General Explanation: This Item Wizard is the first module of our User based System. So at first the user has to sign into the system and can then select this module if he has the permissions to do so. The User may save the item after step 3,4,5,6 and continue the work on it later (or another user). After saving the user has to specify if the item is ready for generation. As long as he edits the item it is not ready for generation. Our data structure is based on dropdowns and data fields. Dropdowns are named lists which references certain values to be part of it. Most data is entered via dropdowns. As new data in a dropdown may be needed during the process there should be a very easy and fast way to add a new value to the dropdown which will by asynchronously saved to the database for the future. If the dropdown is part of the 'from'-Part of a matching, the missing matching entities have to be set. One Dropdown must only be selected once (with exception of a 'is_many_times' dropdown). So if a certain Dropdown is part of multiple Steps or Patterns it will be auto filled. In case of 'is_many_times' dropdowns there exists multiple values in a ranked order. Then for the pattern set the value that matches the rank in pattern part. Matchings can be defined for Dropdowns. So e.g. the Dropdown ManufacturerColor matches automatically the Dropdown GermanSimpleColor. The user only has to define ManufacturerColor and the dropdown GermanSimpleColor is determined automatically (live) via the matching. Data Fields are e.g. Numbers or free text where it wouldn't make sense to use a dropdown like e.g. the Item Number which is almost always different. Via Dropdowns and Data Fields we are able to determine which Basic Data the Item has and e.g. filter items with a particular color,... Via Patterns we are able to collect Basic Data and to write texts in a very simple and structured way for the user. If the flag 'is_many_times' is set for a dropdown, then this does not state that it's a multi dropdown but multiple dropdowns of the same kind which can be selected once each. All these values get a rank, so that an unique order is defined. Set Items are Items which are only selled together e.g. a purse and a keychain. They share the basic independent data like item number,texts... but for any further data they are different. The user defines how many set items are part of this item and then he returns to Step 4 if there are more set items to be filled. Step 1: The user is asked if he wants to copy an item to use it as template for the current one. Therefore all fields are pre selected like it was with the copied item. In the best case the user only has to change a few things to create a whole new item.

Step 1: The user is asked if he wants to copy an item to use it as template for the current one. Therefore all fields are pre selected like it was with the copied item. In the best case the user only has to change a few things to create a whole new item.

Step 2: The user enters the EAN. Technical Remark: he may also use a technical device to scan the EAN but this should not be relevant for you since it only simulates keypresses.

Step 3: see explanation there

Step 4: see explanation there

Step 5: see explanation there

Step 7: The user may directly create a variant of the currently created item

Step 1 - Entry Point for User search for items as template with: item number model manufacturer If an item is copied all dropdowns values, patterns item text etc. should be adopted and already preinitialised. **L-----**Parameter: ean to copy Item Wizard Enter EAN save to DB save into table **items** the EAN with is_ready=false Step 3 - basic independent data of Item/Set show all datafields and dropdowns from tables 'data_fields' and 'dropdowns' with importance='obligatory' Remark about is_many_times : Normally the user has to select one value in the dropdown. But something there may me several These Basic Data are obligatory for each item and set and for each set unique. values to set for one attribute. Therefore we introduced the 'many_times' dropdowns. Note that Nonetheless they also work with the dropdown / data field structure. we **don't** mean a HTML select with multiple attribut. It should be more like e.g.: All inputs shall be generated from DB where is specified which dropdown/data field A user selects a value, the same dropwdown appears again below to select a second value. That is also the reason why we need a ranking for saving values for 'many_times' dropdowns. is needed here and which type it has The user must have the ability to modify the ranking of all is_many_times dropdowns and data fields. Similiar for data fields. Each datafield and dropdown belongs to a group, which is defined by the group_id. Dropdowns/data_fields of the same group should be displayed together (maybe seperate groups by headings with *group_name* found in table **groups**). The groups should be ordered by there position found in table **groups**. The dropdowns/data_fields inside a group should also be ordered depending on their own column position. number of set items (integer) Packaging Measures Height (double purchase price (double) ackaging Measures Width (double) Packaging Measures Depth (double manufacturer (dropdown from DB) Weight (double) item number (string) product name (string) (Dropdown from DB) (multiple) occasions season type (dropdown from DB) (Dropdown from database (multiple) Visibility type (Dropdown from DB) Product type (dropdown from db) or each value save values in tables: • 'item_values': with EAN from step 2, set_no = NULL, either the dropdown_value_match_id or data_field_value_match_id of the value and the rank in case of is_many_times = true (else rank = 1) 'dropdown_value_matches': in case of a dropdown value, save dropdown_id and the value's text_value_id 'data_field_value_matches': in case of a data_field value, save data_field_id and value as a text value is ready for generation? This question should not be a seperate view but e.g. a button at the end of the page (next to the continue button). number of set items > 1? ______ set_product_type dropdown. Set a set item's product type. Every item in a set has it's own type set item's product type (multiple) _ **- - - - - - - - - - 1** Step 4 - dependent further data of item Matchings: Matchings are defined in matchings with a name and a unique id. The matching definitions are defined in matching_definitions. These dropdowns/data fields are dependent from the basic data of Step 3 (and 3.5). A matching relates a set of dropdowns ('from') to another set of dropdowns ('to'). They also match the dropdown/data field structure and are generated from the DB. Whether a dropdown/data field is shown here may dependent on the product type For each matching we have several groups defined by a group_id. A group of a matching represents a relation between specific values of the dropdowns. Therefore there exists a column for the *matching_id*, the matching *group_id*, a *dropdown_id*, the *text_value_id* belonging to the dropdown_id and an optional rank (in case of many_times=true for dropdowns) and the matching_dir (matching direction) which is either 'from' or 'to'. For an example see the mail I wrote some time ago. The idea is now: For every product we have a set of dropdowns and data_fields (see the other boxes how this set is defined). If now any matching is fullfilled that means that for all 'from' dropdowns with same matching_id and group_id the related text_values_ids match with the one the user selected for this product, then the corresponding 'to' dropdowns for the given matching_id,group_id pair should be set automatically to the values defined by the matching rule. > no translation needed because of generation via DB -> only examples here E.g. the same example that was already send via Mail: If the user selects 'black and white' as 'manufacturer color', then we have matching in our database (matching_id = 1, group_id=1) that is fullfilled. Therefore we set the 'German color' dropdown to 'Schwarz' and to 'Weiß' (remember that this is a many_times dropdown). If the user changes the color to 'royal blue' then the matching with matching_id=1 and group_id=2 is fullfilled, therefore we remove the german colors and set it to 'Blau'. depending on the product_type_id which will be set in step 3 via an obligatory dropdown, show all dropdowns/data fields that are listed in tables: -> aus DB relevant_dropdowns and relevant_data_fields Kunden Maße (Zahlen) So all entries which matches with the product_type_id zu deutscher Farbe Attention: If number_of_set_items > 1, then a set_items_product_type is defined. In this case: use the set_items_product_type from step 3.5 for determining the relevant dropdowns /data fields **Remark:** Both product_type and set_item_products_type are is_many_times dropdowns. You have to display all data_fields/dropdowns that belong to at least one of these product_type_ids or in case of sets only to at least one of the set_items_product_type_id . Tabletgröße (Tablethülle, Tablettasche) Geldbörse Maße halboffen (Geldbörse) und noch mehr... for each value save values in tables: • 'item_values': with EAN from step 2, the set no, either the dropdown id or data field id of the value and the rank in case of is_many_times_dropdown/datafield = true (else rank = 1) the set_no is =1 by the default and may be decreased in later iterations save to DB 'dropdown_value_matches': in case of a dropdown value, save dropdown_id and the value's text_value_id 'data_field_value_matches': in case of a data_field value, save data_field_id and the text value is ready for generation? Step 5 - dependent Patterns Patterns are also text templates which also uses the dropdown/data field structure. Depending on the product_type_id which will be set in step 3 via an obligatory dropdown, A Pattern may look like e.g. This purse has the color [ManufacturerColor] ([GermanSimpleColor])." show all patterns that are listed in relevant_patterns, matching the product_type_id . Attention:If number_of_set_items > 1, then a set_items_product_type is defined. "This bag has a [SidebagType] with a [LockType] lock and is suitable for [SuitableFor].". In this case: use the set_items_product_type from step 3.5 for determining the relevant dropdowns / data fields Note again, that product_type and set_items_product_type are is_many_times dropdowns, that All Patterns are defined in database and are generated from it. contain multiple values, so check for all. There may be live matchings needed e.g. GermanSimpleColor is determined from ManufacturerColor via a matching so the user only has to select the ManufacturerColor. Only show the ones which are not optional (=> optional=false => and so obligatory) and add a possibility to add patterns with optional=true. For searching a pattern use the pattern_name in patterns and the keywords in pattern_keywords Some Patterns are compulsory and some are optional. The user decides which optional patterns he will use. Note that it should be possible to use patterns multiple times, so we need an option for the user to duplicate the pattern. The visibility of the patterns depends on the dropdown/data fields which are part of it. If a dropdown/data field In relevant_patterns you get the pattern_id which refers to the patterns table. matches for e.g. a product type the pattern is shown for the user to be selected. Patterns should be ordered according to their *priority*. In some patterns declension of dropdowns will be needed because of the german language. Therefore the pattern parts are merged to the complete pattern in order defined by position. add_right_space defines if we need a space to the right of one pattern part. A pattern part can either be a dropdown or a data_field or just a constant text value which is defined by column pattern_part_type_id which refers to the corresponding type in pattern_part_types. In the first two cases either a dropdown_id or a data_field_id is set. In the last case only the text_value is set. It is possible that a pattern part contains dropdowns or datafields that were already set in step 3, 4 or already by an other pattern. (so dropdowns/data fields with the same ids). In these cases the dropdowns/data fields should be initialized with the values set in the steps before. In case of a dropdown/data_field with is_many_times = true there may be multiple values. For these cases pattern part has a column rank that defines which value should be used for initializing. If rank = NULL or greater than the current maximum value rank, then the data field / dropdown should stay empty. If the ranking of a dropdown/data field is changed, the preset values should update automatically. In case of is_many_times=true the user is allowed to change the preset value. If a user changes a value, this value should stay fixed and should not be overwritten when the ranking changes. select and config patterns For each value that was not already saved in a previous step save values in tables: "item_values": with EAN from step 2, the set_noeither the dropdown id or data field id of the value and the rank (in case of is_many_times_dropdown/datafield = true, else always 1) 'dropdown_value_matches': in case of a dropdown value, save dropdown_id and the value's text_value_id 'data_field_value_matches': in case of a data_field value, save data_field_id and the text value so in case the value was already selected in a previous step (3,4) the value should not be saved twice. save to DB For each used pattern, save a pattern instance in **pattern_instances** with the product's *EAN*, the *pattern_id* and the rank of the pattern. The pattern instance's rank defines in which order the patterns are sorted if the same pattern is used multiple times. For each pattern instance we have to save the values that belong to the pattern. These values are saved in item_pattern_values. For an entry we need the item_pattern_instance_id . Each pattern consists of pattern parts, so for each pattern part its value have to be saved. Therefore set additionally the pattern_part_id. Note that all values belonging to the product are saved in item_values. If the user has selected a value for a pattern part, we only have to reference the item_value_id in item_values is ready for generation? If the value was set automatically, set item_value_id=NULL . (Note that in this case we can automatically detect set value by searching item values for the dropdown_id and rank.) Whether the value is automatically inserted is determined through the rank in pattern part. If the rank is NULL then the value won't be set automatically otherwise it will. further Set Items required? If an set item is added, the set_no is increased by 1 for the next savings. (because of number of set items) Step 6 - texts Item texts are only stored once for one product (independent of number of set items). Item texts are stored in **item_texts** with ean from step 2. For every column in the table we need a corresponding text field. For the short description and item text we need a text areas. save to DB is ready for generation? *** * * *** set is_ready=true in table items for the EAN from step 2.

use item as template for next item?

Yes, Parameter: created EAN

These yellow boxes gives some hint about the database structure:

bold words relate to table names

italique words relate to column names/values