The System Design Document for Setting Up a Self-Service Checkout System

1 Introduction

1.1 Background

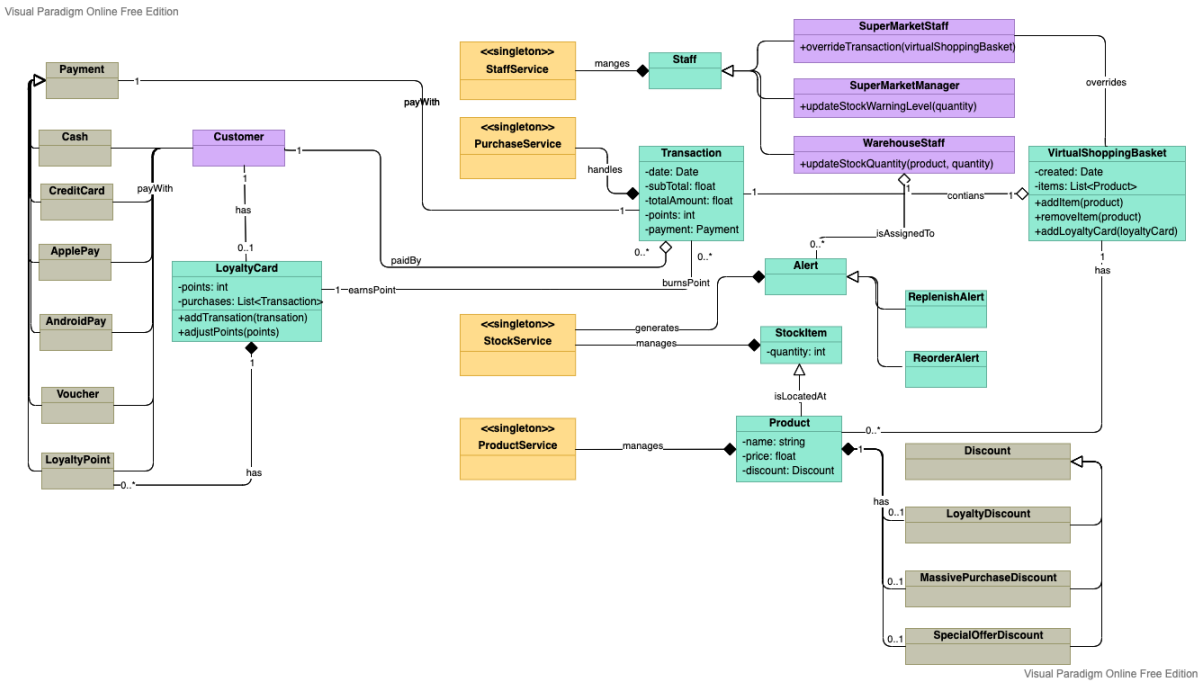
We recently received a request from a well-known high street supermarket that asked us to develop a self-service checkout system for them. The self-service checkout system allows customers to add goods to the virtual shopping basket by scanning the barcode or weighting goods with an integrated scale by themselves. It also allows customers to record the purchase and earn points to their loyalty card. In addition, the system supports multiple ways to finish the purchase during the checkout process, such as cash, credit card, and vouchers.

1.2 Purpose

The purpose of this document aims for providing the overview of the design for the self-service checkout system.

2 The Object Class Design

2.1 The Class Diagram



2.2 The List of Classes

Name	Description
Staff	It is the generalized staff class definition. This class contains the general information for different staff types.
SuperMarketStaff	It is the supermarket staff class definition. This class inherits from the generalized staff class. It also can override the transaction from the virtual shopping basket instance.
SuperMarketManager	It is the supermarket manager class definition. This class inherits from the generalized staff class. It also can update the warning level of stock items.
WarehouseStaff	It is the warehouse staff class definition. This class inherits from the generalized staff class. It also can update the number of stock items.
StaffService	It is the singleton staff service definition. This service can manage all staff objects.
VirtualShoppingBasket	It is the virtual shopping basket class definition. It is the core class definition in the self-service checkout system. The system mainly manages the life-cycle of the virtual shopping basket instance.
Customer	The customer class definition. This class contains the general information of customers.
LoyaltyCard	It is the loyalty card class definition. It contains the information of customers' loyalty cards. Such as available points and purchases. It can adjust points inside the loyalty card and add a transaction to the purchase history.
Payment	It is the payment class definition. It is an abstract class which means that the system cannot initialize the class independently.
Cash	It is the cash class definition. It is one of the payment methods used while making the transaction. It inherits from the generalized payment class.
CreditCard	It is the credit card class definition. It is one of the payment methods used while making the transaction. It inherits from the generalized payment class.
ApplePay	It is the Apple Pay class definition. It is one of the payment methods used while making the transaction. It inherits from the generalized payment class.
AndroidPay	It is the Android pay class definition. It is one of the payment methods used while making the transaction. It inherits from the generalized payment class.
Voucher	It is the voucher class definition. It is one of the payment methods used while making the transaction. It inherits from the generalized payment class.
LoyaltyPoint	It is the loyalty point class definition. It is one of the payment methods used while making the transaction. It inherits from the generalized payment class.

Transaction	It is the transaction class definition. When customers complete the payment, the system creates an instance of the transaction class. It contains the transaction information, such as date, sub-total, and total amount.
PurchaseService	The singleton purchase service. It handles the transaction process, such as adjusting the loyalty card points and notifying the stock control system.
Product	It is the product class definition. It contains information about the product, such as name and price. It inherits from the stock item class.
ProductService	The singleton product service. It manages product instances.
StockItem	The stock item class definition. It contains the quantity information of a product.
StockService	The singleton stock service. It generates alerts to warehouse staff when the stock level is in the warning zone. This service creates an alert and assigns it to warehouse staff to follow up.
Alert	The alert class definition. It is an abstract class.
ReplenishAlert	The replenish alert class definition. It inherits from the alert class. It contains the product that needs to be replenished by warehouse staff.
ReorderAlert	The reorder alert class definition. It inherits from the alert class. It contains the product that needs to be reordered by warehouse staff.
Discount	The discount class definition. It is an abstract class.
LoyaltyDiscount	The loyalty discount class definition. It inherits from the discount class.
MassivePurchaseDiscount	The massive purchase discount. It inherits from the discount class.
SpecialOfferDiscount	The special offer discount. It inherits from the discount class.

2.3 The Relationships Among Classes

Relationship Name	Description
StaffService manages staffs	This relationship indicates that the staff service is responsible for managing the CRUD-related operations for the staff object.
SuperMarketStaff overrides VirtualShoppingBaskets	This relationship represents that supermarket staff can override the transaction in the virtual shopping basket. For example, the customer purchasing a restricted item has not fulfilled the requirement of purchasing the restricted item. The supermarket staff is responsible for cancelling this item from the virtual shopping basket.
VirtualShoppingBasket has a product	This relationship represents that the virtual shopping basket can have zero or many products during its lifetime.
PurchaseService handles transactions	This relationship indicates that the purchase service is responsible for handling the transaction made by the customer. It handles the whole transaction process with the payment gateway.
VirtualShoppingBasket contains a transaction	This relationship represents that the virtual shopping basket contains a transaction. When the transaction is completed, the basket gets the transaction details.
Transaction paid by a payment method	This relationship represents that the transaction contains a sub-type of the payment method abstract class and the details of the payment method.
ProductService manages products	This relationship indicates that the product service manages all CRUD-related operations. Also, when the customer is ready to check out, the purchase service asks the product service to get all available discounts for each product.
Customer pays with a type of the payment method	This relationship represents that customers can complete the transaction with different payment methods. Such as cash, mobile payment, and loyalty points.
Customer has a loyalty card	This relationship represents that customers can apply and own a loyalty card to persist loyalty points for each purchase. However, each customer can only have one loyalty card for persisting/burning loyalty points while completing the transaction process.
Loyalty card has loyalty points	This relationship represents that system can store/burn loyalty points of the loyalty card while completing the transaction.
Product has a discount	This relationship represents that each product has multiple discounts simultaneously, such as exclusive discounts for loyal customers and massive purchase discounts for customers purchasing a large volume of a product in a single transaction.

Product is located at stock item	This relationship represents that the product instance is a sub-type of the stock item class, and the product instance contains the information of the stock item.
StockService generates alerts	This relationship indicates that the stock service generates alerts to warehouse staff. For example, when a product's stock level is in the warning level, the stock service will generate a replenish or reorder alert to the dedicated warehouse staff for further follow-up.
StockService manges stock items	This relationship indicates that the stock service also has a responsibility for managing the CRUD-related operations for the stock item.
Alert is assigned to warehouse staffs	This relationship indicates that the system generates the alert and assigns it to dedicated warehouse staff. When the warehouse staff receives an alert from the stock service, the warehouse staff has obligations to follow up the alert.

2.4 Attributes And Operations Among Classes

2.4.1 The Product Class

The product class has three main attributes: the product class's name, price, and discounts attributes. The product class contains the data of the product. The self-service checkout system reads its information when dealing with CRUD operations of product instances in the virtual shopping basket. The system shows the name and price of the product on the screen of the self-checkout machine. The system also uses the price and the list of discount attributes to calculate the product's final price while checking out. For example, the product may contain multiple discounts simultaneously, such as the significant volume purchase discount, loyalty member exclusive discount, and special offer discount. The system checks the condition of each discount instance. Once the transaction matches the discount requirement, the system will apply the offer to the transaction. The final price will be re-calculated instantly.

2.4.2 The Virtual Shopping Basket Class

The virtual shopping basket class has two attributes and three operations. First, it has the created date and the list of the product attributes. The created date is the virtual shopping basket object initialized date. The system uses this attribute to record the purchase date during the transaction process. The product attribute list is the container of product items that the customer wants. The system also uses this attribute to calculate the final price, apply discounts to the product, and estimate loyalty points from the transaction. It also has three primary operations for this class. It provides the adding product item, removing it, and adding loyalty card operations to the system. The add product operation allows the customer to scan the product with the barcode scanner or weigh the product with the integrated scale to add the product into the virtual shopping basket. The remove product operation allows the customer to cancel the added item from the virtual shopping basket. Finally, the add loyalty card operation allows the customer to scan the loyalty card before checking out.

2.4.3 The Transaction Class

The transaction class has five attributes. It contains the date, sub-total, total amount, points, and payment attributes. The date attribute is the date of the transaction, and the sub-total attribute is the amount of the product's price multiplied by total units. The offer of the qualified discount will be applied while calculating the sub-total of the product. The total amount attribute is the sum of all sub-total of products which is the final amount that the customer has to pay. Finally, the payment attribute is the customer's payment method used for the transaction.

2.4.4 The Staff Class

The staff class is a general class of the supermarket. It contains general information for the staff. It has three children classes in the system: the SuperMarketStaff, the SuperMarketManager, and the WarehouseStaff classes inherit from the Staff class. All classes represent different roles in the system. The SuperMarketStaff class represents the staff of the supermarket. This type of staff can override the virtual shopping basket instance transaction. The SuperMarketManager class represents the manager of the supermarket. This type of staff can update the warning level of a particular product. For example, when the stock level of a particular product is running low, the stock service will alert warehouse staff to handle it. The WarehouseStaff class is presenting the staff of the warehouse. This type of staff can update the stock quantity of a product to the system by replenishing or reordering the products.

2.4.5 The Alert Class

The Alert class is a general class of the system. It has two child classes in the system. The ReplenishAlert and the ReorderAlert classes inherit from the Alert class and can be assigned to dedicated warehouse staff to follow up.

2.4.6 The Loyalty Card Class

The Loyalty Card class represents the physical card that customers may have. This class has two attributes: loyalty points and the history of purchases, and two methods: adding a transaction and adjusting loyalty points. The system uses the attribute of loyalty points to store the currently available loyalty points of the card. Customers can use it to pay the amount of the transaction. It also can be gained from the transaction. The list of the history of purchases attribute is the container that contains all purchases. It also shows the point gained from each purchase specifically. The loyalty card class provides two primary operations. First, it provides an operation to add a transaction into the purchase history. It also provides an operation to adjust the points information of the loyalty card instance. Such as customers using loyalty points to pay the transaction amount and gaining points from the transaction.