

Problem:

Soap, detergent, medicine. There are some things people need to keep on buying consistently. Oftentimes people don't realize that they're about to run out of the item or forget to buy the item before they have to go buy it again.

Objective:

Create a database that stores items you have to buy frequently, when you last bought the item, approximate time you should buy the item next and the locations of where you can buy these items. With these applications, the user can check every now and then to see if you are about to run out of something and where they can go buy the items.

Rules:

Users can register with a unique username and a password

Users can create and manage their inventory of items they purchase

Each store has names and addresses

Each item is categorized between consumable items and household item

Consumable Item and householdItem both have an item name

Consumable Item has a value of days till expiration

Consumable Item has a consumable type

Household Item has a brand name

Household Item has a household item type

Reminders and inventories keeps track of the number of items

Each item is sold a specific store

Items depletes a certain amount each day

Items in inventory has the amount of servings remaining

Reminders can be set for each item you need to purchase again

Reminders include the reminder's name and the days till the next purchase

Users can own multiple inventories

Items can be associated with multiple inventories

Stores can sell multiple items

Inventory manages items

### Nouns:

Users  
Username  
Password  
Items  
Store  
Name  
Address  
Inventory  
Reminder  
Consumable items  
Household items  
Brand name  
Servings

### Verbs;

Register  
Create  
Manage  
Track  
Purchase  
Buy  
Sell  
Own  
Associate  
Depletes

User table:

userID: PK

Name: TEXT

Username: TEXT

Password: TEXT

Inventory table:

inventoryID: PK

userID: FK

inventoryName: TEXT

Item table:

itemID: PK

storeID: FK

inventoryID: FK

itemTypeID: FK

itemName: TEXT

itemsLeft: INT

servingPerDay: INT

Store table:

storeID: PK

Name: TEXT

Address: TEXT

Reminder table:

reminderID: PK

itemID: FK

reminderName: TEXT

daysTillPurchase: INT

ItemType table:

itemTypeID: PK

daysTillExpiration: INT

consumableTypeID: FK

householdItemTypeID: FK

ConsumableType table:

consumableTypeID: PK

consumableType: {medicine, drinks, produce, meatProducts, seafood, snacks, other}

HouseholdItemType table:

productTypeID: PK

productType: {cleaningProducts, hygieneProducts, skinCareProducts, makeupProducts,  
other}

With redis users will be able to...

Create an account

(They will have a account that can be accessed with a unique username and password. It will also be initialized with your name)

hGetAll will be used to make sure that a account has not been created with the same username yet

hSet will be used to help create the user account to set their name, username, and password for their accounts

Update their password

(If you want to change your password, you will be able to update your password by first typing your username, type your old password, and type your new password you want to change to)

hGetAll will be used to find that specific user

(if not found an error will be thrown)

(if password does not match for the account an error will be thrown)

hSet will be used to update the new password

Delete their account

(If you want to delete your account you will need your username and your current password to completely delete the account from redis client)

hGetAll will be used to find the specific user

(if not found an error will be thrown)

(if the passwords do not match an error will be thrown)

del will be used to delete the user data from redis

Get user inventory count

(Users will be able to find your inventory count if you input your username)

hGetAll will be used to find that specific user's inventory count you want to find

Create new inventories

(Users can create new inventories by inputting username and an inventoryName)

sAdd will be used to add a specific inventoryId to a specific collection with the username

hSet will be used to set inventoryName for the inventoryId

Add items to the inventories

(Users can input a item into their inventory by inputting their username, inventoryName, and a list of items)

sPop will be used to check if the inventory already exists for the user

(if it does not it will create a new inventory)

sAdd will be used to add specific items to a specific inventory

### Create new reminders

(Users can create a reminder by inputting username, inventoryName of the item, itemName, and reminderData that you want to input for the reminder)

hSet will be used to set reminderData to reminder

### Delete reminder

(User can delete a reminder by inputting username, inventoryName and the itemName to delete the reminder for that specific item)

hGetAll will be used to find that specific user's inventory

Exists will be used to see if inventory exists

hDel will be used to delete the reminder