

통합 응용SW개발 반 (java, C#)
김소현, 하이안

도서 관리 프로그램 C#

목차

- 도서 관리 프로그램 기본구조
- 도서 관리 프로그램 Winform
- 도서 관리 프로그램 흐름도
- 도서 관리 프로그램 실행 화면

개발환경



개발 언어

C#



IDE 툴

Visual Studio 2019



참고 문헌

윤인성의 C# 프로그래밍

도서 관리 프로그램 C# : 프로젝트 일정

4/16 ~ 17일

주제 선정

4/18 ~ 19일

화면 설계

4/20 ~ 24일

화면 구현

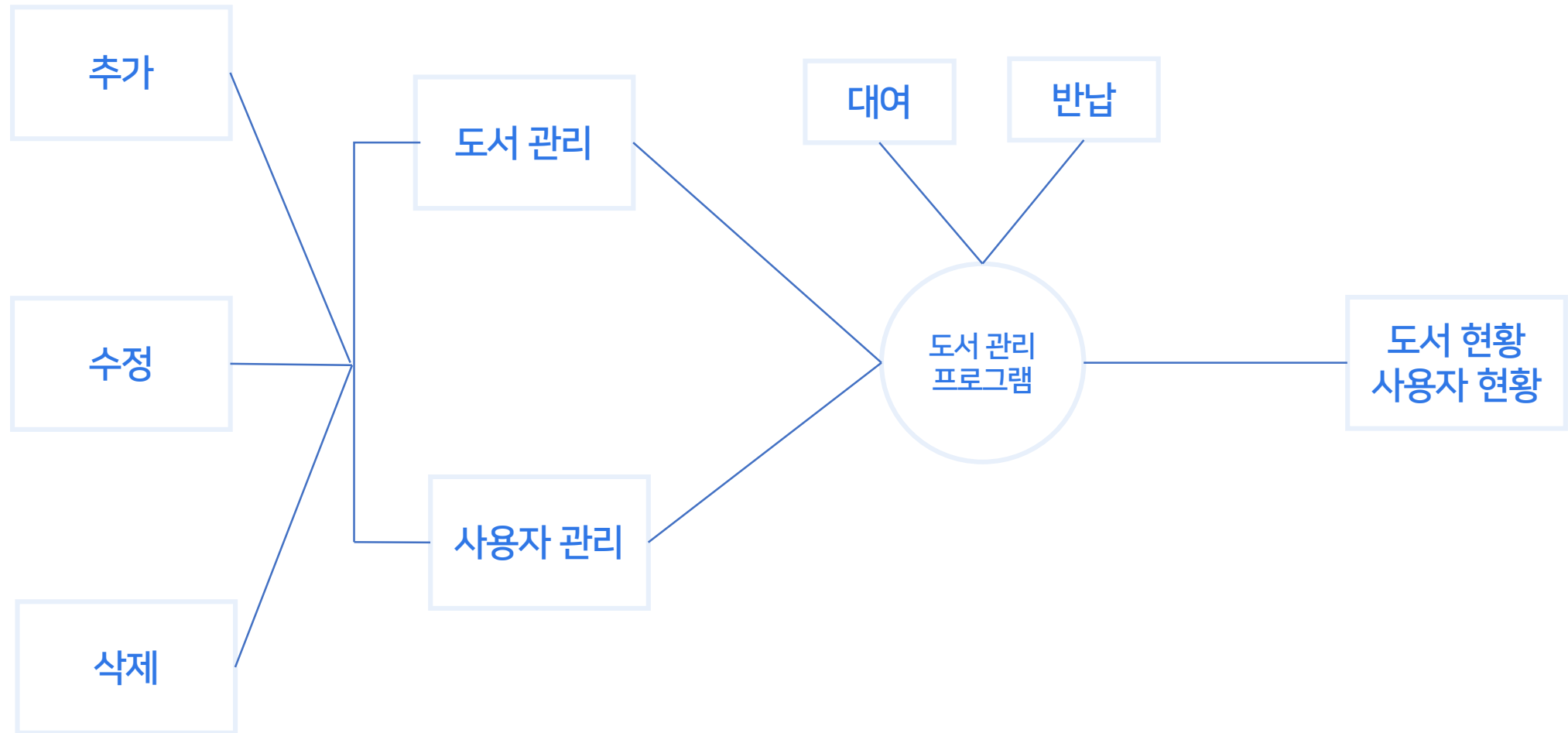
및 데이터 베이스 설계

4/25 ~ 26일

통합 테스트

SUN	MON	TUE	WED	THU	FRI	SAT
				1	2	3
4	5	6	7	8	9	10
11	12	13	14	15	16	17
18	19	20	21	22	23	24
25	26	27	28	29	30	

도서 관리 프로그램 C# : 기본 구조



도서 관리 프로그램의 기능

1. 도서를 대여 및 반납 할 수 있다.
2. 관리 할 도서를 등록 하고 추가, 수정, 삭제 할 수 있다.
-책 이름, 코드 번호, 출판사, 페이지 등록 가능
3. 사용자를 관리 하고 추가, 수정, 삭제 할 수 있다.
-사용자 ID, 이름 등록 가능
4. Log기록을 남겨서 사용자의 사용 기록을 확인 할 수 있다.
5. 등록된 도서의 현황을 확인 할 수 있다.
-전체 도서 수, 사용자 수, 대출 중인 도서 수, 연체 중인 도서 수
6. XML, Json, Mssql 세가지 방식으로 데이터를 저장하고
관리 할 수 있다.

도서관 관리 모드

도서 관리

사용자 관리

도서관 현황

전체 도서 수 : 0
사용자 수 : 0
대출 중인 도서의 수 : 0
연체 중인 도서의 수 : 0

대여/반납

isbn

대여

도서 이름

반납

사용자 ID

도서현황

	isbn	책이름	Publisher	Page	UserId	UserName	isBorrowed	BorrowedAt
--	------	-----	-----------	------	--------	----------	------------	------------

	Id	Name
--	----	------

도서 관리 모드

도서 관리

도서 추가/수정/삭제

isbn

도서 이름

출판사

페이지

도서현황

	isbn	Name	Publisher	Page	UserId	UserName	isBorrowed	BorrowedAt

사용자 관리 모드

사용자 관리

사용자 현황

Id	Name
----	------

사용자 추가/수정/삭제

사용자 ID

이름

추가 수정 삭제

XML 연결

```
public static void Load()
{
    try
    {
        string booksOutput = File.ReadAllText(@"./Books.xml");
        XElement booksXElement = XElement.Parse(booksOutput);
        Books = (from item in booksXElement.Descendants("book")
                select new Book()
                {
                    Isbn = item.Element("isbn").Value,
                    Name = item.Element("name").Value,
                }).ToList<Book>();

        string usersOutput = File.ReadAllText(@"./Users.xml");
        XElement usersXElement = XElement.Parse(usersOutput);
        Users = (from item in usersXElement.Descendants("user")
                select new User()
                {
                    Id = int.Parse(item.Element("id").Value),
                    Name = item.Element("name").Value,
                }).ToList<User>();
    }
    catch (Exception exception)
    {
        // 파일이 없으면 예외 발생: 새로운 파일 생성
        CreateFile();
    }
}

private static void CreateFile()
{
    //Cars.xml 파일을 만들겁니다.

    string fileName = @"./Books.xml";
    StreamWriter writer = File.CreateText(fileName); //파일이 없으면 해당 파일 생성
    writer.Dispose(); //메모리 해제
    fileName = @"./Users.xml";
    writer = File.CreateText(fileName); //파일이 없으면 해당 파일 생성
    writer.Dispose(); //메모리 해제
}
```

Book과 User Xml 데이터를 불러옵니다.
그리고 가져온 데이터를 각각의 클래스 객체를 넣어 줍니다.

Xml 파일이 없을 때 xml 파일을 만들어 줍니다.

Json 연결

```
public static void LoadJson()
{
    try
    {
        string strBookJsonValue = File.ReadAllText(@"./Books.json");
        JObject jsonObjectBook = JObject.Parse(strBookJsonValue);
        Books = (from item in jsonObjectBook["books"]["book"]
                 select new Book()
                 {
                     Isbn = item["isbn"].ToString(),
                     Name = item["name"].ToString(),
                     Publisher = item["publisher"].ToString(),
                     Page = int.Parse(item["page"].ToString()),
                     BorrowedAt = DateTime.Parse(item["borrowedat"].ToString()),
                     isBorrowed = item["isborrowed"].ToString() != "0" ? true : false,
                     UserId = int.Parse(item["userid"].ToString()),
                     UserName = item["username"].ToString().Replace("{", "").Replace("}", "")
                 }).ToList<Book>();

        string strUserJsonValue = File.ReadAllText(@"./Users.json");
        JObject jsonObjectUser = JObject.Parse(strUserJsonValue);
        Users = (from item in jsonObjectUser["users"]["user"]
                 select new User()
                 {
                     Id = int.Parse(item["id"].ToString()),
                     Name = item["name"].ToString()
                 }).ToList<User>();
    }
    catch (Exception e)
    {
        MessageBox.Show("파일이 누락되었습니다!!!");
        CreateFile();
        SaveJson();
        LoadJson(); //다시 불러들이기
    }
}
```

Book과 User Json 데이터를 불러옵니다.
그리고 가져온 데이터를 유저 클래스 객체를 넣어 줍니다.

Json 파일이 없을 때 Json 파일을 만들어 줍니다.

```
public static void selectQuery_User(int Id = -1)
{
    ConnectDB();
    SqlCommand cmd = new SqlCommand();
    cmd.Connection = conn;
    if (Id < 0)
    {
        cmd.CommandText = "SELECT * FROM bookuser";
        SqlDataAdapter da = new SqlDataAdapter(cmd);
        ds = new DataSet();
        da.Fill(ds, "bookuser");
        dt_user = ds.Tables[0];
    }
}
```

SQL 명령어 선언

DataAdapter와 DataSet으로 DB table 불러오기
DataSet은 메모리상의 하나의 DB 객체이며
DataTable은 메모리상의 하나의 테이블 객체

도서 관리 프로그램 C# : Winform

```
public static void printlog(string contents, string name = "History")
{
    DirectoryInfo di = new DirectoryInfo("History");
    if (!di.Exists)
    {
        di.Create();//메모장 파일 생성
    }

    using (StreamWriter writer = new StreamWriter(@"History\" + name + ".txt", true))
    {
        writer.WriteLine(contents);//메모장 log 입력
    }
}

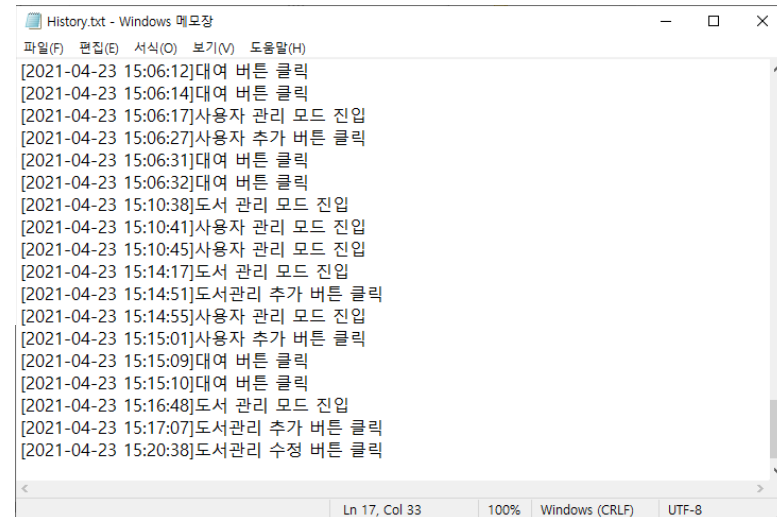
참조 10개
public static void writelog(string contents, ListBox lb)
{
    string logContents = $"[{DateTime.Now.ToString("yyyy/MM/dd HH:mm:ss")}] {contents}";

    lb.Items.Insert(0, logContents);
    DataManager.printlog(logContents);//listbox에 log 입력
}
```

Log 파일을 생성해주는 코드입니다.
Log에 클릭한 버튼의 내용과 입력 시간을 호출해주는 함수입니다.



```
[2021-04-23 11:34:36]반납 버튼 클릭
[2021-04-23 11:34:34]반납 버튼 클릭
[2021-04-23 11:34:29]대여 버튼 클릭
[2021-04-23 11:34:27]대여 버튼 클릭
[2021-04-23 11:34:16]도서 관리 모드 진입
[2021-04-23 11:34:08]도서 관리 모드 진입
[2021-04-23 11:25:49]도서 관리 모드 진입
```



```
try
{
    Book book = DataManager.Books.Single((x) => x.Isbn == textBox1.Text);
    if (book.isBorrowed)
    {
        MessageBox.Show("이미 대여 중인 도서입니다.");
    }
    else
    {
        User user = DataManager.Users.Single((x) => x.Id.ToString() == textBox3.Text);
        book.UserId = user.Id;
        book.UserName = user.Name;
        book.isBorrowed = true;
        book.BorrowedAt = DateTime.Now;

        dataGridView1.DataSource = null;
        dataGridView1.DataSource = DataManager.Books;
        DataManager.Save();

        MessageBox.Show "\"" + book.Name + "\"이/가 \"" + user.Name + "\"님께 대여되었습니다.");
    }
}
catch (Exception exception)
{
    MessageBox.Show("존재하지 않는 도서 또는 사용자입니다.");
}

if (book.isBorrowed)
{
    User user = DataManager.Users.Single((x) => x.Id.ToString() == book.UserId.ToString());
    book.UserId = 0;
    book.UserName = "";
    book.isBorrowed = false;
    book.BorrowedAt = new DateTime();

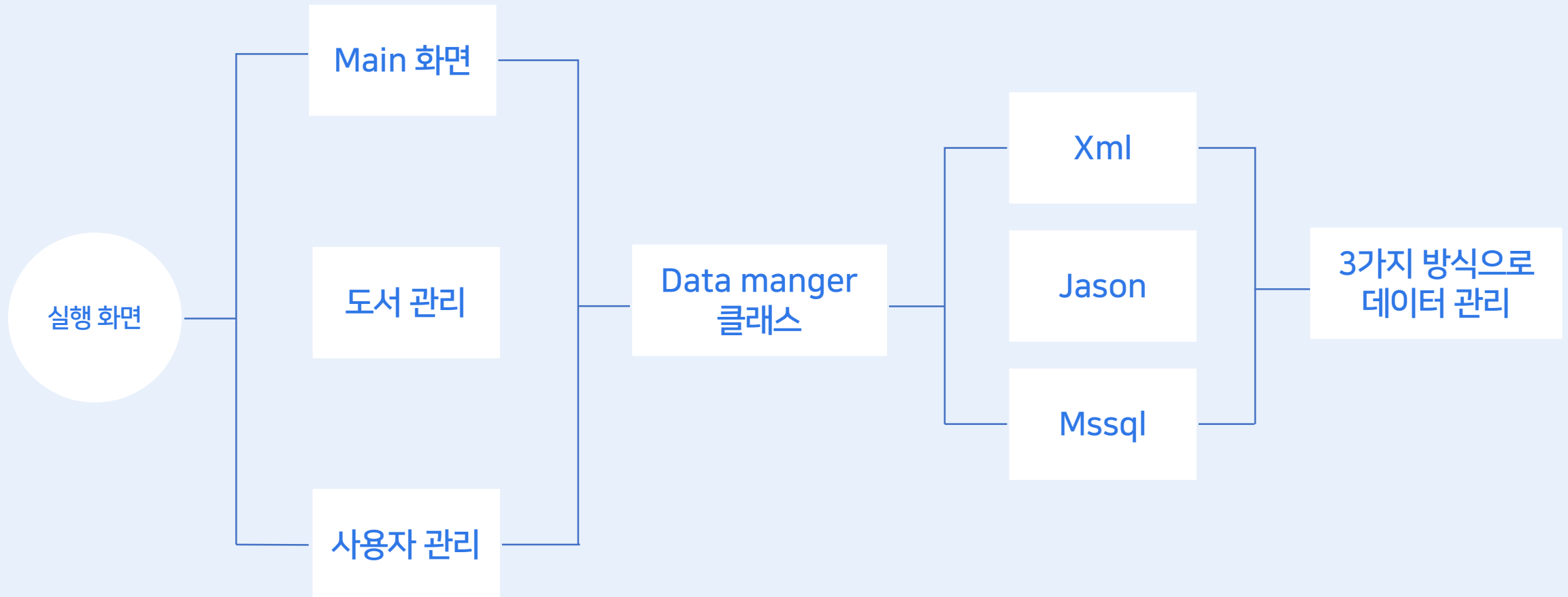
    dataGridView1.DataSource = null;
    dataGridView1.DataSource = DataManager.Books;
    DataManager.Save();
}
```

Main Form 에서 대여/반납 버튼 클릭

Borrowed가 체크 되어 있을 때는
"대여 중 입니다"라는 Message box를 띄우고, 체크가 안되
어 있을 시에는
User data를 Book table의 User data쪽에 넣어 줍니다.

반납 시에는 초기 값을 넣어 주는 방식으로 되어 있습니다.

도서 관리 프로그램 C# : 흐름도



프로그램 실행 결과

도서관 관리

도서 관리 사용자 관리

도서관 현황

전체 도서 수 : 5
사용자 수 : 2
대출 중인 도서의 수 : 0
연체 중인 도서의 수 : 0

대여/반납

Isbn 9791165214432 대여 반납

도서 이름 2021 시나공 정보 사용자 ID 0

도서현황

Isbn	책이름	Publisher	Page	UserId	UserName	isBorrowed	BorrowedAt
9791165214432	2021 시나공 정보	길벗	1044	0		<input type="checkbox"/>	
9791156645238	C# 프로그래밍	한빛아카데미	480	0		<input type="checkbox"/>	
9791158392475	시작하세요! C...	위키북스	510	0		<input type="checkbox"/>	
9791162243770	이것이 C#이다	한빛미디어	856	0		<input type="checkbox"/>	
9791165211615	C# 교과서	길벗	844	0		<input type="checkbox"/>	

Id Name

1	하미안
2	김소현

도서관 관리

도서 관리 사용자 관리

도서관 현황

전체 도서 수 : 5
사용자 수 : 2
대출 중인 도서의 수 : 0
연체 중인 도서의 수 : 0

대여/반납

Isbn 9791165214432 대여 반납

도서 이름 2021 시나공 정보 사용자 ID 1

도서현황

Isbn	책이름	Publisher	Page	UserId	UserName	isBorrowed	BorrowedAt
9791165214432	2021 시나공 정보	길벗	1044	1	하미안	<input checked="" type="checkbox"/>	2021-04-21 오...
9791156645238	C# 프로그래밍	한빛아카데미	480	0		<input type="checkbox"/>	
9791158392475	시작하세요! C...	위키북스	510	0		<input type="checkbox"/>	
9791162243770	이것이 C#이다	한빛미디어	856	0		<input type="checkbox"/>	
9791165211615	C# 교과서	길벗	844	0		<input type="checkbox"/>	

Id Name

1	하미안
2	김소현

[2021-04-21 11:13:07]대여 버튼 클릭

2021 시나공 정보처리기사 물기 "이/가 "하미안"님께 대여되었습니다.

확인

프로그램 실행 결과

도서 관리

도서 추가/수정/삭제

Isbn 추가

도서 이름 수정

출판사 삭제

페이지

도서현황

	Isbn	Name	Publisher	Page	UserId	UserName	isBorrowed	BorrowedAt
	9791165214432	2021 시나공 정...	길벗	1044	0		<input type="checkbox"/>	
	9791156645238	C# 프로그래밍	한빛아카데미	480	0		<input type="checkbox"/>	
	9791158392475	시작하세요! C...	위키북스	510	0		<input type="checkbox"/>	
	9791162243770	이것이 C#이다	한빛미디어	856	0		<input type="checkbox"/>	
	9791165211615	C# 교과서	길벗	844	0		<input type="checkbox"/>	

사용자 관리

사용자 현황

	Id	Name
▶	1	하미만
	2	김소현

사용자 추가/수정/삭제

사용자 ID

이름

추가 수정 삭제

도서 관리 프로그램 C#
감사합니다.

통합 응용SW개발 반 (java, C#)
김소현, 하이안