**LIBRARY MANAGEMENT SYSTEM**

**A PROJECT REPORT**

***Submitted by***

**Dev Singh [RA2211031010147]**

**Nikhil Kumar [RA2211031010097]**

***Under the Guidance of***

**Mr. K. Manikandan**

**Assistant Professor, Dept of Network and Communication [NWC]**

***In partial satisfaction of the requirements for the degree of***

## **BACHELOR OF TECHNOLOGY**

**in**

**COMPUTER SCIENCE ENGINEERING**

**with specialization in IT**

## Logo, company name Description automatically generated

**DEPARTMENT OF NETWORKING AND COMMUNICATION**

# **COLLEGE OF ENGINEERING AND TECHNOLOGY**

**SRM INSTITUTE OF SCIENCE AND TECHNOLOGY**

**KATTANKULATHUR - 603203**

**MAY 2023**

Logo, company name

Description automatically generated

SRM INSTITUTION OF SCIENCE AND TECHNOLOGY

KATTANKULATHUR-603203

**BONAFIDE CERTIFICATE**

Certified that this Project Report titled **“UML for Library Management System”** is the Bonafide work done by **Dev Singh [RA2211031010147] & Nikhil Kumar [RA2211031010097]** who completed the project under my supervision. Certified further, that to the best of my knowledge the work reported herein does not form part of any other work.

| **SIGNATURE**  **Mr. K. Manikandan**  **OODP – Course Faculty**  **Assistant Professor**  **Department of NWC**  **SRMIST** | **SIGNATURE**  **Dr M. Annapurani. K**  **Head of the Department**  **Department of NWC**  **SRMIST** |
| --- | --- |

**TABLE OF CONTENTS**

|  |  |  |
| --- | --- | --- |
| **S.No** | **CONTENTS** | **PAGE NO** |
| 1. | Problem Statement | 5 |
| 2. | Modules Description | 6 |
| 3. | Diagrams | 7 |
|  | * 1. Use case Diagram | 8 |
|  | * 1. Class Diagram | 9 |
|  | * 1. Sequence Diagram | 10 |
|  | * 1. Collaboration Diagram | 11 |
|  | 3.5 State Chart Diagram | 12 |
|  | * 1. Activity Diagram | 13 |
|  | * 1. Package Diagram | 14 |
|  | 3.8 Component Diagram | 15 |
|  | 3.9 Deployment Diagram | 16 |
| 4. | Coding | 17 |
| 5. | Output | 54 |
| 6. | Conclusion and Results | 57 |
|  | References | 58 |

1. **PROBLEM STATEMENT**

The Library Management System is a software solution that automates the processes involved in managing a library. The current manual system used by the library is time-consuming and prone to errors, which affects the efficiency of the library operations. The library staff spends a significant amount of time on tasks such as book issuing, record-keeping, and generating reports. The system also lacks real-time access to information, making it difficult for users to check the availability of books or for staff to keep track of the status of books.

To address these challenges, a computerized Library Management System is needed that can automate the library processes, reduce errors, and provide real-time access to information. The system should allow users to search for books, place requests, check book availability, and borrow books. It should also provide staff with the ability to manage books, members. Overall, the system should be user-friendly, secure, and scalable to accommodate future growth.

The Library Management System project aims to develop and implement a comprehensive software solution that will enhance the efficiency of the library operations and improve the user experience for both staff and users.

1. **MODULE DESCRIPTION**

The modules for the Library Management System project could be:

**Book class:** This module defines the Book class and its properties such as title, author, publisher, year, and stock. It also provides methods to create a Book object and access its properties.

**Library class:** This module defines the Library class and its properties such as a vector of Book objects. It provides methods to add a book to the library, search for a book by title, borrow a book, and return a book.

**User interface:** This module provides the user interface for the library management system. It interacts with the user and calls the appropriate methods of the Library class based on the user's input.

**Database:** This module could be used to store and manage the data of the Library system, such as book information and user information. It could use a database management system or file-based storage to manage the data.

**Reporting:** This module could generate various reports based on the data stored in the system, such as books borrowed by a user, books available in the library, etc. It could provide these reports in a printable format or export them to a file.

1. **UML DIAGRAM**

Unified Modelling Language (UML) diagrams are visual representations used for modelling object-oriented software systems. UML diagrams provide a standardized way to visualize and communicate software designs, making them easier to understand and implement. There are two main types of UML diagrams: behavioral and structural. Structural UML diagrams describe the static structure of a system, including its components, classes, interfaces, and relationships between them. Behavioral UML diagrams describe the dynamic behavior of a system, including the interactions between objects and the changes in their states.

**Structural UML Diagrams include:**

1. Class Diagrams

2. Object Diagrams

3. Component Diagrams

4. Deployment Diagrams

5. Package Diagrams

**Behavioral UML Diagrams include:**

1. Use Case Diagrams

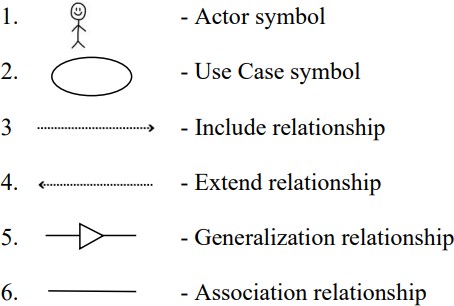
2. Sequence Diagrams

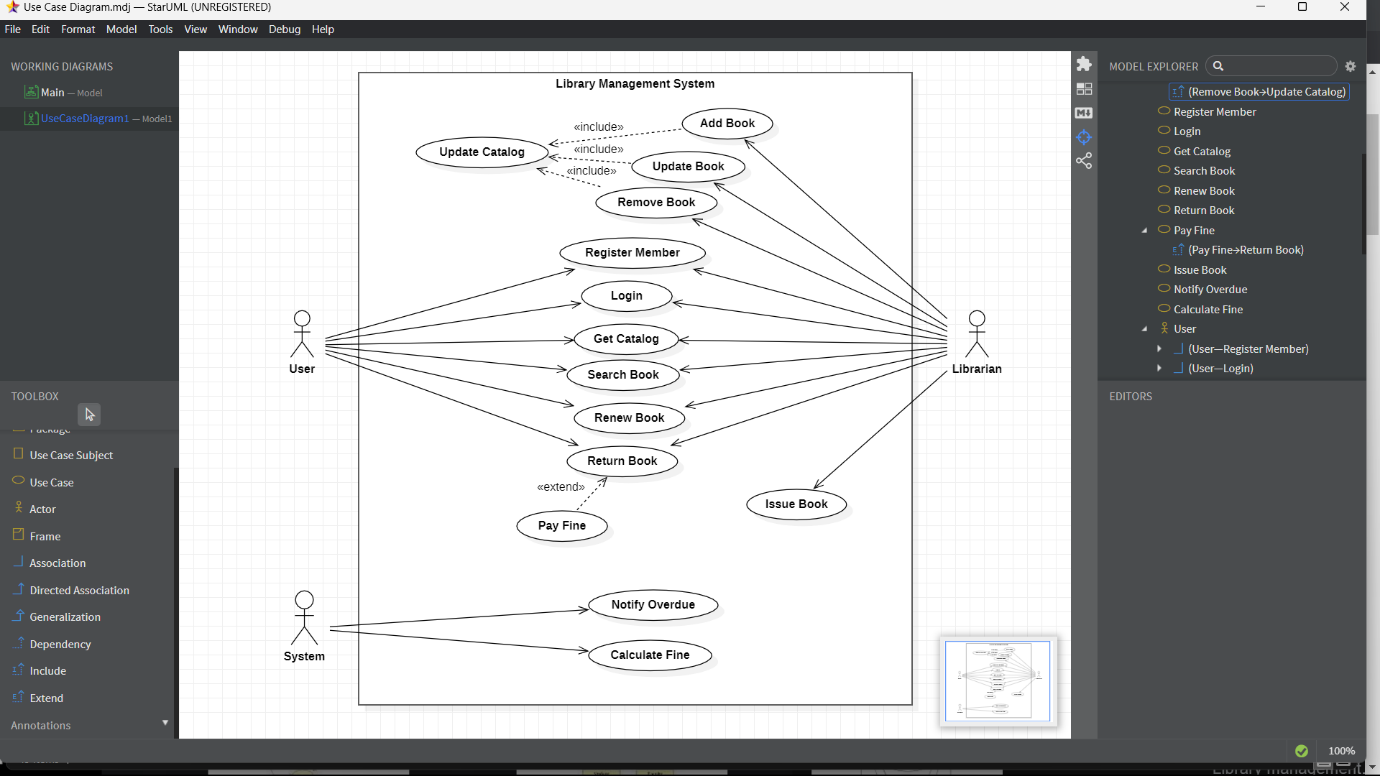
3. Collaboration Diagrams

4. State Diagrams

5. Activity Diagrams

**3.1 USE CASE DIAGRAM**

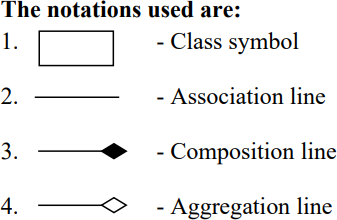
 The [use case diagram](https://itsourcecode.com/uml/student-information-system-use-case-diagram/) represents the main processes in Library management system. Then they will be broken down into more specific use cases depending on the included processes of the main use case. Each of these use cases explains how the system handles the actions or scenarios requested by the user.

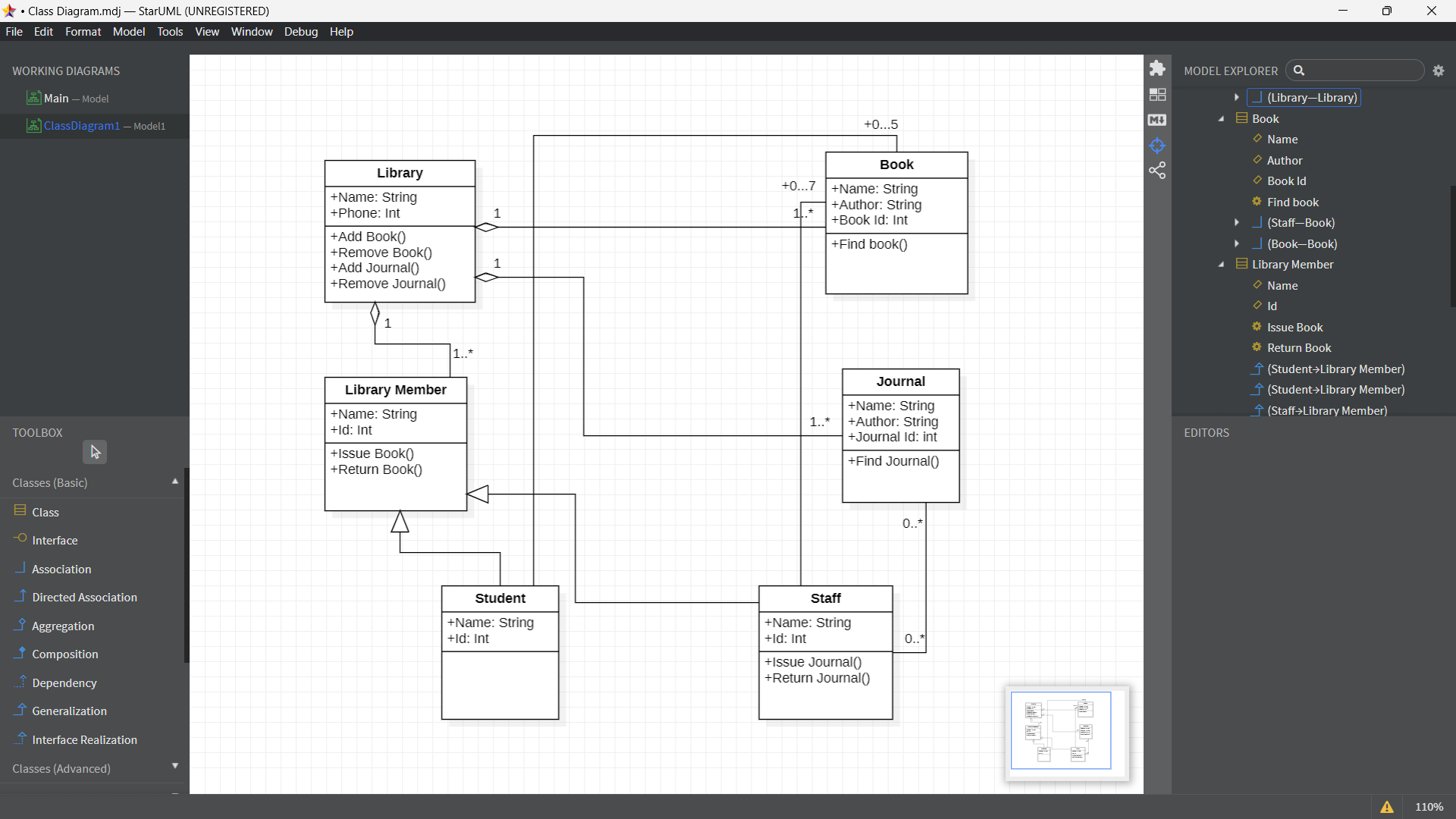


The UML Use Case Diagram is a design used as one of the Methodology on Library Management System development. It represents the main functions or processes of the system as well as the specific processes included. They were also labelled properly to guide programmers and users about the structure of Library Management System.

In this UML Use Case Diagram we have taken three actors a user, librarian, and system.

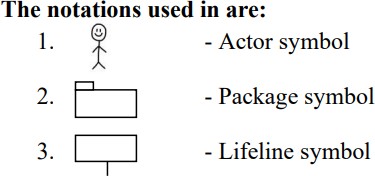
**3.2 CLASS DIAGRAM**

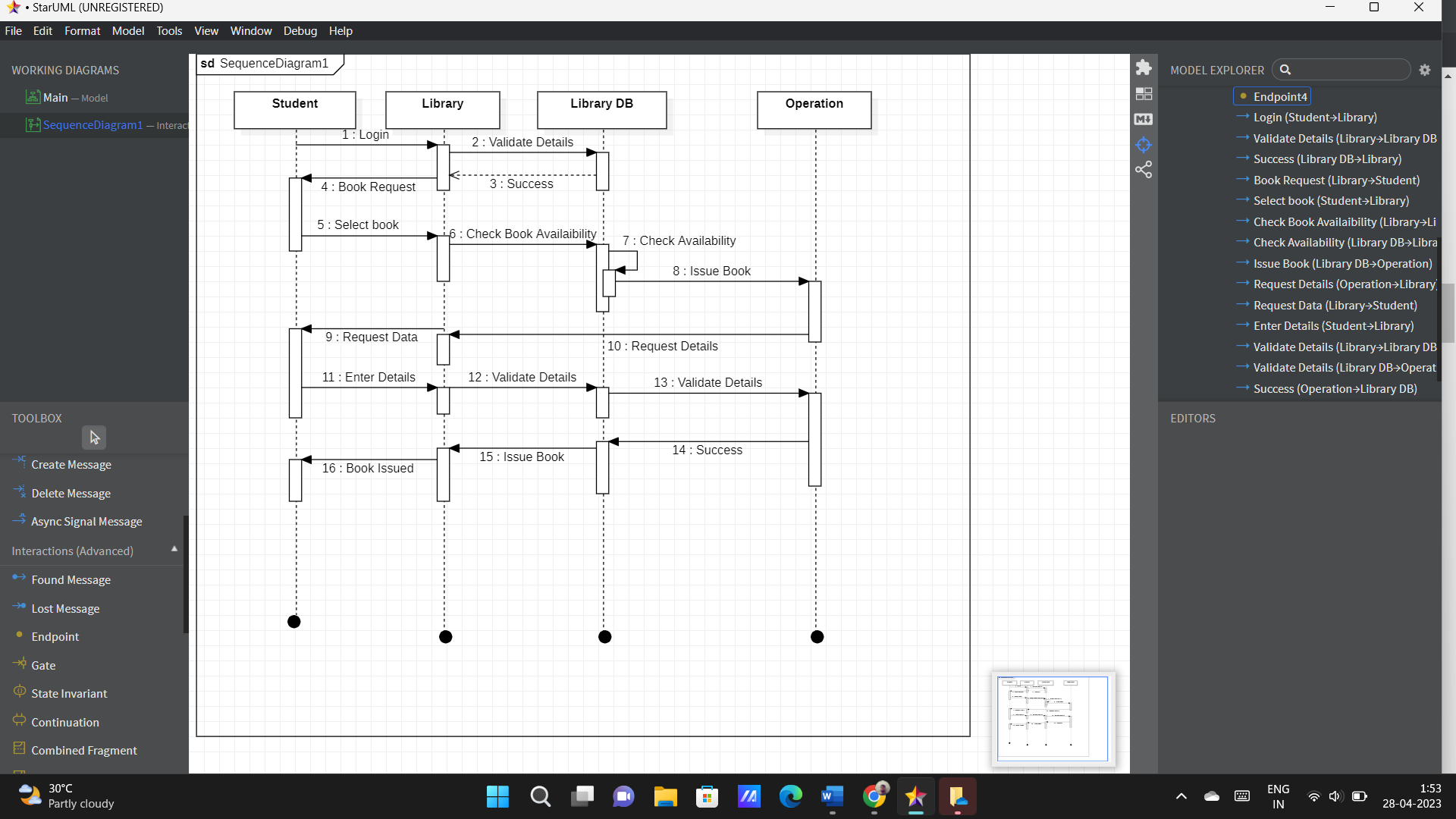
 The [Class diagram for Library Management System](https://itsourcecode.com/uml/student-management-system-class-diagram-uml/) shows the structures of information or data that will be handled in the system. These data or information will be represented by classes. Each of the classes will have their attributes in accord to the methods they will use.



So the classes that must be made in a Library Management System are the **Library**, **Library Member**, **Books**, **Journal**, **Student** and **Staff**. The mentioned classes were just general. If you want more complex or wider scope of your Library management system, then you can add your desired classes.

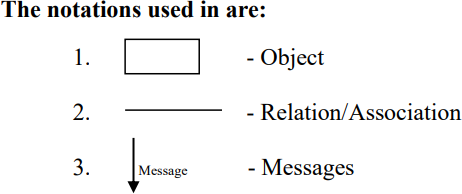
**3.3 SEQUENCE DIAGRAM**

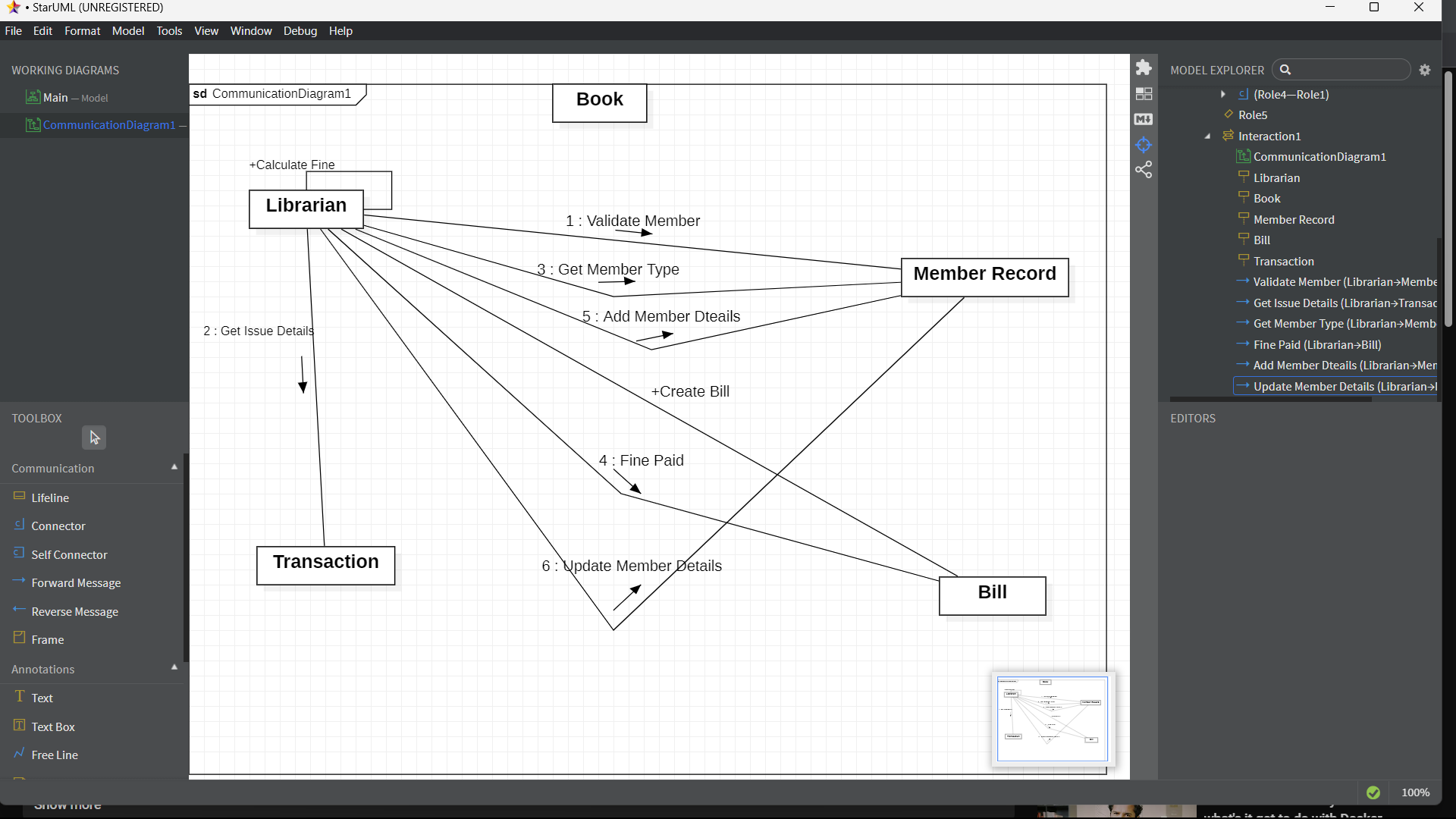
The designed sequence diagram illustrates the series of events that occurs in Library Management System. In this illustration, the actors are represented by a stick man and the transactions or classes are represented by objects. It will give you clear explanation about the behaviour of a Library Management System in terms of processing the flow of instructions.



The Sequence Diagram for Library Management System represents the scenario and the messages that must be passed between objects. This is done for the scenario’s functionality to be realized. It’s an interaction diagram that shows how activities are carried out, including when and how messages are sent.

**3.4 COLLABORATION DIAGRAM**

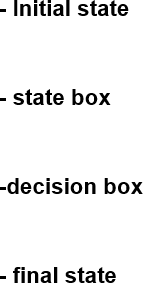
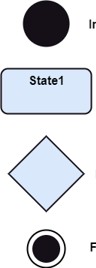
A collaboration diagram, also known as a communication diagram, is an illustration of the relationships and interactions among software objects in the Unified Modelling Language (UML). Developers can use these diagrams to portray the dynamic behaviour of a particular use case and define the role of each object.

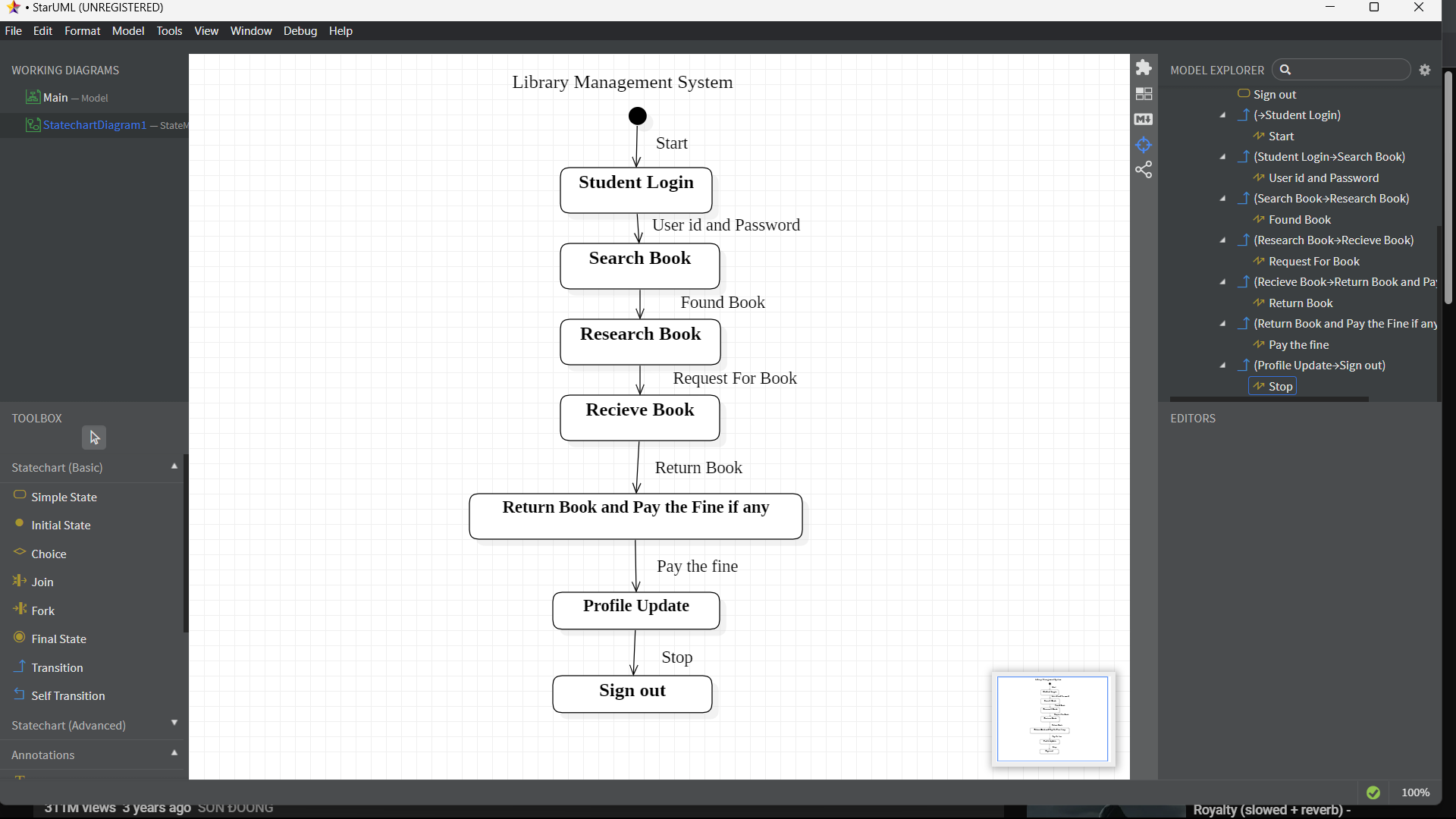


A collaboration diagram shows interactions between objects in a system. For example, it can show how a student searches for a book, the system retrieves book information from the database server, how the librarian assigns the book to the student, and how the student borrows the book. The diagram displays the communication between different system components through method calls, messages, or signals. It helps identify potential issues and ensures the system is efficient, effective, and user-friendly.

**3.5 STATE CHART DIAGRAM**

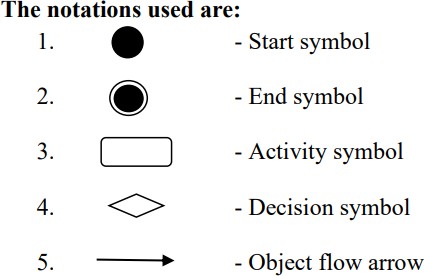
A state diagram is used to represent the condition of the system or part of the system at finite instances of time. It’s a behavioural diagram and it represents the behaviour using finite state transitions. State diagrams are also referred to as State machines and State-chart Diagrams. These terms are often used interchangeably.

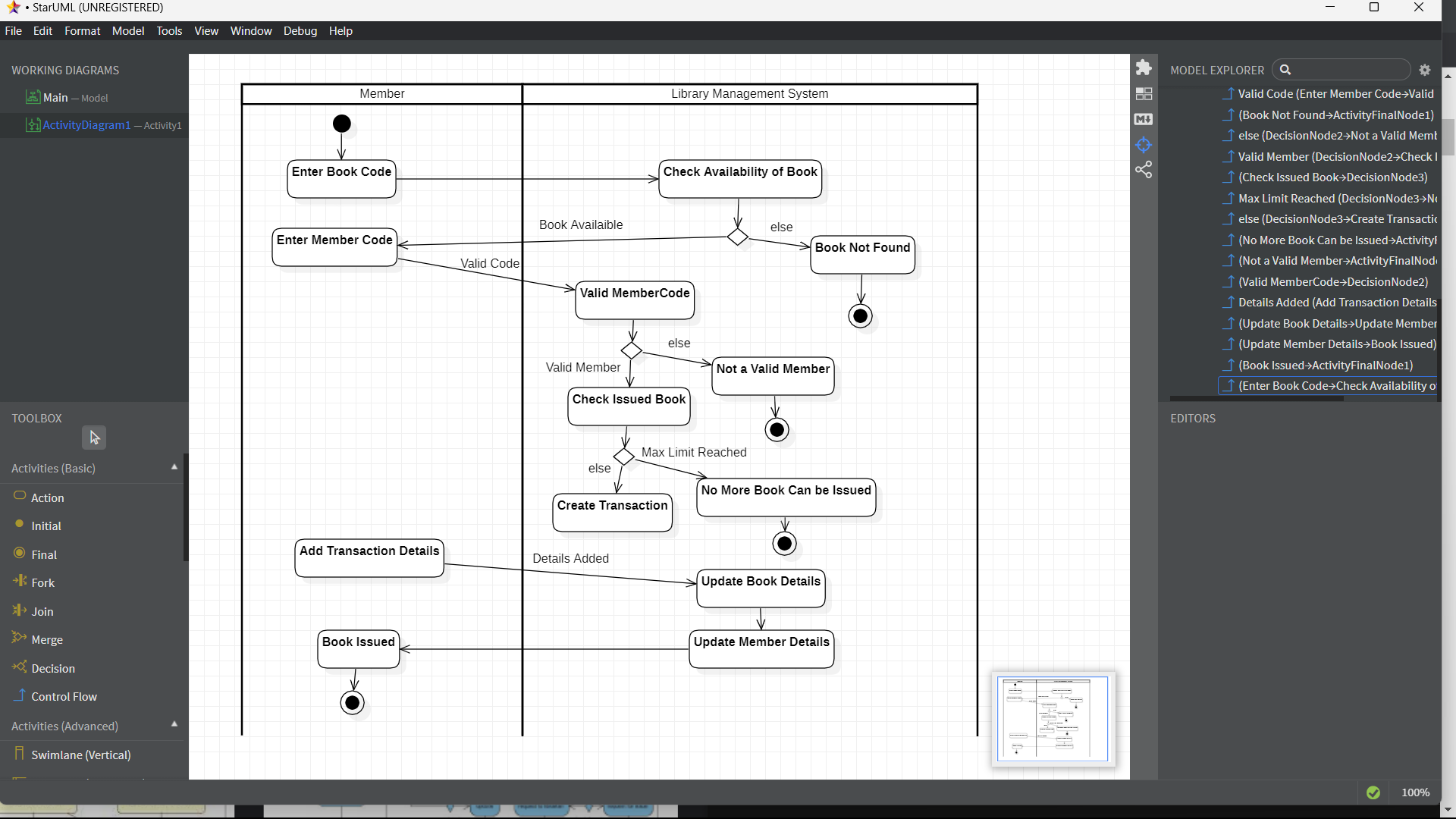




So simply, a state diagram is used to model the dynamic behaviour of a class in response to time and changing external stimuli. We can say that each and every class has a state but we don’t model every class using State diagrams. We prefer to model the states with three or more states.

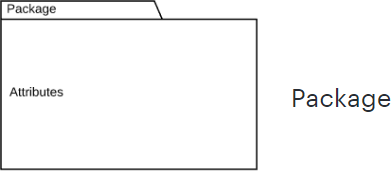
**3.6 ACTIVITY DIAGRAM**

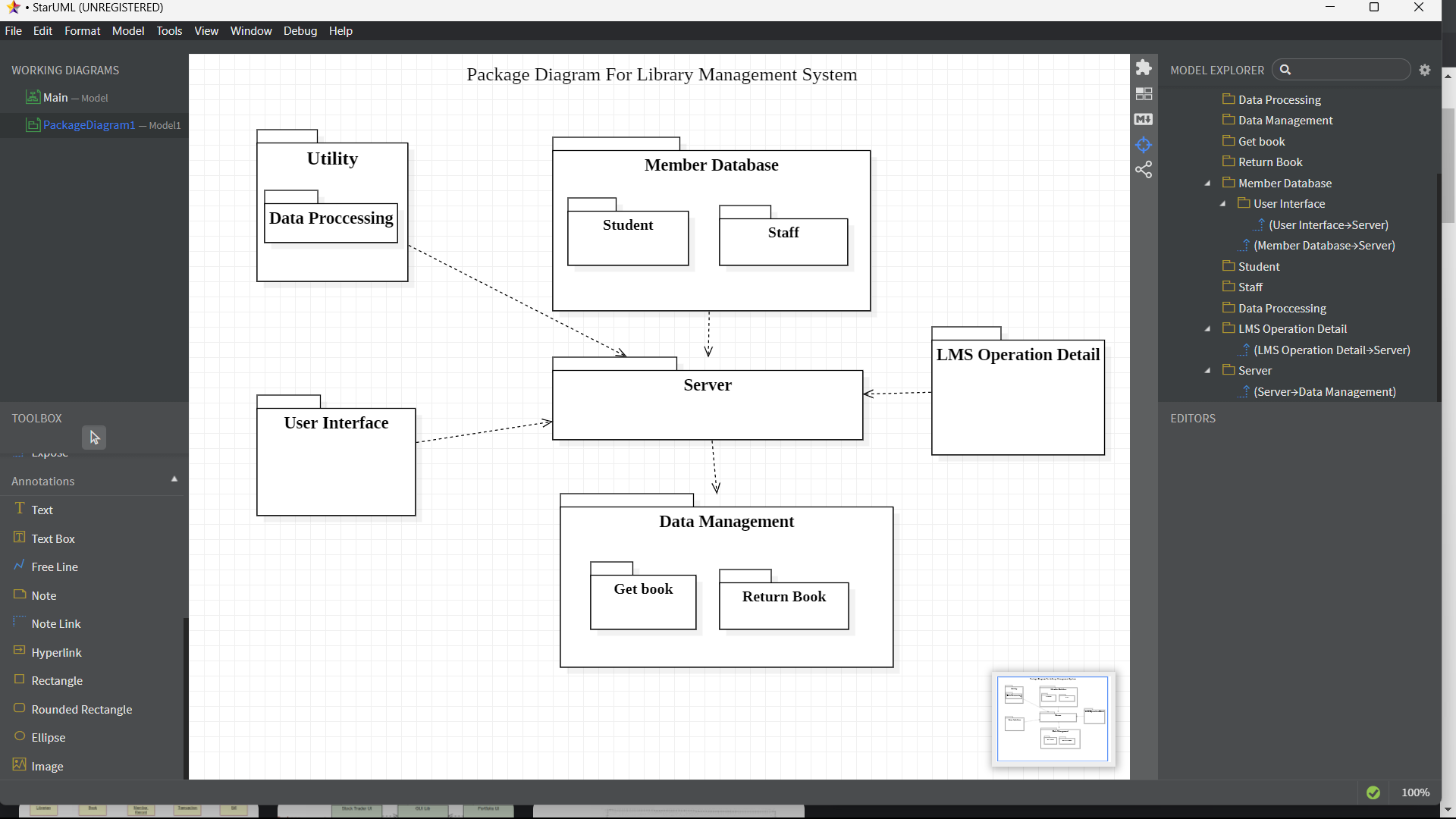
 In UML, an activity diagram provides a view of the behaviour of a system by describing the sequence of actions in a process. Activity diagrams are similar to flowcharts because they show the flow between the actions in an activity; however, activity diagrams can also show parallel or concurrent flows and alternate flows.



Activity diagram for library management system which shows the flows between the activity of members, librarian, library system and books.

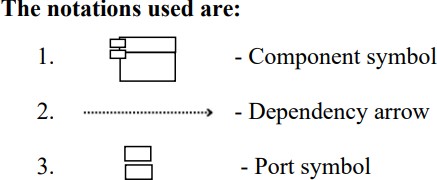
**3.7 PACKAGE DIAGRAM**

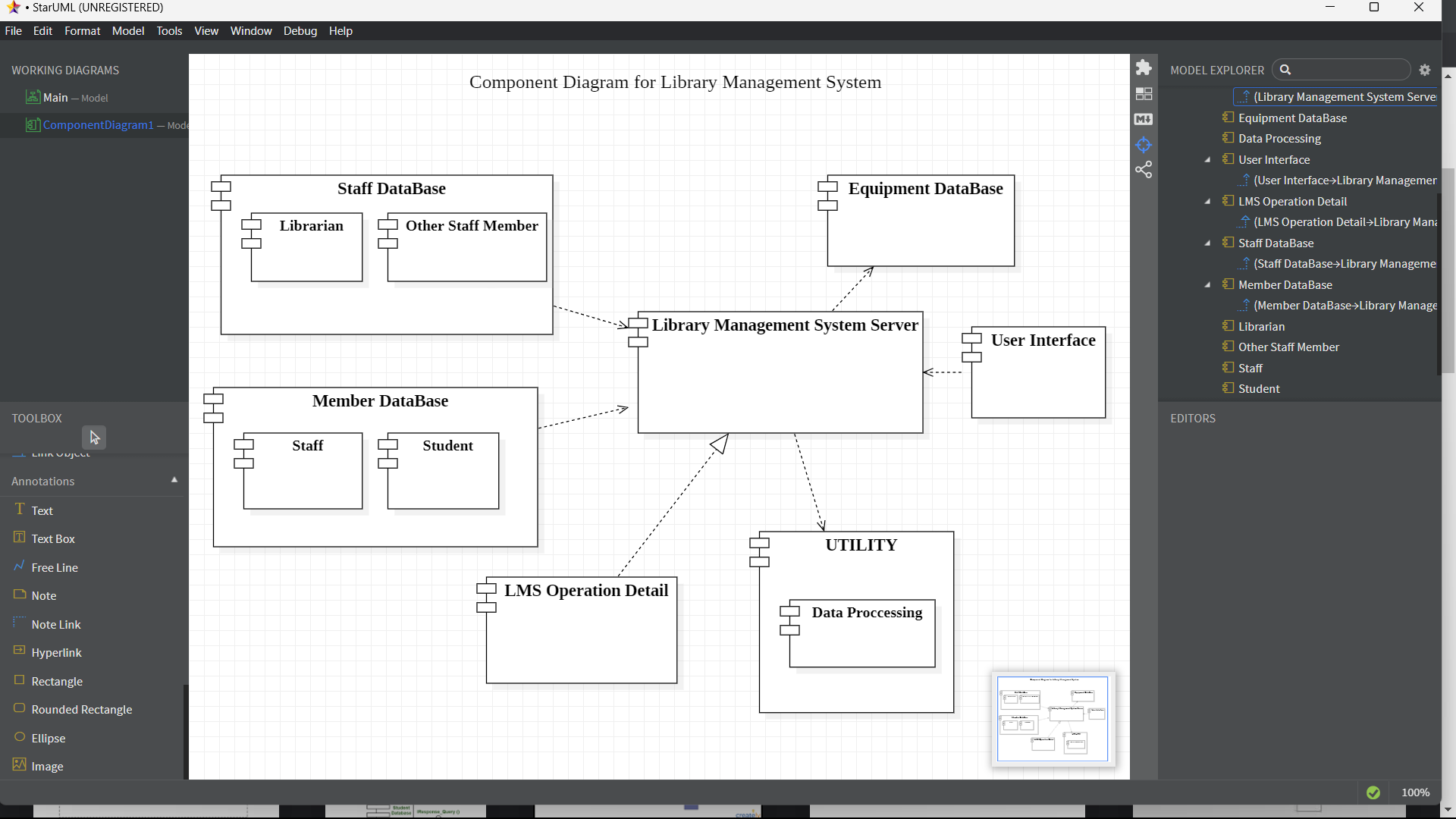
A package is a group of related classes, interfaces, and other elements that are used together to provide a specific functionality or service. A package can also contain other sub-packages.



A package diagram for a library management system might include packages for the user interface, the database access layer, the library services, reporting, security, and configuration. Each of these packages would contain classes and other elements that are related to their specific area of functionality.

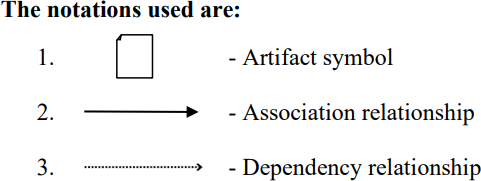
**3.8 COMPONENT DIAGRAM**

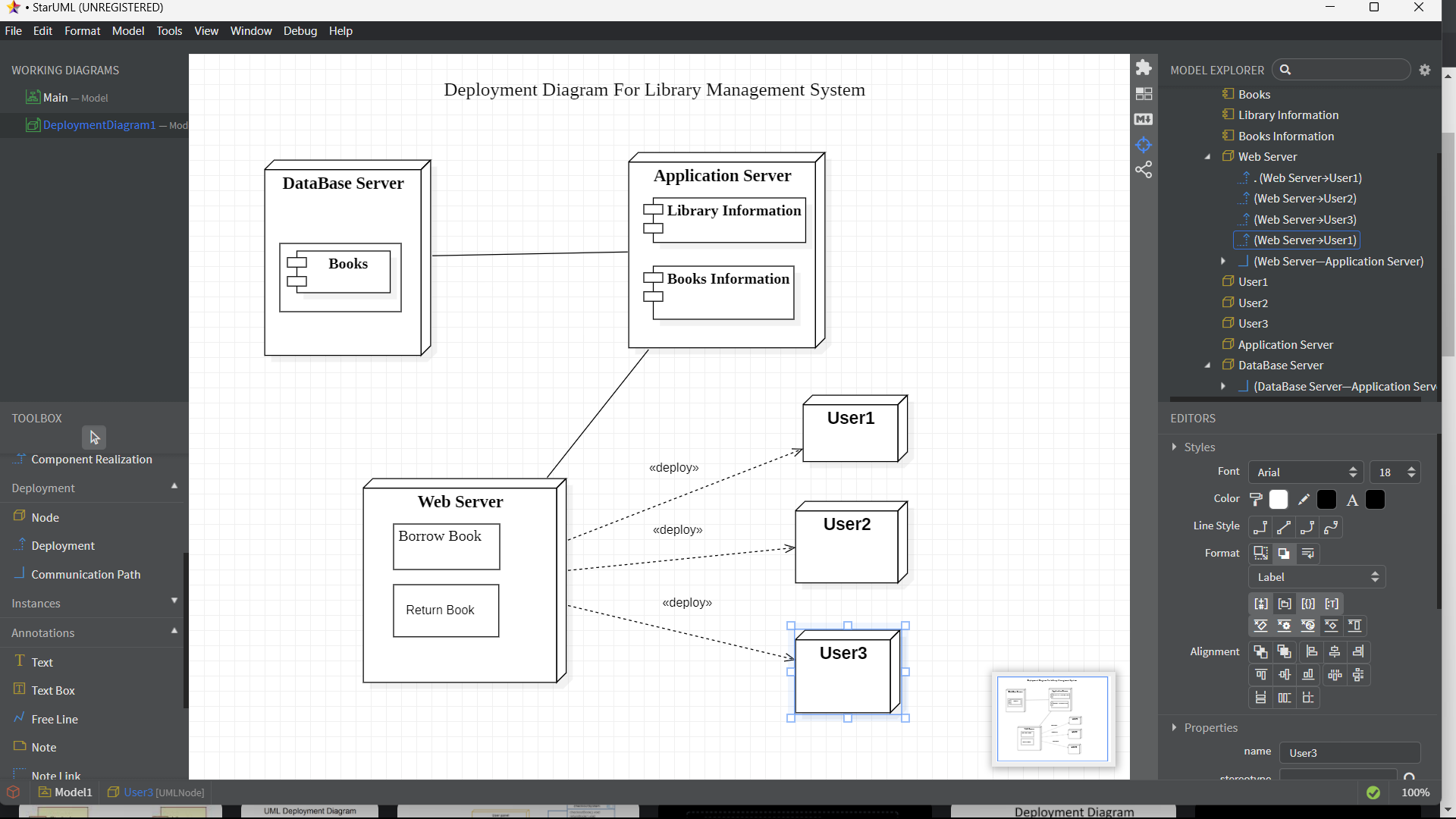
A component diagram is a type of UML diagram that shows the organization of components and their relationships in a software system. A component is a modular, self-contained unit of software that performs a specific function or service. Components can be implemented as classes, libraries, executables, or other software artifacts.

. 

A component diagram for a library management system might include components for the user interface, the database access layer, the library services, reporting, security, and configuration. Each of these components would have well-defined interfaces and would interact with other components through these interfaces.

**3.9 DEPLOYEMENT DIAGRAM**

A deployment diagram is a UML diagram type of the system that represents the execution architecture of the components of a system of the objects, including nodes or modes such as hardware or software execution environments or worlds, and the middleware connecting them. Diagram types mostly outline the logical components of a system.



A deployment diagram in library management system shows how system components are connected in a physical environment. It illustrates the connection between servers, routers, databases, and web browsers. The diagram also displays the communication protocols such as TCP/IP, HTTP, and FTP. It helps ensure the correct design and efficient deployment of the system.

1. **CODING**

//Learnprogramo

#include<iostream>

#include<stdio.h>

#include<stdlib.h>

#include<fstream>

#include<string.h>

#include<conio.h>

using namespace std;

class Lib

{

public:

char bookname[100],auname[50],sc[20],sc1[50];

int q,B,p;

Lib()

{

strcpy(bookname,"NO Book Name");

strcpy(auname,"No Author Name");

strcpy(sc,"No Book ID");

strcpy(sc1,"No Book ID");

q=0;

B=0;

p=0;

}

void get();

void student();

void pass();

void librarian();

void password();

void getdata();

void show(int);

void booklist(int);

void modify();

void see(int);

int branch(int);

void issue();

void der(char[],int,int);

void fine(int,int,int,int,int,int);

};

void Lib::getdata()

{

int i;

fflush(stdin);

cout<<"\n\t\tEnter the details :-\n";

cout<<"\n\t\tEnter Book's Name : ";

cin.getline(bookname,100);

for(i=0;bookname[i]!='\0';i++)

{

if(bookname[i]>='a'&&bookname[i]<='z')

bookname[i]-=32;

}

cout<<"\n\t\tEnter Author's Name : ";

cin.getline(auname,50);

cout<<"\n\t\tEnter Publication name : ";

cin.getline(sc1,50);

cout<<"\n\t\tEnter Book's ID : ";

cin.getline(sc,20);

cout<<"\n\t\tEnter Book's Price : ";

cin>>p;

cout<<"\n\t\tEnter Book's Quantity : ";

cin>>q;

}

void Lib::show(int i)

{

cout<<"\n\t\tBook Name : "<<bookname<<endl;

cout<<"\n\t\tBook's Author Name : "<<auname<<endl;

cout<<"\n\t\tBook's ID : "<<sc<<endl;

cout<<"\n\t\tBook's Publication : "<<sc1<<endl;

if(i==2)

{

cout<<"\n\t\tBook's Price : "<<p<<endl;

cout<<"\n\t\tBook's Quantity : "<<q<<endl;

}

}

void Lib::booklist(int i)

{

int b,r=0;

system("cls");

b=branch(i);

system("cls");

ifstream intf("Booksdata.txt",ios::binary);

if(!intf)

cout<<"\n\t\tFile Not Found.";

else

{

cout<<"\n\t \*\*\*\*\*\*\*\*\*\*\*\* Book List \*\*\*\*\*\*\*\*\*\*\*\* \n\n";

intf.read((char\*)this,sizeof(\*this));

while(!intf.eof())

{

if(b==B)

{

if(q==0 && i==1)

{

}

else

{

r++;

cout<<"\n\t\t\*\*\*\*\*\*\*\*\*\* "<<r<<". \*\*\*\*\*\*\*\*\*\* \n";

show(i);

}

}

intf.read((char\*)this,sizeof(\*this));

}

}

cout<<"\n\t\tPress any key to continue.....";

getch();

system("cls");

if(i==1)

student();

else

librarian();

}

void Lib::modify()

{

char ch,st1[100];

int i=0,b,cont=0;

system("cls");

cout<<"\n\t\t>>Please Choose one option :-\n";

cout<<"\n\t\t1.Modification In Current Books\n\n\t\t2.Add New Book\n\n\t\t3.Delete A Book\n\n\t\t4.Go back\n";

cout<<"\n\n\t\tEnter your choice : ";

cin>>i;

if(i==1)

{

system("cls");

b=branch(2);

ifstream intf1("Booksdata.txt",ios::binary);

if(!intf1)

{

cout<<"\n\t\tFile Not Found\n";

cout<<"\n\t\tPress any key to continue.....";

getch();

system("cls");

librarian();

}

intf1.close();

system("cls");

cout<<"\n\t\tPlease Choose One Option :-\n";

cout<<"\n\t\t1.Search By Book Name\n\n\t\t2.Search By Book's ID\n";

cout<<"\n\t\tEnter Your Choice : ";

cin>>i;

fflush(stdin);

if(i==1)

{

system("cls");

cout<<"\n\t\tEnter Book Name : ";

cin.getline(st1,100);

system("cls");

fstream intf("Booksdata.txt",ios::in|ios::out|ios::ate|ios::binary);

intf.seekg(0);

intf.read((char\*)this,sizeof(\*this));

while(!intf.eof())

{

for(i=0;b==B&&bookname[i]!='\0'&&st1[i]!='\0'&&(st1[i]==bookname[i]||st1[i]==bookname[i]+32);i++);

if(bookname[i]=='\0'&&st1[i]=='\0')

{

cont++;

getdata();

intf.seekp(intf.tellp()-sizeof(\*this));

intf.write((char\*)this,sizeof(\*this));

break;

}

intf.read((char\*)this,sizeof(\*this));

}

intf.close();

}

else if(i==2)

{

cout<<"\n\t\tEnter Book's ID : ";

cin.getline(st1,100);

system("cls");

fstream intf("Booksdata.txt",ios::in|ios::out|ios::ate|ios::binary);

intf.seekg(0);

intf.read((char\*)this,sizeof(\*this));

while(!intf.eof())

{

for(i=0;b==B&&sc[i]!='\0'&&st1[i]!='\0'&&st1[i]==sc[i];i++);

if(sc[i]=='\0'&&st1[i]=='\0')

{

cont++;

getdata();

intf.seekp(intf.tellp()-sizeof(\*this));

intf.write((char\*)this,sizeof(\*this));

break;

}

intf.read((char\*)this,sizeof(\*this));

}

intf.close();

}

else

{

cout<<"\n\t\tIncorrect Input.....:(\n";

cout<<"\n\t\tPress any key to continue.....";

getch();

system("cls");

modify();

}

if(cont==0)

{

cout<<"\n\t\tBook Not Found.\n";

cout<<"\n\t\tPress any key to continue.....";

getch();

system("cls");

modify();

}

else

cout<<"\n\t\tUpdate Successful.\n";

}

else if(i==2)

{

system("cls");

B=branch(2);

system("cls");

getdata();

ofstream outf("Booksdata.txt",ios::app|ios::binary);

outf.write((char\*)this,sizeof(\*this));

outf.close();

cout<<"\n\t\tBook added Successfully.\n";

}

else if(i==3)

{

system("cls");

b=branch(2);

ifstream intf1("Booksdata.txt",ios::binary);

if(!intf1)

{

cout<<"\n\t\tFile Not Found\n";

cout<<"\n\t\tPress any key to continue.....";

getch();

intf1.close();

system("cls");

librarian();

}

intf1.close();

system("cls");

cout<<"\n\t\tPlease Choose One Option for deletion:-\n";

cout<<"\n\t\t1.By Book Name\n\n\t\t2.By Book's ID\n";

cout<<"\n\t\tEnter Your Choice : ";

cin>>i;

fflush(stdin);

if(i==1)

{

system("cls");

cout<<"\n\t\tEnter Book Name : ";

cin.getline(st1,100);

ofstream outf("temp.txt",ios::app|ios::binary);

ifstream intf("Booksdata.txt",ios::binary);

intf.read((char\*)this,sizeof(\*this));

while(!intf.eof())

{

for(i=0;b==B&&bookname[i]!='\0'&&st1[i]!='\0'&&(st1[i]==bookname[i]||st1[i]==bookname[i]+32);i++);

if(bookname[i]=='\0'&&st1[i]=='\0')

{

cont++;

intf.read((char\*)this,sizeof(\*this));

}

else

{

outf.write((char\*)this,sizeof(\*this));

intf.read((char\*)this,sizeof(\*this));

}

}

intf.close();

outf.close();

remove("Booksdata.txt");

rename("temp.txt","Booksdata.txt");

}

else if(i==2)

{

cout<<"\n\t\tEnter Book's ID : ";

cin.getline(st1,100);

ofstream outf("temp.txt",ios::app|ios::binary);

ifstream intf("Booksdata.txt",ios::binary);

intf.read((char\*)this,sizeof(\*this));

while(!intf.eof())

{

for(i=0;b==B&&sc[i]!='\0'&&st1[i]!='\0'&&st1[i]==sc[i];i++);

if(sc[i]=='\0'&&st1[i]=='\0')

{

cont++;

intf.read((char\*)this,sizeof(\*this));

}

else

{

outf.write((char\*)this,sizeof(\*this));

intf.read((char\*)this,sizeof(\*this));

}

}

outf.close();

intf.close();

remove("Booksdata.txt");

rename("temp.txt","Booksdata.txt");

}

else

{

cout<<"\n\t\tIncorrect Input.....:(\n";

cout<<"\n\t\tPress any key to continue.....";

getch();

system("cls");

modify();

}

if(cont==0)

{

cout<<"\n\t\tBook Not Found.\n";

cout<<"\n\t\tPress any key to continue.....";

getch();

system("cls");

modify();

}

else

cout<<"\n\t\tDeletion Successful.\n";

}

else if(i==4)

{

system("cls");

librarian();

}

else

{

cout<<"\n\t\tWrong Input.\n";

cout<<"\n\t\tPress any key to continue.....";

getch();

system("cls");

modify();

}

cout<<"\n\t\tPress any key to continue.....";

getch();

system("cls");

librarian();

}

int Lib::branch(int x)

{

int i;

cout<<"\n\t\t>>Please Choose one Branch :-\n";

cout<<"\n\t\t1.Class 12th\n\n\t\t2.CS\n\n\t\t3.EC\n\n\t\t4.CIVIL\n\n\t\t5.MECHANICAL\n\n\t\t6.1ST YEAR\n\n\t\t7.Go to menu\n";

cout<<"\n\t\tEnter youur choice : ";

cin>>i;

switch(i)

{

case 1: return 1;

break;

case 2: return 2;

break;

case 3: return 3;

break;

case 4: return 4;

break;

case 5: return 5;

break;

case 6: return 6;

break;

case 7: system("cls");

if(x==1)

student();

else

librarian();

default : cout<<"\n\t\tPlease enter correct option :(";

getch();

system("cls");

branch(x);

}

}

void Lib::see(int x)

{

int i,b,cont=0;

char ch[100];

system("cls");

b=branch(x);

ifstream intf("Booksdata.txt",ios::binary);

if(!intf)

{

cout<<"\n\t\tFile Not Found.\n";

cout<<"\n\t\t->Press any key to continue.....";

getch();

system("cls");

if(x==1)

student();

else

librarian();

}

system("cls");

cout<<"\n\t\tPlease Choose one option :-\n";

cout<<"\n\t\t1.Search By Name\n\n\t\t2.Search By Book's ID\n";

cout<<"\n\t\tEnter Your Choice : ";

cin>>i;

fflush(stdin);

intf.read((char\*)this,sizeof(\*this));

if(i==1)

{

cout<<"\n\t\tEnter Book's Name : ";

cin.getline(ch,100);

system("cls");

while(!intf.eof())

{

for(i=0;b==B&&q!=0&&bookname[i]!='\0'&&ch[i]!='\0'&&(ch[i]==bookname[i]||ch[i]==bookname[i]+32);i++);

if(bookname[i]=='\0'&&ch[i]=='\0')

{

cout<<"\n\t\tBook Found :-\n";

show(x);

cont++;

break;

}

intf.read((char\*)this,sizeof(\*this));

}

}

else if(i==2)

{

cout<<"\n\t\tEnter Book's ID : ";

cin.getline(ch,100);

system("cls");

while(!intf.eof())

{

for(i=0;b==B&&q!=0&&sc[i]!='\0'&&ch[i]!='\0'&&ch[i]==sc[i];i++);

if(sc[i]=='\0'&&ch[i]=='\0')

{

cout<<"\n\t\tBook Found :-\n";

show(x);

cont++;

break;

}

intf.read((char\*)this,sizeof(\*this));

}

}

else

{

cont++;

cout<<"\n\t\tPlease enter correct option :(";

getch();

system("cls");

see(x);

}

intf.close();

if(cont==0)

cout<<"\n\t\tThis Book is not available :( \n";

cout<<"\n\t\tPress any key to continue.....";

getch();

system("cls");

if(x==1)

student();

else

librarian();

}

void Lib::issue()

{

char st[50],st1[20];

int b,i,j,d,m,y,dd,mm,yy,cont=0;

system("cls");

cout<<"\n\t\t->Please Choose one option :-\n";

cout<<"\n\t\t1.Issue Book\n\n\t\t2.View Issued Book\n\n\t\t3.Search student who isuued books\n\n\t\t4.Reissue Book\n\n\t\t5.Return Book\n\n\t\t6.Go back to menu\n\n\t\tEnter Your Choice : ";

cin>>i;

fflush(stdin);

if(i==1)

{

system("cls");

b=branch(2);

system("cls");

fflush(stdin);

cout<<"\n\t\t->Please Enter Details :-\n";

cout<<"\n\t\tEnter Book Name : ";

cin.getline(bookname,100);

cout<<"\n\t\tEnter Book's ID : ";

cin.getline(sc,20);

//strcpy(st,sc);

der(sc,b,1);

cout<<"\n\t\tEnter Student Name : ";

cin.getline(auname,100);

cout<<"\n\t\tEnter Student's ID : ";

cin.getline(sc1,20);

cout<<"\n\t\tEnter date : ";

cin>>q>>B>>p;

ofstream outf("student.txt",ios::binary|ios::app);

outf.write((char\*)this,sizeof(\*this));

outf.close();

cout<<"\n\n\t\tIssue Successfully.\n";

}

else if(i==2)

{

ifstream intf("student.txt",ios::binary);

system("cls");

cout<<"\n\t\t->The Details are :-\n";

intf.read((char\*)this,sizeof(\*this));

i=0;

while(!intf.eof())

{

i++;

cout<<"\n\t\t\*\*\*\*\*\*\*\*\*\* "<<i<<". \*\*\*\*\*\*\*\*\*\* \n";

cout<<"\n\t\tStudent Name : "<<auname<<"\n\t\t"<<"Student's ID : "<<sc1<<"\n\t\t"<<"Book Name : "<<bookname<<"\n\t\t"<<"Book's ID : "<<sc<<"\n\t\t"<<"Date : "<<q<<"/"<<B<<"/"<<p<<"\n";

intf.read((char\*)this,sizeof(\*this));

}

intf.close();

}

else if(i==3)

{

system("cls");

fflush(stdin);

cout<<"\n\t\t->Please Enter Details :-\n";

cout<<"\n\n\t\tEnter Student Name : ";

cin.getline(st,50);

cout<<"\n\n\t\tEnter Student's ID : ";

cin.getline(st1,20);

system("cls");

ifstream intf("student.txt",ios::binary);

intf.read((char\*)this,sizeof(\*this));

cont=0;

while(!intf.eof())

{

for(i=0;sc1[i]!='\0'&&st1[i]!='\0'&&st1[i]==sc1[i];i++);

if(sc1[i]=='\0'&&st1[i]=='\0')

{

cont++;

if(cont==1)

{

cout<<"\n\t\t->The Details are :-\n";

cout<<"\n\t\tStudent Name : "<<auname;

cout<<"\n\t\tStudent's ID : "<<sc1;

}

cout<<"\n\n\t\t\*\*\*\*\*\*\* "<<cont<<". Book details \*\*\*\*\*\*\*\n";

cout<<"\n\t\tBook Name : "<<bookname;

cout<<"\n\t\tBook's ID : "<<sc;

cout<<"\n\t\tDate : "<<q<<"/"<<B<<"/"<<p<<"\n";

}

intf.read((char\*)this,sizeof(\*this));

}

intf.close();

if(cont==0)

cout<<"\n\t\tNo record found.";

}

else if(i==4)

{

system("cls");

fflush(stdin);

cout<<"\n\t\t->Please Enter Details :-\n";

cout<<"\n\n\t\tEnter Student's ID : ";

cin.getline(st,50);

cout<<"\n\t\tEnter Book's ID : ";

cin.getline(st1,20);

fstream intf("student.txt",ios::in|ios::out|ios::ate|ios::binary);

intf.seekg(0);

intf.read((char\*)this,sizeof(\*this));

while(!intf.eof())

{

for(i=0;sc[i]!='\0'&&st1[i]!='\0'&&st1[i]==sc[i];i++);

for(j=0;sc1[j]!='\0'&&st[j]!='\0'&&st[j]==sc1[j];j++);

if(sc[i]=='\0'&&sc1[j]=='\0'&&st[j]=='\0'&&st1[i]=='\0')

{

d=q;

m=B;

y=p;

cout<<"\n\t\tEnter New Date : ";

cin>>q>>B>>p;

fine(d,m,y,q,B,p); //fn1

intf.seekp(intf.tellp()-sizeof(\*this)); //fn3

intf.write((char\*)this,sizeof(\*this)); //fn5

cout<<"\n\n\t\tReissue successfully."; //fn3

break;

}

intf.read((char\*)this,sizeof(\*this));

}

intf.close();

}

else if(i==5)

{

system("cls");

b=branch(2);

system("cls");

fflush(stdin);

cout<<"\n\t\t->Please Enter Details :-\n";

cout<<"\n\t\tEnter Book's ID : ";

cin.getline(st1,20);

der(st1,b,2);

cout<<"\n\n\t\tEnter Student's ID : ";

cin.getline(st,20);

cout<<"\n\t\tEnter Present date : ";

cin>>d>>m>>y;

ofstream outf("temp.txt",ios::app|ios::binary);

ifstream intf("student.txt",ios::binary);

intf.read((char\*)this,sizeof(\*this));

while(!intf.eof())

{

for(i=0;sc[i]!='\0'&&st1[i]!='\0'&&st1[i]==sc[i];i++);

for(j=0;sc1[j]!='\0'&&st[j]!='\0'&&st[j]==sc1[j];j++);

if(sc[i]=='\0'&&sc1[j]=='\0'&&st[j]=='\0'&&st1[i]=='\0'&&cont==0)

{

cont++;

intf.read((char\*)this,sizeof(\*this));

fine(q,B,p,d,m,y);

cout<<"\n\t\tReturned successfully.";

}

else

{

outf.write((char\*)this,sizeof(\*this));

intf.read((char\*)this,sizeof(\*this));

}

}

intf.close();

outf.close();

getch();

remove("student.txt");

rename("temp.txt","student.txt");

}

else if(i==6)

{

system("cls");

librarian();

}

else

cout<<"\n\t\tWrong Input.\n";

cout<<"\n\n\t\tPress any key to continue.....";

getch();

system("cls");

librarian();

}

void Lib::fine(int d,int m,int y,int dd,int mm,int yy)

{

long int n1,n2;

int years,l,i;

const int monthDays[12] = {31, 28, 31, 30, 31, 30,31, 31, 30, 31, 30, 31};

n1 = y\*365 + d;

for (i=0; i<m - 1; i++)

n1 += monthDays[i]; //fn1353

years = y;

if (m <= 2)

years--;

l= years / 4 - years / 100 + years / 400;

n1 += l;

n2 = yy\*365 + dd;

for (i=0; i<mm - 1; i++)

n2 += monthDays[i];

years = yy;

if (m <= 2)

years--;

l= years / 4 - years / 100 + years / 400;

n2 += l;

n1=n2-n1;

n2=n1-15;

if(n2>0)

cout<<"\n\t\tThe Total Fine is : "<<n2;

}

void Lib::der(char st[],int b,int x)

{

int i,cont=0;

fstream intf("Booksdata.txt",ios::in|ios::out|ios::ate|ios::binary);

intf.seekg(0);

intf.read((char\*)this,sizeof(\*this));

while(!intf.eof())

{

for(i=0;b==B&&sc[i]!='\0'&&st[i]!='\0'&&st[i]==sc[i];i++);

if(sc[i]=='\0'&&st[i]=='\0')

{

cont++;

if(x==1)

{

q--;

}

else

{

q++;

}

intf.seekp(intf.tellp()-sizeof(\*this));

intf.write((char\*)this,sizeof(\*this));

break;

}

intf.read((char\*)this,sizeof(\*this));

}

if(cont==0)

{

cout<<"\n\t\tBook not found.\n";

cout<<"\n\n\t\tPress any key to continue.....";

getch();

system("cls");

issue();

}

intf.close();

}

void Lib::get()

{

int i;

cout<<"\n\t\*\*\*\*\*\*\*\*\*\*\* LIBRARY MANAGEMENT SYSTEM \*\*\*\*\*\*\*\*\*\*\*\n"<<"\n\t\t\t Learnprogramo <<LMS>> C++\n";

cout<<"\n\t\t>>Please Choose Any Option To login \n";

cout<<"\n\t\t1.Student\n\n\t\t2.Librarian\n\n\t\t3.Close Application\n";

cout<<"\n\t\tEnter your choice : ";

cin>>i;

if(i==1)

{

system("cls");

student();

}

else if(i==2)

pass();

else if(i==3)

exit(0);

else

{

cout<<"\n\t\tPlease enter correct option :(";

getch();

system("CLS");

get();

}

}

void Lib::student()

{

int i;

cout<<"\n\t\*\*\*\*\*\*\*\*\*\*\*\* WELCOME STUDENT \*\*\*\*\*\*\*\*\*\*\*\*\n";

cout<<"\n\t\t>>Please Choose One Option:\n";

cout<<"\n\t\t1.View BookList\n\n\t\t2.Search for a Book\n\n\t\t3.Go to main menu\n\n\t\t4.Close Application\n";

cout<<"\n\t\tEnter your choice : ";

cin>>i;

if(i==1)

booklist(1);

else if(i==2)

see(1);

else if(i==3)

{

system("cls");

get();

}

else if(i==4)

exit(0);

else

{

cout<<"\n\t\tPlease enter correct option :(";

getch();

system("cls");

student();

}

}

void Lib::pass()

{

int i=0;

char ch,st[21],ch1[21]={"learnprogramo"};

cout<<"\n\t\tEnter Password : ";

while(1)

{

ch=getch();

if(ch==13)

{

st[i]='\0';

break;

}

else if(ch==8&&i>0)

{

i--;

cout<<"\b \b";

}

else

{

cout<<"\*";

st[i]=ch;

i++;

}

}

ifstream inf("password.txt");

inf>>ch1;

inf.close();

for(i=0;st[i]==ch1[i]&&st[i]!='\0'&&ch1[i]!='\0';i++);

if(st[i]=='\0'&&ch1[i]=='\0')

{

system("cls");

librarian();

}

else

{

cout<<"\n\n\t\tWrong Password.\n\n\t\ttry again.....\n";

getch();

system("cls");

get();

}

}

void Lib::librarian()

{

int i;

cout<<"\n\t\*\*\*\*\*\*\*\*\*\*\*\* WELCOME LIBRARIAN \*\*\*\*\*\*\*\*\*\*\*\*\n";

cout<<"\n\t\t>>Please Choose One Option:\n";

cout<<"\n\t\t1.View BookList\n\n\t\t2.Search for a Book\n\n\t\t3.Modify/Add Book\n\n\t\t4.Issue Book\n\n\t\t5.Go to main menu\n\n\t\t6.Change Password\n\n\t\t7.Close Application\n";

cout<<"\n\t\tEnter your choice : ";

cin>>i;

switch(i)

{

case 1:booklist(2);

break;

case 2:see(2);

break;

case 3:modify();

break;

case 4:issue();

break;

case 5:system("cls");

get();

break;

case 6:password();

break;

case 7:exit(0);

default:cout<<"\n\t\tPlease enter correct option :(";

getch();

system("cls");

librarian();

}

}

void Lib::password()

{

int i=0,j=0;

char ch,st[21],ch1[21]={"learnprogramo"};

system("cls");

cout<<"\n\n\t\tEnter Old Password : ";

while(1)

{

ch=getch();

if(ch==13)

{

st[i]='\0';

break;

}

else if(ch==8&&i>0)

{

i--;

cout<<"\b \b";

}

else

{

cout<<"\*";

st[i]=ch;

i++;

}

}

ifstream intf("password.txt");

intf>>ch1;

intf.close();

for(i=0;st[i]==ch1[i]&&st[i]!='\0'&&ch1[i]!='\0';i++);

if(st[i]=='\0'&&ch1[i]=='\0')

{

system("cls");

cout<<"\n\t\*\*The Password Should be less than 20 characters & don't use spaces\*\*\n\n";

cout<<"\n\t\tEnter New Password : ";

fflush(stdin);

i=0;

while(1)

{

j++;

ch=getch();

if(ch==13)

{

for(i=0;st[i]!=' '&&st[i]!='\0';i++);

if(j>20 || st[i]==' ')

{

cout<<"\n\n\t\tYou did't follow the instruction \n\n\t\tPress any key for try again.....";

getch();

system("cls");

password();

librarian();

}

st[i]='\0';

break;

}

else if(ch==8&&i>0)

{

i--;

cout<<"\b \b";

}

else

{

cout<<"\*";

st[i]=ch;

i++;

}

}

ofstream outf("password.txt");

outf<<st;

outf.close();

cout<<"\n\n\t\tYour Password has been changed Successfully.";

cout<<"\n\t\tPress any key to continue......";

getch();

system("cls");

librarian();

}

else

{

cout<<"\n\n\t\tPassword is incorrect.....\n";

cout<<"\n\t\tEnter 1 for retry or 2 for menu";

cin>>i;

if(i==1)

{

system("cls");

password();

}

else

{

system("cls");

librarian();

}

}

}

int main()

{

Lib obj;

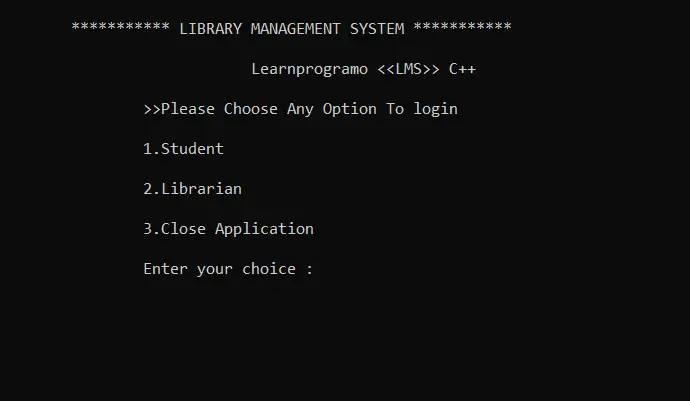
obj.get();

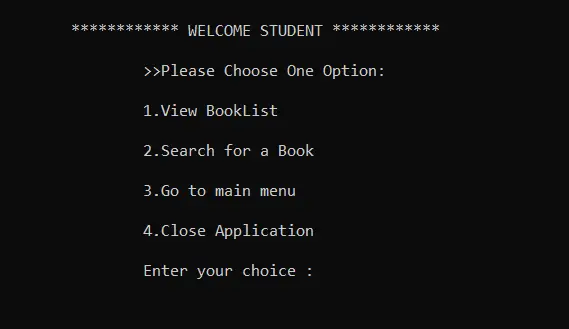
getch();

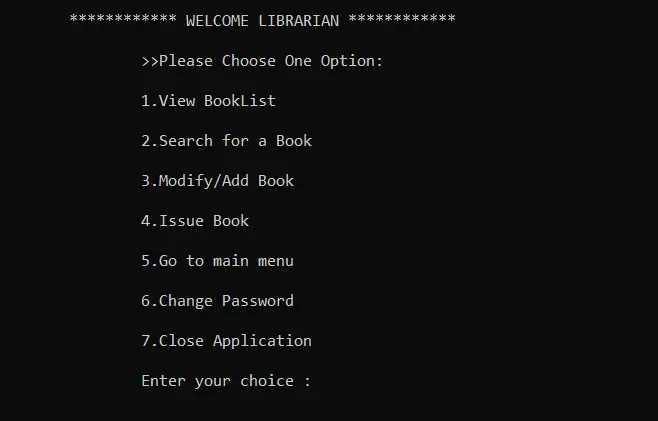
return 0;

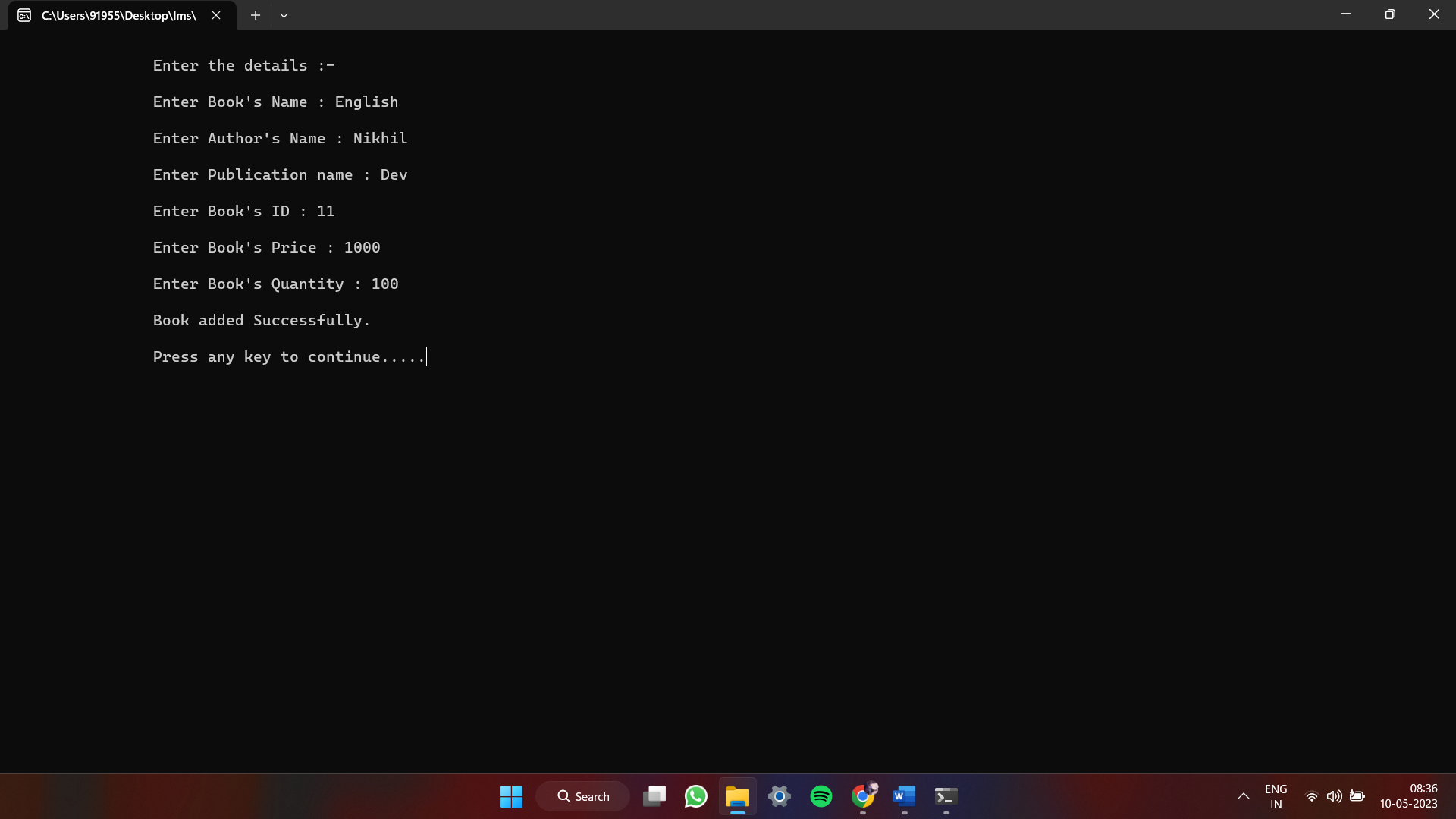
}

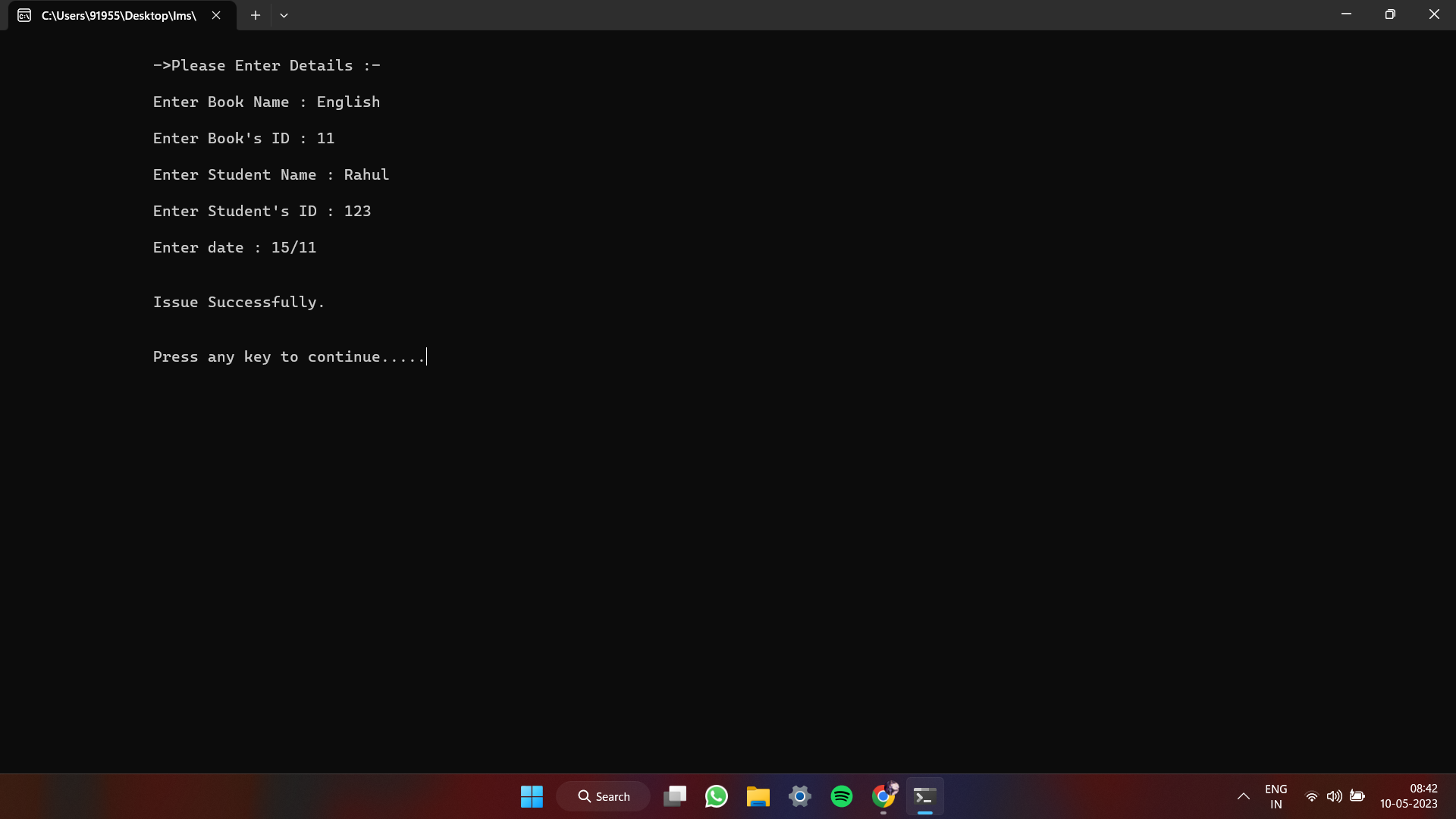
1. **Output**

****

****

****





1. **CONCLUSION/RESULT**

The Library Management System project developed in C++ is a comprehensive and user-friendly system that enables librarians to manage books, and students to access and borrow books from the library. With separate logins for librarians and students, the system provides a secure environment for managing library resources. The librarian has full control over the system, including adding, updating, deleting, and creating books, assigning books to students, changing the password, and modifying books. The use of file handling to store data ensures that the system can manage large volumes of books efficiently.

In addition, the system enables students to view the available books in the library and access information about the books, including which books are currently on loan. Reissuing and returning books are also essential features of this system, which ensures that the books are returned on time and the system maintains accurate records. Overall, the Library Management System project in C++ is an effective tool for managing library resources and promoting efficient operations.

**REFERENCE**

Here are some references we have used for our Library Management System project:

* **SRM UNIVERSITY LIBRARY**: We have noticed and reviewed our own library management system by issuing a book and taking collective information from the staff.
* **GOOGLE:** We have also used google for our information searching.
* **STAR UML:** We have used Star UML application for making UML diagram.