```cpp
#include <iostream.h>
#include <fstream.h>
#include <iomanip.h>
#include <stdlib.h>
#include <stdio.h>


class Book {
    long bookNumber;
    char bookName[100];
    char author[50];
public:
    void add_new_book() {
        cout << "\n\n\t\t\t** ADD NEW BOOK **\n\n";
        cout << "\nBOOK NUMBER: ";
        cin >> bookNumber;
        fflush(stdin);
        cout << "\nNAME OF THE BOOK: ";
        gets(bookName);
        fflush(stdin);
        cout << "\nNAME OF THE AUTHOR: ";
        gets(author);
        fflush(stdin);
        cout << "\n** BOOK SUCCESSFULLY ADDED! **\n";
    }

    void view_book() {
        cout << "\nBOOK NUMBER: " << bookNumber;
        cout << "\n\nNAME OF THE BOOK: " << bookName;
        cout << "\n\nNAME OF THE AUTHOR: " << author;
    }

    void update_book() {
        cout << "\n\n ** ENTER NEW DETAILS!**\n\n";
        cout << "\nBOOK NAME: ";
        gets(bookName);
        cout << "\nNAME OF THE AUTHOR: ";
        gets(author);
    }

    long get_book_number() {
        return bookNumber;
    }

    void report() {
        cout << endl;
        cout << setw(10);
        cout << bookNumber;
        cout << setw(40);
        cout << bookName;
        cout << setw(30);
        cout << author;
    }
};


class Student {
    long admissionNumber;
    char studentName[50];
    long issuedBookNumber;
    int booksIssued;
public:
    void add_new_student() {
        cout << "\n\n\t\t\t** ADD NEW STUDENT **\n\n";
        cout << "\nADMISSION NUMBER: ";
        cin >> admissionNumber;
        fflush(stdin);
        cout << "\nNAME OF THE STUDENT: ";
        gets(studentName);
        fflush(stdin);
        booksIssued = 0;
        cout << "\n\n** STUDENT SUCCESSFULLY ADDED! **\n\n";
    }

    void view_student() {
        cout << "\nADMISSION NUMBER: " << admissionNumber;
        cout << "\nNAME OF THE STUDENT: " << studentName;
        cout << "\nNUMBER OF BOOKS ISSUED: " << booksIssued;
        if (!booksIssued) {
            cout << "\n\n** THE STUDENT HAS ISSUED NO BOOKS!**\n\n";
        }
    }

    void update_student() {
        cout << "\n\n ** ENTER NEW DETAILS!**\n\n";
        cout << "\nNAME OF THE STUDENT: " << studentName;
    }

    long get_admission_number() {
        return admissionNumber;
    }

    long get_issued_books() {
        return issuedBookNumber;
    }

    int get_issued_number() {
        return booksIssued;
    }

    void add_issued_books() {
        booksIssued++;
    }

    void return_book() {
        booksIssued--;
    }

    void set_book_number(long book_number) {
        issuedBookNumber = book_number;
    }

    void report() {
        cout << endl;
        cout << setw(10);
        cout << admissionNumber;
        cout << setw(30);
        cout << studentName;
        cout << setw(10);
        cout << booksIssued;
    }
};


fstream f;
Book book;
Student student;


void new_book() {
    char choice;
    f.open("books.dat", ios::app);
```

```cpp
    do {
        system("cls");
        book.add_new_book();
        f.write((char *) &book, sizeof(Book));
        cout << "\n\nDO YOU WANT TO ADD MORE RECORDS? [Y/n] ";
        cin >> choice;
        fflush(stdin);
    } while (tolower(choice) != 'n');
    f.close();
}


void new_student() {
    char choice;
    f.open("students.dat", ios::app);
    do {
        system("cls");
        student.add_new_student();
        f.write((char *) &student, sizeof(Student));
        cout << "\n\nDO YOU WANT TO ADD MORE RECORDS? [Y/n] ";
        cin >> choice;
        fflush(stdin);
    } while (tolower(choice) != 'n');
    f.close();
}


void display_book(long bookNumber) {
    system("cls");
    cout << "\n\n\t\t\t** BOOK DETAILS ** \n\n";
    int flag = 1;
    f.open("books.dat", ios::in);
    while (f.read((char *) &book, sizeof(Book))) {
        if (bookNumber == book.get_book_number()) {
            book.view_book();
            flag = 0;
        }
    }
    f.close();
    if (flag) {
        cout << "\n\n** BOOK DOES NOT EXIST! **\n\n";
    }
    else {
        cout << "\n\n";
    }
    system("pause");
}


void display_student(long admissionNumber) {
    system("cls");
    cout << "\n\n\t\t\t** STUDENT DETAILS ** \n";
    int flag = 1;
    f.open("students.dat", ios::in);
    while (f.read((char *) &student, sizeof(Student))) {
        if (admissionNumber == student.get_admission_number()) {
            student.view_student();
            flag = 0;
        }
    }
    f.close();
    if (flag) {
        cout << "\n\n** STUDENT DOES NOT EXIST! **\n\n";
    }
    else {
        cout << "\n\n";
    }
    system("pause");
}


void update_book_record() {
    long bookNumber;
    int found = 0;
    cout << "\n\n\t\t\t** MODIFY BOOK RECORD **\n\n";
    cout << "\nBOOK NUMBER OF THE BOOK TO BE UPDATED: ";
    cin >> bookNumber;
    fflush(stdin);
    f.open("books.dat", ios::in | ios::out);
    while (f.read((char *) &book, sizeof(Book))) {
        if (book.get_book_number() == bookNumber) {
            book.view_book();
            book.update_book();
            int pos = -1 * sizeof(Book);
            f.seekp(pos, ios::cur);
            f.write((char *) &book, sizeof(Book));
            cout << "\n\n** BOOK SUCCESSFULLY UPDATED! **
\n\n";
            found = 1;
        }
    }
    f.close();
    if (!found) {
        cout << "\n\n** BOOK NOT FOUND!! **\n\n";
    }
    system("pause");
}


void update_student_record() {
    long admissionNumber;
    int found = 0;
    cout << "\n\n\t\t\t** MODIFY STUDENT RECORD **\n\n";
    cout << "\nADMISSION NUMBER OF THE STUDENT TO BE UPDATED: ";
    cin >> admissionNumber;
    fflush(stdin);
    f.open("student.dat", ios::in | ios::out);
    while (f.read((char *) &student, sizeof(Student))) {
        if (student.get_admission_number() == admissionNumber) {
            student.view_student();
            student.update_student();
            int pos = -1 * sizeof(Student);
            f.seekp(pos, ios::cur);
            f.write((char *) &student, sizeof(Student));
            cout << "\n\n** STUDENT SUCCESSFULLY UPDATED! **\n\n";
            found = 1;
        }
    }
    f.close();
    if (!found) {
        cout << "\n\n** STUDENT NOT FOUND!! **\n\n";
    }
    system("pause");
}


void remove_student() {
    long admissionNumber;
    int flag = 0;
    cout << "\n\n\t\t\t** DELETE STUDENT **\n\n";
    cout << "\nADMISSION NUMBER OF THE STUDENT TO BE DELETED: ";
    cin >> admissionNumber;
    fflush(stdin);
    f.open("students.dat", ios::in | ios::out);
    fstream fil;
    fil.open("temp.dat", ios::out);
    f.seekg(0, ios::beg);
    while (f.read((char *) &student, sizeof(Student))) {
        if (student.get_admission_number() != admissionNumber) {
            fil.write((char *) &student, sizeof(Student));
        }
        else {
            flag = 1;
        }
    }
    fil.close();
    f.close();
    remove("students.dat");
    rename("temp.dat", "students.dat");
    if (flag) {
        cout << "\n\n** STUDENT-RECORD DELETED!! **\n\n";
    }
    else {
        cout << "\n\n** STUDENT NOT FOUND!! **\n\n";
    }
    system("pause");
}


void remove_book() {
    long bookNumber;
    int flag = 0;
    cout << "\n\n\t\t\t** DELETE BOOK  **\n\n";
    cout << "\nBOOK NUMBER OF THE BOOK TO BE DELETED: ";
    cin >> bookNumber;
    fflush(stdin);
    f.open("books.dat", ios::in | ios::out);
    fstream fil;
    fil.open("temp.dat", ios::out);
    f.seekg(0, ios::beg);
    while (f.read((char *) &book, sizeof(Book))) {
        if (book.get_book_number() != bookNumber) {
            fil.write((char *) &book, sizeof(Book));
        }
        else {
            flag = 1;
        }
    }
    fil.close();
    f.close();
    remove("books.dat");
    rename("temp.dat", "books.dat");
    if (flag) {
        cout << "\n\n** BOOK-RECORD DELETED!! **\n\n";
    }
    else {
        cout << "\n\n** BOOK NOT FOUND!! **\n\n";
    }
    system("pause");
}


void student_list() {
    f.open("students.dat", ios::in);
    if (!f) {
        cout << "\n\n** THERE ARE NO STUDENTS IN THE SYSTEM! **\n\n";
        cout << "\n\n";
        system("pause");
        return;
    }

    cout << "\n\n\t ** LIST OF THE STUDENTS IN THE SYSTEM **\n\n";
    cout << "==========================================================\n";
    cout << "ADMISSION NUMBER" << "     " << "NAME OF THE STUDENT" << "     " << "BOOKS ISSUED\n";
    cout << "==========================================================\n";

    while (f.read((char *) &student, sizeof(Student))) {
        student.report();
    }
    f.close();
    cout << "\n\n";
    system("pause");
}


void book_list() {
    f.open("books.dat", ios::in);
    if (!f) {
        cout << "\n\n** THERE ARE NO BOOKS IN THE SYSTEM ! **\n\n";
        cout << "\n\n";
        system("pause");
        return;
    }
    cout << "\n\n\t\t\t** LIST OF THE BOOKS IN THE SYSTEM **\n\n";
    cout << "=========================================================================================\n";
    cout << "-- BOOK NUMBER --" << "             " << "-- NAME OF THE BOOK --" << "           " << "-- NAME OF THE AUTHOR --\n";
    cout << "=========================================================================================\n";

    while (f.read((char *) &book, sizeof(Book))) {
        book.report();
    }
    f.close();
    cout << "\n\n";
    system("pause");
}


void issue_book() {
    long admissionNumber, bookNumber;
    int found = 0, flag = 0;
    fstream fil;
    cout << "\n\n\t\t** BOOK ISSUE **\n\n";
    cout << "\n\nADMISSION NUMBER OF THE STUDENT: ";
    cin >> admissionNumber;
    fflush(stdin);
    f.open("students.dat", ios::in | ios::out);
    fil.open("books.dat", ios::in | ios::out);
    while (f.read((char *) &student, sizeof(Student))) {
        if (student.get_admission_number() == admissionNumber) {
            found = 1;
            if (student.get_issued_number() < 1) {
                cout << "\n\nBOOK NUMBER OF THE BOOK TO BE ISSUED: ";
                cin >> bookNumber;
                while (fil.read((char *) &book, sizeof(Book))) {
                    if (book.get_book_number() == bookNumber) {
                        book.view_book();
                        flag = 1;
                        student.add_issued_books();
                        student.set_book_number(bookNumber);
                        int pos = -1 * sizeof(Student);
                        f.seekp(pos, ios::cur);
                        f.write((char *) &student, sizeof(Student));
                        cout << "\n\n** BOOK ISSUED SUCCESSFULLY! **\n\n";
                    }
                }
                if (!flag) {
                    cout << "\n\n** BOOK NUMBER DOES NOT EXIST! **\n\n";
                }
            }
            else {
                cout << "\n\n** YOU CAN ONLY ISSUE ONE BOOK AT A TIME! **\n\n";
            }
        }
    }
    if (!found) {
        cout << "\n\n** STUDENT DOESN'T EXIST! **\n\n";
    }
    system("pause");
    f.close();
    fil.close();
}

```

```

void return_book() {
    long bookNumber, admissionNumber;
    int found = 0, flag = 0;
    fstream fil;
    system("cls");
    cout << "\n\n\t\t\t** BOOK DEPOSIT **\n\n";
    cout << "\n\nENTER ADMISSION NUMBER: ";
    cin >> admissionNumber;
    fflush(stdin);
    f.open("students.dat", ios::in | ios::out);
    fil.open("books.dat", ios::in | ios::out);
    while (f.read((char *) &student, sizeof(Student))) {
        if (student.get_admission_number() == admissionNumber) {
            found = 1;
            if (student.get_issued_number() == 1) {
                while (fil.read((char *) &book, sizeof(Book))) {
                    if (student.get_issued_books() == book.get_book_number()) {
                        book.view_book();
                        flag = 1;
                        student.return_book();
                        int pos = -1 * sizeof(student);
                        f.seekp(pos, ios::cur);
                        f.write((char *) &student, sizeof(Student));
                        cout << "\n\n** BOOK DEPOSITED SUCCESSFULLY! **\n\n";
                    }
                }
                if (!flag) {
                    cout << "\n\n** BOOK NUMBER DOES NOT EXIST! **\n\n";
                }
            }
            else {
                cout << "\n\n** NO BOOKS ISSUED. PLEASE CHECK! **\n\n";
            }

        }
    }
    if (!found) {
        cout << "\n\n** STUDENT DOESN'T EXIST! **\n\n";
    }
    system("pause");
    f.close();
    fil.close();
}


void administrator() {
    int choice;
    cout << "\n\n\n\t\t\t** ADMINISTRATOR MENU **";
    cout << "\n\n\t1- ADD STUDENT";
    cout << "\n\n\t2- DISPLAY ALL STUDENTS";
    cout << "\n\n\t3- DISPLAY SPECIFIC STUDENT";
    cout << "\n\n\t4- UPDATE STUDENT RECORD";
    cout << "\n\n\t5- DELETE STUDENT";
    cout << "\n\n\t6- ADD BOOK";
    cout << "\n\n\t7- DISPLAY ALL BOOKS";
    cout << "\n\n\t8- DISPLAY SPECIFIC BOOK";
    cout << "\n\n\t9- UPDATE A BOOK";
    cout << "\n\n\t10- DELETE BOOK";
    cout << "\n\n\t11- BACK TO MAIN MENU";
    cout << "\n\n\tENTER YOUR CHOICE: ";
    cin >> choice;
    switch (choice) {
        case 1:
            system("cls");
            new_student();
            break;
        case 2:
            system("cls");
            student_list();
            break;
        case 3:
            long admissionNumber;
            system("cls");
            cout << "\n\n\t\t** STUDENT-DETAILS ** \n\n";
            cout << "\n\nADMISSION NUMBER: ";
            cin >> admissionNumber;
            fflush(stdin);
            display_student(admissionNumber);
            break;
        case 4:
            system("cls");
            update_student_record();
            break;
        case 5:
            system("cls");
            remove_student();
            break;
        case 6:
            system("cls");
            new_book();
            break;
        case 7:
            system("cls");
            book_list();
            break;
        case 8:
            long bookNumber;
            system("cls");
            cout << "\n\n\t\t** BOOK-DETAILS ** \n\n";
            cout << "\n\nBOOK NUMBER: ";
            cin >> bookNumber;
            fflush(stdin);
            display_book(bookNumber);
            break;
        case 9:
            system("cls");
            update_book_record();
            break;
        case 10:
            remove_book();
            break;
        case 11:
            return;
        default:
            cout << "\a";
    }
    system("cls");
    administrator();
}


int main() {
    char ch;
    do {
        system("cls");
        cout << "\n\n\t\t\t** LIBRARY MANAGEMENT SYSTEM **\n\n";
        cout << "\n\n** MAIN MENU **";
        cout << "\n\n\t1- ISSUE BOOK";
        cout << "\n\n\t2- RETURN A BOOK";
        cout << "\n\n\t3- ADMINISTRATOR\'S MENU";
        cout << "\n\n\t4- EXIT";
        cout << "\n\n\tCHOICE: ";
        cin >> ch;
        fflush(stdin);
        switch (ch) {
            case '1':
                system("cls");
                issue_book();
                break;
            case '2':
                system("cls");
                return_book();
                break;
            case '3':
                system("cls");
                administrator();
                break;
            case '4':
                exit(0);
            default :
                cout << "\a";
        }
    } while (ch != '4');
    return 0;
}
```