Gabriel Agbo

Indiana University Kokomo

SPRING 2022

Virtual Library/Library Management System

Table of Contents:

I. Objectives:

Create a virtual library that will allow users to store, view, add and remove eBooks in a system. The mission of this program is to provide speedy and wide access to update information and save the time of its users or members. This will provide easy ways for librarians to create and store their library collections. It enables students and staff to communicate and collaborate through multiple channels.

II. System requirements:

- Only authentic user with a valid member card must have the access to the system.
- User must be able to:
  - ➢ Search for a book from database
  - ➢ Add new book to the database
  - ➢ Update a book's information
  - ➢ Rent and return a book
  - ➢ See the list of available books from database
- User must enter issue and return date in database.
- Same user ID cannot rent more than 5 books.
- User must sign a "DO NOT COPY DISCLAIMER" consent before checking out.

a. Functional Requirements

No.	Priority Weight (1 to 5)	Description
REQ-1 Login	5	Both Admin and members must be logged in before they modify any information.
REQ-2 Add Book	5	This will allow a library staff member to navigate in the

		system as an Admin and add a new book.
REQ-3 Delete/update book	5	This will allow a library staff member to navigate in the system as an Admin and remove/update a book.
REQ-4 Validate user account	3	When a new member signs up then he should wait for acceptance by Administrator according to library policies.
REQ-5 Delete member	3	Admin can delete a member due to some specific rules.
REQ-6 Rent book	5	This function will allow an active member of the library to borrow a book.
REQ-7 Return book	3	This function will allow an active member of the library to return a book.
REQ-8 User Search	5	This function will allow you to Search for a book from database.
REQ-9 See list of available books	3	This function will show you a list of books that are available for rent.

b. Traceability Matrix

Req't	PW	UC1	UC2	UC3	UC4	UC5	UC6	UC7	UC8	UC9
REQ-1	5	X								
REQ-2	5		X							
REQ-3	5			X						
REQ-4	3				X					
REQ-5	3					X				
REQ-6	5						X			
REQ-7	3							X		
REQ-8	5								X	
REQ-9	3									X
MAX PW		5	5	5	3	3	5	3	5	3
TOTAL PW		5	5	5	3	3	5	3	5	3

c. Nonfunctional Requirements

- Error handling
  - WLMS product shall handle expected and non-expected errors in ways that prevent loss in information and long downtime period.
- Performance Requirements
  - The system shall accommodate high number of books and users without any fault.
  - Responses to view information shall take no longer than 5 seconds to appear on the screen.
- Safety Requirements
  - System use shall not cause any harm to human users.
- Security Requirements
  - System will use secured database

- Normal users can just read information but they cannot edit or modify anything except their personal and some other information.
- System will have different types of users and every user has access constraints.

d. User Interface Requirements

- Library staff member/Librarian Model

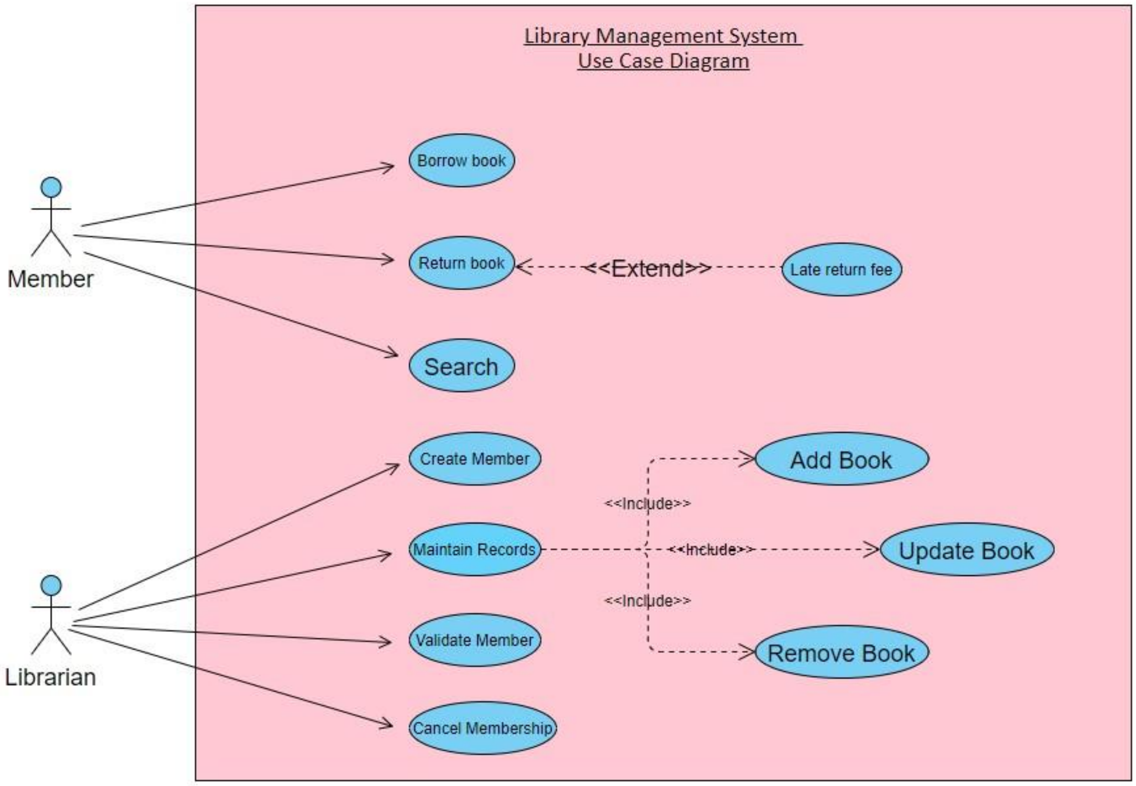
1. Login Page: Librarian model also has a login page in which he or she has to add their authorized ID in order to make the interface accessible.
2. Books List: Librarian has to understand what books are available in the library and which books are issued and which are accessible. It has a detail list of all books that are kept in the library catalog.
3. Books Issued: Librarian will be directed to another page which can have all the details of the readers who have issued a particular book.
4. Available Books: There is another list for the books that are available and student can get them issued without going through the period of waiting.
5. Reminder List: This list will keep a check on which students need to be reminded that their date to return the book is upcoming or otherwise they will have to pay a fee.

- Library Member/Student Model User Interface

1. Login Page: The reader has to enter its particular user ID and a keyword as password.
2. Book List: It includes all the available books in the library management system catalog.
3. Checkout Section: This section will display the book you've selected to rent. This will also display how many more books you can borrow or gives you an error when you are over the limit. And inform you about when are the books due for return based on the checkout date.

4. Reminder Section: This section comes with a warning saying and the date on which you have to return a book. This is very useful since it can help you avoid paying fees.

e. Use Case Diagram:

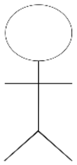
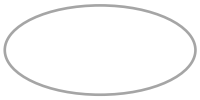


https://online.visual-paradigm.com/app/diagrams/#diagram:proj=0&type=ClassDiagram&width=11&height=8.5&unit=inch

Description:

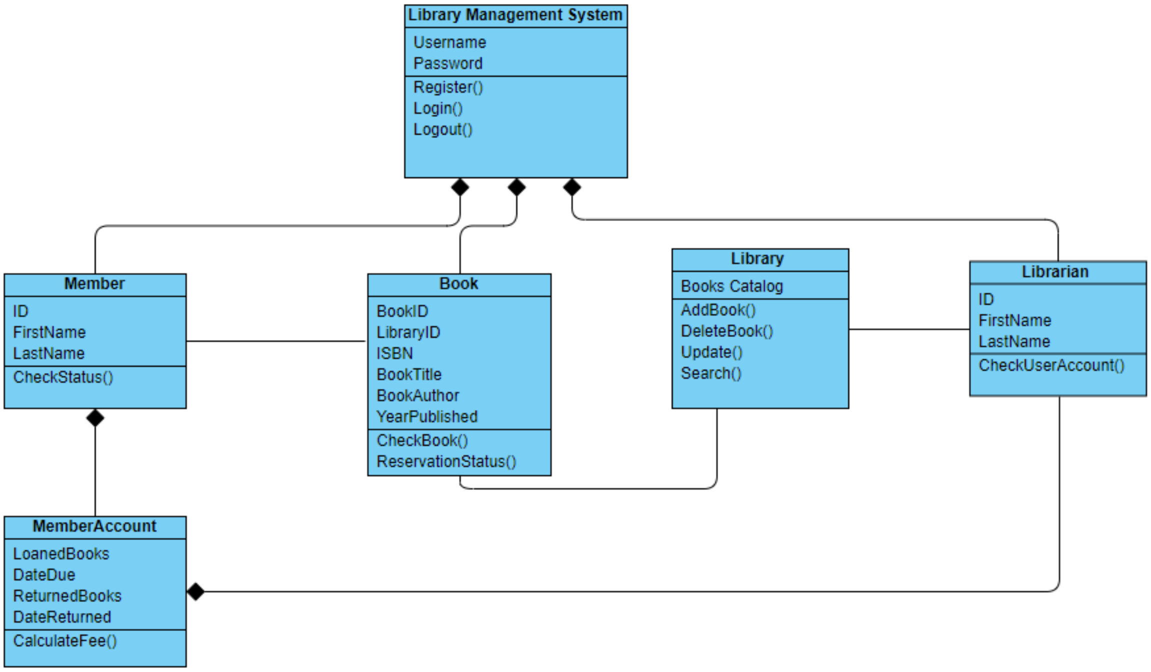
We have three main actors in our system:

1. Librarian: Mainly responsible for adding and modifying books, and users. The Librarian can also issue, reserve, and return book items as a regular member.
2. Member: All members can search the catalog, as well as check-out, borrow, and return a book.
3. System: Mainly responsible for sending notifications for overdue books, canceled reservations, and late return fee, etc.

	Actor This the representation of the librarian and Member that will launch actions in the program.
	Use case Users and processes that interact with information system.
———	Association Communication between actor and use cases.

----------- <<include>>--------→	Include Additional use case relations that are added to carry out functions.
←----------<<Extend>>---------	Extend Additional use case relations that are supplementary behavior extending use case.

f. Class Diagram / Data Model Structure:



https://online.visual-paradigm.com/app/diagrams

Description:

Main Classes of our Library Management System:

- **Library Management System Class:** The central part of the organization for which this software has been designed. It handles all transactions of the library.
- **Book:** The basic building block of the system. Every book will have ISBN, Title, Author, and Year Published.
- **Member:** It manages and stores the user's transactions.
- **Librarian:** Responsible for managing reservations and member's account status.
- **Library:** This class will store and manage all the books in the library.
- **MemberAccount:** This class will verify if the member is active and has a balance to pay off before borrowing a book.

Attributes of Library Management System:

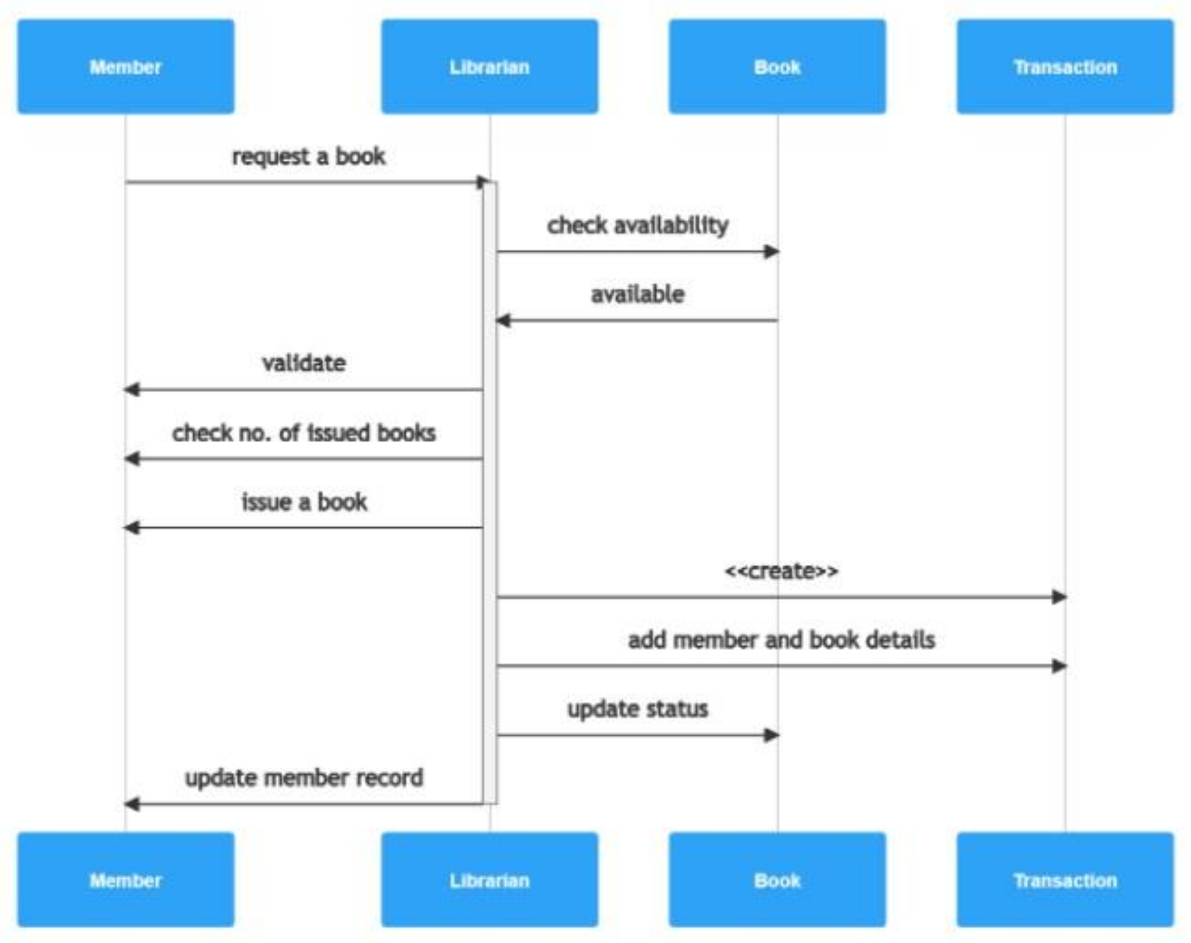
- **Library Management System Attributes** – Username, Password
- **Book Attributes** – BookID, LibraryID, ISBN, BookTitle, BookAuthor, Year Pubished
- **Member** – MemberID, FirstName, LastName
- **Librarian Attributes** – ID, FirstName, LastName
- **Library Attributes** – BooksCatalog
- **MemberAccount Attributes** – LoanedBooks, DateDue, ReturnedBooks, DateReturned

Methods of Library Management System:

- **Library Management System Methods** – Register (), Login (), Logout ()
- **Book** – CheckBook (), ReservationStatus ()
- **Member** – Checkstaus ()
- **Library** – AddBooks (), DeleteBooks (), Update (), Search ()

- **Librarian** – CheckUserAccount ()
- **MemberAccount** – CalculateFee ()

g. <u>System Sequence Diagram:</u>

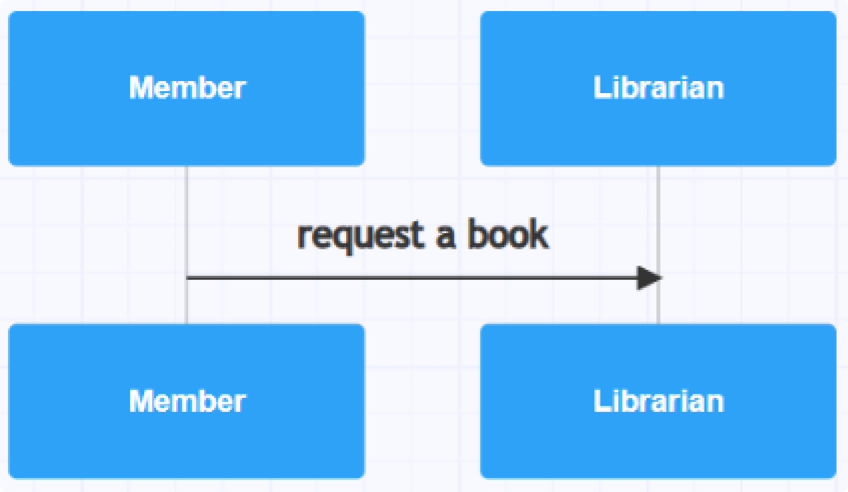


https://app.gleek.io/diagrams/gNP8sUvCtvQ5Dx0cbSNQkQ

<u>Description:</u>

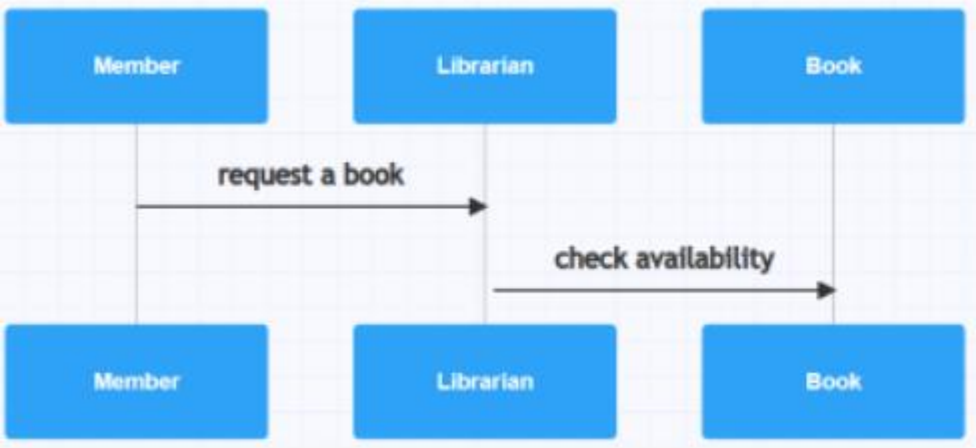
- **Member-request a book -> + Librarian:**

  We are going to break this transaction down and describe the different interactions when a Member launch the application by requesting or looking for a book to rent from library.



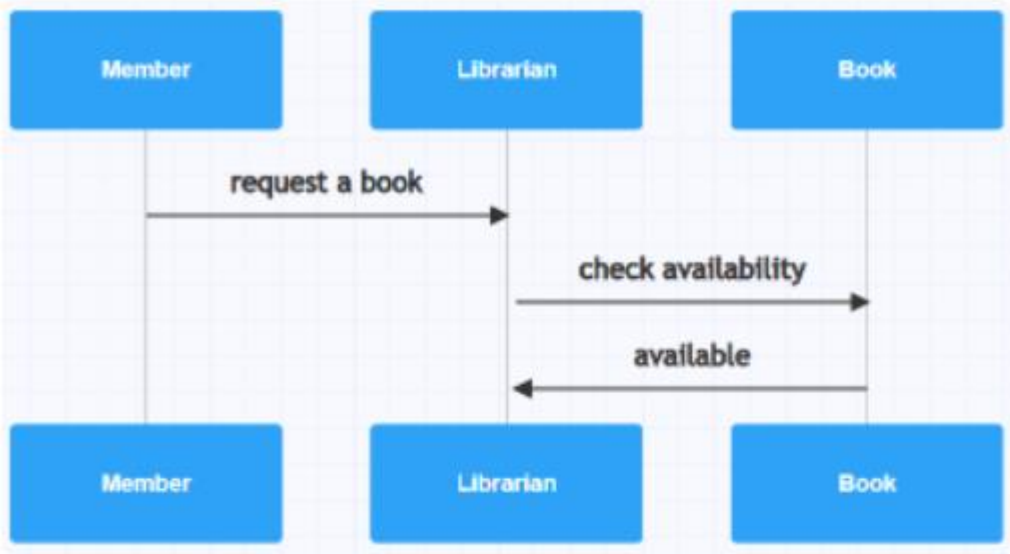
- **Librarian-check availability -> Book**

  The librarian will check the system to see if the book is available. This means an interaction with the Book object in the library management system. Again, we'll use a labeled arrow and the Book object will automatically be created.



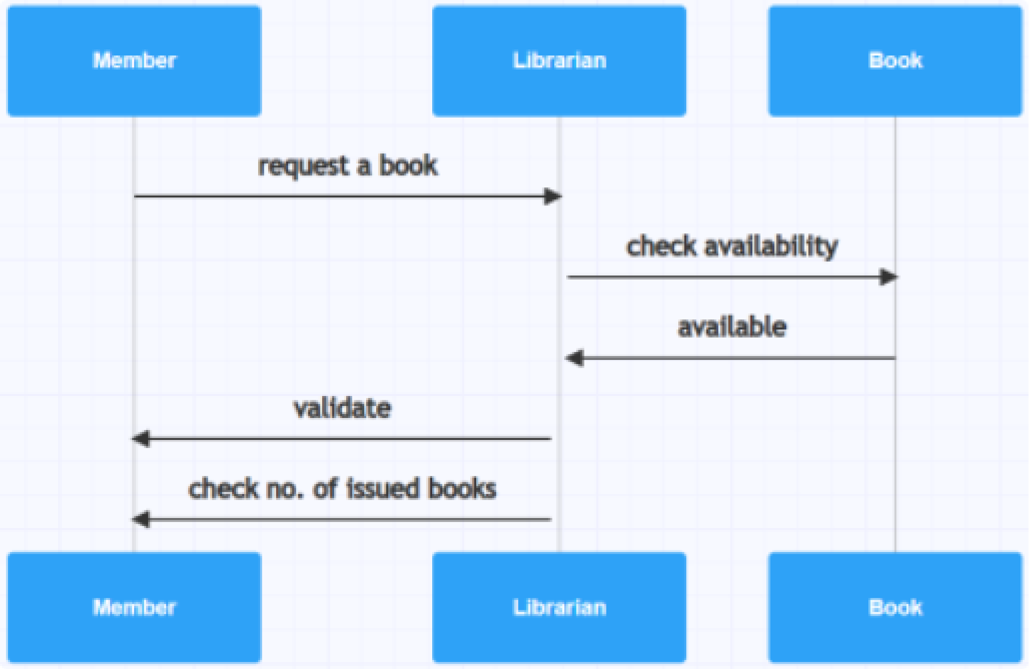
- **Book-available -> Librarian**

If the book is available for the member to rent, the Book object sends a message back to Librarian object for notification.



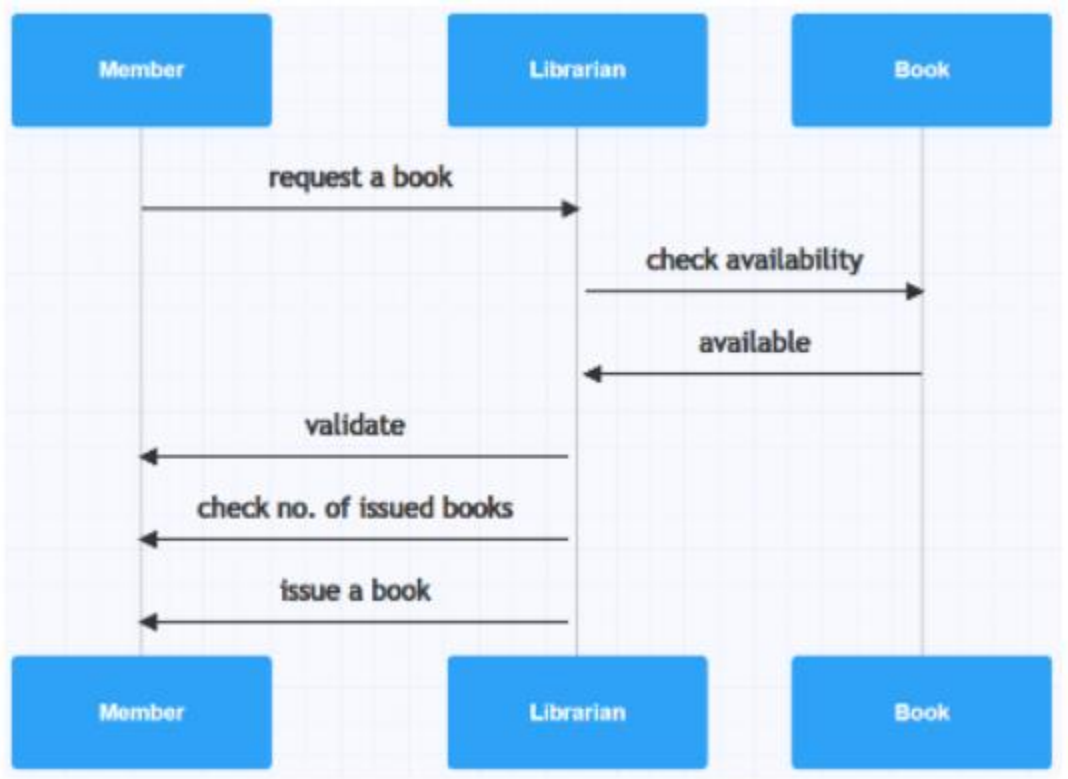
- **Librarian-validate -> Member, and Librarian-check no. of books -> Member**

  The librarian will now have to validate the member request by checking two things, the membership status and also check the number of books already rented from library.



- **Librarian-issue a book -> Member, and Librarian-add member and book details -> Transaction**

  If everything checks out fine, librarian will give member the authorization to rent a book.

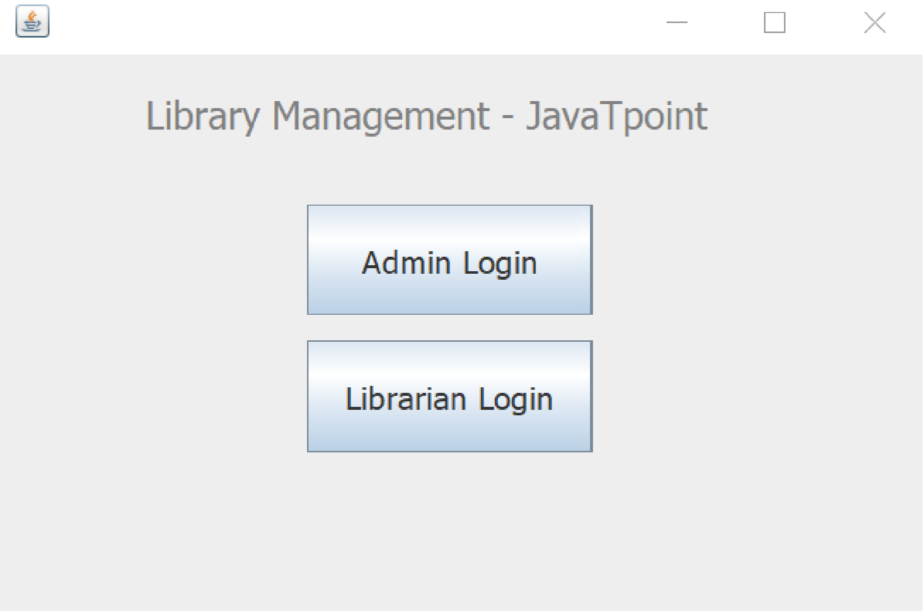


- **Librarian-update member record -> Member**

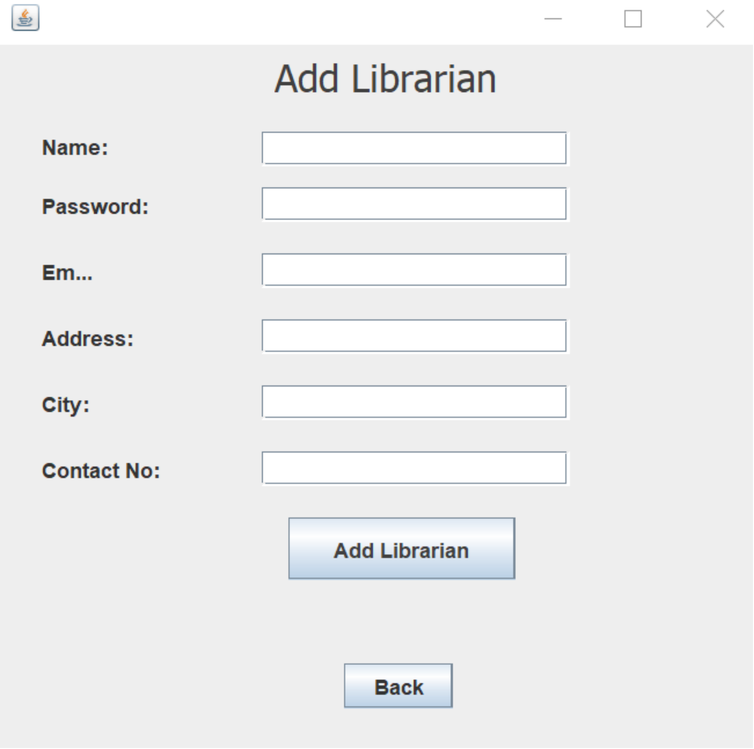
  Librarian next needs to update our system's database, so the rented book isn't available until returned. The librarian also needs to update the member's record, to add rented additional book. That gives us the final lines in our simple library management sequence diagram as shown above.

h. User Interface Specification:

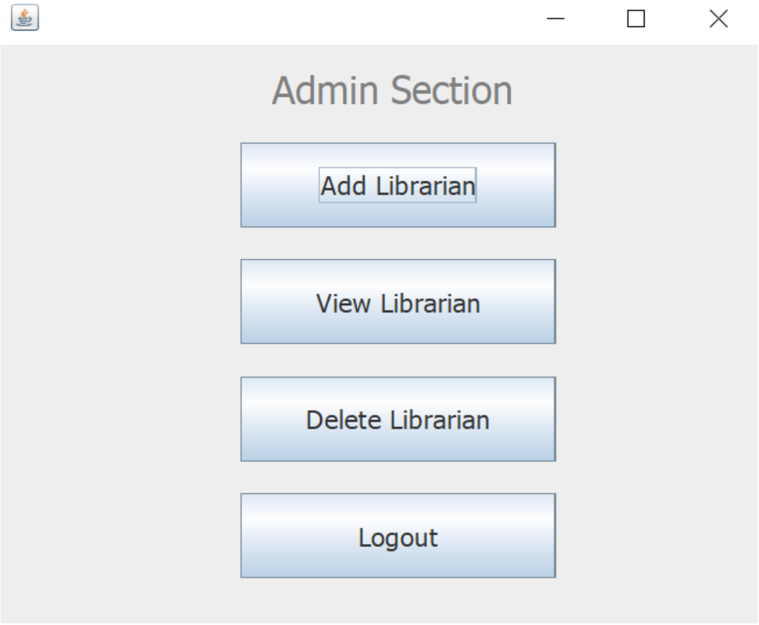
1. Login



2. Add Librarian



3. Admin Section:



i. User Effort Estimation

Based on the prototype we have designed. We can confirm that it will only take about 3 to 4 clicks for a member to borrow a book after login in the system. This is very important for the development of this system. Because it shows how user friendly is the system.

III. Typical customers:

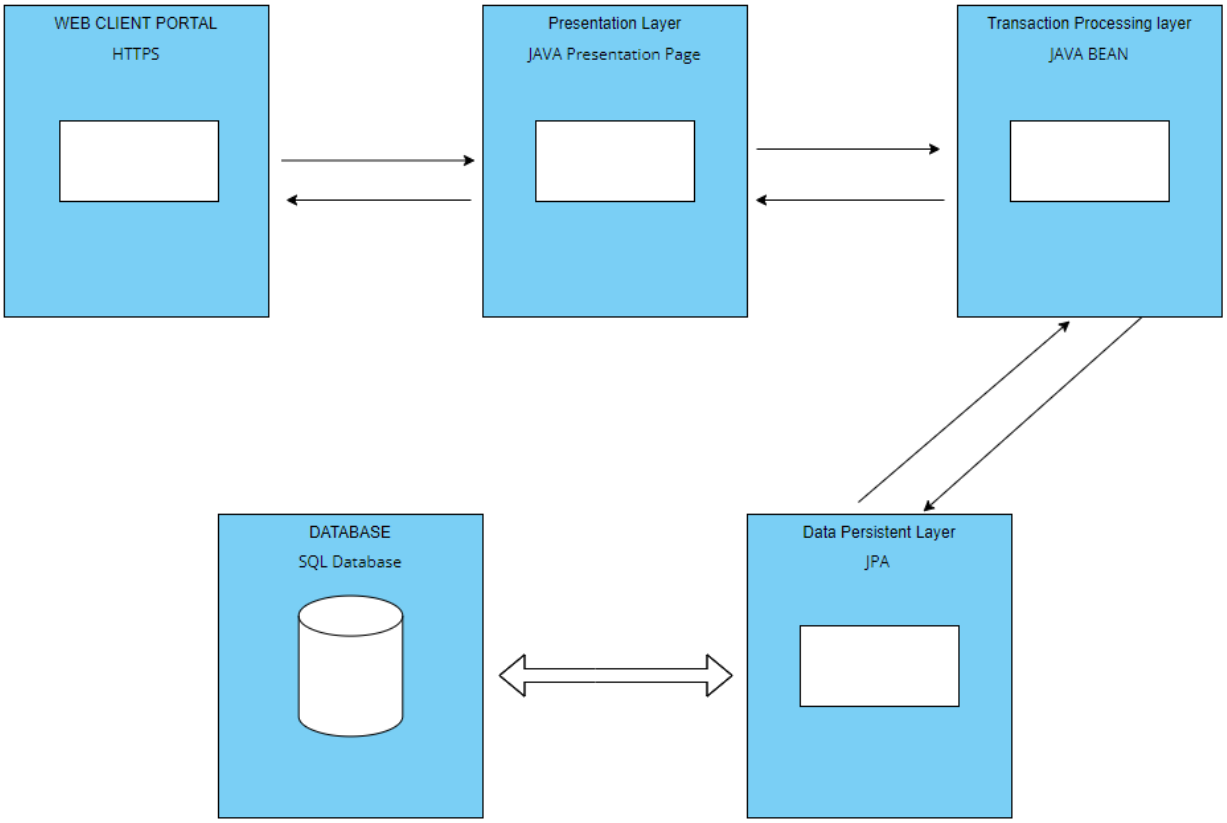
This system will be proposed to public schools and public library in order to facilitate the books renting process for students. This system will be used by students and the library staff. Students are going to be the customers that will rent, return, and browse in the system to find a book in the system. And the library staff will be the special users that will maintain the database by adding/removing/updating books and create/remove library members (students). This also means that they will need an administrator access that will allow them to have functions listed above than the members.

IV. Customer Problem Statement:

- Libraries are used to store books, but require a system to navigate to a specific book or specific content within a book. A library database system is an infrastructure that allows users to search books and book content, add/remove, and download selected books.
- The problem faced is that library users require an efficient method to find a specific book or keyword(s) within a book given a continuously expanding library. Efficiency requires that the processing time should stay relatively the same even as the library contents increases.

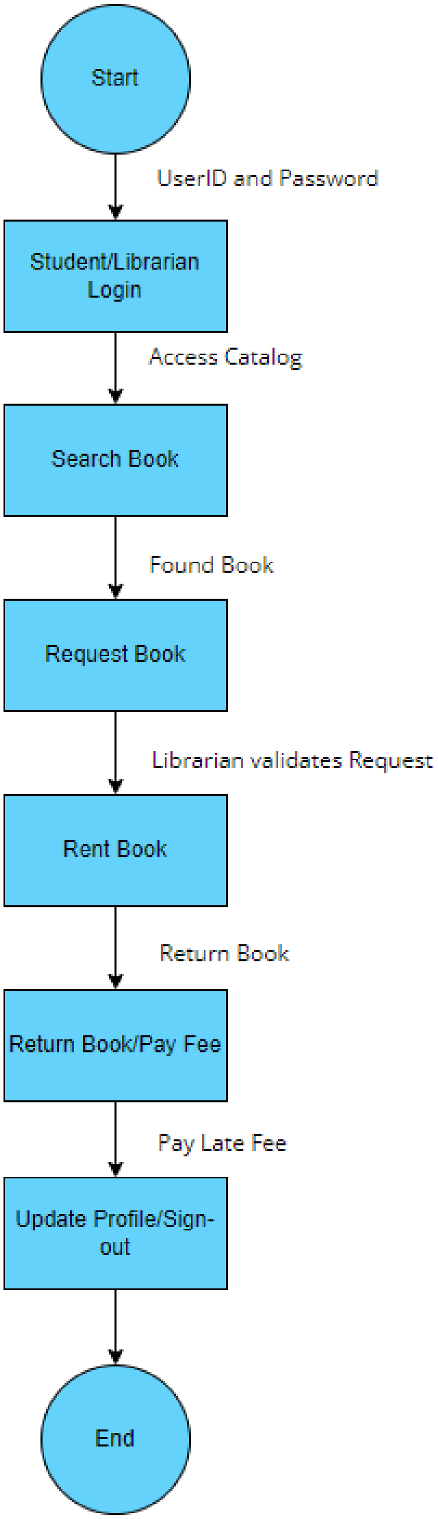
V. Interaction Diagrams

- Diagram 1: **This is an illustration of the Application's System Layers.**

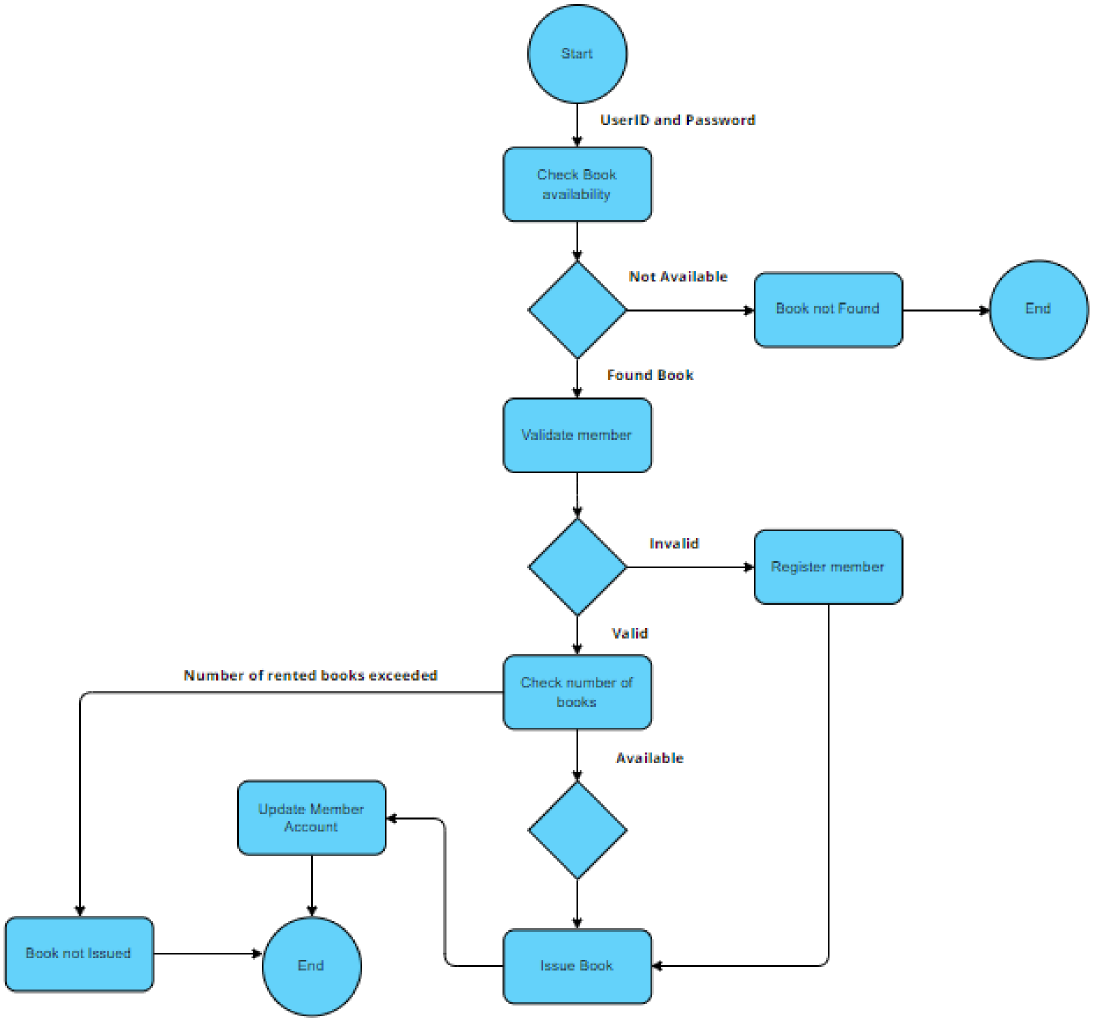


- Diagram 2: **This diagram shows the process of renting a book after you log in to the system**

-



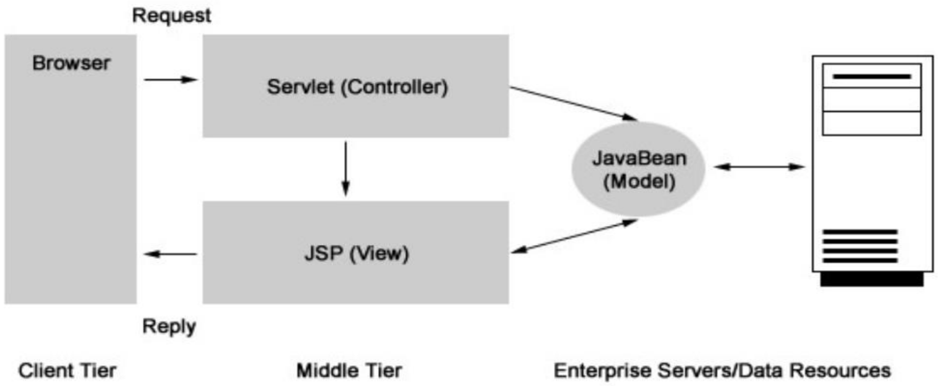
- Diagram 3: **This diagram shows the full activity process of the program when renting a book**



VI. System Architecture and System Design:

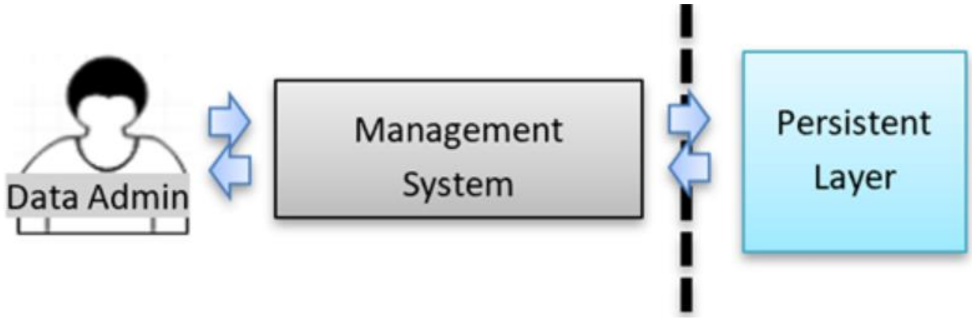
a. Architectural Styles

With the rapid increase of the network applications, MVC paradigm is a very advanced design idea for Web application development. Which should be used for the architecture of this system we're designing. By using MVC we will separate the input, processing, and output of an application according to the Model, View and Controller, to make it a three-tier design mode (namely model layer, view layer and control layer).



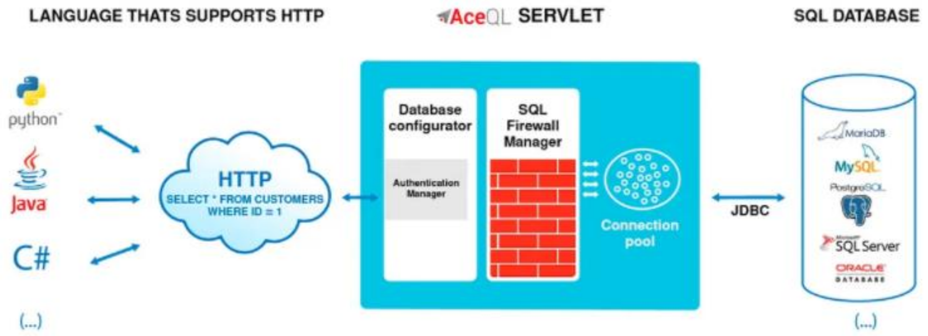
b. Persistent Data Storage

This Library Management System will use an extra server computer in a different location for persistent data storage and backup over the internet. This will be used as a cloud storage which is very common nowadays.



c. Network protocol

The communication protocol we will use for this system will a JAVA JDBC for the program since it is written in JAVA language and also this will provide a complete set of interfaces, allowing portable access to the underlying database, so different types of executable files can be written in Java as well. But also, the HTTPS protocol will be used to provide secured internet connection if we decide to add a website as an extension to existed application for the users.



VII. <u>User</u>

<u>Interface Design and Implementation</u>

I am still trying to implement the screens mock-ups designed in report #1 but I don’t think that I have the necessary tools and experience in order to complete that task. But I plan on using a similar interface design for the program with the same time effort as well.

X. Design of Tests

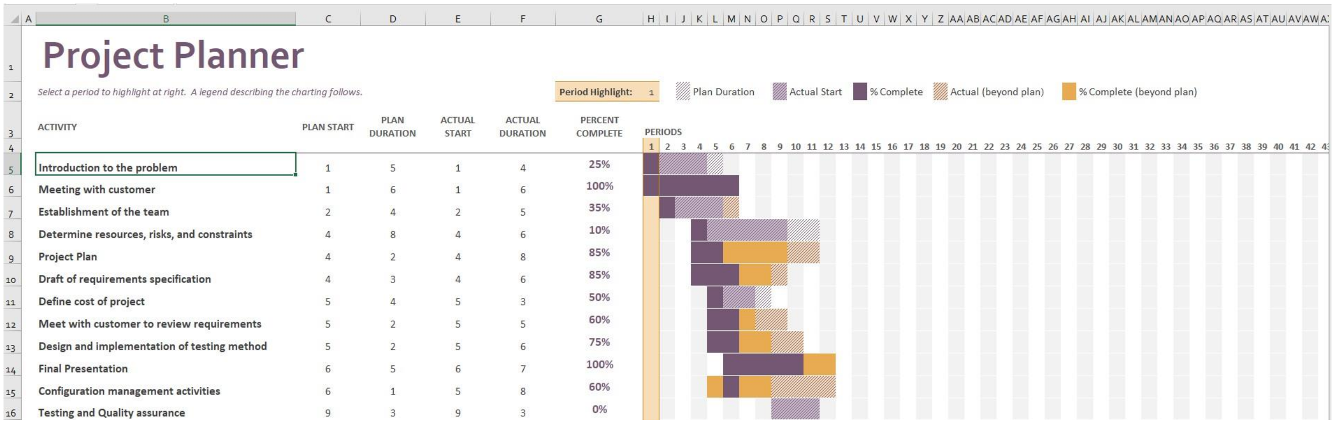
Test Case No:	REQ	Test Case ID	Description	Status
1	REQ-1	UC1	LOGIN	Pass
2	REQ-2 & REQ-3	UC2 & UC3	ADD/UPDATE/DELETE BOOK	Pass
3	REQ-4 & REQ-5	UC4 & UC5	VALIDATE/DELETE USER ACCOUNT	Pass
4	REQ-6 & REQ-7	UC6 & UC7	RENT/RETURN BOOK	Pass
5	REQ-8 & REQ-9	UC8 & UC9	USER SEARCH/LIST AVAILABLE BOOK	Pass

IX. Project planning and development approach:

This system will be developed with Visual Studio from Microsoft in .NET, C++, and the support of SQL databases. This will also require a Windows server with all the specific required capacity we will develop later. And a high-speed internet to make it live and accessible through a website as well it is also very important to have one extra server for backup.

X. Development plan:

Excel sheet



Project Planner

Select a period to highlight at right. A legend describing the charting follows.

Period Highlight: 1

▨ Plan Duration ▨ Actual Start ■ % Complete ▨ Actual (beyond plan) ■ % Complete (beyond plan)

ACTIVITY	PLAN START	PLAN DURATION	ACTUAL START	ACTUAL DURATION	PERCENT COMPLETE
Introduction to the problem	1	5	1	4	25%
Meeting with customer	1	6	1	6	100%
Establishment of the team	2	4	2	5	35%
Determine resources, risks, and constraints	4	8	4	6	10%
Project Plan	4	2	4	8	85%
Draft of requirements specification	4	3	4	6	85%
Define cost of project	5	4	5	3	50%
Meet with customer to review requirements	5	2	5	5	60%
Design and implementation of testing method	5	2	5	6	75%
Final Presentation	6	5	6	7	100%
Configuration management activities	6	1	5	8	60%
Testing and Quality assurance	9	3	9	3	0%

a. Project size estimation based on use case points

Based on this report, our project doesn’t require a big group of professionals. But we must be sure that the project team consists of the minimum experts and experience professionals, such as:

- Project Manager
- .Net and C++ developer
- SQL developer
- UX/UI designers
- Database Expert
- IT technician

b. Plan of Work:

This group of professionals will basically work on developing the system in a SLDC model stages as shown below.

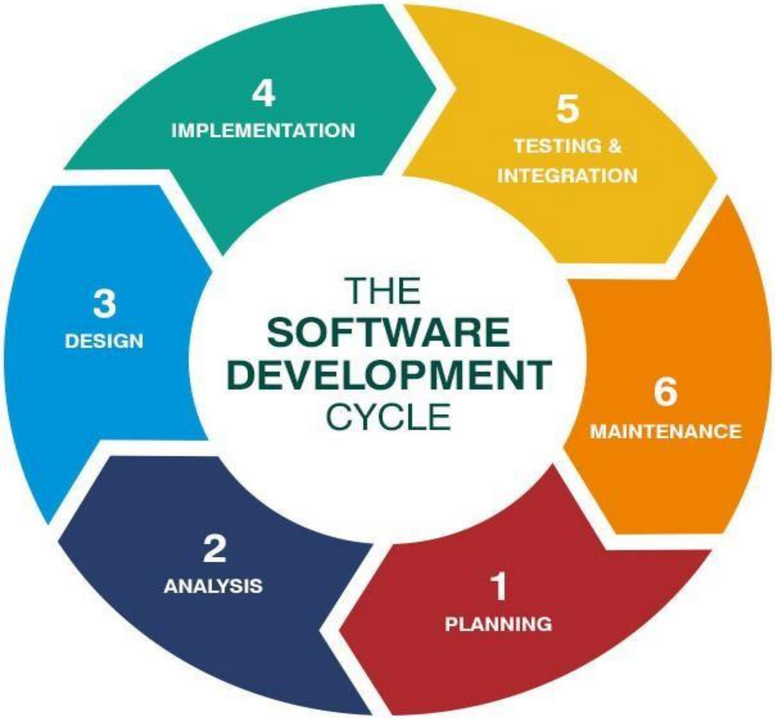


Image from synotive.com

c. Schedule for upcoming weeks:

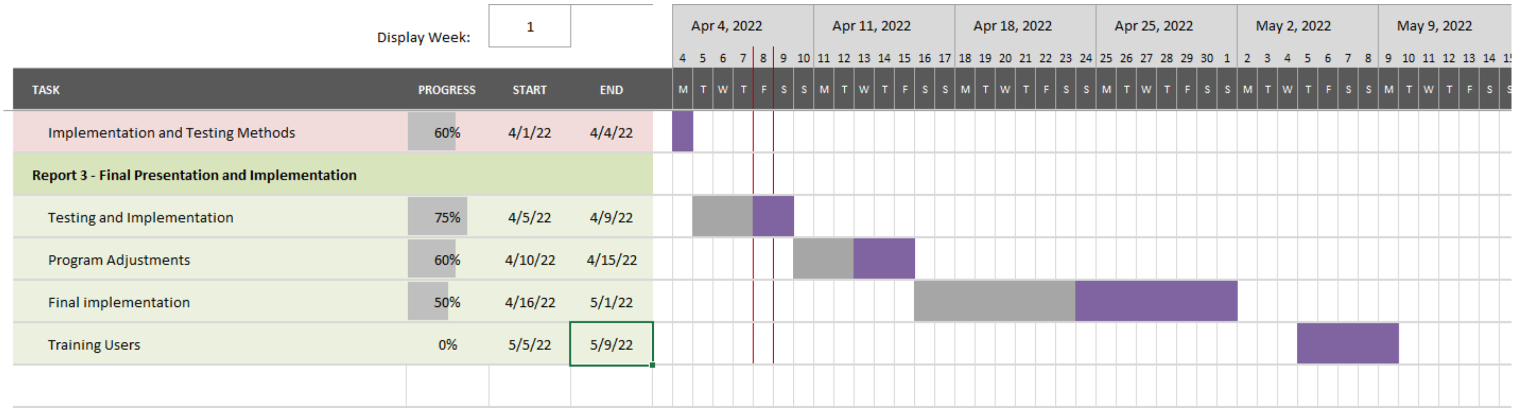
3rd and 4th week of March – Check the coding for project

1st week of April – Work on project UI's prototype

2nd week of April – Work on UI's development

3rd and 4th week – Wrap up project

d. Project Coordination and Progress Report (Gant chart):



Display Week: 1

TASK	PROGRESS	START	END
Implementation and Testing Methods	60%	4/1/22	4/4/22
Report 3 - Final Presentation and Implementation			
Testing and Implementation	75%	4/5/22	4/9/22
Program Adjustments	60%	4/10/22	4/15/22
Final implementation	50%	4/16/22	5/1/22
Training Users	0%	5/5/22	5/9/22

- Implemented use cases:

  We have already implemented in the program how to store a book, search a book, add/delete members and rent a book as described on the different diagrams shown above.

- What is functional and what is currently being tackled?

  The whole program is currently functional from Eclipse and NetBeans. However, after developing the SQL server we run into some issues connecting it to Java. Otherwise the whole system is functional and was successfully tested.

- Other related project management activities:

  I am trying to present the project to some peers and friends in order to get some feedbacks on how to improve it. But also, been doing a lot of research on different tools that can help me wrap up the project smoothly.

IX. References

Collaborate & Create Amazing Graphic Design for free - CANVA. (n.d.). Retrieved March 9, 2022, from https://www.canva.com/

Design a library management system - grokking the object-oriented design interview. Educative. (n.d.). Retrieved February 21, 2022, from https://www.educative.io/courses/grokking-the-object-oriented-design-interview/RMlM3NgjAyR

Nym. (2022, February 23). Library Management System use case diagram UML. Itsourcecode.com. Retrieved March 12, 2022, from https://itsourcecode.com/uml/library-management-system-use-case-diagram-uml/

Tilley, S., Rosenblatt, H. J., & Zhang, J. (2017). Information Systems Analysis, design and implementation (Eleventh edition) / Xi Tong Fen Xi Yu she Ji / Sikete Dili, Hali Luosenbulate Zhu ; Zhang Jin Bian Yi. Zhongguo ren min da xue chu ban she.