**Tru-Medical: design alternative #2 / Working Prototype**



**1. footer on all pages** ( in application.html.erb )

**2. catelog request:**

1. instead of having category request create a new user, have it create a new catelog\_request record. It should also check to see if this is a user we already know about, and if so, link the catelog request record to that user, assocation called 'catelog\_requests'?

2. what happens after creating a new catelog\_request? right now it is trying to render views/users/show.html. Want it to give the user a quick confirmation that his request has been received, and will be mailed shortly. implies sending the form in using ajax; putting confirmation msg in the popup, and then having the popup close, or saying 'To close and return to ... click x" . See code created for AJAX-based variant of 'Sign Up' form.

3. expand the form to include a choice: does he want printed copy/copies of the catelog mailed to him, or would he like a pdf copy emailed to him? Maybe a 'click here to download a pdf copy' / fill in the form below if you would like us to mail you a physical copy: in which case we'll need his physical mailing address — something we are not capturing now.

4. try other form builders: formtastic, its newer cousin: simple\_form

5. check to see if anyone else thinks its odd to have to create a dummy @user in order to pass it to the form\_for when the form is going to be used to create an instance of the type. Trick is to create those users in memory, but not save them to the database unless the user actually fills out and submits the form.

**3. search for Sammons part number equivalent**

1. How going to do this?

2. Once you find it what? : go to page for the equivalent part in TruMedical catelog? What if the match is not exact?

**4. header on all pages (actually fields on the upper right )**

1. register/login: build own using Authlogic Rails plugin (or whatever used on aaox); want stuff to come up in popups above the home page, not go off to separate pages — and especially not the slow netsuite-generated ones.

**5. 2nd, 3rd level category navigation**

1. update the category\_tree\_v3.xls file to add (i) the new images, (ii) the 3rd level categories. [DONE] 12/10

NOTE: Although this is a tree with a maximum depth of three, it does not go uniformly down to level 3 in all branches. Many go only to level-2. Force it to 3 by adding a dummy 3rd level 'All' category ito each 2nd level node that does not have 3rd level chldren as did in VMS? No, in that case, it was only a very small % of the 2nd level categories that required a dummy third level category 'All'. Here, well over half of the categories bottom out at level 2. Time for n-level category tree software. Will need both a nav-HTML builder and menu ( and/or category landing pages ). In this case I've hand-built the left-nav so the .rb script that builds the HTML for the left-name won't be necessary.

2. Make sure that we have all of the 'cateogry images' referenced in the excel file. Make sure they have been resized (to 120x80 jpgs), and are in the correct directory, eg.., /app/images/categories [DONE] 12/10

3. Reload the categories table in the database so that the categories pages will pull in these images. [DONE] 12/10

4. Restart the application and dbms servers, bring up a browser against the local version of the application, see how the category landing pages look. [DONE]

5. Fix anything that is breaking —

— Modalities

references to:

710025-000 (Accessories ) [ image file missing

710016-000 (Combination) [ image file missing ]

710054-00 (Electrodes) [ short a digit in the filename ]

710065-000 (Lotions and Gels) [ image file missing ]

problem: /public/images/categories contains only the thumbnails made from the -XXX files Bill sent; it does not contain the 710016-000 files.

fix: Find them in /public/images/[products, and make 120x80 thumbnails from them. Place the thumbnails in /public/images/categories. Do this for /images/products/modalities for now. Batch resize all of the files in that directory, copying them into /public/images/categories. Then go through /images/categories and delete all but the -XXX and -000 files.

problem: leaf-level category (modalities/clinical electrotherapy/ultrasound) is showing several images of e.g., Dynatron 125. Should it be? Yes: these are all of the individual products in this leaf-level category. Image of Dynatron Intellect Transport 710032-000 is missing. Why? Have 710031-00, 710033-000 and 710034-000 but not 710032-000. Why not?

6. insert the images for all of the top level product categories so that if Bill shows the Working Prototype to his partners, the left nav will work across the full product tree. This will also let us surface any 'missing-image' issues now, rather than later.

Within public/images/categories create a directory for each of the top level product categories. Put inside this directory a 120x80.jpg image of all products in that category, and of all of the composite images being used as sub-category images within this category. [ Where are the xxx's now? all in Modalities, or does Modalities contain only the ones that are really for subcategories of Modality. If the later, then batch convert the rest from the composite\_png files that Bill had Mandy send. Eventually go through them and pick out only the ones that are being used as category names, put those in /public/categories/all and adjust the program (views/categories/show.html.erb) to find them there. For now, add the subdirectory name into the pathname being used to find the category images. WARNING: this may not be possible with a category tree in which either the 2nd or 3rd level nodes can be leaf-level. If that proves to be a problem, copy them all in the /public/images/products/all. That was the case, since the category 'names' in the spreadsheet ( and therefore the database ) did not have underscrores, and the filenames did. ( And I didn't want to insert an obscure a regular expression to replace the spaces with underscores. ) However, images/categories/all only has the XXX images for modalities, not for the other top level categories. Where are they? Got them; have inserted the thumbnails that are used in the category landing pages in in /views/categories/alls. [DONE] 12/9

Go back and recreate all of the 120x80 thumbnails, shrinking the 169x179 jpgs (or the original pngs to 80 vertical, and then growing the canvas to 120x80, and adding a white background. Then sve the file to /images/cartegories/<correct category> or just /images/categories/all if don't want to keep a copy in the directories also.

— modalities [DONE] 12/10

— clinical electrotheraphy (6 images: 7100xx-000 where xx=16/25/31/33/54/65) [DONE] 12/10

— [ etc ]

— clinical supplies

— tables and traction

— fitness productds

— orthopedic products

— AT/Taping

— Evaluation

— Dailing LIving Aids

— Lymphedema

— Wound Care

7. insert images for all of the products

within leaf-level category pages

Shrink the roughly rectangular PNGs to 80 high wide and whatever width setting the 'constrain to same height/width ratio' yields. Then place that in the center of a canvas 120x80px in size, and set the canvas background color to white. Export as a jpg image, storing the result in a new directory: /public/images/products/thumbnails.

modalities

clinical electrotherapy

ultrasound [DONE] 10/12

stim

combination

for product\_types and products

Right now they are in /images/products/<top\_category\_name>. They are all 169x179 jpgs converted from the larger pns is /TruMedical/Content/.PNG Images all categories from Mandy Aug4 Sept27. ~706 of them in total across all categories. Normalize to rectangle in golden-mean ratio — or a rounded-to-the closest 10 pixel equivalent. Check GettyImages standard ratios.

Have leaf-level category pages find them within their subdirectories by adding the subcategory name to the image\_filename? means first substituting underscores for spaces in [sbu] category name.

7. Within the partials:

views/home/\_subcategories.html.erb

views/categories/\_leafCategories.html.erb, and

views/leaf\_categories/\_products.html.erb

— allow user to click anywhere on the subcategory block (image OR text) to get to the next level of subcategory or product.

— same for links to product pages from leaf-level category pages: clicking anywhere on the block should take you to the product page, not just on the live text link.

— highlight the text link on hover of the entire block (not just the text link); use the site orange rather than red that is currently in there; No: replace it with turning on the orange background for the entire blocks as doing on level-2 category pages.

8. the partial \_products.html.erb currently assumes that it will find the product images in images/products/modalities: generalize this by point it at images/products/all\_small and moving images on 120x80 (120wide,80 high) white pads into /images/products/thumbnails.

9. The show action in the LeafCategoriesController makes a similar test assumption: it always returns in @products the set of products from category 9. It needs to accept the category number from an argument on the call. Thta argument needs to be set up in the link\_to macros in views/categories/show.html.erb.

10. sort [sub]categories within category so that things are shown on the page in the order specified in the category-tree entries.

11. get left-nav partial to work on category pages as well as home page.

12. remove the sibling nav: just confusing; Instead make the entries in the bread crumb trail live so that a user who wants to see other altneratives can go back up the tree to the [sub]category landing page that got him to where he is now.

13. left nav currently closes the fly-out panel for one menu item when you hover over another. That doens't work for the last one, if you, e.g., move your mouse from payor to either of the blocks in the rectanagle with Tru-Value products; in that case, since you haven't moused over another menu option, the 'payor' overlay remains up. Could we add a special-case mouse leave event to just the last item in the menu list without screwing up the browser's event stack state? If not, how about explicitly flushing the browser event queue? The API is probably unique to each different browser. Alternatively, put the code to close existing menu overlays into the event handlers for mousing over Promotions or Tru-Value Products. Clearer — and need to define what Bill wants to happen when the user does mouse over these elements anyway. One potential gottcha on this tack: Promotions and True-Value Products are bulit in on partial, the menu in another. Would need browser state to be visible across the two. Should be Ok.

**6. product pages**

1. insert a simple product page, so that Bill can walk down the tree and get to the bottom. Make sure we can handle branches of the product tree that are only 2 levels deep as well as those that go down the full 3 levels; this is a navigation issue, not something for the product page.

2. Note: the product images in images/products/all\_small and all\_medium DO NOT include all fo the products. I could not even find the ten images for modalities/electrotherapy/ultasound — unless for the product 7102490-002 and 710249-005 and 710249-010 we always use the image with name 7100249-000.jpg ( which is what the \_product partial does not ). WE ARE MISSING one image for our test subtree: 710032-000.png

3. insert working buttons to add-to-cart and checkout so that we can test the full path to ordering a product or set of products ( before we necessarily have the product group pages working as in the catelog. If we cleaned up the scrollers on the home page, in inserted the search, the sammon-part-number search, and made the catelog request form create [user]<-->>Catelog\_Request records so it would work with existing users rather than always assuming that they are new, we would have a site that Bill could put up, deciding the add the product group pages in a subsequent point release. [ Footnote: we would have to get the matrix item support working so we could handle colors and sizes in any such release 1.0. ]

**7. product set pages ( aka 'matrix' products pages; currently, in the code base, 'base\_products'**

1. reread my notes

2. review the design of the database classes/tables: reinsert the 4th table: line, type, model, configured\_product(\*), rather than trying to collapse it to 2? If insert Rails-side Admin subsystem then modeling the product class hierarchy completely is not much harder to administer than trying to keep it brief enough that we can push it into the Netsuite 'Item' table in an 'encoded' form. The downside of the encoded-in-the-Item table approach is that the site administrator would always have to remember why it was there, and how to work with it. Would also add all four the record types to Netsuite, but if we put the Admin subsystem on the Rails side, then we really wouldn't need anything on the Netsuite side other than the single predefined 'Item' record type with the built-in 'matrix' option enabled for the products that come in color/size combinations — essentially the 'configured products' that are actually being ordered, shipped, and tracked in the Netsuite financials.

3. create them with migrations

4. load them with data

5. write the HTML/CSS/Ruby for the pages/overlays involved.

(\*) How about Line, Product, Model where:

Manufacturer <->>ProductLine

ProductLine<->>Product

Product<->>Model

If a product has models, then it is Models that correspond to Netsuite Items

If it does not, then it is Products that correspond to netsuite Items.

Note that some products may come in sizes, and colors. This is handled by defining 'matrix items' on the Item in netsuite, one for size, one for color, and passing in orders for a particular Product (=Netsuite Item) that comes in a particular size and/or color.

Accessories are defined on Products ( not Models ) — in the case of the Intelec Transport Ultrasound unit, the unit is the Product; it comes in 4 models depending on the size of the applicator, and it has as accessories: cart, battery pack, carry bag, and any of the independently orderable applicators.

The only odd bit about this, is that the extract of items from Netsuite would be loaded into what? the Model table? And lots of Products would have a single model? The other alternatives is to use the word Product for what is here Model, and then introduce Product\_Type for what was here Product. Then netsuite items ( or at least that those don't have matrix parameters) map to Products. And we have to invent a term for the combination of an item in a given size and color. I suppose we could refer to that as a Product in a given size and color, or a Product Configuration. The problem is that English tends to use the simple term ( in this case 'Product') for the thing you think of most immediately, and qualifiers for less common things. And in this product set, when you are thinking about the simpler ones, model#2 fits this bill better. When you are thinking about complex ones, model#1 fits this bill better. I'm going to go with model #2 because it fits 80% of Tru-Medical's product line, and accept the oddness of having Product\_Types in the complex case.

So for the complicated case we have:

Manufacturer <->> ProductLine

ProductLIne <— contains —>>ProductType

ProductType <— has—>> Product

Product <— comes in —>> [ Sizes, Colors ]

and

Category <—>> Category, and we distinguish two subtypes: top-level and leaf-level categories

leaf-level Category <—>> Product\_Type

or?

leaf-level Category <—>> Product

We could hand this by introducing product types even for the products that don't have them ( that is, in the language of model#2, products that don't have models )

Or (option B) we could do it by defining on the Category record, both the associations product\_types and products. Then in the leaf-level category show.html.erb page, if the product\_types association is empty for the category, we list all of the products. Otherwise (if the product\_types assocation IS populated) we list the product\_types. And the product\_type show.html.erb page will have on it all of the products and accessories that go with this product\_type. Or, to try to bridge the two sets of terminology, we might say that a product\_type has ( or 'comes in' ) one or more models, each of which is referred to by the inventory and financial systems as a separate 'product'.

**Ok, so we're going to go with model #1 (ProductType <—>> Product ) and Option B for mapping leaf level categories to ProductTypes and Products.**

Mechanics of implementing it, from where the software stands now:

1. We already have the models: Category, ProductType and Product.

2. Category already has a has\_many :products relationship defined on it; add a has\_many: product\_types relationship. [DONE] 12/10

3. ProductType already has a has\_many :products relationship defined on it. Add a belongs\_to: category relationship. [DONE] 12/10

4. Product has a belongs\_to: product\_type. Add a belongs\_to: category. [DONE] 12/10

5. For the complex case, add classes Manufacturer and ProductLine. Each has a name, and a logomark. ProductLine has a description also. link ProductLine <->> ProductType. [DONE] 12/10

6. Create/apply a migration to add foreign key for product\_type ( in addition to the existing one for leaf-level category) to products. Add column for this to the excel sheet from which the products table is being loaded; and reload the database [ populate this column and reload the database again when get to building/testing the complex case below ] [DONE] 12/10 used products\_all\_wModels\_1018.xls as base.

7. Modify views/leaf\_categories/show.html to see whether it should be listing product\_types or products, and call the appropriate partial. How? For now, put a column has\_product\_types? in the Category table. leaf\_category/show.html.erb line 216ff. [ Test the if ] [ in Modalities/Electrotheraply, chose Accessories or Lotions and Gets. Nether has the type/model distinction and should therefore render the partial 'products'. Does it?

8. A partial for Products exists. Create a new partial for Product Types. [DONE] 12/10

9. There is a views/product\_types/show.html page. Check it when get to step 13.

10. Build a views/products/show.html.erb page [DONE] 12/10 —

— get it to compile; [DONE]

— get it to work: may need step 12 before can test it; need a path of pictures through the tree of landing pages to get here.

problem: the category\_id' foreign key in the file used to load the products table is not filled in, so we don't know have the llc<-—> products mapping we need.

solution: Build a ruby script that can be run in the Rails console to fill this field in in the database records after the products spreadsheet is loaded, to wit:

Product.all.each do |p|

p.category\_id = Category.where( :name => p.category\_l3).first

p.save

end

To avoid problems with l3 category names that occur multiple times in the categories table (e.g., 'Accessories' ) will have to do this more rigorously eventually, but this will serve for the testing of the product and product\_type pages I am trying to do now. It will return the correct Accessories for the initial product subcategory: Modalities/Clinical Electrotherapy.

Returning nil. Why?

Strip products file down to only the products in top level category Modalities to make it easier to look at the loaded file using Razor or Rails console

Reload products table from this file. OK

Confirm that it has 192 products by looking at it through Razor. OK

Try rerunning Ruby script through Rails console against this smaller table again: Runs, but sets all category ids to 1. Why? That would be the case if I were matching against p.category\_l1, but I'm not, I'm matching against p.category\_l3. Look at excel file used to load categories table. Look at categories table.

solution: original Ruby macro confused the foreign\_key (category\_id) and the instance of the class Category that this id pointed to. Macro should have been:

Product.all.each do |p|

p.category = Category.where(:name=>p.category\_l3).first

p.save

end

11. Create a products controller: products\_controller.rb, and put a show action in it. [DONE] 12/10

12. Put in the pictures to get us down to an example of a simple product ( no models, no sizes, no colors).

— pick one: [DONE] 12/12: Modalities/Clinical Electrotherapy/Lotions & Gels

need 4 additional pictures: 710063-00, 710064-000 710066-000 and 710072-000 in the directory /public/images/products/all\_small.

problem: There were no PNG images for 710063-00 and 710072-000 in the directory Bill resent through yousendit last week. Resized the others (..64 and ...66) and placed them on 120x80 white pads in images\_small and 169x179 white pads in images\_medium.

— test the products page built in step 7 above: confirm that you can

(i) get to it from leaf-level category page, and

problem: 1st category in sibling nav is always highlighting, but not set to Lotions & Gels. [ forced for test; rebuild or remove sibling nav ]

problem: one product title is running to a 3rd line and therefore falling outside of the product block in the leaf-level category page. Limit to <n> characters, or change entry in database.

problem: @product instance variable getting passed to Product#show.html.erb is nil. Check controller; check argument being passed through link\_to macro in \_product.htm.erb partial. ( and once confirm that that code works, move it back into partial; it has now been pulled forward in to the body of leaf\_category/show.html.erb ). Was calling ProductTypes#show not Products#show. Products#show was not setting up the instance variables @product and @category needed by products/show.html.erb.

problem: smashed page layout: nothing in main section; footer floating up into main section. Solution: removed divs involved in tabbed interface triggered by sibling nav. Old code kept in show\_wMavics\_Tabs.html.erb.

OK: better: page layout no longer broken, table of products appearing, but individual product blocks are stepping down and to the right rather than lining up; 'a product' looks like its outside the block, etc [FIXED] 12/12

problem: views/products/show.html.erb breaking: not showing product image/description. [FIXED] see below

Removed the tabs at the bottom, cleaned up div structure. Reinsert them using jquery-ui tabbed interface? Mavics one stored in views/products/show\_wMavicsTabs.html.erb. Still no content showing up. Why? answer: unclosed <script> tag.

problem: image not in #left column, description in #right. [FIXED] 1212: redefine them locally within the heading of this file

problem: (TM) not being interpreted; showing up as '&trade'. [FIXED] 12/12

(ii) push orders from it into the standard Netsuite shopping cart

(iii) create a more subtle acknowledgement that you have put an order in the cart

(iv) checkout and confirm that the order flows correctly through to the Netsuite backend. This means that we will need skins for netsuite generated pages. What should they look like?

(v) UI/layout changes:

— large image pictures in product pages?

— zoom on product pages

— use 169x179 images in leaf\_category pages?

— reinsert set of tabs in the bottom of products page. Eventually where will want to put clinicial\_comments, users\_comments, videos on how to use, msds, ... . Ideally would make the presence of individual tabs conditional on whether content for that tab existsfor this product, and the whole tabbed block conditional on whether some tab has content for this product.

— large background images per the Mavics site — but healthcare focused. Pick a couple samples from GettyImages and show Bill what they will look like in the site.

13. Put in the pictures to allow a site user to get to an example of a product that comes in sizes and/or colors:

— pick one: Modalities/Accessories/Nylatex Wraps [DONE]

— Revise the database model to carry model product sets explicitly [DONE] 12/29

introduce class ProductSet with 1:m relationship between

ProductSet <—>> Product [DONE ]— I named it 'BaseProduct' though rather than 'ProductSet' [ Consider going back and using BaseProduct for ProductType, and ProductSet as initially specified here: for the matrix products. In a clothing company you might think of these as the same product that comes in differen sizes; here they really are more like product sets: if the cold wraps come in different sizes, they look very different from one another, and are used for different parts of the body, i.e., they are a set of products, not a single product ( e.g., a T-Shirt) that comes in different sizes and colors. This would eliminate the awkward term 'Product Type' . ]

Modify define of Product to add foreign key: base\_product\_id, and a boolean field to say, for each product, whether it participates in a product set. [DONE] 12/29

create a separate table for product sets — copying information from the products with extension 000 that also have 001, 002, ... 00Ns. [DONE] 12/29

Populate it from the Products table using a Ruby script run in the Rails console

use a Ruby script to set the foreign keys in each of the product records that participate in a product set to point to the product set tuple using a Rails 'id'.

use Rails console to test that for the product set Nylatex Wraps, we can retrieve all of the individual products in that set. [ Do we want to? Or do people just buy from the list as in the catalog? Since there is no additiona information associated with the individual products, I think we leave it that they buy from the ProductSet page. ]

also test that what shows up in the leaf-category for Modalities/Accessories is only the product set Nylatex Wraps and not the individual products within that set. This will mean modifying the view/leaf\_categories/show.html.erb file

Built a views/product\_set/show.hmtl.erb page that displays the product set: image, description, and the list of tms-part-no, size [ and color? ] , price for each of the products in the product set. [DONE] 12/29.

Issues this raised:

1. Add a column to the left of the tms-part-no for letters used to tie the individual line item to one of the images in the collage picture if it is one

2. ProductSets with long names are breaking the sibling nav

3. Images in ProductSet (aka BaseProduct) pages need to be unsquished.. Seems like this could be done as a batch process from Mandy's png originals.

DONE 12/31 for images in the Modalities top level category that we are using to test the three types of product page: simple, matrix, complex

Do it for each of the other top level categories

4. Zoom? on ProductSet image as well as on individual Product images?

5. Some ProductSet descriptions (eg., Nylatex: asterisking which productds are in some group ... ) need to be edited to remove stuff that is done in the catalog but not on the web site

6. Reduce vertical spacing between lines in the products table. ( Tried without much luck 12/31. )

7. Formatting: width of the @products.description field in products that have either short descriptions or ones for whom a wider field works better. Made them all 300px: put <br> tags in descriptions that aren't going to fit in this. Or make them 100% width and insert variable width line of dots before the price.

— test the path from category to leaf-category that contains one or more product sets

— test path from leaf-category image for a product set to the product set page

— test that you can push a suitably qualfied order from the product set page to Netsuite shopping cart ( Might eventually, on Rails-side cart, allow peopel to change the color of something already in the cart, but with NS it will require delting the old items nd inserting a new one. )

**8. complex product pages ( currently 'product\_types' )**

1. hand build the database for the electrotherapy example of the complex case. See that you can get to the product\_type page, and order products of this type, accessories, etc direcltly from that page (without going to a 'product' page).

1. DONE 2/31: but with accessories just as products that are references from complex product types through an 'accessories' association.

2. Make the product\_type <<->> accessories relation an M:N one. And if that creates problems with the 2nd association (product-type.accessories), then pull accessories out into their own table (with exactly the same fields as the 'products' table). Actually do this first before trying to make it work with two associations defined on the same two tuple types. [DONE] 1/2

2. Code the #show page for product-types: views/product\_types/show,

1. Code image, product description, accessories, and ancillary information. DONE 12/30.

2. Cleanup:

1. Swap out jquery-ui tabs library for jquery tools tab library so it can be styled it to match the rest of the site. [DONE] 1/1/2012

2. Remove declaration for the association product\_types.accessories that goes directly to products from product\_types.rb. See if this allows p.save when adding timestamps to all records in the 'products' table to work. [DONE] 1/2. Yes script to add timestamps works again.

3. Insert new 120x80 category images when Mandy sends them. [DONE] 1/2

4. Pull PK's code for add-to-cart, checkout from heroku and insert into latest page templates for simple, matrix and complex product types. [DONE] 1/2

— addToCart button on views/product\_type/show.html.erb not working

5. Pull PK's code for register, login, show\_cart, checkout for the upper right and insert into views/layouts/application.html.erb. [DONE] 1/2

/////////////////////// as of 1/2/2012 //////////////////////////////////////////////////////////////////////////////////////////////////////////

6. Go through each of the complex product-types within Modalities and see that the correct products and accessories are being displayed. Check for missing images.

7. Consider renaming: base\_product —> product\_set ( so that we can later replace product-type with base\_product ).

8. Remove unused columns from products and accessories tables, e.g., product\_line\_description, and foreign keys used in 1:n product\_types.accessories that preceded m:n version of this association.

8. Check to see where there is information in the catalog about product-types, product-lines, or manufacturers that is not in the database and so is not being displayed on the web site. Should we add it?

**9. Get left-nav to take us down to these three types of product page**

1. Clean up highlighting in 2nd 3rd level categories so that it matches that in 1st level categories — ie the background of the visible portion of the block ( the portion containing the words below the image) turns orange when you mouse over the picture or the text, and a single click anywhere within the block takes you to the next level down, or to a particular simple, matrix, or complex product page. [DONE] 1/3

2. Get views/leaf\_categories/show.html.erb linking through to each of the three types ( simple, matrix, complex) by hard-coding the partial(s) it uses to link to the 'product' , 'product set', or 'product type' pages it uses for the three types. DONE 1/3

1. for simple products (using category 7 as test case ) DONE 1/3

— names not going white when hover over the block

— at leaf level should be using -001, -002 jpgs rather than -000 used at levels top, middle

2. for matrix products (using category 3 as text case) DONE 1/3

3. for complex products (using category 9 as test case) DONE 1/3

— protect code against attempt to reference a missing product\_line\_description; causes a page crash on

3. insert the logic — in either views/leaf\_categories/show.html.erb — to determine for each image shown on the page, whether it represents a simple, matrix, or complex product, and call the correct partial (which will in turn call the correct product controller: products.rb, base\_products.rb, or product\_types.rb ) DONE 1/3

— push to heroku

/////////////////////// as of 1/3/2012 //////////////////////////////////////////////////////////////////////////////////////////////////////////

4. Are there cases where we have only a 2-level category tree? If so, remove the assumption from the code that the leaf-level category always occurs at level 3.

5. go back and remove all of the testing code in leaf\_categor#show, Products and ProductTypes controllers and views/products/show.html.erb and views/product\_types/show.html.erb. Test that navigation works properly when it has to deal with 3-level hierarchy with these 3 different types of product page

6. Make left-nav work on pages other than the home page. Turn it into a partial that can be included in each of the templates. This means that the popups for each of the top-level categories that are in the left nav have to be available on all pages that show the left-nav.

6-1. pulled menu out into a partial; pulled generation of hidden overlay content into a partial; included the on-document-ready javascript to unhide the overlays as user mouses over a menu item. Worked in views/home/index.html, but in views/categories/show.html.erb there were conflicts on the instance variable @category.

6-2. wrote two separate pairs of partials:

— one for the home page ( views/shared/\_menu\_popups\_home.html.erb and views/home/\_subcategories ), and

— one for pages built from the categories/show template (views/shared/\_menu\_popups.html.erb and /views/categories/\_subcategories\_of\_top\_category )

The first pair uses the variable @category, and shows all of the menu items visible on the home page: the top level categories plus the company overlays.

The second pair uses the variable @top\_category, and shows only the menu items visible on the category pages: the top level categories plus the search item.

The categories/show pair is also used (as of 1/6) in leaf\_categories/show.

Note: the two partials, \_subcategories.htm.erb and \_subcategories\_of\_top\_category.html.erb are both pulled in by render statements in shared/\_menu\_popups\_home.html.erb and shared/\_menu\_popups.html.erb respectively. views/leaf\_categories/show.html.erb also useses shared/\_menu\_popups.html.erb, and \_subcategories\_of\_top\_category.html.erb has been copied into views/shows/leaf\_categories so that it can find it there. It is a duplicate of the partial \_subcategories\_of\_top\_category.html.erb in views/categories. Eventually one of these copies should be eliminated.

6-3. problems that showed up in 6-2.

If you hover over a menu item it comes up properly as a modal overlay, graying out the underlying page content, but:

— the subcategory boxes within the overlays that come up are slight (5px?) further to the right than those in the base page. This doesn't matter as long as the underlying page is grayed out but:

— if you mouse out of the menu without either moving over another menu item or selecting something from the overlay that the menu item brought up, the modal graying out of the underlying page goes away, but the overlay itself does not get hidden again; it is left on top of the base page.

This is occuring because the jQuery on-hover event handlers set up in the on-document-ready section of applications.html.erb, hide all of the popups initially, and then, when you hover over one, hide all the others, and show the one you are hovering over. But there is no second function supplied to the hover which hides the popup when you leave it. If I supplied this, then I should be able to remove from the first function, the hide of all overlays with class popup.

— No; the problem is that as soon as mouse leaves the menu item, the overlay goes away; the user does not have a chance to pick something from the overlay — which has the next level of menu items in it.

— When user moves mouse off of the menu item onto the modal overlay containing the next level of menu item, we lose the graying out of the underlying page. Would be better if it stayed grayed out.

Sort of want to say: if once you have hovered over a menu item and its overlay has come up, if you move out of the menu item, but NOT onto the overlay, then the overlay goes way. A frequently encountered special case of this is when the user moves his mouse slowly toward and then over the overlay. If the time between when he leaves the menu item and gets to the popup overlay is long enough, the modal shading of the page below the overlay will go away, and then things look strange, especially because of the misalignment of the overlay menu blocks and the similiar looking blocks in the base page (but fixing the alignment, won't really solve the real problem: that the modal shading of the base page is going away). The question is: once he moves the mouse off of the menu, how do we know that he 'intends' to go to the menu, and leave it up, versus having him mouse off to the left. In that case the gray masking goes away, but the overlay currently stays up. Another poor alternative to a real solution: Add a button in the upper right of each overlay that lets the user explicitly 'close' it for the (hopefully rare) case where he exits the menu top, bottom, or left, rather than moving to the right over the overlay. ( actually even right, but above the overlay, causes the same problem: the overlay stays up. )

1st step: run down where the modal behavior is coming from — presumably an option on the popup.show call. Were these popups impelmented jquery-ui displays or jquery-tools overlays?

interim solution: put close image in upper right corner of each of the overlays that comes up; add a jQuery handler that closes the overlay on click. ( Currently closing all overlays class 'popup'; overkill but it works. Need to get a white version of the close image for company overlays that have orange or blue backgrounds; gray one used on white popups doesn't show up very well on the popups with colored backgrounds. DONE 12/7

— hrefs values in 'more' buttons need to be set so links work. Where should they go? to the views/categories/show.html.erb page for the corresponding top level category? Is there one?

6-4 make left-nav active in the four product page templates:

simple

base\_product

product\_set

complex

/////////////////////// as of Saturday 1/7/2012 //////////////////////////////////////////////////////////////////////////////////////////

**10. Other Stuff**

The following tasks were added ( or made explicit) by TA on 1/1/2012:

1. Breadcrumb trail [DONE] 1/2

1. Rebuild it trail to walk up category tree to root regardless of how deep it is. [DONE]

2. Make it into a shared partial, and rewrite all views that include a breakcrumb to render it as a partial. [DONE] 1/2

2. Clean up the UI associatedwith Google CSE search button, either:

2-1. leave it where it is and make google text input box, search button and 'X' fit. [DONE] 12/7

2-2. leave it where it is and do something like Mavics does — essentially unwrapping it when someone clicks on the button sitting in front of it, and at that point exposing the text entry field, the search button, and the clear-text field icon ('x').

2-3. alternatively, move search, register/login and view-cart / checkout to the upper right, above the floating white panels on the screen.

3. Insert **search by competitor's part number**. Assume there will be multiple columns within the spreadsheet used to load the database, for at least 4 competitors. When user types a number or string in the search box in the page footer, search for matching numbers in any of the columns of competitor part nos within the product table. For R1, assume a single column: sammons\_part\_no. Bring up a page to say the search is in progress, and then either a 'no matching part no's found; please call us on our toll-free line at 800-999-9999' or a line that pairs the # entered, [ the compeittor name], and the tms-part\_no, and says whether the match is exact or only similiar. Give user a button he can click to go to the product page for the TMS product shown as a match. = top level item #3. [DONE] 12/8

3-1 add metasearch to gem file [DONE] 12/8

3-2 add code to create search form in footer to views/layouts/application.html.erb [DONE] 12/8

— not putting cursor in field, or showing what you type into the field when you type the sammons part number there.

3-3 add search action to products controller [DONE] 12/8

3-4 add views/products/search.html.erb as template for returning search results [DONE] 12/8

— Broke this when inserted devise-Worden-based registration/login system. Moved out to a partial, and render of the partial currently commented out of application.html.erb

4 Get **scroller on the home page** to work, with text blocks appropriately positioned on each of the images used.

4-1 using Nivo slider: name clash with admin

4-2 using simple slider: too simple; need text and link overlays.

4-3 using royal slider

— slider set up; individual slides lined to 800x800 png images, allowing the background (currently a light gray) to show through.

— titles, text and 'more' button coming in over the slides after slide has slid in from the right as it should. Have the per-slide control of placement that we need.

— remaining problems"

— slides are not scrolling automaticall [FIXED]

— left back-arrow is under the menu on the left [FIXED]

— backgrounds appear to be under the left menu as I want them to be , but since they are not black, don't notice them sliding in. Not as much visual drama as I want.

— backgrounds don't vary, black or white

— at some point in the process of inserting slider, lost the modal nature of left-menu popups; it is working properly in trumedical2.heroku.com which does not have the slider

— cross browser problems: working on Firefox, but not in webkit-based browsers, i.e., Chrome or Safari. related to the following problem:

— if screen width less than 1280, the background images don't appear

— readjust position of left arrow

— remove orange slide blocks at upper right of the screen [DONE] 10/11

4-4. Male 'more' buttons on slides 2 ... 8 active: go to corresponding category page [DONE]12/7

4-5 Remove mavic js code in mavic.js and in headers of ... that has been replaced by royal slider.

5. Get sibling nav to work properly or eliminate it ( Currently commented out )

6. Layout: line up top of the #right-content thing, when it is on the right, with top of middle column. DONE 1/7

/////////////////////// as of 1/9/2012 //////////////////////////////////////////////////////////////////////////////////////////////////////////

7. Put in **Admin subsystem** for controlling the content in the Rails-side DBMS. [DONE] 1/10

7-1 finalized database schema by creating models for USER w/subtypes CONSUMER, CLINICIAN, and CLINIC, that will be used in 8: different price levels for different categories of user.

7-2 include admin gem; run bundle install; run gem installer

7-3 configure: fields to show for each class in the schema, searches, etc. Focus on categories, product\_types, base\_products, products, user, clinician, clinic.

7-4 [ Discuss the Rails -> Netsuite auto-synch as an R2 addition with Bill. ]

8. Deal with **different price levels** being displayed for different categories of user, and users from different clinics which have negotiated different price levels. This seems to imply a register/login subsystem on the Rails front-end so that we know which prices to show the user, and then pass prices to netsuite on the add-to-cart.

8-1 insert devise-based authorization system DONE 01/11

8-2 pull all views in from Rails engine into apps/views/devise/... so can customize them DONE 01/11

8-3 embed views within content\_for body and three nested divs: #outerContainer, #innerContainer, #mainContent. DONE 01/11

[[ This was done in a new directory on my local machine: tm3ror; merged branch (devise) back into master 01/12, and pushed resulting master to origin (git@github.com:cloudsuite/tm2ror.git), so PK could create new javascript for pushing items into Netsuite cart with prices. This included the new stuff on the home page scroller and images modified to work with it. ]]

/////////////////////// as of 1/14/2012 //////////////////////////////////////////////////////////////////////////////////////////////////////////

8-4 Wire it up so that user login controls the prices he sees.

— make Register create a corresponding Customer or Clinician record (or put a 'type\_of\_user' field in the User class defined by devise. ( Rail's awkward handling of subtypes )

— when user logs in, set the global variable @@price\_schedule to something that defines the price schedule he should see.

— Modify the pages that show prices to prices from the correct price schedule.

9. Cleanup the code base, consolidating the css libraries and removing any css libraries that are not being used, and any javascript libraries that are not being used.

— cut style.css, style3.css, compare.css, news.css., ... from application.html.erb.

— When sure, remove the files from public/stylesheets as well.

— Think can also cut webform.css. Yes. DID

` — Renamed left\_nav.css to left\_nav\_popups.css to avoid the suggestion that it contained styling info for the left nav menu itself.

— remove all/most of mavic.js; note that it contains jquery and jquery UI. Have already had to comment some parts of it out. Broke it back down into the original set of files from which it had been aggregated. Remove whichever of these are unnecessary.

— pull declarations out of individual files and consolidate them into tms\_base.css.

10. Home page overlays that run off the bottom of the page: scroll bars within them? DONE 1/14

— Bill to supply content for 'providers' section

— Question for Bill: Get some images into the Tru-Medical, and Patients/Providers/Payors overlays? They are a bit text-heavy at this point.

11. Cleanup overlays that appear on home page

11-1 Contact us: insert email tag (see Bavia 1.0, 1.1)

11-2 Request a Catelog: capture the physical address user wants (printed) catelogs to go to.

11-3 Reduce width of Product overlays to be 20px beyond 4-5 of the in tha row? Done in tm3ror\_bad code base. Put that back into tm3ror. Seems to be an uwanted dot in these overlays.

12. Go through all of the subcategories and products within Modalities: resolve missing images, sibling nav, etc. ...

‑ began that; inserted images of 710018, 710249, 710252; made product images for leaf-level category pages from 710063-000, 710064-000 and 710065-000.

13. Promotions and Tru-Value product blocks in the #right-content section: What does Bill want us to do when someone clicks on one of these:

1. Promotions: [ ... ] Discuss with Bill Monday 1/16 1PM

2. Tru-Value Products: [ ... ]

14. Visual nits:

1. Left menu: orange block that appears when you mouse over a menu item extends 5-10 px beyond the left and right ends of the underlines; make them match. FIXED 12/7 removed left-right margins from div.menu\_separator in menu.css, line 48

2. losing the 2+ px vertical space between left nav and main panel

3. put in large background images on category.show, leaf\_category.show, and the three product pages: simple, matrix, complex.

4. highlight 'more' buttons on home page sliders when hover over them [DONE] 12/7

15. content: product data / images

All, not just category images for modalities

1. Modalities

product data: DONE: files:

pictures: DONE; missing: { <images filenames> }

2. Clinic Supplies

— as only category (using /desktop/tm3ror )

— review this top level category in printed catelog

— There are cases ( like Thera-Band Exercise Products) that is a Product-Set with two Base-Products where each of the two Base-Products has 8 Products. However, without the Product-Set notion in the model, it is going to show up as two separate Base Products: the image and the description will be the same for both, except we could add the "50 Yard Dispenser Box" and "6-Yard Dispenser Box" to their respective text descriptions. Note: this is NOT what Bill is expecting. He is expecting that we have one Product Set, that has 16 Products in it, and we just add a single line to the Product Set description, e.g., "Comes in both 50 yard and 6-yard dispenser boxes".

— Is there a way that we can add a column to the Products spreadsheet that allows us to say which Base Products are in the same Product Set? No, this spreadsheet is really designed to model the BaseProduct<->>Product relationship. We need another to model the ProductSet<->>BaseProduct relationship. Add a column to the BaseProducts spreadsheet, not the Products spreadsheet.

— base\_products in file: products\_base\_1220\_v5\_tlc02\_clinic.xls

— There are 73 BaseProducts in tlc2 Clinic Supplies. How create this spreadsheet? When created the Modalities one, just copied the first product line item for each BaseProduct. This means that the BaseProduct description is in the 'long-description' field. Would be clearer if renamed this field 'description' and dropped the specific product description column. If make that change, then make sure the code for the BaseProduct page template uses 'description' rather than 'long\_description'. It uses 'description' now: views/base\_products/show.html.erb line 144. Test that can remove it and get properly formatted pages before doing so.

— products in file: products\_1229\_v5\_all\_sorted.xls?

—Hand sort entries for Clinic Supplies (72xxxx) on TMS\_part-no - since Excel 'Sort' function not working on column containing TMS\_part\_no (column G) not working. DONE

— go back and hand sort those for Modalities (71xxxx) also. DONE

— Fix load errors [DONE] 02/02

— run application on the loaded data

— subcategories that have no level 3 sub-categories are not loading product images. New to this top level category Clinic Products & Suppliles? No, occured in one place in Modalities: Personal Electrotherapy; was it failing there too? **FIX THIS 02/03!**

24 Balance and Stability — no leaf categories

inserted render 'leaf\_categories/show/24' into categories/show.html.erb for testing. If invoke with /localhost:3000/categories/24 ... . (failing). The modified leaf\_categories/show.html to see if I could make it work, assuming that I got there. Commented out the 3-deep title and 3-deep breadcrumb. Then modified categories\_06 file to include 'complex' or 'sinple' annotation. Invoke test with http://localhost:3000/leaf\_categories/24". See if can get it to render blocks for each of the products contained in this subcategory.

Also tried filling in the has\_product\_type columns for the 3rd-level subcategories of category 36 (Therapeutic Exercise) while I was at it. See if they work.

25 Clinic Equipment — no leaf categories

26 Clinic Supplies — no leaf categories

27 Education — no leaf categories

28 Hand Therapies — no leaf categories

32 Pillows & Cushions — no leaf categories

42 Topical Analygesics — no leaf categories

— subcategories that DO have level-3 sub-categories are loading the images for those leaf-level (level 3) subcategories, but the pages for those leaf-level categories are not loading images for the products **or base\_products** they contain. This was failing in the one Modalities subcategory, Personal Electrotherapy, also. Is this failing only when loading base\_product set images, and working for individual products? i.e., are these leaf level categories ones that ONLY contain base\_product sets, and therefore are showing up empty on the screen?

29 Massage Produts — leaf categories loading, but no products are showing in those categories

**Fix this 02/03** — insert category type in categories\_v6, et al

33 Resistive Exercises — leaf categories loading, but ...

FIXED matrix subcategory;

**not fixed: simple subcategory**

36 Therapeutic Exercise — leaf categories loading, but ...

**Fix this 02/03** — insert category type in categories\_v6, et al

42 Topical Analygesics — leaf categories not loading

**Fix this 02/03** — insert category type in categories\_v6, et al

— views/leaf\_categories/show.html was deciding which partial to call (simple, complex, matrix) based on a field in the categories table, but that field had not been filled in for anything other than one leaf category of each of the three types in Modalities. When I filled it in (with respectively 'matrix' and 'simple') for categories 34 and 35 within Clinic Supplies, reloaded the categories table from the modified excel spreadsheet, restarted Rails, and attempted to open the application in a browser, it failed with the error: " undefined method 'children' for nil:NilClass" in line 3 of views/home/index.html.erb. This is occuring in the (initially hidden) popups for the top level categories ( Modalities, Clinic Supplies, ... ). Why? Why did reloading the category table trigger it? Why does the instance variable @caetegory not refer to an actual category? Because I corrected the name from 'Clinical Products & Supplies' to 'Clinic Products & Supplies' and the code that set up the instance variable @category was doing a search on the name field for each top-level category. Similiar changes to names used in views/shared/\_menu\_popups.html.erb.

In Clinic Products & Supplies > Restive Exercise > Band & Tubing

base product image for 720001-000 exists.

base-product images missing for 720002-000, ...03-000, ...05-000, ...10-000, ...14-000, ...15-000, ...19-000, ...20-000. FIXED 02/02

When I go to page for 720001 (Thera-Band Exercise Bands), two problems: (1) no image, (2) no list of specific products for this base-product.

(1) FIXED: change code to look for images in /images/products/all\_large and batch convert .png images to jpgs with width set to 250px, storing them in both the top-level category specific directory and also copying them into all\_large.

(2) FIXED: foreign keys that linked products to base\_products had been hand set for the one example of a matrix product in Modalities. Do it for all products by running a script in Rails console once base products for each new top level category have been added to the base\_products table that implements the extent of the Rails class BaseProduct. Remember to rerun this macro after adding base\_products for each of the <n> top level categories. [ **or else hand-insert the foreign keys into the excel file that is being used to load the products table — did this for tlc2 ( Clinic Supplies); go back and do it for Modalities.** ]

/////////////////////// as of 2/02/2012 //////////////////////////////////////////////////////////////////////////////////////////////////////////

Batch convert all images/products/clinic\_supplies/large images to .../small height:80px; width: unconstrained. Then copy these into /images/products/all\_small. (Actually we don't need the small and medium sized ones within each category. Keep only the original pngs. Move all of the derived images directly into /images/productds/all\_small, all\_medium, all\_large. )

DONE for Clinic 02/03

Rework this. I changed all of the filenames in the first 2 top level categories (Modalities and Clinic Supplies) to -001 etc. Take them back to -000 and reinstate the code in view/products/show.html.erb that strips the last three characters from the image\_filename stored in the products table, and appends "-000" and then ".jpg". Rename all of the yellow files whose names I changed, back to -000.png. And rerun the batch conversion to small and large files and insertion in public/images/products/all\_small and ... /all\_large. This will avoid having to change the file names for the simple products in all of the remaining 9 top level categories; the pngs are all named with -000.png extensions. ( Actually confirm this before backing out the changes to top level category #2 Clinic. ) We can get away with this because we aren't showing (and don't have) separate images for each item in a matrix-product, e.g., each x in the colors yellow and red. In this release the -000.png images get used for everything: the base\_products and the individual products.

Figure out why the leaf-level category for simple products within /Clinic/Resistive/... is not showing any products. ( remember to set category type in categories\_v6. xls ] FIXED 02/04

Figure out why other matrix products within the subcategories of .../Clinic/... are not showing up. FIXED for leaf categories 33 and 36. 02/04. ( category\_tree\_v6\_ contained leaf-category type only for subclasses of Modalities, not Clinic Products & Supplies ) Stopped at these two rather than all of the subcategories of Clinic, because of problem#2 outlined below.

Problem#1: Expand views/category/show.html to handle either intermedidate categorys or leaf-level categories. This means moving the partials that the old leaf\_level\_categories/show used to invoke into app/views/category. [DONE] 02/04.

— I may have unresolved Git issues. It seemd to have trouble merging the category branch back into master. I have not yet tried to push the modified master to github.

— Why am I still seeing page URLs that include 'leaf\_categories'?

Problem#2: all of the products within a given leaf-level category may not be matrix or simple; a single leaf-level category, may contain both matrix and simple products ( and concievably product types ). YES, THIS IS A PROBLEM: See http:/local\_host:3000/leaf\_categories/41: it has 4 simple products and 3 complex ones: 720094, 720100, and 720310. Since this category is defined as a 'simple' one in categories\_v6, all of the individidual products for the complex ones are rendering with their own individual block, rather than a single block for each complex product. Should be iterating through all of the product referenced by the leaf\_category, and for those that are simple invoking products/n and for those that are complex, invoking base\_products/m where n ends in -001 and m ends in -000. This means that not every tuple in the Product table should contain a foreign key that references a leaf-level category. Each stand-alone product should, and each base\_product ( or eventually each product\_set — should I clean this up now as well ), but not each variant in which the base\_product is offered. This replaces the Category<->>ProductType links and the the Category<->>BaseProduct links. Those associations, and the foreign key's in the ProductType and BaseProduct tables that support them, will no longer be necessary. Instead:

Introduce a class, ProductReference; leaf-categories contain one or more ProductReferences; each ProductReference is of one of 4 subclasses: ProductType, ProductSet (each of which has one or more BaseProducts), or Product. Or, renaming things slightly, have a single Product table which contains fields for 4 subtypes: ProductType,ProductSet, BaseProduct, IndividualProduct. Does this mean that the separate ProductType and BaseProduct tables go away? If so, then the long\_description field in Product becomes the BaseProduct description (not clear: is it also used for the ProductType description? reread that code). Is there any other information in the BaseProduct table that needs to find its way back into the Product table?

Keep class Product. Add fields 'type', 'product\_line\_id' (for product\_type), and product\_set\_id ( for base\_product upgrade to product\_sets each of which has one or more base\_products ). No additional fields needed for base\_products ( several of the fields now in Product, e.g., long\_description, are actually only used in products of type 'base'. They can be left null in products of type 'simple'. Drop the unused columns 'comes\_in' and 'comes\_in2". No such columns. What about the redundant accessory\_for and accessory\_of columns? Drop which?

Decide how to handle image\_filename: should it have \_001 etc extensions on products of type 'sinple', or should we just use the \_000 for now, since for the most part what we have is image files named with \_000.png that can be easily made into files with \_000.jpg names. Note that for Clinic Supplies and Products, I went back and changed the filenames for the simple products to \_001.jpg. Eventually that is what you want if you want to be able to show products in multiple colors/sizs. But if the only images we have now are the generic ones with \_000.jpg extensions, maybe the code that fills in the page templates should be set up to expect images with names \_000.jpg. Change this when/if get separate pictures for individual simple products. Could allow incremental migration, but having the code use whatever image filename is in the database. For base\_products this will be a \_000.jpg; for simple products it could be either \_000.jpg if that's all we've got, or \_001.jpg, \_002.jpg, ... if we have images for each of the individual simple products within a base\_product. this would allow updating of the image-filenames in the excel spreadsheet rather than changing the image file names.

Will have to go through the products\_v7 file and change all of the base\_product\_ids to be the id of the correct product of type = "base" within this file, rather than a reference to the rails-id of a tuple within the separate table base\_products.

— created a test file for products from tlc1 and tcl2. Only edited the first few product sets for tcl2;

— added 3 columns

— added separate rows for BaseProducts

— ran the migrations

— edited the excel file to introduce BaseProduct tuples for the first few matrix products in tlc#2, Clinic Supplies. (thru line 283) Note: BaseProducts are now modeled as a subclass of the abstract class Product, and Single Table Inheritance (STI) is used to implement the class Product and its subclasses.

— changed value in base\_product\_id to be the rails id of the base product for this simpleproduct.

— reloaded the products table to contain these modified tuples for the products within the tlc#1 and tlc#2 ( Note that those for tlc#1 have NOT been modified yet ).

— One thing we are testing for is what should be used in the type field if either the top level class or its subclasses have a multi-word name.

(Note: I have not yet added rows for ProductSets — ProductSets have one or more BaseProducts; lets see if we can get the code back to correctly showing page templates for tlc#2 matrix products with product subsets grouped around BaseProducts. — leave off the case statement for ProductSet )

When get to a leaf level category, search the products table for all products that contain a foreign key reference to this category (in field ' category\_id') , and then, for each product p, do a case statemnt on p.type field, rendering the appropriate partial for products of this type. Make sure that only the SimpleProducts that are not part of a matrix with a BaseProduct have a value for the category\_id field. Where does this code go? Start with views/categories/show.html.erb line 180 ff.

<%=

case p.type

when "SimpleProduct"

render "simple\_product"

when "set"

render "set" # as yet undefined

when "BaseProduct"

render "base\_product"

else "complex"

render "error2"

end

%>

Add the class definitions for each of the subtypes., but you can test it in the rails console by having an initially empty products table, Do they have to be the same as the string in the type field? i.e., "simple, base, set, complex" ? Or can I use SimpleProduct as the class name and 'simple" as the value of the type field in the products table? I think they have to be the samecreating and instance of one of the subtypes of product and see what it puts in the type field by listing the table from razorSQL.

Retain the has\_many declaration in BaseProduct, and the belongs\_to declaration in Product (now SimpleProduct? ) because still need to know which simple products are buyable examples of an abstract 'BaseProduct'. Similarly, need to know which ProductSets contain which BaseProducts. Note: One way of thinking of this would be to say that all of what we have been calling 'matrix' products are ProductSets, and that a given ProductSet may have one or more BaseProducts. Then the template for rendering ProductSets would have to handle ProductSets that had a single base product correctly — i.e., not attempt to print lines for single line characterizations of each of the sbusets defined by a BaseProduct, if there really were no subsets within a ProductSet.

Test this by

1. gettting some simple products to display properly. http://localhost:3000/categories/35 \_ has 3 simple products (ids: 259, 260, 261) and 2 matrix ones. (ids 262,263). Failing when it calls the partial to display the base product 262. Eliminate the partials by moving the code into the body of categories/show.html.erb. DONE 2/6. Works.

2. getting some base products to display properly. http://localhost:3000/categories/34 should display a page with the 9 base products in leaf category 34 (Clinic/Restive/Band&Tubing). Rails console shows that c24.products returns 9 products (check: 9 simple products, or 9 BaseProducts?). And the code calls the partial to display the first of them. Fails because id=nil. Partials apparently cannot see local variables defined within page from which it is called. Eliminate the partial by moving the code into the body of views/categories/show.html.erb. DONE 2/6. Works.

— try passing local variable id from show.html.erb to the partials as follows:

<%= render :partial => 'simple\_product', :locals => { :product => p } %>, or

<%= render :partial => 'base\_product', :locals => { :p => p } %>,

This does not work. The partial does not render: not even the debugging <p>...</p>. Have examples of passing strings to partials and binding them to variables used within the partial from StackOverflow. Is there a way of doing it for variables of type integer.

////////////////////////////////// Monday 2/6 7PM ////////////////////////////////////////////////////////////////////////////////////////

3. gettting some complex products (from Modalities) to display properly within this new structure

— test with 710032-xxx and 710033-xxx in category=9

— insert tuple with type = 'ComplexProduct' into products\_1229\_v8\_thru\_tlc2\_test\_complex.xls

— remove category\_id entry from tuples for 710032\_001, \_002, \_005, and \_010. DONE 2/7

— add product\_type\_id (=24 now) to same tuples DONE 2/7

— fill in product\_line\_id (=3) to same tuples DONE 2/7

— reload products table DONE 2/7

— modify views/product\_types/show.html.erb to display @product\_type.long\_description not @product\_type.description below pic and above table of models that this product\_type comes in. DONE

— accessories not showing up in product\_type page: rebuild the join table (accessories\_product\_types) used to implement m:n relationship between accessories and product\_types to use the new ids of product\_types when the latter are stored within the products table.

— changing the code for images in leaf-level category caused them to be squished. Change it back, or fix the images.

— problems surfacing on product-type page (for product\_type=29)

— on individual product-type page (for product\_type 29) image is missing. Looking for 730033-000.jpg in public/images/products/modalities\_large; change this to .../products/all/all\_large and make sure image is there. DONE 2/7

— why is there a limit on vertical space available for the models and accesories table? DEBUG

4. get code in categories/show.html.erb to walk to level-3 categories ( the 'else' option in case statement, ~ line 212. [DONE] 2/7

Clean up the formatting of product images in the partials for \_simple\_product and \_matrix\_product called from views/leaf\_categories/show.html.erb. DONE. Eliminated horizontal stretching of the image; now center it within the 120px available.

— merged with Modalities

— base\_product data in file: <?products\_base\_1229\_v5\_tlc01\_thru\_tlc02.xls>

— products in file: products\_1229\_v5\_all\_sorted.xls [DONE]

— (R2) adding the abstraction Product\_Sets — at least the code while reworking this section of the site

— pick test product set: 720000-000. with base\_products 720001 and 720002. DONE. 2/7

— new controller ( ProductSets with 'show' action), and page template to display product\_sets containing more than one base\_product (views/product\_sets/show.html.erb). DONE 2/7

— new code in category.show.html.erb to handle product sets as well as the current 'SimpleProducts', 'BaseProducts' and 'ComplexProducts'. DONE. 2/7: additional 'when' clause in 'case' statement.

— confirm that app still works with the new code in it but no entries in the products file tagged with the type 'ProductSet': so none of the new code should be invoked. DONE 2/7. It does.

— new version of products file that has entries for our test product set, and where both of the base\_products within that product set, contain foreign keys linking them to that product-set. Decide whether to model all matrix products as product-sets: some with a single base-product, some with two or more base products. Current code does not work that way; for product sets that contain a single base\_product, the code uses the base\_products/show controller/action the views/base\_producxts/show.html.erb file directly.

— modify file for the test product set 720000. Note: don't really need a tms\_part\_no for product-sets. DONE 2/7

— reload the database DONE 2/7

— test the new code DONE 2/7

— clean up formatting on single and multiple base-product product sets DONE 2/7

— Rework handling of accessories

(added 2/8) There is currently a separate table for accessories, and a join table between accessories and product\_types. Since the dbms implementation of product\_types has been folded into products STI, at least update join table to reference product\_type tuple ids within the products STI table. ( assuming that the join table is not accidentally working because the stand alone product\_types table still exists). Decide what to do about the accessories table: either (1) make its columns match those of products, so I can easily copy lines from products table into the accessories table, or (2) implement accessories within the STI also. The latter seems like the best long-term solutiion, since accessories are a subtype of product: actually a subtype of simple\_product. However, to get tlc#1 (modalities) running, start with option (1). Or even a more limited subset of (1): use the existing accessories table without making its columns the same as those in the products table. Might be able to get away with this if the only accessories are in tlc#1, modalities. Check catalog. Unknown: can an STI implementation participate in the join tables used to model an m:n relationship?

— rebuilt join table accessories\_product\_types using new ids for product\_types from STI Products table; reloaded table; [DONE] 2/8

— if there are no accessories for a type (e.g. Transport? ) check, and then do not put empty table on the page.

— go back thru products within tlc#1 Modalities, and add in tuples for product-sets and base-products. [DONE] 2/8

— since this will push numbers for things over ~200 down, rebuild accessories\_product\_types spreadsheet and database table to use the correct product\_type ids. [DONE] 2/8

— if there are additional accessories ( that are not in the subcategory modalities/accessories ) add entries for them to the accessories\_product\_types join table. Reading the printed catelog, it looks like some things COULD be added the the accessories\_product\_types table. Not going to do so for R1 though; low priority.

— Hand check xls spreadsheet being used to load products table to confirm that (1) only simple-products, product-types, base-products, and product-sets have entries in the column (category\_id) used as the foreign key to link entries in this table to leaf-level categories. Check that these category\_ids are correct for products within tlc#1, Modalities. [DONE] 2/8

— reload the products table from test file of products from tlc#1 and tcl#2 that I have been using to keep load times within reason — products\_1229\_v8\_thru\_tlc2\_test\_complex.xls (\*)(\*\*). Problems: accessories showing up within leaf level category pages.

(\*) will having the word 'Accessories' in the type column when Accessories is not defined as a type yet, cause problems?

(\*\*) Why are there 6 extra entries in the categories\_id column? Major alignment problem?

— test tlc#1 from the web site. Confirm that products are now showing up for the leaf-level categories that were not showing products earlier — and that they are showing the correct products. Record results of testing in tm3ror/test/Testing\_tcl1.xlsx.

— Clean up tlc#2, Clinic Supplies/Products, entries also: (1) all entries that should show up in llc do, (2) only those that should show up have a category\_id foreign key, (3) category\_ids used as foreign keys are correct — that is, they match the ids in categories\_v6.

— reload the products table, and test tlc#2 from the web site

////////////////////////////////// Tuesday 2/7 1 PM ////////////////////////////////////////////////////////////////////////////////////////

3. Tables & Traction

4. Fitness

5. Orthopedics

6. ATC/Taping

7. Evaluation

9. Dailing liing Aids

10. Lymphedema

11. Wound Care

Recap (1/12) of principal UI stuff to clean up:

1. Tru-Medical, payor, provider, ... overlays on home page that go below bottom edge of page — and currently give user no way of scrolling content within the overlay. FIXED 01/14: put in scroll bars.

2. alignment problem on home page

— remove 15px left margin from 'subcontainer' containing 'menu' so left side of menu on home page and others lines up. [DONE] 1/14

3. alignment of the register/login, cart/checkout stuff in top right of pages

4. making left menu 'live' in the 3 types of product pages: simple, matrix (with one or more product-sets), complex

5. register/login pages:

— remove left-nav to make them look like add-to-cart / checkout pages on Netsuite backend? No: Bill prefers to make left-menu live on these pages, and add Tru-Medical logo;

— on register add option to email or call to set up a new clinic newclinic@tru-medical.com.

clinic# or id, or type the name of the clinic; verifiable for us and simple for them. or main-phone number of the ABC Physical Theraphy. Handle misspelling? Verify them. Don't have the email addresses for existing clinics. 12345 Apple777: walking into that account to set them up online; here's your account number, put in your email's; show them price drop.

6. product\_type.html.erb

— make description column within table long enough that products fit on single table lines. [DONE] 01/14: 390px.

— remove tabs that don't have content for: patient and clinician referrals

7. modal nature of overlays when click menu items in left menu

— failing on home page; now failing on inner pages too. ( works for inner pages in version on heroku. )

8. reinsert search-by-competitor's-part-no when resolve clash between authorization subsystem and/or admin subsystem

9, modify views/devise/register/new.html.erb to let user put in clinic he is associated with (if any); set global session variable @@applicable\_price\_list; use this to generate clinic-specific prices on simple, matrix, and complex product pages ( product/show.html.erb, base\_product/show.html.eb, and product\_type/show.html.erb, [ and product\_set/show.html.erb ]

**Tru-Medical: Release 2 goodies**



**R2 UI changes/additions**

1. build a version fo the site with colored circles on the main menu items, and background colors in each of the sections that match: perhaps just color backgrounds rather than the images used in R1.

2. complex types: with multiple thumbnails and zoom on product\_type page. Same for matrix and simple products?

3. hide Google CSE search until someone clicks bold SEARCH entry in left menu; see if you can avoid loading the Google search until that point; will improve download performance for pages that contain the Google search significantly.

**R2 functionality**

1. add support for multiple price levels by replacing Nesuite register/login subsystem, with one we control in the Rails front-end

1-1 Allow Tru-Medical to grant a clinic one discount structure, and then to discount a set of individual products still further to meet local competition

1-2 Dealer pricing?: see 3 below

2. Add ability to search against part numbers from multiple (up to 4?) competitors, not just Sammons.

3. Dealers? / Affiliates?

4. add Rails->Netsuite autosych of changes to products, categories, images

4-1. Three options for synchronizing product data: (i) general schema mapper built around open-source T\_\_; (ii) schema mapper for TruMedical database and Netsuite setup; (iii) break data on the Rails side out into the same tables we are using to support the object model in the Rails site.

4-2. Pushing images to Amazon S3 and synchronizing.

4-3. [ Pushing Video to a CDN: current HDDN, Akami/Brightcove, [ ... ]

1697; this customer is a company

william king physical therapy

in to account

to financial

price-list: tru-blue

Alan and

William King Jr.

Tom Atwood PT:

terms: ne 30

price list: corpoate 2

credit limit

addresses: