

Use cases for a simple library system

Add Item

Use case name: Add Item

Participating actor: Librarian

Parameters: in Isbn: Integer, out ItemID: Integer

Entry condition:

1. The *librarian* requests the *title* information from the *librarianterminal*. The Find Title use case is used. If the *title* doesn't exist, the system will ask the *librarian* to add the *title*, the Add Title use case is used.

Flow of events:

2. The *title* information is displayed on the *librarianterminal*.

Exit condition:

3. The *librarian* adds *item*, an *itemId* is generated for it.

Add User

Use case name: Add User

Participating actor: Librarian

Parameters: in UserID: Integer

Entry condition:

1. The *librarian* requests to add *user* from the *librarianterminal*.

Flow of events:

2. The *librarian* inputs the user information to the system. If the user information has already been recorded in the system, the *librarian* will not be allowed to add the *user*.

Exit condition:

3. The user information has been stored in the system, a *userId* is generated for it or the *librarian* is not allowed to add the *user*.

Add Title

Use case name: Add Title

Participating actor: Librarian

Parameters: in Isbn: Integer

Entry condition:

1. The *librarian* requests to add *title* from the *librarianterminal*.

Flow of events:

2. The *librarian* inputs the title information such as *isbn* to the system. If any information requested is missing or the same with other *title*, the system will prompt an error message and go to the the beginning of this step; if the *title*'s information is same with other *title*, the *librarian* will not be allowed to add the *title* and go to 3.

Exit condition:

3. A new *title* has been added to the system or the *librarian* is not allowed to add the *title*.

Borrow Loancopy

Use case name: Borrow Loancopy

Participating actor: Librarian

Parameters: in UserID: Integer, in ItemID: Integer

Entry condition:

1. The *librarian* requests the *user* information from the *librarianterminal*. The Find User use case is used. If the *user* doesn't exist, then proceeds to 5.

Flow of events:

2. The *librarian* requests the *title* information. The Find Title use case is used. If the title doesn't exist, then proceeds to 5.

3. The *librarian* requests add *loan*. The system judges if the *user* is able to borrow book. If the *user's* privilege is revoked, the system will prompt a corresponding message. The *librarian* can collect the fine. The Collect Fine use case is used. If the *user's* fine is not cleared, proceeds to 5. If the *item* is a *referencecopy*, or, the *user* already has 10 *loans*, then the *user* is not allowed to borrow *loancopy* and proceeds to 5. The system also checks the book reservation. The Check Reservation use case is used. If the system denies the borrow request, then proceeds to 5.

4. The *library* lends the *loancopy* to the *user*. If the user has reached loan limit, the system updates the *user's* loan privilege. If the *user* has a *reservation* on that *title*, the *reservation* is removed. The Remove Reservation use case is used.

Exit conditions:

5. The system denies the borrow request or a new *loan* is registered, a loanId is generated for the *loan*.

Collect Fine

Use case name: Collect Fine

Participating actor: Librarian

Parameters: in UserID: Integer, in ItemID: Integer

Entry condition:

1. The *librarian* requests the *user* information from the *librarianterminal*. The Find User use case is used. If the *user* doesn't exist, then proceeds to 3.

2. The *librarian* enters the fine collected, the system updates the *user's* fine information. If the *user's* privilege is revoked, then system updates the *user's* privilege.

Exit condition:

3. The *user's* fine is cleared or the *librarian* is unable to collect the *user's* fine.

Remove Item

Use case name: Remove Item

Participating actor: Librarian

Parameters: in ItemID: Integer

Entry condition:

1. The *librarian* requests the *title* information from the *librarianterminal*. The Find Title use case is used. If the *title* doesn't exist, then proceeds to 4.

Flow of events:

2. The *title* information is displayed on the *librarianterminal*.

3. The *librarian* specifies the *item* should be removed. If the *item* is loaned, the *librarian* is not allowed to remove the *item*.

Exit condition:

4. The specified *item* has been removed or the *librarian* is not allowed to remove the *item*.

Remove Title

Use case name: Remove Title

Participating actor: Librarian

Parameters: in Isbn: Integer

Entry condition:

1. The *librarian* requests the *title* information from the *librarianterminal*. The Find Title use case is used. If the title doesn't exist, then proceeds to 3.

Flow of events:

2. The title information is displayed on the *librarianterminal*. If the *title* is reserved or some of the title's *loancopies* are loaned, the *librarian* is not allowed to remove the *title*.

Exit condition:

3. The *title* has been removed, along with all the *items* associated with the *title* or the *librarian* is not allowed to remove the *title*.

Remove User

Use case name: Remove User

Participating actor: Librarian

Parameters: in UserID: Integer

Entry condition:

1. The *librarian* requests the *user* information from the *librarianterminal*. The Find User use case is used. If the user doesn't exist or the user's privilege is revoked, proceeds to 3.

Flow of events:

2. The user information is displayed on the *librarianterminal*. If the *user* has *loan*, the *librarian* is not allowed to remove the *user*.

Exit condition:

3. The *user* has been removed along with his/her reservation if any or the *librarian* is not allowed to remove the *user*.

Renew Loan

Use Case Name: Renew Loan

Participating actor: Librarian

Parameters: in UserID: Integer, in ItemID: Integer

Entry condition:

1. The *librarian* requests the *loan* information from the *librarianterminal*. The Find Loan use case is used. If the loan doesn't exist, or the loan owner's privilege is revoked, proceeds to 5.

Flow of events:

2. The *loan* information is displayed on the *librarianterminal*.

3. The *librarian* requests renew *loan*. If the *user's* privilege is revoked, the system will prompt a corresponding message. The *librarian* can collect the fine. If the *user's* fine is not cleared, then proceeds to 5. The system checks whether the *title* is reserved. The Check Reservation use case is used. If the *title* is reserved, the *librarian* is not allowed to renew the *loan* and proceeds to 5.

4. The system identifies the *loan* of the *loancopy*. If the *loan* is overdue, the system prompts loan status information. The system checks the times of the renewing the *loan*, if the *loan* has been renewed 2 times, the *librarian* is not allowed to renew the *loan*.

Exit condition:

5. The corresponding *loan* is renewed, a new dueDate is generated for the *loan* or the *librarian* is not allowed to renew the *loan*.

Return Loancopy

Use case name: Return Loancopy

Participating actor: Librarian

Parameters: in UserID: Integer, in ItemID: Integer

Entry condition:

1. The Librarian requests the *loan* information from the *librarianterminal*. The Find Loan use case is used. If the *loan* doesn't exist, then proceeds to 4.

Flow of events:

2. The *loan* information is displayed on the *librarianterminal*.

3. The *librarian* requests remove *loan*. The system checks whether the *loan* is overdue. If the *loan* is overdue, the system prompts the corresponding message to notify the *librarian* and the fine information will be recorded. If the loan has been overdue for more than 3 days, the loan owner's privilege will be revoked. Only after the user's fine is cleared, can the user's privilege be retained.

Exit condition:

4. The corresponding *loan* is destroyed or the request is denied.

Monitor System

Use case name: Monitor System

Participating actor: Librarian

Parameters:

Entry condition:

1. The *librarian* requests to monitor the system from the *librarianterminal*.

Flow of events:

2. All the book *titles* and the *users* of the library are displayed on the *librarianterminal*.

Exit condition:

3. The *librarian* completes monitoring the system.

