Systems Analysis

and Design

Instructor : Huang, Chuen-Min

**Teamwork1 ver.2**

Group 6

|  |  |
| --- | --- |
| ID | Name |
| A10523006 | Maggie |
| A10523049 | Peggy |
| B10423003 | Kurumi |
| B10423029 | Bean |
| B10523020 | Kendy |
| B10523030 | Jerry |
| B10523053 | Lynn |
| M10723001 | Joe |
|  |  |
|  |  |
| Date 2018/5/21 | |

Content

[**Introduction** 1](#_Toc514537050)

[**Use case diagram** 2](#_Toc514537051)

[**Use case description** 3](#_Toc514537052)

[1. Search Book Information 3](#_Toc514537053)

[2. Read E-Book 4](#_Toc514537054)

[3. Borrow Book 5](#_Toc514537055)

[4. Return Book 6](#_Toc514537056)

[5. Manage Paper Book (the most important) 7](#_Toc514537057)

[6. Manage Member 10](#_Toc514537058)

[7. Manage E-Book 12](#_Toc514537059)

[**Activity diagram** 14](#_Toc514537060)

[1. Search Book Information 14](#_Toc514537061)

[2. Read E-Book 15](#_Toc514537062)

[3. Borrow Book 16](#_Toc514537063)

[4. Return Book 17](#_Toc514537064)

[5. Manage Paper Book (the most important) 18](#_Toc514537065)

[6. Manage Member 19](#_Toc514537066)

[7. Manage E-Book 20](#_Toc514537067)

[**Sequence diagram** 21](#_Toc514537068)

[**Class diagram** 25](#_Toc514537069)

[**Behavior state machine** 26](#_Toc514537070)

[**Participate In Assignments** 27](#_Toc514537071)

**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. If you are not in the library, you still can use website to search book information. 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 users are in the library, the system will identify their identity by membership card;  
  when users are not in the library, the system will identify their identity by account and password.
* 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 book information or reading books in the library.
* Librarian can administrate data of all the books, E-books and memberships.
* Returning books must be done through the librarian.

**Use case description**

1. Search Book Information

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Use Case Name:** | Search Book Information | **ID:** | 1 | **Importance Level:** | | Medium |
| **Primary Actor:** | User | **Use Case Type:** | | | Essential, Detail | |
| **Stakeholders and Interest:** | | | | | | |
| User – 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:** | User | | | | | |
| **Include:** |  | | | | | |
| **Extend:** |  | | | | | |
| **Generalization:** |  | | | | | |
| **Normal Flow of Event:** | | | | | | |
| 1. The user login in system.   If input wrong account or password, input it again.  If the user is guest, he/she needs to choose “Guest” button.   1. The user clicks search-book button to use search-book function. 2. The user input key word and find book he wants.   If didn’t find anything, the user need to input other key word.   1. Screen will show any book searched by the user.   If the user doesn’t want to view book, stop in show book list.   1. The user chooses the book he wants to view. 2. The user can view book summary,   If user wants to view more book, go to step 3.  If user don’t want to view other book, end search book function. | | | | | | |
| **Sub Flow:** | | | | | | |
| S-1: | | | | | | |
| **Alternative / Exception Flow:** | | | | | | |
|  | | | | | | |

1. Read E-Book

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Use Case Name:** | Read E-Book | **ID:** | 2 | | **Importance Level:** | | Medium |
| **Primary Actor:** | Member | | | **Use Case Type:** | | Essential, 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.   1. The member clicks “Read E-Book” button and executes read e-book process. 2. The member needs to input key word.   If system doesn’t find the book id, then the member needs to input key word again.   1. The system will show book list about what member finds.   If member don’t want to view it, then stop in show book list.   1. The member chooses E-book from book list. 2. The member view E-book information he/she chooses.   If member doesn’t want to read, stop in view E-book information.   1. The member can read E-book he chooses. 2. 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:** | | | | | | | |
|  | | | | | | | |

1. Borrow Book

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Use Case Name:** | Borrow Book | **ID:** | 3 | **Importance Level:** | | High |
| **Primary Actor:** | Member | **Use Case Type:** | | | Essential, 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, Librarian | | | | | |
| **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.   1. The member clicks “Borrow Book” button and executes borrowing process. 2. 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.   1. The system will check book’s state.   If the book is unavailable, show book’s state message and go to step 3.   1. The system will check member’s state.   If member has overdue book, show member’s state message and end the function.   1. If borrow book successful, show successful message. 2. 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:** | | | | | | |
| S-1 | | | | | | |
| **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. | | | | | | |

1. Return Book

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Use Case Name:** | Return Book | **ID:** | 4 | **Importance Level:** | | High |
| **Primary Actor:** | Librarian | **Use Case Type:** | | | Essential, 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:** | Member, 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.   1. The system will check book’s state.   If the book is overdue, reduce member’s number of overdue book.  If member number of overdue book is zero then change member state.  If the book is unavailable, show book’s state message and go to step 3.   1. The system will change book’s borrower, book time and overdue time. | | | | | | |
| **Sub Flow:** | | | | | | |
|  | | | | | | |
| **Alternative / Exception Flow:** | | | | | | |
|  | | | | | | |

1. Manage Paper Book (the most important)

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Use Case Name:** | Manage Paper Book | **ID:** | 4 | | **Importance Level:** | | High |
| **Primary Actor:** | Librarian | | | **Use Case Type:** | | Real, Detail | |
| **Stakeholders and Interest:** | | | | | | | |
| Librarian – someone needs to manage book in the system. | | | | | | | |
| **Brief Description:** | | | | | | | |
| This use case describes when librarian get new book, how librarian add new book data in database. If book’s information is wrong or need to be change, librarian can edit exist book information or change state about book. For example, book is missing. Book’s state will change to missing, such as damaged, repaired, and deregistered. | | | | | | | |
| **Trigger:** | Librarian get new book and add new book information or edit exist book data or change book state. | | | | | | |
| **Type:** | External | | | | | | |
| **Relationship:** | | | | | | | |
| **Association:** | Librarian | | | | | | |
| **Include:** |  | | | | | | |
| **Extend:** |  | | | | | | |
| **Generalization:** |  | | | | | | |
| **Normal Flow of Event:** | | | | | | | |
| 1. The librarian logins into the system.   If librarian inputs wrong account or password, input it again.   1. The librarian chooses function.   If the librarian wants to add 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 update book state,  then go to S-3: update book state is performed. | | | | | | | |
| **Sub Flow:** | | | | | | | |
| S-1: Add book information   1. The librarian presses add book button. 2. The system will create new book id. 3. The system will show add book form to librarian. 4. The librarian can input book information on add book form.   If librarian does not want add book, then press cancel button and go to step 2.   1. The librarian press save button after input book information. 2. The system will alarm librarian does he really want to add this book?   If librarian presses “No”, go to step S-1-3.   1. The system will save data which from add book form to database.   S-2: Edit book information   1. The librarian press edit book button. 2. The librarian input book ID he want to edit.   If system doesn’t find book, then show message tell the librarian didn’t find books, go to step 2.   1. The system will show edit book form for librarian. 2. The librarian can change book information.   If librarian does not wants to edit book, then press cancel button and go to step 2.   1. The librarian press save button after input book information 2. The system will alarm librarian did he really want to change this book’s information?   If librarian presses ”No”, then go to step S-2-3   1. The system will save data from edit book form to database.   S-3: Update book state   1. The librarian press update book state button. 2. The librarian input book ID he want to update.   If system doesn’t find book, then show message tell librarian didn’t find book, go to step 2.   1. The system will show book state choice form for librarian.    1. If book is available, then librarian chooses available state.       1. The system will alarm librarian does he really want to change this book’s state?   If librarian presses “No”, go to step S-3.   * + 1. The system will change book’s state to available.   1. If book is unavailable, then librarian chooses unavailable state.   3-2-1. The system will alarm librarian did he really want to change this book’s state?  If librarian presses “No”, go to step S-3.  3-2-2. The system will change book’s state to unavailable.   * 1. If book is damaged, then librarian chooses damage state.      1. The system will show a manage damage book form.      2. If book is damaged by member, the librarian need input member ID.      3. If the librarian press cancel button, go to step S-3.   If the librarian press ok button, the system will change book’s state to damage. And if member ID isn’t null, system will change member’s state   * 1. If book is repaired, then librarian chooses repaired state.      1. The system will alarm librarian did he really want to change this book’s state?   If librarian press ”No”, go to step S-3.   * + 1. The system will change book’s state to be repaired.   1. If book is missing, then librarian chooses missing state.  1. The system will alarm librarian did he really want to change this book’s state?   If librarian press “No”, go to step S-3.   1. The system will change book’s state to missing.    1. If book is deregistered, then librarian chooses deregistered state.       1. The system will alarm librarian did he really want to change this book’s state?   If librarian presses ‘No”, go to step S-3.   * + 1. The system will change book’s state to deregistered. | | | | | | | |
| **Alternative / Exception Flow:** | | | | | | | |
|  | | | | | | | |

1. Manage Member

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Use Case Name:** | Manage Member | **ID:** | 6 | **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:** | If the guest wants to join to member, and create new member data. If the member wants to change the password, member needs to edit the data. If the member doesn’t want to use the system anymore so I asked to librarian to delete the data. | | | | | |
| **Type:** | External | | | | | |
| **Relationship:** | | | | | | |
| **Association:** | Librarian | | | | | |
| **Include:** |  | | | | | |
| **Extend:** |  | | | | | |
| **Generalization:** |  | | | | | |
| **Normal Flow of Event:** | | | | | | |
| 1. The librarian logins to system,   if he inputs wrong account or password, input it again.   1. 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.   1. The librarian inputs any member’s information it needs to be saved. 2. 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.   1. Saving every member’s information that the librarian has input.   S-2: Edit member 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.   1. The librarian can update any member information except member id. 2. 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.   1. Saving every member’s information that the librarian has input.   S-3: Delete member 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.   1. 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.   1. 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:** | | | | | | |
|  | | | | | | |

1. Manage E-Book

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Use Case Name:** | Manage E-book | **ID:** | 7 | **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 logins 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: 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.   1. The librarian inputs any E-book’s information it needs to be saved. 2. 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.   1. Saving every E-book’s information that the librarian has input.   S-2: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.   1. The librarian can update any E-book information except E-book id. 2. 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.   1. Saving every E-book’s information that the librarian has input.   S-3: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.   1. 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:** | | | | | | |
|  | | | | | | |

**Activity diagram**

1. Search Book Information

|  |
| --- |
|  |

1. Read E-Book

|  |
| --- |
|  |

1. Borrow Book

|  |
| --- |
|  |

1. Return Book

|  |
| --- |
|  |

1. Manage Paper Book (the most important)

|  |
| --- |
|  |

1. Manage Member

|  |
| --- |
|  |

1. Manage E-Book

|  |
| --- |
|  |

**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.





**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 library gets new books, the book will be adding to collection and the state will start at “available”. Member can borrow book and the book state will change to “borrowed”. If the member return book in time, state will change back to “available”. If not, the state will change to “Overdue” until member return book.

The member may lose the book. He must inform the librarian that the book is missing, then the state will change to “missing”. If it can’t be find then the state will go to the end, but if it is found then the state will go back return to “available”.

When the member returns the book and it is damaged, the state will change to “damaged” and it will go to “being repaired” state. If the book is too damaged and can’t be repaired, the state will go to the end.

The book will change to “available” or “unavailable” if the librarian needs.

|  |
| --- |
|  |

**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 |