Loyalty Points, Coupons, Promotion Codes, Smart Cart Stuff, and other sales options – 2/8/17

Steve, Alan, Eric, Brandon

* The QR codes might end up helping us take rules and settings into the cart. Meaning, it may take some sort of id and/or instructions to the cart. The cart would then break things apart, parse what is needed, and then apply the values and/or look-up the rules accordingly. Basically, let the QR code help transport some of the logic pieces and/or directions of where to look things up at.
* One of the older methods was built using flex grid and a small self-contained list of prices for known quantities. For example: say you knew that it went 1/8 oz, ¼ oz, ½ oz, 1 oz (full). They, the users, would just add a pricing list such as: 5,9,16,27.
  + One of the benefits of the flex grid was it was a simple list
  + It still needs to be maintained
  + They have to be in a certain order and could break if setup wrong
  + This is somewhat of a hack of sorts
* The ideal would be a table or database type model so that it will be more set fields and structure
* One of the goals is checking the rest of the cart and seeing what else plays into the pricing tiers.
  + This needs to be a pre logic type action. This logic could then change the values in the cart.
* Possible problem with QR codes – If something changes, the QR code (old label) could go stale. If the QR code held all of the logic, it could change. If it only held an id number, it could always just look it up and then roll from there. That would be better.
* Calendar based sales and promotions – date and time based pieces.
* The smart cart is the logic side of the puzzle. The brains.
* Loyalty points – basics – clear back from a paper and pencil based model – usually you have a start date, a way to set an initial balance if needed, some sort of ratio between dollars spent to points. This could items sold, dollars spent, repeat items, etc.
* We need a stored value that is able to searched and manipulated via an actual one-to-many type relationship. We need the concept of an account of some sorts. If you only rely on a on-the-fly calculation, it gets too crazy. You run into crazy logic, time spans, rules, etc.
* What about special deals, double points, points earned in a certain time period.
* We need a place to store the earned points and the redeemed points. It really needs to be tied to promotion codes, campaigns, specific sales, etc.
* What about family members? What about sub customers who get to use a parent discount? Other crazy models.
* If we store points and/or credits, we could then start basing logic off of that. If we do store points and/or credits, that would need to take place when the cart converts from session into a real database.
* We may need to control the usage of points and/or credits. We don’t want to go into a negative points or negative credit. Image a person who has 2,000 points but wants to use 2,225 points. That shouldn’t work and/or happen. We have a max of 2,000 for this transaction. We just need to watch things.
* We need a centralized service that watches things. We are getting loyalty points in the cart, in the receipt, and in the customer log. We need to standardize things.
* We may need this same centralized logic to show up in ecommerce and outside the secured environment.
* We need to help catch part of the story… not just showing points, but why do they get certain points for certain things. That is huge and great logic that helps tell the story.
* Can we combine loyalty points, gift cards, in-store credits, coupons, punch cards, etc. ??? They are all some kind of a liability of sorts
* What about vendor credits? What about customer credits? What about general non-denominational credits (not tied to a specific vendor or customer)?
* We may need to marry the in-line discount campaigns with the QR codes, with the coupons, with the possible loyalty points. Basically, sew things together.
* Is this standard and/or an add-on that needs to be setup and/or charged extra?
* We get into max, min, expiration dates, time periods
* On printing QR codes and promotions, we need to generate our own codes and then help the publishers either generate their own QR codes or help them know what is needed.
* What about a serialized coupon or a single one-by-one code. Or do you just record when the coupons are used and the system could check to see if it had been used by the current client or not.
* What about a digital punch card?
* This whole thing has been a huge problem since the beginning of time. We just need to help use the system to help track what we can and then show them what we have.
* You could get away from print ads and get them into a digital world where each QR code and/or coupon would be customer specific. That takes it to a tighter level.
* Eric was talking about coupons on coupons… basically, a simple entry level, then prompting for some info, and then creating a more specific code.
* If we make a new table, we just need a category and/or type to differentiate between loyalty points, in-store credits, coupons, punch-cards, gift cards, other…
* Coupons and promotion codes need to break out to a sub table that has rules and other restrictions and/or details.
* We need to show some of these liabilities and assets (vendor credits) on the balance sheet.
* What about special line items that could get skipped and/or mapped directly to the balance sheet. We need to make this as smooth as possible.
* We need to be able to search gift cards, search coupons, search promotion codes, etc. Maybe its own home page. We need to see balances. We need to be able to add to and subtract from an existing balance (load and unload).
* We need some reporting on what is still outstanding and/or still owed.
* At one time do you ever write things off and clear the accounts. For example, is there a expiration date on the points/credits?
* Do we want to assign any dollar values to those points and/or credits that are out there? The owners may want to see this in their balance sheet.
* We may need to look into some other products to see what they did. Basically, doing some research on other products.
* Some of the people use my cart favorite buttons to do things like bogo (buy one get one). We may want to tie in some of the my cart favorite logic into promotions, sales, discounts, etc.
* As a note, we have some other notes dealing with discounts, exceptions, and other tracking stuff.
* One of the goals of all of this sales and promotion stuff is getting the data back to see what is working and what is not working. Tracking a campaign is huge. What to use or which ones are working…. That’s what they want to know.
* What about special reports that group and show the campaigns, the promotions, the sales, etc. Help the owners and managers see what is working and what is going on. Basic analyst stuff (sums, max, min, averages, trends, etc.). How effective was the money that I spent for my marketing campaign?
* If we use the same table, we may want to have a flag that helps us know if we are dealing with assets (who owes us) vs. liabilities (who do we owe or potentially owe).
* What about in-house deals… single use, multi-use, time-based, reoccurring, etc.
* Make things configurable… this deals with naming, passwords, aliases, rules, assignments, etc.
* We are seeing a new table for the main account. This can be tied to a vendor, a customer, or a generic account. We then need a table that allows for transactions (load, unload, get, use, etc.). We are also seeing another table that deals more with the promotion codes and campaigns. We need to interconnect all of these pieces.
* Exclusions, per item, per category, rules, and logic. Take things out of the hardcoded level and make it data driven logic and rules. That is somewhat of the goal.
* We could still allow free-form discounts and such, but some of the new pieces will really start helping to define the usage. Another way is to define one of the campaigns as fully open. Either way, we need to group things and bring back the data in a report.
* See a small drawing… we had key players including vendors, users, customers, unknown people (generic), invoices, PO’s, and items. Those pieces already exist. We then create an special tracking account table that could tie to any of those people (persons including the general or unknown). That table would hold the who and the what type of account was it (asset or liablility – who owes who money and/or a money equivalent). We then need a sub table that would allow for the accounts to be loaded and unloaded (positive and negative amounts). The other main table would be the promotion codes and/or campaign table. This holds special rules and assignments for items, parts, discounts, pricing, etc. We then would tie them all up together.
* Eric is in favor of bitesize chunks and multiple deliverables… keep it simple.
* Just a question… what about multi-tiered affiliate programs. It has similar needs (account, transactions, and promotions).
* As we create these new tools and features – who will babysit things? maintain things? and what kind of setup is needed? We can make some good money doing custom code, but it might be really awesome to make it part of the whole package.
* As far as funding… Is this an add-on or do we just take into our family model and kick back a percentage back out to the developers who are playing? Let’s look into a monthly percentage kickback type model.
* If we can easily add it to the standard, then let’s do it…
* As a note, we have put some rough costs (development prices) in the adilas community funded projects.
* Support? Who is going to do support? Or do we make it tight and then just roll it in to the whole system. Get more people involved into the project.
* Our plan for next time is reviewing the notes, making a plan, and starting layout database tables. Take small steps, kick things around, let it simmer, and build on last pieces. Treat it like the sub inventory project. Multiple doses, dipping in and out to figure out the game plan.
* Part of this plan is pulling in some of what Eric has already done and making it part of the bigger picture. That is pretty cool. As a side note, instead of paying for one-time custom code model, our goal is somewhat pulling things into a reoccurring revenue model. Pretty cool! ☺
* Steve was talking about a family type model, a trust, and a virtual coop type model.
* Not part of this conversation but… Eric has some expertise in servers, load balancing, database stuff, performance tuning, monitoring, checking for bad code or checking for non-efficient pieces.
* There is not a handbook for what we are doing… We have to look to other companies, history, and we understand that we are somewhat of a pioneer and/or explorer of one way or another.