Mid-Module Assignment: System Design

1 Classes

Class	Reason for Inclusion
Item	Object the Customer buys and the Supermarket sells
Person	Superclass for Customers and Employees, since both have common attributes
Customer	Subclass of Person with unique attributes for Customers
Employee	Subclass of Person with attributes that all employees have
Supermarket Staff	Subclass of Employee that interacts with Virtual Shopping Basket
Warehouse Staff	Subclass of Employee that interacts with Stock Control System
Self Service Checkout	Combines Virtual Shopping Basket, Barcode Reader, Integrated Scale
Virtual Shopping Basket	Interface between Customers, Barcode Scanner, Integrated Scale, Stock Control System
Barcode Reader	Scans Item for input into the Virtual Shopping Basket
Integrated Scale	Sends Item weights to the Virtual Shopping Basket
Loyalty Scheme	Stores information about member Customers, scheme discounts
Stock Control System	Interface with Virtual Shopping Basket, Warehouse Staff, Items
Alert	Eliminates redundancy between types of alerts
Payment	Enables updates to payment methods

2 Relationships

Relationship Type	Reason Chosen
Inheritance	Customer is a type of Person
Inheritance	Employee is a type of Person
Inheritance	Warehouse Staff is a type of Employee
Inheritance	Supermarket Staff is a type of Employee
Composition	Self Service Checkout is composed of a Virtual Shopping Basket, Barcode Reader, Integrated Scale; Checkout does not exist without components
Aggregation	Customer has many Virtual Shopping Baskets, but Virtual Shopping Baskets exist without Customer
Aggregation	Loyalty Scheme has many Customers, but Customers exist without Loyalty Scheme
Aggregation	Stock Control System has many Items, but Items exist without Stock Control System
Aggregation	Stock Control System has many Virtual Shopping Baskets, but Virtual Shopping Baskets exist without Stock Control System
Association	Loyalty Scheme and Virtual Shopping Basket exchange data
Association	Payment and Virtual Shopping Basket exchange data
Directed Association	Supermarket Staff updates Virtual Shopping Basket

Relationship Type	Reason Chosen
Directed Association	Barcode Reader sends data to Virtual Shopping Basket
Directed Association	Integrated Scale sends data to Virtual Shopping Basket
Directed Association	Warehouse Staff updates Stock Control System
Directed Association	Stock Control System triggers Alert
Directed Association	Alert sends data to Warehouse Staff

3 Class Diagram

The class diagram on page 4 uses UML's Specification perspective, showing detail whilst avoiding language-specific design; this is commonly accepted as the most robust perspective to use for object-oriented programming (Fowler & Scott, 2000).

4 Rationale for System Design

This system design is intended to minimize code redundancy and to enable future changes, such as implementing different loyalty schemes, upgrading checkout devices, and customising alerts (Costal et al, 2002; Larman, 1998). The system is organised into three primary groups: people, purchasing activities in the supermarket, and stock activities in the warehouse. The Virtual Shopping Basket and the Stock Control System are the heart of the design, synchronising information about the items being purchased and ultimately driving actions to both customers and employees.

5 Key Choices Made

1. Defining a Self Service Checkout class composed of the Virtual Shopping Basket, Barcode Reader, and Integrated Scale allows the supermarket to upgrade any of the components with minimum cost and technological impacts.

2. Establishing Loyalty Scheme as a class enables modular updates to the loyalty scheme or implementing multiple loyalty schemes as supermarkets change.

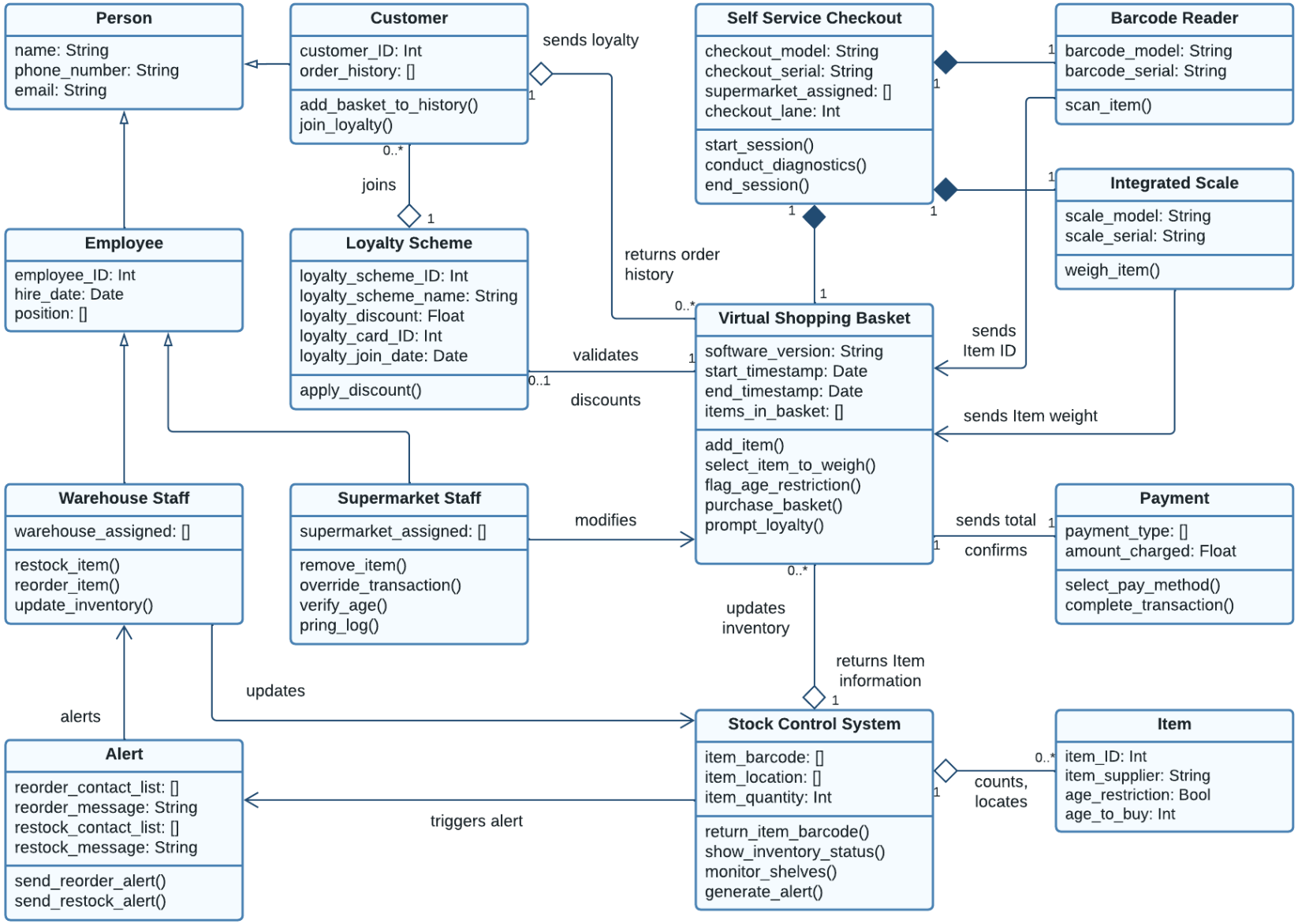
3. Defining Alert as a class allows for customisation for future requirements and reduces redundancy between methods.

6 References

Costal, D., Sancho, M.-R. and Teniente, E. (2002). *Understanding Redundancy in UML Models for Object-Oriented Analysis. Notes on Numerical Fluid Mechanics and Multidisciplinary Design*, pp.659–674. Available from: https://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.12.1137&rep=rep1&type=pdf [Accessed 21 April 2022].

Fowler, M. and Scott, K. (2000). *UML distilled (2nd ed.): A Brief Guide to the Standard Object Modeling Language*. Addison-Wesley Longman Publishing Co., Inc., USA. Available from: https://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.183.2984&rep=rep1&type=pdf [Accessed 05 April 2022].

Larman, C. (1998). *Applying UML and Patterns*. Prentice Hall. Available from: https://personal.utdallas.edu/~chung/SP/applying-uml-and-patterns.pdf [Accessed 21 April 2022].


Person
name: String
phone_number: String
email: String
Customer
customer_ID: Int
order_history: []
add_basket_to_history()
join_loyalty()
sends loyalty
Self Service Checkout
checkout_model: String
checkout_serial: String
supermarket_assigned: []
checkout_lane: Int
start_session()
conduct_diagnostics()
end_session()
Barcode Reader
barcode_model: String
barcode_serial: String
scan_item()
joins
Integrated Scale
scale_model: String
scale_serial: String
weigh_item()
Employee
employee_ID: Int
hire_date: Date
position: []
Loyalty Scheme
loyalty_scheme_ID: Int
loyalty_scheme_name: String
loyalty_discount: Float
loyalty_card_ID: Int
loyalty_join_date: Date
apply_discount()
returns order history
Virtual Shopping Basket
software_version: String
start_timestamp: Date
end_timestamp: Date
items_in_basket: []
add_item()
select_item_to_weigh()
flag_age_restriction()
purchase_basket()
prompt_loyalty()
validates
discounts
sends Item ID
sends Item weight
Warehouse Staff
warehouse_assigned: []
restock_item()
reorder_item()
update_inventory()
Supermarket Staff
supermarket_assigned: []
remove_item()
override_transaction()
verify_age()
pring_log()
modifies
Payment
payment_type: []
amount_charged: Float
select_pay_method()
complete_transaction()
sends total
confirms
updates inventory
returns Item information
updates
alerts
Stock Control System
item_barcode: []
item_location: []
item_quantity: Int
return_item_barcode()
show_inventory_status()
monitor_shelves()
generate_alert()
Item
item_ID: Int
item_supplier: String
age_restriction: Bool
age_to_buy: Int
counts, locates
Alert
reorder_contact_list: []
reorder_message: String
restock_contact_list: []
restock_message: String
send_reorder_alert()
send_restock_alert()
triggers alert