Point of Sale (POS) Software

Black text – Gray

Blue text – Adam

Considerations

* Audience (end user): customer or sales rep?
* A POS would be used exclusively by sales reps, no? So design should center around their needs for completing a sales traction, tracking inventory, and commissions(can omit)
* Tie into ecommerce or standalone POS?
* Initial I would design a standalone POS that feeds backend to a basic inventory management system. However, I would create a mock-up of some ecommerce need in order to structure products, pricing, and provide some header information to POS transactions; could build out this idea to a fully-fledged ecommerce if Schwartz deems initial idea to small.
* Pricing – varied between customer accounts or same across the board?
* To add an additional layer of complexity I think the idea of tiered pricing based on customer loyalty or membership might be perceptive
* Different permissions for different POS user accounts?
  + This is really an infinitely complex idea but we could dumb it down and do two permission categories: one “read only” permission that could only perform transactions but not edit customer / product information and one “all powerful” admin permission
  + Two user system, one for standard employees and then a manager for pricing overrides, void transactions, returns, ect.

Infrastructure

* Web application
* Python / Django

Functionality

* Customer management
  + View / add / delete / edit
  + Discount management?
* Product management
  + View / add / delete / edit
  + Subtract from inventory if something is bought
  + Idea could extend to tracking sales by date/timestamp and used to forecast business trends. Could be built out to include a PO system for product replenishment if Schwartz feels initial POS project is too small based on sales history or alerts of low inventory levels. Spit balling on low inventory alerts, could include snapshot of weekly sales trends.
* Perform and manage transactions
* Create standardized system to import data from different software? (possibly too ambitious)
* What? ^
* ^^^^^ Terrible wording, but the idea here is a dilemma between two forms of packaging our software: 1) if you buy our software, you MUST start from scratch and individually add each product and customer into the system, and 2) if you buy our software and are switching from a different software, there is an importation process in place that can transfer all of your stored data for products and customer into our software.

I think we have a lot to work with here and can extend these ideas to give ourselves plenty of work. Great concept!