Sri Lanka Institute of Information Technology

Assignment

MLB_07.01_4

Library Management System

Object Oriented Concepts – IT1050

B.Sc. (Hons) in Information Technology

Topic: Library Management System

Group no: Malabe_07.01_04

Campus: Malabe

Submission Date: 17/05/2022

We declare that this is our own work, and this Assignment does not incorporate without acknowledgment any material previously submitted by anyone else in SLIIT or any other university/Institute. And we declare that each one of us equally contributed to the completion of this Assignment.

Registration No	**Name**	**Contact Number**
IT21159626	Basnayake B R C K	0771743181
IT21349010	Senavirathna P M H M	0711511501
IT21346286	Amaratunga R Y S	0719540830
IT21348570	Ranaweera R K I P	0764068173
IT21347030	Samarasinghe S S	0774019914

Table of contents

Introduction

The main goal of a library management system is to do all the tasks efficiently within a short period of time. Tasks such as borrowing of books, cataloging, recording member details, updating items, indexing, etc. can be very easily done with the help of an automated system. Not only it reduces the huge amount of labor needed it diminishes the chances of mistakes that can happen manually.

Library management system saves time of all users, with just one tap on the screen a member can browse the catalogue and reserve their needed books, librarian can quickly record member details and do the necessary updates real time. Supervising the book stocks can be done in a couple of hours.

UML diagram simply demonstrates the relationship between the different classes of the library management system. It describes the different activities done by different users of the system to achieve their goal.

The attached UML diagram and CRC cards and class implementation codes are some samples to give a rough idea of the intended library management system.

User Requirements

- Guests can register to the system using their personal details.
- Guests can search books from the system.
- Members can login to the system using their user ID and password.
- System will verify the user ID and password.
- Members can search books using book title, author, book ID, and ISBN number.
- Members can see the details of books such as title, author, book ID, ISBN no, publisher, price, and quantity.
- Members can also check availability of the books.
- Members can reserve books using the system.
- Members can check reserved, borrowed, and returned book details in their profile.
- Members can also check payment details in profile.
- Members can update personal details in profile.
- Members can change their passwords.
- Librarian can enter to the system using their user ID and password.
- Librarian can add new books to the database.
- Librarian can also update book details and delete books from database.
- Librarian verifies the membership and check availability status before issuing books.
- Members can pay membership fees, and fines using payment portal.
- System admin can enter to the system using their name.
- System admin has access to the system database.
- Admin can check security, update the system, and change or update the database.
- System can generic a detailed report of daily issued, returned and overdue books and membership details.

Noun Verb Analysis (Nouns in red and verbs in green)

- Guests can register to the system using their personal details.
- Guests can search books from the system.
- Members can login to the system using their user ID and password.
- System will verify the user ID and password.
- Members can search books using book title, author, book ID, and ISBN number.
- Members can see the details of books such as title, author, book ID, ISBN no, publisher, price, and quantity.
- Members can also check availability of the books.
- Members and staff can reserve books using the system.
- Members can check reserved, borrowed, and returned book details in their profile.
- Members can also check payment details in profile.
- Members can update personal details in profile.
- Members can change their passwords.
- Librarian can enter to the system using their user ID and password.
- Librarian can issue books to members .
- Librarian can also update book details and delete books.
- Librarian verifies the membership and check availability status before issuing books.
- Members can pay membership fees, and fines using payment portal.
- System admin can enter to the system using their name.
- System admin has access to the system database.
- Admin can check security, update the system, and change or update the database.
- System can generate a detailed report of daily issued, returned and overdue books and membership details.

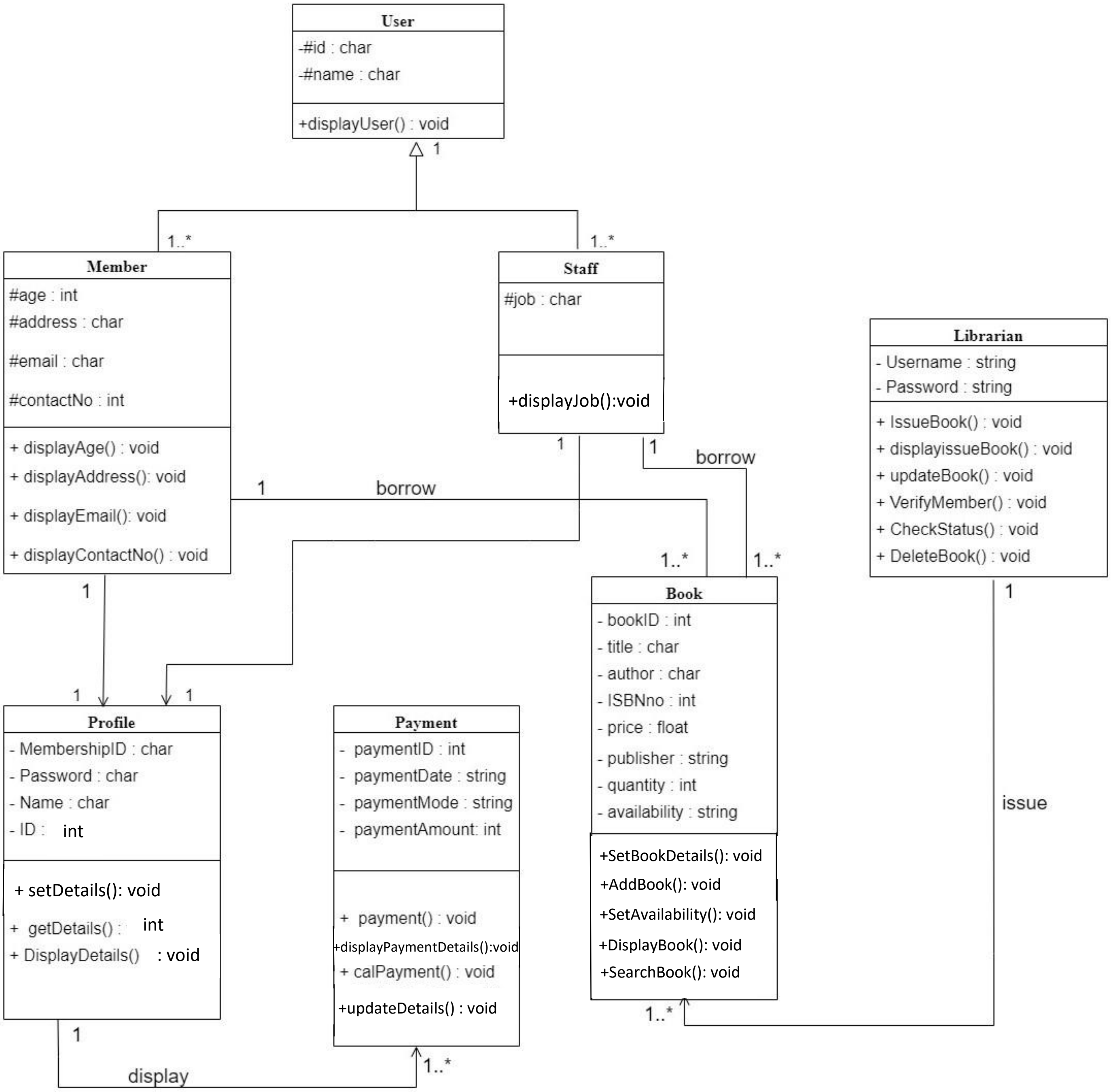
Classes identified through the noun verb analysis are;

- User
- Member
- Staff
- Book
- Librarian
- Payment
- Profile

The other nouns were not considered as there were many redundant nouns, attributes, events, meta language nouns and some were out of scope.

UML diagram was drawn according to the identified classes found above through the noun verb analysis.

UML Diagram


User
-#id : char
-#name : char
+displayUser() : void
1
1..*
1..*
Member
#age : int
#address : char
#email : char
#contactNo : int
+ displayAge() : void
+ displayAddress(): void
+ displayEmail(): void
+ displayContactNo() : void
Staff
#job : char
+displayJob():void
Librarian
- Username : string
- Password : string
+ IssueBook() : void
+ displayissueBook() : void
+ updateBook() : void
+ VerifyMember() : void
+ CheckStatus() : void
+ DeleteBook() : void
1
1
borrow
1
borrow
1..*
1..*
1
1
1
1
Book
- bookID : int
- title : char
- author : char
- ISBNno : int
- price : float
- publisher : string
- quantity : int
- availability : string
+SetBookDetails(): void
+AddBook(): void
+SetAvailability(): void
+DisplayBook(): void
+SearchBook(): void
issue
1..*
Profile
- MembershipID : char
- Password : char
- Name : char
- ID : int
+ setDetails(): void
+ getDetails() : int
+ DisplayDetails() : void
Payment
- paymentID : int
- paymentDate : string
- paymentMode : string
- paymentAmount: int
+ payment() : void
+displayPaymentDetails():void
+ calPayment() : void
+updateDetails() : void
1
display
1..*

CRC Cards

IT21159626

Class: Librarian	
Responsibilities	**Collaborations**
Verify member	Member
Check availability status	Book
Issue books	Book
Update issued books	Book
Delete Book	Book

IT21349010

Class: Book	
Responsibilities	**Collaborations**
Know the details of book title, Author, price, publication, ISBN no and book ID	
Know the book availability	
Know the quantity of the books	
Search book details	Member
Add book details	Librarian

IT21346286

Class: User	
Responsibilities	**Collaborations**
Login/Logout.	
Register/update account.	

Borrow a book.	Librarian
Search catalogue.	
Renew a book.	Librarian
Reserve a book.	Librarian
Remove reservation.	Librarian
Renew the membership.	Librarian, Payment
Return a book.	Librarian
Pay fines.	Payment

IT21348570

Class: Profile	
Responsibilities	**Collaborations**
Update personal details	
Change the password	
Check payment details	Payment
Check reserved, borrowed, and returned books	Book, Member

IT21347030

Class: Payment	
Responsibilities	**Collaborations**
Add payment methods	
Add extra charges	Librarian
Update payment information	
Display payment details	

Codes

IT21159626 – Librarian class

Librarian.h

```
#include "Book.h"
#define SIZE 10
class Librarian{
    private:
          char username[20];
          char password[10];
          char libID[5];
         Book *book[SIZE];
    public:
       Librarian();
       Librarian(char *name,char *pw,char *ID);
       void issueBook(Book *B);
       void displayissueBook();
       void updateBook();
       void verifyMember();
       void checkStatus();
       void deleteBook();
       ~Librarian();
};
```

Librarian.cpp

```
#include <iostream>
#include "Book.h"
#include "Librarian.h"
#include <cstring>
#define SIZE 10
using namespace std;
//implementation of methods of librarian class

//default constructor
Librarian::Librarian(){
    strcpy(username,"");
    strcpy(password,"");
    strcpy(libID,"");
}
//overloaded constructor
Librarian::Librarian(char *name,char *pw, char *ID){
    strcpy(username,name);
    strcpy(password,pw);
    strcpy(libID,ID);
}

//implemetation of methods
void Librarian::issueBook(Book *B){
    strcpy(book,B);
}
```

```
void Librarian::displayissueBook(){


}


void Librarian::updateBook(){


}
void Librarian::verifyMember(){


}


void Librarian::checkStatus(){


}
void Librarian::deleteBook(){


}
//destructor
Librarian::~Librarian(){
  cout<<"Destructor for Librarian class"<<endl;
}
```

Main.cpp

```
//uni directional association between book and librarian
#include <iostream>
#include "Librarian.h"
#include "Book.h"
#include <cstring>
#define SIZE 10
using namespace std;
int main() {

  //creation of dynamic objects
  Librarian *L1;
  Librarian *L2;

  Book *B1;
  Book *B2;

  L1 = new Librarian(); //default constructor
  L2 = new Librarian("NimaliPerera","neeper","LIB01");//overloaded
constructor

  B1 = new Book(); //default constructor
  B2 = new Book();

  //calling the methods
  L1-> issueBook();
  L2-> issueBook();
  cout<< "Display Book Issues" <<endl;
  L1-> displayissueBook();
  L2-> displayissueBook();

  L1-> updateBook();
  L2-> updateBook();

  L1-> verifyMember();
  L2-> verifyMember();

  L1-> checkStatus();
  L2-> checkStatus();

  L1-> deleteBook();
  L2-> deletebook();

  //deleting the created objects of class librarian
  delete L1;
  delete L2;

  return 0;
}
```

IT21349010 – Book class

Book.h

```
#include <iostream>
#include <cstring>
using namespace std;

class Book{
                char bookID[10];
                char title[100];
                char author[100];
                int ISBNno;
                float price;
                int quantity;
                string publisher;
                string availability;
        public:
                book();
                void SetBookDetails(char ID[10],char name[100],char
author[100],int isbnNo,float price,int qty,string publisher,string
availability);
                void AddBook();
                void SetAvailability();
                void DisplayBook();
                void SearchBook();

};
```

Book.cpp

```
#include <iostream>
#include <cstring>
#include "Book.h"
using namespace std;

//Implementation
//Default constructor
Book::Book()
{
        ISBNno = 0;
        price = 0;
        publisher = "";
        quantity = 0;
}
void Book::SetBookDetails(char ID[10],char name[100],char author[100],int
isbnNo,float price,int qty,string publisher,string Availability)
{
        bookID[10] = ID[10];
        title[100] = name[100];
        author [100]= author[100];
        ISBNno = isbnNo;
        price = price;
        publisher = publisher;
        quantity = qty;
        availability = Availability;
}
```

```
void Book::AddBook()
{

        system("cls");
        cout<<"Book Title   : ";
        gets(title);
        cout<<"Author       : ";
        gets(author);
        cout<<"Book ID      : ";
        cin>>bookID;
        cout<<"ISBN number  : ";
        cin>>ISBNno;
        cout<<"Publisher    : ";
        cin>>publisher;
        cout<<"Price        : ";
        cin>>price;
        cout<<"Quantity     : ";
        cin>>quantity;
        system("cls");
}
void Book::SetAvailability()
{

}
```

```
void Book::DisplayBook()
{
        int i;
        cout<<"Title of the book : " <<title<<endl
                <<"Author           : " <<author<<endl
                <<"Book ID          : " <<bookID<<endl
                <<"ISBN no          : " <<ISBNno<<endl
                <<"Publisher        : " <<publisher<<endl
                <<"Price            : " <<price<<endl
                <<"Quantity         : " <<quantity<<endl
                <<"Status           : " <<availability<<endl;
}

void Book::SearchBook()
{


}
```

Main.cpp

```
int main()
{
        Book book;

        book.AddBook();
        book.SetAvailability();
        book.DisplayBook();
        book.SearchBook();

        return 0;
}
```

IT21346286 – User class

```
#include <iostream>
#include <cstring>
using namespace std;

class User
{
	protected:
		char name[20];
		char id[10];

	public:
		void assignId(char *usId);
		void displayId();
		void assignName(char *usName);
		void displayUser();
};

void User::assignName(char *usName){
	strcpy(name, usName);
}

void User::displayUser(){
	cout<<"User Name : "<<name<<endl;
}

void User::assignId(char *usId){
        strcpy(id, usId);
}

void User::displayId(){
        cout<<"User id : "<<id<<endl;
}

class Member : public User
{
        protected:
                int age;
                char address[50];
                char email[20];
                int contactNo;
        public:
                void assignAge(int a);
                void displayAge();
                void assignAddress(char *mAddress);
                void displayAddress();
                void assignEmail(char *mEmail);
                void displayEmail();
                void assignContactNo(int no);
                void displayContactNo();
};
```

```
void Member::assignAge(int a){
	age = a;
}

void Member::displayAge(){
	cout<<"Age : "<<age<<endl;
}

void Member::assignAddress(char *mAddress){
	strcpy(address, mAddress);
}

void Member::displayAddress(){
	cout<<"Address : "<<address<<endl;
}

void Member::assignEmail(char *mEmail){
	strcpy(email, mEmail);
}

void Member::displayEmail(){
	cout<<"Email address : "<<email<<endl;
}

void Member::assignContactNo(int no){
	contactNo = no;
}

void Member::displayContactNo(){
        cout<<"Contact no : "<<contactNo<<endl;
}



class Staff : public User
{
        protected:
                char job[20];
        public:
                void assignJob(char *sJob);
                void displayJob();
};

void Staff::assignJob(char *sJob){
        strcpy(job, sJob);
}

void Staff::displayJob(){
        cout<<"Library position : "<<job<<endl;
}
```

```
int main(){
    Member m1, m2;

    m1.assignName("Kamal Gunaratne");
    m1.assignId("LMS1000234");
    m1.assignAddress("No:72, Galle Road, Colombo");
    m1.assignAge(12);
    m1.assignEmail("kamal65@gmail.com");
    m1.assignContactNo(0701234567);
    m1.displayUser();
    m1.displayId();
    m1.displayAddress();
    m1.displayAge();
    m1.displayEmail();
    m1.displayContactNo();

    Staff s1;
    s1.assignJob("Librarian");
    s1.displayJob();

    return 0;
}
```

IT21348570 – Profile class

Profile.h

```
class Profile{

        private:

                int ID;
                char Name[50];
                char password[20];
                char membershipID[20];

        public:

                void setDetails(int inputID, char inputName[], char  inputpassword[], char
m_membership[]);
                void displayDetails();
                void getDetails();
};
```

Profile.cpp

```
#include "Profile.h"

#include <iostream>

#include <cstring>

using namespace std;


void Profile::setDetails(int inputID, char inputName[], char  inputpassword[], char
m_membership[]){


        ID = inputID;

        strcpy(Name, inputName);

        strcpy(password, inputpassword);

        strcpy(membershipID, m_membership);


}

void Profile::displayDetails(){


        cout << "ID :" << ID << endl;

        cout << "Name :" << Name << endl;

        cout << "Password :" << password << endl;

        cout << "Membership :" << membershipID << endl;

}


void Profile::getDetails(){


}
```

Main.cpp

```
#include "Profile.h"
#include <iostream>
using namespace std;

int main()
{
        Profile u1,u2,u3,u4,u5;
        u1.setDetails(3570, "Alan", "A3570", "yes");
        u2.setDetails(5512, "James", "J5512", "yes");
        u3.setDetails(9090, "Jack", "J9090", "yes");
        u4.setDetails(8563, "Noah", "N8563", "yes");
        u5.setDetails(0012, "David", "D0012", "yes");

        return 0;

}
```

IT21347030 – Payment class

Payment.h

```
#include <iostream>
using namespace std;
class Payment {

        private:

                int paymentID;
                string paymentDate;
                string paymentMode;
                int paymentAmount;

        public:

                void payment(int P_ID, string P_Date, string P_Mode, int P_Amount);
                void displayPaymentDetails();
                void updateDetails();
                void calPayment();

};
```

Payment.cpp

```
#include"Payment.h"
#include<iostream>
#include<cstring>
using namespace std;

 void Payment::payment(int P_ID, string P_Date, string P_Mode, int P_Amount){

   paymentID = P_ID;
   paymentDate = P_Date;
   paymentMode = P_Mode;
   paymentAmount = P_Amount;
}
void Payment::displayPaymentDetails(){

        cout<<"Pyment ID: "<<paymentID<<endl;
        cout<<"Payment Date "<<paymentDate<<endl;
        cout<<"Payment Mode "<<paymentMode<<endl;
        cout<<"Payment Amount "<<paymentAmount<<endl;
}

void Payment::calPayment(){

}
void Payment::updateDetails(){
}
```

Main.cpp

```
#include"Payment.h"
#include<iostream>
using namespace std;
int main()
{
	Payment cp1;
	cp1.payment(21348,"2022/01/07","Card payment", 17000);



	Payment cp2;
	cp2.payment(21148,"2022/01/10","Card payment", 5000);



	Payment cp3;
	cp3.payment(21577,"2022/01/15","Cash payment", 7500);



	Payment cp4;
	cp4.payment(77848,"2022/01/11","Check payment", 80000);



	Payment cp5;
	cp5.payment(57912,"2022/01/24","Card payment", 10000);
	return 0;
}
```

Individual Contribution

	Student ID	Student Name	Individual Contribution
1	IT21159626	Basnayake B R C K	• Noun verb analysis • Created the UML Librarian class diagram • Created the CRC card for Librarian class • Implemented the coding for Librarian class
2	IT21349010	Senavirathna P M H M	• Documented the user requirements • Created the UML Book class diagram • Created the CRC card for Book class • Implemented the coding for Book class
3	IT21346286	Amaratunga R Y S	• Created the UML User class diagram • Created the CRC card for User class • Implemented the coding for User class
4	IT21348570	Ranaweera R K I P	• Created the UML Profile class diagram • Created the CRC card for Profile class • Implemented the coding for Profile class
5	IT21347030	Samarasinghe S S	• Created the UML Payment class diagram • Created the CRC card for Payment class • Implemented the coding for Payment class