Assignment 1

Name: Zain Zahoor

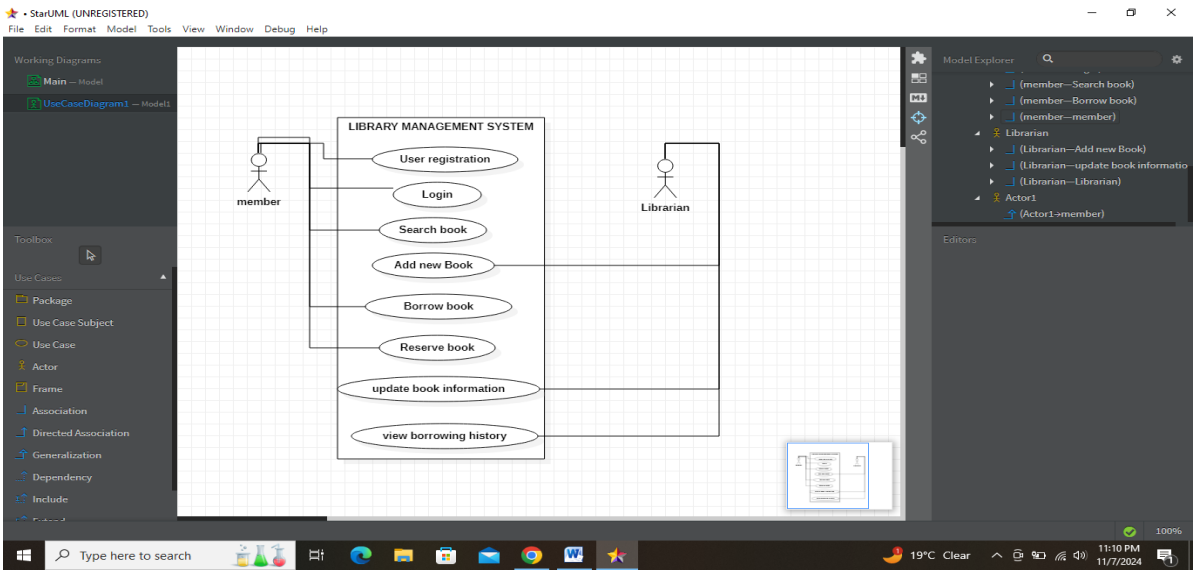
Reg No: Fa21-Bse-166

Submitted To: Sir Mukhtiar Zamin

Introduction to Library Management System (LMS)

A **Library Management System (LMS)** is a software application designed to streamline the management of a library's resources. It automates tasks such as managing books, borrowing books, and tracking return records. An LMS helps librarians and staff to manage various operations efficiently and provides users (students, members, etc.) with an easy way to search for, borrow, and return books.

Use Case:



MODULE: **Add New Book**

Fully Dressed Use Case:

Use Case Name: Add New Book
ID: UC002
Actors: Librarian (Primary), Library System (Secondary)
Description: This use case describes the process where a librarian adds a new book to the library system.

Preconditions:

- The librarian must be authenticated and authorized to add new books.
- The system must be functional and capable of recording book data.

Postconditions:

- The new book is added to the system's catalog.
- The librarian receives confirmation that the book has been successfully added.

Triggers: The librarian initiates the process by selecting the "Add New Book" option.

Basic Flow:

1. The librarian logs into the system.
2. The librarian selects the "Add New Book" option from the menu.
3. The system displays the form for entering book details.
4. The librarian enters the details, including title, author, ISBN, publisher, and genre.
5. The librarian submits the form.
6. The system validates the input and checks for required fields.
7. The system adds the new book to the catalog and stores it in the database.
8. The system generates a confirmation message.
9. The librarian receives the confirmation that the book was successfully added.

Alternative Flow:

- **Duplicate ISBN:** If the ISBN entered already exists in the system, the system displays an error message: "Book with this ISBN already exists."

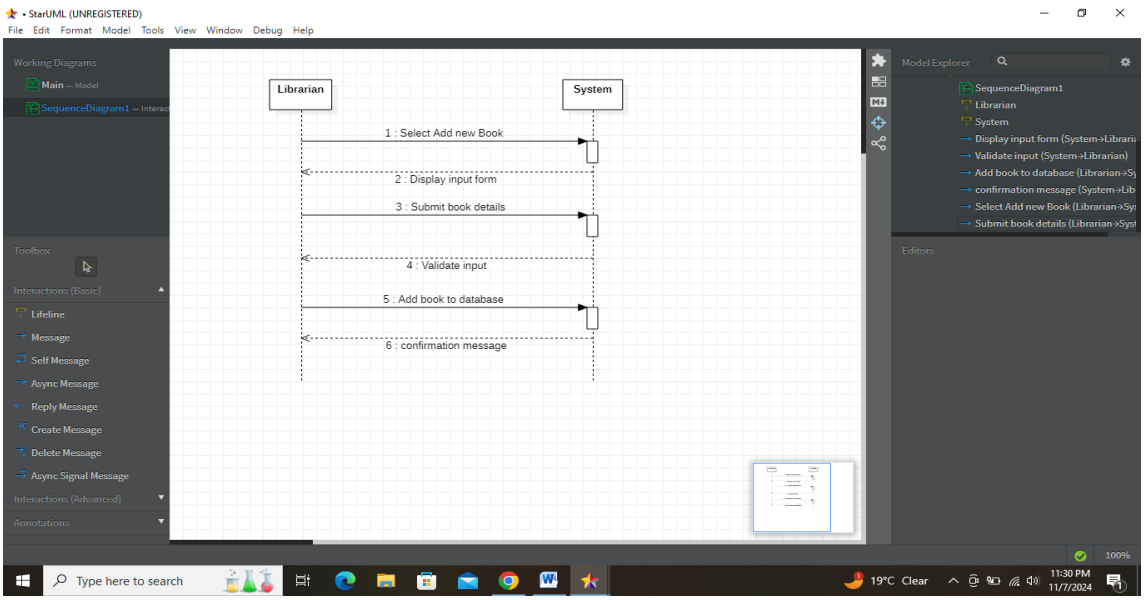
- **Missing Information:** If any required information is missing (e.g., title, ISBN), the system asks the librarian to fill in the missing fields.

Exceptions:

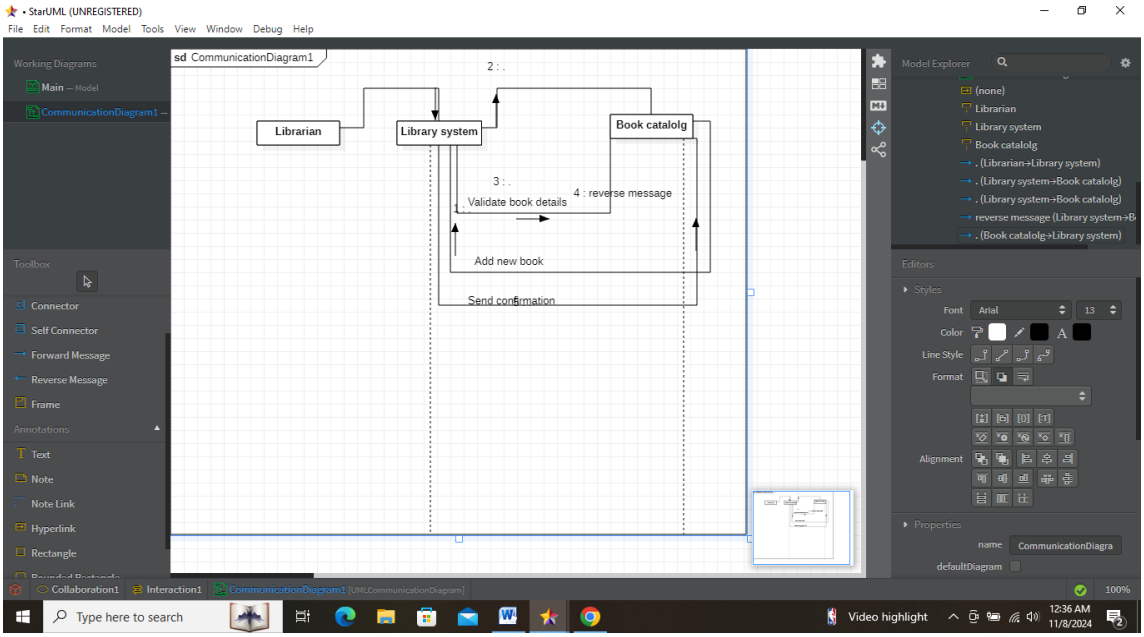
- **System Error:** If there is a technical issue or database failure, the system informs the librarian and asks them to try again later.

- **Input Validation Error:** If the ISBN or date format is incorrect, the system provides a specific error message indicating what needs to be corrected.

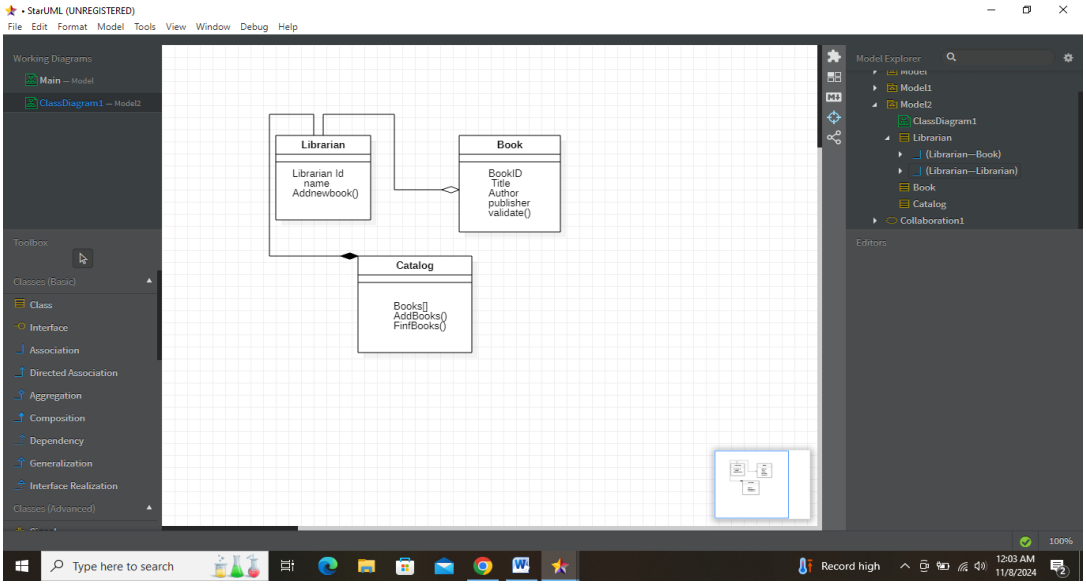
SSD:



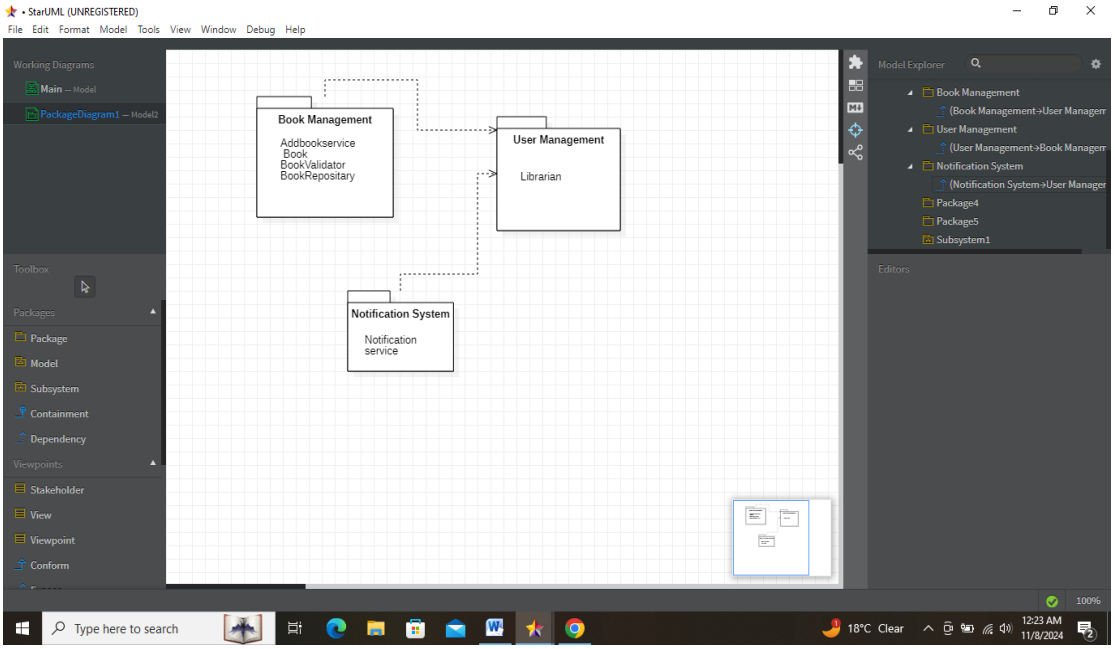
Communication Diagram:



Class Diagram:



Package Diagram



Applying Grasp Principle "Information Expert" on This Module "Add New Book".

Code:

```
public class Book {
    private String title;
    private String author;
    private String isbn;
    private int publicationYear;
    public Book(String title, String author, String isbn, int publicationYear) {
        this.title = title;
        this.author = author;
```

```
        this.isbn = isbn;
        this.publicationYear = publicationYear;
    }
    public String getTitle() { return title; }
    public String getAuthor() { return author; }
    public String getIsbn() { return isbn; }
    public int getPublicationYear() { return publicationYear; }
}
public class BookValidator {
    public boolean isValid(Book book) {
        if (book.getTitle() == null || book.getTitle().isEmpty()) {
            System.out.println("Invalid book: Title is missing");
            return false;
        }
        if (book.getAuthor() == null || book.getAuthor().isEmpty()) {
            System.out.println("Invalid book: Author is missing");
            return false;
        }
        if (book.getIsbn() == null || !isValidISBN(book.getIsbn())) {
            System.out.println("Invalid book: ISBN is invalid");
            return false;
        }
        return true;
    }
    private boolean isValidISBN(String isbn) {
```

```
        return isbn.matches("\\d{10}|\\d{13}");
    }
}
public class BookRepository {
    public void save(Book book) {
        System.out.println("Book titled '" + book.getTitle() + "' has been saved to the database.");
    }
}
public class AddBookService {
    private BookValidator bookValidator;
    private BookRepository bookRepository;
    public AddBookService() {
        this.bookValidator = new BookValidator();
        this.bookRepository = new BookRepository();
    }
    public void addBook(Book book) {
        if (bookValidator.isValid(book)) {
            bookRepository.save(book);
            System.out.println("Book has been added successfully.");
        }
else {
            System.out.println("Book could not be added due to validation errors.");
        }
    }
}
```

```
public class InformationExpert {
    public static void main(String[] args) {
        Book newBook = new Book("Effective Java", "Joshua Bloch", "9780134685991", 2018);
        AddBookService addBookService = new AddBookService();
        addBookService.addBook(newBook);
    }
}
```