## Tabled Items:

**Editing Price of items during a Sale:** Discussed and not implemented was a process to change the price of an item during a sale so they didn’t have to enter a negative discount for some items since that looks bad on the invoice. I see this as more of a data integrity issue. Constantly changing the price of an item (No Cost Service item) can cause more problems.

**Alphanumeric SKU numbers:** Changing SKUs to be alphanumeric and to designate where they were entered into the system still needs to be determined if it is a need. Including Vendors in the item table so that they can identify where the item came from will need to be addressed if they want to include that or not. Finally including a creation date and last update date to the item table has not fully been decided.

**Edit Multiple Items at Once During Sale:** Upon investigation it is determined that the ability to edit multiple items at the same time during a sale will require a whole new function separate from how it currently works. There are two possible options. To fully rebuild the way sales are processed or to just build a new process for possible “Bulk Sales.” Some key questions are how often do they need the ability to edit multiple items at once? What are the fields they would like to update? If this is only happening a couple of times then there may not be any added value in completing this request.

**Bug Fix for Sales Process:** Carlo mentioned that after so many items on the sale they processed it then went onto a second sale to make sure they didn’t lose any information. With this fear that will mean that we need to fully flush out the sale process so that everything is protected, verified, and backed up properly to prevent the lose of information.

**Invoice with Too Many Items:** It has been discovered that when an invoice is too large and a user is trying to view it that the application makes too many calls to the database and thus causes the system to freeze. This is another issue that has been logged in our bug folder for repair. Some time will need to be taken to go through current bugs and make a more streamlined process for many functions in the application.

**Return Trade Ins:** Upon investigation into this process since Trade Ins are not being added into the club table and have a separate process that it is not a simple task to just complete a refund for these Trade Ins. When the application was first created the discussion surrounded that when they accept a Trade In it would then automatically be added to the clubs table with it’s own new SKU number and could then be sold right away. Why was this process abandoned? Was there a specific reason for the Trade Ins to stay separate? If we were able to easily identify them in the clubs table then it might make this process more simple and we could implement it easier. If they decide that they do not want these Trade Ins in the club table then the full return process will need to be reworked to make it possible to process these as returns.

**Negative Discounting:** Is this something that should be in the application? Or should the system prevent a negative discount? Is there a way around this? What does the system need to do in place of a negative discount?

**Cashout Revamp:** Is this the proper process to do an actual cashout? What specifically are they trying to achieve here? A normal cashout would just be the employee counting and recording the money left in the till and then processing the figure. Is there a reason they need to compare their other forms of payment with each other. For Credit Cards and Debit purchases there should be no reason why these wouldn’t match. Do they want to track back to 8/01? If yes, then need their printouts for each location back to then.

**Overall Clean Up and Bug Fix:** There are currently errors that occurring in multiple places of the application. We need to figure a way to trap these specific errors and locate a fix for them. There is also some process that could probably be cleaned so they run more smoothly.