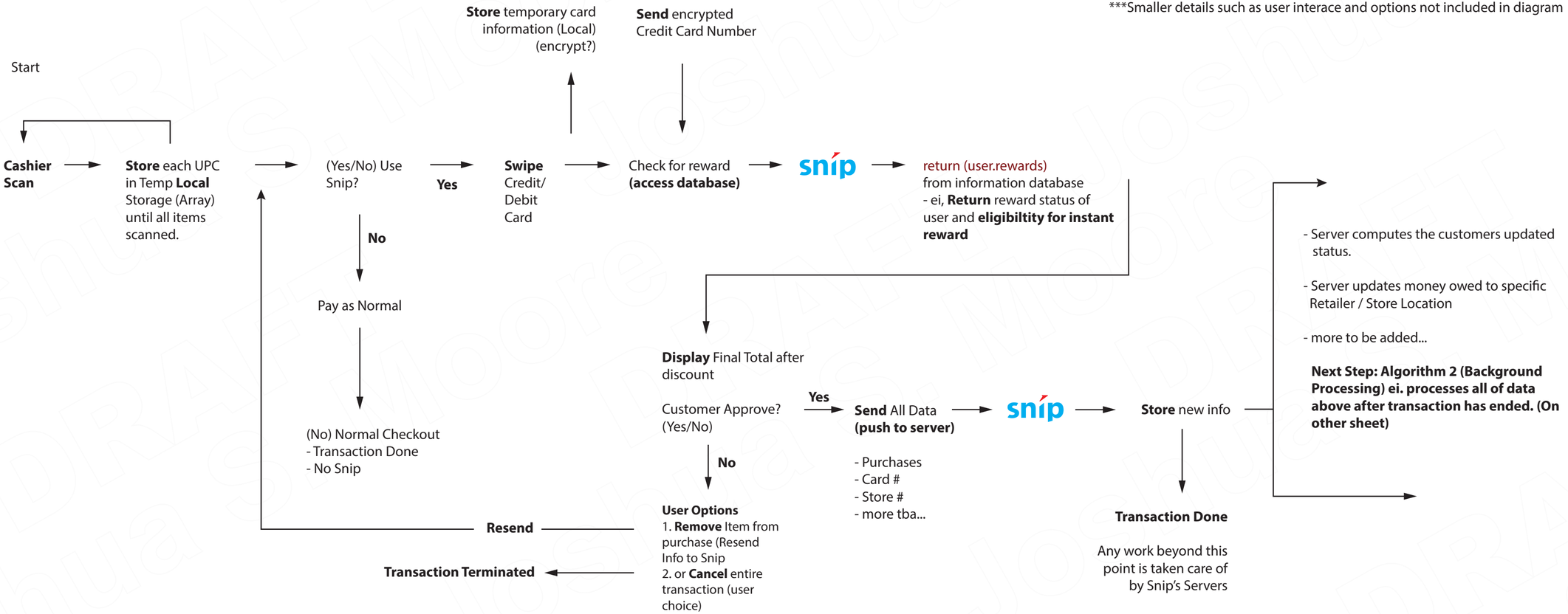


***Smaller details such as user interace and options not included in diagram

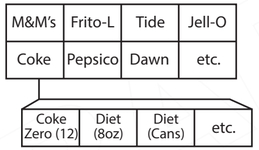
snip

Main General Algorithm | 1

Summary:
Instore Purchase

Joshua Moore
803-521-5411
jsm4@g.clemson.edu

Brands Who Sign Up W/ Snip: How Their Info is Stored | Hash Table Structure



Hash Table With All Brands

Brand can be found by using first section of UPC Code

Hash Table With Products that Brand Sells

Product Hash Table lives with Brand Location

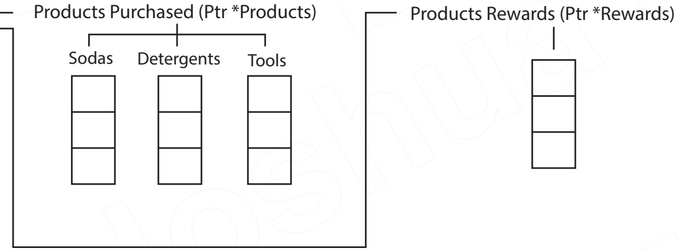
Product can be found by using second section of UPC Code

Maybe keep track of parent companies seperately, ex:

Pepsico = FritoLay + Pepsi + Gatorade + Quaker + ...

User Profile (more to be added/ revised) (Separate Structure)

```
struct UserProfile {
   Ptr *Products;     -- All products bought
   Ptr *Rewards;      -- Commited Brands/ Rewards: STATUS
   int  ID;
   int Password;
   struct Card {  .... };
   .....
   .....
}
```



General Algorithm for Updating User Program Progress. Fine Details and steps in between steps need to be added.

All UPC Codes from Trasaction and User + Store Info

1 01234 45678 2
1 82884 45879 5
8 12577 97618 8
3 07532 42478 1

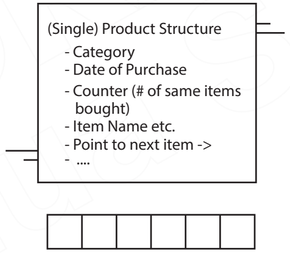
→ Find User Data Location Based on Card# or User ID

→ 1. Only add purchase data to User Profile if user signed up for the Particular Brand Incentive

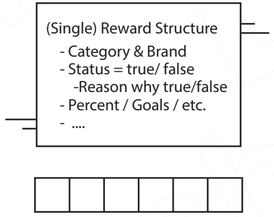
-- or -- (better, but more space needed)

2. Add purchase data regardless.

→ Add Products to User's Purchase Data
- If product exists, just update it's counter.
- Else, Allocate memory for Product structure with it's attributes ex:



→ Update Reward Status Based on Purchases
- If No Conflict Reward; status = true
- If Conflicting Condition; status = false



→ Push update to user's wireless devices

snip

Post Purchase Algorithm | 2

Summary:
What server does after transaction complete.

Updates user incentive status.
Updates user purchases.
etc.

--

How Info Stored

Joshua Moore
803-521-5411
jsm4@g.clemson.edu