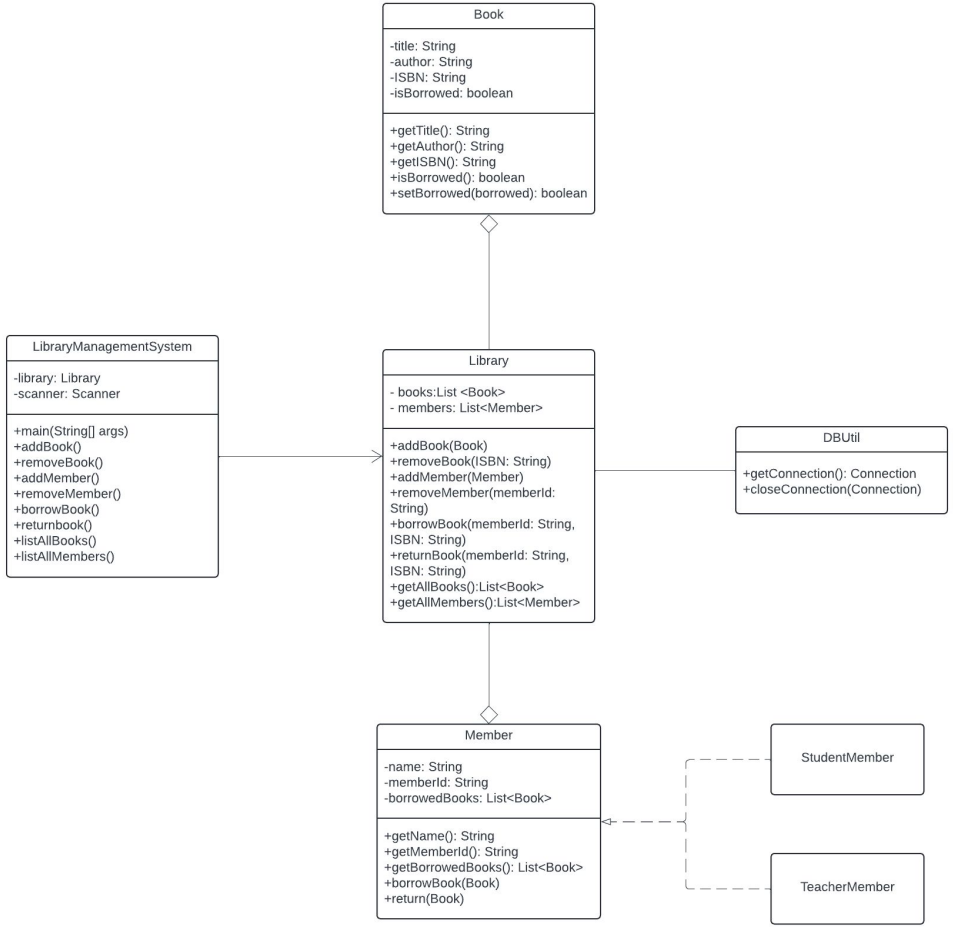
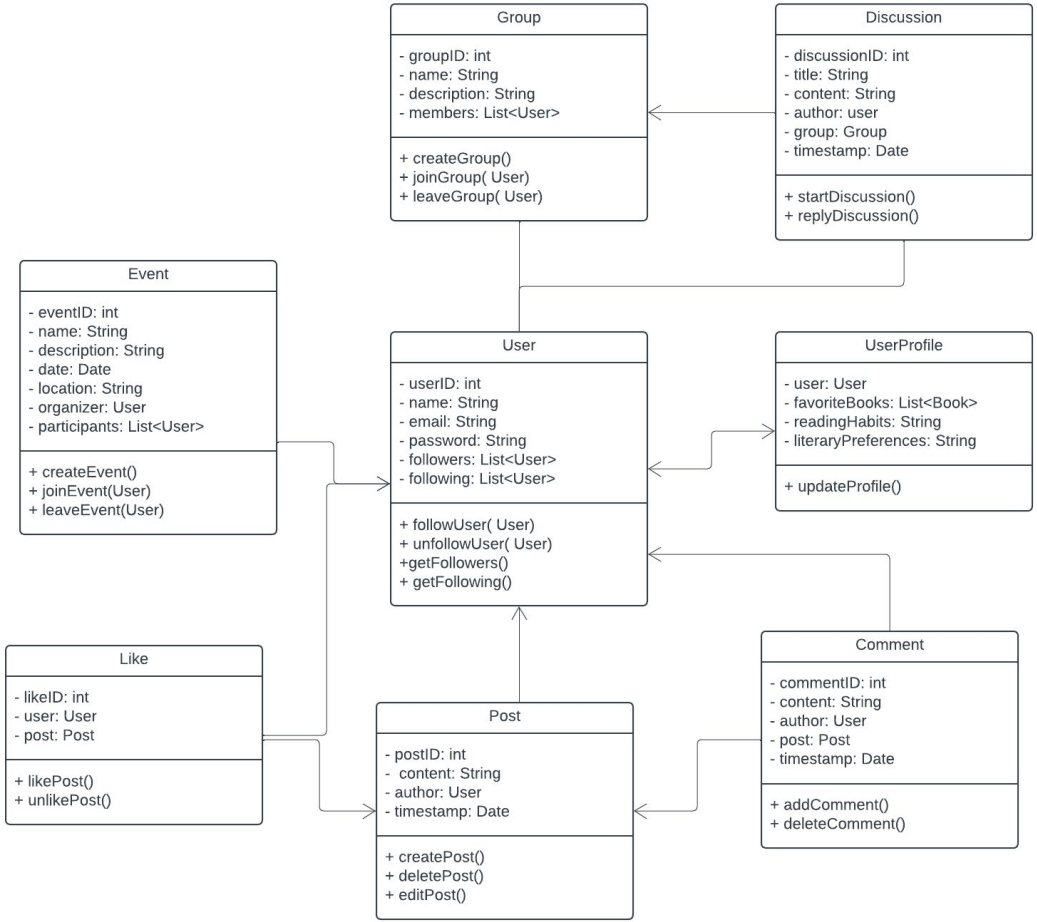
Library System Presentation

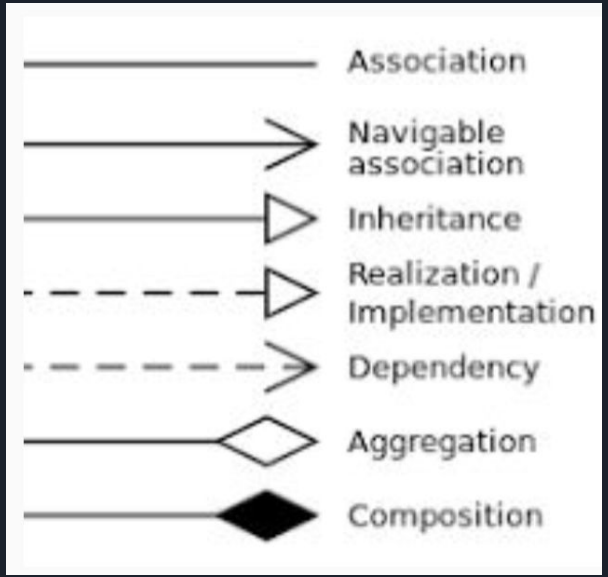
By: Minh Tran, Thu Pham, and Ruben Plascencia


Book
-title: String
-author: String
-ISBN: String
-isBorrowed: boolean
+getTitle(): String
+getAuthor(): String
+getISBN(): String
+isBorrowed(): boolean
+setBorrowed(borrowed): boolean
LibraryManagementSystem
-library: Library
-scanner: Scanner
+main(String[] args)
+addBook()
+removeBook()
+addMember()
+removeMember()
+borrowBook()
+returnbook()
+listAllBooks()
+listAllMembers()
Library
- books:List <Book>
- members: List<Member>
+addBook(Book)
+removeBook(ISBN: String)
+addMember(Member)
+removeMember(memberId: String)
+borrowBook(memberId: String, ISBN: String)
+returnBook(memberId: String, ISBN: String)
+getAllBooks():List<Book>
+getAllMembers():List<Member>
DBUtil
+getConnection(): Connection
+closeConnection(Connection)
Member
-name: String
-memberId: String
-borrowedBooks: List<Book>
+getName(): String
+getMemberId(): String
+getBorrowedBooks(): List<Book>
+borrowBook(Book)
+return(Book)
StudentMember
TeacherMember


Group
- groupID: int
- name: String
- description: String
- members: List<User>
+ createGroup()
+ joinGroup(User)
+ leaveGroup(User)
Discussion
- discussionID: int
- title: String
- content: String
- author: user
- group: Group
- timestamp: Date
+ startDiscussion()
+ replyDiscussion()
Event
- eventID: int
- name: String
- description: String
- date: Date
- location: String
- organizer: User
- participants: List<User>
+ createEvent()
+ joinEvent(User)
+ leaveEvent(User)
User
- userID: int
- name: String
- email: String
- password: String
- followers: List<User>
- following: List<User>
+ followUser(User)
+ unfollowUser(User)
+getFollowers()
+ getFollowing()
UserProfile
- user: User
- favoriteBooks: List<Book>
- readingHabits: String
- literaryPreferences: String
+ updateProfile()
Like
- likeID: int
- user: User
- post: Post
+ likePost()
+ unlikePost()
Post
- postID: int
- content: String
- author: User
- timestamp: Date
+ createPost()
+ deletePost()
+ editPost()
Comment
- commentID: int
- content: String
- author: User
- post: Post
- timestamp: Date
+ addComment()
+ deleteComment()

Overall System Design in UML

- In our UML Design, we decided to split both parts of the project into two UML Designs for each part(Library Management and Library Social Network)
- In the design for the Library Management System, we utilized Association, Navigable Association, Realization/Implementation, and Aggregation to display the relationships that each class and its attributes may have amongst each other
-

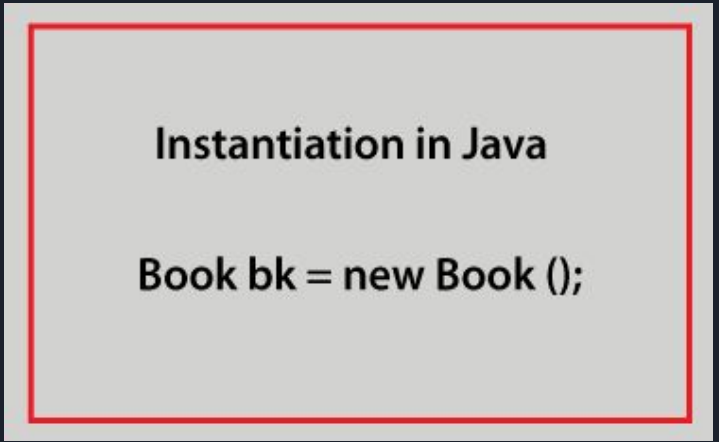


Experience of Extending the Design

- While we were extending the design from the part 1(Library Management System) to part 2(Literary Social Network), we used some similar Class Structures and utilized some attributes from part 1 to part 2
- In the design for the Library Social Network, we took a more simplistic design approach(compared to Library Management System) with utilizing only Navigable Association to showing the relationship each class can have and how they can directly access one another.

How We Applied Our Learning

- We utilized our understanding of differentiating classes and their relationships from the different examples we've seen and the assignments we've done.
- Applying Abstract classes, utilizing constructors and override methods, extending class attributes from one class to another with Inheritance, and finally; utilizing implementation functions to transfer methods and attributes from one class to another
- We additionally applied our understanding of instantiating objects from assignment 2 in order to create objects for the integral classes and to print them.



```
Library Management System
1. Add Book
2. Remove Book
3. Add Member
4. Remove Member
5. Borrow Book
6. Return Book
7. List All Available Books
8. List All Members
9. Search Books
10. Update Member Information
11. Generate Borrowed Books Report
12. Generate Overdue Books Report
13. Exit
Enter choice: 3
Enter name: Alice
Enter member ID: S123
Enter member type (1 for Student, 2 for Teacher):
1
Member added successfully.
```

```
Library Management System
1. Add Book
2. Remove Book
3. Add Member
4. Remove Member
5. Borrow Book
6. Return Book
7. List All Available Books
8. List All Members
9. Search Books
10. Update Member Information
11. Generate Borrowed Books Report
12. Generate Overdue Books Report
13. Exit
Enter choice: 5
Enter member ID: S123
Enter ISBN of the book to borrow: 978-0134685991
Enter due date (yyyy-MM-dd): 2024-08-01
Book borrowed successfully.

Library Management System
```

```
1. Add Book
2. Remove Book
3. Add Member
4. Remove Member
5. Borrow Book
6. Return Book
7. List All Available Books
8. List All Members
9. Search Books
10. Update Member Information
11. Generate Borrowed Books Report
12. Generate Overdue Books Report
13. Exit
Enter choice: 11
Borrowed Books:
Book: Book{title='Effective Java', author='Joshua Bloch', ISBN='978-0134685991', isBorrowed=tru
024-08-01}, Borrowed by: Alice (ID: S123), Borrowed Date: 2024-07-03
Book: Book{title='Python Basic', author='Mac Culan', ISBN='978-0134685993', isBorrowed=true, du
5-20}, Borrowed by: Bob (ID: T456), Borrowed Date: 2024-07-03

Library Management System
```

```
Library Management System
1. Add Book
2. Remove Book
3. Add Member
4. Remove Member
5. Borrow Book
6. Return Book
7. List All Available Books
8. List All Members
9. Search Books
10. Update Member Information
11. Generate Borrowed Books Report
12. Generate Overdue Books Report
13. Exit
Enter choice: 7
All Available Books:
Book{title='Java Concurrency in Practice', author='Brian Goetz', ISBN='978-0134685992', isBorro
ueDate=null}

Library Management System
1. Add Book
```

```
1. Add Book
2. Remove Book
3. Add Member
4. Remove Member
5. Borrow Book
6. Return Book
7. List All Available Books
8. List All Members
9. Search Books
10. Update Member Information
11. Generate Borrowed Books Report
12. Generate Overdue Books Report
13. Exit
Enter choice: 8
All Members:
Name: Alice, Member ID: S123, Type: student
Name: Bob, Member ID: T456, Type: teacher

Library Management System
1. Add Book
2. Remove Book
```

```
Library Management System
1. Add Book
2. Remove Book
3. Add Member
4. Remove Member
5. Borrow Book
6. Return Book
7. List All Available Books
8. List All Members
9. Search Books
10. Update Member Information
11. Generate Borrowed Books Report
12. Generate Overdue Books Report
13. Exit
Enter choice: 2
Enter ISBN of the book to remove: 978-0134685992
Book removed successfully.

Library Management System
1. Add Book
```

```
Library Management System
1. Add Book
2. Remove Book
3. Add Member
4. Remove Member
5. Borrow Book
6. Return Book
7. List All Available Books
8. List All Members
9. Search Books
10. Update Member Information
11. Generate Borrowed Books Report
12. Generate Overdue Books Report
13. Exit
Enter choice: 4
Enter member ID of the member to remove: T456
Member removed successfully.

Library Management System
1. Add Book
```

```
Library Management System
1. Add Book
2. Remove Book
3. Add Member
4. Remove Member
5. Borrow Book
6. Return Book
7. List All Available Books
8. List All Members
9. Search Books
10. Update Member Information
11. Generate Borrowed Books Report
12. Generate Overdue Books Report
13. Exit
Enter choice: 6
Enter member ID: T456
Enter ISBN of the book to return: 978-0134685992
Book returned successfully.

Library Management System
1. Add Book
```

```
Library Management System
1. Add Book
2. Remove Book
3. Add Member
4. Remove Member
5. Borrow Book
6. Return Book
7. List All Available Books
8. List All Members
9. Search Books
10. Update Member Information
11. Generate Borrowed Books Report
12. Generate Overdue Books Report
13. Exit
Enter choice: 10
Enter current member ID: T456
Enter new name: Bobby
Enter new member ID: T007
Member information updated successfully.

Library Management System
```