

Software Requirements Specification

for

BooksBeep

Version 1.0 approved

Prepared by Debasmita Saha

Heritage Institute of Technology

20.01.2023

USE CASE Diagram
Requirements
31/12/23

13/01/23

Design
24/01/23

Class Diagram
14/02/2023

Testing & Modularization
14/03/23

Table of Contents

Table of Contents	ii
Revision History	iii
1. Introduction.....	1
1.1 Purpose.....	1
1.2 Document Conventions.....	1
1.3 Intended Audience and Reading Suggestions.....	1
1.4 Product Scope	1
1.5 References.....	2
2. Overall Description.....	2
2.1 Product Perspective.....	2
2.2 Product Functions	2
2.3 User Classes and Characteristics	3
2.4 Operating Environment.....	3
2.5 Design and Implementation Constraints	3
2.6 User Documentation	3
2.7 Assumptions and Dependencies.....	4
3. External Interface Requirements	4
3.1 User Interfaces	4
3.2 Hardware Interfaces.....	4
3.3 Software Interfaces	4
3.4 Communications Interfaces	5
4. System Features	5
4.1 User Log in	5
4.2 System Log in	6
4.3 Register New User	6
4.4 Register New Book.....	7
4.5 Delete Book	8
4.6 Search Book.....	9
4.7 Issue Book.....	10
4.8 View Issued Details	11
4.9 Return Book	11
4.10 Renew Book.....	12
4.11 Pay Fine.....	13
4.12 Due Date Alerts.....	14
4.13 Log out.....	14
5. Other Nonfunctional Requirements	15
5.1 Performance Requirements.....	15
5.2 Safety Requirements	16
5.3 Security Requirements	16
5.4 Software Quality Attributes	16
5.5 Business Rules.....	16
6. Other Requirements	16
Appendix A: Glossary.....	16
7. Use Case Diagram.....	17
8. Sequence Diagram.	18
9. Class Diagram.....	25

10. Black Box testing.....	26
11. Modularization.....	27
12. List of Functions.....	28
13. IFPUG.....	32
14. COCOMO.....	33
15. Intermediate COCOMO.....	34

Revision History

Name	Date	Reason For Changes	Version
BooksBeep	20.01.2023	-	1.0

1. Introduction

1.1 Purpose

BooksBeep is a web-based application for organizing and managing library tasks like entering the records of new books, retrieving the details of books available in the library, issuing, renewing, and returning books along with some unique features. This Library Management System allows even users like students or teachers to login and view the status of books issued by them, pay fine online and renew books remotely without the need to visit the library. For other library management tasks, system log in is required and is used by the librarians and admins. Users can also choose to get an SMS alert 3 days prior to the due date of returning books to - either renew or return them on time . Thus, making it very easy for them to avoid late charges and beneficial for the library as well since when users hold back books for a long time and make them inaccessible for others, it is quite troublesome.

1.2 Document Conventions

This Document was created based on the IEEE template for System Requirement Specification Documents.

1.3 Intended Audience and Reading Suggestions

- The system can be used or incorporated in any colleges, universities or medium to large sized libraries where most of the work of keeping records of books and members is done manually on pen and paper. The system's user interface is designed in such a way that anyone can easily handle it and use all the features it provides at ease. Anyone who has the most basic knowledge or skill to use a computer can use this web application. The system does not demand too much computational resources and can run on very old models too if they can support the minimum requirements.*
- BooksBeep can also act as a replacement for already existing Library Management System used by an institute due to its added on features. It can be very useful for students and faculty members who need assistance in keeping track of their issued books , easily at anytime and from anywhere. It can save them a lot of time visiting the library and also avoid long queues for fine payment or renewal of books.*

1.4 Product Scope

The problems that can be addressed and solved through this project includes elimination of paper-work in the library, recording every transaction in a computerized manner to avoid mismanagement, implementation of online late payment to ease the process of paying fines and increase transparency ,avoid mistakes caused mostly due to human error ,the database design can handle and store information about all the books in the inventory and also has a secure and transparent environment for keeping track of books from librarian's as well as users' perspective. Admins and librarians(system users) can easily through the web application with minimal time requirements easily add, update, delete books , manage users and other library tasks.

Even normal users like students can login into this Library management system, see the details of issued books and get due date alert on time to return or renew their books which is an added on feature and a really necessary one as due to the busy lifestyles, forgetting due dates for returning books is very common and sometimes returning them very late also results in a significantly large fine heavy on pockets. Plus holding back books for a very long time may result in losing them and it becomes difficult for other people who may want to avail those books from the library otherwise.

1.5 References

IEEE Software Requirement Specifications Template:
https://web.cs.dal.ca/~hawkey/3130/srs_template-ieee.doc

2. Overall Description

2.1 Product Perspective

Traditional Library management systems do not allow a normal user to login and check details of his/her books. Only a librarian can check and tell the status of books, that too when one visits the library. But knowing how many books one has issued and their due dates helps one keep them responsibly. Also remembering due dates is very difficult, especially when books are issued at different times, checking on the books frequently to see their return due dates individually is cumbersome but returning them on time is important as they are library's property and many people may need to access those. Holding on to books for a long time troubles both the user and the library as the user will have to pay a significantly large amount of fine and the library books can get misplaced or lost.

2.2 Product Functions

The major functions of the product are: -

- Register New User
- Register New Book
- Delete Book
- Search Book
- Issue Book
- View Issued details
- Renew Book
- Return Book
- Late fine payment
- Due date alerts

2.3 User Classes and Characteristics

<u>USER CLASSES</u>	<u>CHARACTERISTICS</u>
User	A <i>normal user like student/teacher in case of an educational institute or any other person who is member of the library, has access to the features like-</i> <i>Search, log in, view issued book details, pay fine, renew books, turn-on/off due date alerts.</i>
Librarian	A <i>librarian can log into the system through System log in and can access many more features compared to a normal user. Librarians like the usual library staffs of any library, are in charge of viewing ,issuing books to the users, returning books, renewing books, paying late fine on user's behalf. All such functions belong to this class .</i>
Admin	An <i>admin has access to the entire library management system and all its features and can also modify the database ,add or delete books from the library inventory and manage users.</i>

2.4 Operating Environment

The product is a web-based application and can run on any browser accessible devices that support the minimum requirements to run the application.

2.5 Design and Implementation Constraints

The whole Project is divided in two parts: the front end and the back end.

Front end

The front end is designed using HTML , CSS, JavaScript.

Backend

The backend is designed using SQL, Python(Django) Framework.

2.6 User Documentation

A document containing the step by step manual of how to use the application for different functions should be provided in the web application.

2.7 Assumptions and Dependencies

- In the proposed system, we assume that each member will be having an identity card with an user id(roll number /identity number) which is unique and can be used for the library book issue, fine payment etc. Whenever users wish to take a book, the system will check both the book details as well as the user details ,add the book to user account and then update library database.
- It is also assumed that the users have proper internet connection on their devices to access this web application.

3. External Interface Requirements

3.1 User Interfaces

The web based Graphical User Interface(GUI) provided will be completely user friendly and will ensure minimum number of clicks(or touches in case of a hand-held device) to perform the desired task.

3.2 Hardware Interfaces

For Desktop/Laptop:-

- Intel core 2 duo / AMD Ryzen 3 or higher processors to provide reliability and stability and run the web application on the pc for a long time.
- A minimum of 2 GB RAM is necessary for fast reading and writing capabilities it and will in turn support processing transactions smoothly.

For Mobiles/Tablets:-

- Snapdragon 430 /Mediatek Helio g35/p20 processor or higher chipset.
- Minimum 1 GB RAM for the Web application to operate smoothly.

3.3 Software Interfaces

For Desktop/Laptop: -

- Windows XP or higher, ChromeOS, Linux ,MacOS.

For Mobiles/Tablets: -

- Android 4.1 or higher.
- iOS 7 or higher.

3.4 Communications Interfaces

- This product needs to communicate with the payment partner bank server. Communication can be done using HTTP requests and APIs.
- Sending alert for due dates can be done through SMS APIs.

4. System Features

This section of the SRS describes the various system features and the functional requirements of the product.

4.1 User Log in

4.1.1 Description and Priority

This feature is used to log into the system. Users are required to enter their user id and password .The user id and password will be verified in order to be allowed to enter the system.

4.1.2 Stimulus/Response Sequences

S- From the list of options available on the page after opening the web application for the first time , 'User Log in' is clicked.

R- A log in page appears.

4.1.3 Functional Requirements

- **REQ-1:**The log in page should open on clicking the log in option where the user can fill in his/her user id and password to log in.
- **REQ-2:**System should allow the user to log in only when correct credentials are provided.If wrong user id or password is entered, the system should not allow the user to log in and it will display an error message- "Invalid user id or password" and the log in page should refresh for the user id and password to be entered again. Users should have an option change or update their credentials if they want (This step requires verification in case of a normal user but for an admin, for changing credentials can be done in 'User management' section of the system homepage and should not require verification.)
- **REQ-3:**After successful log in, the user should be taken to the homepage inside the library management system.

4.2 System Log in

4.2.1 Description and Priority

This feature is used by the admin or librarian to log into the system.

4.2.2 Stimulus/Response Sequences

S- From the list of options available on the page after opening the web application for the first time , ' System Log in ' is clicked.

R- System Log in page appears.

4.2.3 Functional Requirements

- **REQ-1:** The System log in page should open on clicking the 'System log in' option where the admin /librarian can fill in the system user id and password to log in.
- **REQ-2:** System should allow the log in only when correct credentials are provided.If wrong user id or password is entered the system should not allow log in and it will display an error message-“Invalid user id or password” and the log in page should refresh for the user id and password to be entered again.
- **REQ-3:** After successful system log in ,the admin/librarian should be taken to the system homepage which contains all the options for management of books and users.

4.3 Register New User

4.3.1 Description and Priority

This feature can be used to register new users to create accounts.

4.3.2 Stimulus/Response Sequences

S- From the list of options available on the page after opening the web application , 'Register New User' is clicked.

R- The Register New User page opens.

4.3.3 Functional Requirements

- **REQ-1:** Register New User page should open upon clicking this option, which should to enter the following:-
 - New Member's name, department, section(if student)
 - Unique college id ,Email id,phone number.
 - A password for his/her account
- **REQ-2:** System should send an OTP to the provided phone number and email and ask to enter the same in order to verify the mobile number and email provided.
- **REQ-3:** System should create a new user account if no previous accounts exist with the same details and all fields are filled properly including the OTP and should show a "User registered successfully" message.
- **REQ-4:** If wrong college id is entered or user with same id already exists then system should display an error message-“Invalid id/ Account with same id already exists” and the page should refresh for the details to be entered again.

4.4 Register New Book

4.4.1 Description and Priority

This feature allows the admin to add new books to the library .

4.4.2 Stimulus/Response Sequences

S - On the system home page, under the 'Book management' section, 'Register New Book' is clicked.

R- Register New Book page appears.

4.4.3 Functional Requirements

- **REQ-1:** Only an admin should be able to access this feature after log in.
- **REQ-2:** The Register New Book page should open on clicking and the books details should be allowed to enter like:
 - Book Name, Author, Publisher's Name, Date of Purchase
 - ISBN Code, Book Price, Edition Number, Volume Number (if any)
 - Genre or the subject to which the book belongs to.
- **REQ-3:** If the book already exists, then the total number should be incremented . Otherwise the new book entry should be created along with an accession number (book id) and added to the library database and a message – “Book added successfully” should be displayed.

4.5 Delete Book

4.5.1 Description and Priority

This feature allows the admin to remove books from the library database.

4.5.2 Stimulus/Response Sequences

S - On the system home page, under the ‘Book management’ section, ‘Delete Book’ option is clicked.

R- Delete Book page appears.

4.5.3 Functional Requirements

- **REQ-1:** Only an admin should be able to access this feature after system log in.
- **REQ-2:** The Delete Book page should open on click by the admin and book name can be typed in the search bar. Additional details like:-
 - Accession number, ISBN Code, Edition Number, Volume Number
 - Author, Publisher's Name

can be entered in the spaces for fields provided. Once the search fields match a particular book, option should be there to select and delete it.

- **REQ-3:** The Admin should be asked if he really wants to delete the book from the library database. Option to ‘confirm deletion’ or ‘cancel’ should be given.
- **REQ-4:** The copy of the book should be erased from the library database after confirming deletion and a message “Book removed from database successfully” should be displayed.
- **REQ-5:** If the book is not found ,then “Book not found message” should be displayed.

4.6 Search Book

4.6.1 Description and Priority

This feature allows anyone to search books based on book name , author name , publication etc .The feature does not require log in.

4.6.2 Stimulus/Response Sequences

S - On the first page opened after launching the web application , ‘Search Book’ is clicked.

R- Search page opens.

4.6.3 Functional Requirements

- **REQ-1:** The search page should open on clicking the ‘Search Book’ tab.
- **REQ-2:** The page should ask to enter one or more of details of the book like:-
 - Book Name
 - Author Name
 - Publisher
 - Subject to which the book belongs to

And based on the one or more fileds entered by the user, the keywords should be matched and results of search should be displayed.

- **REQ-3:** System should show a message if no book is found matching the keywords – “no matching books found”.

4.7 Issue Book

4.7.1 Description and Priority

This feature is for issuing books for users by a librarian ,as per their choice within a limited number of books.

4.7.2 Stimulus/Response Sequences

S - The ‘Issue Book’ option is clicked on the System home page .

R- The ‘Issue Book’ page opens.

4.7.3 Functional Requirements

- **REQ-1:** The ‘Issue Book’ page should open on clicking the ‘Issue Book’ option.
- **REQ-2:** The page should allow the librarian to fill in the details to issue the book :-
 - Accession number (can be typed or scanned if books have barcodes on them)
 - User id of the user to whom the book will be issued.
- **REQ-3:** Against the accession code, from the database the book name, author , ISBN code ,publisher’s name etc should be displayed for the librarian to check if it is the correct book that is being issued and asked for confirmation.
- **REQ-4:** Upon confirmation, system should record this change in the user’s account and library database. After that, a message should be displayed – “Book Issued successfully”.
- **REQ-5:** System should ensure that a user cannot have more than 5 books issued to the account if trying to do so ,an error message should be displayed – “Cannot issue book, maximum number of books already issued”.

4.8 View Issued Details

4.8.1 Description and Priority

This feature is used to view the details of all the books currently issued to an user.

4.8.2 Stimulus/Response Sequences

S- *The ‘View Issued Details’ option is clicked on the home page after system/user log in.*

R-A *A page opens which has all the information about the user account and issued books.*

4.8.3 Functional Requirements

- **REQ-1:** *The ‘Issued Details’ page should open on clicking the option and should contain all the information about issued books and user’s account.*
- **REQ-2:** *For librarians ,an option to enter the user id for whom the he/she wants to see all the issued book details should be provided.Upon entering the user id,the issued books details of that user should be displayed.*

4.9 Return Book

4.9.1 Description and Priority

This feature is for returning books for a user by a librarian before, on or after the due date.

4.9.2 Stimulus/Response Sequences

S - *The ‘Return Book’ option is clicked on the home page after system log in.*

R- *The ‘Return Book’ page opens.*

4.9.3 Functional Requirements

- **REQ-1:** The ‘Return Book’ page should open on clicking the ‘Return Book’ option.
- **REQ-2:** The page should allow the librarian to fill in the details to return the book :-
 - Accession number (can be typed or scanned if books have barcodes on them)
 - User id of the user to who is returning the book .
- **REQ-3:** Against the accession code ,from the database ,it should be checked that it is the same user id to whom the book was issued, if so the librarian should be asked to confirm the return.
- **REQ-4:** Upon confirmation, System should check and show a message if the due date has expired and it goes to the fine payment section. Then after successful fine payment ,system should record this change in the user’s account and library database and after that, a message should be displayed –“Book Returned successfully”.

4.10 Renew Book

4.10.1 Description and Priority

This feature is used for renewing books . It can be used by the librarian as well as the user himself/herself .

4.10.2 Stimulus/Response Sequences

S - The ‘Renew Book’ option is clicked on the home page after system log in / User log in.

R- The ‘Renew Book’ page opens.

4.10.3 Functional Requirements

- **REQ-1:** The ‘Renew Book’ page should open on clicking the ‘Renew Book’ option.

- **REQ-2:** For librarians ,an option to enter the user id for whom the renewal of a book will be done should be provided.Upon entering,the issued books details of that user should open.And for normal users directly the list of issued books should be displayed for selection of the book which needs to be renewed.
- **REQ-3:** System should show a message if the due date has expired takes the user to the fine payment section.
- **REQ-4:** System should keep a count against the user id to ensure he/she cannot renew a book more than three times. If so an error message should be displayed that “Book cannot be renewed more than three times”.
- **REQ-5:** System will update the due date for the books if conditions are satisfied and display a message “Book renewed successfully”.

4.11 Pay Fine

4.11.1 Description and Priority

This feature is used for paying late fine for books . It can be used by the librarian to pay fine on behalf of the user ,as well as by the user himself/herself .

4.11.2 Stimulus/Response Sequences

S- *The ‘Pay Fine’ option is clicked on the home page after system log in / User log in.*

R-*The user is navigated to payment partner’s page.*

4.11.3 Functional Requirements

- **REQ-1:** The user or librarian should be redirected to the fine payment page after clicking the option.
- **REQ-2:** The outstanding dues should be calculated forwarded to the payment page where the user can fill in payment mode and make payment,then if successful,a message –“Fine payment successful” should be shown and user should be navigated back to BooksBeep’s homepage.The transaction should be noted and status should be updated in the database.

- **REQ-3:** If for any reason payment fails the system should be able to make sure after redirection the user should again be able to go back and pay the fine Account and the fine should be updated after that.

4.12 Due Date Alerts

4.12.1 Description and Priority

This feature is used to alert the user 3 days before the due date for returning books.

4.12.2 Stimulus/Response Sequences

S- After log in , the 'Due Date Alerts' option is clicked.

R - Due Date Alerts page opens.

4.12.3 Functional Requirements

- **REQ-1:** The 'Due Date Alert' page should open on clicking the option.
- **REQ-2:** The user should be allowed to set a due date alert destination,if he/she is turning on the feature for the first time or wants to change the alert destination.An OTP is sent to the provided phone number and email to verify the alert destination.The user should be allowed to enter the same for verification,upon verification a message—"Alert destination set successfully" should be displayed.
- **REQ-3** The user should be allowed to turn on/off this feature.
- **REQ-4:** The system should store the details and that this user requested to receive alerts.The data will be stored in the database and fetched by SMS APIs to generate SMS reminders for the user 3 days before the due date of returning books.

4.13 Log out

4.13.1 Description and Priority

This feature is used to log out of the Library management system after finishing using it.

4.13.2 Stimulus/Response Sequences

S- *The ‘Log out’ option is clicked on the home page .*

R- *User is logged out and the first page which opens after launching the Library Management System is relaunched.*

4.13.3 Functional Requirements

- **REQ-1:** *The user should be logged out from the web application after clicking the log out option.*
- **REQ-2:** *A message-“Logged out successfully” should be displayed after log out.*

5 Other Nonfunctional Requirements

5.1 Performance Requirements

- *The web application should be fast and its load time for every page should not be more than three seconds.*
- *Booksbeep should be reliable and members should be able to access their account 95% of the time without failure.*

5.2 Safety Requirements

- *The system should be recoverable, in the case of unplanned system downtime, all features should be made available again within 1 working day.*
- *The system should be able to handle up to 500 members logging into the system and using it at the same time.*
- *There must be regular checks to ensure maintainability of the due date alerts.*

5.3 Security Requirements

- Only the Admin should have privilege to manage books- add or delete book, modify user details and change fine if ever required.
- Normal users should not be able to access features provided specially after system log in.

5.4 Software Quality Attributes

- The web application's interface must be user-friendly .
- Response time should not be more than two second for users.
- When making minor changes to the database by the Admin, the rest of the system should be up and running.

5.5 Business Rules

- Bank partner services
- SMS API services

6 Other Requirements

- **Database-** MYSQL is used as a database language as it is easy to maintain and retrieve records by simple queries which are in English language which are easy to understand and easy to write.
 - **Development tools and Programming language-** HTML is used to write the whole code and develop web pages with CSS, JavaScript for styling work and PHP for server side scripting.
 - **Hard Disk Space-** Minimum 1TB Space for database storage.

Appendix A: Glossary

SRS-Software Requirements Specification

S-Stimulus

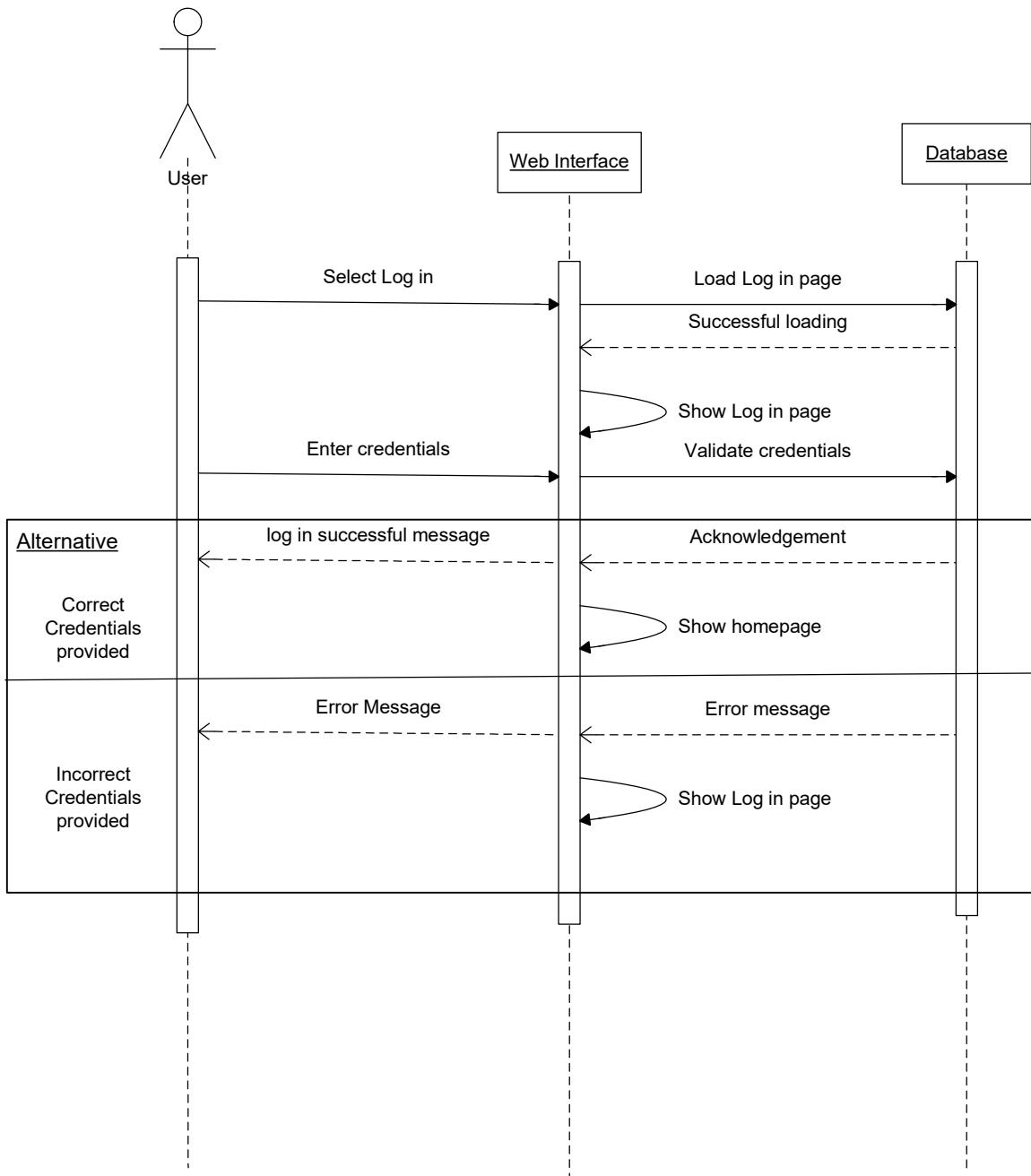
R-Response

REQ-Requirements

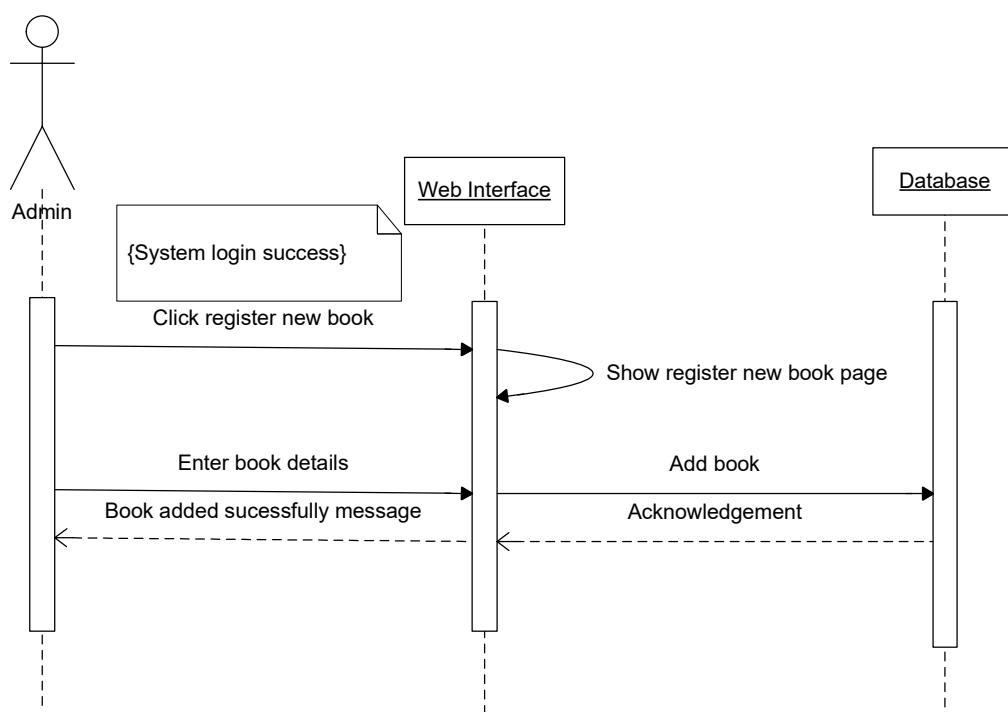
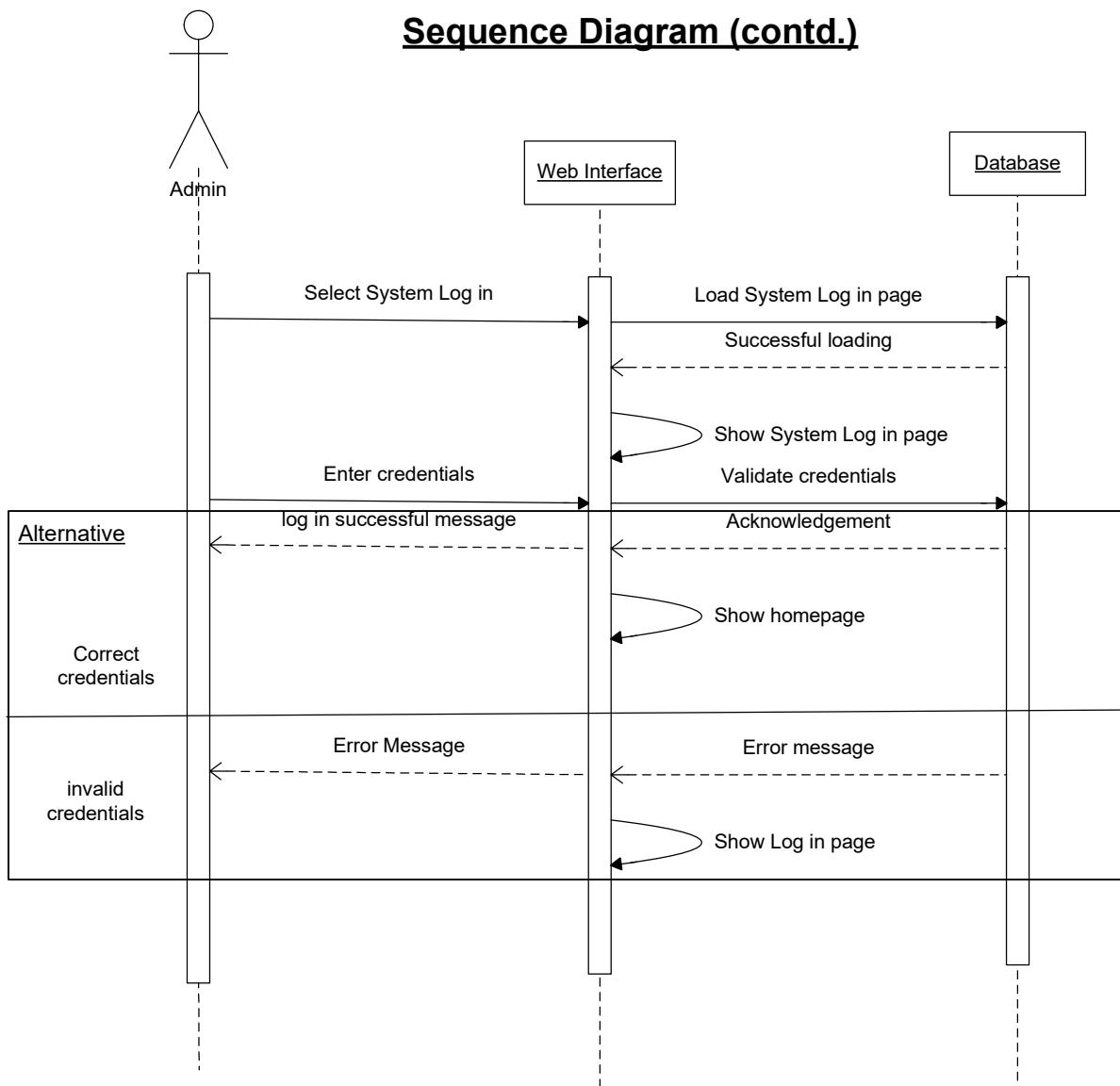
Use Case Diagram

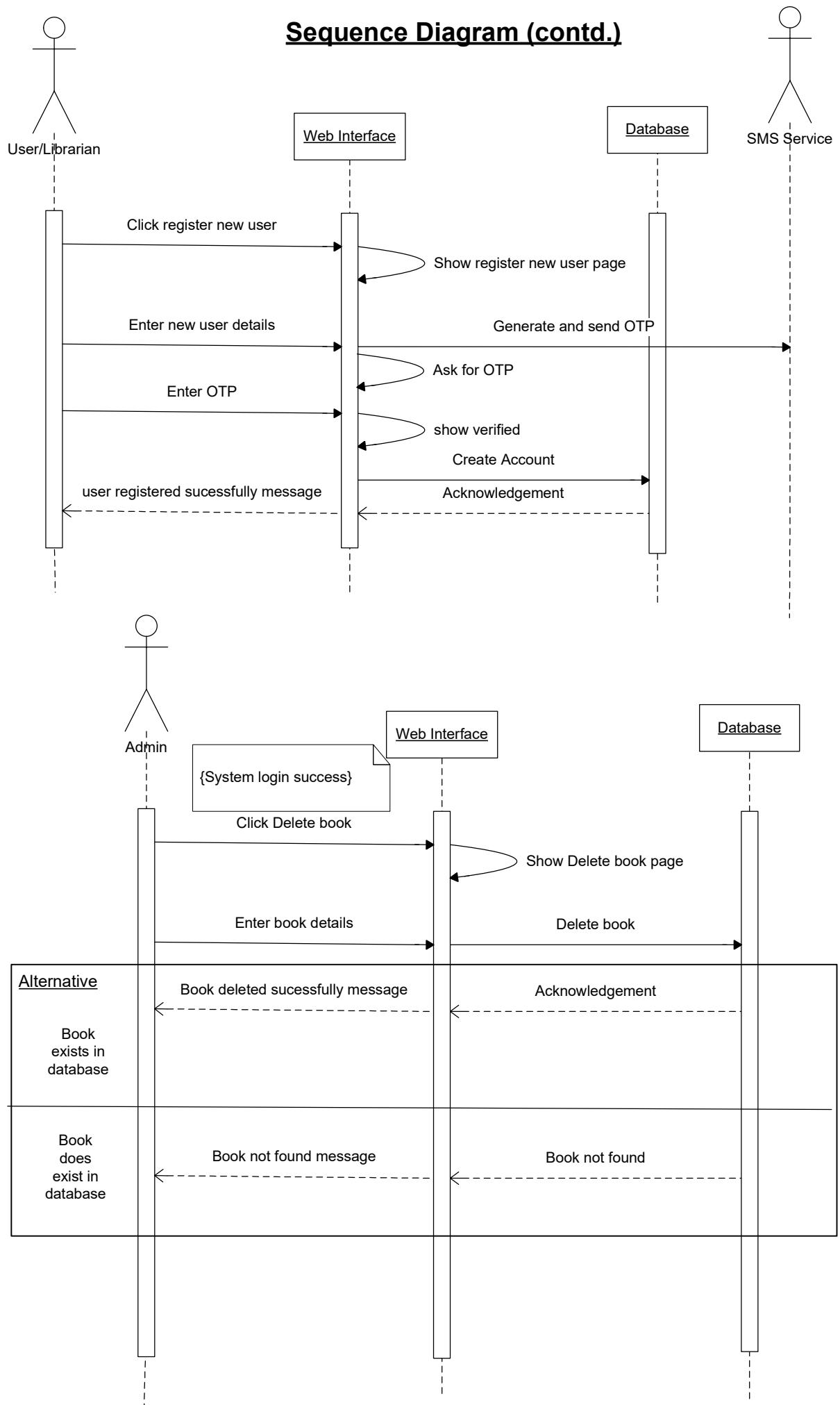


Sequence Diagram

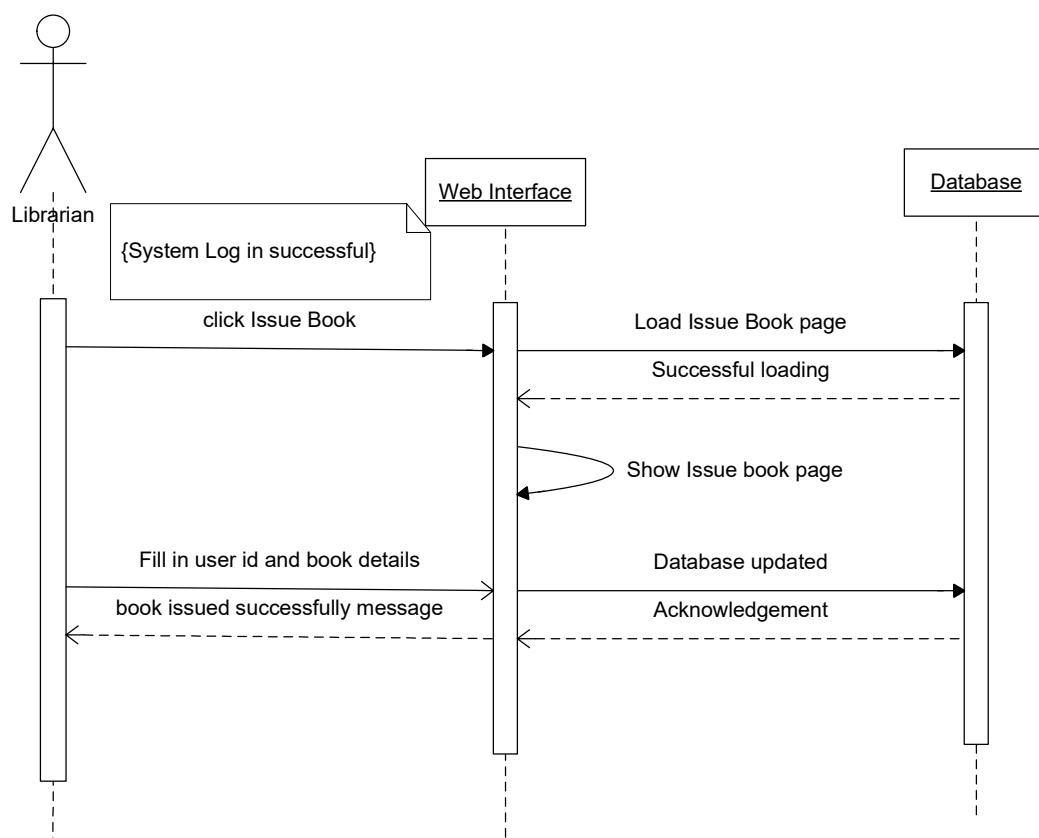
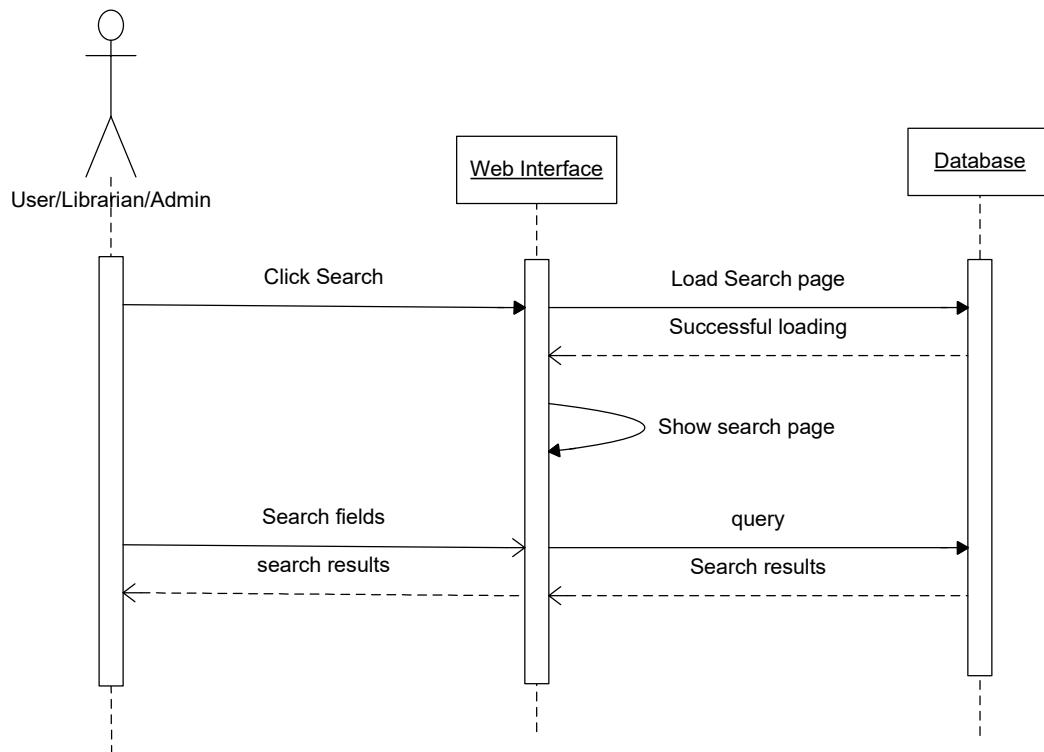


Sequence Diagram (contd.)

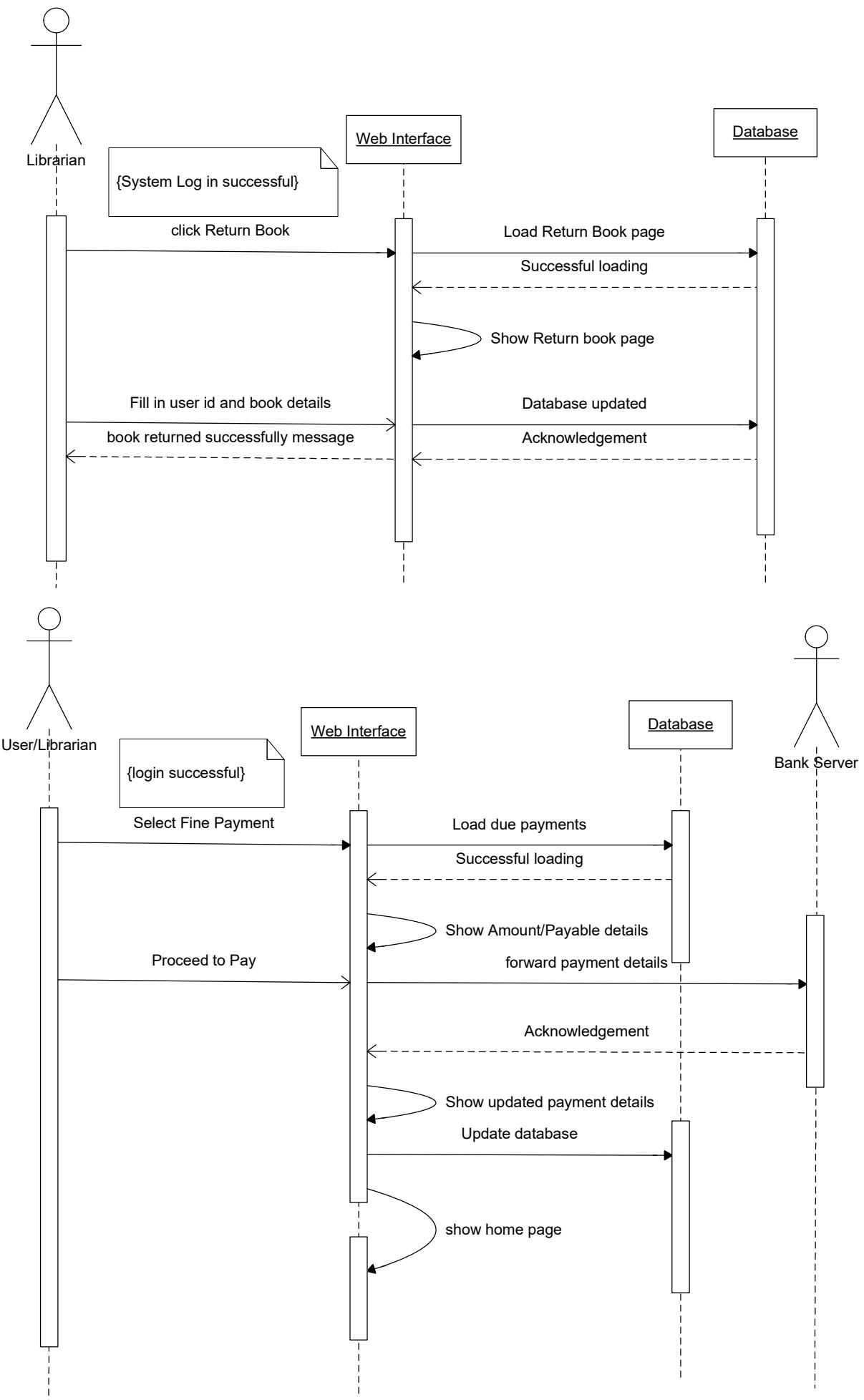




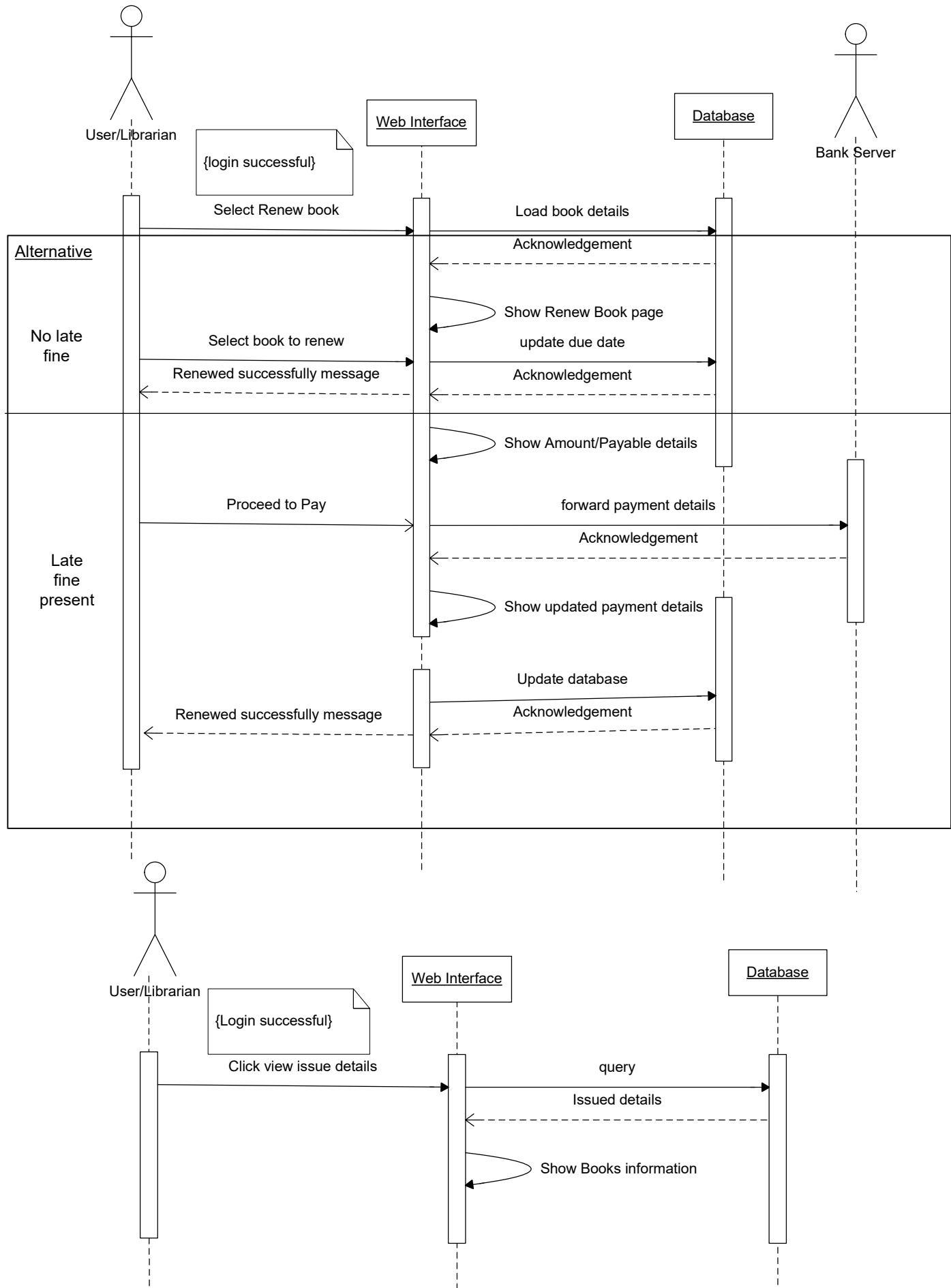
Sequence Diagram (contd.)



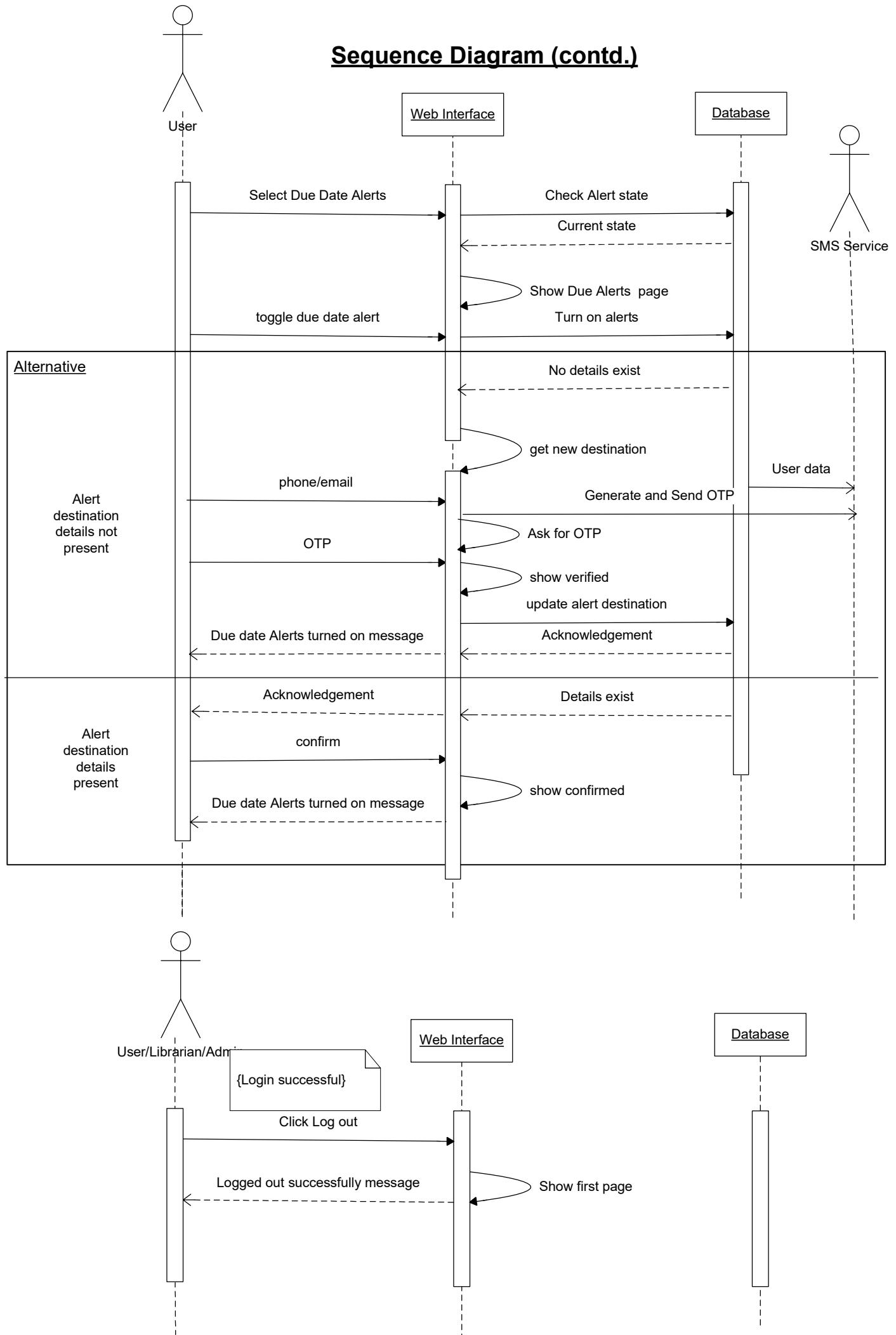
Sequence Diagram (contd.)



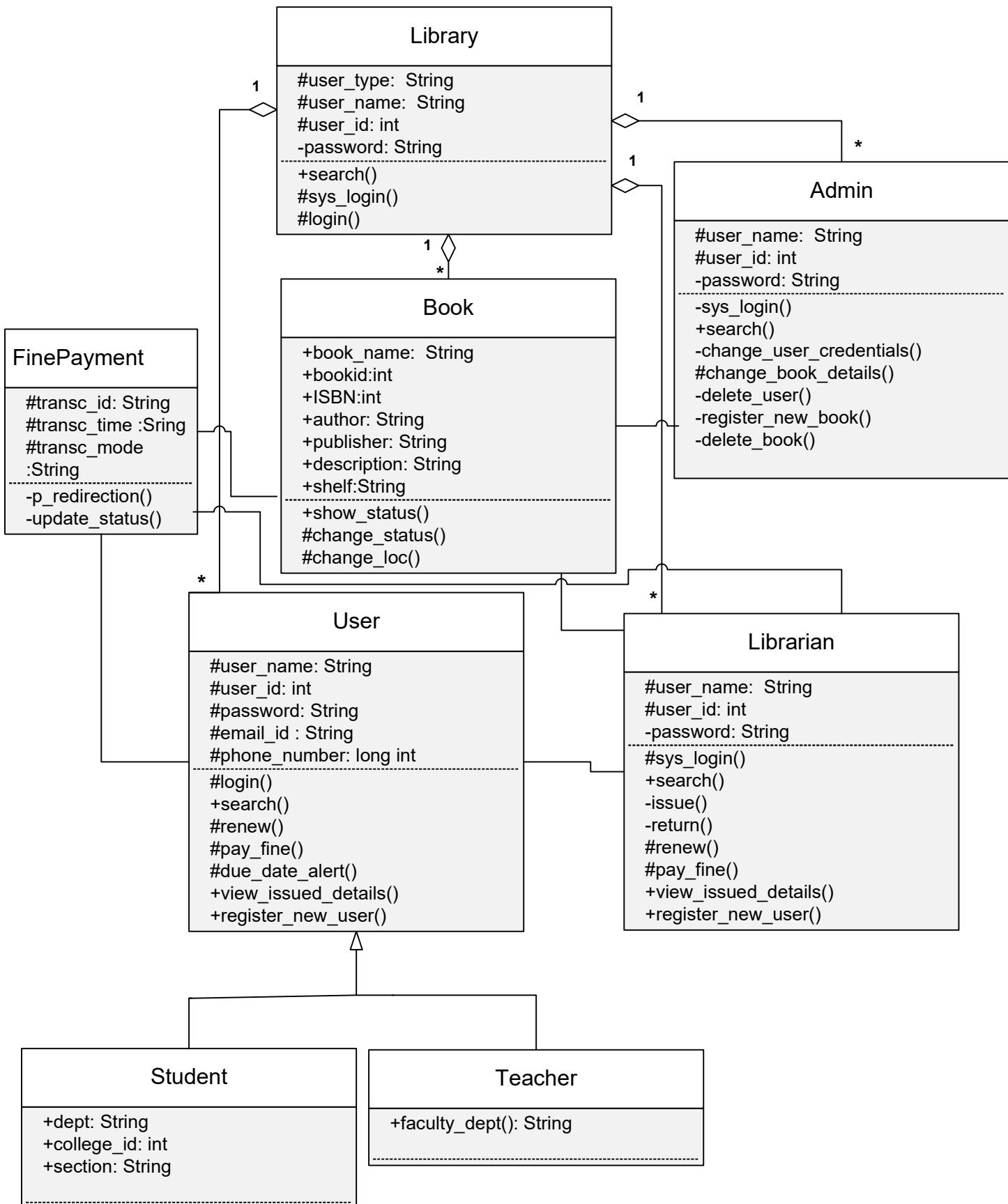
Sequence Diagram (contd.)



Sequence Diagram (contd.)

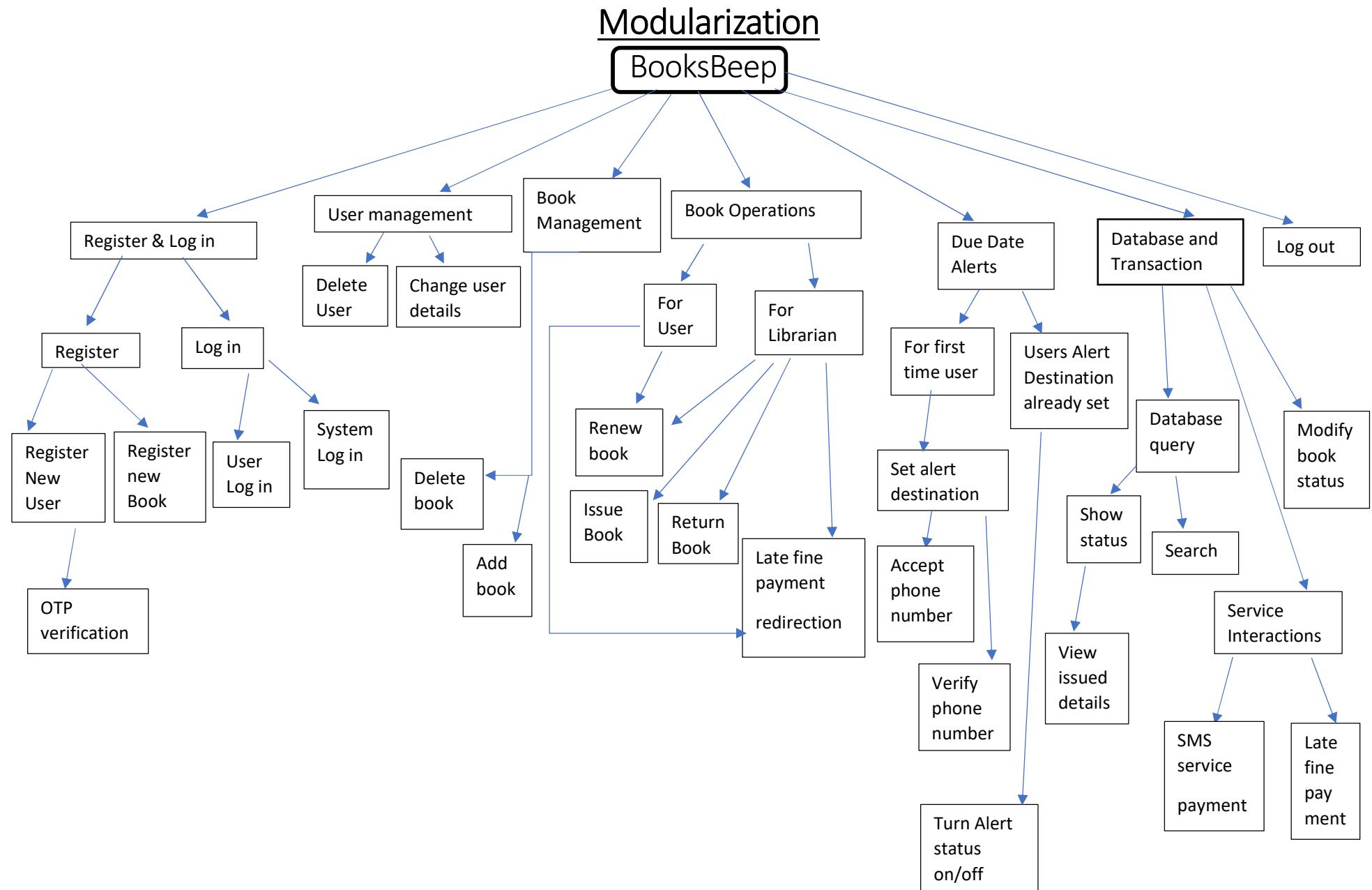


Class Diagram



Testing

Test Cases for:	<i>BooksBeep</i>				
Author	<i>Debasmita Saha</i>				
Date Created	<i>20-02-2023</i>				
Last Updated	<i>Not Applicable</i>				
Prerequisites:	<i>The user has an active internet connection and a web browser installed.</i>				
System Module:	<i>User Log in, Search Book, Issue book and Due Date Alerts.</i>				
Test Case #	Test Case Description	Test Procedure	Input Data	Expected Result	Comments
User Log in	Used to log into the system if correct credentials are provided	The credentials are verified with the database	User id and password	'Logged into the system successfully' message ,navigates to homepage inside the Library Management System. Error message - 'Invalid user id or password',the log in page refreshes for credentials to be entered again.	Correct credentials entered, hence log in successful. Incorrect credentials entered, hence log in unsuccessful.
Search Book	Shows search results upon entering one or more fields related to the book	Based on the fields entered, the book is searched in the database.	Book Name, Author Name, Publisher, Subject(entering at least 1 is must)	The search results-Total no. of copies of the book, issued and available books their ISBN,shelfnumber,bookid. Message shown 'no matching books found'	Search results with similar keywords shown. Incorrect fields entered. Book does not exist in database.
Issue Book	Issues books to a user	Librarian enters book id and user id,Then selects the book and confirms issue.Book gets issued to the user and the database gets updated.	Book id /Accession number ,user id	'Book issued successfully' message. 'Cannot issue book,maximum number of books already issued'.	Book found and issued to user successfully. User already has the maximum no. of books issued to his/her account. (5)
Due Date Alert	Send SMS alert to user 3 days prior to the due date of returning books when turned on	User turns this option on /off and the changes are noted and saved in the database through which the SMS APIs fetch data.	Phone number ,email id (Required only if turning this feature on for first time also OTP for verification)	'Alert destination set successfully' , 'Due date alerts turned on ' message.	Alerting service turned on and noted in the database.



List of Functions

OTP verification:

- accept_phno()
- accept_email()
- generate_otp()
- send_otp_ph()
- send_otp_email()
- tracktime() : run_timer(), show_timer()
- accept_otp()
- check_otp() : match_otp(), show_msg
- resend_otp()
- show_msg()

Register New Book:

- accept_details()
- create_entry()
- add_to_db()
- show_msg

User Log in:

- accept_userid()
- accept_password()
- verify_userid_password() :query_db(), match_results()
- show_msg()
- navigate_to_user_LMS()

System Log in:

- accept_userid()
- accept_password()
- verify_userid_password() : query_db(), match_results()
- show_msg()
- navigate_to_sys_LMS()

Delete User:

- accept_userid()
- delete_user():remove_from_db()
- update_db()
- show_msg()

Change user details:

- accept_userid()
- accept_email()
- accept_phno()
- accept_change():make_change(),update_db()
- show_msg()

Delete Book:

- accept_details()
- remove_from_db()
- update_db()
- show_msg()

Add book:

- register_new_book()
- inc_count()
- update_db()
- show_msg

Renew book:

- accept_details()
- check_fine() :cal_fine(), display_fine()
- check_count():show_count(), show_msg()
- update_count()
- update_datedate()
- update_db()

Issue Book:

- accept_details()
- accept_userid()
- validate_details()
- add_to_user()
- show_msg()
- update_db()

Return book:

- accept_details()
- query_db()

- find_user()
- remove_from_user()
- add_to_db()
- update_db()

Late Fine redirection:

- accept_user_details()
- check_fine():cal_fine(), show_fine()
- open_pp_website()

Late Fine payment:

- accept_payer's_details()
- show_fine()
- confirm_fine()
- accept_payment()
- show_msg()
- record_trans()
- update_db
- redirect_to_homepage()

Modify Book details/status:

- accept_status()
- change_details():update_db
- show_msg()

Turn alert status on/off:

- accept_value()
- update_db()
- show_msg()

Set alert destination:

- accept_email()
- accept_phno()
- send_otp()
- show_msg()
- update_db()

Accept phone number:

- create_variable()
- input_phno()
- assign_value()
- store_in_db()

Verify phone number:

- accept_phno()
- otp_verification()

View issued details:

- accept_userid()
- query_db()
- show_acctn_details() : tabulate_details() , display_table()

Search :

- accept_details()
- show_results() : match_keywords() , display_matched()
- show_msg()

SMS service payment:

- send_money() : fetch_LMS_details() ,fetchAPI_details() , fetch_amt()
,pay_service()
- record_trans()
- update_db()
- show_msg()

Log out:

- navigate_LMS_startpg()
- show_msg()

Function Point Analysis (FPA) Worksheet												
Measurement Parameter	Count			Weightage						FP Count		
	Simple	Average	Complex	Simple	Average	Complex						
# of External Inputs (EI)	72	10	3	X	3	4	6	=	274			
# of External Outputs (EO)	88	2	8	X	4	5	7	=	418			
# of External Inquiries (EQ)	51	9	2	X	3	4	6	=	201			
# of Internal Logical Files (ILF)	17	0	1	X	7	10	15	=	134			
# of External Interface Files (EIF)	68	8	10	X	5	7	10	=	496			
Unadjusted Function Points (UFP)												UFP = 1,523
General System Characteristics (GSC): - rate each factor below on a scale of 0 to 5 for DI	0 = No Influence	1 = Incidental	2 = Moderate									
Degree of Influence (DI)	3 = Average	4 = Significant	5 = Essential									DI
01. Does the system require reliable backup and recovery?												4
02. Are data communications required?												2
03. Are there distributed processing functions?												1
04. Is performance critical?												2
05. Will the system run in an existing, heavily utilized operational environment?												1
06. Does the system require on-line data entry?												5
07. Does the on-line data entry require the input transaction to be built over multiple screens or operations?												2
08. Are the master files updated on-line?												5
09. Are the inputs, outputs, files, or inquiries complex?												2
10. Is the internal processing complex?												1
11. Is the code designed to be reusable?												4
12. Are conversion and installation included in the design?												0
13. Is the system designed for multiple installations in different organizations?												1
14. Is the application designed to facilitate change and ease of use by the user?												4
Total DI = 34												
VAF = 0.99												
Adjusted Function Points (AFP)												AFP = 1,508
Any Assumptions made should be listed below.												

Constructive Cost Model (COCOMO)

Modules	Number of Functions
Register & Log in	24
User Management	12
Book Management	8
Book Operations	27
Due Date Alerts	14
Database and Transaction	10
Service Interactions	16
Log out	2
TOTAL	113

Project Type: Organic

Each function should have approx. 10 lines of code (loc)

Therefore, Total loc = $113 \times 10 = 1130$ loc

KLOC = $1130/1000 = 1.13$

Effort = $2.4 \times (\text{KLOC})^{1.05} = 2.4 \times (1.13)^{1.05} = 2.73 \sim 3$ PM

Development Time(T_{dev}) = $2.5 \times (\text{Effort})^{0.38} = 2.5 \times (2.73)^{0.38} = 3.66 \sim 4$ months

Intermediate COCOMO

Cost Drivers	Ratings
Product Attributes :	
Required Software Reliability	High (1.15)
Size of Application Database	Very High (1.16)
Complexity of the product	Nominal (1)
Hardware Attributes :	
Run-time performance constraints	Nominal (1)
Memory constraints	Nominal (1)
Volatility of virtual machine environment	Low (0.87)
Required turnabout time	Nominal(1)
Personnel Attributes :	
Analyst Capability	High (0.86)
Applications experience	Nominal(1)
Software Engineer Capability	High (0.86)
Virtual Machine Experience	Low (1.1)
Programming Language Experience	High (0.95)
Project Attributes :	
Application of Software Engineering Methods	High (0.91)
Use of Software Tools	High (0.91)
Required Development Schedule	High(1.04)

Effort Adjustment Factor(EAF) = $(1.15 \times 1.16 \times 1 \times 1 \times 1 \times 0.87 \times 1 \times 0.86 \times 1 \times 0.86 \times 1.1 \times 0.95 \times 0.91 \times 0.91 \times 1.04) = 0.77$

Effort= $2.4 * ((\text{KLOC})^1.05) * (\text{EAF}) = 2.4 \times (1.13)^1.05 \times 0.77 = 2.11 \sim 3 \text{ PM}$
(Person Month)