



*Preparing People to Lead Extraordinary Lives*

**LOYOLA  
UNIVERSITY  
CHICAGO**

DEPARTMENT OF  
*Computer Science*

Water Tower Campus, Lewis Towers 416  
820 N. Michigan Avenue  
Chicago, IL 60611 USA

## **Retail Check Out and Return System (Retail Return)**

### Functional Needs Statement

The system automates operations for a retail store check out and return counter. Items to be purchased are scanned by a clerk one at a time. Prices are calculated automatically from the product code scanned. When more than one of the same item is purchased the system allows the clerk to first enter the quantity then scan an item once (the default quantity is one). As each item is scanned the system shows the price of that item (or the price of each and every one of the items if quantity is other than one) and the current total for the entire purchase transaction.

Products are identified by a product code (imagine a SKU scan bar code). Each product is in a product category.

A sales receipt is generated at the end of the transaction with all purchases grouped into categories and giving the date and time of the transaction. The sales receipt shows the actual price for each item purchased. The sales receipt has a scan code as well that is used to automate returns.

Prices change often. Purchasers get the current price in the system. The system keeps track of the prices actually paid when the product was purchased. When an item is returned, the refund is for the amount actually paid, not the current price.

The manager can add or remove products and categories at any time. Products removed can still be returned but can no longer be purchased. The manager can adjust prices at any time; price changes are immediate and the new price is used for any purchase transactions started after the price change. For simplicity, do not address transactions in progress during price changes.

The system supports several kinds of pricing schemes. Unit pricing gives the price of a single product. Multiple pricing allows pricing such as "\$.50 each, 3 for \$1.25" and "Bogo". Hence, price calculations are "sophisticated" and require careful systems thinking. Not all items need to be scanned together to get the special pricing options; they can be scanned any time during the purchase transaction. Pricing schemes can apply to multiple products.

When an item is returned the original receipt and the item being returned are scanned. Quantities greater than one may be returned similarly to purchasing. The correct refund is

automatically generated. The refund price is the price actually paid for that item(s) not the current price of the item. A receipt (with a scan code) is also printed for the return showing all items returned, the amount for each item, and the total of the refund. Return items due a refund are shown as negative values on the receipt. A return receipt total that is negative is a refund to the customer. Multiple return transactions can be made from a single sales receipt (e.g. return one item today and another item tomorrow). When processing a return, the system updates the information stored from the original sale to show what has been returned and what is still purchased (and may be returned in the future).

For simplicity assume only a single type of item is returned in a transaction (in any quantity). If you want to return both shoes and a hat, that is two transactions and two possible refunds. Return processing must take into account the affect of the special pricing schemes (if 3 were purchased at “\$0.50 each, 3 for \$1.25” the first item returned yields a \$0.25 refund plus sales tax if applicable; returning the first of a Bogoy yields nothing). Items returned are handled as “last purchased, first returned” even across multiple return transactions. The ordering of items will be as on the original receipt. Use negative numbers for funds due back to the customer; positive numbers for what the customer needs to pay.

Attempts to return items that were not purchased are detected by the system. For example, if a customer attempts to fraudulently return quantity 4 of an item but only purchased 3, the return is rejected. Likewise, if a customer successfully returns an item one day and later attempts to return the same item again, the return is identified as fraudulent (assuming original sale was for quantity one).

Some products are subject to sales tax and others are not. The system calculates sales tax at the end of purchase and return transactions and displays (and puts on the receipt) the subtotal of taxable and untaxed items, the sales tax calculated, and the grand total of the receipt including any tax. Assume the sale tax rate is 5% for items subject to tax.

Once a day the system generates a report giving the total sales value, total number of items purchased, total return value, total number of items returned, and total number of purchase and return transactions that day. The report also includes the number of any attempted fraudulent returns that were detected. The report includes only transactions that have been completed when the report is run (sales or returns in progress are ignored and will be in the next report).

For simplicity, assume that all purchases are cash. There is no need to consider different payment forms such as credit cards, debit cards, or gift cards.

For simplicity, do not address managers and clerks, their ids, how they log into the system, etc. Assume anyone at the running system can perform all the transactions.

RetailReturnFNS

Version 4 (eliminating category pricing, eliminating cancel an item during purchase)

Version 3 (derived from earlier RetailReceiptSystems with extensive updates and changes)