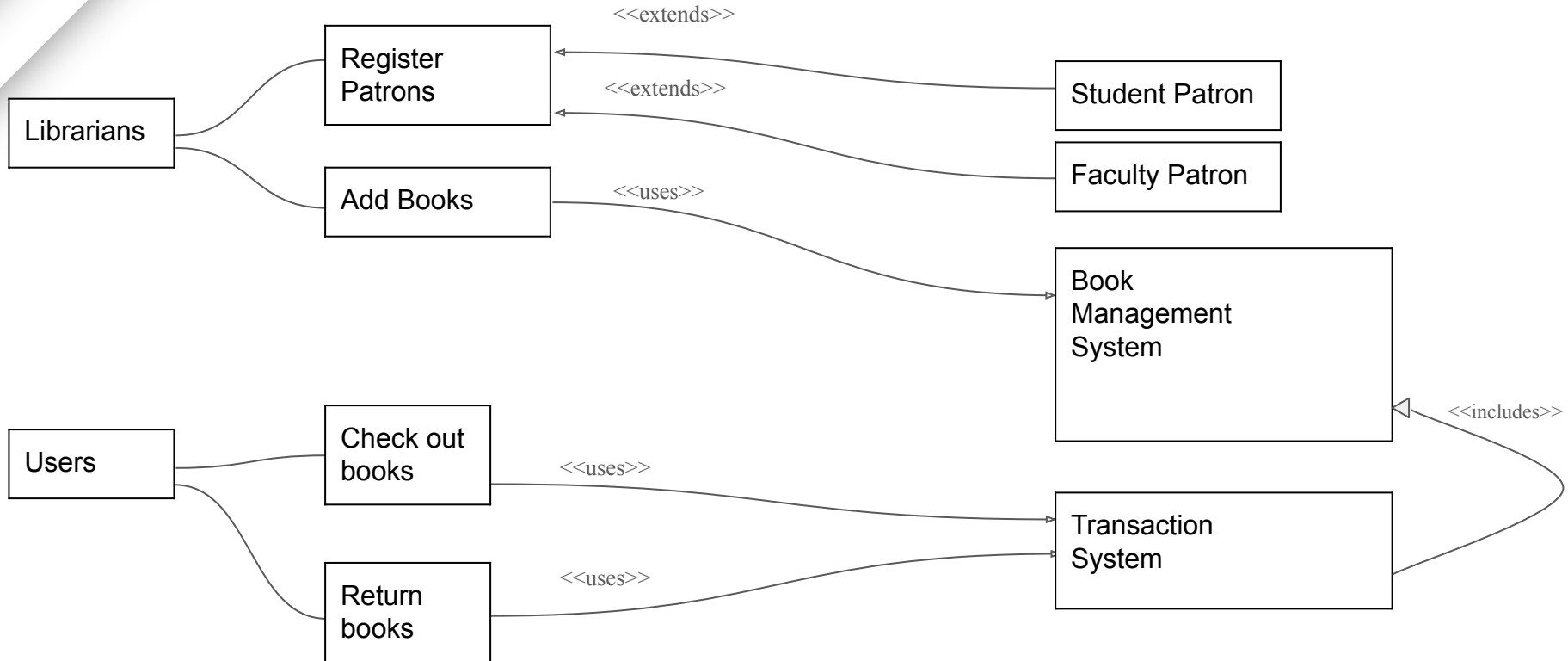


BiblioConnect

Library Management & Literary Social Network

Jesus Aguayo, William Arata, Joshua Patrick Cainglet, Jorge Chavez, Vinny Ngo

Use Case Diagram: Library Management System



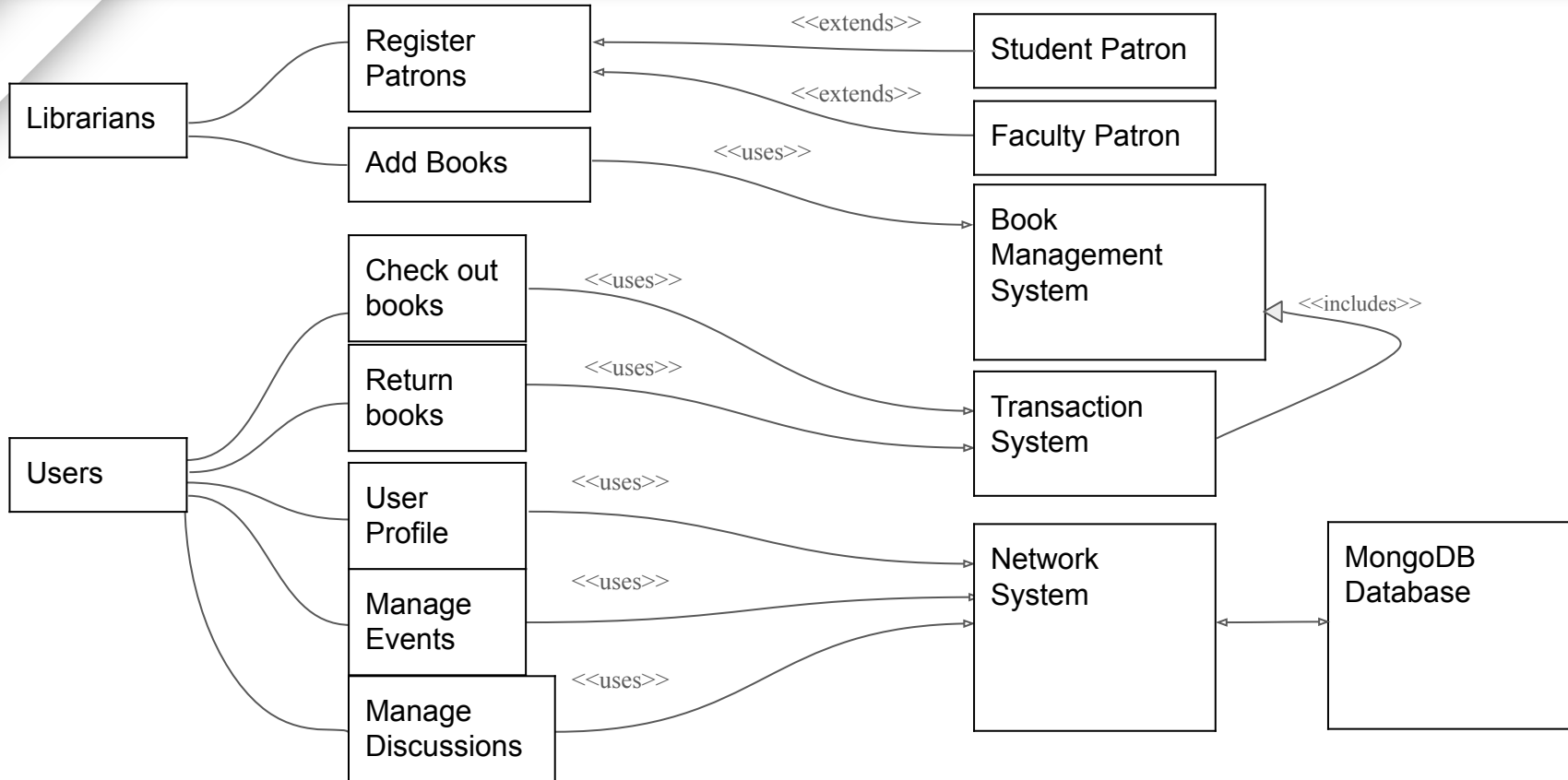
Literary Social Network Extension

Extending the Library Management System, the Literary Social Network functionality allow patrons to interact and connect. Leveraging MongoDB for database implementation, patrons can now seamlessly sign into their accounts via the command line interface, gaining access their profile, groups, events, and more than just checking out books.

Literary Social Network capabilities:

- Showcase their favorite books and genres
- Add an unlimited amount of friends
- Create and join groups and participate in discussions
- Create, view, and participate in events or meetings

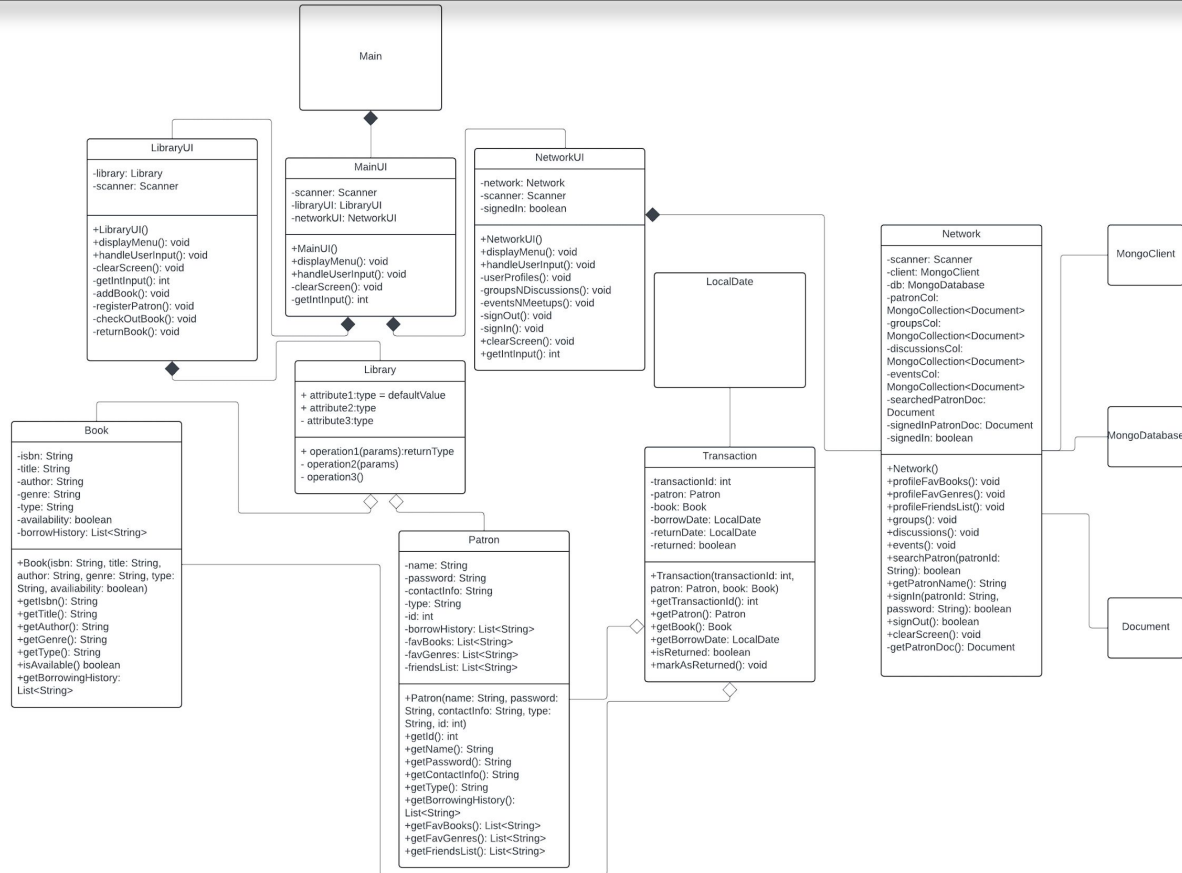
Use Case Diagram: Literary Social Network



Class Diagrams

<i>Book</i>	<i>Patron</i>	<i>Transaction</i>	<i>Network</i>
-isbn : String -title : String -author : String -genre : String -type : String -availability : boolean -borrowHistory : List<String>	-name : String -password : String -contactInfo : String -type : String -id : Int -borrowHistory : List<String> -favBooks : List<String> -favGenres : List<String> -friendsList : List<String>	-transactionId : Int -patron : Patron -book : Book -borrowDate : LocalDate -returnDate : LocalDate -returned : boolean +getTransactionId() : Int +getPatron() : Patron +getBook() : Book +getBorrowDate() : LocalDate +getReturnDate() : LocalDate +isReturned() : boolean +markAsReturned() : void	+userProfile(patronId : String) +groupNDiscussions() : void +eventsNMeetups() : void +signIn(patronId : String, password : String) +signOut() : boolean
+getIsbn() : String +getTitle() : String +getAuthor() : String +getGenre() : String +getType() : String +isAvailable() : boolean +getBorrowHistory() : List<String>	+getId() : Int +getName() : String +getPassword() : String +getContactInfo() : String +getType : String +getBorrowingHistory() : List<String> +getFavBooks() : List<String> +getFavGenres() : List<String> +getFriendsList() : List<String>	<i>Library</i>	<i>LibraryUI</i>
		+addBook(book : Book) +addPatron(patron : Patron) +borrowBook(patronId : int, isbn : String) +returnBook(id : int)	+displayMenu() : void +handleUserInput : void +clearScreen() : void +getIntInput : Int +addBook() : void +registerPatron() : void +checkOutBook() : void +returnBook() : void

Overall System Design in UML



Examples of learned software engineering principles

Design:

In the project's design phase, early challenges emerged as functions and classes became increasingly complicated. Recognizing the need for a better design, we created a UML diagram to visualize an implementation to simplify and extend our existing codebase without extensive modifications

Implementation:

With the improved UML diagram, we followed its blueprint to enhance our existing Library system. Each class and function was separated to adhere to the principle of single responsibility, facilitating better organization and maintainability.

Testing:

Throughout the program's development, testing was conducted on both the application and its database. We compared expected outcomes with actual results, making sure the behavior of each class's functions and sub-classes to ensure it behave as intended with their parent classes.

Live Demo



Thank You!