SCHOOL OF COMPUTING SCIENCE AND ENGINEERING

GALGOTIAS UNIVERSITY, GREATER NOIDA

UTTAR PRADESH

LAB FILE
SUBJECT – SOFTWARE ENGINEERING
COURSE CODE - BCSE2355

Submitted by Adnan Ahmad
Admission No. – 22SCSE1012033

Submitted to Mr. Ravi Sharma

INDEX

ASSIGNMENT- 1

Objective – Draw Entity-Relationship diagram for Library Management System

ER Diagram is known as Entity-Relationship Diagram, it is used to analyse the structure of the Database. It shows relationships between entities and their attributes. An ER Model provides a means of communication.

Step 1: The first step is to identify the entity sets. These are things that have more than one instance of being existent. The entity sets for a library management system are as follows:

- Book
- Publisher
- Member

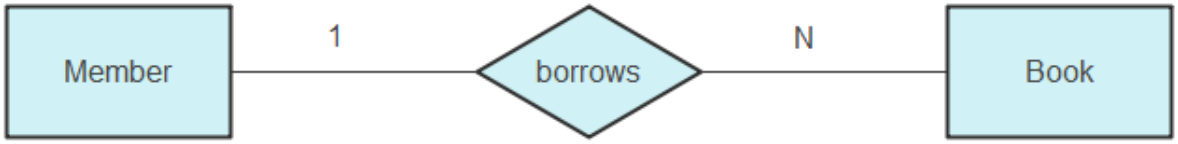
Step 2: The second step is to associate attributes to the entity sets. This is important because entity sets are recurring, such as several books, members, etc. The attributes are needed to differentiate between entities. The respective attributes are as follows:

- Member: Member ID, Name, Birthday, Address, Age
- Book: Author, ISBN, Title, Author, Price
- Publisher: PID, Name, Address, Phone

Step 3: The third step is to find the key attributes of each entity. The key attribute is unique to every single entity component. The key attributes are as follows:

- Member – Member ID
- Book – ISBN
- Publisher – PID

Step 4: The fourth step is to identify the relationship between the different entities present. This is necessary to analyse the database clearly and make a better library management system.



A single member can borrow multiple books if they want. This kind of relationship is known as a one-to-many relationship.

A single publisher can publish several books and supply them to the library. This is another one-to-many relationship.

Step 5: The fifth step is to compile all these steps. Join the entities that go together and form an ER diagram.

E-R Diagram of Library Management System

ASSIGNMENT- 2

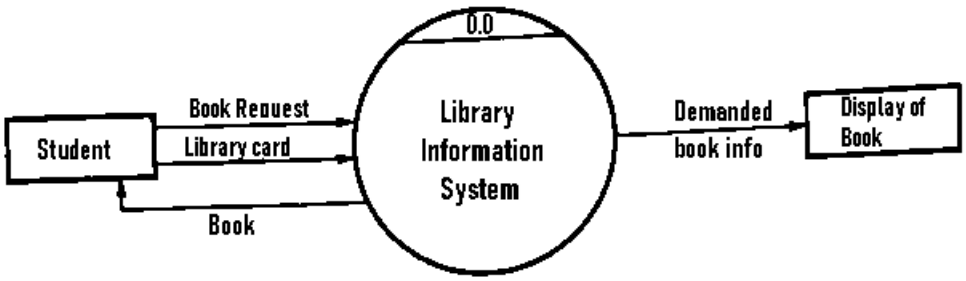
Objective – Draw a flow diagram for Library Management System

The overall processing unit will contain the following output that a system will produce or generate:

- Book will be the output as the book demanded by the student will be given to them.
- Information of demanded book should be displayed by the library information system that can be used by the student while selecting the book which makes it easier for the student.

Level 0 DFD: -

At this level the system has to show very simple and straight forward flow diagram of library management system it mainly contains a small diagram of or fractions of management system.



Level 1 DFD: -

At this level, the system has to show or exposed with more details of processing.
The processes that are important to be carried out are:

- Book delivery
- Search by topic

List of authors, List of Titles, List of Topics, the bookshelves from which books can be located are some information that is required for these processes. **Data store** is used to represent this type of information.

Level 2 DFD: -

Out of scope:

Other activities like purchasing of new books, replacement of old books or charging a fine are not considered in the above system.

ASSIGNMENT- 3

Objective – Draw an activity diagram for Library Management System

The activity diagram used to describe flow of activity through a series of actions. Activity diagram is an important diagram to describe the system. The activity described as an action or operation of the system.

Steps in creating an Activity Diagram for Library Management System.

Step 1: Familiarize Activity Diagram Symbols

First, Activity Diagram Symbols – are used to create an Activity Diagram which was presented here. Their symbols and applications must be familiarized before you build the Activity Diagram.

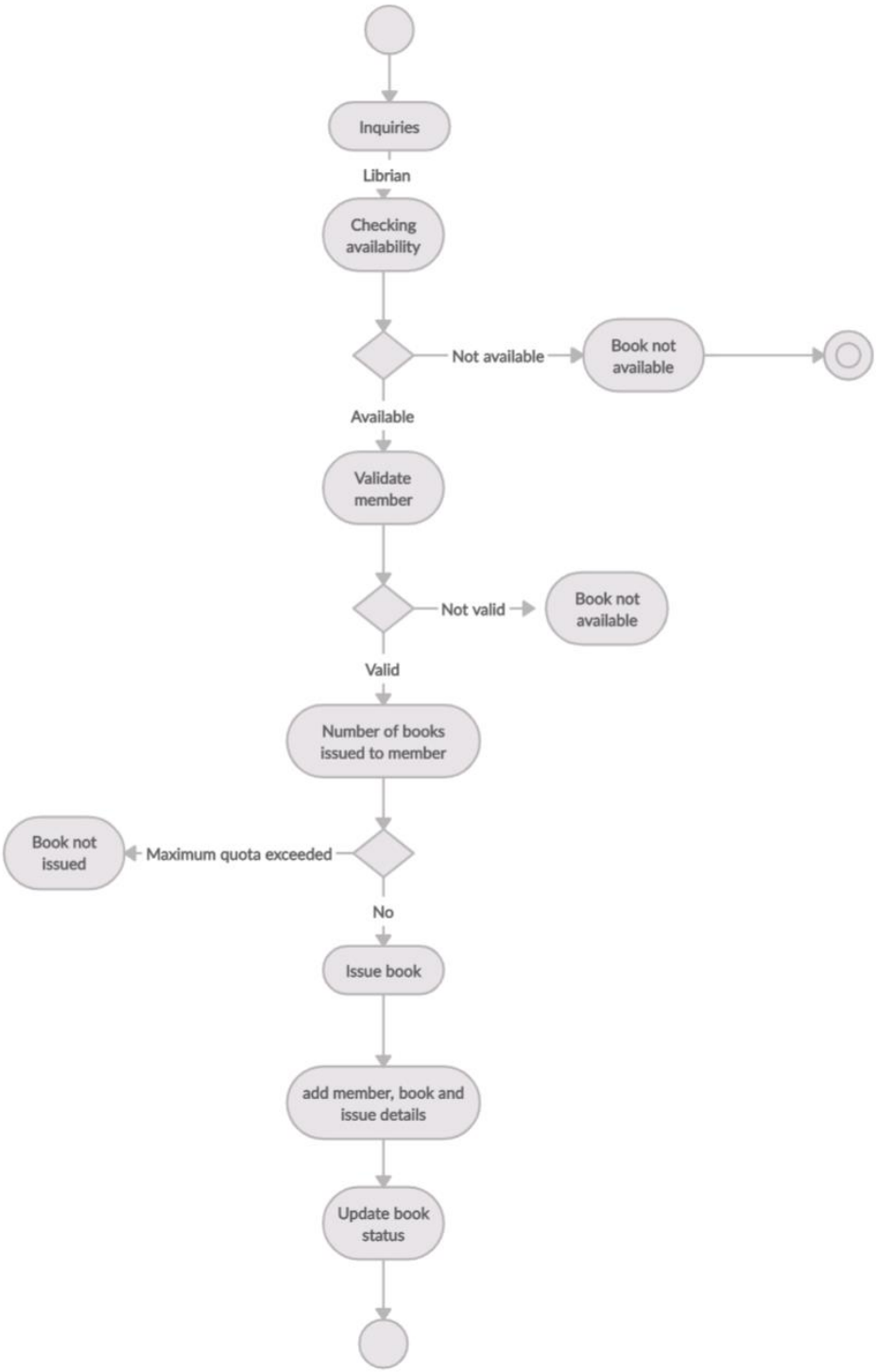
- **Start** – is the beginning of the action is symbolized by a filled circle. A dark circle represents the initial node.
- **Action** – is a stage of an activity in which users or software accomplish a certain task. Actions are represented by round-edged rectangles.
- **Decision** – represented by a diamond and is used to symbolize a conditional branch in the flow. It has two or more outputs and a single input.
- **Swim Lanes** – are the lanes drawn as boundaries, and the activities drawn in the same lane as the organization's activities. Swim lanes must be arranged in a logical order.
- **Activity Flow** – is another name for the connectors in the diagram that illustrate the flow between processes (activity).
- **End or Terminator** – is the last phase of the activity. An indicated black circle represents the final node.

Step 2: Determine the targeted users

Second, After the symbol familiarization, you'll need to determine your targeted users. Your targeted users will be the ones to use your project.

Step 3: Analyse the activities included

Third, analyzation is very important in creating an activity diagram. It will help you understand the work of the diagram and avoid unwanted errors.



Step 4: Plot the Activity Diagram

Finally, to plot the activity diagram you will need the users, activities, decisions, and their flows (paths). You will base the flow of activities on the evaluated information to have the exact Activity Diagram.

Conclusion:

You need to know the diagrams used to design and develop the Library Management System. This is due to the fact that without it, it is impossible to create a fully functional system.

But if you create this activity diagram, you will know the possible inputs and scenarios that the system should process and perform. Not only that, but you will identify the processes that are required and connect them to the other UML Diagrams.

ASSIGNMENT- 4

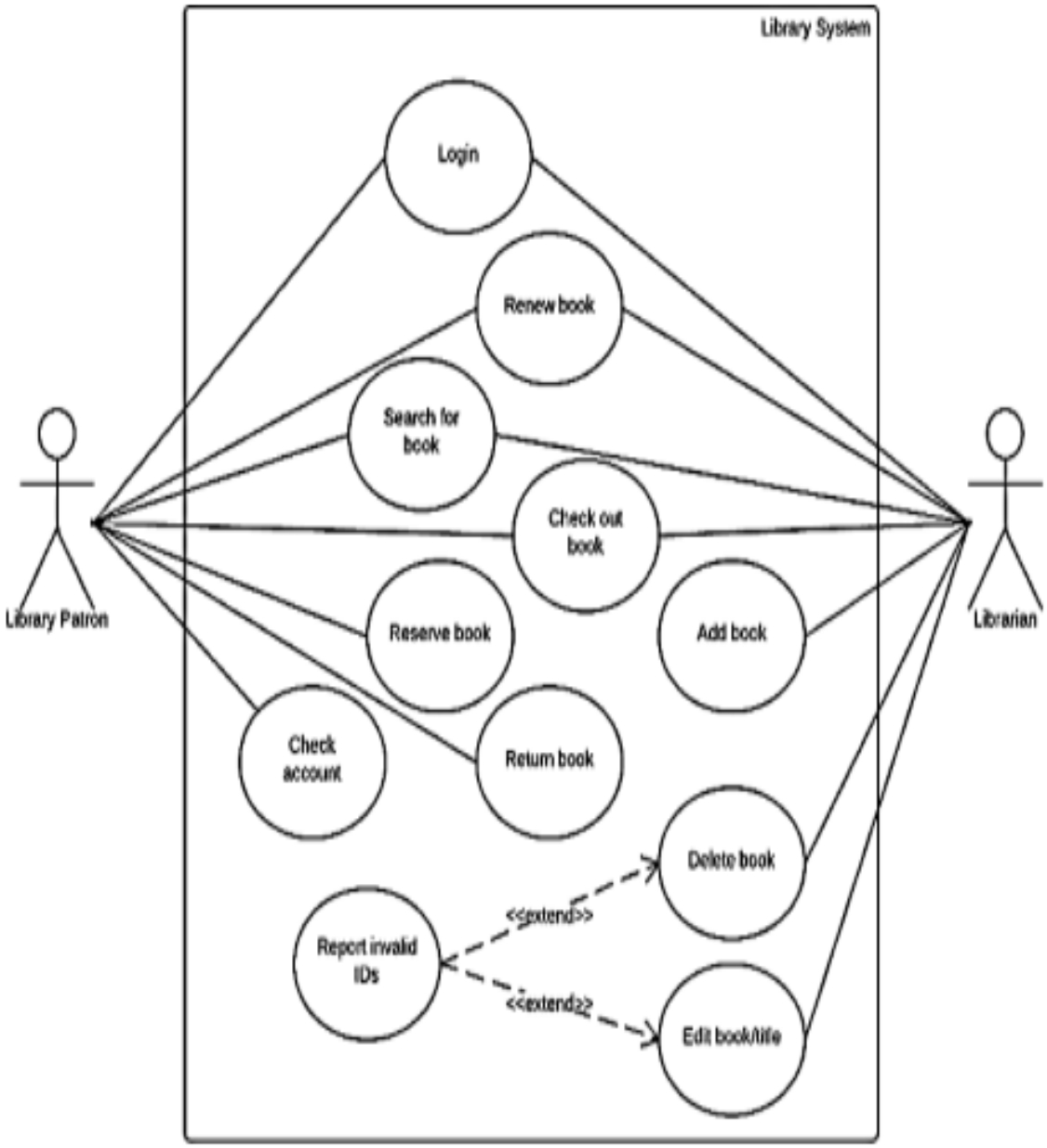
Objective – Draw a use case diagram for Library Management System

Use case diagrams referred as a Behaviour model or diagram. It simply describes and displays the relation or interaction between the users or customers and providers of application service or the system. It describes different actions that a system performs in collaboration to achieve something with one or more users of the system. Use case diagram is used a lot nowadays to manage the system.

Here, we will understand the designing use case diagram for the library management system. Some scenarios of the system are as follows:

- User who registers himself as a new user initially is regarded as staff or student for the library system.
- For the user to get registered as a new user, registration forms are available that is needed to be fulfilled by the user.
- After registration, a library card is issued to the user by the librarian. On the library card, an ID is assigned to cardholder or user.
- After getting the library card, a new book is requested by the user as per there requirement.
- After, requesting, the desired book or the requested book is reserved by the user that means no other user can request for that book.
- Now, the user can renew a book that means the user can get a new due date for the desired book if the user has renewed them.
- If the user somehow forgets to return the book before the due date, then the user pays fine. Or if the user forgets to renew the book till the due date, then the book will be overdue and the user pays fine.
- User can fill the feedback form available if they want to.
- Librarian has a key role in this system. Librarian adds the records in the library database about each student or user every time issuing the book or returning the book, or paying fine.

- Librarian also deletes the record of a particular student if the student leaves the college or passed out from the college. If the book no longer exists in the library, then the record of the particular book is also deleted.
- Updating database is the important role of Librarian.



ASSIGNMENT – 5

Objective - Write a C program to find whether the groups of quadratic equations are real, imaginary or equal

The standard form of a quadratic equation is:

ax2 + bx + c = 0, where

a, b and c are real numbers and

a != 0

The term $b^{2;}$ - 4ac is known as the **discriminant** of a quadratic equation. It tells the nature of the roots.

- If the discriminant is greater than 0, the roots are real and different.
- If the discriminant is equal to 0, the roots are real and equal.
- If the discriminant is less than 0, the roots are complex and different.

Program to Find Roots of a Quadratic Equation

```c
#include <math.h>
#include <stdio.h>
int main() {
    double a, b, c, discriminant, root1, root2, realPart, imagPart;
    printf("Enter coefficients a, b and c: ");
    scanf("%lf %lf %lf", &a, &b, &c);

    discriminant = b * b - 4 * a * c;

    // condition for real and different roots
    if (discriminant > 0) {
        root1 = (-b + sqrt(discriminant)) / (2 * a);
        root2 = (-b - sqrt(discriminant)) / (2 * a);
        printf("root1 = %.2lf and root2 = %.2lf", root1, root2);
    }

    // condition for real and equal roots
    else if (discriminant == 0) {
        root1 = root2 = -b / (2 * a);
        printf("root1 = root2 = %.2lf;", root1);
    }
```

```
    // if roots are not real
    else {
        realPart = -b / (2 * a);
        imagPart = sqrt(-discriminant) / (2 * a);
        printf("root1 = %.2lf+%.2lfi and root2 = %.2f-%.2fi", realPart,
imagPart, realPart, imagPart);
    }

    return 0;
}
```

Output: -

Enter coefficients a, b and c: 2.3

4

5.6

root1 = -0.87+1.30i and root2 = -0.87-1.30i

In this program, the sqrt() library function is used to find the square root of a number.

ASSIGNMENT- 6

Objective - Perform boundary value analysis on a C program to find whether the groups of quadratic equations are real, imaginary or equal

A quadratic equation is an equation which must be in the form of ax2+bx+c where a can't be 0. We use Quadratic formula to find roots and check whether the roots are real or imaginary.

$$\frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$$

This Quadratic formula is applied only when b2— 4ac >= 0.

such that,

If b2 - 4ac > 0, means the eqn. has more than one real roots.

if b2 - 4ac = 0, represent equal or single root.

if b2 - 4ac < 0, represents imaginary root.

Lastly, if a is 0, then the equation would not be considered as a quadratic equation.

So, a quadratic equation can have [Real, Equal, Imaginary, not quadratic]

We are supposing interval [0, 10] where our input values will fall in between this interval and we will create test cases using Boundary Value Analysis accordingly.

Program in C

```
#include <conio.h>
#include <stdio.h>
#include <math.h>
```

```
void main()
{
    float a, b, c, result;
    printf(" ax^2 + bx + c, \n enter the values of a, b and c : = ");
    scanf("%f %f %f", &a, &b, &c);

    result = (b * b) - (4 * a * c);
    if (a == 0)
        printf("not quadratic");
    else if (result > 0)
    {
        result = sqrt(result);
        printf("Real roots are, %f,%f \n", (-b - result) / (2 * a), (-b +
result) / (2 * a));
    }
    else if (result == 0)
    {
        result = sqrt(result);
        printf("equal root, %f,%f \n", (-b) / (2 * a), (-b) / (2 * a));
    }
    else
    {
        printf("Imaginary root");
    }

    getch();
}
```

Output: -

```
1    0    50    50    Not a Quadratic Equation

2    1    50    50    Real Roots

3    50    50    50    Imaginary Roots

4    99    50    50    Imaginary Roots

5    100    50    50    Imaginary Roots

6    50    0    50    Imaginary Roots

7    50    1    50    Imaginary Roots

8    50    99    50    Imaginary Roots

9    50    100    50    Equal Roots
```

```
10    50    50    0    Real Roots

11    50    50    1    Real Roots

12    50    50    99    Imaginary Roots

13    50    50    100    Imaginary Roots
```

Testing the program: -

In Boundary Value Analysis, there will be 4N+1 test cases which means 4*3+1 = be generated 13 test cases will

Test Case	a	b	c	Expected Output
1	0	50	50	Not Quadratic
2	1	50	50	Real Roots
3	50	50	50	Imaginary Roots
4	99	50	50	Imaginary Roots
5	100	50	50	Imaginary Roots
6	50	0	50	Imaginary Roots
7	50	1	50	Imaginary Roots
8	50	99	50	Imaginary Roots
9	50	100	50	Equal Roots
10	50	50	0	Real Roots
11	50	50	1	Real Roots
12	50	50	99	Imaginary Roots
13	50	50	100	Imaginary Roots