Point of Sale System

By: Matthew Smith, Nicholas Stark, Alicia Loya

3/10/2023

1. Brief Overview

The system being designed is a point of sale system used to keep track of inventory that enters and exits both the warehouse and the clothing store. The system will include a user database, a storage database, and a transaction history database. The user database will include the login information of all current employees which will enable them to access the system. The employee login will also have the subclass of administrator if the employee has been logged as an administrator in the database. The administrator users will be able to add or remove employee login information from the login database. The storage database will store the information of all products within the clothing store. An item class will be used to identify the product and its attributes, including but not limited to price, size, and quantity. Functions with user input will be used for searching for available products and updating the database by changing the quantity of the item through sale and cash only refunds. Employees with the administrator class will be able to change both the quantity and price of an item, as well as enter a new item into the system with a unique barcode identification number. The transaction database will store the transaction history for previous purchases. If the system is not being used for refunds, the transaction history will only be accessed by the employees with the administrator class.

2. Software Architecture Overview

Software Architecture Diagram

Description Software Architecture

UML Class Diagram

Description of UML Diagram

Classes

Employee/Admin

The most important class for the user interface of this program is the employee class, along with the admin employee subclass. In order for the user to access the functions of this class, the user will have to utilize the login function, which will take an input string for both the username and password, if the username and password matches, a boolean value of login status will be updated to true, allowing the employee/admin to utilize the classes methods and store the String username of the logged in class. After logging in, the employee will be able to use the employee functions which will only work if the loginStatus variable is true. The class will also store the int numItems for the number of checkout items and an array of Items called itemList which stores items that will be sold at checkout. Here is a list of functions that the employee class will be able to utilize. All functions that update inventory information will also add a record of each transaction to a database that stores all transactions.

Employee

Void createItem(string ID, int quantity, double price) - will create a new instance of the Item class, with the ID set to the input ID that is present on the new item, the specified quantity, and the selected price. If an item of existing ID is found, it will update the quantity and price of the existing item, information for item attributes will be updated for the new instance of the item class by checking using the Item's ID in an external database to retrieve the information about the item with the matching ID. The new item that is created will be stored in the item Inventory.

void updateQuantity(Item item, int amount) - will update the item quantity stored in the inventory with either a positive or negative amount, however the value cannot be changed to negative, instead displaying zero.

void updatePrice(string ID, double price) - this function will update the item with the matching ID to the new price entered.

bool itemPresent(string ID) - this function will return true or false if the item with the matching ID is present in the inventory.

void searchItem(string ID,double price, string size, string color, int quantity)
This function will search for Items in the inventory with the matching characteristics, if a characteristic is left empty, the list of items will not be filtered according to the empty parameter, allowing for precise filtering.

Void addToCheckout(string ID) - adds the item selected by ID to itemList[], and updates the int numItems by 1. The ID is either added manually by the employee or entered by utilizing the mobile devices camera in order to scan the item's barcode.

void checkout() - this function will take all items that have been added to the employee's itemList[] and calculate the price of all the items combined. From there it will access an external online payment system that will take the form of payment, if the payment is accepted, all items in itemList[] will be removed from the Item's quantity stored in the database and the numItems will be set to 0 and the array itemList[] will be cleared of all items.

void refund(string ID) - this function will take the ID of the item that is scanned or entered and will increase the inventory quantity by 1, similar to other transaction functions, it will update the transaction history for record.

void logout() - this function will log the employee out of the app, set the loginStatus to false, clear the username,remove all items from itemList and setnumItems to 0.

Admin subclass

void displayTransactions() - this admin exclusive function will access the transaction history database and display it to the admin user.

Item

The item class is used to store every unique item based on its unique barcode. Each item class is stored in a database filled with every item that exists. Items will generally be accessed by the item ID when they are searched for in operation. Information, such as color, size, ID will not be changed once initialized,as is constant according to the unique barcode initialized according to an external catalog database. However, the item's quantity and price can be changed by the employee's functions. Below are the Item classes existing functions.

string ID() - this function returns the items ID. This function is useful for comparing the input search ID's of the employee function for finding the item in the database.

string price() - returns the Items price

string color() - returns the Items color

string size() - returns the Items size

double price() - returns the Items price

void displayInfo() - display all info pertaining the specific Item, such as the ID, color, size, and price.

void changePrice(double price) - changes items price, this function is used by employee functions that change the price

void changeQuantity(int change) - changes items quantity adding or subtracting by the input change, this function is used by employee functions that change the quantity

Databases

Below is a list of databases that this program will access in order to store information that is used by the aforementioned classes.

Employee Logins

This database is used to store all employee information in the format of a hashmap, storing the username and the associated password for logging in.

Inventory

This database stores all of the existing Item classes as a dynamic list of classes, when an employee function is used to access this databse, it will search the elements of the database for the matching item ID in order to operate on that value.

Item Catalog

This database stores the relevant Item information that is needed for initializing a new Item into the inventory. This database is only accessed when creating a new Item in the inventory as it contains external information from each Item's manufacturer, such as the color, size, and ID. This database is important as it retrieves existing information that is used throughout all stores that sell the same items. As a result, certain information that is used to create an item into our inventory cannot be altered in order to prevent misinformation about our products as our store is a distributor.

Transaction History

This database stores all transactions that are recorded when they happen as a result of employee functions. All transactions are stored as digital receipts that are then uploaded into the transaction database. This information can only be accessed by the store administrators when using the displayTransactions function in order to keep customer information confidential. These transactions include the date and time of the transactions, the form of payment used, the checkout price total along with the items that were purchased for each transaction. This information is required in order to keep track of store earnings and keep record in ase of customer disputes. In order to make sure that the database does not run out of space, each transaction will be removed from the database automatically after 3 years.

External Systems

For external systems, the most important system will be an external payment service that is accessed when the checkout function is used. This system will be required in order to process payments and return whether or not the payment has been accepted. If the payment has been declined, the selected items will not be purchased. In addition, this external system will be able to approve cash payments, contactless payments and credit or debit cards. The external system will be the square mobile reader, which is compatible with ipads which is the device that this program is intended to function on.

3. Development plan and timeline

Partition of Tasks

(task partition here)

Team Member Responsibilities

This project will consist of multiple teams for development,