International Islamic University Chittagong

Department of Computer Science and Engineering

LAB REPORT

Course title	:	Software Engineering Sessional & Software Development 2
Course Code	:	**CSE-3638 & 3640**
Report No	:	**04**
Report Title	:	**UML Diagrams and Basic Data Structures**

Submitted By

Name	:	**Ariful Hasan Adil**
ID No	:	**C223112**
Section	:	**6CM**
Semester	:	6th

Submitted To

Mohammad Arfizurrahman
Adjunct Faculty
Department of CSE, IIUC

Submission Date : 16/03/2024

1. Overview:

In 4th week class I have learned how to work with basic data structures like arrays and objects in a programming language, and also how to interact with those structures using functions.

2. Objectives:

The goals of the lab was:

- Understanding Basic DataStructure
- Understanding Data Representation, Function Design , Encapsulation and many more....
- Creating UML Diagrams.

3. Tools and Setup

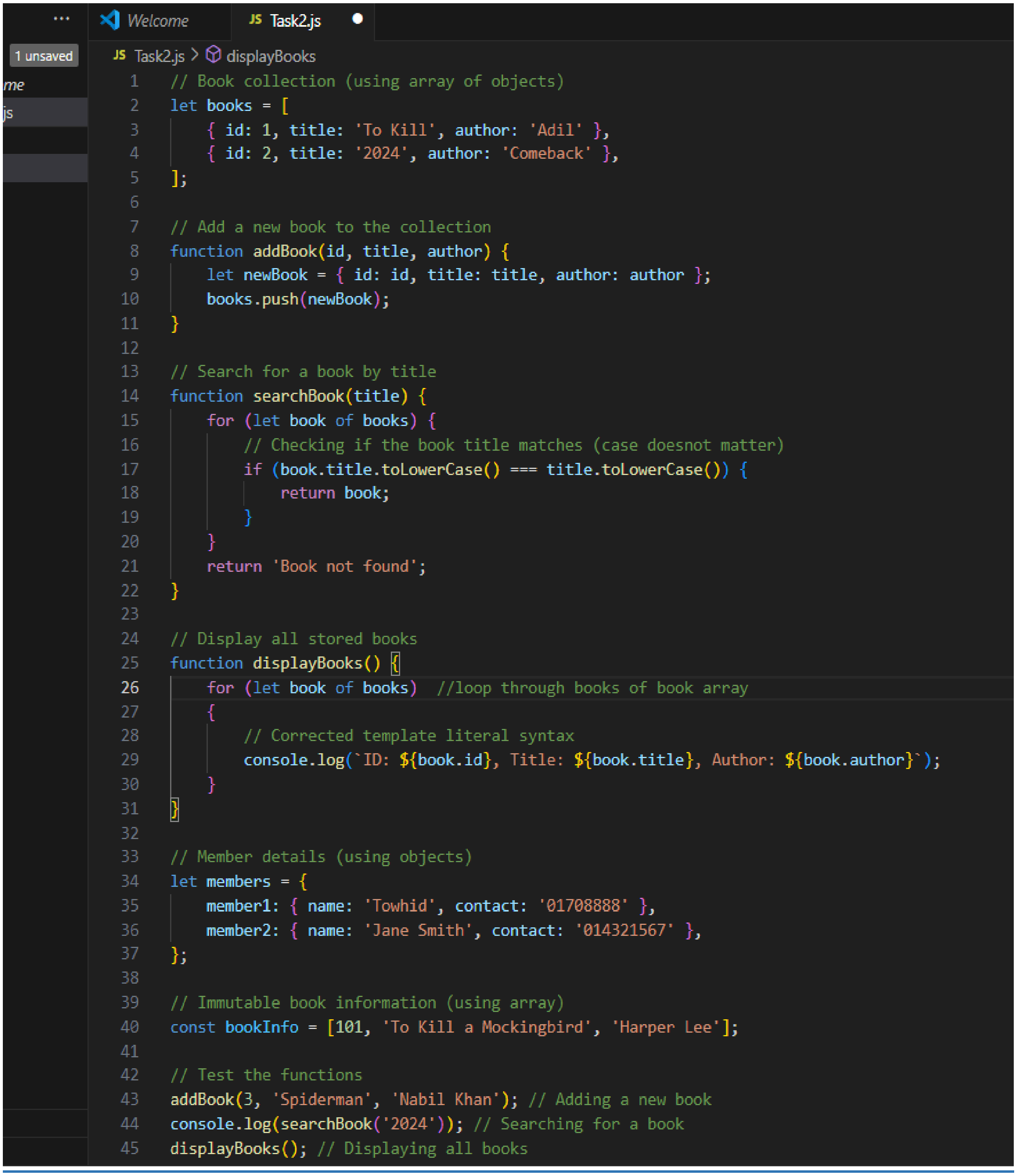
- Programming Language: JavaScript
- Development Environment: VS Code, Node.js

4. Activities Performed (with Examples and Screenshots)

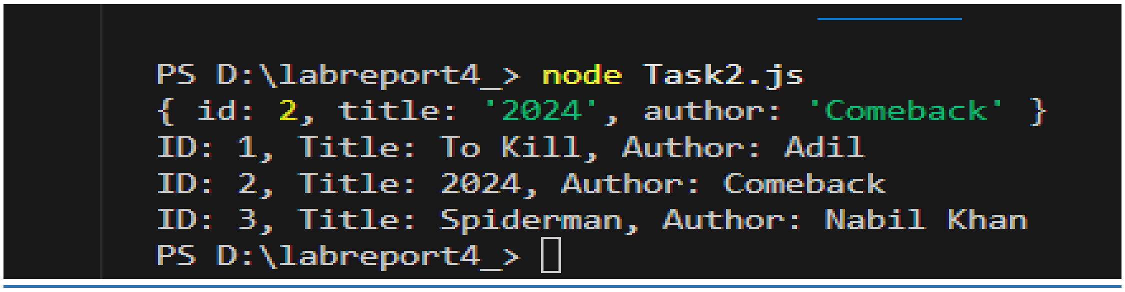
Lab Tasks

Task 2: Implement Basic Data Structures

Data Structure and Prototype Implementation



```javascript
// Book collection (using array of objects)
let books = [
    { id: 1, title: 'To Kill', author: 'Adil' },
    { id: 2, title: '2024', author: 'Comeback' },
];

// Add a new book to the collection
function addBook(id, title, author) {
    let newBook = { id: id, title: title, author: author };
    books.push(newBook);
}

// Search for a book by title
function searchBook(title) {
    for (let book of books) {
        // Checking if the book title matches (case doesnot matter)
        if (book.title.toLowerCase() === title.toLowerCase()) {
            return book;
        }
    }
    return 'Book not found';
}

// Display all stored books
function displayBooks() {
    for (let book of books)  //loop through books of book array
    {
        // Corrected template literal syntax
        console.log(`ID: ${book.id}, Title: ${book.title}, Author: ${book.author}`);
    }
}

// Member details (using objects)
let members = {
    member1: { name: 'Towhid', contact: '01708888' },
    member2: { name: 'Jane Smith', contact: '014321567' },
};

// Immutable book information (using array)
const bookInfo = [101, 'To Kill a Mockingbird', 'Harper Lee'];

// Test the functions
addBook(3, 'Spiderman', 'Nabil Khan'); // Adding a new book
console.log(searchBook('2024')); // Searching for a book
displayBooks(); // Displaying all books
```



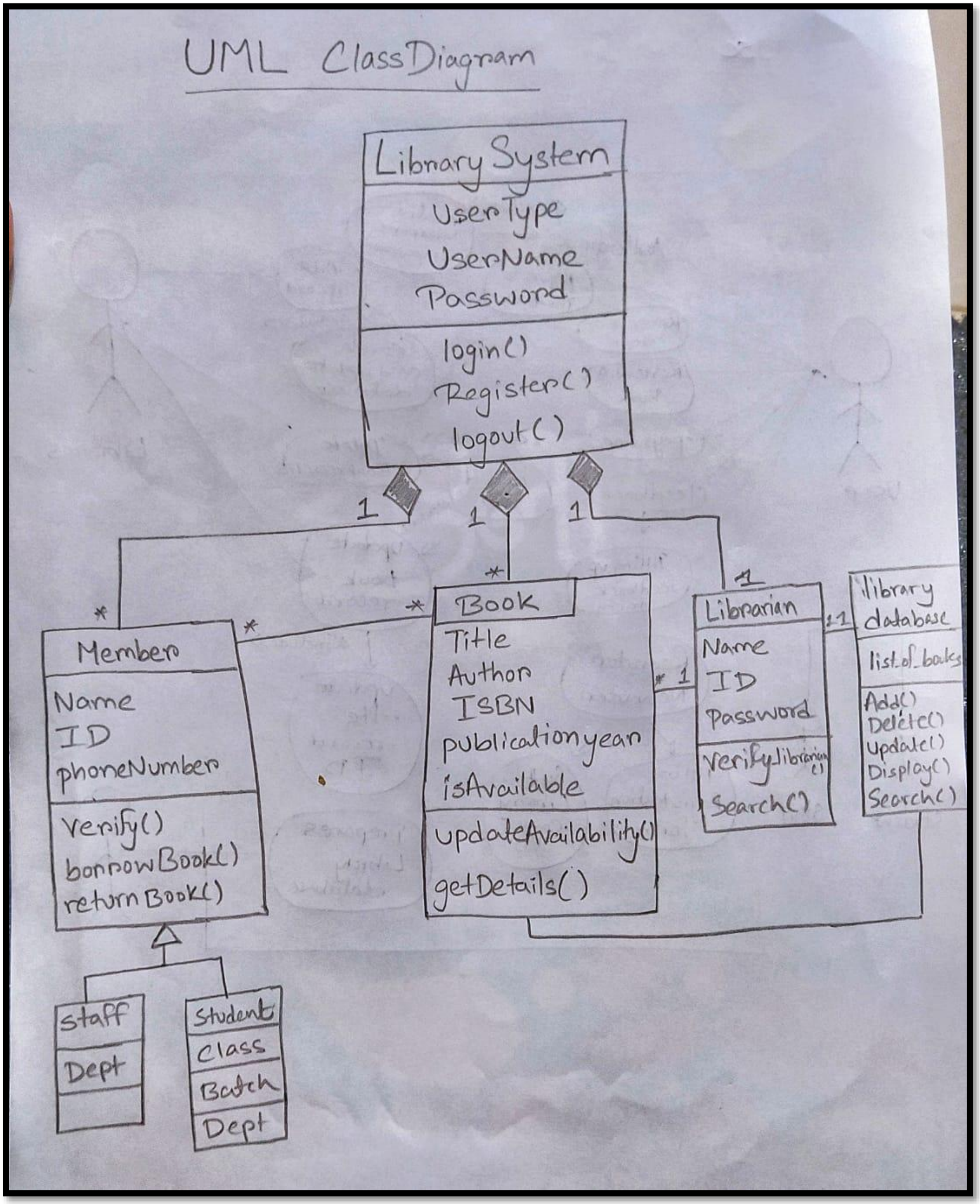
```
PS D:\labreport4_> node Task2.js
{ id: 2, title: '2024', author: 'Comeback' }
ID: 1, Title: To Kill, Author: Adil
ID: 2, Title: 2024, Author: Comeback
ID: 3, Title: Spiderman, Author: Nabil Khan
PS D:\labreport4_> 
```

- **Initializing Lists/Array** to store a collection of books and Writing a small prototype that demonstrates:
  - Initializing a collection of books.
  - Adding a new book to the collection.
  - Searching for a book in the collection.
  - Displaying all stored books.

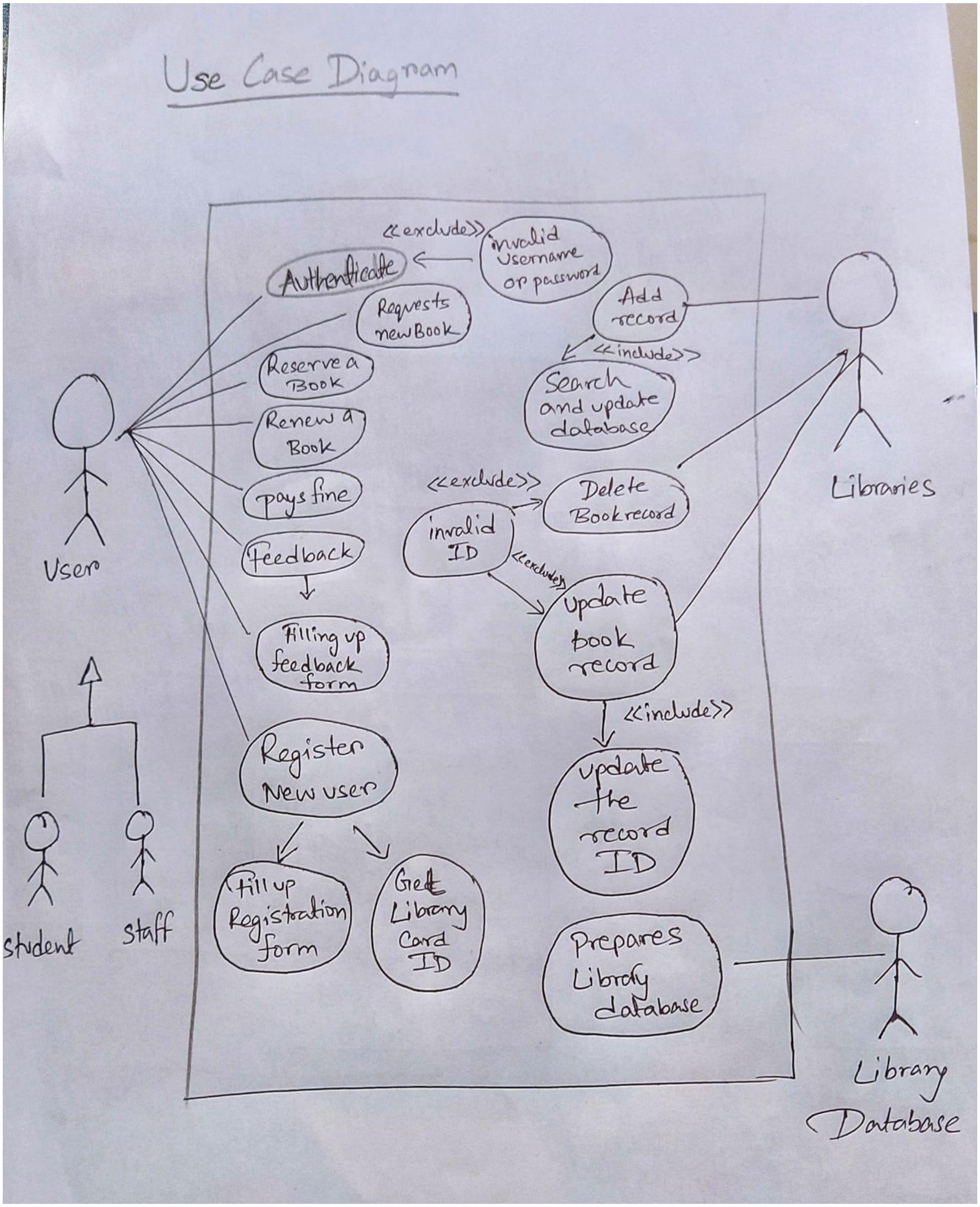
Task 1: Create UML Diagrams

1. **UML Class Diagram:**
   - Identify key classes for a library system (e.g., *Book*, *Member*, *Library*, *Librarian*, *Loan*).
   - Determine and represent relationships (e.g., association, aggregation/composition, inheritance if applicable).
   - Include relevant attributes and methods for each class.
2. **UML Use Case Diagram:**
   - Identify the primary actors (e.g., *Library Member*, *Librarian*).
   - List essential use cases (e.g., *Search Catalog*, *Borrow Book*, *Return Book*, *Manage Inventory*).
   - Show the interactions and relationships between actors and use cases (including any include or extend relationships).

DRAWING UML CLASS DIAGRAM:



DRAWING UML CASE DIAGRAM:



For UML Class Diagram

Key Classes:

1. Book
2. Member
3. Library
4. Librarian

Relationships

- **Association**: `Member` borrows `Book`.
- **Aggregation/Composition**: `Library` contains `Books` and employs `Librarians`.
- **Inheritance** (if necessary): `Staff` (general class) may be extended by `Librarian`.

For UML Case Diagram

Primary Actors

1. **Library Member**
2. **Librarian**

Essential Use Cases

1. **Search Catalog (for Members to search available books)**
   - **Include: `Login` (for secure access)**
2. **Borrow Book (Members check out books)**
   - **Extend: `Notify Overdue` (in case of late returns)**
3. **Return Book (Members return borrowed books)**
4. **Manage Inventory (Librarians oversee book inventory)**
   - **Include: `Add Book`, `Remove Book`**

Relationships

- **Show arrows connecting actors (e.g., `Library Member`) to the use cases they interact with (e.g., `Borrow Book`, `Search Catalog`).**
- **Use `include` and `extend` as needed to connect related use cases.**

. Reflections and Learnings:

FROM THIS LAB CLASS I UNDERSTOOD MORE ABOUT OBJECT-ORIENTED CONCEPTS AND LEARNED HOW TO DRAW UML CLASS AND CASE DIAGRAM. I ALSO GOT TO KNOW MORE ABOUT PROBLEM ANALYSIS AND GOT UML PROFICIENCY.

IN THE FUTURE IT CAN BE HELPFUL FOR SOFTWARE DEVELOPMENT, SYSTEM ANALYSIS AND DESIGN.

IT CAN ALSO BE USED IN TEAM COLLABORATION AND PROJECT MANAGEMENT.