

Systems Analysis and Design

Instructor : Huang, Chuen-Min

Teamwork1 ver.1

Group 6

ID	Name
A10523006	Maggie
A10523049	Peggy
B10423003	Kurumi
B10423029	Bean
B10523020	Kendy
B10523030	Jerry
B10523053	Lynn
M10723001	Joe

Date 2018/5/9

Content

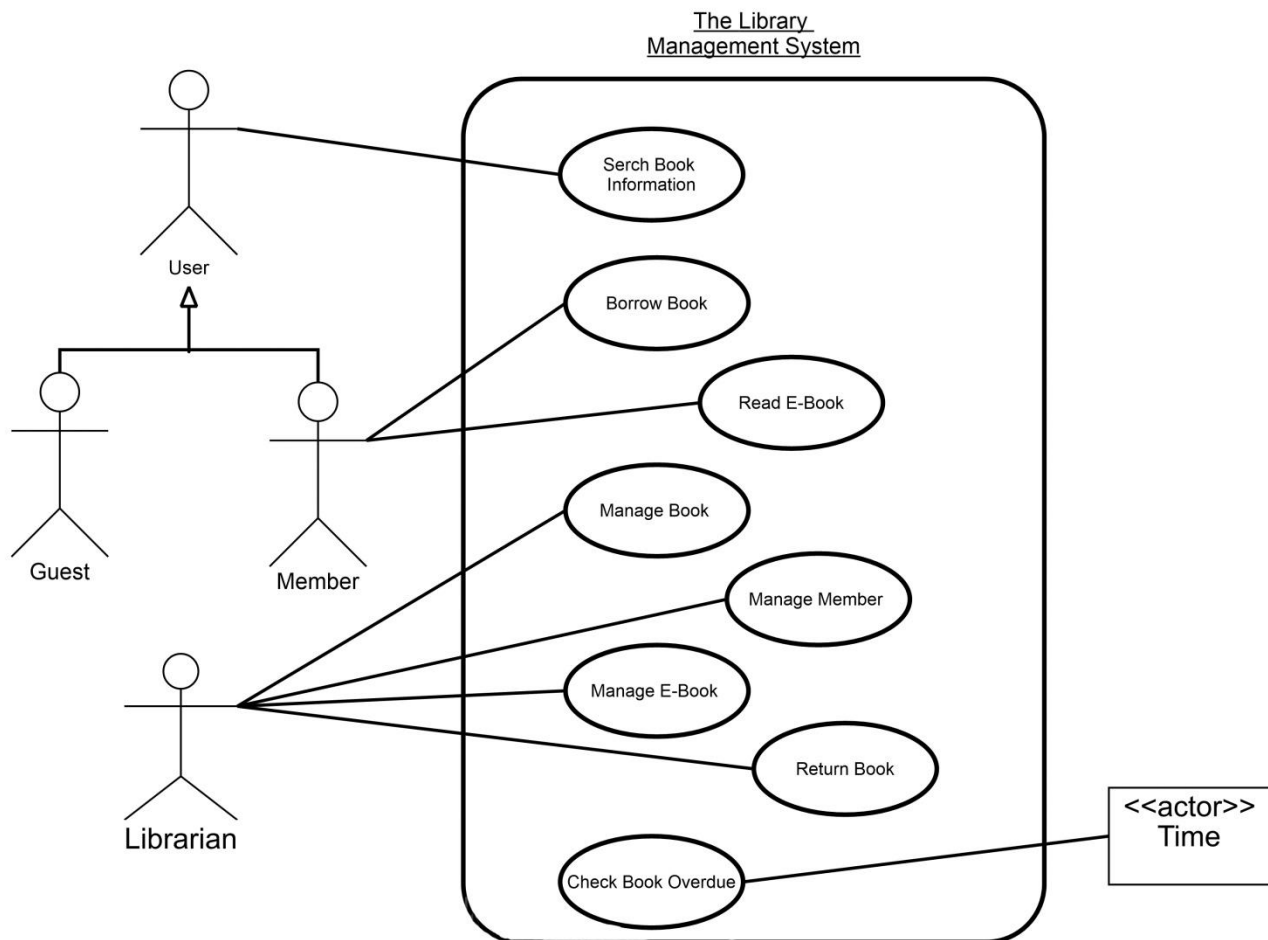
Introduction	1
Use case diagram	2
Use case description	3
1. Search Book	3
2. Borrow Book	4
3. Read E-Book	5
4. Manage Book (the most important)	6
5. Manage Member	8
6. Manage E-Book	10
7. Return Book	12
8. Check Book Overdue	13
Activity diagram	14
1. Search Book Information	14
2. Borrow Book	15
3. Read E-Book	16
4. Manage Book (the most important)	17
5. Manage Member	18
6. Manage E-Book	19
7. Return Book	20
8. Check Book Overdue	21
Sequence diagram	22
Class diagram	25
Behavior state machine	26
Participate In Assignments	27

Introduction

Our subject, The Library Management System, is mainly for the problems or functions you might meet or use when you are at the library. Regarding using the library's website, we excluded the cases outside the library. And we focus on book, ignore other objects' situations. (e.g. tablet, discussion room)

- On the basis of different user, they will get different authority.
- When they are in the building, the system will identify their identity by membership card.
- It's difficult to present the form of "a card", so we'll use "an account" to replace "a card".
- The system will identify user's identity by their account while they're using the system.
- The system can't trace back the records of books and members.

Use case diagram



- Members can have self-borrowing-service (self-serve when borrowing books) or reading E-Books.
- The only things guests can do are searching data or reading books in the library, namely only in the building, otherwise it will be refused.
- Librarian can administrate data of all the books, E-books and memberships.
- Returning books must be done through the librarian.
- Every midnight (00:00), the system checks if there are books out of date and renew the status of borrowed-out books. If books weren't returned on time, it will stop the member's borrow book right and inform members.

Use case description

1. Search Book

Use Case Name:	Search Book	ID:	1	Importance Level:	Medium
Primary Actor:	Member	Use Case Type:	Real, Detail		
Stakeholders and Interest:					
Member – someone can search book and view book.					
Guest – someone can search book and view book.					
Brief Description:					
This use case describes member and guest can search book in system. When they find any book want to know more. They can view that book.					
Trigger:	Member and guest can search book and view data about it.				
Type:	External				
Relationship:					
Association:	Member, Guest				
Include:					
Extend:					
Generalization:					
Normal Flow of Event:					
<ol style="list-style-type: none"> 1. The user login in system. <div style="margin-left: 20px;">If input wrong account or password, input it again.</div> 2. The user clicks search-book button to use search-book function. 3. The user input key word and find book he wants. <div style="margin-left: 20px;">If didn't find anything, the user need to input other key word.</div> 4. Screen will show any book searched by the user. <div style="margin-left: 20px;">If the user doesn't want to view book, stop in show book list.</div> 5. The user chooses the book he wants to view. 6. The user can view book information, <div style="margin-left: 20px;">If user wants to view more book, go to step 3.</div> <div style="margin-left: 20px;">If user don't want to view other book, end search book function.</div> 					
Sub Flow:					
Alternative / Exception Flow:					

2. Borrow Book

Use Case Name:	Borrow Book	ID:	2	Importance Level:	High
Primary Actor:	Member	Use Case Type:	Real, Detail		
Stakeholders and Interest:					
Member – The people who has been registered in the system and wants to borrow paper book.					
Brief Description:					
The member who wants to borrow paper book, needs login system to borrow book.					
Trigger:	Member wants to borrow paper book, then clicks the “Borrow Book” button.				
Type:	External				
Relationship:					
Association:	Member				
Include:					
Extend:					
Generalization:					
Normal Flow of Event:					
1. The member logins to the system. If he inputs wrong account or password, then input it again.					
2. The member clicks “Borrow Book” button and executes borrowing process.					
3. The member needs to input book id to system. If system doesn’t find the book id, then the member needs to input book id again.					
4. The system will check book’s state. If the book is unavailable, show book’s state message and go to step 3.					
5. The system will check member’s state. If member has overdue book, show member’s state message and go to step 3.					
6. If borrow book successful, show successful message.					
7. The system will change book’s state to borrowed. If the member wants to borrow other book, then go to step 3. If the member completes borrowing book, then close the borrow book window.					
Sub Flow:					
Alternative / Exception Flow:					
1. If the member didn’t overdue any book, but the member’s state message tell you have overdue some book. Go to find librarian and solve problem.					
2. If the member wants to borrow the book, but book state’s message tell this is overdue book. Please bring the book to find librarian.					

3. Read E-Book

Use Case Name:	Read Book	ID:	3	Importance Level:	Medium
Primary Actor:	Member	Use Case Type:	Real, Detail		
Stakeholders and Interest:					
Member – The people who has been registered in the system and wants to read E-book.					
Brief Description:					
The member who wants to read E-book, and needs to login system to read E-book.					
Trigger:	Member wants to read E-book, then clicks the button.				
Type:	External				
Relationship:					
Association:	Member				
Include:					
Extend:					
Generalization:					
Normal Flow of Event:					
1. The member logins to the system. If he inputs wrong account or password, then input it again.					
2. The member clicks “Read E-Book” button and executes read e-book process.					
3. The member needs to input key word. If system doesn’t find the book id, then the member needs to input key word again.					
4. The system will show book list about what member finds. If member don’t want to view it, then stop in show book list.					
5. The member chooses E-book from book list.					
6. The member view E-book information he/she chooses. If member doesn’t want to read, stop in view E-book information.					
7. The member can read E-book he chooses.					
8. The system will show book content. If member want to read more book, go to step 3, else end read E-book.					
Sub Flow:					
Alternative / Exception Flow:					

4. Manage Book (the most important)

Use Case Name:	Manage Book	ID:	4	Importance Level:	High
Primary Actor:	Librarian	Use Case Type:	Essential, Detail		
Stakeholders and Interest:					
Librarian – someone needs to create, delete and edit book’s data.					
Brief Description:					
The librarian can create new book data, edit book data, and delete book data when he needs.					
Trigger:	Create new book data, edit or delete when the librarian needs.				
Type:	External				
Relationship:					
Association:	Librarian				
Include:					
Extend:					
Generalization:					
Normal Flow of Event:					
1. The librarian logins to the system. If he inputs wrong account or password, input it again.					
2. The librarian chooses function. If the librarian wants to create a new book data, then go to S-1: create book data is performed. If the librarian wants to edit book data, then go to S-2: edit book data is performed. If the librarian wants to delete book data, then go to S-3: delete book data is performed.					
Sub Flow:					
S-1: Create book information					
1. The librarian needs to input book id.					
2. System will check book id. If the book id has been used, then the librarian needs to input book id again.					
3. The librarian inputs any book’s information it needs to be saved.					
4. The system will alarm the librarian does he/she really want to create book? If the librarian doesn’t confirm, go to step 2 of normal flow.					
5. Saving every book’s information that the librarian has input.					

S-2: Edit book information

1. The librarian needs to input book id to the system.
If system doesn't find book id, then the librarian needs to input book id again.
2. The librarian can update any book's information except book id.
3. The system will alarm the librarian does he really want to edit book?
If the librarian doesn't confirm, go to step 2 of normal flow.
4. Saving every book's information that the librarian has changed.

S-3: Delete book information

1. The librarian needs to input book id to system.
If system doesn't find book id, then the librarian needs to input book id again.
2. The system will check book's state.
If the book has been borrowed, show book's state message and go to step S-3-1.
3. The system will alarm the librarian does he really want to delete book?
If the librarian confirms, then the book will be deleted in database.
If the librarian doesn't confirm, then go to step 2 of normal flow.

Alternative / Exception Flow:

5. Manage Member

Use Case Name:	Manage Member	ID:	5	Importance Level:	High
Primary Actor:	Librarian	Use Case Type:	Essential, Detail		
Stakeholders and Interest:					
Librarian – someone needs to create, delete, and edit member’s data.					
Brief Description:					
The librarian can create new member data, edit and delete member data when he needs.					
Trigger:	Create new member data, edit or delete when the librarian needs.				
Type:	External				
Relationship:					
Association:	Librarian				
Include:					
Extend:					
Generalization:					
Normal Flow of Event:					
1. The librarian logs in to system, if he inputs wrong account or password, input it again.					
2. The librarian chooses function. If the librarian wants to create a new member data, then go to S-1: create member information is performed. If the librarian wants to edit member data, then go to S-2: edit member information is performed. If the librarian wants to delete member data, then go to S-3: delete member information is performed.					
Sub Flow:					
S-1: Create member information					
1. The librarian needs to input member id.					
2. System will check the member id. If the member id has been used, then librarian needs to input member id again.					
3. The librarian inputs any member’s information it needs to be saved.					
4. The system will alarm the librarian, does he really want to create member? If the librarian doesn’t confirm, go to step 2 of normal flow.					
5. Saving every member’s information that the librarian has input.					

S-2: Edit book information

1. The librarian needs to input member id to system.
If system doesn't find member id, then the librarian needs to input member id again.
2. The librarian can update any member information except member id.
3. The system will alarm the librarian does he really want to create member?
If the librarian doesn't confirm, go to step 2 of normal flow.
4. Saving every member's information that the librarian has input.

S-3: Delete book information

1. The librarian needs to input member id to system.
If system doesn't find member id, then the librarian needs to input member id again.
2. The system will check member's state.
If the member has overdue books, show member's state message and go to step S-3-1.
3. The system will alarm the librarian does he really want to delete member?
If the librarian confirms, then the member will be deleted in database.
If the librarian doesn't confirm, then go to step 2 of normal flow.

Alternative / Exception Flow:

6. Manage E-Book

Use Case Name:	Manage E-book	ID:	6	Importance Level:	High
Primary Actor:	Librarian	Use Case Type:	Essential, Detail		
Stakeholders and Interest:					
Librarian – someone needs to create, delete, and edit E-book’s data.					
Brief Description:					
The librarian can create new E-book data in system. Then librarian can edit or delete E-book data when he needs.					
Trigger:	Create new E-book data, edit or delete when the librarian needs.				
Type:	External				
Relationship:					
Association:	Librarian				
Include:					
Extend:					
Generalization:					
Normal Flow of Event:					
1. The librarian logs in to system, if he inputs wrong account or password, input it again.					
2. The librarian chooses function.					
If the librarian wants to create a new E-book data,					
then go to S-1: create E-book information is performed.					
If the librarian wants to edit E-book data,					
then go to S-2: edit E-book information is performed.					
If the librarian wants to delete E-book data,					
then go to S-3: delete E-book information is performed.					
Sub Flow:					
S-1: Create E-book information					
1. The librarian needs to input E-book id.					
2. System will check the E-book id.					
If the E-book id has been used, then librarian needs to input E-book id again.					
3. The librarian inputs any E-book’s information it needs to be saved.					
4. The system will alarm the librarian does he really want to create E-book?					
If the librarian doesn’t confirm, go to step 2 of normal flow.					
5. Saving every E-book’s information that the librarian has input.					

S-2: Edit E-book information

1. The librarian needs to input E-book id to system.
If system doesn't find E-book id, then the librarian needs to input E-book id again.
2. The librarian can update any E-book information except E-book id.
3. The system will alarm the librarian does he really want to create E-book?
If the librarian doesn't confirm, go to step 2 of normal flow.
4. Saving every E-book's information that the librarian has input.

S-3: Delete E-book information

1. The librarian needs to input E-book id to system.
If system doesn't find E-book id, then the librarian needs to input E-book id again.
2. The system will alarm the librarian does he really want to delete E-book?
If the librarian confirms, then the E-book will be deleted in database.
If the librarian doesn't confirm, then go to step 2 of normal flow.

Alternative / Exception Flow:

7. Return Book

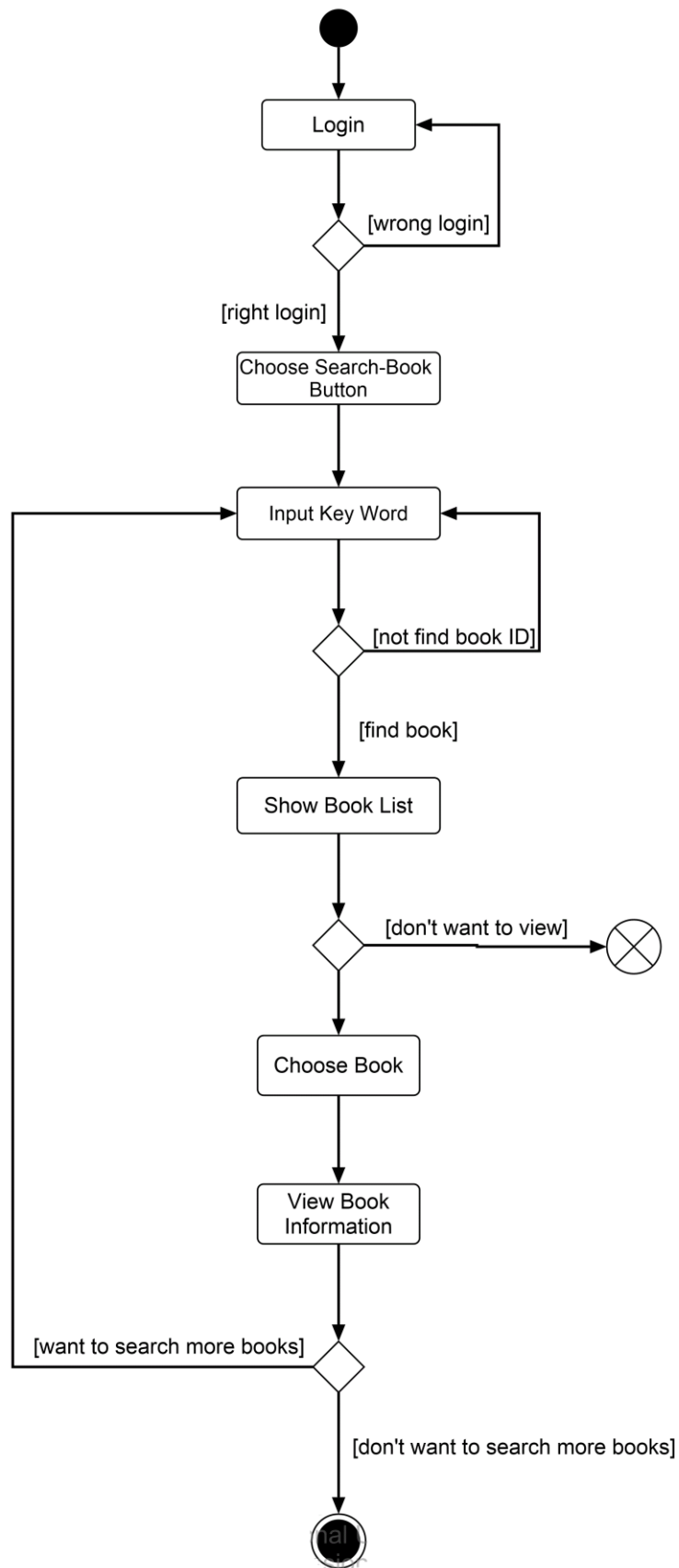
Use Case Name:	Return Book	ID:7	Importance Level:	High
Primary Actor:	Librarian	Use Case Type:	Real, Detail	
Stakeholders and Interest:				
Librarian – someone needs to handle return books.				
Brief Description:				
The use case describes when someone bring back book and the librarian should handle it.				
Trigger:	Librarian handle return book.			
Type:	External			
Relationship:				
Association:	Librarian			
Include:				
Extend:				
Generalization:				
Normal Flow of Event:				
1. The librarian login to system. If he inputs wrong account or password, input it again.				
2. The librarian clicks “return book” button to use return-book function.				
3. The librarian needs to input book id to system.				
If system doesn’t find book id, then the librarian needs to input book id again.				
4. The system will check book’s state.				
If the book is overdue, reduce member’s overdue book.				
If the book is unavailable, show book’s state message and go to step 3.				
5. The system will change book’s borrower, book time and overdue time.				
Sub Flow:				
Alternative / Exception Flow:				

8. Check Book Overdue

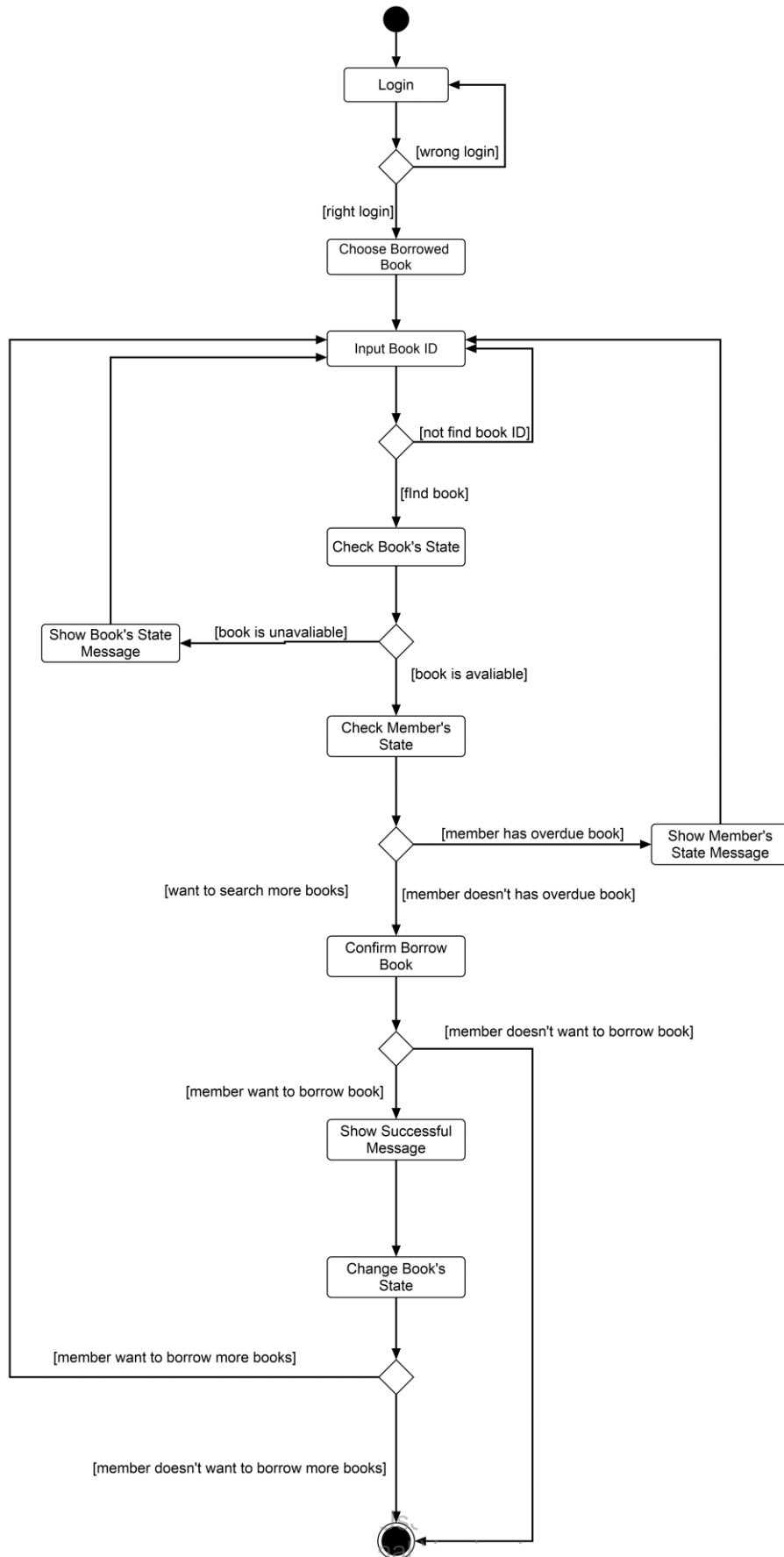
Use Case Name:	Check overdue time	ID:	8	Importance Level:	High
Primary Actor:	Time	Use Case Type:	Essential, Detail		
Stakeholders and Interest:					
Time – When midnight (00:00) comes, will check every borrowed book is overdue or not.					
Brief Description:					
This function will work on midnight (00:00) every day. It will check every borrowed book is overdue or not. If it is overdue, it will change member’s state and book’s state to overdue.					
Trigger:	midnight (00:00)				
Type:	External				
Relationship:					
Association:	Time				
Include:					
Extend:					
Generalization:					
Normal Flow of Event:					
1. The time will check every borrowed book’s state. If the book is overdue, the S-1: change book state is performed.					
Sub Flow:					
S-1: Change Book State					
1. The book’s state will change to overdue.					
2. The member who borrowed this book will plus 1 overdue book in state.					
Alternative / Exception Flow:					

Activity diagram

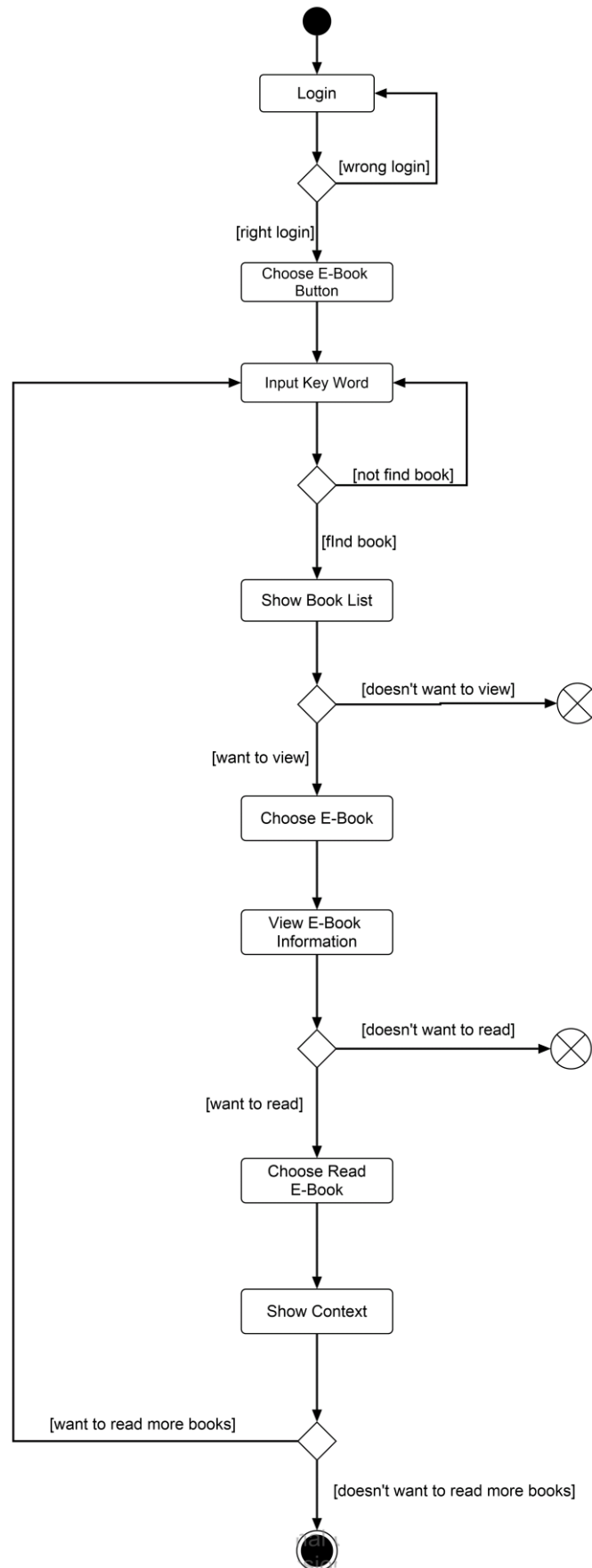
1. Search Book Information



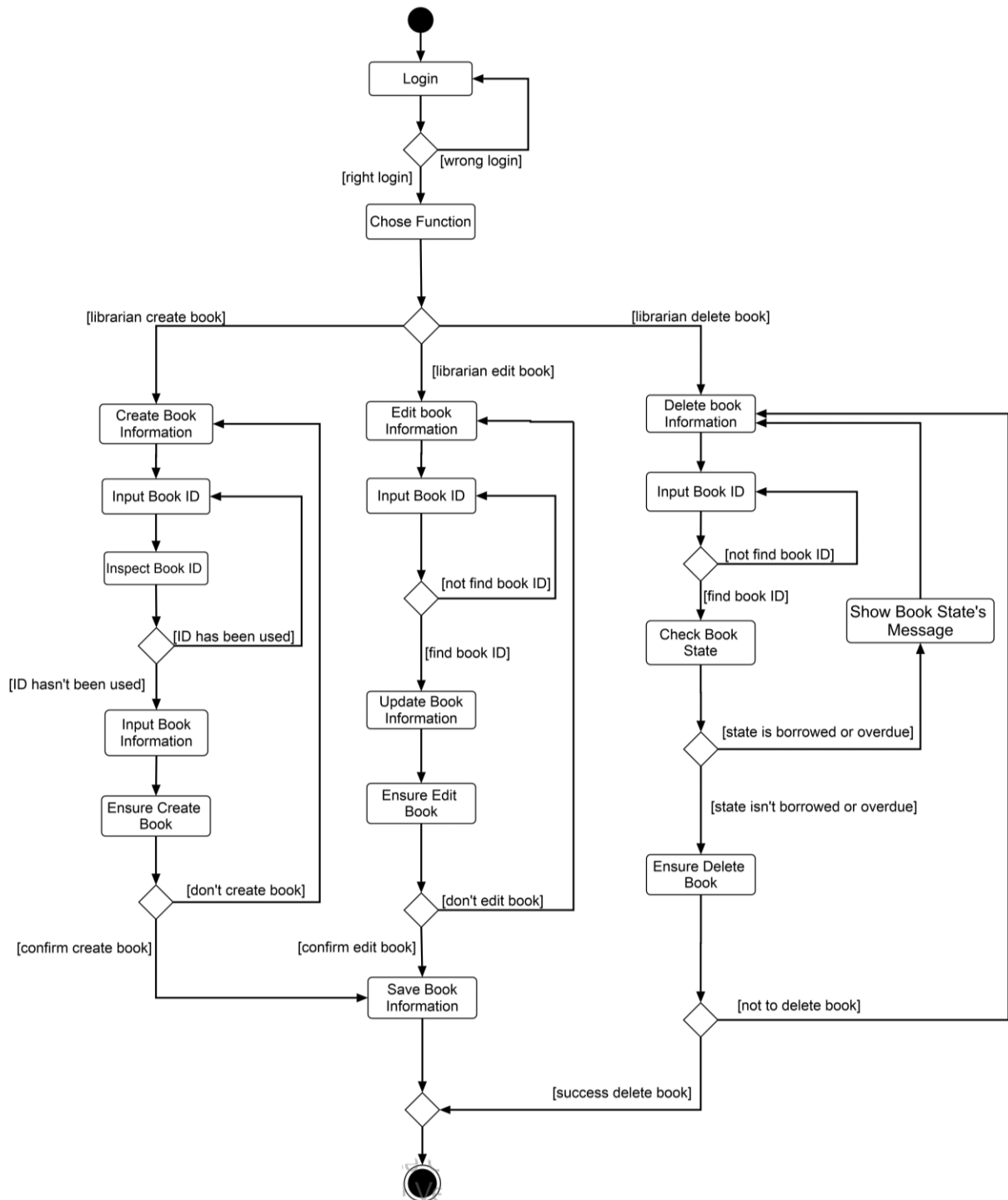
2. Borrow Book



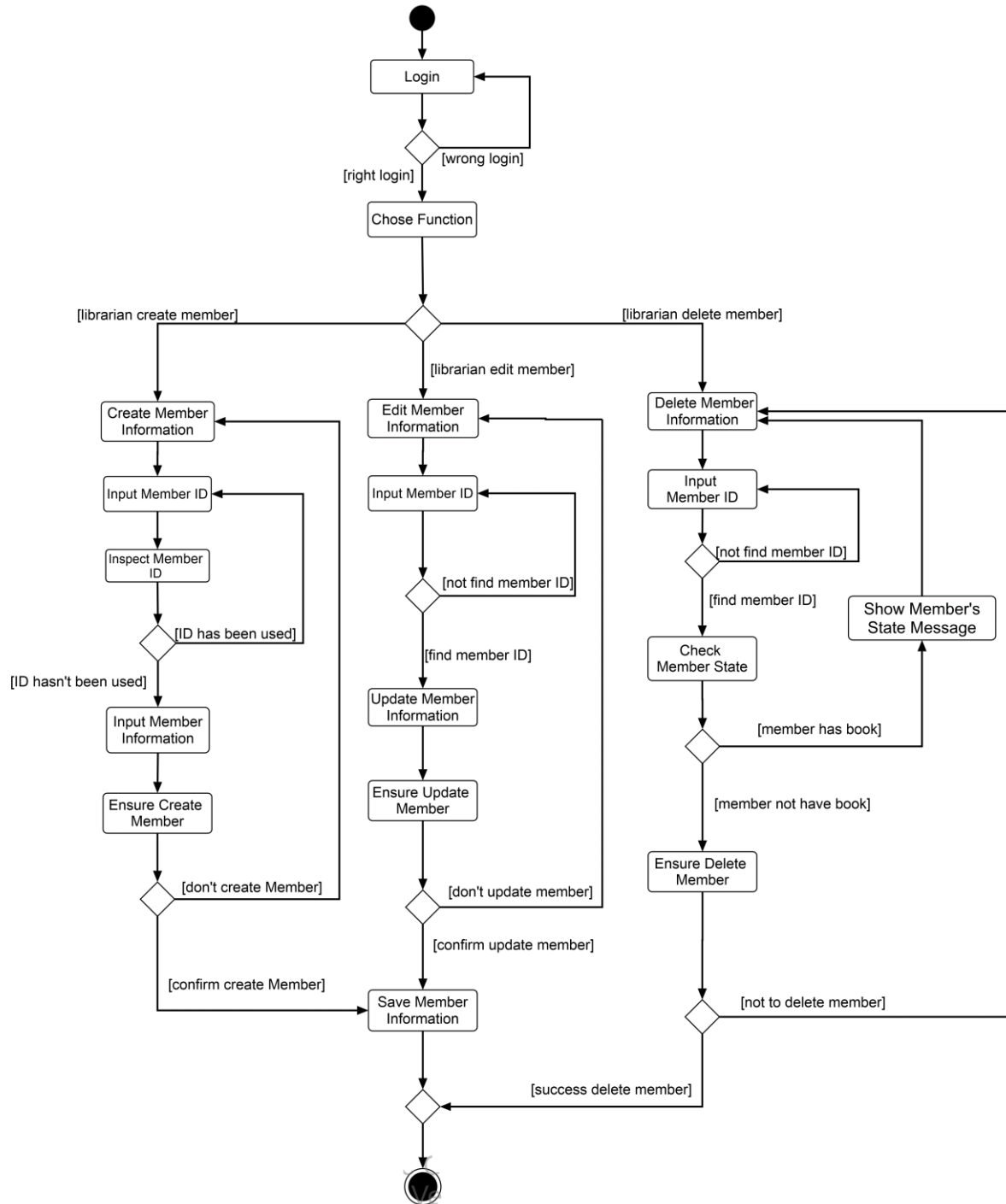
3. Read E-Book



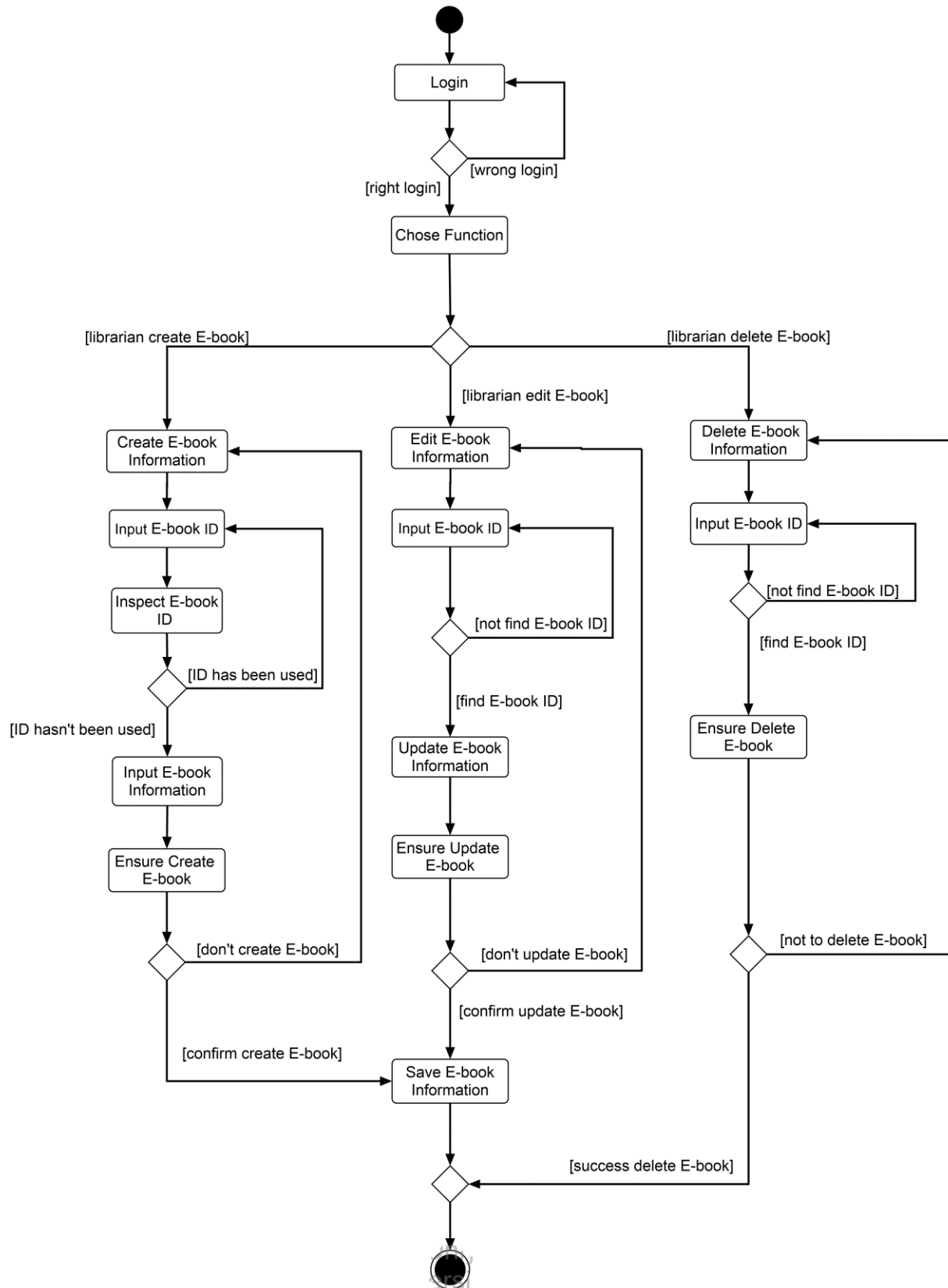
4. Manage Book (the most important)



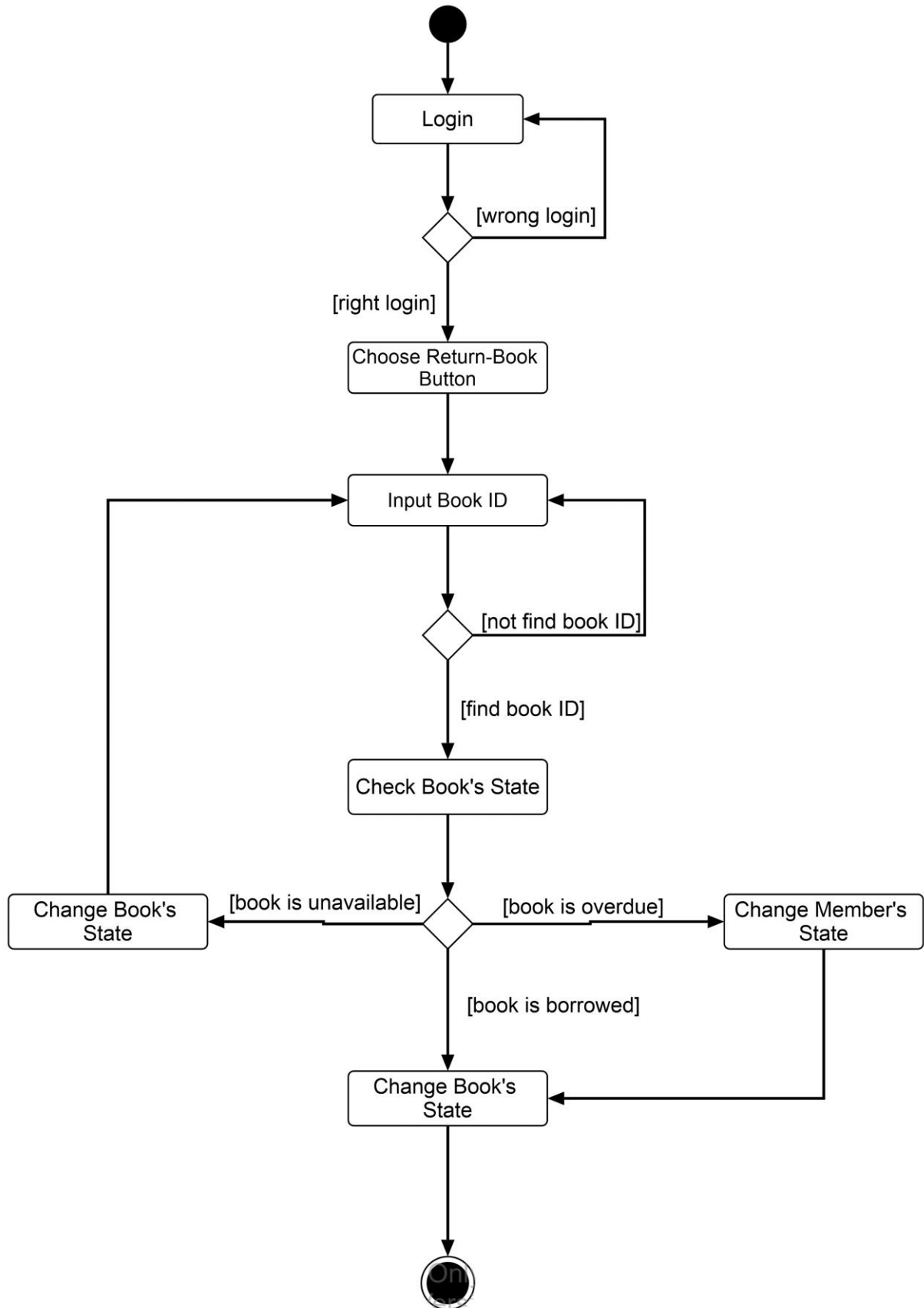
5. Manage Member



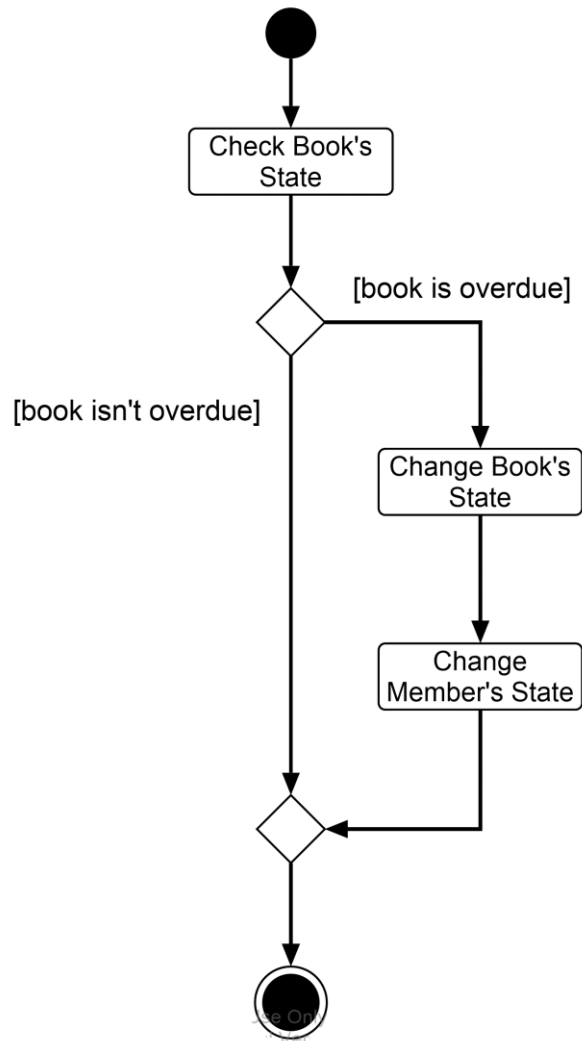
6. Manage E-Book



7. Return Book



8. Check Book Overdue



Sequence diagram

To use Manage Book's function, the Librarian needs to input Account and Password to login the system to verify identity if the user has access to use this function. If identity verify fail, the system will show log-in fail message. If identity verify success, the system shows the librarian GUI for librarian.

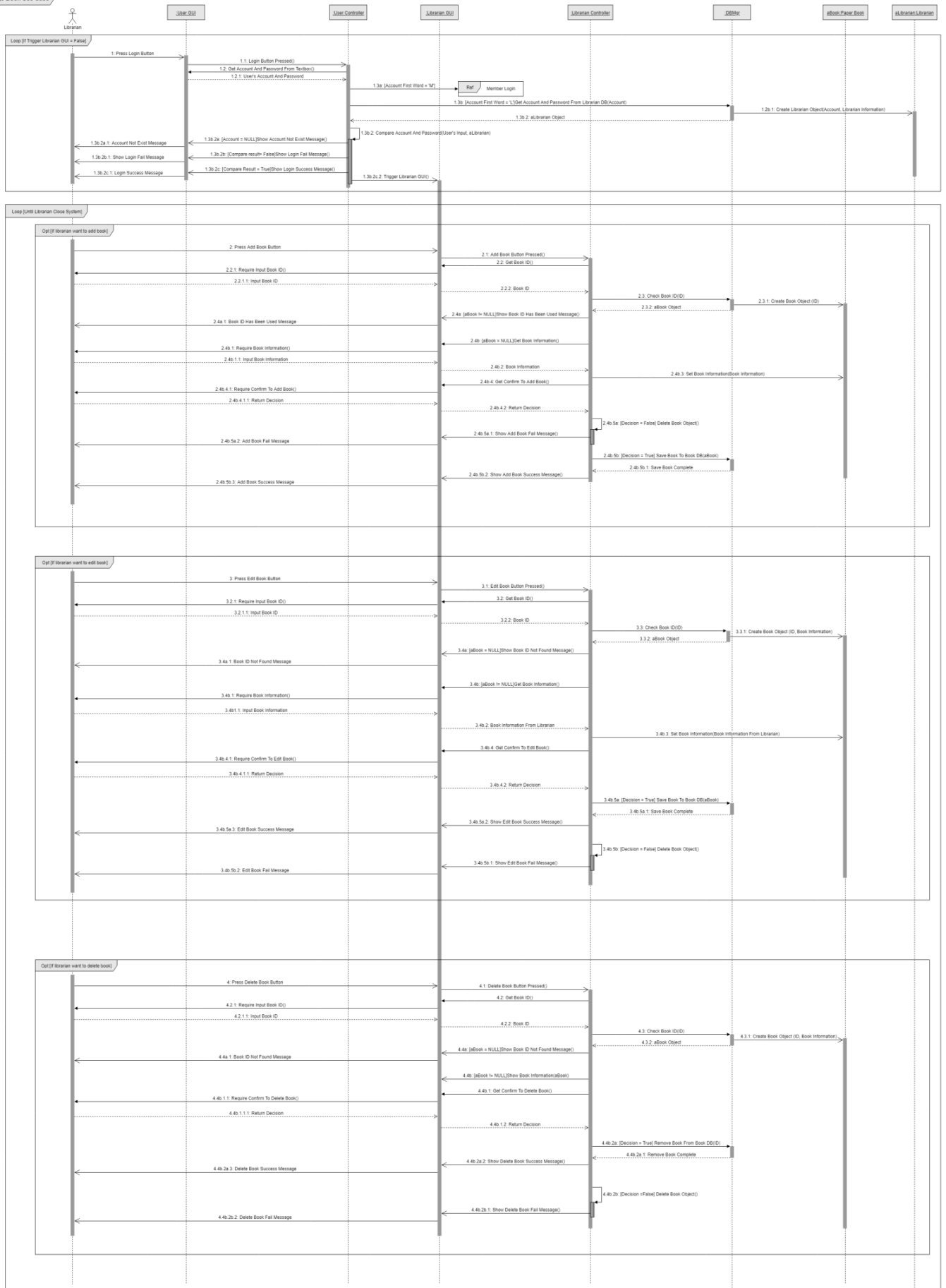
In the librarian GUI, the librarian can press "Add Book", "Edit Book" and "Delete Book" button to choose function he wants to use.

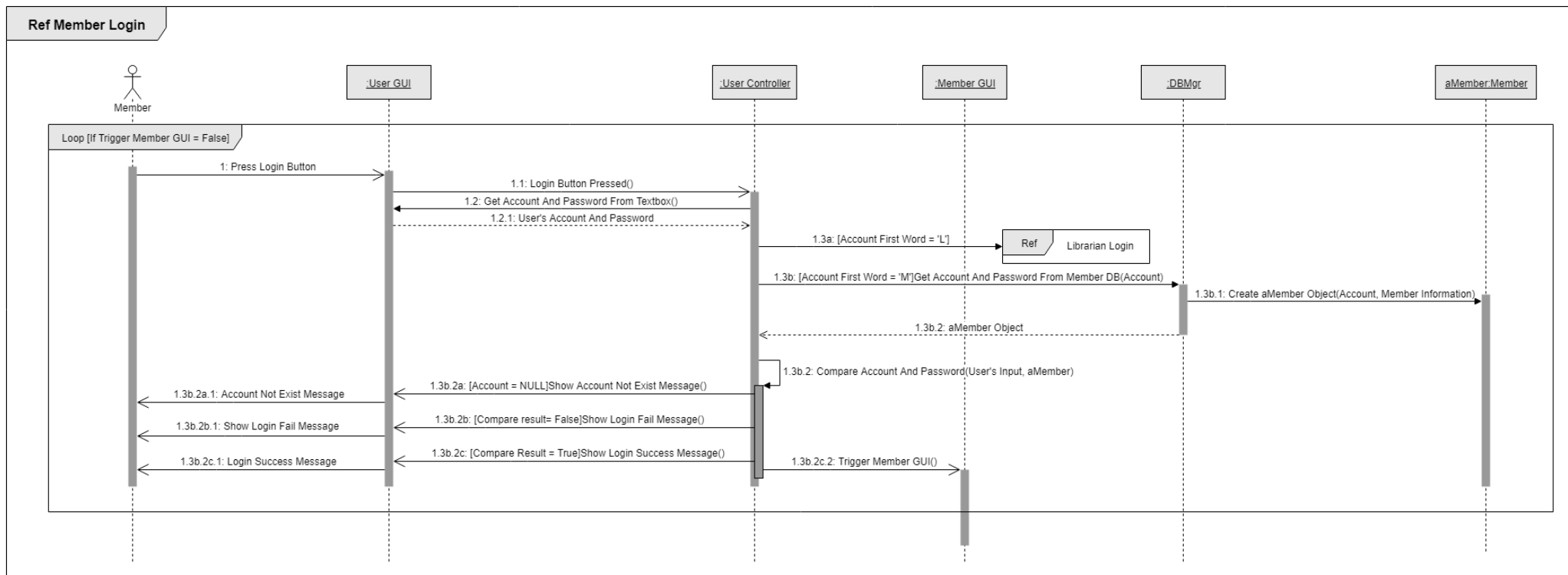
When the librarian presses "Add Book" button, system will show a form to the librarian to input the information about book which he wants to add, the system will check two things before execute add-book process. First, system will notice the librarian confirm to do this book changed, if the librarian doesn't sure to do this, system will cancel add-book form and wait for next action. Second, system will check whether the book's ID has been used, if the book ID has been used, system will show "the Book ID has been used" message. If the book ID hasn't been used, system will execute add-book process then show the "add book success" message when add-book process completed.

When the librarian presses "Edit Book" button, system will show a form to the librarian to input the information about book which he wants to edit, the system will check two things before executing edit-book process. First, system will notice the librarian confirm to do this book changed, if the librarian isn't sure to do this, system will cancel edit-book form and wait for next action. Second, system will check whether the book's ID is existed, If the book ID isn't existed, system will show "book ID is not found" message. If the book ID is existed, system will execute edit-book process then show "edit book success" message when edit-book process completed.

When the librarian press "Delete Book" button, system will show a form to the librarian to input the book's ID which he wants to delete, the system will check three things before executing delete-book process. First, system will notice the librarian confirm to do this, if the librarian isn't sure to do this, system will cancel edit-book form and wait for next action. Second, system will check whether the book's ID is existed. If the book ID isn't existed, system will show "book ID is not found" message. Third, system will check whether the book's state is borrowed or overdue. If the book's state is borrowed, system will show "the book is borrowed" message means the librarian can't delete the book's information before member return the book. If the book's state is overdue, system will show "book is overdue" message means the librarian can't delete book before member return overdue book. If the book's state is neither borrowed nor overdue, system will execute delete-book process then show "delete book success" message when delete-book process completed.

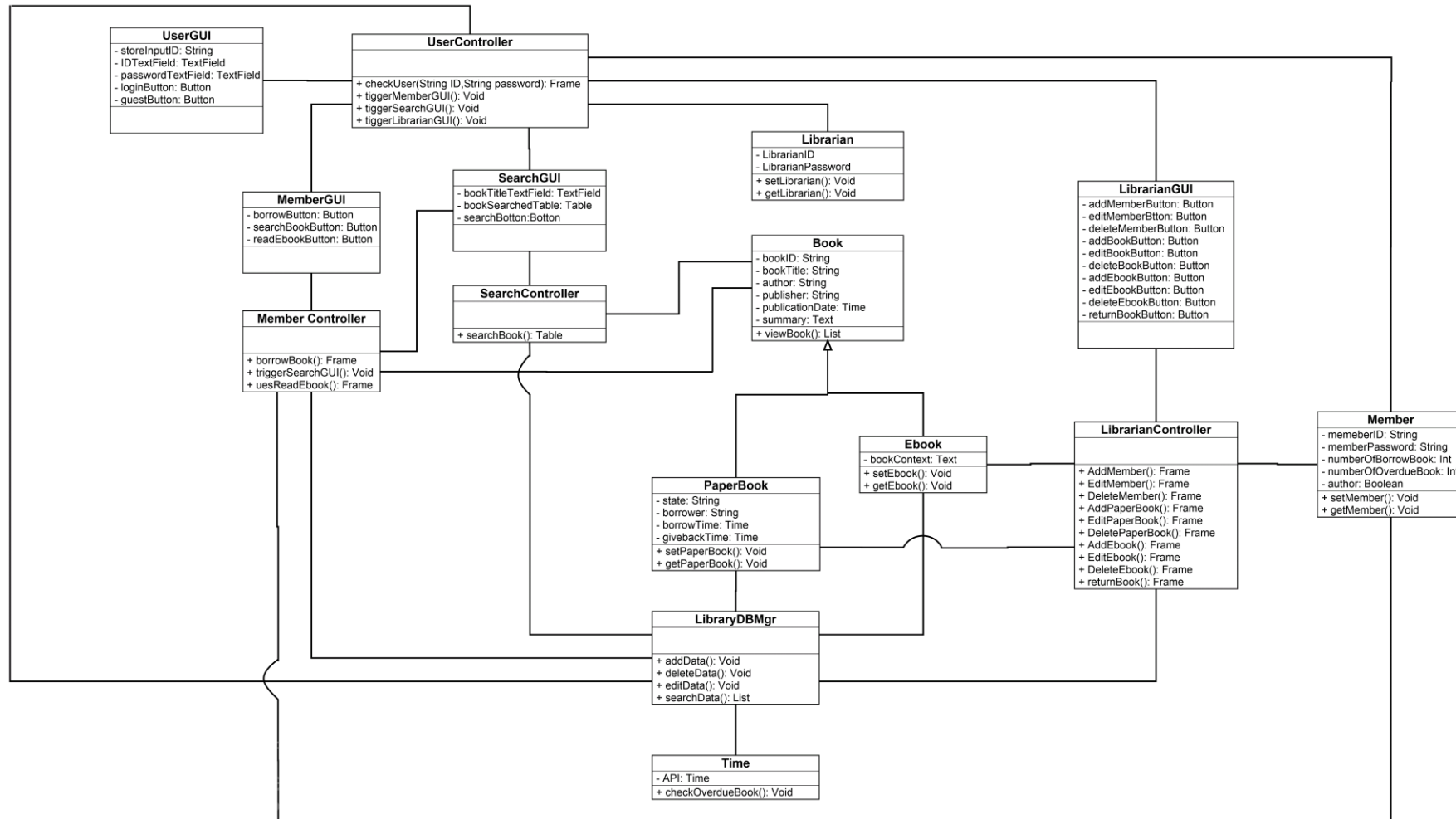
ed Manage Book Use Case





Class diagram

First, system run class Time to check whether time is at midnight (00:00). If it is at midnight (00:00), system will run checkOverdueBook() method to check overdue book, and the system will show class UserGUI let user can login and according to his login to decide which GUI user can use.

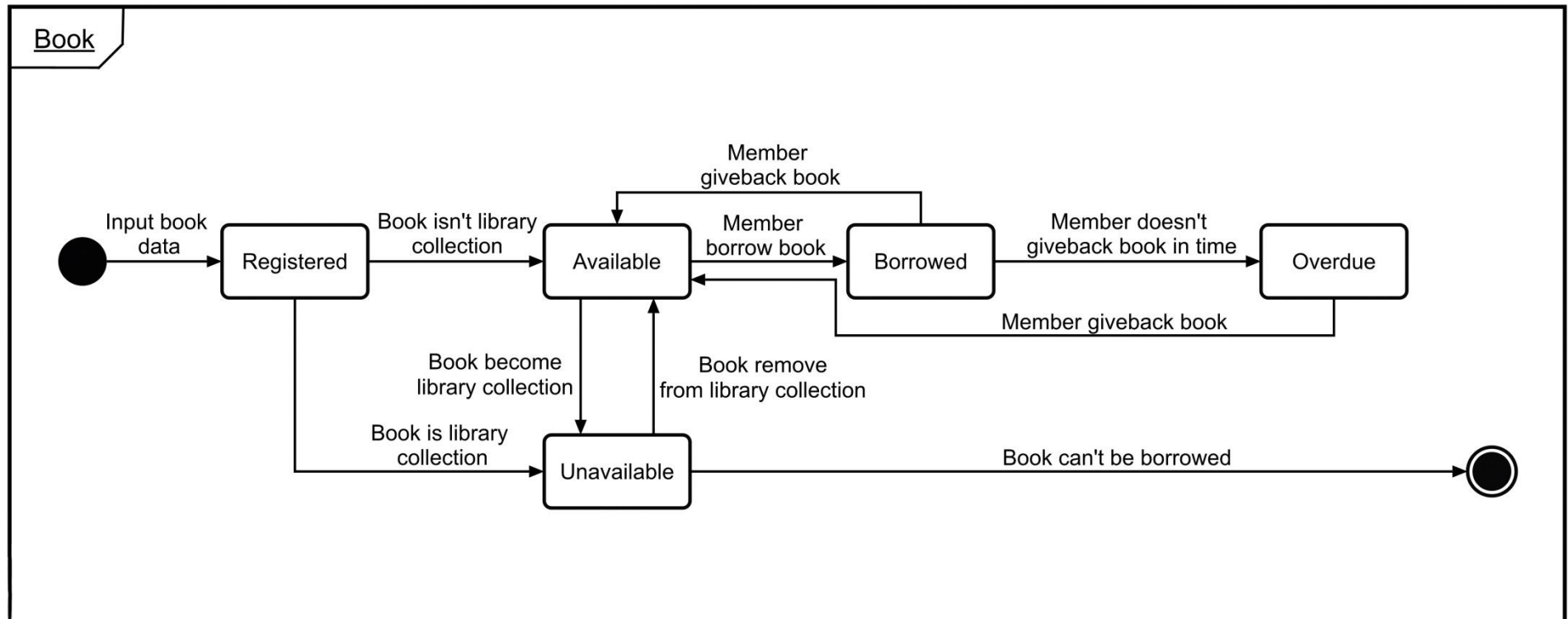


Behavior state machine

When the book is input the information to the system, the book's state changes to registered.

If book is collection, the book's status will change to unavailable. If book isn't collection, the book's status will change to available and the member can borrow book from available status.

If member doesn't giveback book in time, the book status will change to overdue until the member giveback book. When member givebacks book, the book's status will change to available and it can be borrowed.



Participate In Assignments

ID	Name	Participate	Responsibility
A10523006	Maggie	100%	Use case diagram Activity diagram Class diagram
A10523049	Peggy	100%	Use case diagram Use case description Activity diagram Class diagram
B10423003	Kurumi	100%	Word Introduction Use case diagram Use case description Activity diagram Class diagram Sequence diagram Behavior state machine
B10423029	Bean	0%	
B10523020	Kendy	100%	Use case diagram Use case description Activity diagram Class diagram Sequence diagram Behavior state machine
B10523030	Jerry	100%	PPT Use case diagram Activity diagram Use case description Behavior state machine Class diagram Behavior state machine
B10523053	Lynn	100%	Use case diagram Use case description Activity diagram Class diagram Sequence diagram Behavior state machine
M10723001	Joe	100%	Use case diagram Use case description Activity diagram Class diagram Sequence diagram Behavior state machine