UTM

UNIVERSITI TEKNOLOGI MALAYSIA

FACULTY OF COMPUTING

UNIVERSITI TEKNOLOGI MALAYSIA

DATA STRUCTURE & ALGORITHM (SECJ2013)

SEMESTER 1 2022/2023

UTM BOOK MANAGEMENT SYSTEM

Group Members	1. Nur Shuhada Safiah Binti Ayob (A21EC0114) 2. Nurunnajwa Binti Zulkifli (A21EC0121) 3. Mohamad Azri Hadif bin Mohammad Rizal (A21EC0054) 4. Umar Haziq Bin Muhamad Norhisham (A21EC0235)
Group Name	Toye So
Section	02
Lecturer's Name	Dr Pang Yee Yong
Date	30/1/2023

TABLE OF CONTENT

PART 1: INTRODUCTION

1.1 Synopsis Project

Description:

UTM book management system is a system that allows librarians to add new books and view lists of books, number of books, sort books by the categories and remove the books if needed. To view the book lists and number of books, this system will display all the information needed to the viewers. This system is developed by using various types of data structure concepts in the C++ programming language. To make sure this system runs smoothly, a linked list and stack data structure is used in this system .

Data structures applied:

a) Linked list

   Linked list used in all system's functions. For example it is used in adding new books to the list and it creates a new node for each new book and adds it to the list by adjusting the pointer accordingly.

b) Stack

   The concept of stack is Last In First Out. The implementation of stack is linked list stack. The program is able to remove only the last book and display the latest list of the book.

System functionality:

a) Librarians can have the list of books in the library.
b) Librarians can delete any books if needed and update the number of books left in the library.
c) Librarians can add new books to the record and update the number of books in the library.
d) Librarians can sort the books in the library based on type and genre.

1.2 Objective of The Project

a) To allow librarians experience organised library management.
b) To allow librarians to have records of books in the library.

c) To apply data structure into this project and to ensure the methods chosen are capable of ensuring the functions run smoothly.

PART 2: SYSTEM ANALYSIS AND DESIGN (USE CASE, FLOWCHART AND CLASS DIAGRAM)

2.1.System Requirements Use case diagram

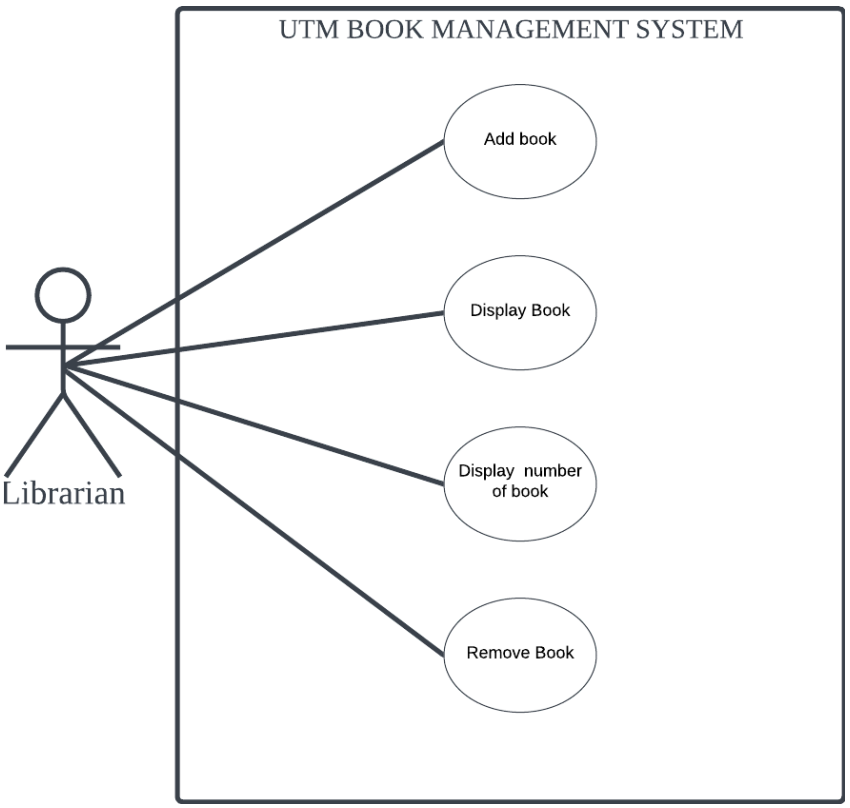


Figure 1: Use case diagram for UTM Book Management System

Use Case Description for Library Management System

The system users are student and librarian

Actor	**Task**
Librarian	Librarians are assigned to add books in the management system. Librarians can also view the total number of books in the system. Additionally, librarians have access to all books' titles stored in the system.A librarian has the authority to remove books from the system.

Detail Description For each use case

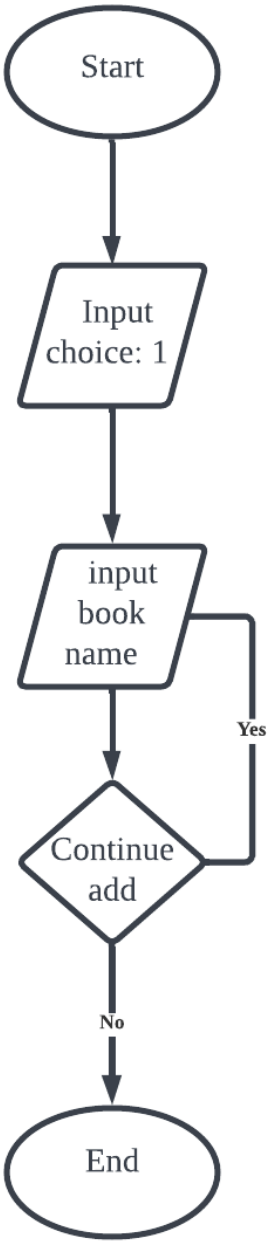
The system has 4 use cases

Use case	**Purpose**
Add books	Add new books in the system
View number of books	View total number of books that is stored in the system
View books' title	View all books' title that are stored in the system
Remove books	Removed stored books from the system.

Algorithm: Flowchart for each module.

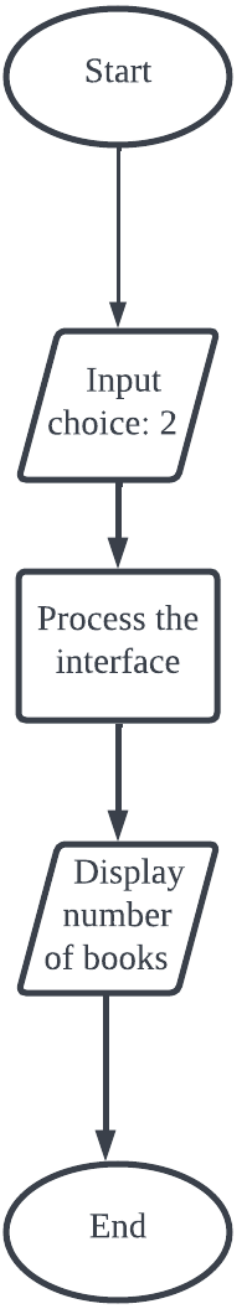
FlowChart 1: Add books

Prepared By: Umar Haziq Bin Muhamad Norhisham



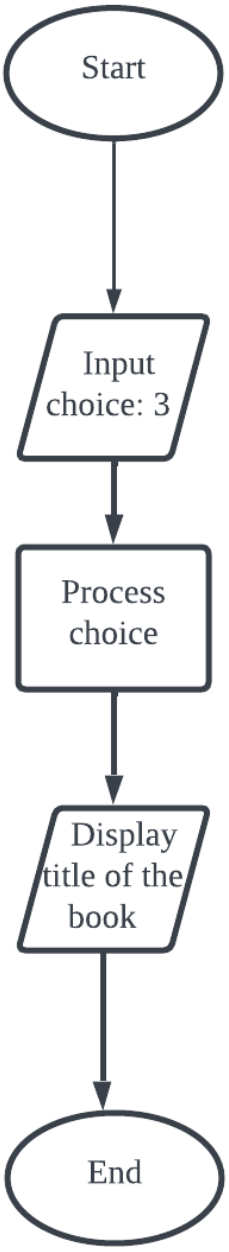
FlowChart 2: Display number of books

Prepared By: Nurunnajwa Binti Zulkifli



FlowChart 3: Display books

Prepared By: Nur Shuhada Safiah Binti Ayob



FlowChart 4: Remove books

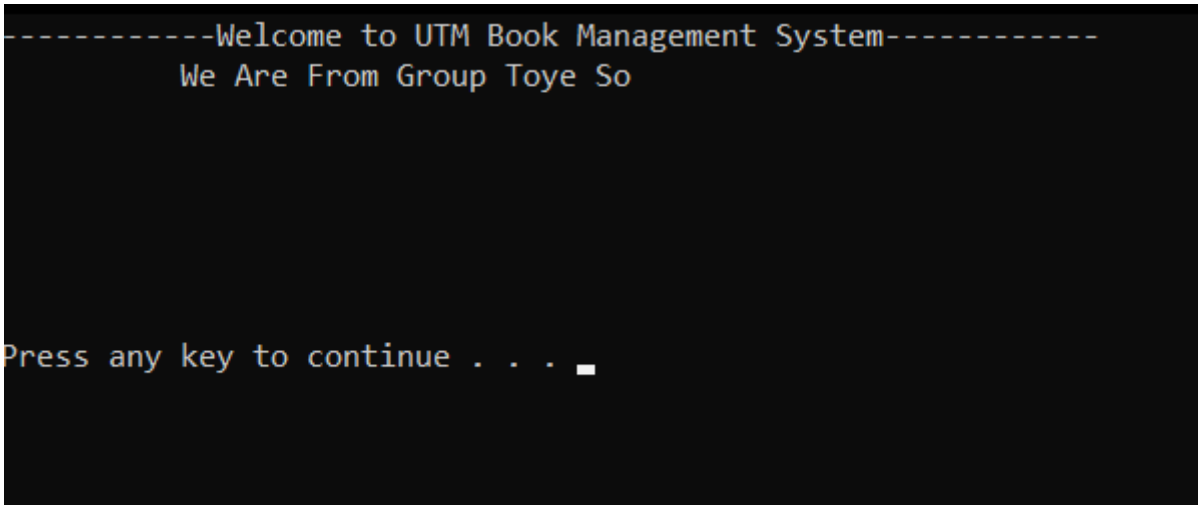
Prepared By: Mohamad Azri Hadif bin Mohammad Rizal

FlowChart 5: Remove books

Prepared By: Mohamad Azri Hadif bin Mohammad Rizal

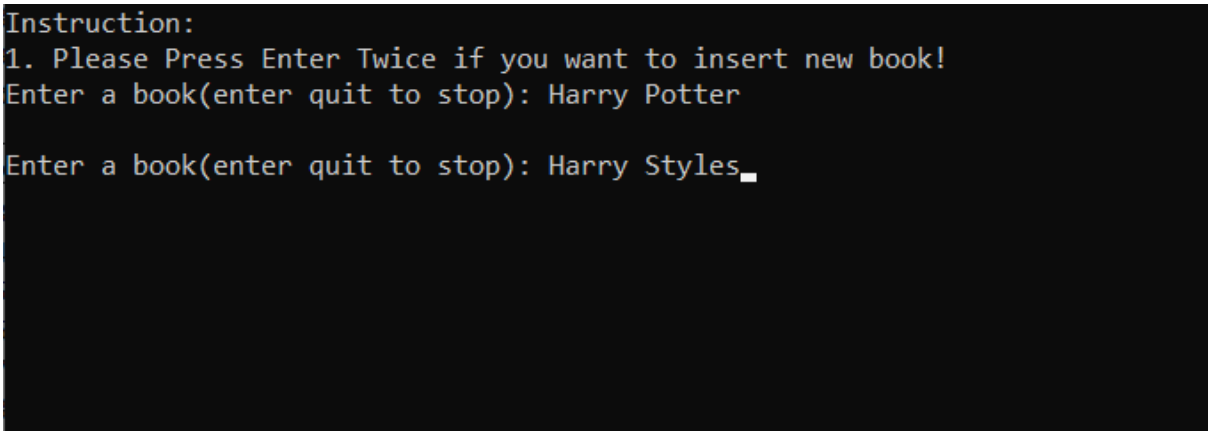
PART 3 : SYSTEM PROTOTYPE

Main Menu



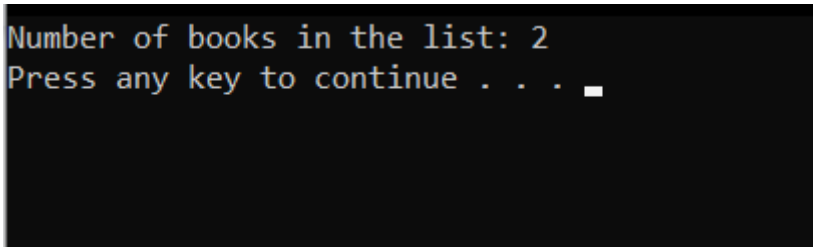
Description: There are 6 choices that users can enter to use the system's Features. User has to determine a choice to begin using the UTM Book Management System.

Register Books Interface



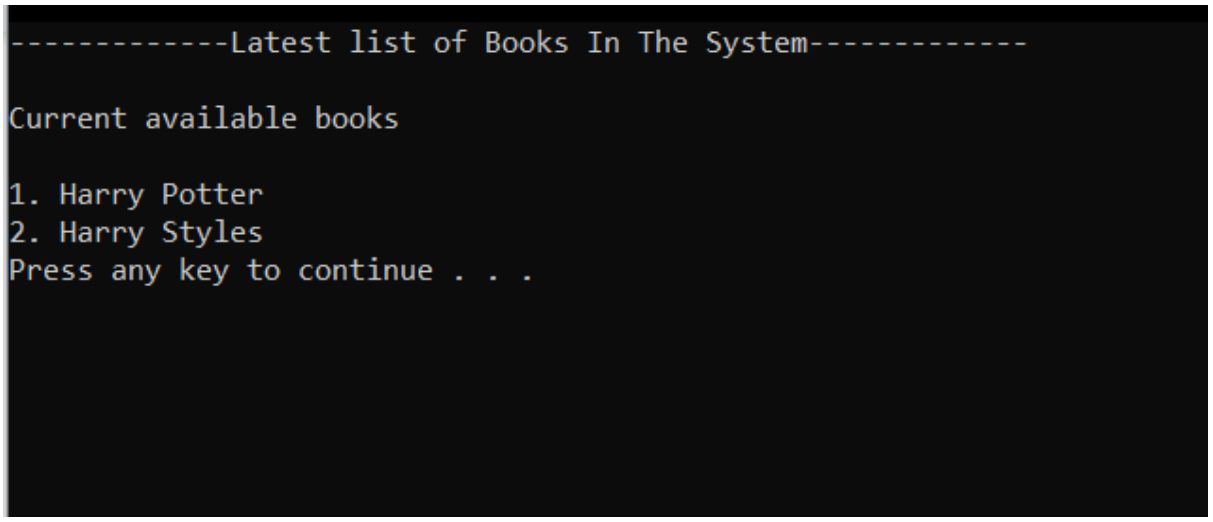
Description: The user must enter the book's title into the system to register the books. Enter must be typed to stop book registration once the user has finished registering all of the books.

Display Number of Books Interface



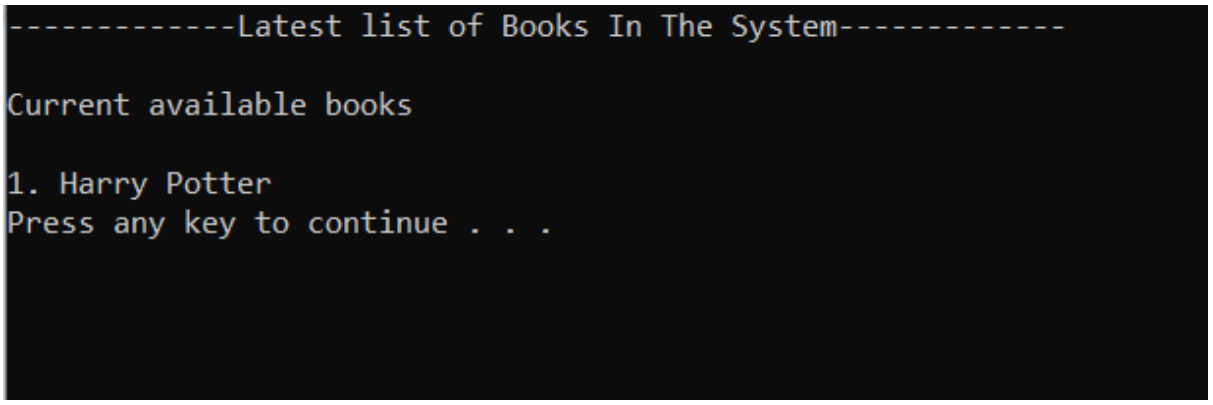
Description: This screen display total amounts of books that have been registered in the system

Display List of Books Interface



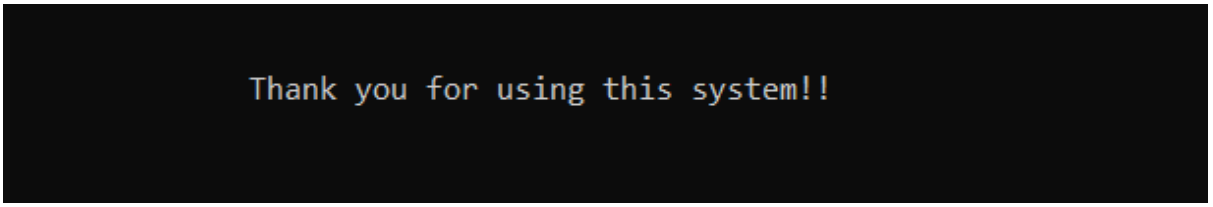
Description: The title of the book that has been registered in the system is shown on this screen.

Display Latest List of Books Interface



Description: This screen displays the latest list of books in the system after the last book has been deleted from the system.

Display Quit Program Interface



Description: This screen displays the interface user will view after quitting the program

PART 4: DEVELOPMENT ACTIVITIES

Meeting Date	Member Participate in the meeting	Activity	Task for each member	Task Achieved (Yes/No)
19/1/2023	**1. Nur Shuhada Safiah Binti Ayob** **2. Nurunnajwa Binti Zulkifli** **3. Mohamad Azri Hadif Bin Mohammad Rizal** **4. Umar Haziq Bin Muhamad Norhisham**	**Discuss the use cases in the system.**	**1. Use case** **2. Flow chart** **3. Synopsis** **4. Description**	**1. Yes** **2. Yes** **3. Yes** **4. Yes**
21/1/2023	**1. Nur Shuhada Safiah Binti Ayob** **2. Nurunnajwa Binti Zulkifli** **3. Mohamad Azri Hadif Bin Mohammad Rizal** **4. Umar Haziq Bin Muhamad Norhisham**	**Combining code for full version project**	**1. Display books** **2. Display number of books** **3. Sort books** **4. Add books** **5. Remove books**	**1. Yes** **2. Yes** **3. Yes** **4. Yes** **5. Yes**

PART 5: APPENDIX

```
#include <iostream>
#include <string>
#include <iomanip>
#include <chrono>
#include <thread>

using namespace std;
using namespace std::this_thread;     // sleep_for, sleep_until
using namespace std::chrono_literals; // ns, us, ms, s, h, etc.
using std::chrono::system_clock;

struct Book
{
  string data;
  Book *next;
};

class linkedList
{
private:
    Book *head,*tail;
public:
  linkedList()
  {
      head = NULL;
      tail = NULL;
  }
  void addBook();
  void displayList();
  void displayList1();
  void listSize();
  void deleteList();
  void deleteLast();
};

void displayMenu();
int getChoice(int & choice1);
int endProgram(bool & start2);
void clear();

void displayMenu()
```

```
{
  cout << setw(35) << "Main Menu" << endl << endl;
  cout << "[1]  Add Book\n"
       << "[2]  Size Of List\n"
       << "[3]  Display List\n"
       << "[4]  Remove Last Book\n"
       << "[5]  Delete List\n"
       << "[6]  Quit Program\n"
       << "Enter Choice:  ";
}
int getChoice(int & choice1)
{
  cin >> choice1;
    while (choice1 < 1 || choice1 > 6) {
      cout << endl;
      cout << "Invalid Entry!!" << endl;
      cout << "Enter Choice:  ";
      cin >> choice1;
    }
  return choice1;
}
int endProgram(bool & start2)
{
  start2 = false;
  cout << "\n\n\t\tThank you for using this system!!\n\n";
  return start2;
}

void linkedList::addBook()
{
  Book *ptr;
  bool quit = false;
  string temp = "";

  cout << "Instruction:" << endl;
      cout << "1. Please press enter twice if you want to insert new
book!" << endl << endl;
  while (!quit)
    {
      cin.ignore();
       cout << "Enter the book name and ISBN Number(enter quit to
stop): ";
      getline(cin, temp);
```

```
    if (temp == "quit")
    {
      quit = true;
      system("CLS");
      break;
    }

    ptr = new Book;
      ptr->data = temp;
    ptr->next = NULL;
    if(head == NULL)
    {
      head = ptr;
      tail = ptr;
    }
    else
    {
      tail->next = ptr;
      tail = tail->next;
    }
  }


}

void linkedList::displayList()
{
  Book *ptr;
  ptr = head;
  int i = 1;
  cout <<"-------------Latest list of Books In The System-------------"
<<endl<<endl;
  cout << "Current available books" << endl << endl;
  while (ptr!= NULL)
  {
    sleep_for(5ns);
    sleep_until(system_clock::now() + 1s);
    cout << i << ". " << ptr->data << endl;
    ptr = ptr->next;
    i++;
  }
  system("pause");
```

```
  system("cls");
}

void linkedList::displayList1() {
  Book *ptr;
  ptr = head;
  int i = 1;
  cout <<"-------------Latest list of Books In The System-------------"
<<endl<<endl;
  cout << "Current available books" << endl << endl;
  while (ptr != NULL) {
  sleep_for(5ns);
    sleep_until(system_clock::now() + 1s);
    cout << i << ". " << ptr->data << endl;
    ptr = ptr->next;
    i++;
  }
  system("pause");
  system("cls");
}

void linkedList::listSize()
{
  Book *ptr;
  int counter = 0;
  ptr = head;
  while (ptr)
  {
    ptr = ptr->next;
    counter++;
  }
  cout << "Number of books in the list: " << counter << endl;
  system("pause");
  system("cls");
}

void linkedList::deleteLast()
{
  if (head == NULL)
    return;

  if (head->next == NULL)
  {
```

```
    delete head;
    head = NULL;
    return;
  }

  // Find the second last node
  Book* ptr = head;
  while (ptr->next->next != NULL)
    ptr = ptr->next;
  // Delete last node
  delete (ptr->next);
  // Change next of second last
  ptr->next = NULL;
  if(ptr == NULL)
    cout << "Last book is cleared!" << endl;

    cout << "Removing the last book";
    int i = 0;
    while (i<5)
    {
      sleep_for(7ns);
      sleep_until(system_clock::now() + 1s);
      cout << ".";
    i++;
  }

    sleep_for(10ns);
    sleep_until(system_clock::now() + 1s);
    cout << "." << endl;

  system("pause");
  system("cls");
}

void linkedList::deleteList()
{
  Book *ptr;
  while (head != NULL)
  {
    ptr = head->next;
    delete head;
    head = ptr;
  }
```

```
  if(head == NULL)
  cout <<"List is cleared!" << endl;
}

int main() {
  int choice = 0;
  bool start = true;
  linkedList a;

     cout << "------------Welcome to UTM Book Management
System------------" <<endl;
   cout << "|" << setw(60) <<"|" << endl;
   cout << "|" << setw(60) <<"|" << endl;
   cout << "|" << setw(42) << "We Are From Group Toye So" << setw(18)
<< "|" << endl;
   cout << "|" << setw(60) <<"|" << endl;
   cout << "|" << setw(60) <<"|" << endl;
   cout <<
"================================================================" <<
endl;
   system("pause");
   system("CLS");

  while(start != false)
  {
    while(choice != 6)
    {
      displayMenu();
      getChoice(choice);
      if(choice == 1)
      {
        system("CLS");
        a.addBook();
         cout << endl;
      }
      else if(choice == 2)
      {
        system("CLS");
        a.listSize();
         cout << endl;
      }
     else  if(choice == 3)
```

```
      {
        system("CLS");
        a.displayList();
        cout << endl;
      }
      else if(choice == 4)
      {
        system("CLS");
        a.deleteLast();
        a.displayList1();
         cout << endl;
      }
      else if(choice == 5)
      {
        system("CLS");
        a.deleteList();
         cout << endl;
      }
      else if(choice == 6)
      {
        system("CLS");
        endProgram(start);
      }
    }
  }
}
```