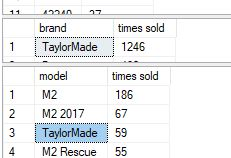
## Transition to Single Item Table

Visual

- Easier to read export items

- Inventory searches won’t need to be defined by item type

- When adding inventory an item won’t need to be defined with an item type



Background

* Quicker searches
* Easier to manage
* Some restructuring will be required to add in the ability to average costs when adding items into stock.
* Additional functionality
* Eliminating erroneous data
* Real locations for items instead of status’
* Cannot distinguish between and actual Item and a Service for COGS.

Better inventory control, to replace Caspio certain data rules must be followed. Inventory must have a proper flow and proper control, this includes: transferring items between locations (packing slips), end to end tracking of where all items are (location, in transit), only selling items that are at the physical location.

The more important question to ask Carlo is: What do they want this system to do? Is it going to be a fully developed application or just something to use to make sales with? Do they want Inventory Management?

Changing to a single table and updating must be completed for proper inventory control and data flow.

To put the change off now and then decide to do it later the more it will cost to complete. The longer that we wait the harder it will be. Right now, the data is more manageable and the change makes sense.

Prep Work for Process

* Verify current Models and Brands to be used with the Sweet Spot (Jim): 6 hours
* Restrict all Items to the Brands and Models listed in the database: 6 hours
* Change Model and Brand table to include a column for Active: 4 hours
* Fully structure the Dependency between Brands, Models, Types, and Product Groups (for Cost Averaging) [ex. Brand = 3 then Model can only be 10 or 11]: 32 hours
* Create process for Admins to add new Brands, Models, Types, and Product Groups with the proper dependencies: 32 hours
* Create a list of functions and procedures that will need to be changed to utilize the new single table structure: 6 hours
* Build New Items table for all types of items: 4 hours
* Manage the list of functions and procedures that are getting updated to avoid duplicating work: 4 hours
* Update all the functions and procedures to use the new single table structure: 16 hours
* Create all conceivable Use Cases for testing of the new structure: 8 hours
* Testing of each Use Case to determine if the new process works or fails: 32 hours
* Review of all failed Use Cases: 4 hours
* Create Fixes for Failed Use Cases: 4 hours
* Build Temporary Table to mark which items have been updated from the start of the sorting process: 6 hours
* Sort through all current Inventory to match it to the new table structure: 32 hours

Total Estimated Work Time Involved = 196 hours

Three Weeks at 40 hours each for two programmers = 240 hours

Giving 44 Flex hours in case of unforeseen problems.

We can also have Jim and/or Stephen complete some additional testing with day to day scenarios.