Online Library Management System

Requirement engineering

Section: 2, Group: 1

Husain Ali Ahmed (LEADER)   –   202106117

Abdullah Mohammed Salman   –   202104275

Haitham Abdullah Taher   –   202107520

Qasim Hasan Ali   –   202107902

Submitted to: DR. QASEM OBEIDAT

1-Purpose

- Library Management System (LMS) is being developed to automate most of the library services. LMS is also developed for documenting and organizing books and users' data. Automating library tasks such as issuing books and returning books lead to generating accurate statistics that help to take important decisions and improve the services provided by the library.

2-Scope

- In this part you can find information about what is in the scope of the system and what is out of scope of the system.

In-Scope:
1. Provide the functionality for adding, editing, and deleting book data.
2. Provide the functionality for adding, editing, and deleting member data, including personal details and membership status.
3. Provide functionality that enables tracking issuing activities.
4. Provide a search function that enables searching for books or members.
5. Provide a payment function to receive payments and link the business account "Bank account" to it.
6. Developing and implementing a QR code reader in the system.
7. Developing and implementing a social media strategy.
8. User authentication to make sure that the user is signed in.
9. Developing and implementing an overdue strategy.
10. Generating payment reports and Issued reports.

Out-Scope:
1. Support a multi-language.
2. Creating the QR codes "Done by third party could be the book publisher".
3. Publishing the system on the internet.
4. Creating an Email account for sending updates and notification for the system users.
5. Creating the Business account "Bank account".

6. Managing the library finances, such as budgeting, accounting, invoicing payment processing.
7. Human resources management, such as recruitment, payroll, performance management.
8. Facility management, such as renovations, physical maintenance repairs.
9. Marking and communications, promoting library services and events, managing social media accounts.
10. The process of inserting data "Done manually by the staff."
11. Creating of managing the library social media profiles.
12. User autoinduction "Done by third party."

3-Product Perspective

- It is a responsive website that fits its content within any screen size.
- A website that works on more than one browser (Chrome, Mozilla Firefox, safari, etc...)
- Simple and user-friendly interface.
- It is a website that allows managing and tracking the different library activities.

4-Product Requirements

Functional Requirements:

Requirement ID	SRS-LMS-001
Title	Login
Priority	High Priority ("Core Requirements").
Description	The user should be able to login to the system by entering username and password.
Version	V1.0

Requirement ID	SRS-LMS-002
Title	Add Member
Priority	High Priority ("Core Requirements").
Description	The user should be able to register into the system within one window.
Version	V1.0

Requirement ID	SRS-LMS-005
Title	Remove Book
Priority	High Priority ("Core Requirements").
Description	The librarians should be able to remove a book from the system within 1 window.
Version	V1.0

Requirement ID	SRS-LMS-003
Title	Remove Member
Priority	High Priority ("Core Requirements").
Description	The administrators should be able to remove a member from the system within 1 window.
Version	V1.0

Requirement ID	SRS-LMS-004
Title	Add Book
Priority	High Priority ("Core Requirements").
Description	The Librarians should be able to add a book to the system within 1 window.
Version	V1.0

Requirement ID	SRS-LMS-006
Title	Issue Book
Priority	High Priority ("Core Requirements").
Description	The system shall allow the Librarians to issue a book to a member.
Version	V1.0

Requirement ID	SRS-LMS-007
Title	Return Book
Priority	High Priority ("Core Requirements").
Description	The system shall allow the Librarians to manage the retention process of the book.
Version	V1.0

Requirement ID	SRS-LMS-008
Title	Search Book
Priority	High Priority ("Core Requirements").
Description	The system shall allow the user to search for a book to view the book's details: author, title, release date, International Standard Book Number and borrow status.
Version	V1.0

Requirement ID	SRS-LMS-009
Title	Purchasing
Priority	High Priority ("Core Requirements").
Description	The system should allow the user to purchase books from the library.
Version	V1.0

Requirement ID	SRS-LMS-010
Title	Search Member
Priority	High Priority ("Core Requirements").
Description	The system shall allow the librarians to search for a member to view: his/her full name, age, department, phone number, email and all the borrowed books.
Version	V1.0

Requirement ID	SRS-LMS-011
Title	Generate Reports
Priority	High Priority ("Core Requirements").
Description	The system shall allow the users to generate reports. Ex. The users shall be able to generate reports regarding borrowed books by them and payments dues. Ex. The admins shall be able to generate reports that contain a user's full name, university identification number, list of their borrowed books and payments dues.
Version	V1.0

Requirement ID	SRS-LMS-012
Title	Implementing Social media Strategy
Priority	Medium Priority ("Optional Requirements")
Description	The system should provide links and functions that let users to go to the library's social media profiles
Version	V1.0

Requirement ID	SRS-LMS-013
Title	QR Code Reader
Priority	Medium Priority ("Optional Requirements")
Description	The system should be able to read the QR code associated with each book.
Version	V1.0

Requirement ID	SRS-LMS-014
Title	Payment
Priority	High Priority ("Core Requirements").
Description	The system shall allow the user to pay his/her payments and dues.
Version	V1.0

Requirement ID	SRS-LMS-015
Title	Read Book
Priority	High Priority ("Fancy requirements")
Description	The system shall allow the user to read the book online.
Version	V1.0

Non-Functional Requirements:

Requirement ID	SRS-LMS-001
Title	Usability
Priority	High Priority ("Core Requirements").
Description	The user should be able to use the system without prior knowledge.
Version	V1.0

Requirement ID	SRS-LMS-002
Title	Performance
Priority	High Priority ("Core Requirements").
Description	The system should response to the user requests within 2 seconds. Ex. The system must response to Adding, Removing, Issuing, Retuning books within 2 seconds.
Version	V1.0

Requirement ID	SRS-LMS-003
Title	Scalability (Measurability)
Priority	High Priority ("Core Requirements").
Description	The System must be scalable enough to support and process 10 thousand visits at the same time while maintaining optimal performance.
Version	V1.0

Requirement ID	SRS-LMS-004
Title	Reliability
Priority	High Priority ("Core Requirements").
Description	The system should not have a break down more than 5 seconds a year.
Version	V1.0

Requirement ID	SRS-LMS-005
Title	Supportability
Priority	High Priority ("Core Requirements").
Description	The system should be supportable and maintainable throughout its life cycle where, adding a new function, or expanding an existing function has NO or low impact on the other functions.
Version	V1.0

Requirement ID	SRS-LMS-006
Title	Recoverability
Priority	High Priority ("Core Requirements").
Description	The system should be able to recover all the data after a system failure to ensure that the system must have a backup database.
Version	V1.0

Requirement ID	SRS-LMS-007
Title	Portability
Priority	High Priority ("Core Requirements").
Description	The system can run on Safari, Chrome, Microsoft Edge, and Firefox browsers with optimal performance.
Version	V1.0

Requirement ID	SRS-LMS-008
Title	Security
Priority	High Priority ("Core Requirements").
Description	The system must be secured where each user has a unique username and password.
Version	V1.0

Requirement ID	SRS-LMS-009
Title	Traceability
Priority	High Priority ("Core Requirements").
Description	The system must be traceable to its requirements and design.
Version	V1.0

Requirement ID	SRS-LMS-010
Title	Accessibility
Priority	High Priority ("Core Requirements").
Description	The system should allow access to its user.
Version	V1.0

5-Principal Stakeholders

The principal stakeholders of an online library management system are the individuals or groups who have an interest or stake in the system and its outcomes.

- **Library Patrons:** Library patrons are the users of the library and use the system to search for library materials, reserve books, and track their borrowing history. They rely on the system to access the library's collections and services.

- **Librarians:** they are the primary users of the system, they are responsible for managing the library's resources, memberships and borrowers. Moreover, they use the system to manage the library collections, track lending and provide services to library patrons.

- **Library Administrators**: Library administrators are responsible for the overall management of the library and use the system to monitor library operations, generate reports, and make informed decisions about resource allocation and service delivery.

- **Book owners:** they are the owners of the books in the library, and they use the system to check on their books.

- **IT Staff:** they are responsible for the maintenance, support, and development of the system. They ensure that the system is running smoothly, troubleshoot issues, and make improvements and upgrades to the system as needed.

- **Vendors:** they are the companies or individuals who provide the software, hardware, or other services related to the system.

6-User Characteristics

Users should have access to users' services, and employees should have access to both user's and management services.

The user (Library Patrons):
- The user should be familiar with the services provided by the system.
- The user must know how to interact with the system.
- The user should know the constraints required for each service.
- The user should be familiar with the necessary information to arrange his/her time for the borrowing and retention processes.
- The user must know all the regulations required to use the system services.

The Employee (Librarians):
- The Employee should be familiar and capable of managing all the system services.
- The employee should be accurate when using the system's services.
- The employees should look after customers' satisfaction.
- The employee should know how to analyze extract the system information.
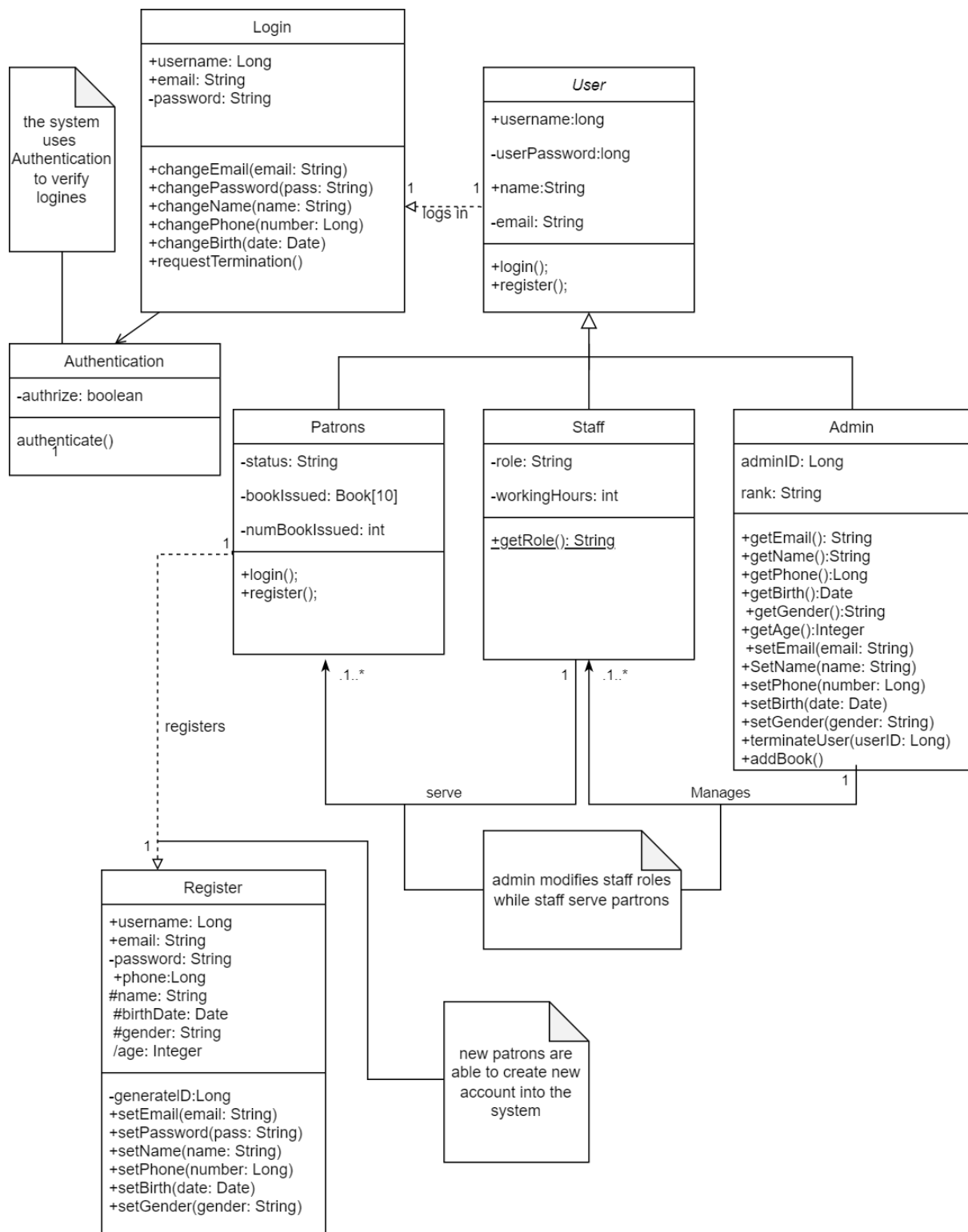
The Administrative (Library Administrators):
- The administrator must be familiar with adding to or deleting books from the system.
- The administrator must be familiar with adding and deleting Employee accounts.
- The administrator should know how to control and modify book prices.
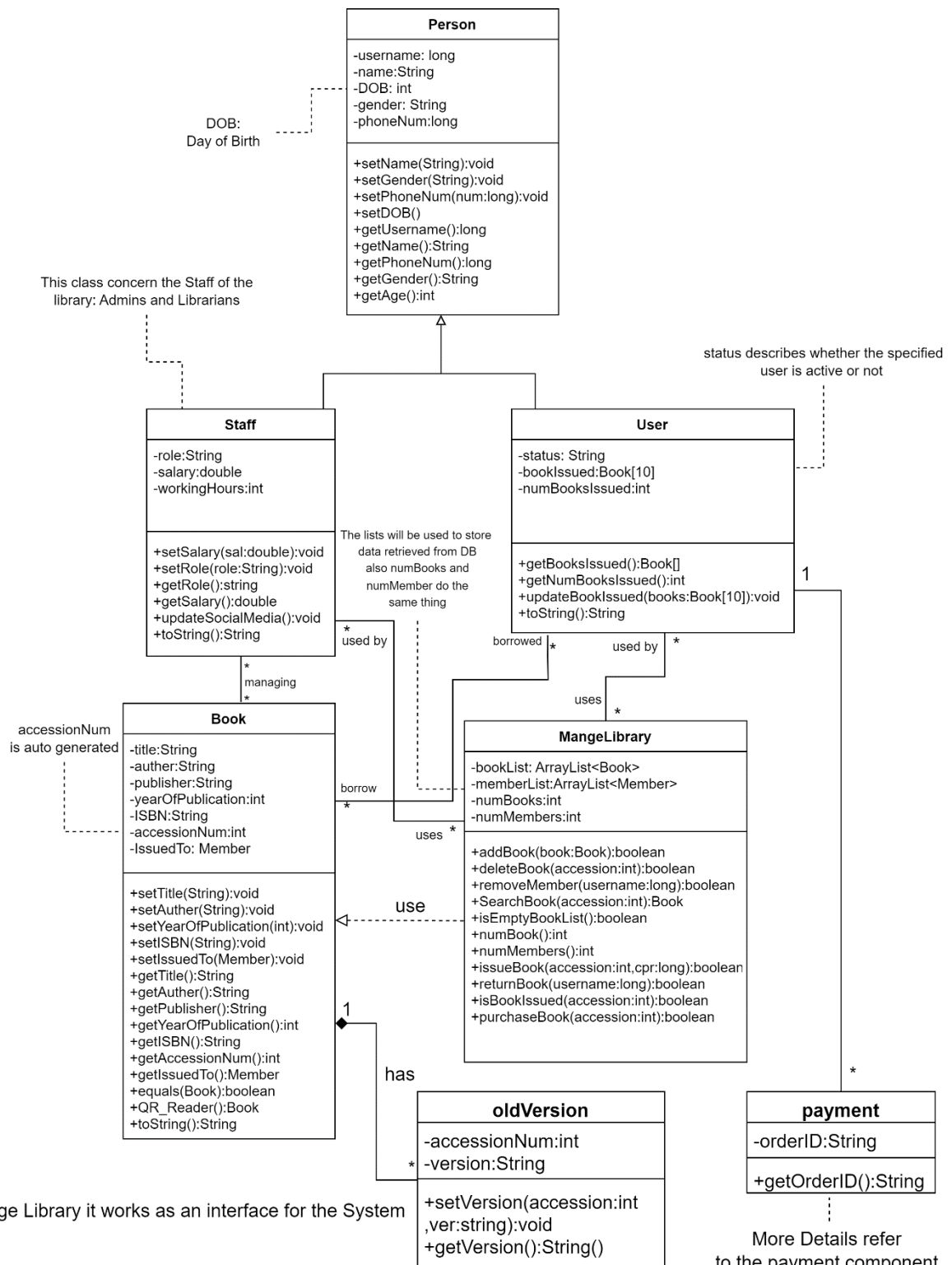- The administrator must know how to generate reports.
-

7-General Constraints

- The system must be used in a web browser.
- The system should work on most laptops and computers with 2 or more.
- The system is intended to run on a browser engine with OS 10 or later.
- The system is limited by operating in servers of a maximum number of users at a given time.
- The system must be user-friendly.
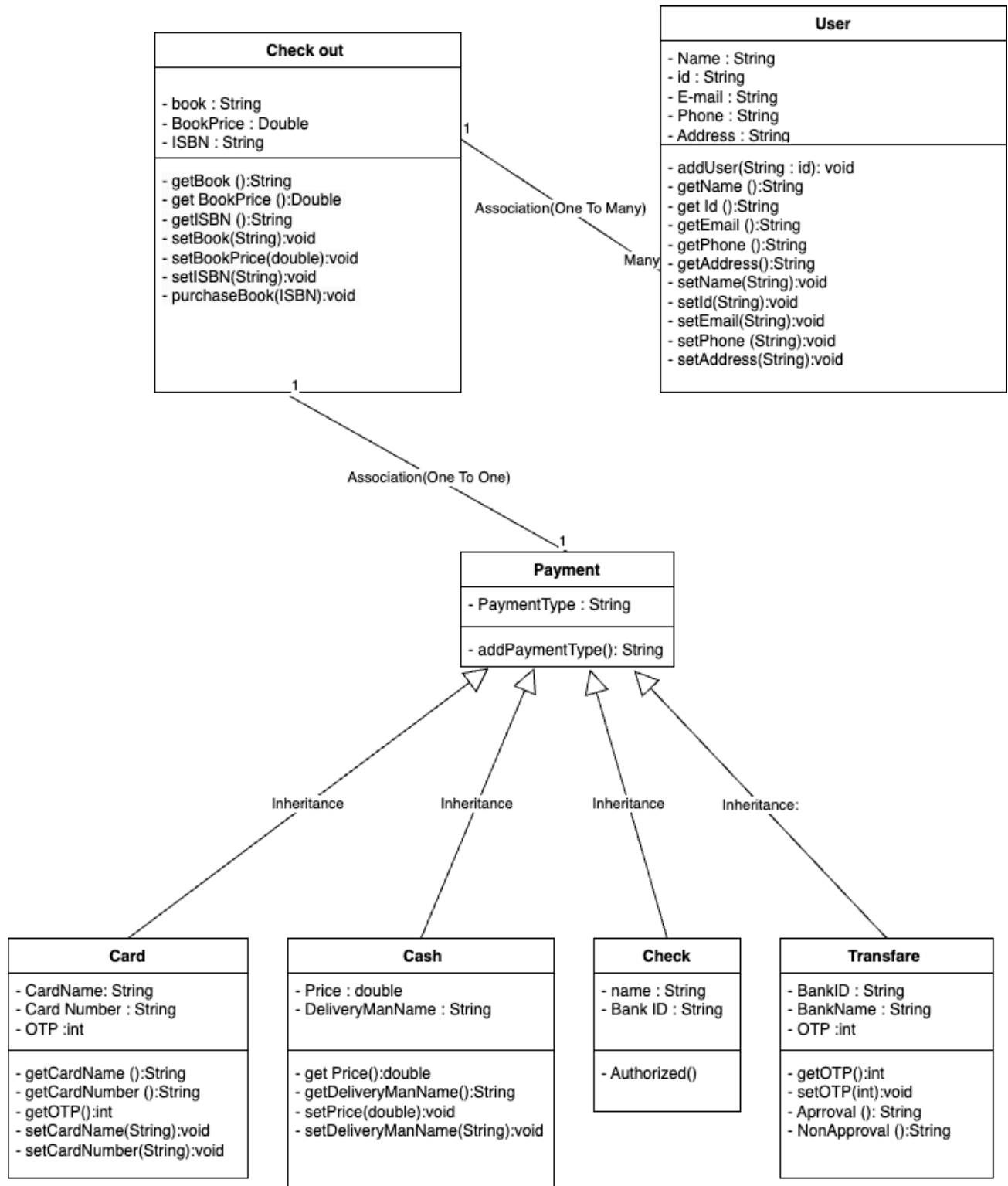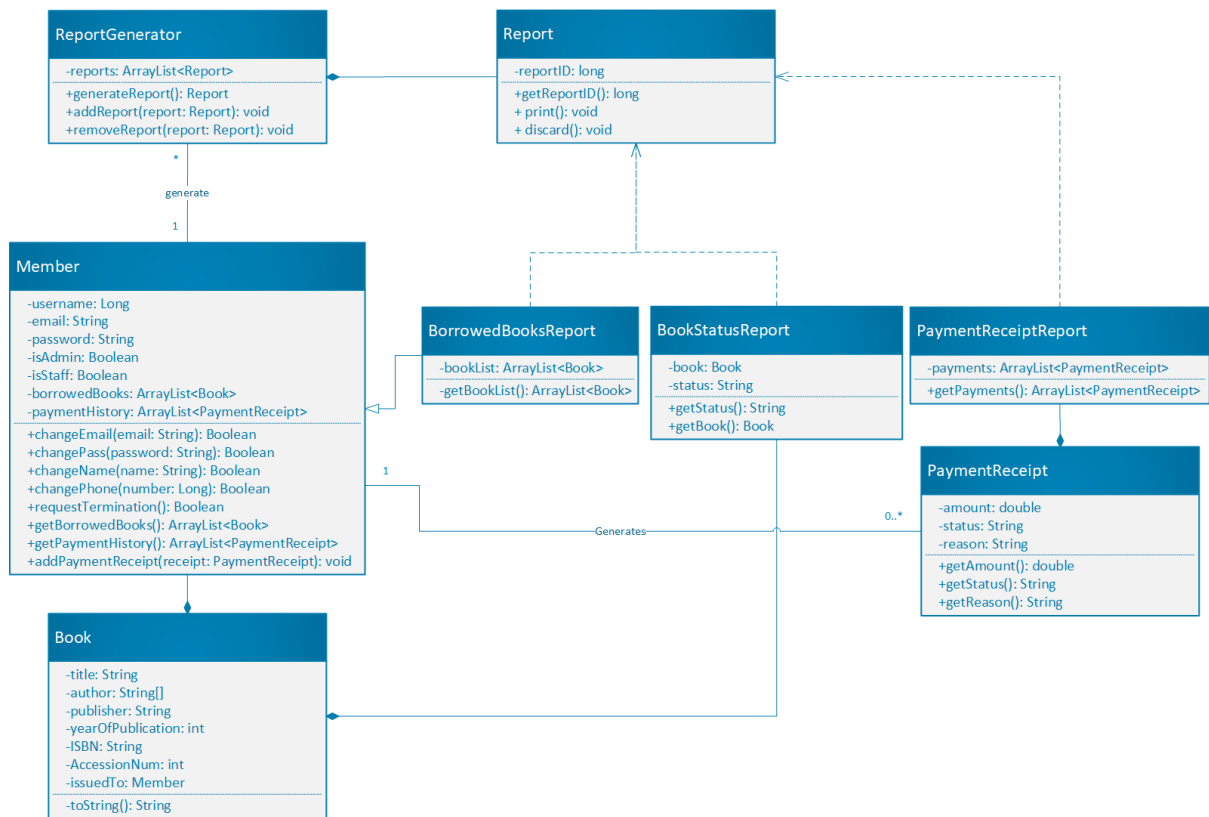- The system requires internet.

Registration:

Login

+username: Long
+email: String
-password: String

+changeEmail(email: String)
+changePassword(pass: String)
+changeName(name: String)
+changePhone(number: Long)
+changeBirth(date: Date)
+requestTermination()

User

+username:long

-userPassword:long

+name:String

-email: String

+login();
+register();

1 ◁ logs in 1

the system uses Authentication to verify logines

Authentication

-authrize: boolean

---

authenticate()

1

Patrons

-status: String

-bookIssued: Book[10]

-numBookIssued: int

+login();
+register();

Staff

-role: String

-workingHours: int

+getRole(): String

Admin

adminID: Long

rank: String

+getEmail(): String
+getName():String
+getPhone():Long
+getBirth():Date
+getGender():String
+getAge():Integer
+setEmail(email: String)
+SetName(name: String)
+setPhone(number: Long)
+setBirth(date: Date)
+setGender(gender: String)
+terminateUser(userID: Long)
+addBook()

1

.1..*   1   .1..*

serve   Manages

1

registers

admin modifies staff roles while staff serve partrons

Register

+username: Long
+email: String
-password: String
+phone:Long
#name: String
#birthDate: Date
#gender: String
/age: Integer

-generateID:Long
+setEmail(email: String)
+setPassword(pass: String)
+setName(name: String)
+setPhone(number: Long)
+setBirth(date: Date)
+setGender(gender: String)

new patrons are able to create new account into the system

Manage Library:

Person

-username: long
-name:String
-DOB: int
-gender: String
-phoneNum:long

+setName(String):void
+setGender(String):void
+setPhoneNum(num:long):void
+setDOB()
+getUsername():long
+getName():String
+getPhoneNum():long
+getGender():String
+getAge():int

DOB:
Day of Birth

This class concern the Staff of the
library: Admins and Librarians

status describes whether the specified
user is active or not

Staff

-role:String
-salary:double
-workingHours:int

+setSalary(sal:double):void
+setRole(role:String):void
+getRole():string
+getSalary():double
+updateSocialMedia():void
+toString():String

User

-status: String
-bookIssued:Book[10]
-numBooksIssued:int

+getBooksIssued():Book[]
+getNumBooksIssued():int
+updateBookIssued(books:Book[10]):void
+toString():String

The lists will be used to store
data retrieved from DB
also numBooks and
numMember do the
same thing

* used by

borrowed *

used by *

* managing
*

uses *

Book

-title:String
-auther:String
-publisher:String
-yearOfPublication:int
-ISBN:String
-accessionNum:int
-IssuedTo: Member

+setTitle(String):void
+setAuther(String):void
+setYearOfPublication(int):void
+setISBN(String):void
+setIssuedTo(Member):void
+getTitle():String
+getAuther():String
+getPublisher():String
+getYearOfPublication():int
+getISBN():String
+getAccessionNum():int
+getIssuedTo():Member
+equals(Book):boolean
+QR_Reader():Book
+toString():String

accessionNum
is auto generated

MangeLibrary

-bookList: ArrayList<Book>
-memberList:ArrayList<Member>
-numBooks:int
-numMembers:int

+addBook(book:Book):boolean
+deleteBook(accession:int):boolean
+removeMember(username:long):boolean
+SearchBook(accession:int):Book
+isEmptyBookList():boolean
+numBook():int
+numMembers():int
+issueBook(accession:int,cpr:long):boolean
+returnBook(username:long):boolean
+isBookIssued(accession:int):boolean
+purchaseBook(accession:int):boolean

borrow
*

uses *

use

has

1

1

*

oldVersion

-accessionNum:int
-version:String

+setVersion(accession:int
,ver:string):void
+getVersion():String()

payment

-orderID:String

+getOrderID():String

More Details refer
to the payment component

Note: Manage Library it works as an interface for the System

Payment:

Check out

- book : String
- BookPrice : Double
- ISBN : String

- getBook ():String
- get BookPrice ():Double
- getISBN ():String
- setBook(String):void
- setBookPrice(double):void
- setISBN(String):void
- purchaseBook(ISBN):void

User

- Name : String
- id : String
- E-mail : String
- Phone : String
- Address : String

- addUser(String : id): void
- getName ():String
- get Id ():String
- getEmail ():String
- getPhone ():String
- getAddress():String
- setName(String):void
- setId(String):void
- setEmail(String):void
- setPhone (String):void
- setAddress(String):void

1   Association(One To Many)   Many

1

Association(One To One)

1

Payment

- PaymentType : String

- addPaymentType(): String

Inheritance   Inheritance   Inheritance   Inheritance:

Card

- CardName: String
- Card Number : String
- OTP :int

- getCardName ():String
- getCardNumber ():String
- getOTP():int
- setCardName(String):void
- setCardNumber(String):void

Cash

- Price : double
- DeliveryManName : String

- get Price():double
- getDeliveryManName():String
- setPrice(double):void
- setDeliveryManName(String):void

Check

- name : String
- Bank ID : String

- Authorized()

Transfare

- BankID : String
- BankName : String
- OTP :int

- getOTP():int
- setOTP(int):void
- Aprroval (): String
- NonApproval ():String

Generate Reports:

ReportGenerator

-reports: ArrayList<Report>

+generateReport(): Report
+addReport(report: Report): void
+removeReport(report: Report): void

Report

-reportID: long

+getReportID(): long
+ print(): void
+ discard(): void

generate

*

1

Member

-username: Long
-email: String
-password: String
-isAdmin: Boolean
-isStaff: Boolean
-borrowedBooks: ArrayList<Book>
-paymentHistory: ArrayList<PaymentReceipt>

+changeEmail(email: String): Boolean
+changePass(password: String): Boolean
+changeName(name: String): Boolean
+changePhone(number: Long): Boolean
+requestTermination(): Boolean
+getBorrowedBooks(): ArrayList<Book>
+getPaymentHistory(): ArrayList<PaymentReceipt>
+addPaymentReceipt(receipt: PaymentReceipt): void

BorrowedBooksReport

-bookList: ArrayList<Book>

-getBookList(): ArrayList<Book>

BookStatusReport

-book: Book
-status: String

+getStatus(): String
+getBook(): Book

PaymentReceiptReport

-payments: ArrayList<PaymentReceipt>

+getPayments(): ArrayList<PaymentReceipt>

PaymentReceipt

-amount: double
-status: String
-reason: String

+getAmount(): double
+getStatus(): String
+getReason(): String

1

Generates

0..*

Book

-title: String
-author: String[]
-publisher: String
-yearOfPublication: int
-ISBN: String
-AccessionNum: int
-issuedTo: Member

-toString(): String

Project Component:

For Library Management System

Components are:

1- Registration: It is the way the users can register through the system and modify their account's details. It's also where they login into the system every time they use it.

2- Manage Library (it is a core component that manages the monitoring and maintenance of the books, whether they are available for borrowing or unavailable.)

3- Payment (it is the process of sending money to the right account, paying for the fees.)

4- Generate Reports (It is used by administrated to generate reports to monitor the system)

System Use Cases:

Component Name	Use Case Name	Description
Registration	Use Case 1 Register	The user creates new account in the system
	Use Case 2 Login	The user login into the system using an existing account
	Use Case 3 Change Password	The user changes his/her account password
	Use Case 4 Terminate Account	The user deletes his/her account. The account must have 0 issued books before deleting
Manage Library Activities	Use Case 1 Add Book	Add a book to the system
	Use Case 2 Remove Book	Remove a book form the system
	Use Case 3 Search Book	Search for books
	Use Case 4 QR code Reader	Read books details using QR code label

	Use Case 5 Issue Book	Use Book for a user
	Use Case 6 Return Book	Manages the book return process
	Use Case 7 Purchasing Book	Enables the users to purchase books
	Use Case 8 Social Media Strategy	Implementing the social media" Crate connections between the library and its social accounts"
Payment	Use case 1 Check out	The user adds books to checkout
	Use case 2 Chose payment method	The user picks a payment method
	Use case 3 Pay by cash	The user chooses to pay using cash
	Use case 4 Pay by card	The user chooses to pay using card
	Use case 5 Generate payment receipt	The user confirms his payment and receives his/her payment receipt
Generate Reports	Use Case 1 View User's Fees	View the fees that a user must pay
	Use Case 2 View Borrowed Books	Users view their borrowed books
	Use Case 3 View Book Status	View the book status if it's borrowed or not
	Use Case 4 View Member Status	View a member's status to see if he/she is blacklisted, has dues and more details
	Use Case 5 View Fees	Users can view their own fees

Registration component	Husain Ali Ahmed	–	202106117
Manage library component	Haitham Abdullah Taher	–	202107520
Payment component	Qasim Hasan Ali	–	202107902
Generate reports component	Abdullah Mohammed Salman	–	202104275

Registration Component:

Use case 1: Register

Primary Actor: User

Pre-condition: User must be adult and not registered with the same email previously.

Postcondition: the user creates a new account and is able to use the system.

Main Scenario:
1- The system asks the user to enter an email, username and password.
2- System will be sending a code to the email address to verify.
3- User will open the link to see a message "Verification successfully".
4- User will login to the system.

Alternate Scenario:
Someone is either under 18 or entered a used email, so the system showed a message to the user to enter valid email.

Quality requirement:
Security:
The system should protect user's personal information and keep it hidden.
The system should not allow 2 active users at the same time.
Responsiveness:
The system should respond within 0.3 seconds once the user clicks login.

Priority level:
High level

Related functional requirement IDs:
 SRS-HMS-001

Login

Primary Actor: User, Admin, Staff

Pre-condition: User must be registered to the system.

Postcondition: The user can login with his account and use the system.

Main Scenario:
1. The system asks the user to enter the username and password.
2. User chose verification by email if new IP detected.
3. User enters his email.
4. System will be sending an email to verify.
5. User will open the link to see a message "Verification successfully".
6. User will login to the system.

Alternate Scenario:
 Someone entered the wrong password or incorrect username.

Quality requirement:
Security: The system should protect user's personal information and keep it hidden.
The system should not allow 2 active users at the same time.
Responsiveness: The system should respond within 0.3 seconds once the user clicks login.

Priority level: High level

Related functional requirement IDs: SRS-HMS-002

Change password.

Primary Actor: User/staff

Pre-condition: User must be logged in

Postcondition: The user can change his/her account password.

Main Scenario:
1- Users click change password under profile menu.
2- System asks the user to his/her old password and new password.
3- System asks the user to reenter the password to confirm it.
4- The system will send a verification code to the user email address.
5- User will enter the verification code and click confirm.
6- System will show a message "password changed successfully" and send verification message through the email.
7- User will be logout out of the system from all devices.

Alternate Scenario:The user failed to confirm his/her new password, so the system repeated all the steps.

Quality requirement:
Security: The system should not allow the user to set an easy password.
The system should log out of all devices after changing the password.
Reliability: The system should change the password once the user clicks confirm in all databases and not fail to change it.

Priority level: High level

Related functional requirement IDs: SRS-HMS-003

Terminate account.

Primary Actor: User/staff

Pre-condition: User must be logged in and have no borrowed books and no unpaid fees.

Postcondition: the user account will be deleted from the active accounts in the database.

Main Scenario:
1- User click terminate account under profile menu.
2- System shows "warning: all data will be lost, enter your password to confirm"
3- System asks the user to enter a reason to terminate the account.
4- System will show a message "your account has been terminated".
5- The user information will be archived.

Alternate Scenario:
User entered wrong password, so the system asked the user to reenter it.
User left the termination reason blank, so the system didn't proceed.

Quality requirement:
Security: The system should not allow the user to terminate his/her account without confirming the password and warning the user.
The system should log out of all devices after changing the password.
Documentation: The system should store all the data related to the user in the old accounts database.

Priority level: High level

Related functional requirement IDs: SRS-HMS-004

Manage Library:
Use Case 1: Add Book

Primary Actor: Admin "Librarian"

Pre-Conditions: The admin should log into the system, Preparing valid book detail for the insertion process.

Post-Conditions:
- A message will bob up stating that "The book has been added successfully."
-

Main Scenario:
1- Log into the system.
2- Open the Manage Books by clicking on "library services"
3- Insert the Book detail:
   - Title
   - Author
   - Publishers
   - Year of publication
   - ISBN
4- Click on the "Add" button
Not: Accession Number "auto generated"

Alternate Scenario: 4.a: When trying to add a book that already exists in the library system "the duplicated book will be considered as a copy of the existing one", The number of copies of that book will be increased by one and an accession number will be generated for the new copy.

Quality Requirement: Usability & performance
This function is designed in a manner that a user can perform without any helping frames. Each insertion process should take at most 2 seconds processing time after pressing the Add button.

Priority Level: High level

Related Functional Requirement IDs: SRS-LMS-004

Use Case 2: Remove Book

Primary Actor: Admin "Librarian"

Pre-Conditions: The admin should log into the system and the book to be removed must be stored in the system.

Post-Conditions: A bob up message will appear stating that the book has been added successfully.

Main Scenario:
1. Log into the system
2. Open the library services menu by clicking on "Manage Library"
3. Select the book to be removed
4. Click on "Remove" button
5. The Book will be removed and a bob up message will appear stating that the book has been removed successfully.

Alternate Scenario:
      Same as Step 1 –4 in the main scenario
6. If the book is issued to a member the deletion process will fail and the system will return the user to the previous step.

Quality Requirement:
Usability & performance
This function is designed in a manner that a user can perform without any helping frames. Each insertion process should take at most 2 seconds processing time after pressing the Add button.

Priority Level: High level

Related Functional Requirement IDs: SRS-LMS-005

Search Book

Primary Actor: User

Pre-Conditions: The user should have internet access.

Post-Conditions: The results of the search will appear to the user.

Main Scenario:
1. Click on the search bar that appears on the main page.
2. Type the accession number of the book in the search bar.
3. Click on the search icon.
4. The Book details will be shown to the user.
5.

Alternate Scenario: 6.a: If the Book the user is looking for is not in the system. When the user presses on the search icon a message should bob up stating that the book is not in the system.

Quality Requirement:
Performance: The result of the searching function must be shown within 2 seconds.

Priority Level: High level

Related Functional Requirement IDs: SRS-LMS-008

<u>Use Case 4:</u> QR Code Reader

Primary Actor: User

Pre-Conditions:
- Each book must have a unique QR code label
- Each QR code must be stored in the system.
-

Post-Conditions: The reading results will be shown.

Main Scenario:
1. Log into the system.
2. Open the library services menu by clicking on "Manage Library"
3. Click on the QR code icon
4. Scan the QR code using a QR code reader
5. The info of the book will appear

Alternate Scenario:
- 5.a: If the book code is not stored in the system, the system will generate a message for the user stating that the book code could not been found and take the user to the previous page.
-

Quality Requirement:
- Performance the response time must not exceed 2 seconds.
- Privacy: the data cannot be accessed by others than the admin and user that hold the account itself

Priority Level: Medium level

Related Functional Requirement IDs: SRS-LMS-013

Issue Book

Primary Actor: User/Admin

Pre-Conditions:

- The user must be registered
- The book to issued must be available in the library
- The user to which the book will be issued must have no late fees.

Post-Conditions:

- The number of books issued to the member who borrowed the book will be updated
- A bob up message will appear at the end of the issuing process to the user stating "your request has been added successfully. Please go to the librarian to get the book."

Main Scenario:

1. Log into the system
2. Open the library services menu by clicking on "Manage Library"
3. Click on the "Issue Book" button
4. Select the book "can be found using the search function"
5. Click on the "send" button
6. A request to borrow the book will be stored in the system
7. The request will be confirmed by the librarian.
8.

Alternate Scenario:

- 7.a1: If the user already has ten books issued to him then a bob up message will be shown stating that "Sorry, you cannot borrow more than ten books." and the request will be denied.
- 7.a2: If the user has late fees the issuing process would be denied, and a message will appear to the user stating "Please, pay your previous fees" and the system will take the user to the payment interface.

Quality Requirement:

- Performance and: The sever_processing is fast where it takes at most 2 seconds.
- Usability: It is easy for a new user to issue a book where all the requirements needed to complete this task are present in one interface "page".

Priority Level: High level

Related Functional Requirement IDs: SRS-LMS-006

Return Book

User/ Admin

- The user logged into the system.
- Book must be issued to the user.

- A conformation message will appear to the user stating, "Thank you for using our services, book has been returned successfully."
- The book returned can be issued to other members.

1. Log into the system.
2. Open the library services menu by clicking on "Manage Library"
3. Click on the "Return Book" button.
4. A list of the books issued to the specified member will be shown.
5. Select the book(s) to be returned from the issued books list
6. Click on "submit" button
7. A request for returning a book will be stored in the system
8. The user should go to the library with the book
9. The librarian will take the book and confirm the book's retention.

- 9.a: If the book is returned after the deadline of the borrowing period, the system will take the user to the payment interface "page" to pay the charge on him.
- 9.a: If the book is damaged, the librarian will not confirm the retention process and will issue a bill asking the user to pay the book's cost "the payment is done through the system".

- Performance and: The server processing is fast where it takes at most 2 seconds.
- Usability: It is easy for a new user to issue a book where all the requirements needed to complete this task are present in one interface "page".
- Privacy: the data cannot be accessed by others than the admin and user that hold the account itself

High level

SRS-LMS-007

<u>Use Case 7:</u> Purchasing Book

Primary Actor: User

Pre-Conditions:
- The user must be registered in the system "when buying through the system"
- Acceptance on terms and conditions "will be shown be for each payment".

Post-Conditions:
- The system will generate a message to the user stating that "The purchase process was done successfully"

Main Scenario:
1. Log into the system.
2. Open the library services menu by clicking on "Manage Library"
3. Click on the "Purchase Book" button
4. Select the book to purchased "Enter the accession number"
5. Click on Purchase Book
6. The payment page will open
7. Choose the payment method
8. If the payment succeeded the system will generate a message stating "Thank you for choosing us, the purchase process succeeded"

Alternate Scenario:
7.a: If the payment failed that system would generate "The purchasing process failed, please try again."

Quality Requirement:
- Security: the members' personal information must be secured during the payment.
- Responsiveness: The system must responses to the user action within at most 2 seconds.
-
Priority Level: High level
Related Functional Requirement IDs: SRS-LMS-011

<u>Use Case 8:</u> Implementing social media strategy

Primary Actor: Admin

Pre-Conditions:
- Logged into the system
- The access must be done through an admin account

Post-Conditions:
- Links that direct the users to the library social media platforms.

Main Scenario:
1- Log into the system
2- Open the library services menu by clicking on "library services"
3- Press on one of the social media icons that appear at the center-bottom
4- A new page "interface" will open contain the specification of the links associated with each icon.
5- Admin changes the specification of each icon "link, color of the icon, change icon, QR code of the profile"
6- After setting the specification the icon selected, the admin should press on apply changes.
7- A pub up message will appear stating "The icon specification has been changed successfully."

Alternate Scenario:
- 7.a: If there is an error in the link that points to the social media platform the system should open a new page that shows the QR code that directs the user to the social profile of the library.
-

Quality Requirement:
- Correctness: the link associated with each icon must direct to the correct social profile
- Performance: the system performs the function within 2 seconds after the user clicks on the icon.

Priority Level: Medium level
Related Functional Requirement IDs: SRS-LMS-014

Payment:

Use case 1: Check out.

Primary Actor: Student, Admin, user.

Pre-condition: User must have an account in system and must be logged in.

Postcondition: The user can select the books he wants to buy and put them in the basket

Main Scenario:
1- Log in into the system.
2- Search for books or authors.
3- Determine the books to be purchased.
4- Add the address
5- Add contact information.

Alternate Scenario:
Someone has no address or no Contact details (phone number, email)

Quality requirement:
Usability:
The system should be easy to use without prior knowledge.
Reliability:
The system should not have a break down more than 5 seconds a year.
Recoverability:
The system should be able to recover all the data after a system failure to ensure that the system must have a backup database.

Priority level: High level

Related functional requirement IDs: SRS-LMS-011

Choose payment method.

Primary Actor: User

Pre-condition: Users must log in and have books in basket.

Postcondition: The user can select the payment method.

Main Scenario:
1-user finish the check out and fill in the required information.
2-system asks the user to choose the payment method.
3- user chose his payment method.

Alternate Scenario:
User does not have any item to purchase or don't have any payment method.

Quality requirement:
Accessibility:
The system should be able to accessibility with user by contact information.
Security:
The system should protect the user information and address.

Priority level:
High level

Related functional requirement IDs:
SRS-LMS-002

Pay by cash on delivery.

Primary Actor: User /Delivery man

Pre-condition: Users must purchase the books and fill all information.

Postcondition: The user can complete the payment process by paying by cash on delivery.

Main Scenario:
The user completes his order online and Fill all the information required.
The user wants to pay by cash.
The delivery man takes the order to the user.
The user opens the wallet and gives the money to delivery man.
The delivery man gives the user the receipt with the change.

Alternate Scenario:

The customer does not have enough money.
The employee will cancel some services from the system.

Quality requirement:
Payment:
 The system shall allow the user to pay by cash.
Performance:
The system should response to the user requests within a few seconds.

Priority level:
High level

Related functional requirement IDs:
SRS-LMS-012

Pay by card

Primary Actor: User, Bank, employee

Pre-condition: Users must be purchased the books and fill all information.

Postcondition: The user can complete the payment process

Main Scenario:
1-user should fill his bank account.
2-the bank will send OTB password to the user number
3- the user should validate his bank account by OTB password.
4- The bank should debit the required amount from the account balance.
5- the user will receive an SMS message from the bank.

Alternate Scenario:The user bank account had no enough balance.

Quality requirement:
Payment: The system shall allow the user to pay.
Performance: The system should response to the user requests within 2 seconds.
Security: The system should protect the user bank information.

Priority level: High level

Related functional requirement IDs: SRS-LMS-012

Generate payment receipt

Primary Actor: User

Pre-condition: User must have finished his payment.

Postcondition: User will receive the payment receipt

Main Scenario:
1- the system will generate the receipt for payment process
2- the system will send an email to user about confirmed payment.
3- The system sends a report to the user if something goes wrong or the process fails for some reason.

Alternate Scenario: If the user contact information is wrong.

Quality requirement:
Payment: The system shall allow the user to pay.

Accessibility: The system should allow access to its user.

Priority level: High level

Related functional requirement IDs: SRS-LMS-010

Generate Reports Component:

Use Case 1: View User's Fees

Primary Actor: Librarians (Admins)

Pre-Conditions: Admin must be existing and logged in to the system

Post-Conditions: Generate report about a user's fees

Main Scenario:
1. Admin must complete the login process.
2. The admin must click on the main menu from the admin's home page.
3. The system will redirect the admin to another page within the same window.
4. The admin must click on View a User Fees.
5. The system will ask for the ID of the user that the admin wants to view his/her fees.
6. The admin must type down the user's ID.
7. A new page will pop up that includes all the user's information with his/her fees.
8. The admin can either print the generated report or just discard it.
9.

Alternate Scenario:

6a: User does not exist
1. Allow the admin to re-enter the ID of the user.

Quality Requirement:

The system should protect the user's personal information and provide an accurately calculated fee (if there is any).

Priority Level: High level.

Related Functional Requirement IDs: SRS-LMS-011

View Borrowed Books

Primary Actor: User/ Admin

Pre-Conditions:
- The user, either an ordinary user or the admin must be logged in to the system to view a user borrowed books details.
-

Post-Conditions:
- A full detailed report that contains all the data of books issued to a specific user.

Main Scenario:
1- Logged into the system successfully
2- The user must click on the main menu on the home page.
3- The system will redirect the user to another page within the same window.
4- The user must click on View Borrowed books.
5- In this event if the user is an admin, then he/she must enter the ID of the member desired to view his borrowed books details. But if the user is an ordinary user, then after he clicked on the view borrowed books a full detailed report containing all the books issued to him/her will be shown.
6- The user can either print the generated report or just discard it.

Alternate Scenario:
- If the function is used by the admin and when he/she entered the member ID, if the ID entered does not exist the system shall allow the admin to re-enter the user ID.
-

Quality Requirement:
- Response time: the report must be generated within at most 2 seconds after the user presses the view borrowed Books button.
- Privacy: the data presented on the report must be hidden from others accept the user himself or the admin
-

Priority Level: High level

Related Functional Requirement IDs: SRS-LMS-011

View Book Status

Primary Actor: Librarians and Admins

Pre-Conditions: Admin must be existing and logged in to the system

Post-Conditions: Generate report about a book's status

Main Scenario:
1. Admin or Librarian must complete the login process.
2. The user must click on the main menu from the staff home page.
3. The system will redirect the user to another page within the same window.
4. The user must click on View a Book's Status.
5. The system will ask for the ID of the book that the user wants to view its status.
6. The user must type down the book ID.
7. A new page will pop up that includes all the book's information with its availability status.
8. The user can either print the generated report or just discard it.

Alternate Scenario:
6a: The book does not exist in the system.
1. Allows the user to re-enter the ID of the book.

Quality Requirement: The system shall respond to the admin within 3 seconds and provide the information of the book accurately.

Priority Level: High level
Related Functional Requirement IDs: SRS-LMS-011

View Member Status

Primary Actor: Librarian (Admins)

Pre-Conditions: The admin must be existing and logged in to the system.

Post-Conditions: Generate a report which includes all a library member's information with his status if he/she is blacklisted or not.

Main Scenario:
1. The admin must complete the login process.
2. The admin must click on the main menu.
3. The admin must click on View Member Status.
4. The system will ask for the ID of the member that the admin wants to view his/her status.
5. The admin must type down the ID of the member.
6. A new page will popup that includes all the member information and his/her status.
7. The admin can either click on "PRINT" to print the generated report, take a punishment if the user is blacklisted or close the page

Alternate Scenario:
5a: Member Does Not Exist
1. Allows the admin to re-enter the ID of the member.

Quality Requirement: The user's personal information must be protected and secured, and the response time must be within less than 4 seconds.

Priority Level: High level

Related Functional Requirement IDs: SRS-LMS-011 and SRS-LMS-014

View Fees

Primary Actor: Librarians and Users

Pre-Conditions: The user must be existing and logged in to the system

Post-Conditions: Pay fees

Main Scenario:
1. The user must complete the login process.
2. The user must click on the main menu.
3. The user must click on View Fees.
4. A new window will popup including all the information of the user and his/her fees.
5. The user can either click on "PAY" to pay his/her fees (if there is any) or leave the window.

Alternate Scenario:
3a: Connection Lost
1. An alert will be displayed showing that the connection was lost.
2. Asks the user to reload the page
3.

Quality Requirement: protects the user's personal information and the system must respond within 3 seconds.

Priority Level: High level

Related Functional Requirement IDs: SRS-LMS-015

Use cases:

Registration: by Hussain Ali Nooh

Online Library Registeration System
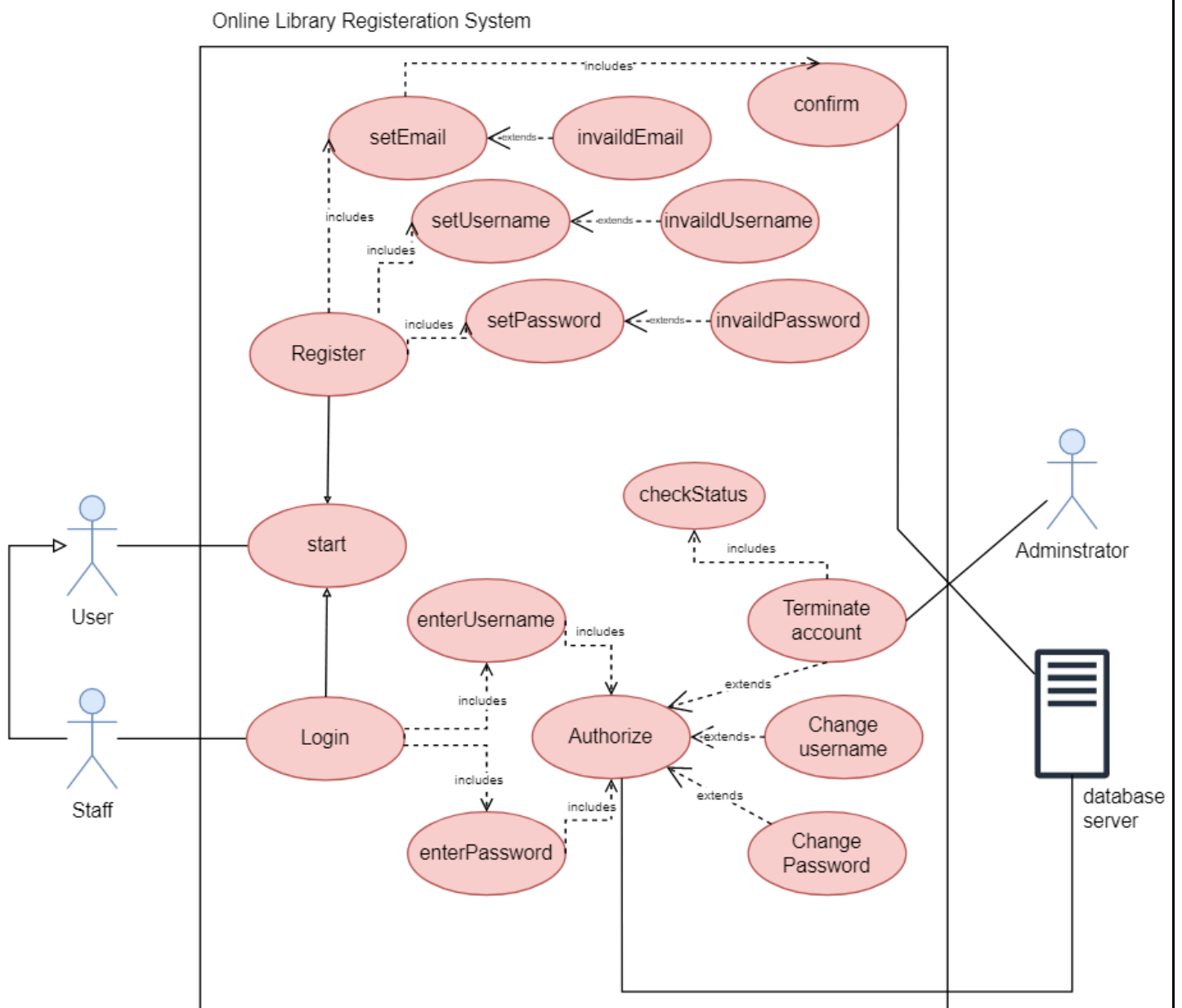
Manage Library: by Haitham Abdullah

Assume that the user has logged into the system successfully
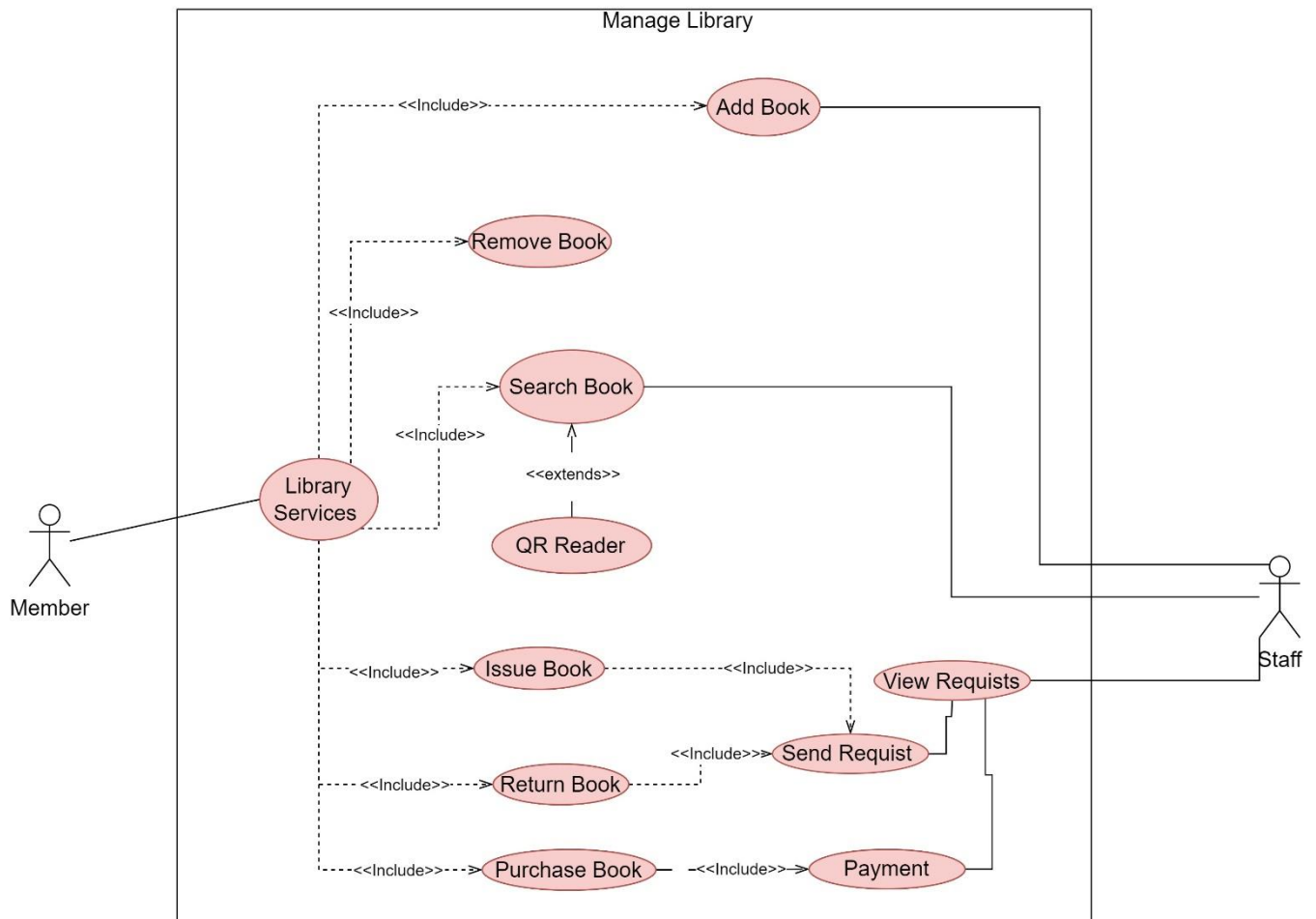
Payment: by Qasim Hasan
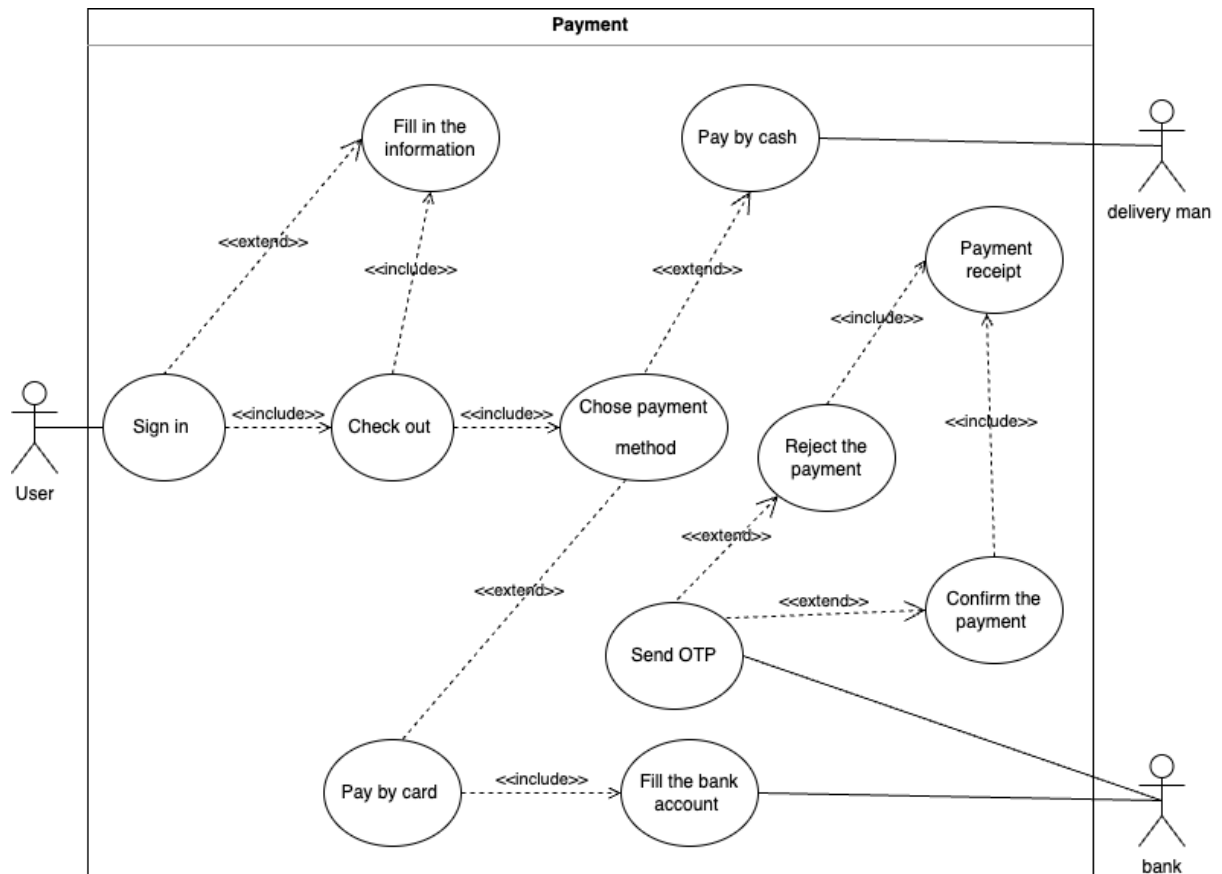
Generate Reports Component

ASSUME THAT THE ACTOR IS ALREADY LOGGED IN

GUI interfaces:

Screen 1
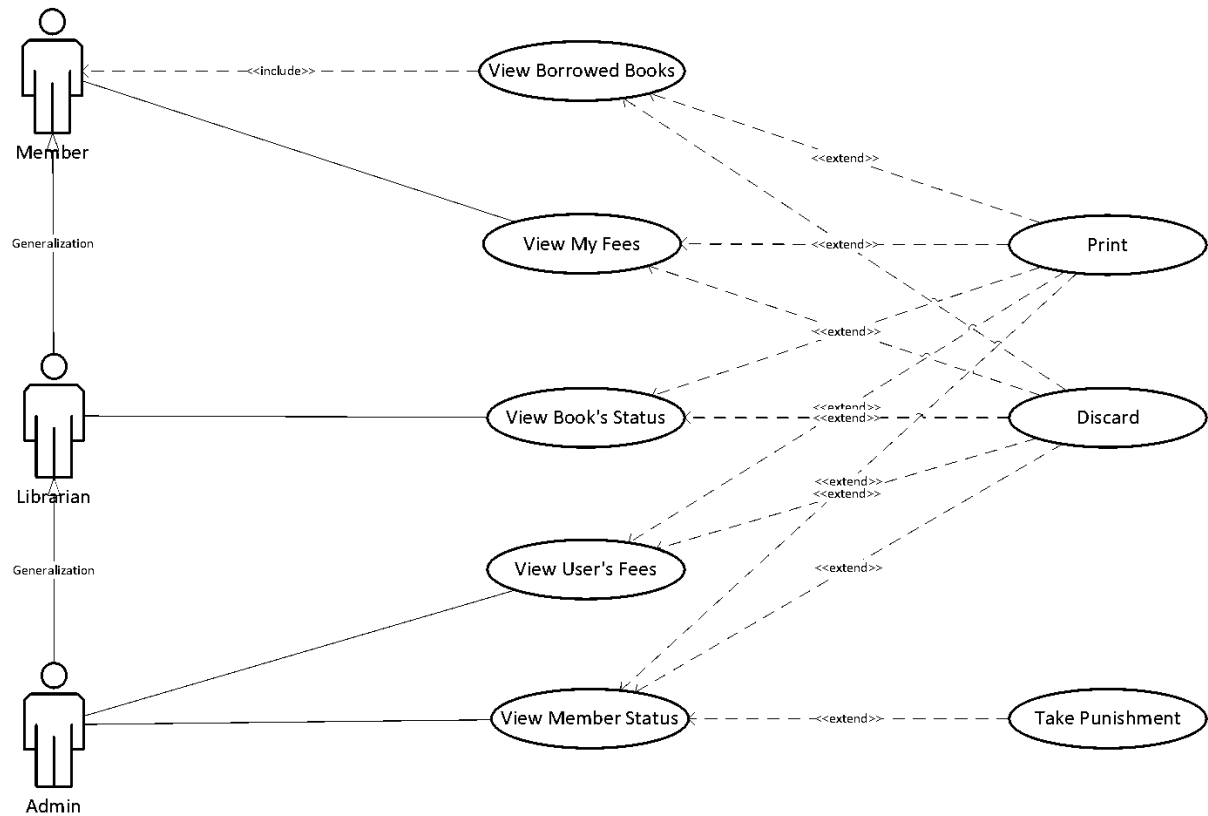
Library of Life

Trending

Search | 2023 |

Title:	Software Engineering
Author:	Ali Nasseri
Publisher:	Hussain Nooh

ISBN:	9780134763415
Status:	Ali Nasseri
Publication year:	12/01/2003

Go to Login

Follow us on

Screen 2

Library of Life

Trending

LOGIN

User Name / Email
HakosWorld

Password
****

Login

Forgot Password?

Create Account

Register

Follow us on

Screen 3

Library of Life

Trending

Create Account

Email
Example@gmail.com

Password
****

Birth Date
MM/DD/YYYY

Gender :

User Name
HakosWorld

Full Name
Husain Ali Nooh

Phone Number
34567890

Register

Follow us on

Library of Life

Manage Books

Search: 2023

Title:	Software Engineering	ISBN:	9780134763415
Author:	Ali Nasseri	Status:	Ali Nasseri
Publisher:	Hussain Nooh	Publication year:	12/01/2003

ADD   **REMOVE**

Menu

Go To

Component 1
▸ Manage Library
  • Manage Books
  • Issue Book
  • Return Book
  • Purchase Book
▸ Payments
  • Check out
  • Customer Information
▸ Reports
  • Books
  • Members

Follow us on

Library of Life

Modify account

Change Email
ahmedli@gmail.com

Change Password
****

Change Birth Date
12/01/2003

Gender : ♂ ♀

Edit User Name
a_librarian

Edit Full Name
Ahmed Ali Ahmed

Change Phone Number
36666666

Submit Change

TERMINATE ACCOUNT ⚠

Follow us on

Library of Life

Issue Book

Search: 2023

Title:	Software Engineering	ISBN:	9780134763415
Author:	Ali Nasseri	Status:	Ali Nasseri
Publisher:	Hussain Nooh	Publication year:	12/01/2003

Send   **Cancel**

Menu

Go To

Component 1
▸ Manage Library
  • Manage Books
  • Issue Book
  • Return Book
  • Purchase Book
▸ Payments
  • Check out
  • Customer Information
▸ Reports
  • Books
  • Members

Follow us on

Library of Life

Issue Book

Requests

#	Customer_ID	Book Number	Copies	Response	
1	03010411	1123	20	Accept	Reject
2	00010466	1123	5	Accept	Reject
3	02010487	1123	2	Accept	Reject
4	04010445	1123	1	Accept	Reject
5	04010445	1123	1	Accept	Reject

Menu

Go To

Component 1
▸ **Manage Library**
- Manage Books
- Issue Book
- Return Book
- Purchase Book

▸ **Payments**
- Check out
- Customer Information

▸ **Reports**
- Books
- Members

Follow us on

Library of Life

Return Book

select	Book Number	Title	Due Date	Notes
☐	20482	Database Management System	20-5-2023	
☐	46280	Linear Algebra Fundamentals	15-6-2023	
☐	58158	Computer Organization	22-6-2023	
☐	19452	Language Development ‖	1-7-2023	
☐	11241	Atomic Habits	4-7-2023	

Submit   **Cancel**

Menu

GO TO

Component 1
▸ **Manage Library**
- Manage Books
- Issue Book
- Return Book
- Purchase Book

▸ **Payments**
- Check out
- Customer Information

▸ **Reports**
- Books
- Members

Follow us on

Library of Life

Return Book

#	Customer_ID	Book Number	Late Fees	Status
1	03010411	1123	1.3 BD	pay
2	00010466	1123	5 BD	pay
3	02010487	1123	2 BD	pay
4	04010445	1123	1 BD	pay
5	04010445	1123	0.0 BD	Done

Menu

Go To

❖ Component 1
▸ **Manage Library**
- Manage Books
- Issue Book
- Return Book
- Purchase Book

▸ **Payments**
- Check out
- Customer Information

▸ **Reports**
- Books
- Members

Follow us on

Check Out :

Library of Life

▸ **Manage Library**
- Manage Books
- Issue Book
- Return Book
- Purchase Book

▸ **Payments**
- ○ Checkout
- Customer Information

▸ **Reports**
- Books
- Members

Cart Summary :
1-Since Book : 000$
QTY : 1
2-Math Book : 000$
QTY : 1
.
.
.
.

Sub-Total :0000$
Shipping : Free
Total : 0000$

○ Credit Card

● Pay Another Way

reCAPTCHA

Choose Your Alternative Payment Type

PAYMENT TYPE

(Default to first payment method) ⬍

Continue

Follow us on

Library of Life

Menu
Go To

▸ **Manage Library**
- Manage Books
- Issue Book
- Return Book
- Purchase Book

▸ **Payments**
- Checkout
- ○ Customer Information

▸ **Reports**
- Books
- Members

Customer Information :

Email Address : Hello @gmail.com

Customer Name : Ali Hassan

Phone number : +973 37777777

Address :
Home : 1333   Block : 777
Round : 4444   City : aaaa

Total Amount of many : 0000 $

Payment Details :
Card Number :
0000 0000 0000 0000   VISA

Expiry :   **CVC :**
00 /00   0000

Card Holder Name :
Ali Hassan Ali Hassan

Pay !

Follow us on

Library of Life

Menu
Go To

▸ **Manage Library**
- Manage Books
- Issue Book
- Return Book
- ○ Purchase Book

▸ **Payments**
- Checkout
- Customer Information

▸ **Reports**
- Books
- Members

Purchase Book

Search   2023

Title: Software Engineering   **ISBN:** 9780134763415
Author: Ali Nasseri   **Status:** Available
Publisher: Hussain Nooh   **Publication year:** 12/01/2003

Purchase   **Cancel**

Follow us on

Library of Life

Menu

Go To

▸ **Manage Library**
- Manage Books
- Issue Book
- Return Book
- Purchase Book

▸ **Payments**
- Checkout
- Customer Information

▸ **Reports**
- Books
- Members

Books

Search | 2023 | 🔲 🔍 |

Title:	**Software Engineering**	ISBN:	**9780134763415**
Author:	**Ali Nasseri**	Status:	**Available**
Publisher:	**Hussain Nooh**	Publication year:	**12/01/2003**

Print   **Discard**

Follow us on
⃝ in f

Library of Life

⌂ 🗛 💬 ? ⤷

Menu

Go To

▸ **Manage Library**
- Manage Books
- Issue Book
- Return Book
- Purchase Book

▸ **Payments**
- Checkout
- Customer Information

▸ **Reports**
- Books
- Members

Members

Search | 202104275 | 🔍 |

Name: Abdullah Mohammed Salman   **Status: Clear**
Fees: 0 BD
Borrowed Books: | ▼ |

DATABASE MANAGEMENT SYSTEMS	20482	Show Details
SOFTWARE REQUIREMENTS ENGINEERING	46291	Show Details
COMPUTER ORGANIZATION	58158	Show Details
LANGUAGE DEVELOPMENT II	19452	Show Details
INTRO. TO BUSINESS ADMINISTRATION	76241	Show Details

General use case:

Library Management System

- Search Book
- Borrow Book
- Return Book
- Purchase Book
- Payment
- Registeration
- Manage Library
- View Reports

User

Staff

Bank