

Project Documentation

on

LIBRARY MANAGEMENT SYSTEM

SAYANIK BHOWMIK

//C++ CODE

```
#include <bits/stdc++.h>
```

```
#include <cstdlib>
```

```
#include <string>
```

```
using namespace std;
```

```
class Library {
```

```
public:
```

```
    int transNumber, transac = 0, ret = 0;
```

```
    string name, password, pay, p, newname, newpass, search, borrow,  
    feedback = "No Feedback provided";
```

```
    string bn[5], rn[5];
```

```
    long long regnum, bal = 10000, amt;
```

```
    void newReg() {
```

```
        cout << "\n-----**Your Profile*-----" << "\n\n"; cout
```

```
        << "Enter Name: ";
```

```
        cin >> newname;
```

```
        cout << "Enter Registration No: "; cin >>
```

```
        regnum;
```

```
        cout << "Enter password: "; cin >>
```

```
        newpass;
```

```
        cout << "\n";
```

```
    }
```

```
    void libInfo() {
```

```
        cout << "\n-----*LIBRARY*-----" << "\n\n"; cout
```

```
        << "Enter Name: ";
```

```
        cin >> name;
```

```
        cout << "Password: "; cin >>
```

```
        password; cout << "\n\n";
```

```
    }
```

```
    void libSearch() {
```

```
        cout << "\nSearch by Book Name: "; cin >>
```

```
        search;
```

```
        cout << "Available for borrowing";
```

```
    }
```

```

    void libBorrow() {
        cout << "\nEnter Book Name to be borrowed: "; cin
        >> bn[transac];
        transac++; cout
        << "\n";
        cout << "Book Borrowed: ";
        cout << bn[transac - 1] << "\n\n";
    }

```

```

void libReturn() {
    cout << "\n";
    cout << "Enter Book Name to be returned: "; cin
    >> rn[ret];
    ret++;
    cout << "Book Returned: "; cout
    << rn[ret - 1] << "\n\n";
}

```

```

    void display() {
        cout << "\nName: " << name << "\n";
        cout << "Registration No: " << regnum << "\n"; cout
        << "Books borrowed\n";
        for (int i = 0; i < transac; i++) {
            cout << bn[i] << "\n";
        }
        cout << "Books returned \n"; for
        (int i = 0; i < ret; i++) {
            cout << rn[i] << "\n";
        }
    }
};

```

```

int main() {
    Library user;
    int ch, a;
    user.newReg
    ();
    user.libInfo()
    ;
}

```

```

    while (true) {
    cout << "----*Enter Your Choice* ";
    cout << "\n1. Search Book"; cout
    << "\n2. Borrow Book"; cout <<
    "\n3. Return Book";
    cout << "\n\nEnter your choice: "; cin
    >> ch;

    switch (ch) {
    case '1':
        user.libSearch();

        break;
    case
    '2':
        user.libBorro
        w(); break;
    case '3':
        user.libReturn
        (); break;
    }

    cout << "\n";
    cout << "Do you wish to continue? Enter 1 to continue, 0 to exit: "; cin
    >> a;
    cout << "\n";
    }

    user.display
    (); return 0;
}

```

Oriented Design for Library management System

A Library Management System is a software built to handle the primary housekeeping functions of a library. Library management systems help libraries keep track of the books and their checkouts, as well as members' subscriptions and profiles.

Library management systems also involve maintaining the database for entering new books and recording books that have been borrowed with their respective due dates.

We have three main actors in our system:

1. Librarian – Mainly responsible for adding and modifying books and users. The Librarian can also issue and receive returned books.

2. Student – All students can check out, issue and return a book.

3. Book

The use cases of the Library Management System are:

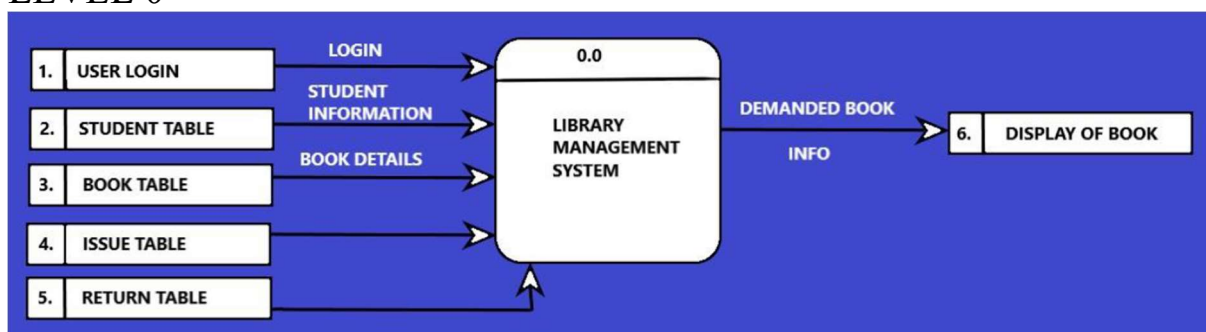
1. Add/ Remove book – To add or remove a book.

2. Check-out book – To issue a book from the library.

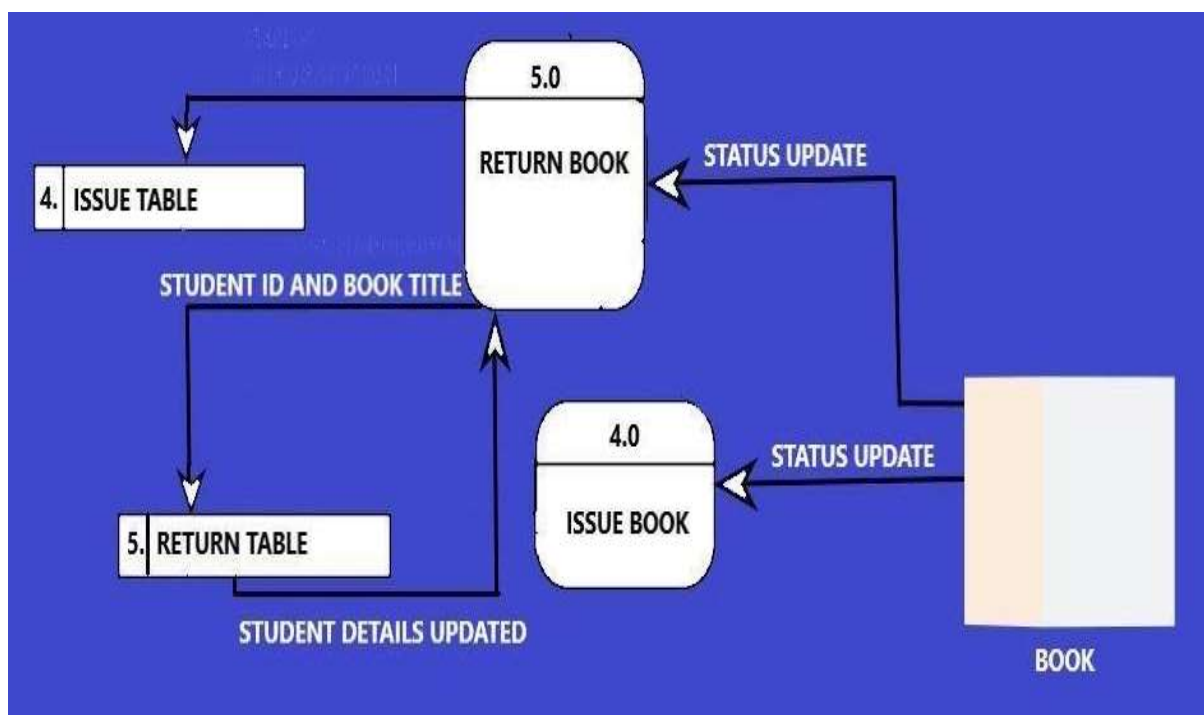
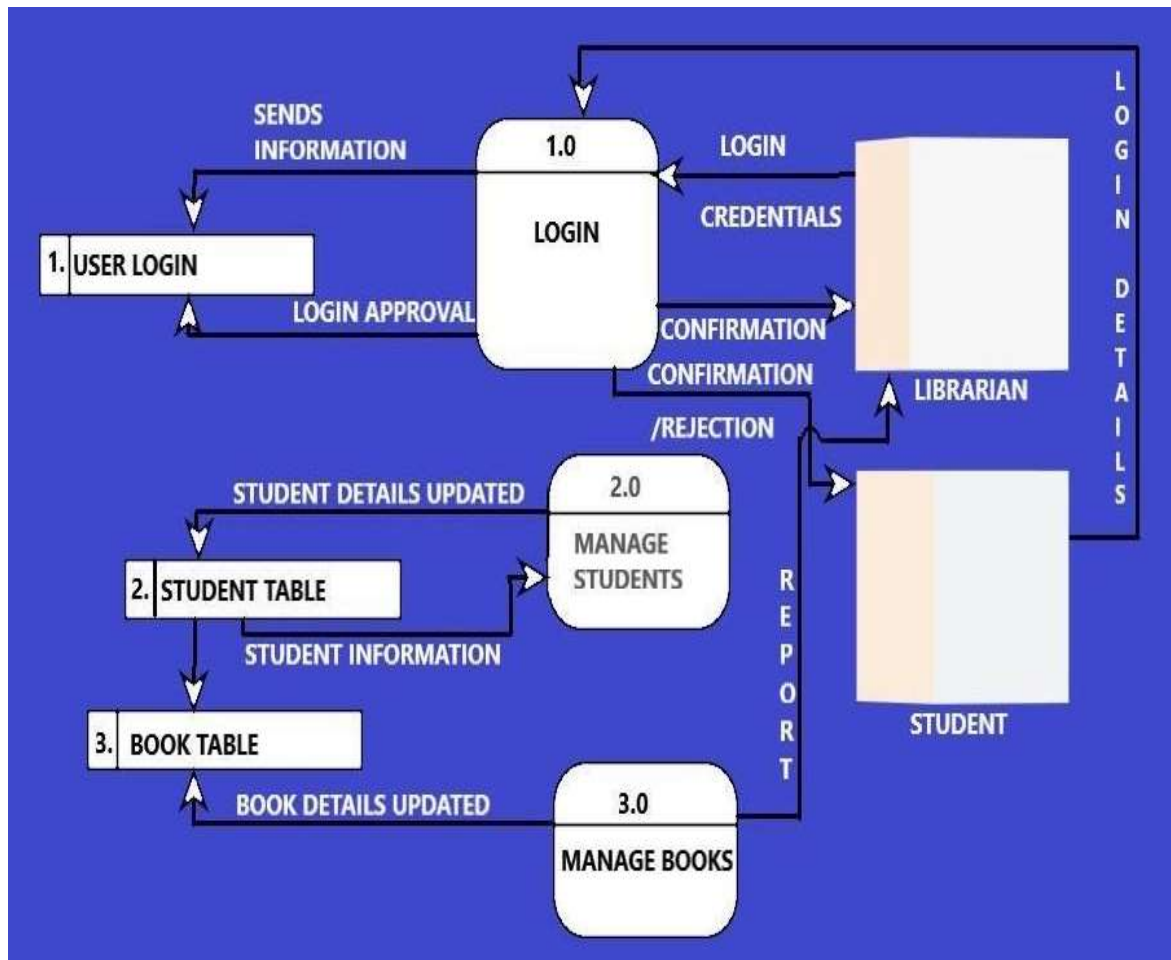
3. Return a book – To return a book to the library which was issued by a user.

DATA FLOW DIAGRAM

LEVEL 0



LEVEL 1



LEVEL 2

