

클래스 다이어그램 작성 사례: Library Application

Prof. Kim, Hyeon Soo



충남대학교

CHUNGNAM NATIONAL UNIVERSITY

SEAL

Software Engineering & Application Lab.
소프트웨어 공학 및 응용 연구실

목차

1. Introduction
2. Class Diagram
3. Use Case와 Class 간의 관계
4. Class Specification

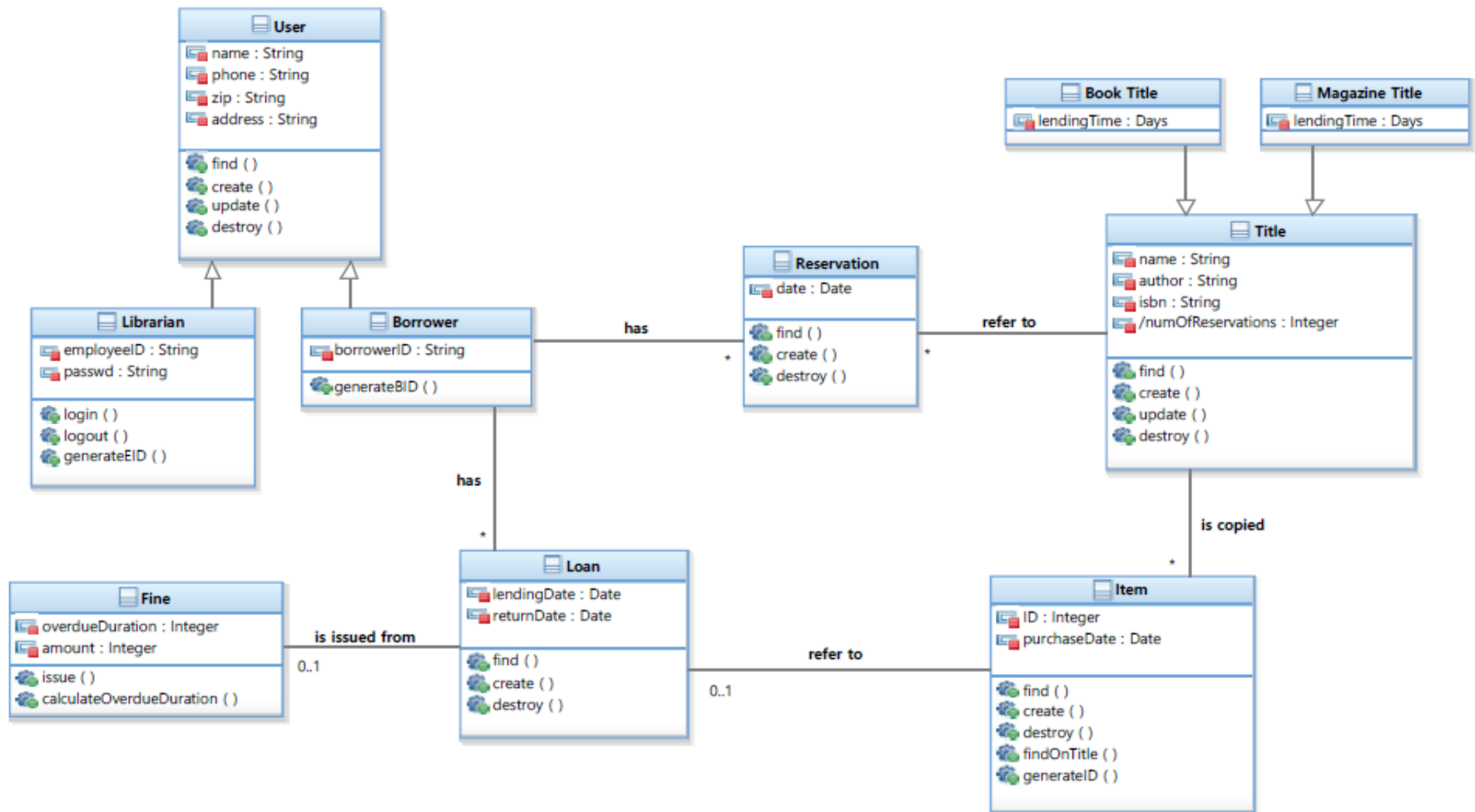
1. Introduction

▪ Objective

- 이 문서는 Library application에 대한 시스템 모델(클래스 다이어그램)에 대한 내용을 기술한다
- 이 문서는 요구사항을 분석하기 위해 수행한 시스템 모델링의 결과인 클래스 다이어그램과 각 클래스에 대한 명세를 포함한다

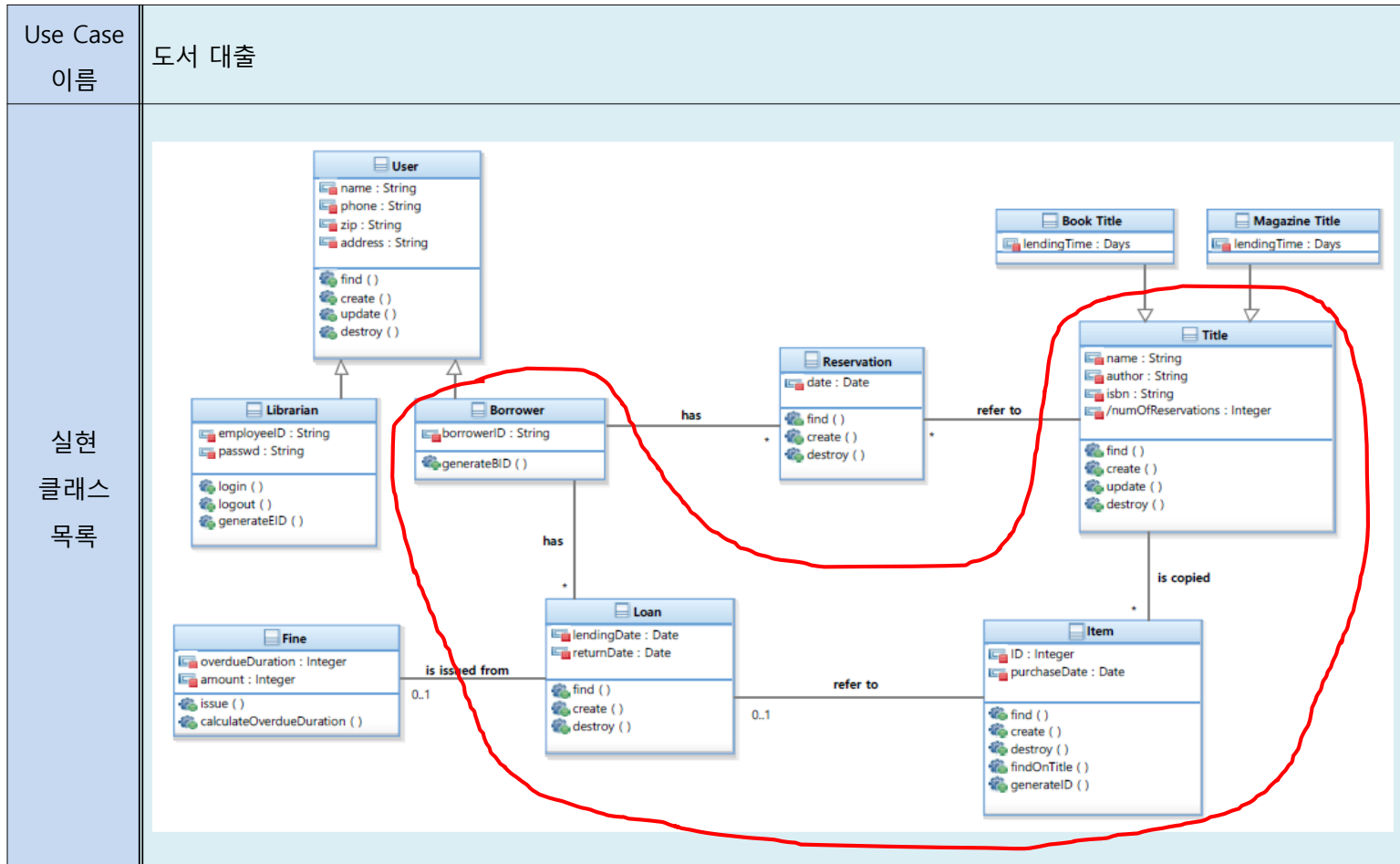
2. Class Diagram

4



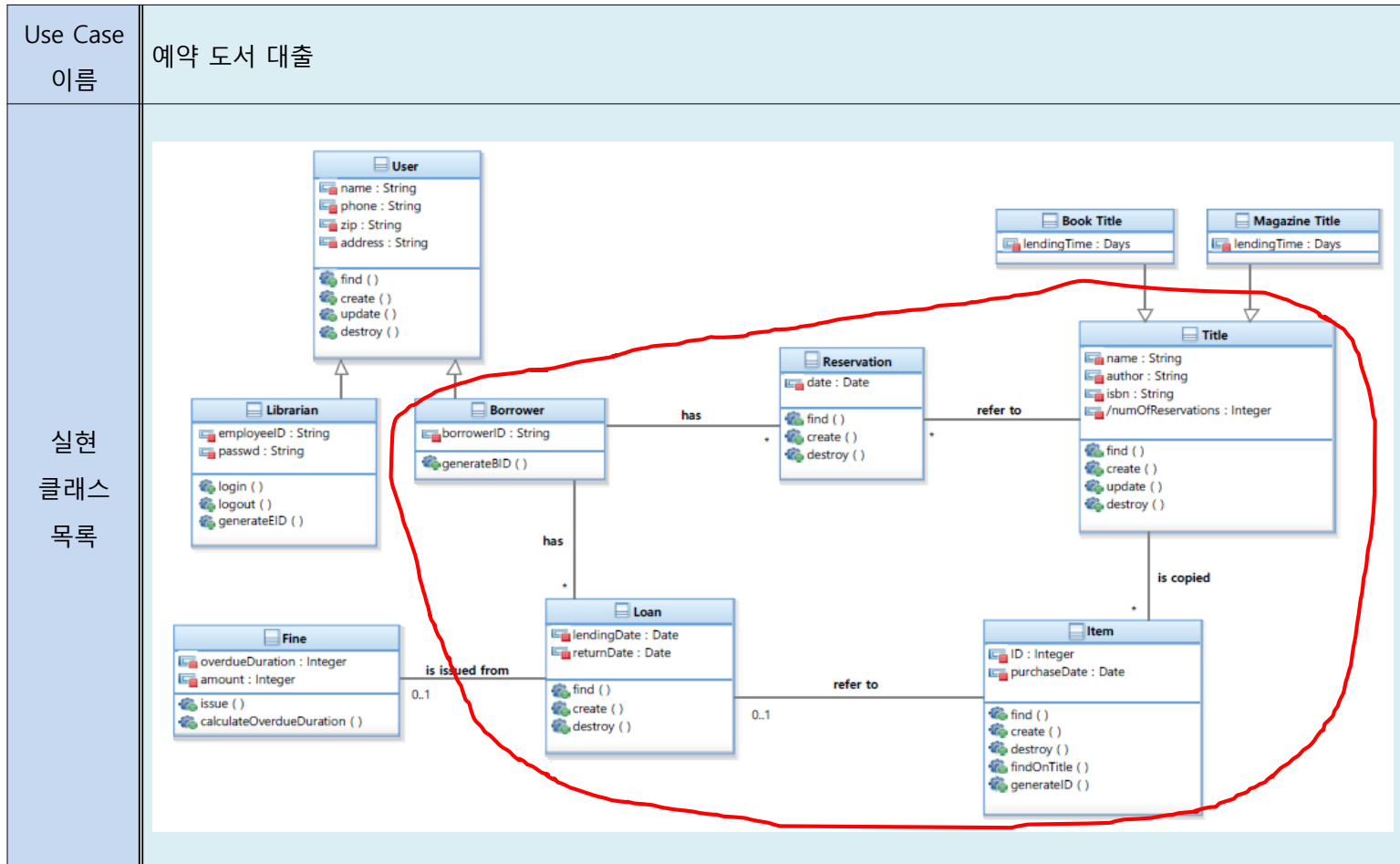
3. Use Case와 Class 간의 관계

3.1. UC: 도서 대출



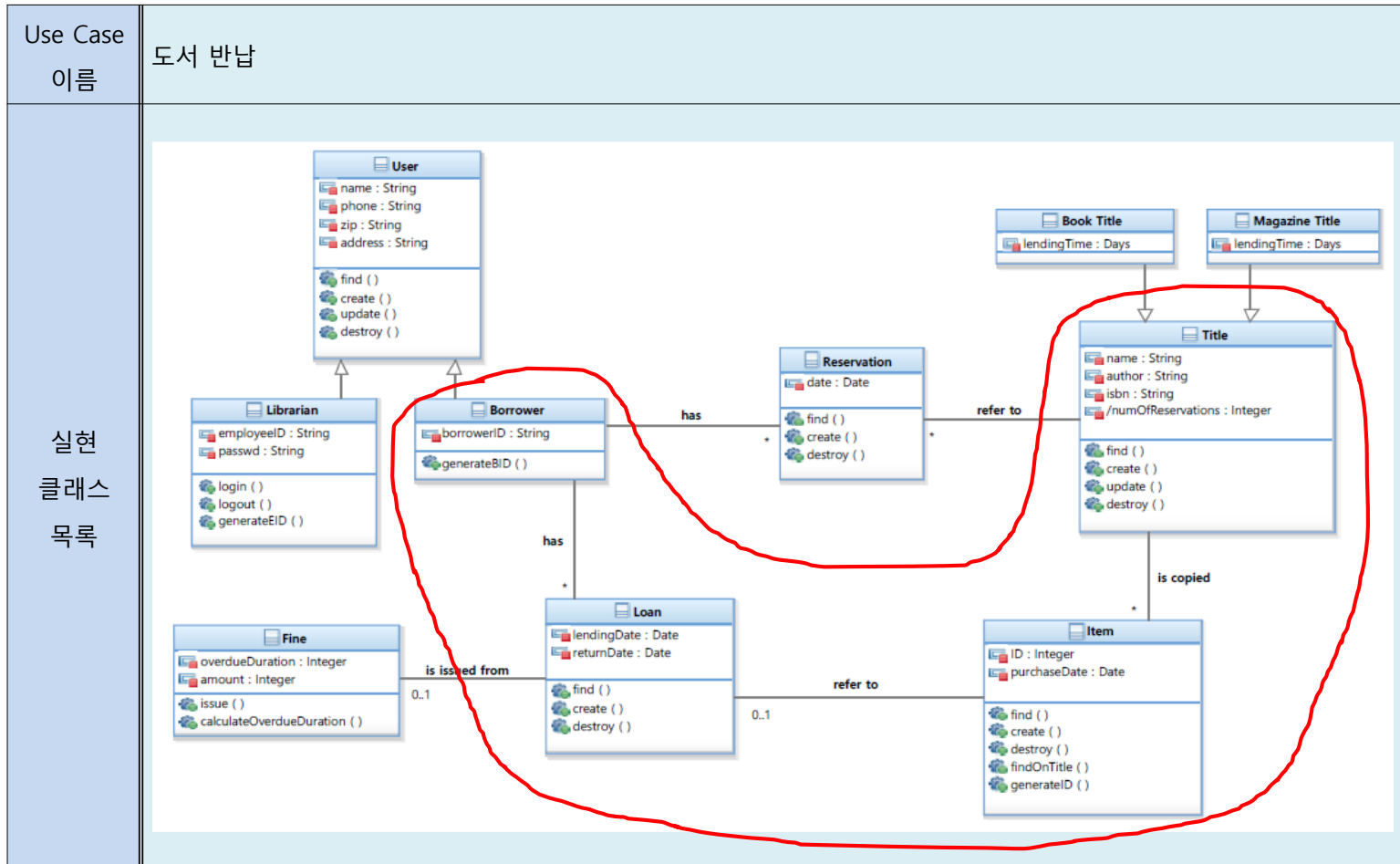
3. Use Case와 Class 간의 관계

3.2. UC: 예약 도서 대출



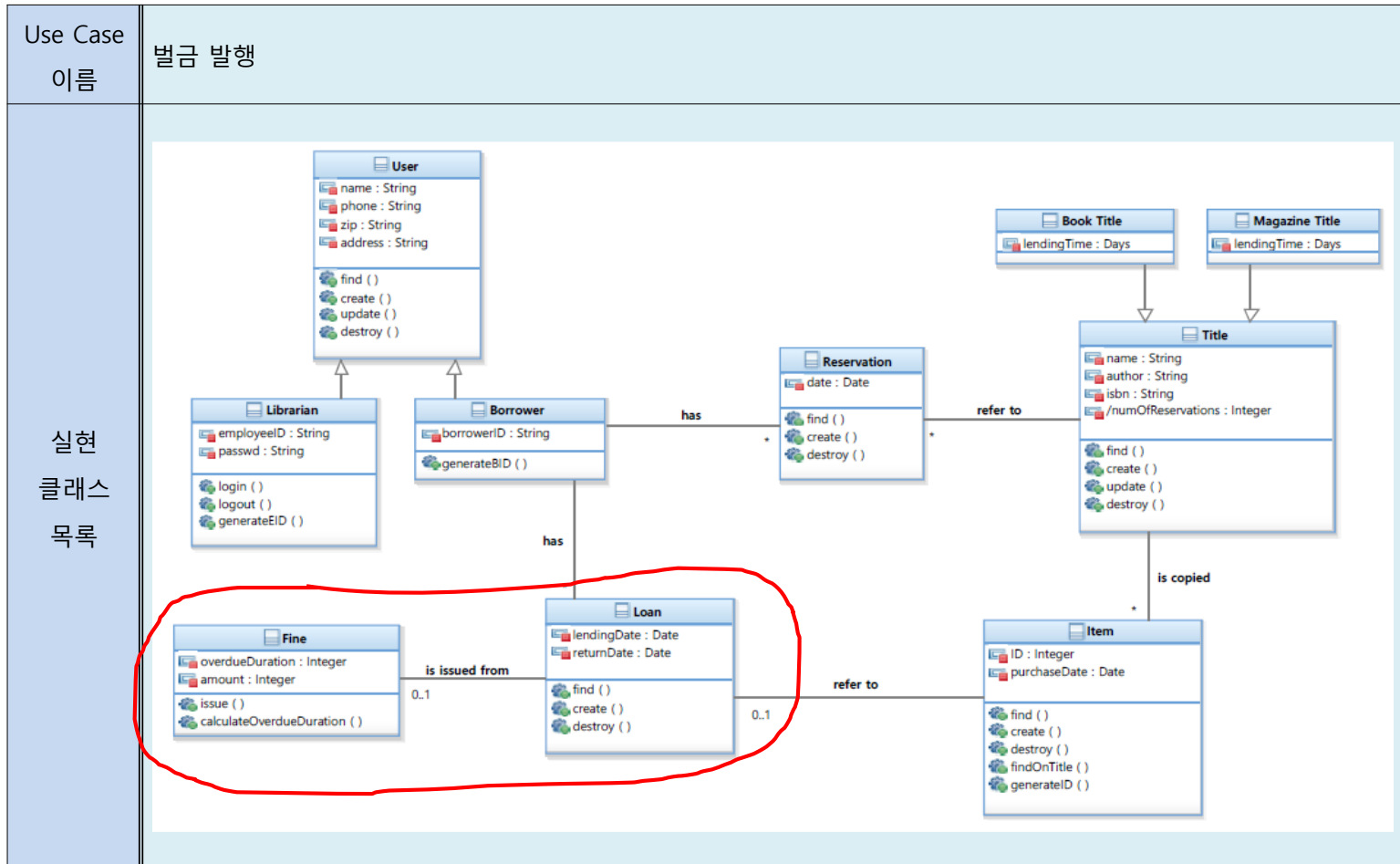
3. Use Case와 Class 간의 관계

3.3. UC: 도서 반납



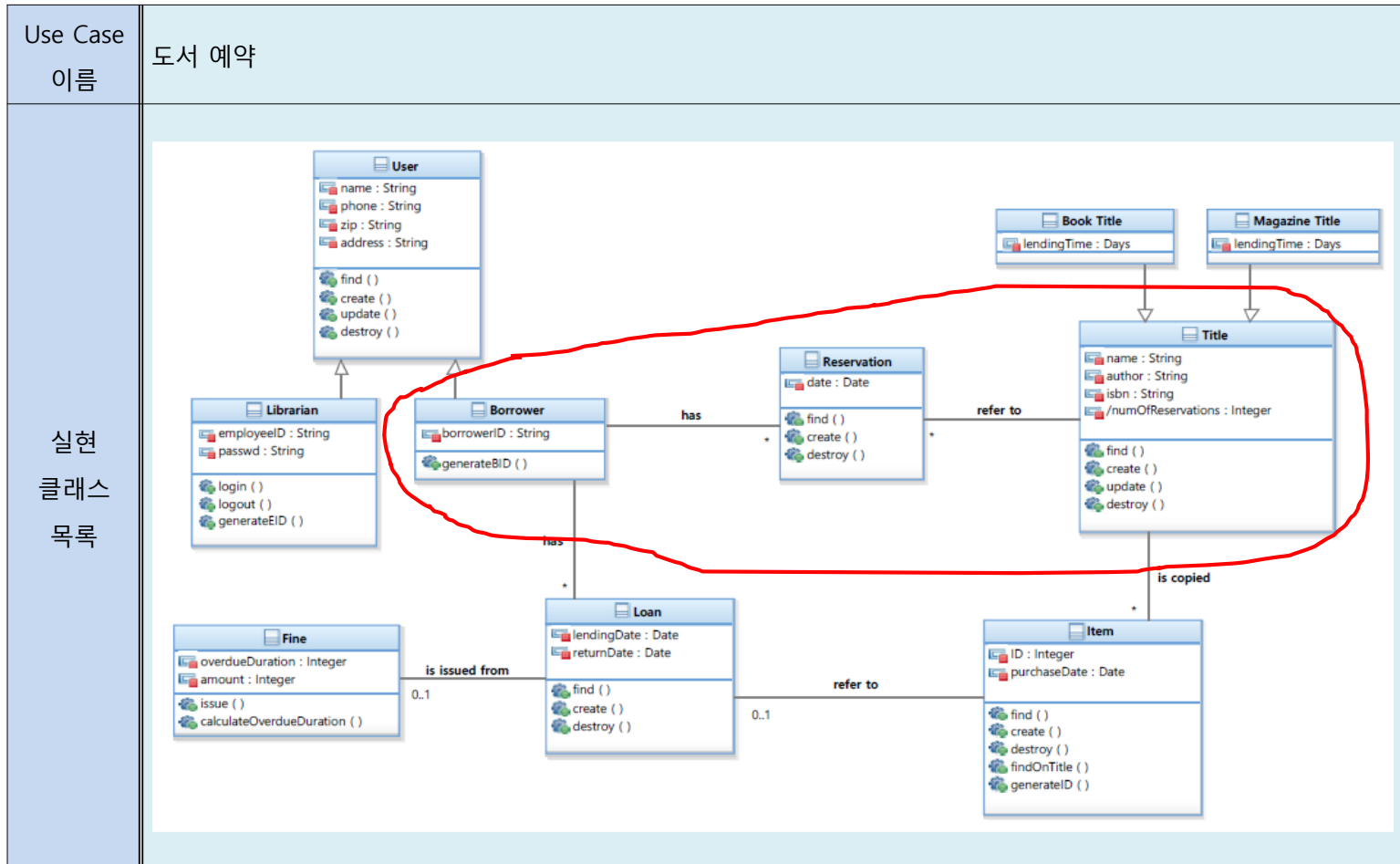
3. Use Case와 Class 간의 관계

3.4. Extension UC: 벌금 발행



3. Use Case와 Class 간의 관계

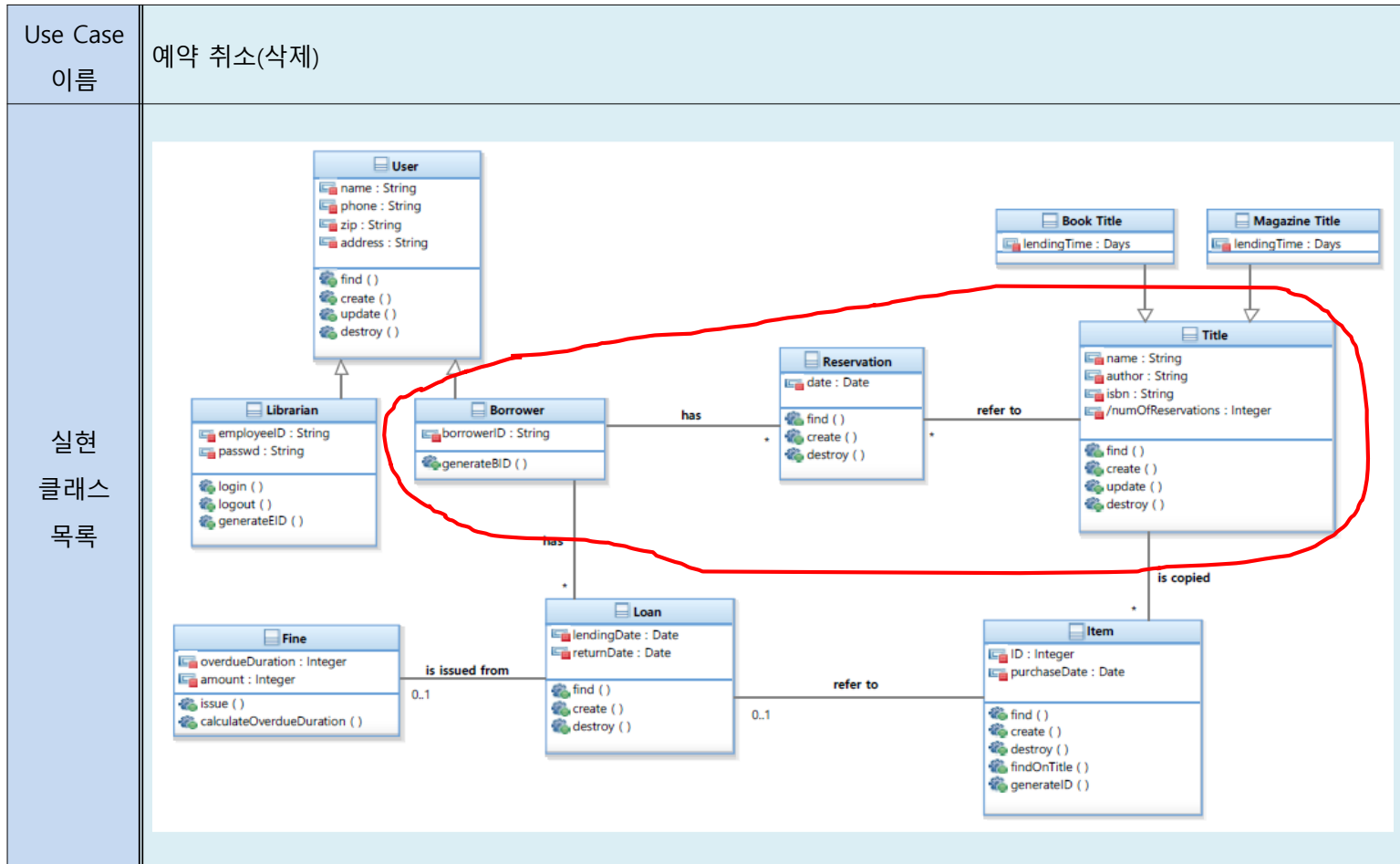
3.5. UC: 도서 예약



3. Use Case와 Class 간의 관계

10

3.6. UC: 예약 취소(삭제)



3. Use Case와 Class 간의 관계

3.7. UC: 도서 추가

Use Case 이름

도서 추가

실현 클래스 목록

```
classDiagram
    class User {
        name : String
        phone : String
        zip : String
        address : String
        find()
        create()
        update()
        destroy()
    }
    class Librarian {
        employeeID : String
        passwd : String
        login()
        logout()
        generateEID()
    }
    class Borrower {
        borrowerID : String
        generateBiD()
    }
    class Reservation {
        date : Date
        find()
        create()
        destroy()
    }
    class Title {
        name : String
        author : String
        isbn : String
        numOfReservations : Integer
        find()
        create()
        update()
        destroy()
    }
    class BookTitle {
        lendingTime : Days
    }
    class MagazineTitle {
        lendingTime : Days
    }
    class Item {
        ID : Integer
        purchaseDate : Date
        find()
        create()
        destroy()
        findOnTitle()
        generateID()
    }
    class Loan {
        lendingDate : Date
        returnDate : Date
        find()
        create()
        destroy()
    }
    class Fine {
        overdueDuration : Integer
        amount : Integer
        issue()
        calculateOverdueDuration()
    }

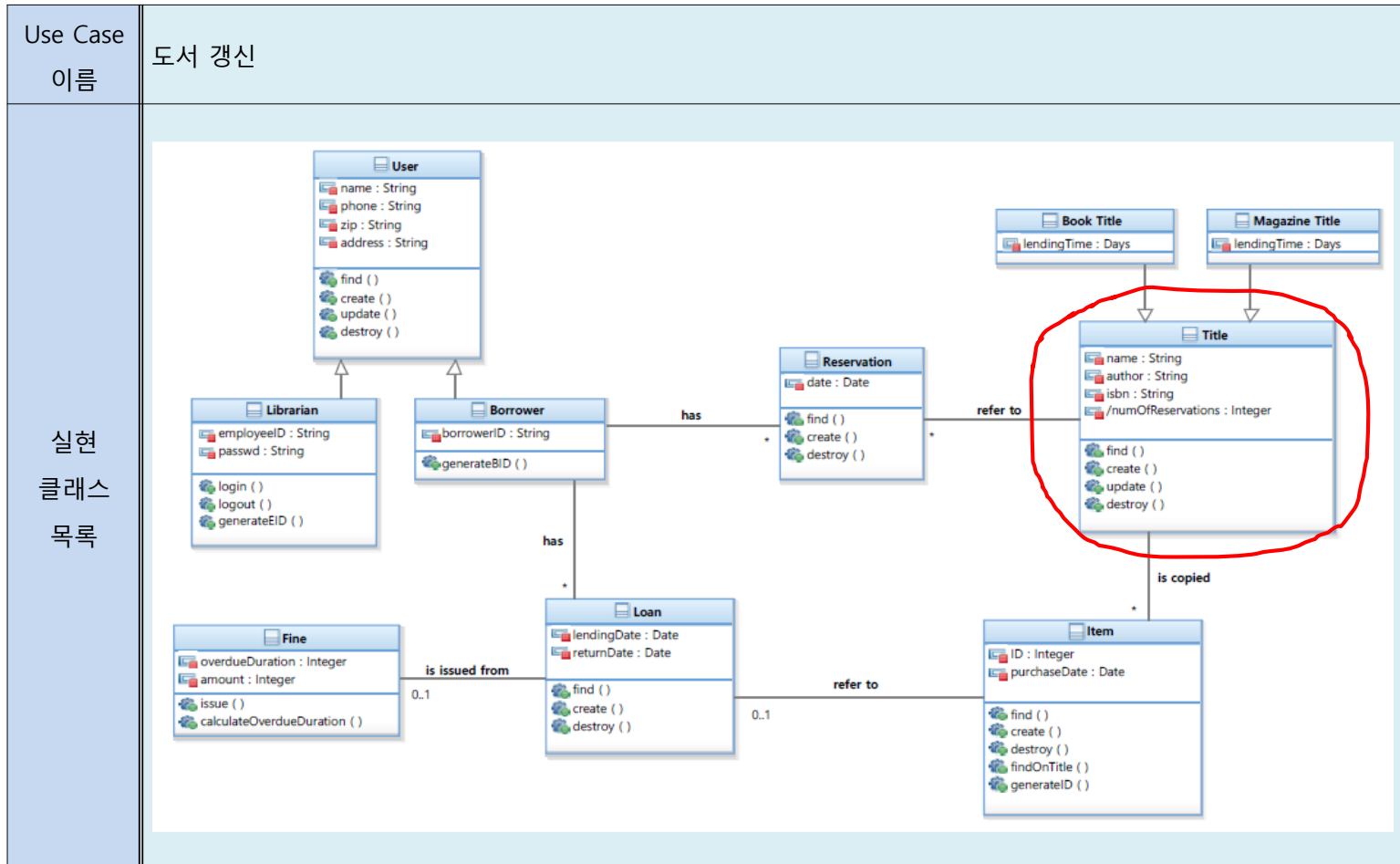
    User <|-- Librarian
    User <|-- Borrower
    Borrower "1" -- "*" Loan : has
    Loan "*" -- "0..1" Fine : is issued from
    Loan "1" -- "*" Reservation : has
    Reservation "*" -- "*" Title : refer to
    Title <|-- BookTitle
    Title <|-- MagazineTitle
    Title "1" -- "*" Item : is copied
    Item "1" -- "0..1" Loan : refer to
```

The diagram illustrates the relationships between various classes in a library system. The **User** class is the base for **Librarian** and **Borrower**. **Borrower** has a **Loan** (1 to many). **Loan** is issued from a **Fine** (many to 0..1). **Borrower** also has a **Reservation** (1 to many). **Reservation** refers to a **Title** (many to many). **Title** is the base for **Book Title** and **Magazine Title**. **Title** is copied into an **Item** (1 to many). **Item** refers to a **Loan** (1 to 0..1). Each class has specific attributes and methods as detailed in the class boxes.

3. Use Case와 Class 간의 관계

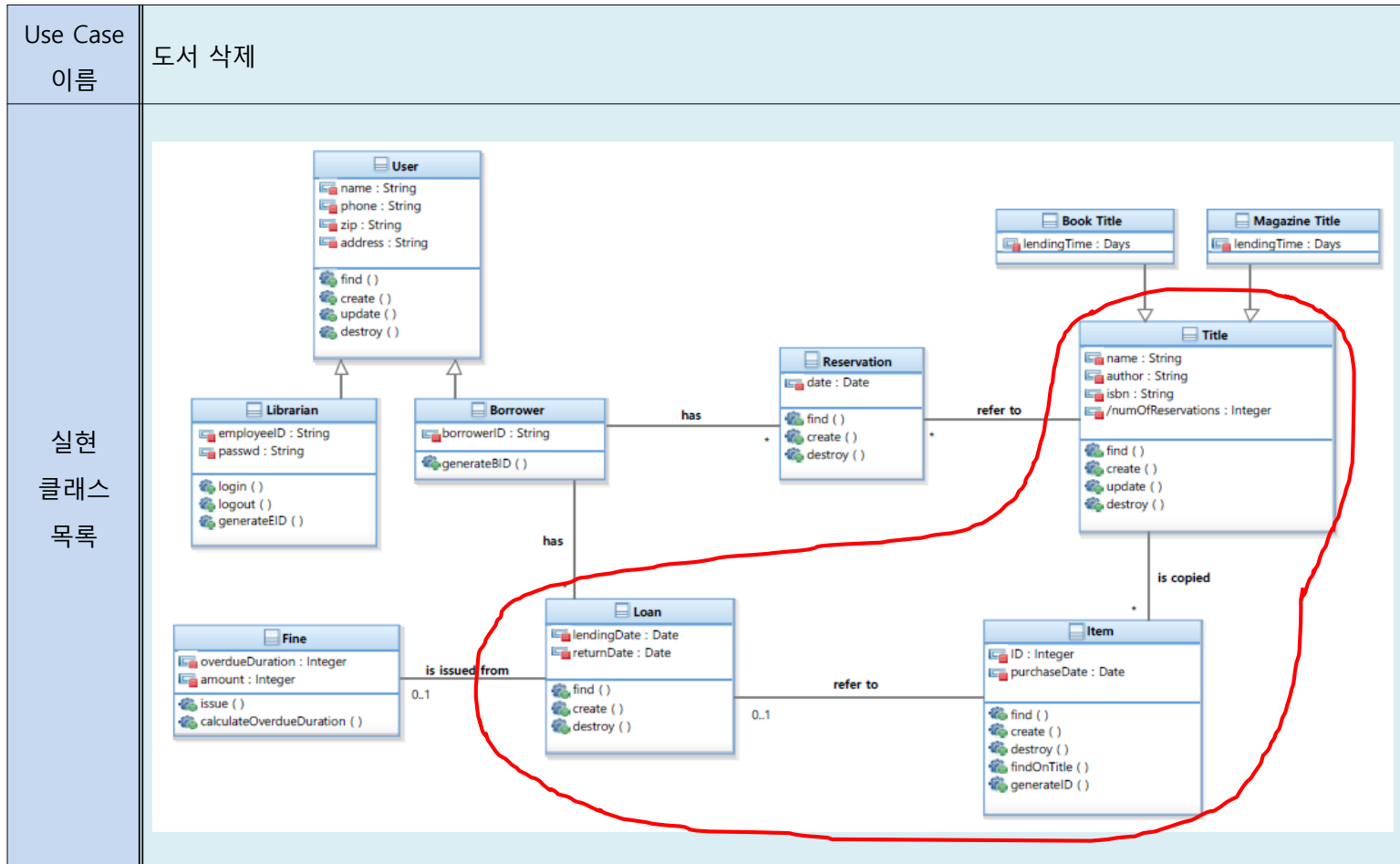
12

3.8. UC: 도서 갱신



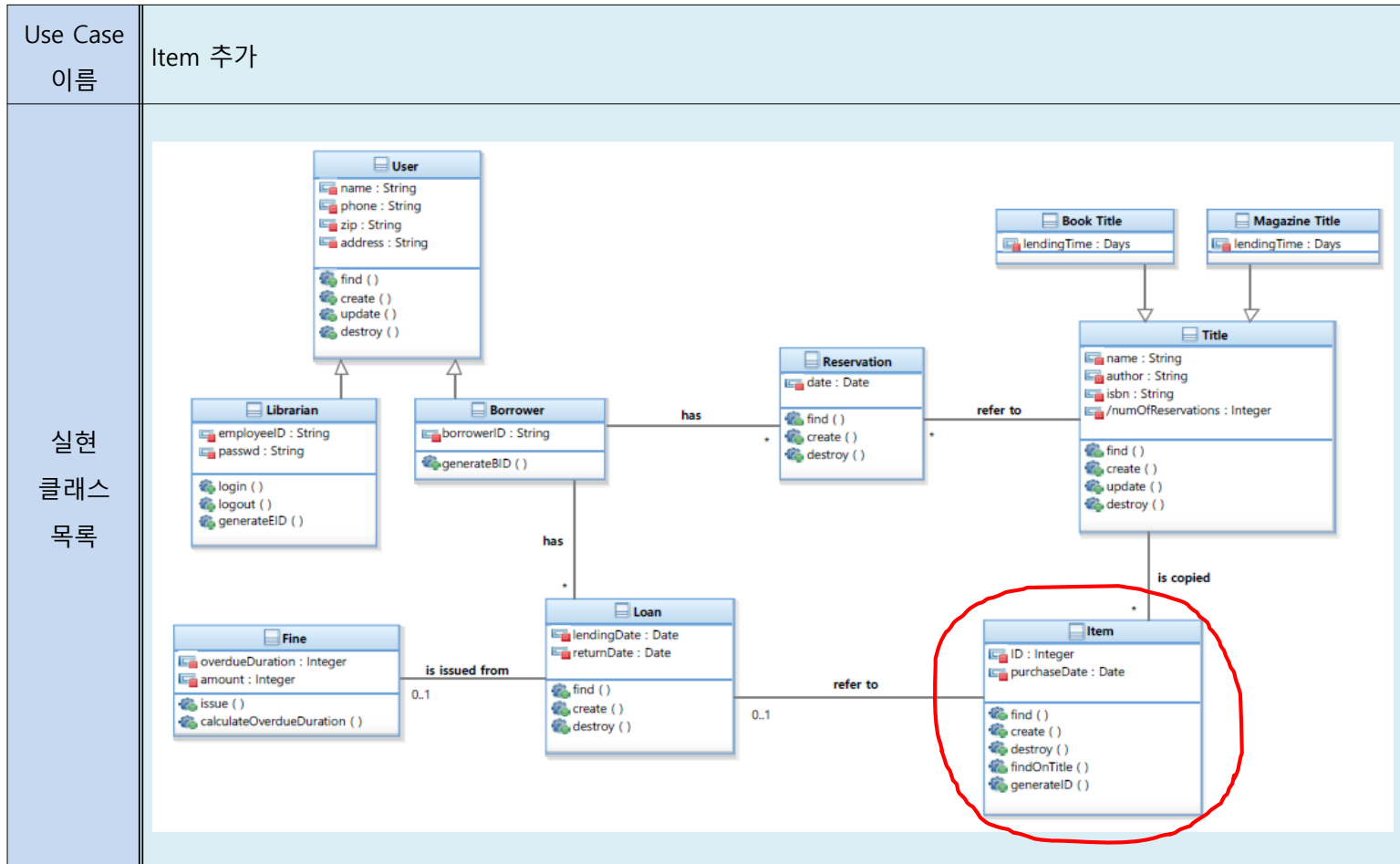
3. Use Case와 Class 간의 관계

3.9. UC: 도서 삭제



3. Use Case와 Class 간의 관계

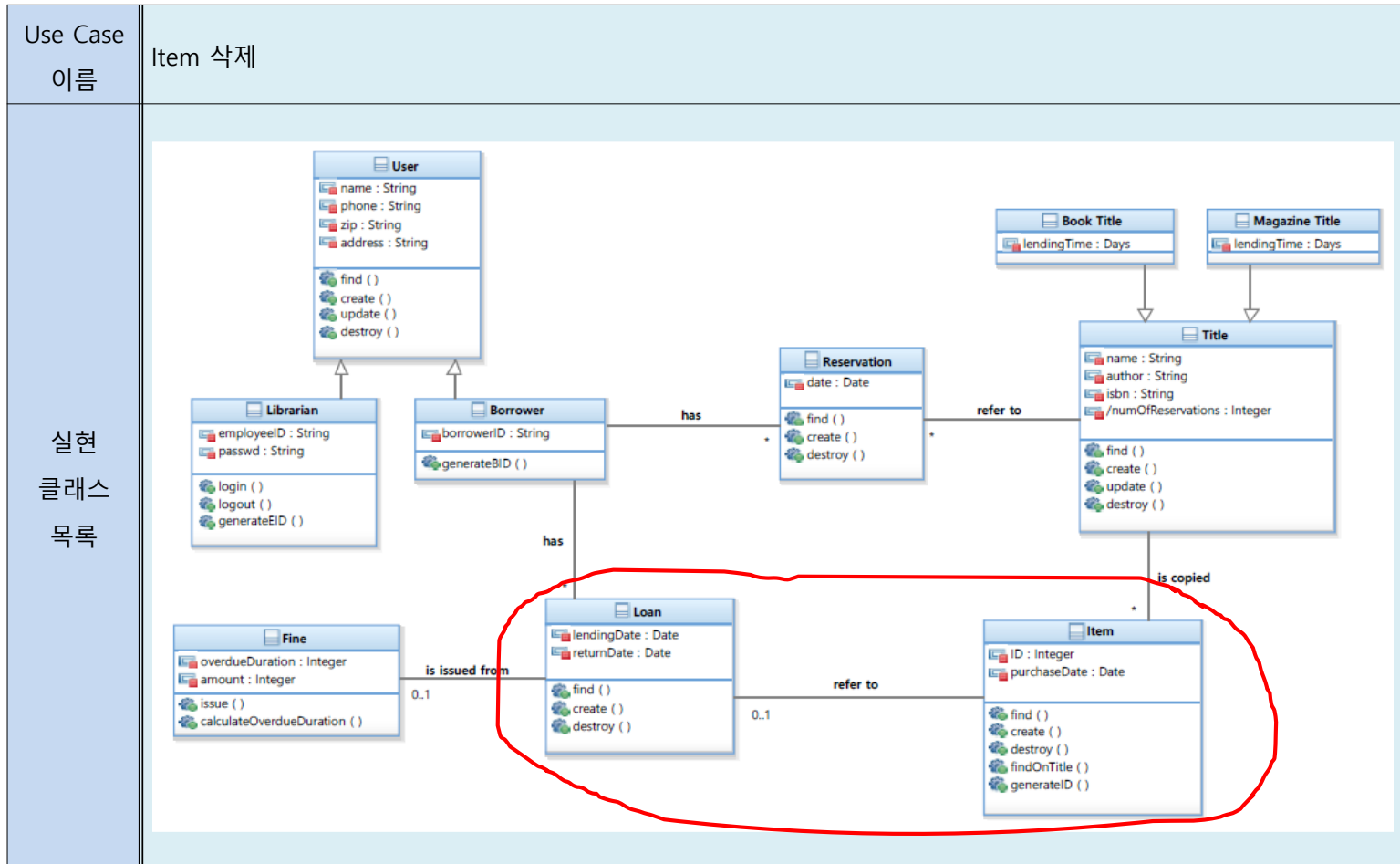
3.10. UC: Item 추가



3. Use Case와 Class 간의 관계

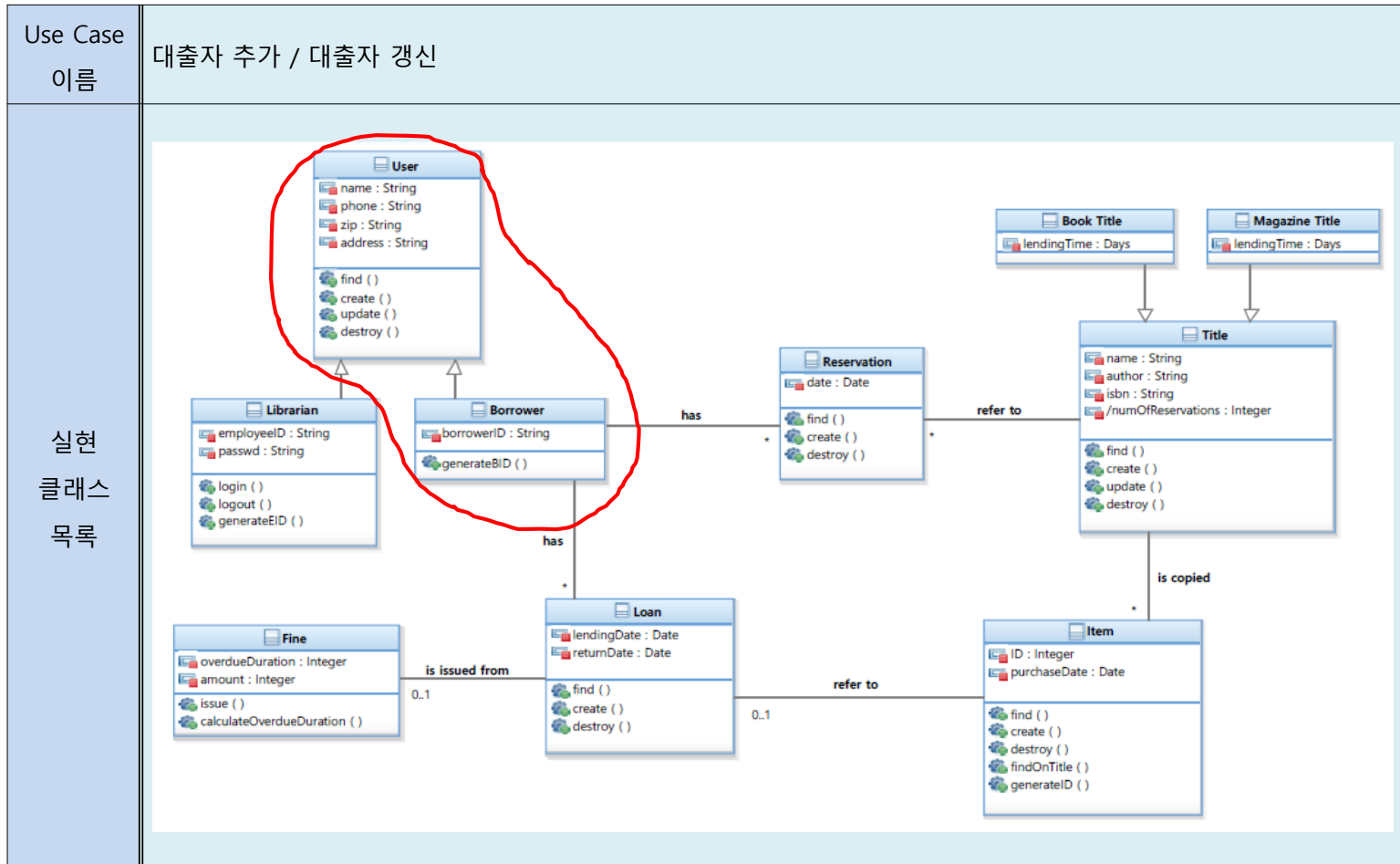
15

3.11. UC: Item 삭제



3. Use Case와 Class 간의 관계

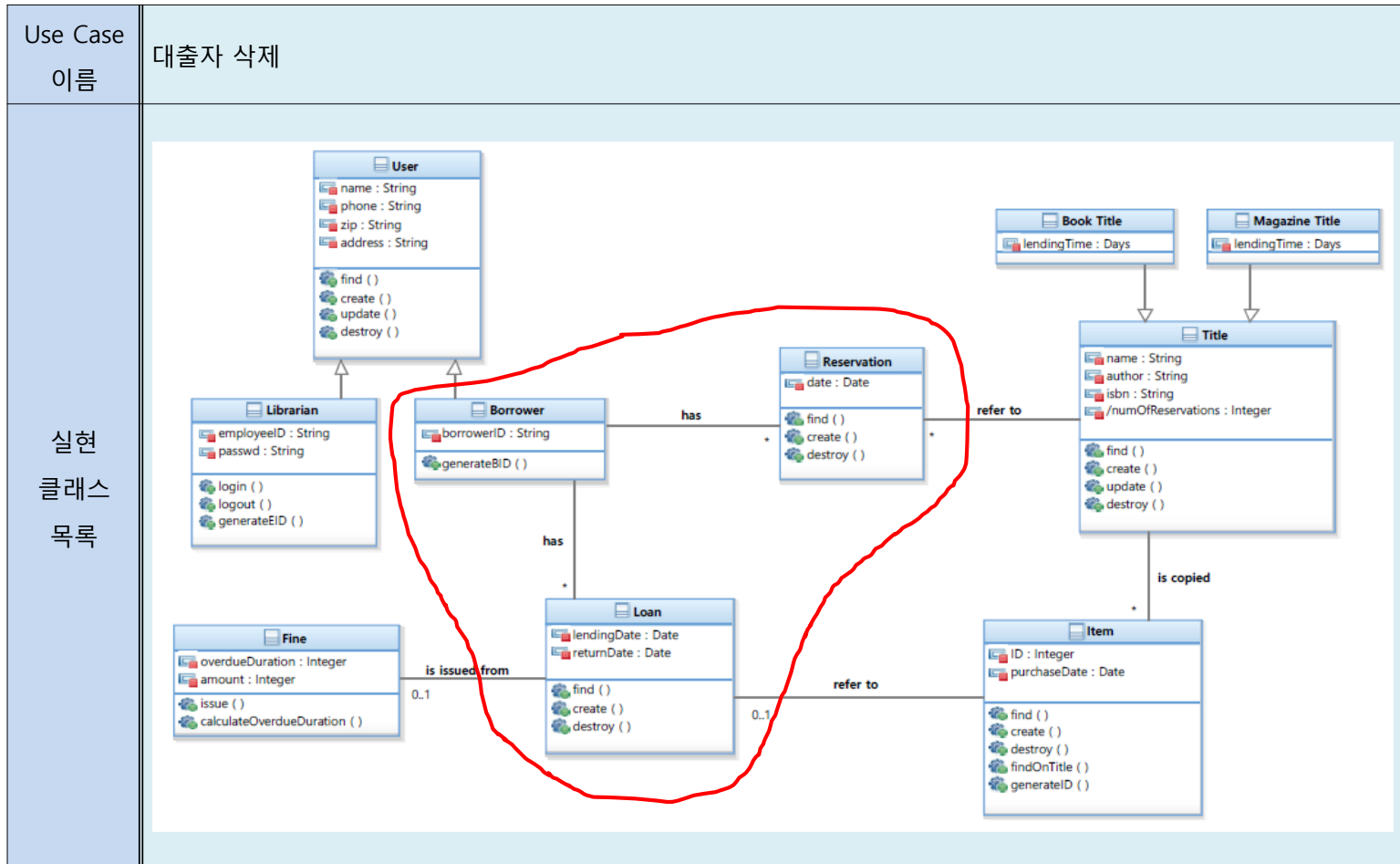
3.12. UC: 대출자 추가 / UC: 대출자 갱신



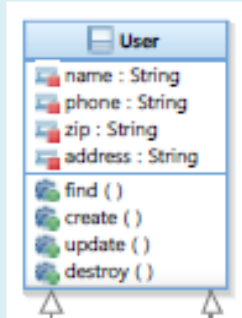
3. Use Case와 Class 간의 관계

17

3.13. UC: 대출자 삭제



4. Class Specification

Class Name	User		
Class Diagram	 <pre> classDiagram class User { name : String phone : String zip : String address : String find () create () update () destroy () } </pre>		
Responsibility	Librarian과 borrower와 같이 도서관을 이용하는 사용자에게 대한 공통 정보를 유지한다.		
Attribute	Type	Name	Description
	String	name	사용자의 이름
	String	phone	사용자의 전화번호
	String	zip	사용자 주소의 우편번호
	String	address	사용자의 주소

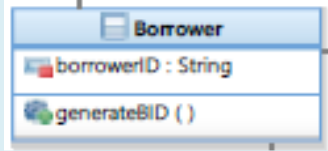
4. Class Specification

Class Name	User			
Operation	Return Type	Method Name	Parameter Type	Parameter Name
		find		
	Description	사용자에 대한 정보를 조회하거나 확인한다.		
		create		
	Description	사용자에 대한 정보를 생성한다.		
		update		
	Description	사용자에 대한 정보를 변경한다.		
		destroy		
	Description	사용자에 대한 정보를 삭제한다.		

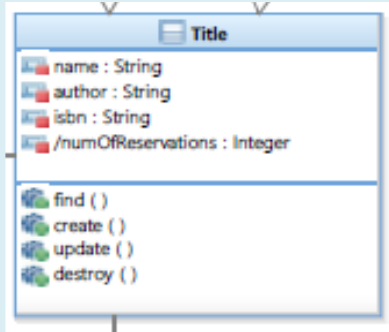
4. Class Specification

Class Name	Librarian			
Class Diagram	<pre> classDiagram class Librarian { employeeID : String passwd : String login () logout () generateEID () } </pre>			
Responsibility	사서(librarian)에 대한 정보를 유지한다.			
Attribute	Type	Name		Description
	String	employeeID		사서의 사원 번호
	String	passwd		사서의 비밀 번호
Operation	Return Type	Method Name	Parameter Type	Parameter Name
		login		
	Description	사서가 시스템을 사용하기 위해 로그인한다.		
		logout		
	Description	사서가 로그아웃한다.		
		generateEID		
	Description	사서에 대한 사원 번호를 생성한다.		

4. Class Specification

Class Name	Borrower			
Class Diagram				
Responsibility	대출자에 대한 정보를 유지한다.			
Attribute	Type	Name		Description
	String	borrowerID		대출자 ID
Operation	Return Type	Method Name	Parameter Type	Parameter Name
		generateBid		
	Description	대출자에 대한 대출자 ID를 생성한다.		

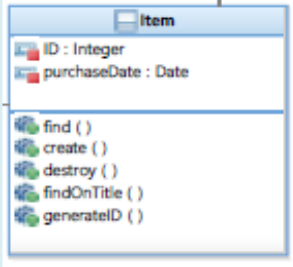
4. Class Specification

Class Name	Title		
Class Diagram	 <pre> classDiagram class Title { name : String author : String isbn : String /numOfReservations : Integer find () create () update () destroy () } </pre>		
Responsibility	책이나 잡지와 같이 도서관에서 관리하는 도서에 대한 공통 정보를 유지한다.		
Attribute	Type	Name	Description
	String	name	도서의 이름
	String	author	도서의 저자 이름
	String	isbn	isbn 번호
	Integer	numOfReservations	예약 개수

4. Class Specification

Class Name	Title			
Operation	Return Type	Method Name	Parameter Type	Parameter Name
		find		
	Description	도서에 대한 정보를 조회하거나 확인한다.		
		create		
	Description	도서에 대한 정보를 생성한다.		
		update		
	Description	도서에 대한 정보를 변경한다.		
		destroy		
	Description	도서에 대한 정보를 삭제한다.		

4. Class Specification

Class Name	Item		
Class Diagram	 <pre> classDiagram class Item { ID : Integer purchaseDate : Date find () create () destroy () findOnTitle () generateID () } </pre>		
Responsibility	책이나 잡지와 같이 도서관에서 관리하는 도서의 copy 본에 대한 정보를 유지한다.		
Attribute	Type	Name	Description
	Integer	ID	copy 본의 id 번호 (도서 관리 번호)
	Date	purchaseDate	copy 본의 구매 일자

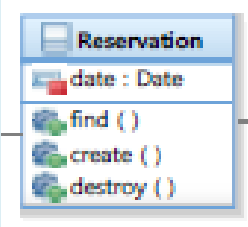
4. Class Specification

Class Name	Item			
Operation	Return Type	Method Name	Parameter Type	Parameter Name
		find		
	Description	Item에 대한 정보를 조회하거나 확인한다.		
		create		
	Description	Item에 대한 정보를 생성한다.		
		destroy		
	Description	Item에 대한 정보를 삭제한다.		
		findOnTitle		
	Description	Item에 대한 정보를 해당 도서 정보를 바탕으로 조회하거나 확인한다.		
		generateID		
	Description	Item에 대한 고유한 도서 관리 번호를 생성한다.		

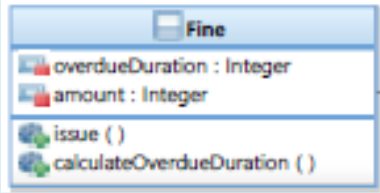
4. Class Specification

Class Name	Loan			
Class Diagram	 <pre> classDiagram class Loan { lendingDate : Date returnDate : Date find() create() destroy() } </pre>			
Responsibility	도서에 대한 대출 정보를 유지한다.			
Attribute	Type	Name		Description
	Date	lendingDate		도서 대출 일자
	Date	returnDate		도서 반납 일자
Operation	Return Type	Method Name	Parameter Type	Parameter Name
		find		
	Description	도서에 대한 대출 정보를 조회하거나 확인한다.		
		create		
	Description	도서에 대한 대출 정보를 생성한다.		
		destroy		
	Description	도서에 대한 대출 정보를 삭제한다.		

4. Class Specification

Class Name	Reservation			
Class Diagram	 <pre> classDiagram class Reservation { +date : Date +find () +create () +destroy () } </pre>			
Responsibility	도서에 대한 예약 정보를 유지한다.			
Attribute	Type	Name		Description
	Date	date		도서 예약 일자
Operation	Return Type	Method Name	Parameter Type	Parameter Name
		find		
	Description	도서에 대한 예약 정보를 조회하거나 확인한다.		
		create		
	Description	도서에 대한 예약 정보를 생성한다.		
		destroy		
	Description	도서에 대한 예약 정보를 삭제한다.		

4. Class Specification

Class Name	Fine			
Class Diagram	 <pre> classDiagram class Fine { overdueDuration : Integer amount : Integer issue () calculateOverdueDuration () } </pre>			
Responsibility	연체 도서에 대한 벌금 정보를 유지한다.			
Attribute	Type	Name		Description
	Integer	overdueDuration		도서 연체 기간
	Integer	amount		도서 연체 벌금
Operation	Return Type	Method Name	Parameter Type	Parameter Name
		issue		
	Description	연체 도서에 대한 벌금을 발행한다.		
		calculateOverdueDuration		
	Description	도서 연체 기간을 계산한다.		