



UTM

UNIVERSITI TEKNOLOGI MALAYSIA

DATA STRUCTURE - SCSJ 2013

MINI PROJECT PROPOSAL

TITLE : BOOK STORE MANAGEMENT SYSTEM

GROUP MEMBERS : SOH JIA JUN
TAN SEE JOU
TAN SHI XUAN

SECTION : 09

LECTURER : DR RUHAIDAH SAMSUDIN

INTRODUCTION

Topic for our Data Structure mini project is about Book-Shop management system. The objective of this system is to allow customers to search any book they desired by insert the author's name or book title and buy. Other than that, book shop staff can use this system to manage the stock inventory, sales for any book and give discount to certain book. This system can also generate report for the sales and inventory which make the process of stock checking become convenient and enable the owner to check his/her business from time to time.

Therefore, we decided to implement sorting, searching, linked list and queue concept in this mini project. We decided to apply linked list concept in storing stock inventory. This is because linked list has no restriction which enable the insertion, deletion and restriction data operation can be done in any parts of the list. The book list has been sorted alphabetically according to the book's title and if after that there is any new book added, it will also be arranging in order. Thus, by using linked list, there is no restriction to store the new book, the new book can be stored in any parts of the list according to the title. Besides, by implement searching concept, customer can easily know the availability of the book they desired quickly and convenient by just inserting the author's name or book's title. Searching concept is then compare the customer input with the stock to check the availability. Finally, we decided to implement queue concept in customer buying process. Once the customer had made payment, the sale record is stored in queue concept, which is first in first out (FIFO). The latest sale record will at the top of the list.

USE CASE DIAGRAM: UNIFIED MODELING LANGUAGE (UML)

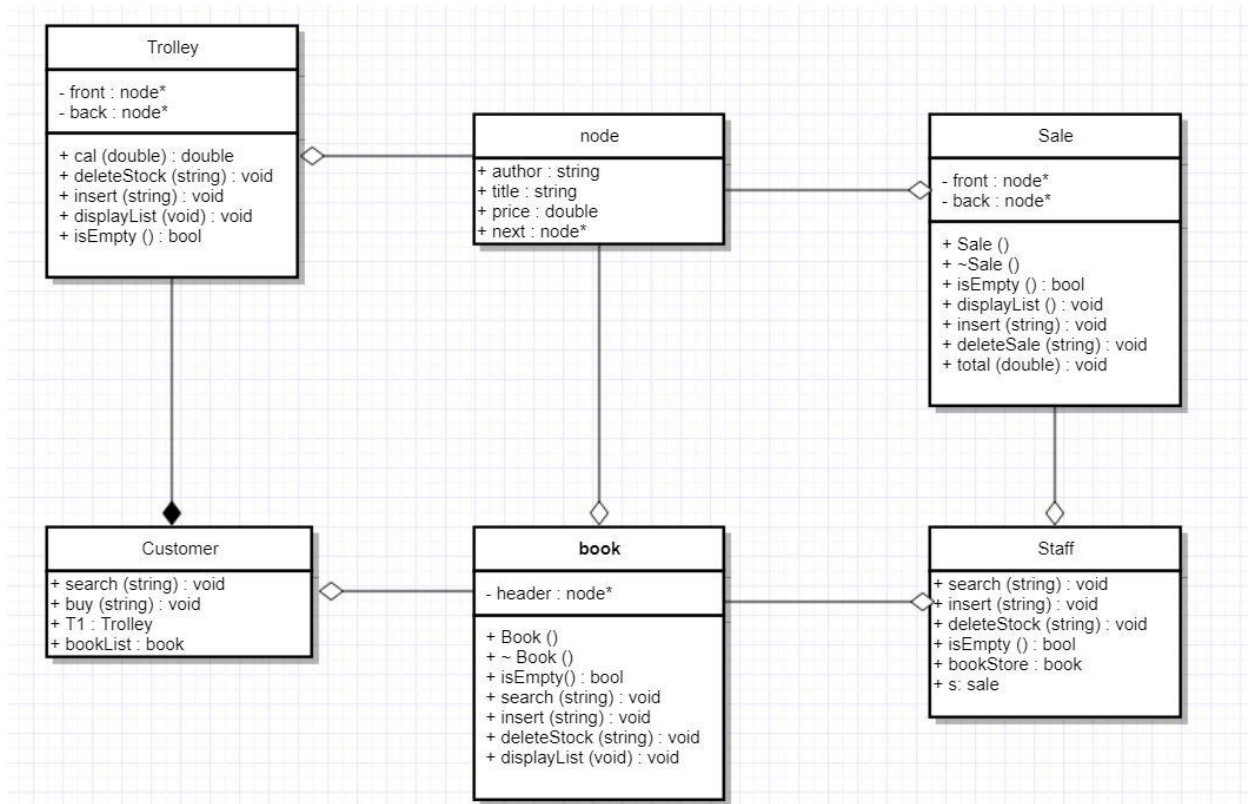


Diagram 1: UML diagram of the book shop management system

Class "node" contains 2 string variables which are author and title, a double price and a pointer named next. There will be classes "nodeTrolley", "nodeBook" and "nodeSale" when implementing the program. They aggregate with class "Trolley", "Book" and "Sale" respectively.

Class "Staff" is also included in the program. It contains s, a variable of class "Sale" and bookStore, a variable of class "Book" datatype. A staff can search a book via search() function and staff can add a new book to the store by using insert() function. If the book store is not selling a particular book already then it can be deleted because function deleteStock() is included in the class "Staff". A boolean function isEmpty() is needed when using deleteStock() function because when a linked list is empty there is nothing to be deleted.

Class "Customer" and class "Trolley" has composition relationship because customer has trolley. It contains T1, a variable of class Trolley datatype and bookList, class book datatype.

Inside class "Customer", function search is added so that customer can search the book either by author or title of the book. Function buy() is to purchase the books and will delete the book in the stock in the same time.

Class "Trolley" has "front" and "back" pointer to indicate the head and tail of the queue. It is a class act like the cart function in any shopping APP. It has to stored a list of chosen book so the function, insert() is needed to add a new chosen book into the list. Function deleteStock() is to allow customer to delete the chosen books if they change their mind. Function isEmpty() will return true when the list empty which means front and back equal to NULL. It is used inside the deleteStock function. Function displayList() is to show the list of book chosen by customer. The cal() function returns a double value after calculating the total price the books.

Class "Sale" has a "front" and "back" pointer which point to the first node and the last node of the list respectively. Class "Sale" also consist of a "Sale" constructor, a "~Sale" destructor, isEmpty(), displayList(), insert(), deleteSale(), and total(). Function isEmpty() is used to check the whether or not the sale is empty while the displayList() is used to print out all the sale details in a list. Insert() function is used to add the sale into the list while deleteSale() is used to delete the sale details in the list. Total() function is used to calculate the total amount of the sales.

Class "Book" has a "head" pointer which point to the first node of the list. Class "Book" also consist of a constructor, a destructor, isEmpty(), search(), insert(), deleteStock() and displayList(). Function isEmpty() is used to check whether or not the book list is empty, search() is used to find a book with a given value either author's name or title, insert() is used to add a new book to the book list, deleteStock() is used to delete a book with a given value and displayList() is used to print out all the books details in the list.

Reference

Alex (2007, July 14) LearnCpp.com 10.3-Aggregation

<https://www.learncpp.com/cpp-tutorial/103-aggregation/>

Alex Marbus (2000, Jun 20) Code Project - OOP and UML

<https://www.codeproject.com/Articles/618/OOP-and-UML>

Jay (2017, August 25) The Coding Delight - Simple Queue Implementation Guide

<https://www.thecodingdelight.com/queue-linked-list/>

Tutorialspoint

https://www.tutorialspoint.com/uml/uml_class_diagram.htm