conversion pieces. We don’t know where that is going to go. We may have to figure that out later.
    - Steve would like to use some of the smart group type logic… basically rules and assignments. We also need some tie-ins to dates, times, promotions, etc.
  + On the conversion action table – we need to cover: make it, add to cart, update cart quantity, remove from cart, clear cart, save as quote, restore to cart, save as other quote, checkout, after the fact adjustments up, after the fact adjustments down.
    - Make it is pretty simple – set the process type to 1, set the qty, set the mode to create, set the app type to 1, the main id to 0, the line id to 0.
    - Add to the cart (scan or some other method) – set the process type to 2, the set the qty to a negative, set the mode to cart, set the app type to 1, set the main id to 0, set the line id to the cart line number (from the array).
    - Update the cart quantity – just change the quantity. Remember, most times it will be negative.
    - When removing a line item – set the quantity to 0 and set the status to 0.
    - When clearing a cart – set all quantities to 0 and set all statuses to 0.
    - When saving as a quote – leave the process type at 2 (claimed), change the mode from cart to quote, change the app type to 13, set the main id to the main quote number, hold the line item id’s.
    - When restoring from quote to cart – just change the mode from quote to restored. Leave everything else alone until they do another action.
    - If saved back as same quote, just update mode and other details.
    - If it is saved as another quote, make a whole new set of entries. This does double up the claimed inventory.
    - If full checkout, change the process type id to a 3 (full checkout or sold or gone), cange the mode to invoice, change the app type id to 3, change the main id to real invoice number, change the line id to the real po/invoice line id.
      * Also as part of the checkout, we also want to record the sub inventory id number (~ num), the conversion id (` num), the show conversion u of m, the show quantity, and the show pkg price. These last 3 fields are to help with the smoke and mirrors stuff.
      * If a normal sale or normal checkout where the item is not tied to sub inventory, what then? We need to change the show conversion u of m to the normal u of m, the show qty to the normal qty (but positive), and the show pkg price to the normal price per.
      * Also behind the scense, we need to split up the costs, the item prices, and set the normal quantity and normal unit of measure to what needs to be recorded behind the scenes.
      * We also need to make sure that the taxes are figured out correctly.
      * As a side question… On PO’s, we use a field called the rounding error. Is there ever a time on an invoice that we might need that field. If yes, what is it and how would it flow through?
    - The adjustments up and down are going to be manual entries. They would hit the conversion action table directly.
    - Eric had an ideas about a quote only… some how be able to make the quote not claim any of the inventory.
    - What about adjusting things and rolling things backwards into the base unit of measure. For example: We created 75 of something to get us through a weekend sale… The weekend is over and we only used 60 of them. We want to take 15 of them and roll them back into the base unit of measure.
  + If we add a number of new fields to the po/invoice line items table, we need a what to go backwards and fill in whatever data we can. This would be a lot of clicking but would bring the data up and into compliance. This will be part of the database update logic.
* Exploratory stuff and ideas from Steve and Danny
  + At first, the multipliers were locked down. That didn’t work very long. We opened that up. Each company may have different multipliers. We are also seeing that even though the conversion rules are set to something, we need to let that piece flex when they do the actual conversion process.
  + Steve’s stuff started with a perfect parent/child relationship
  + We already have the parent item, the vendor, and the sub category
  + He pulled in the package weight or quantity
  + We already know the RFID tag number
  + On the PO line items, show the parent, show the subs, and show an indicator if there are conversions. This indicator could also allow a sub to be sub divided (going to the template or build page). It could be an open circle (no mini conversions) or a filled circle (it has some conversions that have been built out).
  + We may need to use more and more icons to help them get where they need to go. Think icons or something to keep it small. Basically, we need a way for them to move around and see what is going on without taking up tons of space.
  + We have some great ideas on how to shorten up the label functionality. We talked a bunch about ~ and ` symbols. Basically, we are doing some mapping.
  + The old way (when Steve and Danny were working on things) had print buttons. This was the commit or action piece. We need to allow them to make things as needed and then print to their hearts desire. Basically, split the print vs. the action. We need to build things and then allow printing as needed. The building is the action. The printing is just extra.
  + The biggest limitation is pushing up to the limit of the barcode. Technically we can barcode anything… the problem comes back to our internal barcode generator only allows up to 15 characters. We maxed that out pretty quickly. The new `5001 (back tick) will really help that out. It will basically be able to go back and look up the price, the sub inventory id, and the unit of measure all from one id.
  + We do have Chris Dunsey looking into a dynamic barcode. Also we do have an app that helps to do that (BBQ stuff – a Java app). See the custom/trinity fasteners pages.
* Tons of ideas and notes on the adilas developer notebook and adilas shop (elements of time – research and such)
* Ecommerce stuff from Russell and some alternate angles
  + Mostly logic and visual display
  + We may need to build out the internal pieces first and then see what is needed.
  + What about the API socket level? That will be needed at some point and time.
* Eric needs this kind of stuff to help with the pricing engine and even clear down to discount rules and loyalty point stuff
  + This may not be fully ready yet… but we wanted to dream a bit.
  + We know that he base unit of measure and the conversion unit of measure will have an effect on the loyalty points. Certain things will be worth more.
  + Steve keeps saying things about what pricing tier or pricing category does this mini conversion belong to? What about quantity discounts based on some of the conversion unit of measure vs. quantity discounts based on the base unit of measure.
  + It might be nice if we could do things like we have with smart group buttons without being tied into the buttons. Basically, all of the logic, rules, and options, without the actual button being the owner or holder of the connections. As a side note, we have things called smart group rules and smart group assignments. Both of those are huge pieces.
  + What comes first, the conversion or the price?
    - They may have to build the rules and then select what plays into the rules. You may need both, conversion options and the pricing engine pieces to make the full thing run together.
    - As a side note, in Blum (one of our clients) they had to setup the pricing rules first. Then they added those pricing rules to the individual items.
    - As another side note, if they changed the pricing list, they had to click a button to cascade it. The old way was very manual and reliant on the pricing lists and the positioning.
  + As a fun note… We do have some clients that use a form of flex grid to help tie things together as far as pricing models. They are using the flex grid which holds a bunch of numbers. Here is a sample: 210,110,60,30. What this means is $210 for 1 oz, $110 for ½ oz, $60 for ¼ oz, and $30 for 1/8 oz. Basically, they setup a 4 length list which knows that position 1 is for full ounces, position 2 for ½ ounces, position 3 for ¼ ounces, and position 4 for 1/8 ounces.
  + They would then assign the pricing lists to an item. For example: Item X has a pricing tier list such as the 210,110,60,30 list. We then watch to see how much (weight and quantity) and then see what level it plays on. If the total reaches the ½ ounce level, we would then find the position 2 price and play from there. This was a fully built out work around. I think we can gain some good info from this model.
  + We are seeing tons of ways of grouping items and prices… it is based on category, sub categories, sub inventory, mini conversions, parent items, dates, times, parent attributes, sub attributes, per vendor, per location, per event, etc. Lots of ways to group things. We also need to set start and end periods.
* Other
  + We need to add in a number of new standard units of measure. Things like 1/8, ¼, ½, ¾, cones, shake. Keep things generic.
  + What about tracking the history? Is it on the template? The conversion itself? The process?
  + We may want to add user-maintained barcodes for the subs (on the time\_sub\_inventory\_[53] table. We may also want to add a barcode on the conversion\_details table.
  + What about showing discounts?
  + Cascading the smoke and mirrors will be days worth of work… we have carts, invoices, mini invoices, quote, pdf’s, ecommerce stuff, email stuff, and tons of black boxes… This is a huge part of the puzzle.
  + We may want to show some light conversion info back to the person – in the cart mode. Maybe a plus icon that shows underlying base quantity and base unit of measure.
  + What about rounding errors? We are splitting hairs… we have to either force the rounding error, or we just record it and virtually eat it as cost of goods sold.
  + There is a number of new tables that need to be added in when making and creating new corporations. Make sure and add those corp-specific tables to the cfc/maintenance.cfc file.
  + When we over make something and want to destroy it or roll it back, we need a way to push it back to the original child piece. If there is some left over, it will say at the base unit of measure. We may also want to allow the company to do an entry on the update PO to alter the original amounts or left over amounts.
  + We know that people will want to transfer both subs and mini conversion pieces… we need to allow that. That is super important.
  + This is a question on sub attributes and part categories… If something is setup and then changed… what happens to the sub attributes? We could lose some data or orphan some of the pieces.
  + We may need to record when a part number or item is moved between categories – think of a history.
  + What if we made the attributes follow the parent item – currently they follow the part category… what if we made it so that they follow the item instead? Just an idea.
  + We have some other clients that may need to know about the new features and pages… Once the new stuff is launched, we may need to help point the users in the correct direction.
  + We are seeing a number of similarities between build and sell recipes, build and hold recipes, smart group buttons, and these new mini conversions. All of it deals with breaking things into smaller and smaller pieces.
  + We can see that these people will end up making up their own units of measure and then being able to sell and market that.