****

**Mini Project Report**

**Book Management System**

**Data Structure and Algorithm**

**SCSJ2013 – 03**

**Submitted to:**

**Dr Ruhaidah Binti Samsuddin**

**Submitted by:**

**Nazmus Sakib (A17CS0265)**

**Miraj Ahmed (A16CS4017)**

**Md Mirza Shihab (A16CS4017)**

**Emrul Kayser Evan (A17CS4004)**

**Description:**

In this modern world we do everything online. Even to buy a book, we look for it online, and when we look for it online, we get the book more easily. Nowadays very few people have time to go to the store and check the shelf where hundreds or thousands of books are situated. So, for a book shop to manage their books with proper guidance, the management system has to be computerized or programmed from the manual hand-written statements or bills. As this hand-written slips, statements and bills can be lost easily. Moreover the staff should have proper placements of books.

To solve this issue we can make a simple program that can easily let you find the book you are looking for in the book shop. For instance a customer came to “ The C++ Programming Language, 4th Edition by Bjarne Stroustrup” so the programme will get the book name and search it within the book library where all the books are being kept. Not only searching the customer can view the book , buy it.

In this program we used queue implementation to create the project. We gave admin the power to add a book, remove a book, search a book by its name and can also view all the books. The staff can only add or remove books from the server. The customer can search a book by its name or tile then buy it.

The login for admin is : myadmin,

The password for admin is pass1,

The login for staff is : staff1, staff2, staff3, staff4,

Password for staff is : pass1, pass2, pass3 and pass4 respectively.

**Objective:**

**Node :** this class will be used to hold the name, ID, authors and the price of each book in he system. ID refers to the \_ID, name refers to the \_Name, authors refers to the \_publishername and the price refers to the price in rm.

**Queue :** here will be applied the algorithm for the case study relating the book shop management. The enqueue function adds a book and dequeue removes a book from the front. Displaylist will display all the books available, searchbyTitle will search the book by its name and display on he screen. Lastly bookmanagement will divide the admin from the staff.

**Login** : This function manages the admins. An admin can add a book to the server, remove a book from the server, display all the books available and can also search a book by its name.

**Staff** : This function only allows the stuff to add or remove a book from the server. Nothing else.

|  |
| --- |
| queue |
| +bookManagement():void  +deQueue():void  +DisplayList():void  +enQueue():void  +searchByTitle():void  +backPtr:Node\*  +frontPtr:Node\* |

|  |
| --- |
| Node |
| +Node():Node  +ID:int  +Name:string  +Price:double  +publisherName:string |

|  |
| --- |
| Main():int |

|  |
| --- |
| Login(string&admin, string&password):int |

|  |
| --- |
| Staff(string&user , string&pass1):int |

**UML Class Diagram:**