INFO C451 Project Proposal

Sean Wilson

1/28/2024

Table of Contents

Problem Statement

The world today prefers things to be convenient, and to be convenient, and relevant in today culture, a business must be able to conduct its business in a matter that allows the customer to utilize their phone in an easy and effective manner. Studies show that individuals would rather not leave their home to get information or to rent a book and would much prefer being able to do these things from the comfort of their own home.

System Requirements

This platform will run on an apache2 or nginx based server with PHP version 8.0 or better. The system requires MySQL (MariaDB) to be installed and composer as a prerequisite.

Server requirements will depend on site usage. For up to 100 users concurrently, The following is recommended:

- Ubuntu Server 20.04 LTS or better. (Other OS's that run Apache or Nginx may work but are not guaranteed)
- DEV Environments can utilize MAMP or LAMP.
- 1 vCPU
- 1GB Ram
- 20GB Disk Space
- SSL is recommended.

System Features

Index/Home Page:

The homepage of the library will include a simple user interface for users to log in and manage their library account. Without logging in users can still view the catalog, search for books, and see book recommendations right from the home screen.

Admin Panel:

The unique admin panel gives system administrators quick and easy control over managing the site content for their users. Various types of information can be administered from this panel including:

- Users
- Publishers
- Authors
- Catalog Content
- And more!

Catalog Page:

The catalog page aims to simplify the library experience for the user. Instead of walking around aisles with 100s of books to look at, users can simply search for the book their looking for or use one of our various filters to pinpoint the book they are looking for. From there the user can easily check the book out and move on with their day. This provides a lot of ease-of-use aspects, to provide users with a convenient and engaging experience.

Book Page:

When it comes to the book details page, the customer will be able to see easy details of the book front and center. The picture of the cover, brief description/synopsis. There will be identifiable information such as Author, Genre, ISBN, Publisher, and Price information. From here the customer can check out the book instantly or add to cart to continue looking for more books, or click the back button to return to the previous screen to search for other books.

Cart:

The cart portion of the site is all about being helpful to the user. This page will store all cart data for the customer, and this will be a convenient way for the user to manage their cart data. From here the user can check out or modify their cart by removing or reviewing quantity information.

History:

This will be a transactional history page, here the customer will have information regarding the books they've checked out in the past as well as have information on when the book is due back.

Typical Customers

- Librarian
- IT Administrator
- Developers
- Library Customers (Main Users)

Project Planning and Development

Hardware Requirements

- Ubuntu Server 23.10 with Apache2, PHP 8.0, Composer, MariaDB
- Virtual Server with 1vCPU, 2GB Ram, 20GB Storage utilizing VMware Server
- MariaDB will be locally installed and utilized.

Network Requirements

- 100mbps symmetrical required, 1gbps preferred.
- Cat5e Cables or better.
- Virtual Network Adapters with VMware Server

Project Development Plan

- Week 1-2: (Current Step)
    - Determine framework of application and establish outlines and structure. Creating IDE Environments and connecting Environments together, GIT Repository set up.
- Week 3-4:
    - Building the system login page for Users, Administrators, and Librarians
- Week 5-7:
    - Add functionality to add/edit/delete entities via the administration panel. Provide edit buttons on main pages for quick editing for Employees.
- Week 8:
    - Testing Phase, Record Demonstration and present for Mid-Term
- Week 9-11:
    - Gather User Experience data and provide improvements to the current feature set based on customer feedback. Continue working on implementing the basic features for the system:
        - Book Filters
        - Recommendation Algorithms
        - Admin Panel Development
        - History Page
        - Session Refinement
- Week 12-14:
    - Prepare test case information for User Acceptance testing. Provide optional developmental enhancements as time permits.
- Week 15:
    - Record Demo of final project.

System Requirements

Functional Requirements

No.	Priority Weight	Description
REQ-1	High	Database Should be able to store up to 100,000 books.
REQ-2	High	There should be a Login, Register, Catalog, and Admin page.
REQ-3	High	If two people check out the book at the exact same time, the system should be able to handle that scenario.
REQ-4	High	There should be a recommended books section on the front page that recommends new books to the users.
REQ-5	High	Errors should be handled in a friendly manner.
REQ-6	High	Customers should be able to conduct the entire transaction online

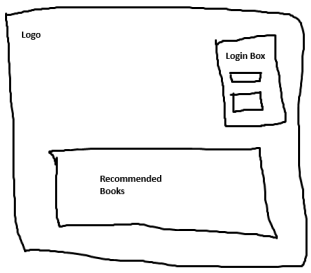
REQ-7	Medium	The website should be available on Desktop and Mobile.
REQ-8	Low	There should be a filtering aspect of the catalog allowing users to easily filter through all the books in the catalog.

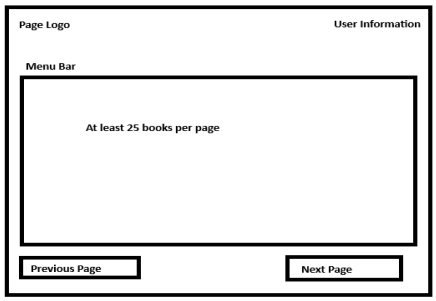
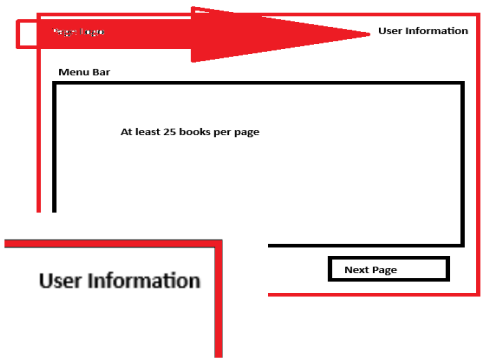
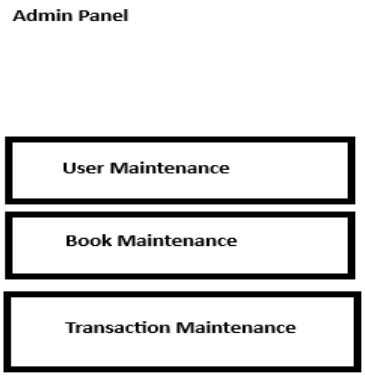
Non-Functional Requirements

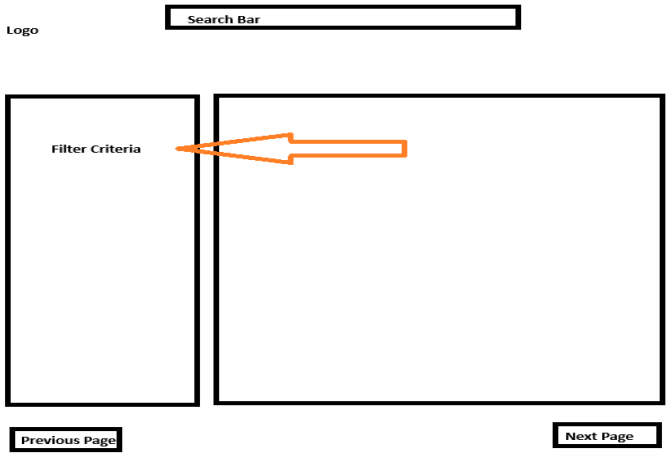
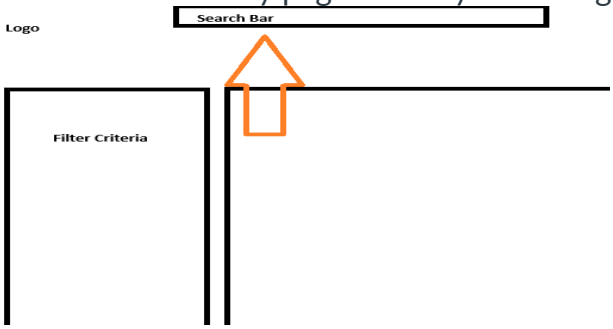
No.	Priority Weight	Description
REQ-1	High	System Should be Scalable
REQ-2	High	System should have 99.9% reliability
REQ-3	High	System should have necessary support.
REQ-4	High	System should be able to handle at least 50 users at one time.

REQ-5	High	System should be secure from the latest cybersecurity attacks.
REQ-6	High	UI should be easy to use and to understand for the customer

User-Interface Requirements Requirements

No.	Priority Weight	Description
REQ-1	High	Home Page should be a Login Screen, and have Recommended books section. 

REQ-2	High	Catalog should be able to show at least 25 books  per page.
REQ-3	High	System should display user information in the top right hand corner 
REQ-4	High	Admin Panel should have buttons for each maintenance Item 

REQ-5	Medium	Book List will have filters to aid in finding books. 
REQ-6	Medium	Search Bar on every page for easy searching 

Sequence and Activity Diagrams

Late Payment

States

- Pending Late Charge: The charge hasn't been charged yet because the book isn't late yet.
- Charged: The late payment is now charged because the Current Date > Due Date
- Billed: The late payment is now invoiced to the customer.
- Past Due: The late charge is past due.
- Collect: The invoice is > 60 days in age past Due Date and customer is no longer able to check out books until debt is collected.

Actions

The customer checks out a book, and then the late payment process begins on that book. The charge does not actually charge until the current date > due date and the book has not been returned yet. There are automated monitoring processes in place that check dates and will pause the account from being active until all debts are paid.

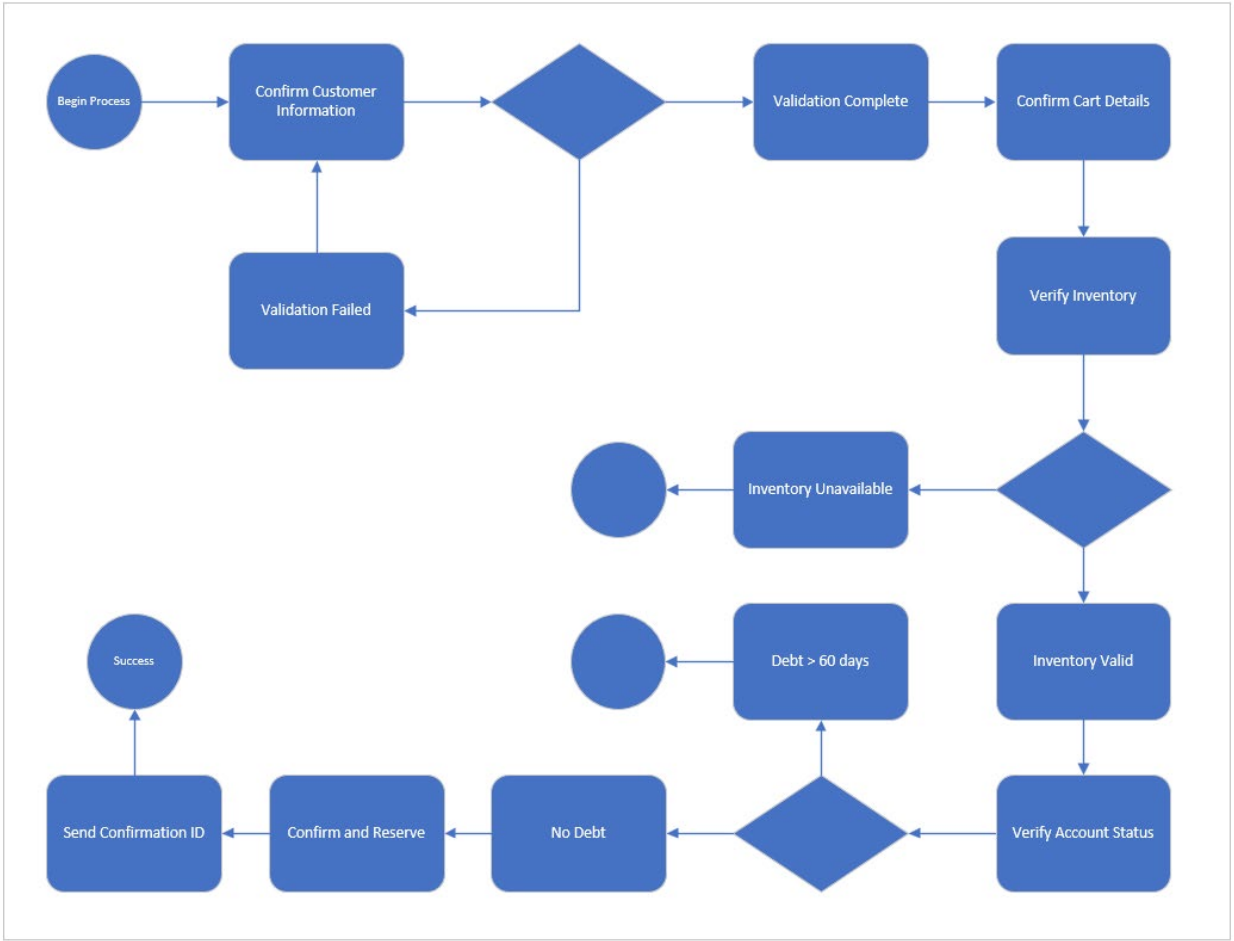
Reserve Book

States

- Initial State: Customer initiates the check out process.
- Final State: 1. Customer fills out the information required and proceeds to check out. 2. The customer is unable to check out books due to debt.

Actions

The customer browses for a book and requests to check out the book online. If the book is available, the system proceeds with the check out and if the customer does not have any debt associated with the account that is > 60 days the rental is processed.


Begin Process
Confirm Customer Information
Validation Complete
Confirm Cart Details
Validation Failed
Verify Inventory
Inventory Unavailable
Inventory Valid
Debt > 60 days
Success
Send Confirmation ID
Confirm and Reserve
No Debt
Verify Account Status

Sequence Diagrams

Add New Book to Inventory

Actor: Librarian, Administrator

Objects: Book, Library System

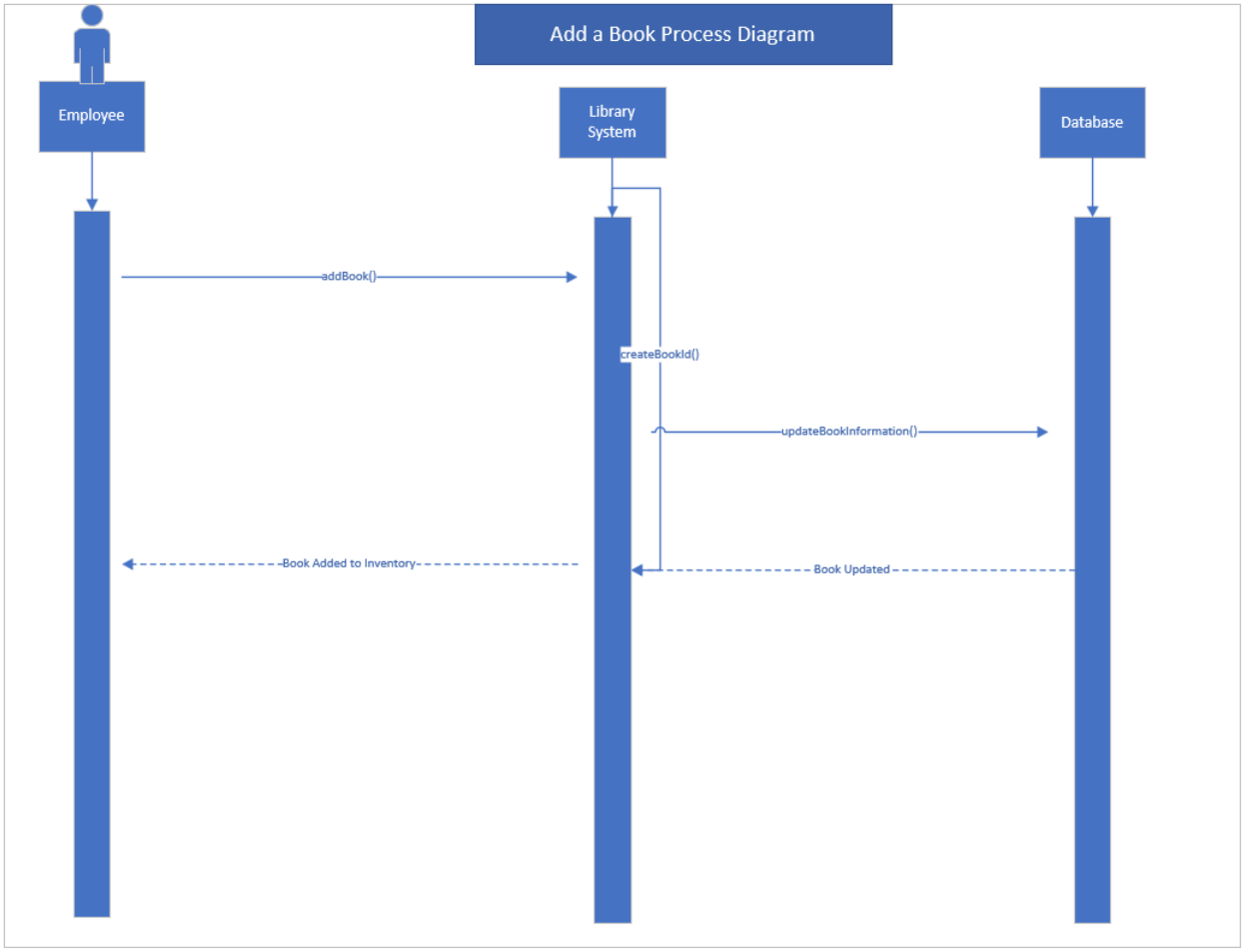
1. Employee adds book to system utilizing administrative panel.
2. The library system will create new book ID.
3. Employee fill out the information for the book.
4. The system validates the book details and saves the book in the database.

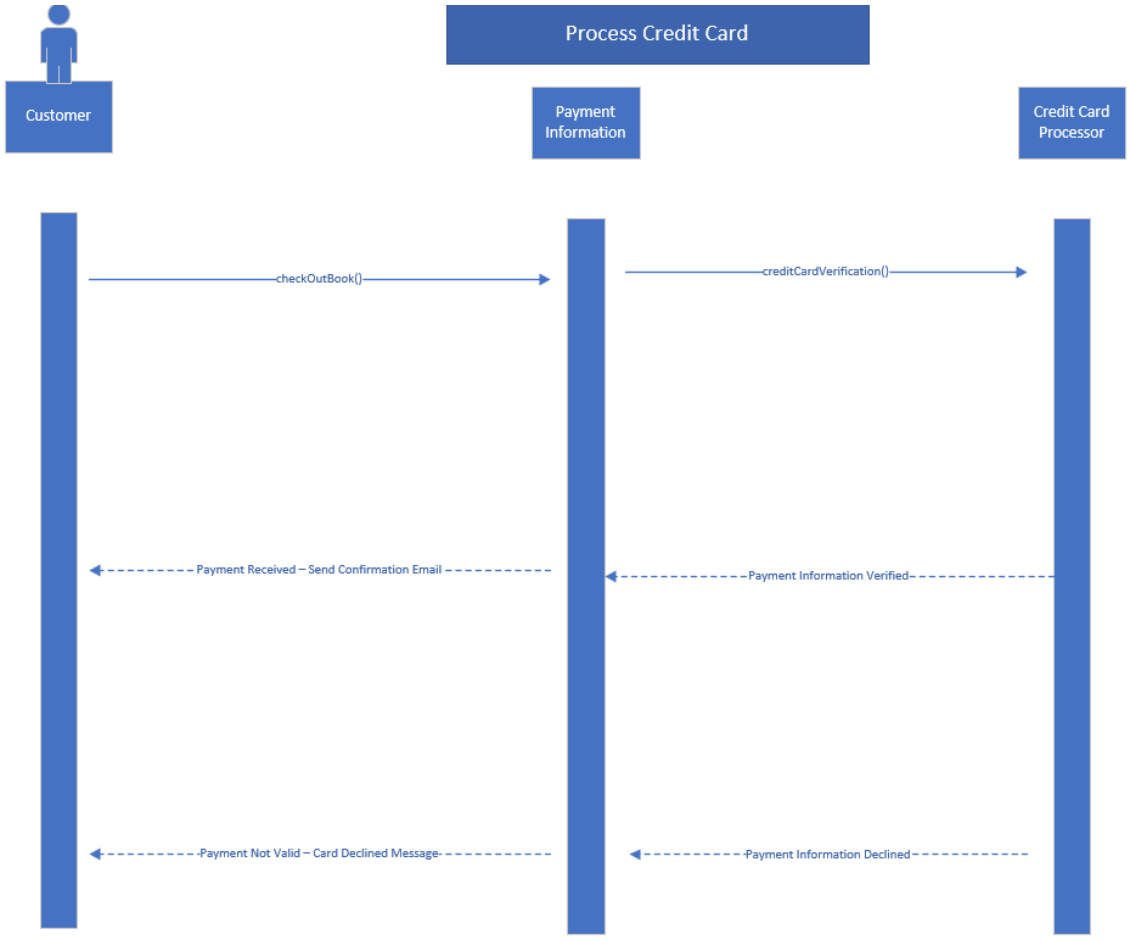
Process Credit Card

Actor: Customer

Objects: Payment Processor, Credit Card

1. Customer enters information on payment screen
2. Payment processor received credit card information and begins to verify payment details.
3. If information is unable to be validated, credit card denied message appears.
4. If information is validated, confirmation is sent to the customer via email and success page displays.


Add a Book Process Diagram
Employee
Library System
Database
addBook()
createBookId()
updateBookInformation()
Book Added to Inventory
Book Updated


Process Credit Card
Customer
Payment Information
Credit Card Processor
checkOutBook()
creditCardVerification()
Payment Received – Send Confirmation Email
Payment Information Verified
Payment Not Valid – Card Declined Message
Payment Information Declined

Functional Requirements

Stakeholders

- Business Owners: Can run reports and manage business finances.
- Managers: Help manage employees and customer issues with admin access.
- The Public: This stakeholder can utilize this system to easily check out books in a more convenient manner.

Primary Actors

- Customer: This actor is essentially the public who can check out books from the comfort of their home and pay all their dues easily from a web interface without going into the library.

Secondary Actors:

- Admin: This person can add, remove, edit all entries within the library site as well as perform necessary modifications when issues arise.
- System: This is required for the operation of the site and is responsible for all the calculations and management of books, inventory, sales, transactions, and historical data.

Use Cases

Here we list out all the different responsibilities for all the use cases and how long it will take the engineers to develop each portion of the use case. In these estimates, 2 indicates a full days' worth of work.

Admin (total: 14)

- Add/Modify book: This is to add a new book (2)
- Add User: This is to add a new user to the system (2)
- Add/Modify late fees: This is to add or modify late fees added to books (4)
- Login/Logout: This is to securely log in and out of the site, the site will automatically detect admin account and show required dashboard upon login (4)
- View Historical Information: To view information such as check outs, check ins, etc. (2)

Customer (total: 16)

- Check Out Book: The ability to check out a book (4)
- Pay Late Fees: Gives customer a breakdown of fees and ability to pay online (4)
- Credit Card: Add the ability to pay by card (2)
- PayPal: Add the ability to pay with PayPal (2)
- Return Book: Add the ability to return the book (4)

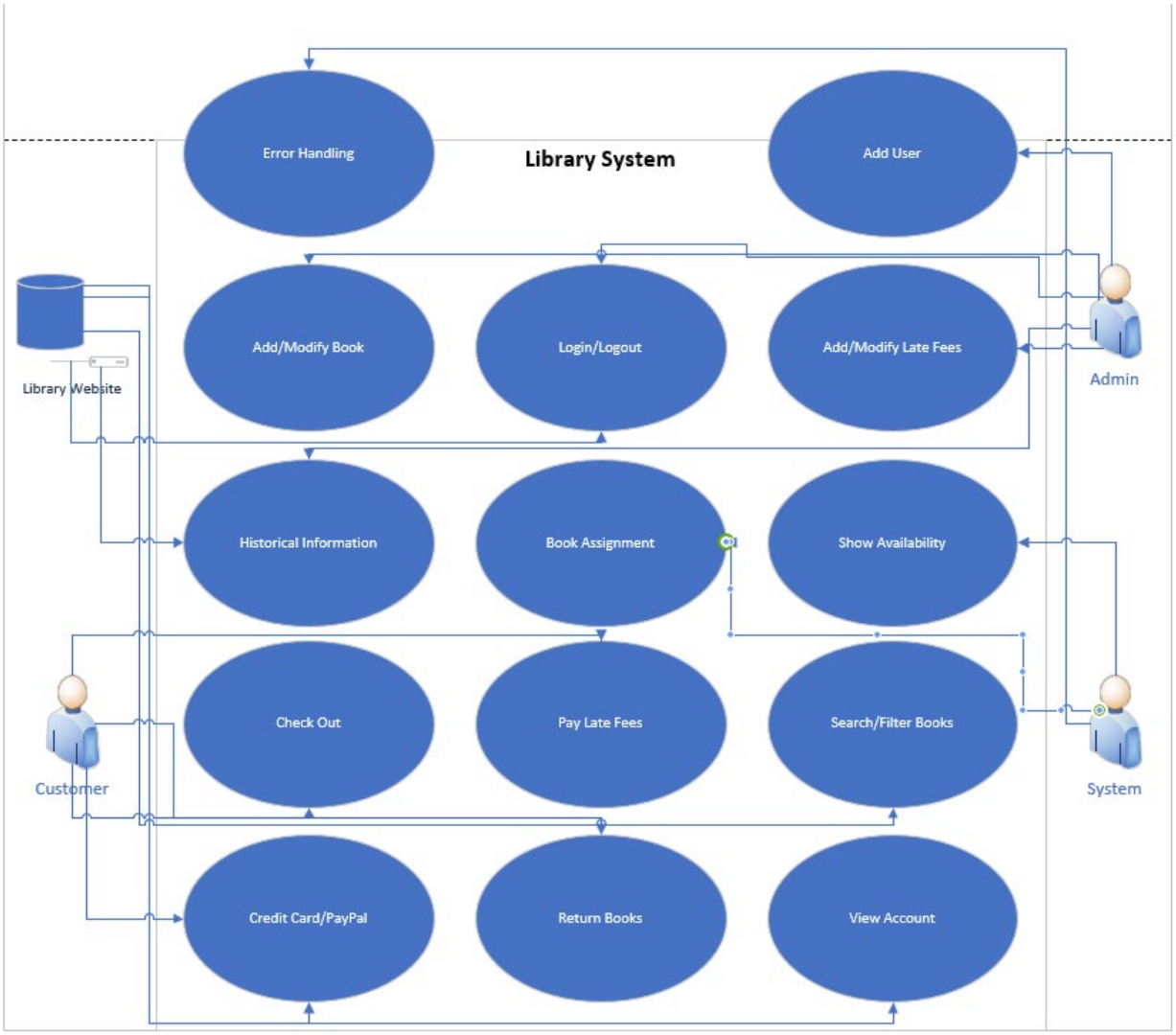
Library Website (total: 16)

- Transactional History: Keeps a record of all transactions made by account (4)
- Login/Logout: The ability to login or logout via the site (2)
- View Account: The ability to see late fees, checked out books, and other information related to account (2)
- Search Book: The ability to search for a specific book (2)
- Filter Books: The ability to filter all books (2)
- Credit Card: Add the ability to pay by card (2)
- PayPal: Add the ability to pay with PayPal (2)

System (total: 8)

- Assigning books to users: ensure book has not been checked out before letting transaction through (4)
- Show available/unavailable: The ability to show if a particular book is available or not (2)
- Show friendly errors: Show a friendly error to the user when necessary (2)

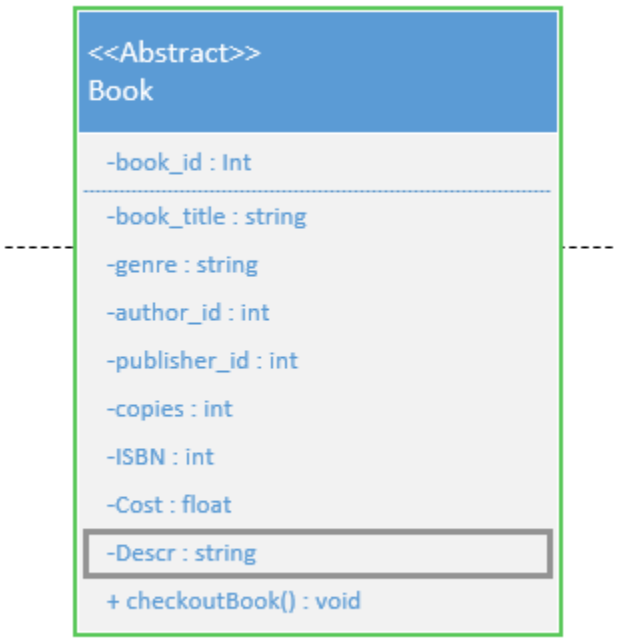
Use Case Diagram



Class Diagram

Book

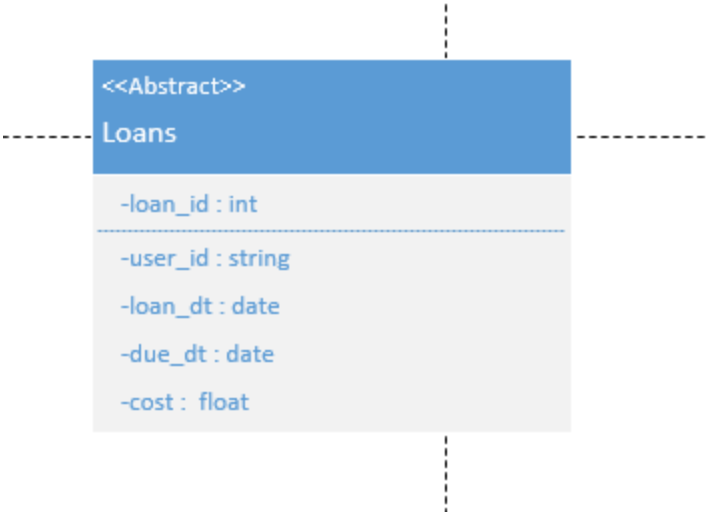
In the library system, we are only to hold book type items, and we provide a method in order for customers to be able to check the books out based on availability.



User

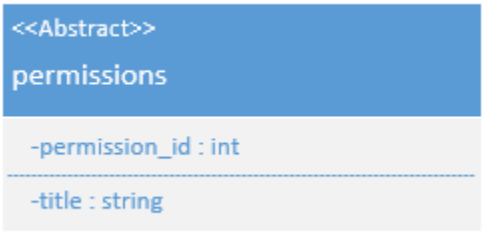
We have 3 types of users here, which are basic user, employee, and admin users.

Loans

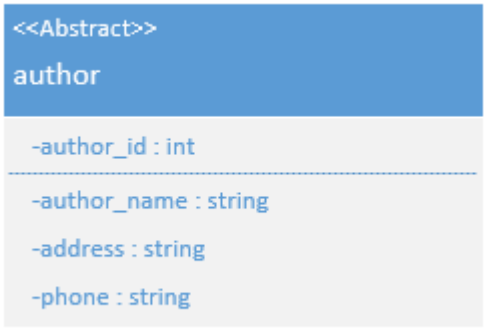


Permissions

Having a separate permissions table allows us to expand the permissions in the future if necessary. It is also reusable.

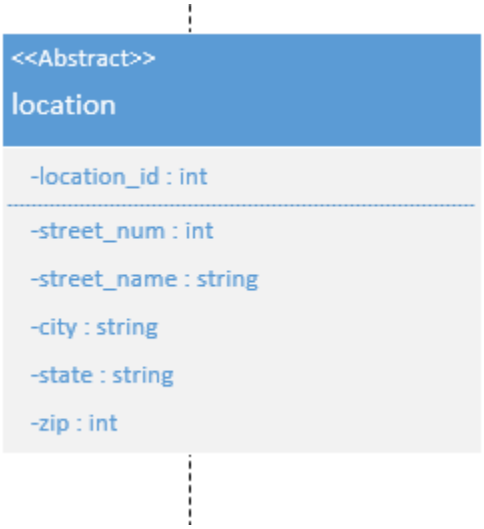


Author

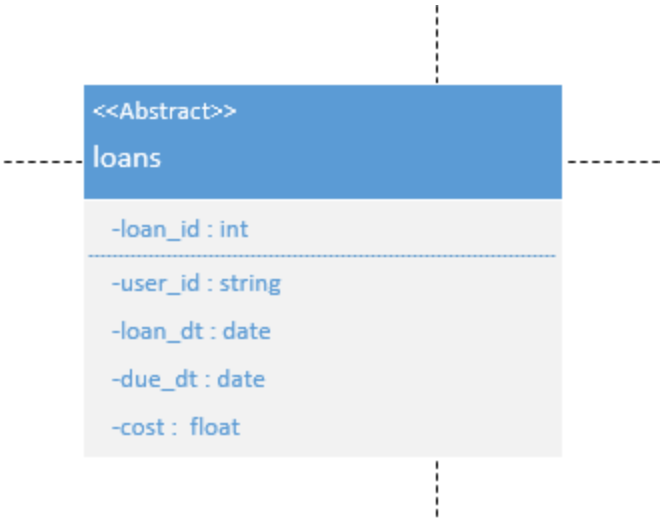


Publisher

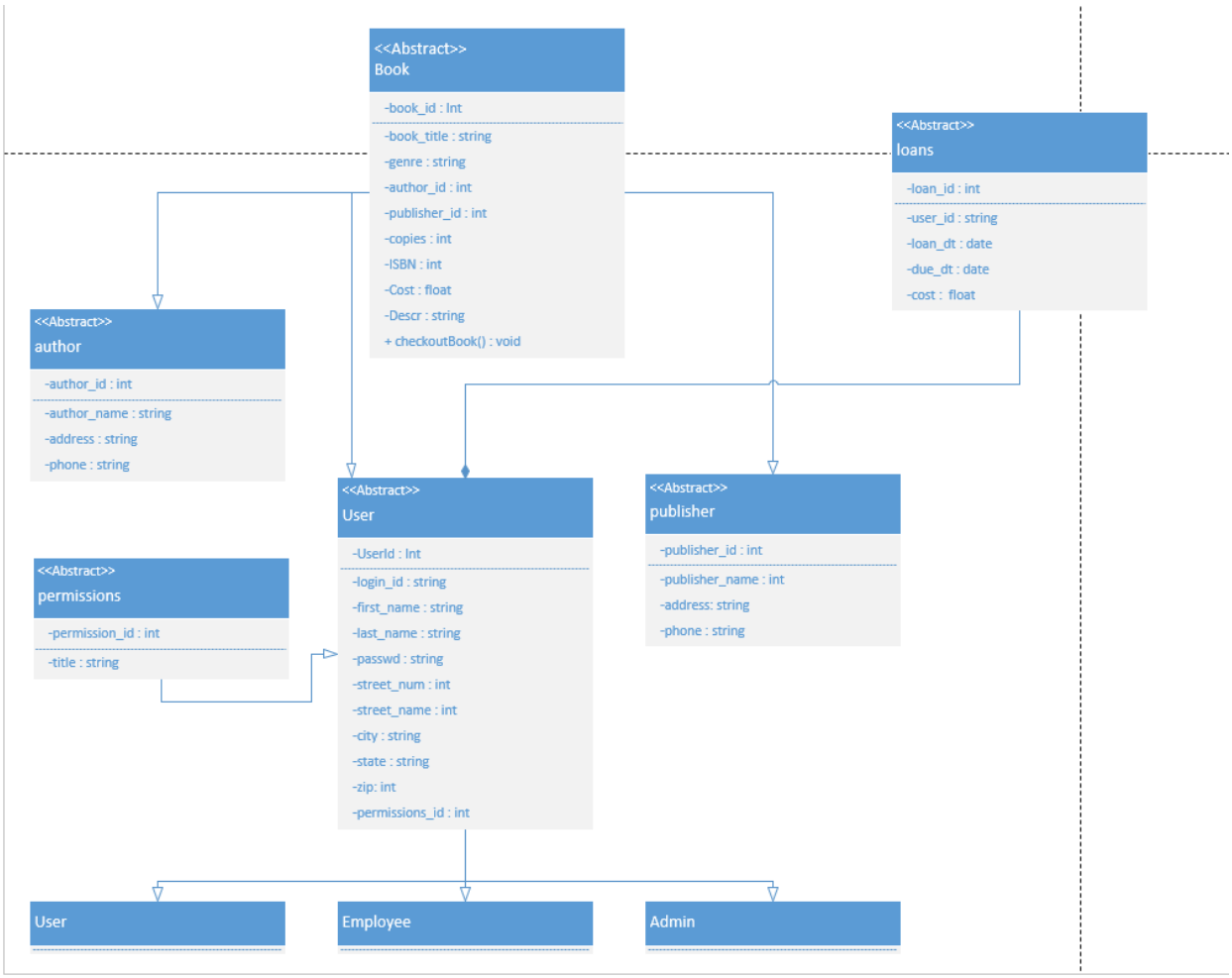
Location



System will utilize a transactional table, for each transaction (credit, debit, etc) there will be a row added and its foreign key will be the account ID of the user. This will be the Loans table in our case.



Complete Diagram:


<<Abstract>>
Book
-book_id : Int
-book_title : string
-genre : string
-author_id : int
-publisher_id : int
-copies : int
-ISBN : int
-Cost : float
-Descr : string
+ checkoutBook() : void
<<Abstract>>
loans
-loan_id : int
-user_id : string
-loan_dt : date
-due_dt : date
-cost : float
<<Abstract>>
author
-author_id : int
-author_name : string
-address : string
-phone : string
<<Abstract>>
User
-UserId : Int
-login_id : string
-first_name : string
-last_name : string
-passwd : string
-street_num : int
-street_name : int
-city : string
-state : string
-zip: int
-permissions_id : int
<<Abstract>>
publisher
-publisher_id : int
-publisher_name : int
-address: string
-phone : string
<<Abstract>>
permissions
-permission_id : int
-title : string
User
Employee
Admin