

Library Book Management System R1

Design Documentation

Prepared by TEAM C:

- Ryan Tytka rdt7867@rit.edu
- Mikayla Wishart mcw7246@rit.edu
- Bryan Wang jw3513@g.rit.edu
- Yug Patel ydp4388@rit.edu

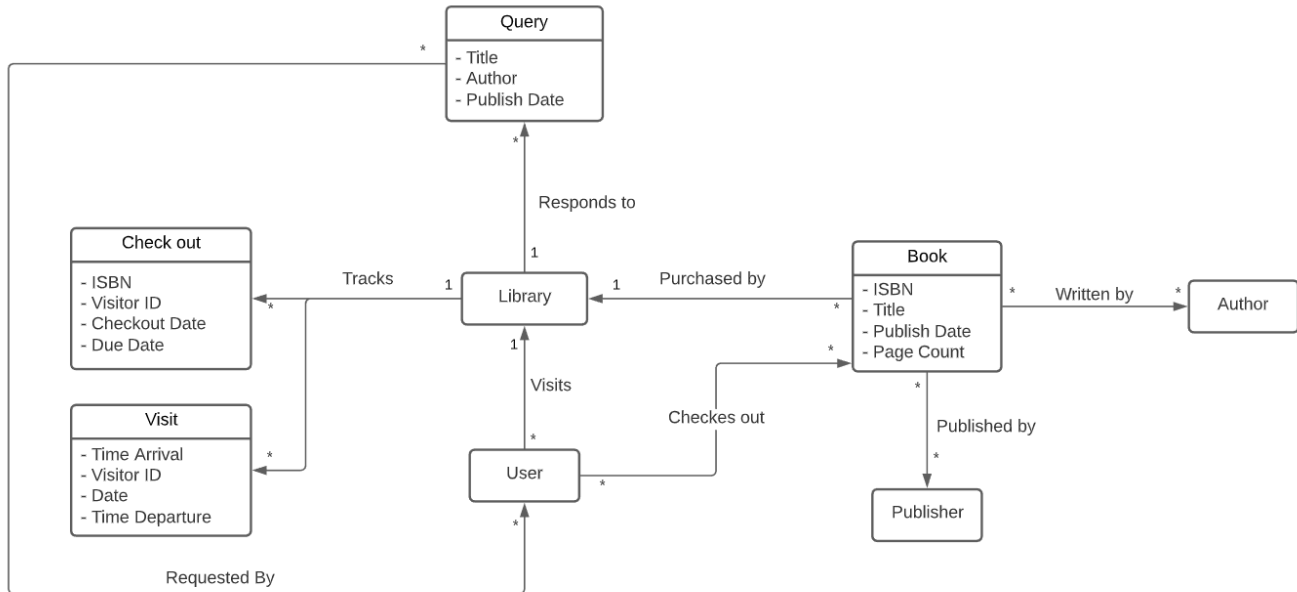
Summary	2
Domain Model	3
Subsystems	4
Library State	4
Library Product	6
Status of the Implementation	7
Appendix	8

Summary

Our team was tasked with designing and implementing the Library Book Management System (LBMS). The LBMS is Book Worm Library's (BWL) system for providing book information to users, tracking library visitor statistics for a library statistics report, tracking checked out books, and allowing the library inventory to be updated. It is the server-side system that provides an API used by client-side interfaces that BWL employees use.

Domain Model

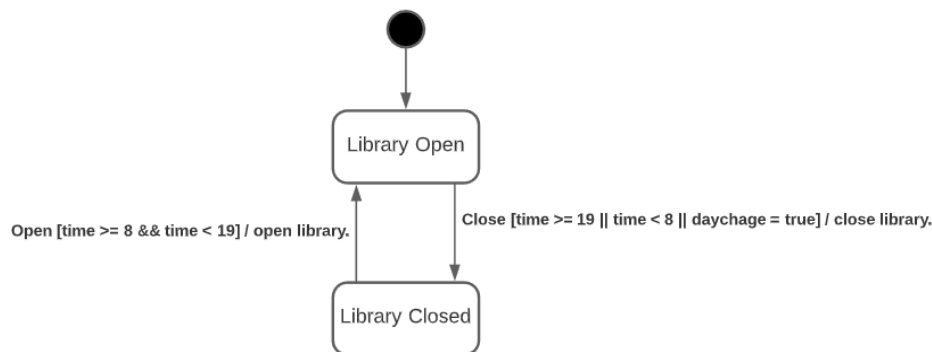
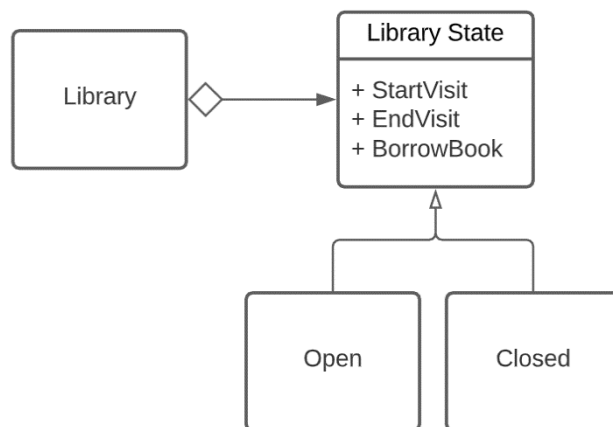
This section provides a domain model for the project. It should follow the guidelines discussed in class and the design project activity sheets. For it to be readable, you may need to turn this page into landscape mode.

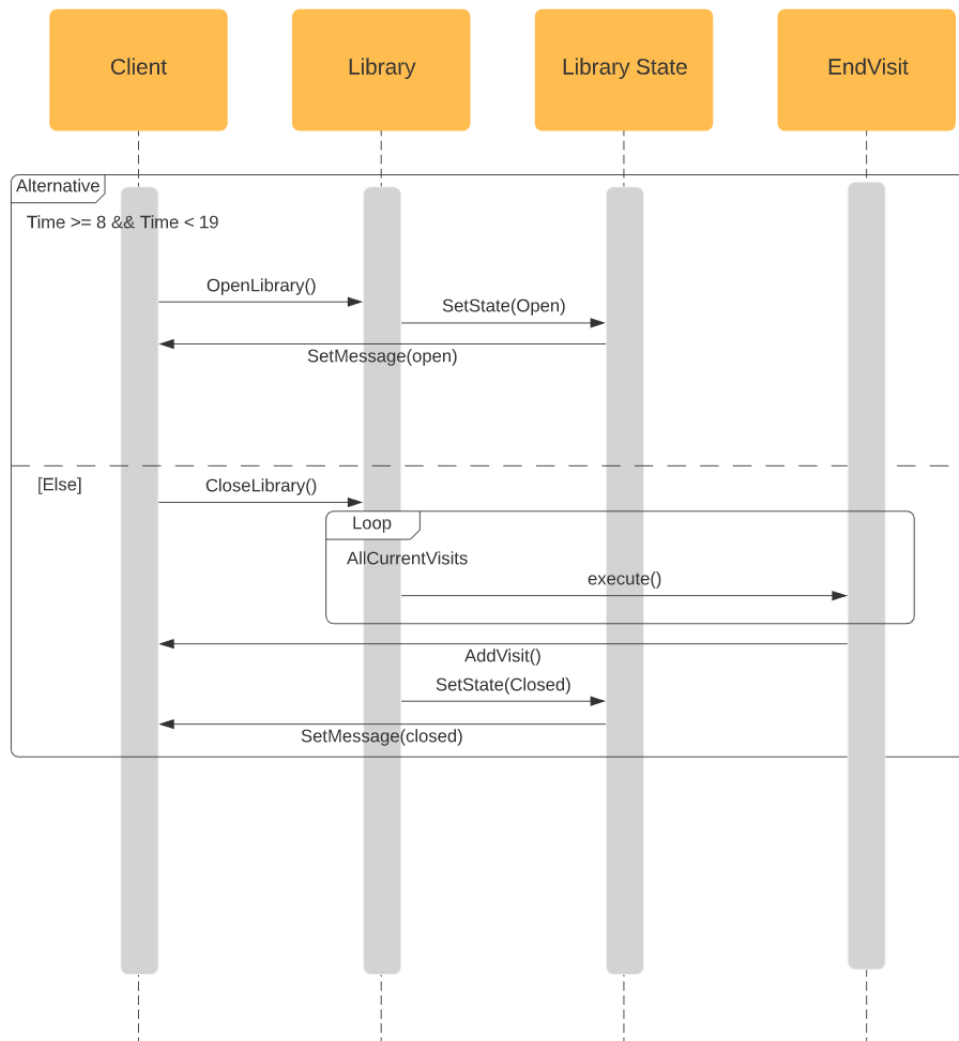


Subsystems

Library State

The library state change functionality uses the most basic form of the State Pattern. There are two states that the library object will switch between. The change of states are dependent on the calendar object in the Client class. Everytime there is a change in time or day, the method class in Client checks for the current state of the library and changes it through library methods if needed. In the Open state, the library will allow the visitors to start and end the visit, or allow the user to borrow books that the library has purchased. In the Closed state, all the current visits are ended by creating EndVisit commands. The library will not allow visitors to start or end visits, or allow the user to borrow any books from the library.

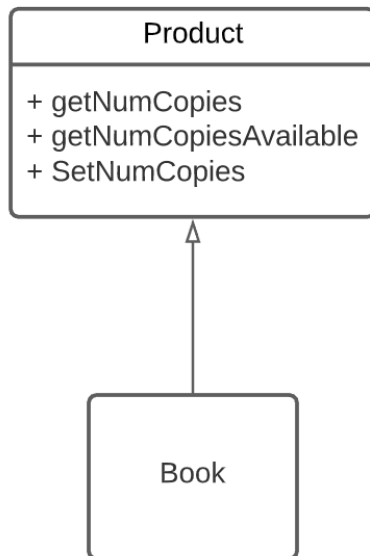




Name: Library State		GoF pattern: State Pattern
Participants		
Class	Role in GoF pattern	Participant's contribution in the context of the application
Library	Context	Library class holds all the information about purchased books, registered visitors, current visitors; and handles all the methods that may access them.
Library State	State	LibraryState acts as an interface for the 2 states of the library.
Open	Concrete State	Open class holds all methods that are exclusive to the open state.
Closed	Concrete State	Closed class includes all methods that are exclusive to the closed state.
Deviations from the standard pattern:		
Requirements being covered: The library will not allow the following transactions while closed		

Library Product

The library product functionality has a product interface that any current or future products in a library. They may have the same behaviors, which will make it easier to perform requests.



Name: Library Product		GoF pattern: Strategy Pattern
Participants		
Class	Role in GoF pattern	Participant's contribution in the context of the application
Book store	Context	BookStore holds all the products in the system through parsing a file.
Library	Context	Library holds all the products that I has purchased.
Product	Strategy	Product acts as an interface for all the products that are available now or in the future.
Book	Concrete Strategy	Book uses the Product interface to represent a single book in the system.
Deviations from the standard pattern:		
Requirements being covered: The system shall store book data for all books currently in BWL's possession, The system will store the data of the book check out transaction		

This section provides detailed design for specific subsystems described in the system architecture.

Status of the Implementation

Requirements:

Register Visitor - Complete

Start Visit - Complete

End Visit - Complete

Generate Report - Partially complete (fines collected - not complete, fines outstanding - not complete)

Advance Time - Complete

Display Date Time - Complete

Store Search - Complete

Library Search - Partially complete (output for "*" is not completed)

Pay fine - Partially complete. Implemented but not tested

Return Book - Partially completed. Implemented but not tested.

Appendix

This section provides fine-grained design details for all of the classes in your design. We captured this information using the CRC (Class-Responsibilities-Collaborators) card format below.

Class: Book	
Responsibilities: This class contains the behavior for the book object. It holds the title, ISBN, publisher, publish date, and page count.	
Collaborators	Author: Team C
Users: None	Used By: BookSearch, BookStore, CheckOut, Client, Library, Sort, StoreSearch, Visitor

Class: BookStore	
Responsibilities: The class that stores all information of books from the text file.	
Collaborators	Author: Team C
Users: Client, Book	Used By: Client, Library

Class: Client	
Responsibilities: Updates interface, sends requests, Tracks time	
Collaborators	Author: Team C
Usages: Library, BookStore, CommandParser, Visit, Book, Client	Used By: BookStore, AdvanceTime, CommandParser, DateTime, Report, Closed, Library, Open Main

Class: AdvanceTime	
Responsibilities: Command that moves date forward specified amount of time	
Collaborators	Author: Team C
Usages: Client	Used By: CommandParser

Class: BeginVisit	
Responsibilities: The BeginVisit class adds a new visit to the Library by giving Visitor IDs. If it's a new visitor, it requires you to register to the Library. If the visitor ID is invalid, it will show the error message.	
Collaborators	Author: Team C
Usages: Library	Used By: CommandParser

Class: BookSearch	
Responsibilities: Implements the Command interface and takes given parameters and searches the given library for a book that matches these parameters.	
Collaborators	Author:
Usages: Library, Book	Used By: CommandParser

Class: BorrowBook	
Responsibilities: The BorrowBook class enables the books that visitors begin to borrow. This class also provides the date that the books got borrowed.	
Collaborators	Author: Team C
Users: Calendar, Library, Command, BookSearch	Used By: CommandParser

Class: BorrowedBooks	
Responsibilities: The BorrowedBooks helps to find out the specific book borrowed from a given Visitor ID, and also calculate the time(the number of days) that visitor borrowed when the book got returned.	
Collaborators	Author: Team C
Users: Command, Library, BookSearch, CheckOut, Calendar	Used By: CommandParser

Class: Command	
Responsibilities: Interface for all request commands	
Collaborators	Author: Team C
Users: None	Used By: AdvanceTime, BeginVisit, BookSearch, BorrowBook, BorrowedBook, CommandParser, DateTime, EndVisit, NewVisitor, PayFine, PurchaseBook, Report, ReturnBook, StoreSearch

Class: CommandParser	
Responsibilities: Parses through commands that are given	
Collaborators	Author: Team C
Users: Library, Client, BookStore, AdvanceTime, BeginVisit, BookSearch, Borrow-Book, BorrowedBook, DateTime, EndVisit, NewVisitor, PayFine, PurchaseBook, Report, ReturnBook, StoreSearch	Used By: Client, main

Class: DateTime	
Responsibilities: The DateTime class outputs the current date and time in the library.	
Collaborators	Author: Team C
Users: Client	Used By: CommandParser

Class: EndVisit	
Responsibilities: The EndVisit class removes a visitor from the library.	
Collaborators	Author: Team C
Users: Library	Used By: CommandParser

Class: NewVisitor	
Responsibilities: The NewVisitor class adds a new visitor that registered in the library along with first name, last name, address and phone number.	
Collaborators	Author: Team C
Users: Library	Used By: CommandParser

Class: PayFine	
Responsibilities: Command that allows a visitor to pay off accumulated fines.	
Collaborators	Author: Team C
Users: Library	Used By: CommandParser

Class: PurchaseBooks	
Responsibilities: The PurchaseBooks class shows the search result of books ID in the library inventory. The user can buy any of those books along with available quantity from the inventory.	
Collaborators	Author: Team C
Users: Command, Library, Book, BookSearch	Used By: CommandParser

Class: Report	
Responsibilities: Generates a report of statistics about the library.	
Collaborators	Author: Team C
Users: Client	Used By: CommandParser

Class: ReturnBook	
Responsibilities: Command that allows a visitor to return borrowed books.	
Collaborators	Author: Team C
Users: Library	Used By: CommandParser

Class: StoreSearch	
Responsibilities: Implements the Command interface and takes given parameters and searches the BookStore books that match the parameters.	
Collaborators	Author: Team C
Users: BookStore	Used By: CommandParser

Class: Closed	
Responsibilities: The closed state of the library	
Collaborators	Author: Team C
Users: Library, Visitor, Client	Used By: Client, Library

Class: Library	
Responsibilities: Contains the behavior for the Library object.	
Collaborators	Author: Team C
Users: LibraryState, CheckOut, Book, BookStore, Client, Visitor, Command	Used By: Client, BeginVisit, BookSearch, Command, CommandParser, EndVisit, NewVisitor, PurchaseBook, Closed, Open

Class: LibraryState	
Responsibilities: Interface used to track when the library opens and closes	
Collaborators	Author: Team C
Users: None	Used By: Closed, Library, Open

Class: Open	
Responsibilities: The open state of the library	
Collaborators	Author: Team C
Users: Visitor, Library, Client, Visit	Used By: Library

Class: Visit	
Responsibilities: Object representing a visit to the library	
Collaborators	Author: Team C
Users: None	Used By: Client, Open

Class: Visitor	
Responsibilities: Contains the behavior for the Visitor object. Holds the information for first name, last name, address, phone number, and visitor ID. Also keeps track of a specific visitor's fines due/paid and books that are checked out.	
Collaborators	Author: Team C
Users: CheckOut, PayFine, UnpaidFine, Book	Used By: NewVisitor, Closed, Library, Open