|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Use Case Name:**  Return Book |  | **ID:**7 |  | **Importance Level:** | | Middle |
| **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 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 in system, if input wrong account or password, input it again. 2. The librarian clicks return-book button to use return-book function. 3. The librarian need to input book id in system, if system didn’t find book id, then librarian need to input book id again. 4. The system will check book’s state.   If the book is overdue, reduce member’s overdue book.   1. The system will clean book’s borrower, book time and overdue time. | | | | | | |
| **Sub Flow:** | | | | | | |
|  | | | | | | |
| **Alternative / Exception Flow:** | | | | | | |
|  | | | | | | |