Clothing Store Point of Sale System

Tiffany Truc
Antonio Ambriz
Danny Kim

CS-250-1002

System Description

The Clothing Store Point of Sale System allows companies to organize store inventory alongside the customer transaction history. The point of sale system is designed to make inventory management more efficient by automatically updating the item inventory given purchases and returns. The item inventory system is searchable given an item's unique ID, price, size, color, and quantity. By working with external payment processors, employees are able to accept debit, credit, and cash payments with purchase totals being automatically calculated. Transaction history and sales numbers are stored separately in a secure cloud-based database which can only be accessed by administrative employees. This system is also supported by iOS and Android operating systems, and utilizes phones, tablets, and barcode scanners to connect to the shared item inventory.

Software Architecture Overview

Architectural Diagram Description

Clothing Store POS System

User Interface – The design implemented to make the interface intuitive and simple to use.

Credential System – User login system that determines if the user is an administrative user or regular user.

USE CASE – A bundle of features implemented into the POS application that all connect to one another. In this the Purchase/Return, Employee Tracking, Inventory Management, and General Transaction Report systems will all communicate with each other in some way.

- **Purchase/Return System** – Implementation for the process of purchasing/returning clothing store products from/to the store.

- **Employee Tracking System** – A system that will allow employees to clock in/clock out and keep track of their general actions on the POS system.

- **Inventory Management System** – Implementation of managing inventory in various manners ranging from unique ID to the variables that consists of clothing item or other purchasable items from the clothing store. It allows the user to add/remove or customize various inventory items.

- **General Transaction Report System** – A log that records all transactions that were made within the system.

External Device Compatibility System – An abstract block of compatibility features required to be able to allow access to the POS system by various external hardware.

- Barcode Scanner
- Receipt Printer
- Keyboard and Mouse
- Tablet/PC
- Payment System

Point of Sale API – A storage of data that holds all manner of information regarding the product.

Credit/Debit API – A storage of all data involving payment methods such as credit cards, debit cards, and the name of the holders.

Security System – A system made to combat hacking of store information regarding their products and payment information.

Cloud Service Server/BackUp – A cloud implementation that automatically updates and backs up data involving anything that happens in the POS system.

Real Time Analytics System – The part of the POS system that calculates any mathematical functions instantly. It will calculate net costs for purchases which includes tax and update inventory after every purchase/returns. Another example would be the ability to calculate the total time an employee has worked in that week at the clothing store.

POS Application Patch/Update System – The system that allows for the POS application to be updated through a series of patches after the initial release into the market.

UML Diagram and Descriptions

The Item class deals with in store items and their associated characteristics, the class itself has eight public attributes. These attributes include idNumber, itemName, price, quantity, size, color, dateAdded, and itemType. The attributes itemName, size, color, and itemType are strings. The idNumber, quantity, and dateAdded are ints, and the price is a float. These attributes help categorize the items in the Clothing Point of Sale System, and are connected to the inventory, and payment classes. The employee and administrator classes are connected to this class, as employees and administrators can edit, add, and remove items. The operations in the item class include buyItem(), returnItem(), addItem(), and removeItem(). These operations all return as void.

The Administrative User class deals with management and employees granted administrative privileges. This class has four private attributes: name, adminID, password, and email. Password, name and email is stored as a string, and adminID is stored as an int. The Administrative User class has three operations: viewTransactions(), viewSales(), and viewItems(). These operations all return void, and allow administrative users to access the transaction history, sales numbers, and item inventory databases. Only administrative users are allowed the permission to access these databases.

The Employee class deals with employee login information, and the ability to add, remove, and edit item information. The Employee class has four private attributes: name, staffID, password and email. The attributes name, password, and email are strings, and the staffID is stored as an int. The operations of the Employee class include: viewItem(), searchItem(), addItem(), removeItem(), assignAttributes(), clockIn(), and clockOut(). These operations are used to check in and check out employees, view items, and assign item characteristics, edit items, and add and remove items. These operations all return as void.

The Payment class deals with customer transactions, and recording the payment, as well as updating the inventory, and automatically getting the total for the transaction. The Payment class has five private attributes: amount, saleDate, type, saleTime, and transactionID. The attributes saleDate, saleTime, and transactionID are ints, while type is a string, and amount is a float. The operations of the Payment class include updateInventory(), getTotal(), updateSales(), updateTransactionHistory(), and readBarCode(int idNumber). All of the operations return as void, and readBarCode takes in the item idNumber to input into the system and add to the customer's transaction. The operations automatically update the item inventory, sales numbers, and transaction history after every customer transaction. Returns are included under customer transactions. Furthermore, the operations automatically get the total of the transaction, including sales tax. The readBarCode operation receives the item's idNumber as a parameter, and returns void.

The Transaction History class has records detailing each customer transaction given their payment type, store location, cashier information, name, sale time and date, and transaction ID. eight private attributes. The attributes are the name, saleDate, saleTime, customerPaymentType, customerCardInfo, staffID, storeID, and transactionID. The attributes saleDate, saleTime, staffID, storeID, transactionID, and customerCardInfo are all ints. The name and customerPaymentType are strings. The operations include getTransaction(int transactionID) and returnItem(int idNumber), which both return as void. The operations receive transaction information given the transaction ID, and as the operation to return an item.

The Sales Number class records the sales per store given the transaction history of the store, and is accessible by administrative users. The class contains three public attributes, and three private attributes. The public attributes include quantity, idNumber, and itemName, and the private attributes include saleDate, saleTime, and storeID. All of the attributes are ints except for itemName, which is a string. There is one operation: getSales(). This operation returns as void, but retrieves the sale analytics.

The Inventory class deals with the inventory of items per store location, and the characteristics of items, alongside operations making the inventory system more efficient, organized, and searchable for employees and management. The public attributes include itemNumber, quantity, itemName, itemAttributes, dateAdded, and itemType. The private attribute is the storeID. The itemNumber, quantity, storeID, and dateAdded are all ints. The itemType, itemAttributes and itemName are strings. The operations include searchInventory(), updateInventory(), removeItem(), and addItem(). These operations all return as void. The operations allow employees and management to search the inventory, update the item inventory, and remove and add items to the inventory.

There is a Cash class, which is utilized in the Payment class. There are no attributes, and there are no operations for this class.

The Credit Card class stores customer information, specifically their credit card information. There are five private attributes to this class: name, creditNumber, expiryDate, cvcCode, and cardType. The name and cardType are stored as strings, with the creditNumber, expiryDate and cvcCode stored as ints. There are no operations in this class.

The Debit Card class stores the customer debit card information when used in a transaction. There are five private attributes to this class: name, debitNumber, expiryDate, cvcCode, and cardType. The name and cardType are stored as strings, with the debitNumber, expiryDate and cvcCode stored as ints. There are no operations in this class.