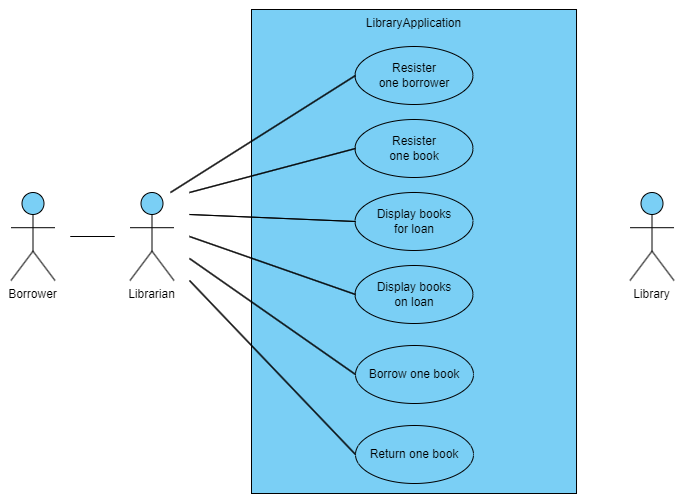
TeamProject(Iteration#1)

3팀

|  |  |  |
| --- | --- | --- |
| 학번 | 성명 | 팀장 |
| 2020315024 | 이용우 | O |
| 2021320033 | 황주원 | X |
| 2020210002 | 김무원 | X |
| 2022320029 | 이상민 | X |

Use Case Diagram



Use Case Description<1> registerOneBorrower

|  |  |
| --- | --- |
| Use Case Name | registerOneBorrower |
| Primary Actor | Librarian |
| Secondary Actor | Borrower |
| 개요 | 등록되지 않은 Borrower가 Librarian에게 이용자 등록을 신청한다. |
| Main Scenario | |
| Actor Action | System Response |
| 1.등록되지 않은 Borrower가 이용자 등록을 위해 이용자 등록신청서(u\_name)를 작성하고 Librarian에게 제출한다. |  |
| 2.Librarian은 Borrower에게 받은 이용자 등록신청서(u\_name)를 바탕으로 Borrower의 정보를 시스템에 등록한다. |  |
|  | 3.u\_name을 담은 Borrower 객체를 생성한다. |
|  | 4.Borrower 객체를 BorrowerCollection 객체에 저장 후 속성을 출력한다. |
| 5.Librarian은 Borrower에게 등록되었음을 고지하고 회원증을 발급해준다. |  |
| Alternative Scenario |  |

Use Case Description<2> registerOneBook

|  |  |
| --- | --- |
| Use Case Name | registerOneBook |
| Primary Actor | Librarian |
| Secondary Actor | None |
| 개요 | Librarian이 등록되지 않은 Book을 Library에 등록한다. |
| Main Scenario | |
| Actor Action | System Response |
| 1.Librarian이 등록되지 않은 Book의 title, author, c\_num를 확인한다. |  |
| 2.Librarian은 Book의 title, author, c\_num를 시스템에 등록한다. |  |
|  | 3.title, author, c\_num를 담은 Book 객체를 생성한다. |
|  | 4.Book 객체를 BookCollection 객체에 저장하고 속성을 출력한다. |
| Alternative Scenario |  |

Use Case Description<3> displayBooksForLoan

|  |  |
| --- | --- |
| Use Case Name | displayBooksForLoan |
| Primary Actor | Librarian |
| Secondary Actor | Borrower (Option) |
| 개요 | 시스템이 대출가능한 Book들을 화면에 출력한다. |
| Main Scenario | |
| Actor Action | System Response |
| 1.Librarian은 시스템에 대출가능한 Book의 목록을 요청한다. |  |
|  | 2.BookCollection 객체에서 Book객체들을 오름차순으로 정렬한다. |
|  | 3.정렬된 BookCollection 객체에서 Book 객체들 각각 isBookForLoan을  확인하고, true일 경우 속성을 출력한다. |
| Alternative Scenario |  |

Use Case Description<4> displayBooksOnLoan

|  |  |
| --- | --- |
| Use Case Name | displayBooksOnLoan |
| Primary Actor | Librarian |
| Secondary Actor | Borrower (Option) |
| 개요 | 시스템이 대출중인 Book들을 화면에 출력한다. |
| Main Scenario | |
| Actor Action | System Response |
| 1.Librarian은 시스템에 대출중인 Book의 목록을 요청한다. |  |
|  | 2.Book Collection 객체에서 Book 객체들을 오름차순으로 정렬한다. |
|  | 3.정렬된 BookCollection 객체에서 Book 객체들 각각 isBookOnLoan을 확인하고, true일 경우 속성을 출력한다. |
| Alternative Scenario |  |

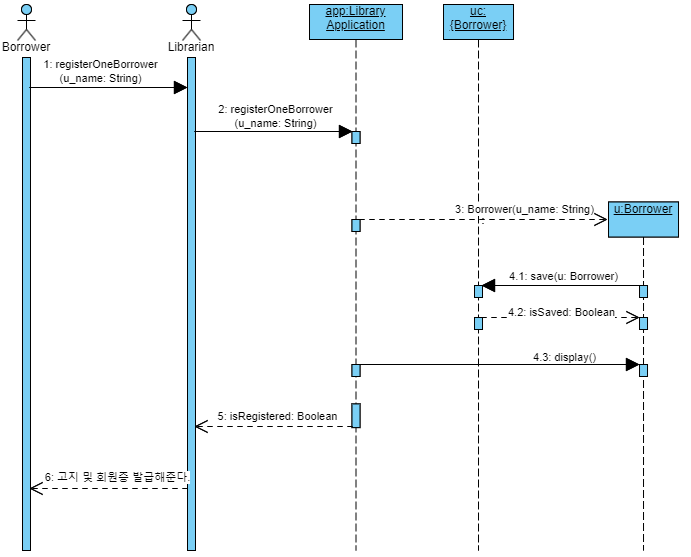
Use Case Description<5> borrowOneBook

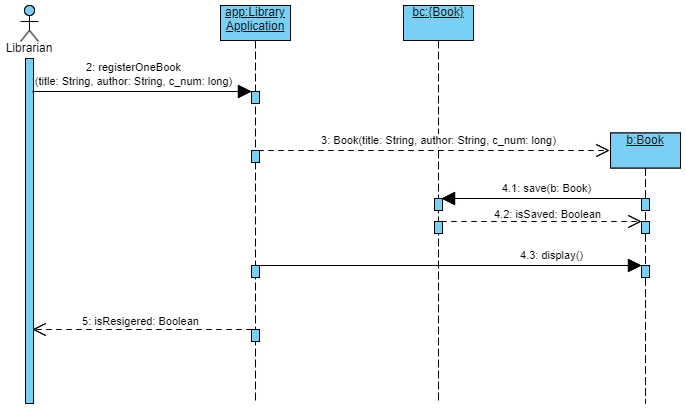
|  |  |
| --- | --- |
| Use Case Name | borrowOneBook |
| Primary Actor | Librarian |
| Secondary Actor | Borrower |
| 개요 | Borrower가 Librarian을 통해 one book을 대출한다. |
| Main Scenario | |
| Actor Action | System Response |
| 1.Borrower가 대출하려는 Book을 들고 대출창구에 도착한다. |  |
| 2.Borrower가 대출하려는 Book과 회원증을 Librarian에게 제출한다. |  |
| 3.Librarian이 대출할 Book과 회원증의 바코드(c\_num, u\_name)를 인식시킨다. |  |
|  | 4.BookCollection 객체에서 나온 대출할 Book 객체를 검색하고, BorrowerCollection 객체에서 Borrower 객체를 검색한다. |
|  | 5.Book 객체의 속성과 Borrower 객체의 속성을 화면에 출력한다. |
|  | 6.찾아낸 Book 객체와 Borrower 객체가 대출 가능한지 확인한다. |
|  | 7.Loan 객체를 생성하고 Book 객체와 Borrower 객체를 양방향으로 링크한다. (Borrower-Loan-Book) |
|  | 8.Loan 객체에서 대출일자를 현재시간으로 하고 반납기한일자를 계산하고 저장한다. |
|  | 9.Loan 객체를 LoanCollection 객체에 저장한다. |
|  | 10.Loan 객체의 속성을 화면에 출력한다. |
| 11. Librarian은 Borrower에게 Book과 회원증을 건네고, 반납일자를 고지한다. |  |
| 12. Borrower는 Book과 회원증을 받고 대출창구를 떠난다. |  |
| Alternative Scenario |  |

Use Case Description<6> returnOneBook

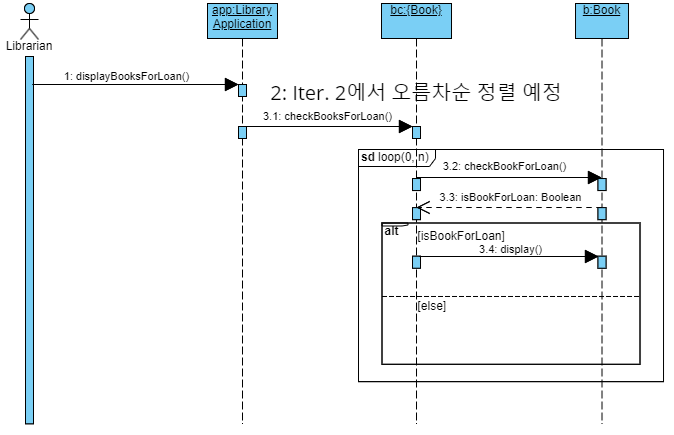
|  |  |
| --- | --- |
| Use Case Name | returnOneBook |
| Primary Actor | Librarian |
| Secondary Actor | Borrower |
| 개요 | Borrower가 borrow한 Book을 Librarian을 통해 Return한다. |
| Main Scenario | |
| Actor Action | System Response |
| 1.Borrower가 반납하려는 Book을 들고 반납창구에 도착한다. |  |
| 2.Borrower가 반납하려는 Book을 Librarian에게 제출한다. |  |
| 3.Librarian이 반납할 Book의 바코드(c\_num)를 인식시킨다. |  |
|  | 4.BookCollection 객체에서 반납할 Book 객체를 찾는다. |
|  | 5.Book 객체에 연결된 Loan 객체를 찾는다. |
|  | 6.Loan 객체가 링크를 해제하는 메시지를 받고, Loan객체와 연결된 Book 객체, Borrower 객체 각각의 양방향 링크를 해제한다. |
|  | 7.Loan 객체에서 반납된 일자를 현재시간으로 저장한다. |
|  | 8.Loan 객체에서 속성을 출력한다. |
| 9.Librarian은 Borrower에게 반납이 완료되었음을 고지한다. |  |
| 10.Borrower는 반납창구를 떠난다. |  |
| Alternative Scenario |  |

Sequence Diagram<1> registerOneBorrower

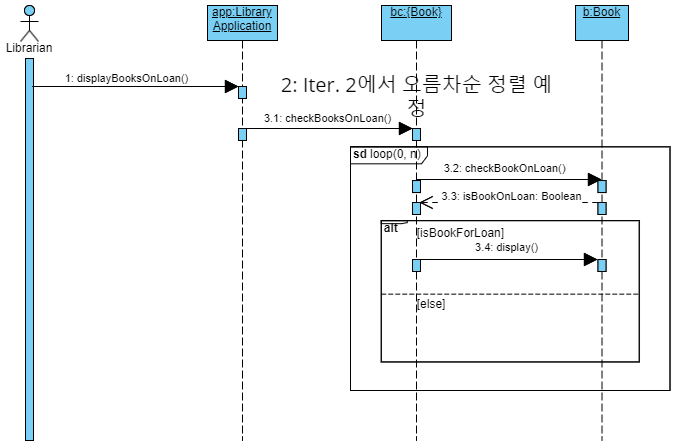


Sequence Diagram<2> registerOneBook

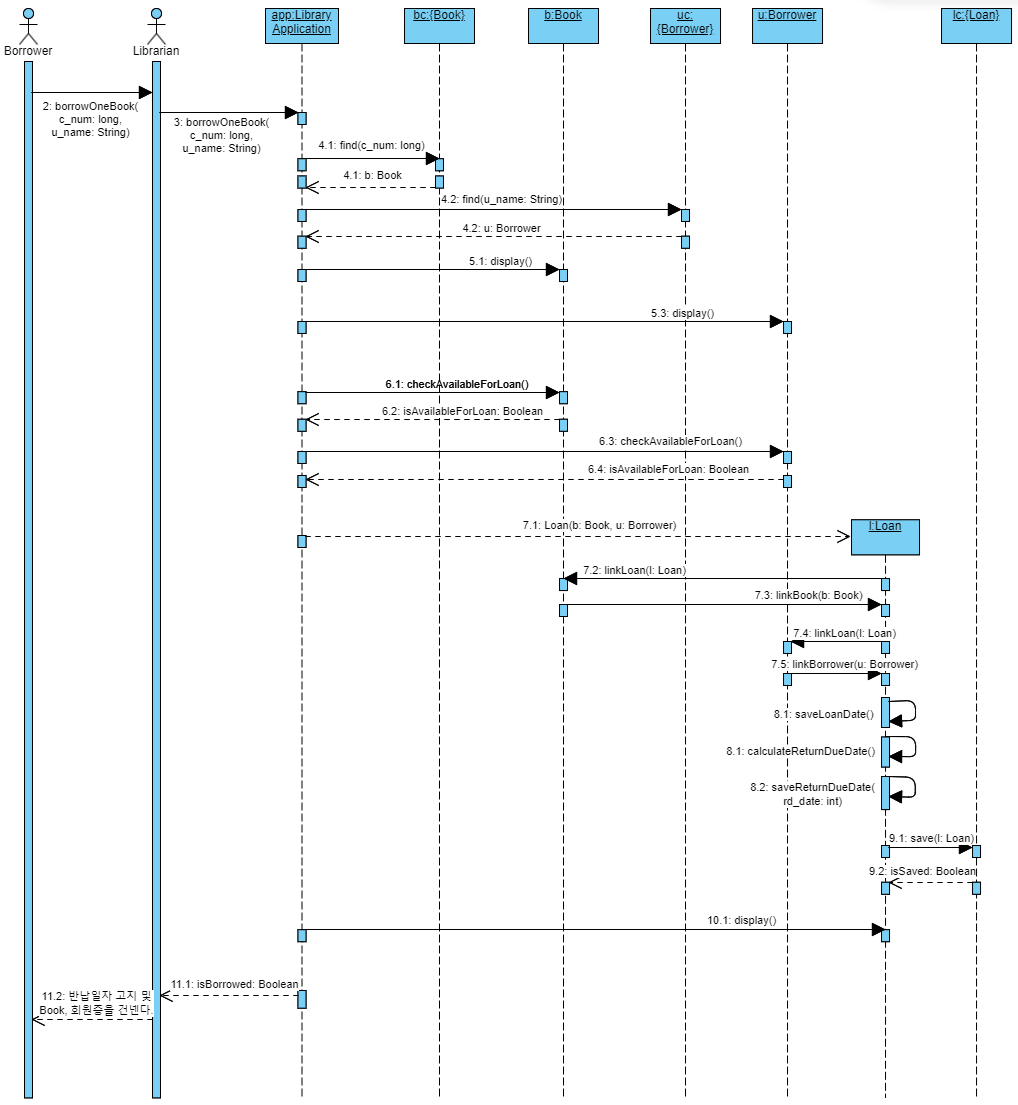
Sequence Diagram<3> displayBooksForLoan



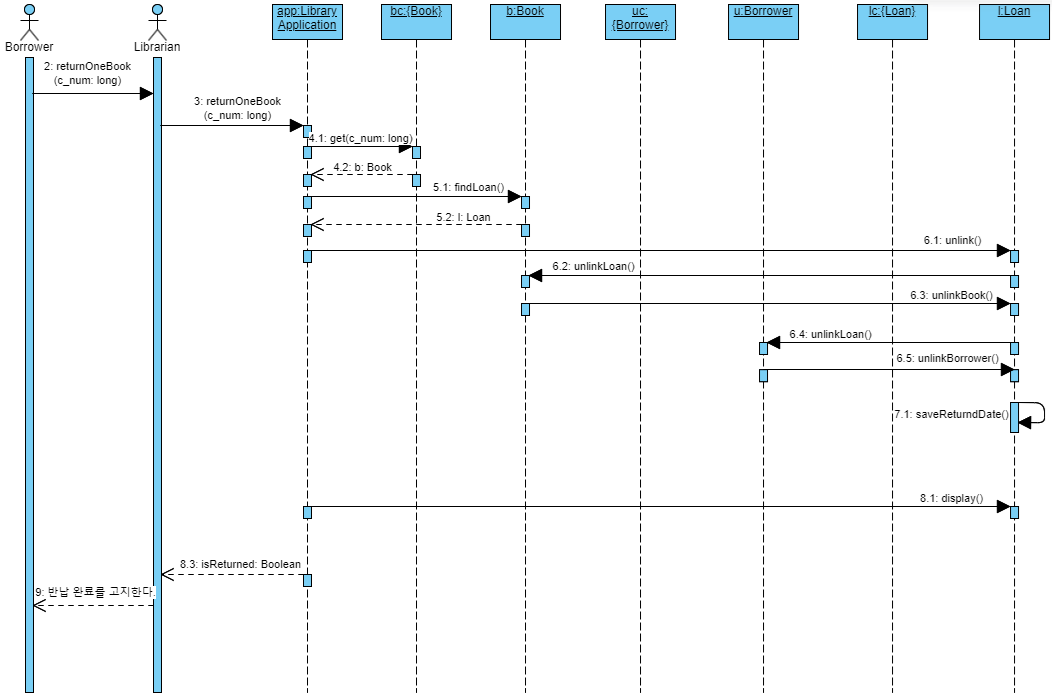
Sequence Diagram<4> displayBooksOnLoan



Sequence Diagram<5> borrowOneBook



Sequence Diagram<6> returnOneBook



Class Diagram

