Pierre Garcia

BUS591

06/24/2019

Domain dictionary: perpetual inventory control system

Item:

An item is anything that can be legally sold at the store location.

Attribute:

* ID
* Name
* Brand
* Selling cost
* UPC

Methods:

* Add item:
  + We need to be able to add an item
* Archive item:
  + If we don’t want to sell an item in the store anymore we archive it and after 2 years the item will be deleted

Authorization:

An item must be authorized at the location

Attributes:

* License ID

Store location:

Correspond to where the store is located

Attributes:

* Coordinate
* Zip code
* City

Perpetual inventory:

The perpetual inventory class is a singleton class. For security purpose and to eliminate confusion, it is the only place in the system where we can access and modify an item quantity on hand

Attributes:

* Quantity on hand:
* Discrepancy range:
  + the discrepancy range helps figure out to what extend the quantity on hand can be off compared to the quantity counted. It is based on historical data about the item selling data.

Methods:

* Get POS updates:
  + Request the transactions data from the POS so the item quantity on hand can be adjusted. It is set to be requesting every 5 minutes.
* Update QOH
  + Updating QOH can be done in two ways: the first one is when the reconciliation requests to adjust QOH after a count.
  + the second way to adjust QOH is after transaction arrive from the POS.

reconciliation:

it reconciliates the quantity counted and the quantity on hand.

Attributes:

* Quantity on hand
* Quantity counted
* Reconciliated quantity

Methods:

* Get quantity on hand:
  + The system accesses the quantity on hand to be able to compare it with the quantity counted.
* Get quantity counted
  + Quantity counted is input during the count assignment. The reconciliation is launched as the employee input the item count.
* Calculated reconciliation:
  + Reconciliation is calculated by comparing Qcounted and QOH. If the difference the two of them is small. Then the QOH is not updated. However, if the difference in quantity is bigger than the threshold of acceptance, then a recount request is triggered.
* Request recount
  + Triggered when discrepancy between Q counted and QOH is bigger than the threshold.

Count assignment:

An employee requests a count assignment.

Attributes:

* Assignment ID
* Assignment time
* Priority

Methods:

* Create new assignment:
  + after being authentified, the employee can create an employee
* Pause assignment
  + Assignment can be paused. If it is pause for more than 30 minutes, then the assignment is automatically canceled.
* Cancel assignment
* Reschedule assignment
  + A canceled assignment with items of high priority must be rescheduled.

Assignment building

Once the assignment is created, an employee builds the assignment to fit requirements. The requirements include selecting items with highest priority and also the kind of count request that is going to performed.

Attributes:

* Item list
* Assignment type

Methods:

* Edit the assignment
  + the employee builds the assignments and then he reviews to see if it matches the time available. He can then edit the assignment by adding or deleting items.

Count result:

The count results are entered as soon as the employee is done counting the particular item.

Attributes:

* Assignment ID
* Quantity counted

Methods:

* Input results:
  + The results of the count are entered in the system as the employee finishes his cout.
* Update quantity counted
  + If the discrepancy between the quantity counted and an approximative value of what should be there, requires the employee to confirm and update the quantity counted. The quantity counted can be update if the employee miss wrote the item counted.

Counting route:

The counting route is calculated by taking a fixed point in the store and finding the item in the counting list that is the closest to the one that is counted prior to itself.

Attributes:

* Length of route
* Item coordinate

Methods:

* Calculate route:
  + The route is calculated by finding the smallest path to an item in the list.
* Edit route:
  + The route can be edited to fit the time requirement of the employee.

Count location:

The location of count is important for the counting assignment because it allows to make sure that the employee is at the right place, counting the right item in the store.

Attribute

* Aisle ID
* Section ID
* Shelve ID

Count request:

A count request can arise for several reason, it can be requested by the system as a verification count, a cycle count or an exception count.

Attributes:

* Date request time
* Priority

Method:

* New count request
* Update count request

Count Well

The count well keeps track of the items that are in the counting queue and assign them priority depending on the urgency in which they need to be counted.

Attribute:

* Count priority

Method:

* Update priority queue:
  + An item priority can be updated, for instance if it is a recount because an important item quantity on hand is really off.

Employee:

An employee can request a count assignment during the counting time period. The employee must be authorized.

Attributes:

* EmployeeID
* Name

Methods:

* Request\_count\_assignment

Item allocation

* number of facings of front
* number of facing deep
* number of facing height

stock Area

Backroom

Attributes:

* Quantity on hand in the backroom.

Floor:

Attributes:

* Quantity on hand on the floor

Area

Attributes:

* Area ID

Aisle

Attributes

* Area ID

Section

Attributes

* letter
* section ID

Shelve:

Attributes

* shelve number
* side letter
* section ID

Item allocation

Attributes

* number of facings of front
* number of facing deep
* number of facing height

Sellable

Case

Innerpack

Content

Attribute

* ID
* Supplier
* Perishable date

Container:

Attribute:

* Volume
* Value
* Turnover

Item description

Methods:

* Modify description:
  + Allows to modify the description of an item.
* Modify state:
  + Allows to modify the state of an item.

POS description

Attributes:

* POS item transaction ID

Shelve description

Attribute:

* Shelve tag

-