

도서관리 프로그램

이현덕

목차

- 개요
- 개발환경
- 정의서
- 화면 설계
- 유즈 케이스
- 데이터 베이스
- 기능설계 및 구현
- 기능 테스트

기능

- 회원가입
- 로그인,로그아웃,회원정보수정
- 게시판 글 목록
- 게시판 글 작성
- 게시판 첨부파일 다운로드
- 게시판 글 수정 및 삭제
- 게시판 글 답변
- 게시판 페이지 블록설정

개요

사용자, 도서 관리 페이지를 구현하고 사용자가 도서를 대여하는 페이지를 구현

Ms-sql을 통해 데이터를 관리

Button을 이용하여 구동

도서의 재고와 반납, 대여 여부를 Datagridview를 통해 실시간으로 확인 가능

리스트 박스로 반납,대여 여부를 실시간으로 기록

개발 환경

Visual Studio 2019

MSSQL Server 2012

개발 기간

2020.04.06 ~ 04.12

테이블 정의서

테이블에 대한 구체적인 내용은 다음과 같이 다룸

엔티티 타입명	도서				작성일	2020년 4월 13일		
테이블 명	BookTable				작성자	이현덕		
테이블 설명	도서의 정보를 추가, 생성 및 삭제 기능을 관리한다.							
번호	컬럼명	속성명	도메인	데이터타입	길이	Null	기본값	key
1	Isbn	번호	번호(n,50)	nvarchar	50	Not Null		primarykey
2	Name	도서명	도서명(n,50)	nvarchar	50	Not Null		
3	Publisher	저자	저자(n,50)	nvarchar	50	Not Null		
4	Page	페이지	페이지(int)	int		Not Null		
5	BorrowedAt	대여날짜	날짜(DATE)	date		Null		
6	IsBorrowed	대여여부	대여여부(bit)	bit		Not Null		
7	UserId	유저Id	아이디(int)	int		Null		
8	UserName	유저이름	이름(n,7)	nvarchar	7	Null		

테이블 정의서

테이블에 대한 구체적인 내용은 다음과 같이 다룸

엔티티 타입명		유저			작성일		2020년 4월 13일	
테이블 명		UserTable			작성자		이현덕	
테이블 설명		사용자의 정보를 추가, 생성 및 삭제 기능을 관리한다.						
번호	컬럼명	속성명	도메인	데이터타입	길이	Null	기본값	key
1	Id	아이디	아이디(int)	int		Not Null		primarykey
2	Name	이름	이름(n,7)	nvarchar	7	Not Null		

인터페이스 정의서

인터페이스에 대한 구체적인 내용은 다음과 같이 다룸

도서 추가

환경	개발환경	서비스 그룹	기능구현
서비스 명	B-M-001	작성자	이현덕
설명	추가버튼을 눌러 데이터베이스에 저장된 Isbn의 중복 값을 확인 후 값을 추가한다.		
코드	<pre>selectQuery(textBox_isbn.Text); if (DBHelper.dt_book.Rows.Count > 0) { MessageBox.Show("이미 존재하는 도서입니다."); } else { int.TryParse(textBox_page.Text, out int page); if(page ≤ 0) { MessageBox.Show("Page 값이 잘못되었습니다. 0 이상의 숫자를 입력하여 주십 return; } insertQuery(); selectQuery(); }</pre>		

인터페이스 정의서

인터페이스에 대한 구체적인 내용은 다음과 같이 다룸

도서 수정

환경	개발환경	서비스 그룹	기능구현
서비스 명	B-M-002	작성자	이현덕
설명	도서를 선택하고 isbn의 중복 값을 확인하고 수정이 가능하다.		
코드	<pre>selectQuery(textBox_isbn.Text); if (DBHelper.dt_book.Rows.Count == 0) { MessageBox.Show("존재하지 않는 도서입니다."); } else { updateQuery(); selectQuery(); }</pre>		

인터페이스 정의서

인터페이스에 대한 구체적인 내용은 다음과 같이 다룸

도서 삭제

환경	개발환경	서비스 그룹	기능구현
서비스 명	B-M-003	작성자	이현덕
설명	SelectQuery값에서 같은 isbn 값을 찾아 해당 데이터를 삭제한다.		
코드	<pre>selectQuery(textBox_isbn.Text); if (DBHelper.dt_book.Rows.Count == 0) { MessageBox.Show("존재하지 않는 도서입니다."); } else { deleteQuery(); selectQuery(); } } catch(Exception) { MessageBox.Show("존재하지 않는 도서입니다."); }</pre>		

인터페이스 정의서

인터페이스에 대한 구체적인 내용은 다음과 같이 다룸

사용자 추가

환경	개발환경	서비스 그룹	기능구현
서비스 명	B-M-004	작성자	이현덕
설명	도서와 마찬가지로 추가버튼을 눌러 데이터베이스에 저장된 사용자ID의 중복 값을 확인 후 값을 추가한다.		
코드	<pre>int.TryParse(textBox_ID.Text, out int id); if (id ≤ 0) { MessageBox.Show("Id 값이 잘못되었습니다. 0 이상의 숫자를 입력하여 주십시오."); return; } //해당 id 데이터가 존재하는지 체크(중복유저체크) selectQuery(id); if (DBHelper.dt_user.Rows.Count > 0) { MessageBox.Show("이미 등록된 사용자입니다."); } else { insertQuery(); selectQuery(); }</pre>		

인터페이스 정의서

인터페이스에 대한 구체적인 내용은 다음과 같이 다룸

사용자 수정

환경	개발환경	서비스 그룹	기능구현
서비스 명	B-M-005	작성자	이현덕
설명	사용자를 선택하여 사용자 ID가 일치하면 사용자의 이름을 변경할 수 있다.		
코드	<pre>int.TryParse(textBox_ID.Text, out int id); if (id ≤ 0) { MessageBox.Show("Id 값이 잘못되었습니다. 0 이상의 숫자를 입력하여 주십시오."); return; } //해당 id 데이터가 존재하는지 체크(중복유저체크) selectQuery(id); if (DBHelper.dt_user.Rows.Count == 0) { MessageBox.Show("등록되지 않은 사용자입니다."); } else { //BookTable의 UserName도 UserTable 값에 맞추기 updateQuery_Book(); //설정한 이름으로 변경 updateQuery_User(); selectQuery(); }</pre>		

인터페이스 정의서

인터페이스에 대한 구체적인 내용은 다음과 같이 다룸

사용자 삭제

환경	개발환경	서비스 그룹	기능구현
서비스 명	B-M-006	작성자	이현덕
설명	사용자를 선택하고 데이터베이스에서 ID값을 불러와 일치하면 등록된 사용자 데이터를 삭제한다.		
코드	<pre>int.TryParse(textBox_ID.Text, out int id); if (id ≤ 0) { MessageBox.Show("Id 값이 잘못되었습니다. 0 이상의 숫자를 입력하여 주십시오."); return; } //해당 id 데이터가 존재하는지 체크(중복유저체크) selectQuery(id); if (DBHelper.dt_user.Rows.Count == 0) { MessageBox.Show("등록되지 않은 사용자입니다."); } else { bool isRemove = true; updateQuery_Book(isRemove); deleteQuery(); selectQuery(); }</pre>		

인터페이스 정의서

인터페이스에 대한 구체적인 내용은 다음과 같이 다룸

도서 대여

환경	개발환경	서비스 그룹	기능구현
서비스 명	B-M-007	작성자	이현덕
설명	대여를 눌러 사용자 ID값과 도서 Isbn값이 데이터베이스에서 일치하면 대여한다. isBorrowed를 bool값으로 두어 true,false 로 대여 진행상황을 구분한다.		
코드	<pre>bool isBorrowed = bool.Parse(DBHelper.dt_book.Rows[0]["isBorrowed"].ToString()); if (isBorrowed) { //2-1. true이면 이미 대여중이라고 할 것 MessageBox.Show("이미 대여 중인 도서입니다."); } else { //대출할 책 이름 가져옴 string bookName = DBHelper.dt_book.Rows[0]["Name"].ToString(); // 대출자의 id를 이용하여 대출자의 이름을 가져옴 selectQuery_User(userid); string userName = DBHelper.dt_user.Rows[0]["Name"].ToString(); bool doBorrow = true; updateQuery(userid, doBorrow); MessageBox.Show("\'" + bookName + "\"이/가\'" + userName + "\"님께 대여되었습니다."); log = \$"{DateTime.Now.ToString("yyyy/MM/dd hh:mm:ss")} {userName} 이/가 {bookName}을 대여 "; listBox1.Items.Insert(0, log); refreshStatus(); //대출 하고 나서 db table에 적혀 있는 값들을 다시 보여줌 }</pre>		

인터페이스 정의서

인터페이스에 대한 구체적인 내용은 다음과 같이 다룸

도서 반납

환경	개발환경	서비스 그룹	기능구현
서비스 명	B-M-008	작성자	이현덕
설명	대여처럼 도서를 선택 후 반납을 눌러 DB에서 isbn과 사용자ID값이 일치하는지 확인 후 isBorrowed의 값으로 대여 여부를 판별한다. 그리고 대여중인 날짜가 저장되어있는 데이터값을 현재날짜 Datetime.today에서 빼준뒤 7일이 넘으면 연체, 그렇지 않으면 반납 이라고 표시		
코드	<pre>borrowCount = 0; foreach (DataRow item in DBHelper.dt_book.Rows) { if (bool.Parse(item["isBorrowed"].ToString())) { DateTime oldDay = DateTime.Parse(item["BorrowedAt"].ToString()); TimeSpan timeDiff = DateTime.Today - oldDay; int diffDays = timeDiff.Days; if (diffDays > 7) { borrowCount++; } } }</pre>		

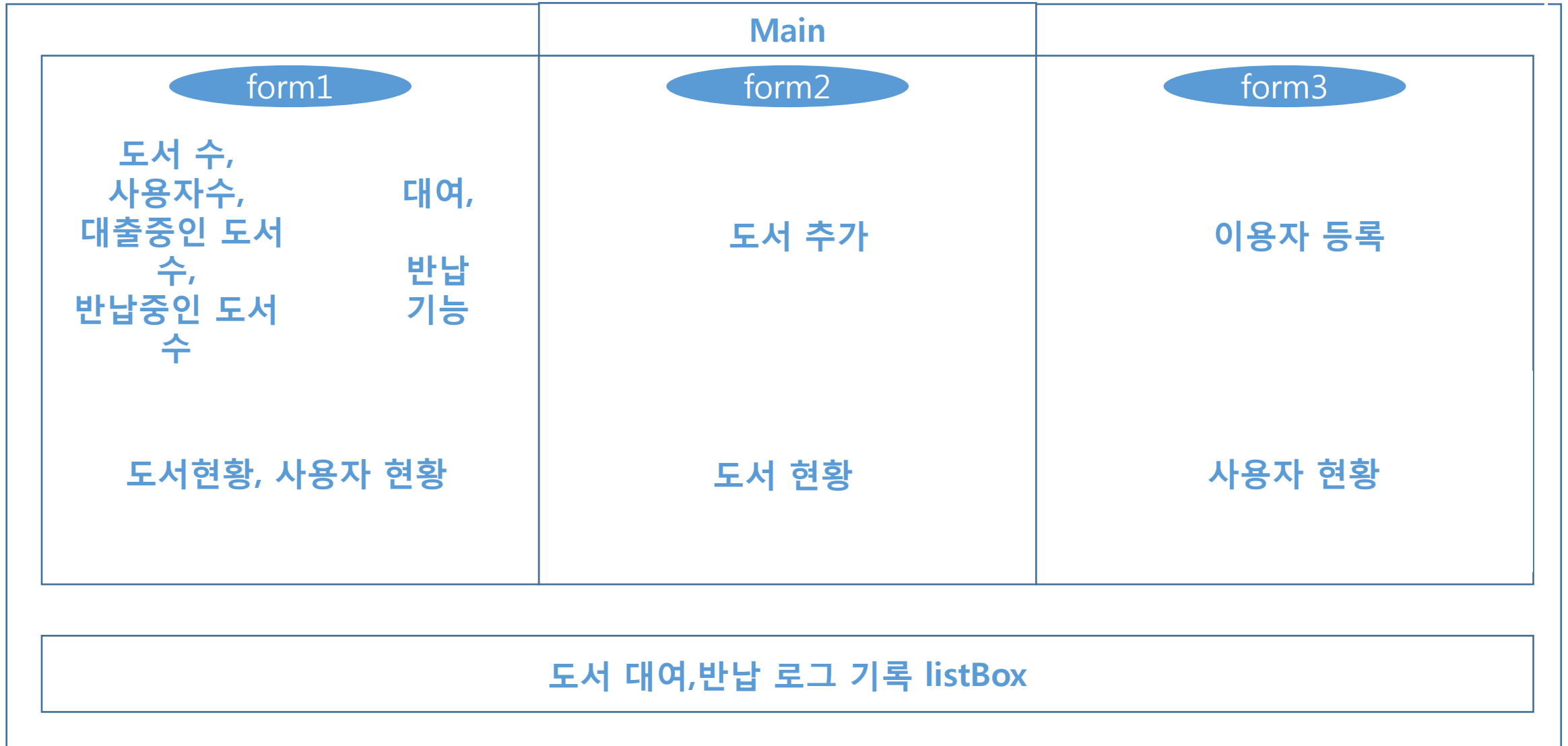
코드

```
string bookName = DBHelper.dt_book.Rows[0]["Name"].ToString();
if (isBorrowed)
{
    DateTime oldDay = DateTime.Parse(DBHelper.dt_book.Rows[0]["BorrowedAt"].ToString());
    updateQuery();

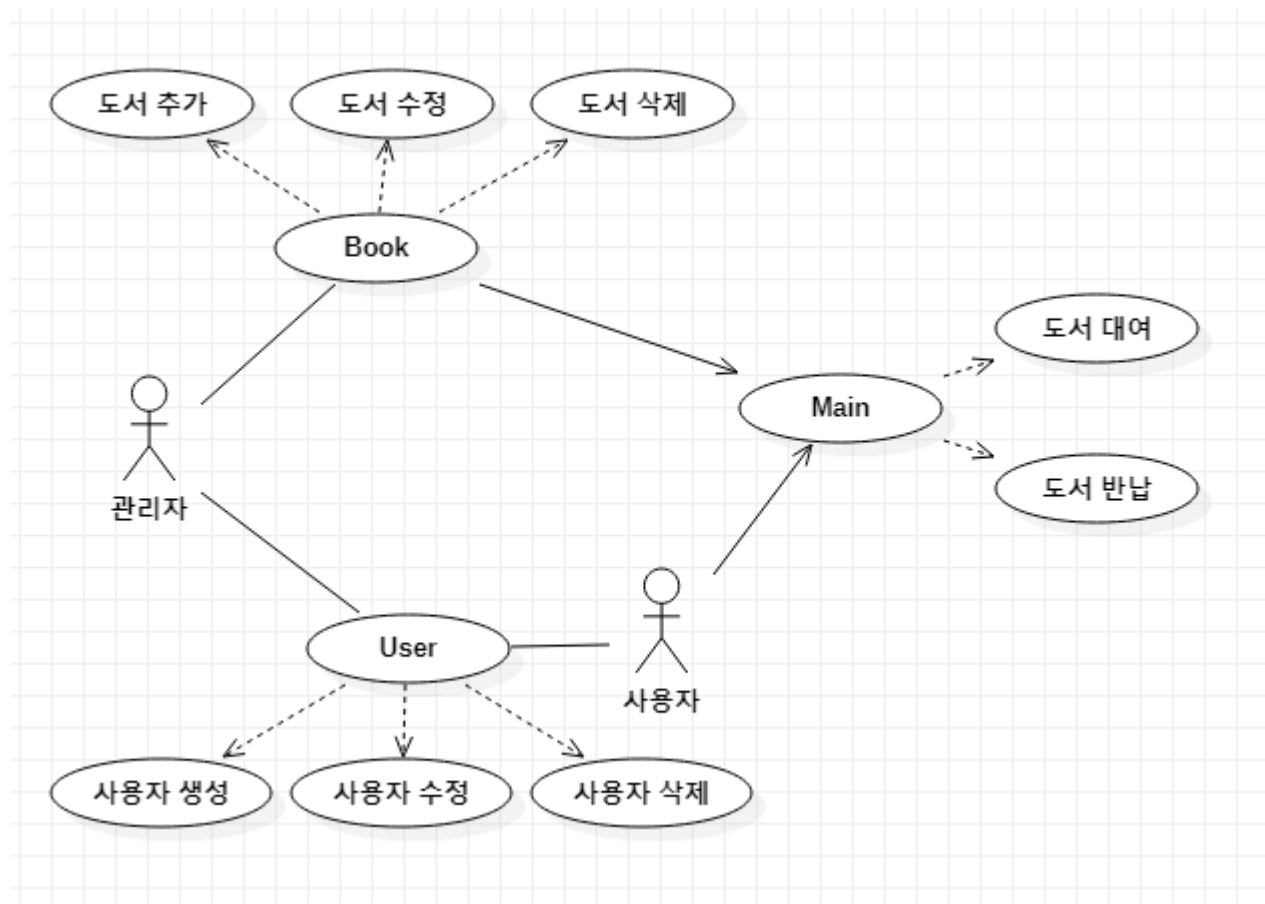
    TimeSpan timeDiff = DateTime.Today - oldDay;
    int diffDays = timeDiff.Days;

    if (diffDays > 7)
    {
        MessageBox.Show("\"" + bookName + "\"이/가 연체 상태로 반납되었습니다.");
    }
    else
    {
        MessageBox.Show("\"" + bookName + "\"이/가 반납되었습니다.");
        log = $"[{DateTime.Now.ToString("yyyy/MM/dd hh:mm:ss")}] {bookName}을 반납했습니다. ";
        listBox1.Items.Insert(0, log);
    }
    refreshStatus();
}
```


화면 설계





유즈 케이스



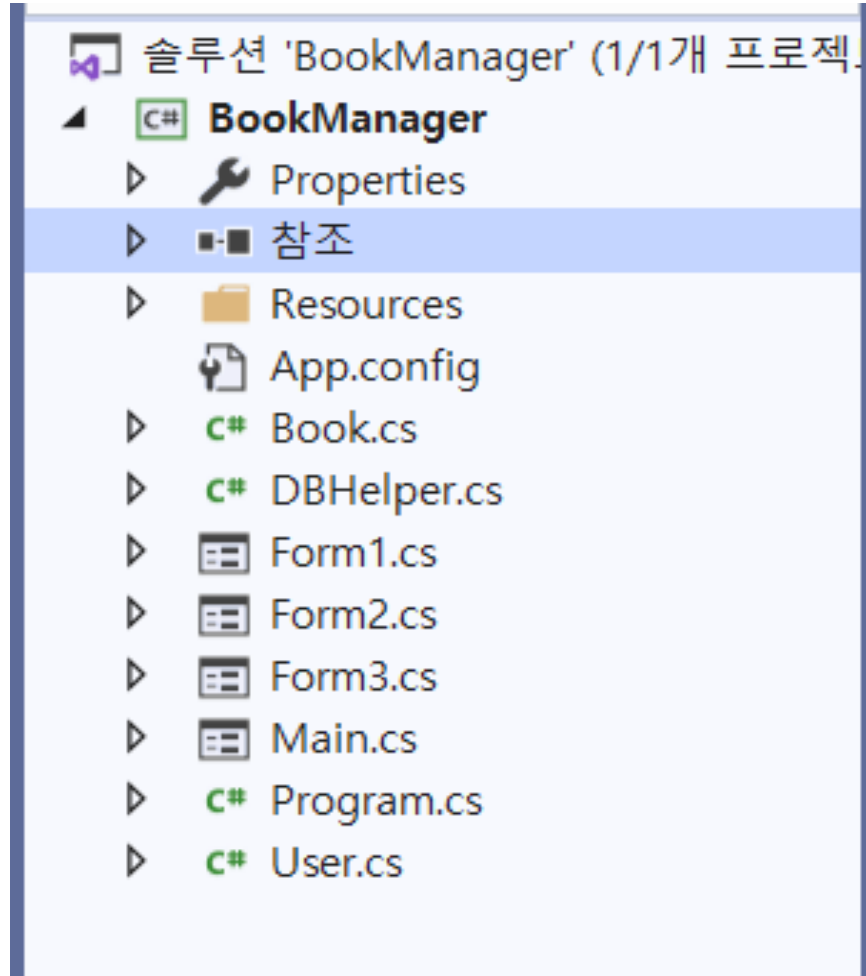
데이터 베이스

Ms-sql 테이블

	열 이름	데이터 형식	Null 허용
	Isbn	nvarchar(50)	<input type="checkbox"/>
	Name	nvarchar(50)	<input type="checkbox"/>
	Publisher	nvarchar(50)	<input type="checkbox"/>
	Page	int	<input type="checkbox"/>
	BorrowedAt	date	<input checked="" type="checkbox"/>
	IsBorrowed	bit	<input type="checkbox"/>
	UserId	int	<input checked="" type="checkbox"/>
	UserName	nvarchar(7)	<input checked="" type="checkbox"/>
			<input type="checkbox"/>

	열 이름	데이터 형식	Null 허용
	Id	int	<input type="checkbox"/>
	Name	nvarchar(7)	<input type="checkbox"/>
			<input type="checkbox"/>

기능 설계



```
class DBHelper
```

```
{
```

```
    public static SqlConnection conn = new SqlConnection();
```

```
    public static DataSet ds;
```

```
    public static DataTable dt_book;
```

```
    public static DataTable dt_user;
```

참조 11개

```
    public static void ConnectDB()
```

```
    {
```

```
        conn.ConnectionString = string.Format("Data Source={0}); " +
```

```
            "Initial Catalog = {1};" +
```

```
            "Integrated Security = {2};" +
```

```
            "Timeout = 3"
```

```
            , "local", "MYDB1", "SSPI");
```

```
        conn = new SqlConnection(conn.ConnectionString);
```

```
        conn.Open();
```

```
    }
```

```

public static void selectQuery_Book(string Isbn = "")
{
    ConnectDB();

    SqlCommand cmd = new SqlCommand();
    cmd.Connection = conn;
    if (Isbn == "")
    {
        cmd.CommandText = "SELECT * FROM BookTable";

        SqlDataAdapter da = new SqlDataAdapter(cmd);
        ds = new DataSet();
        da.Fill(ds, "BookTable");

        dt_book = ds.Tables[0];
    }
    else
    {
        cmd.CommandText = "SELECT * FROM BookTable Where Isbn = " + Isbn;
        SqlDataAdapter da = new SqlDataAdapter(cmd);
        ds = new DataSet();
        da.Fill(ds, "BookTable");

        dt_book = ds.Tables[0];
    }

    conn.Close();
}

```

참조 4개

```

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

        dt_user = ds.Tables[0];
    }
    else
    {
        cmd.CommandText = "SELECT * FROM UserTable Where Id = " + Id;
        SqlDataAdapter da = new SqlDataAdapter(cmd);
        ds = new DataSet();
        da.Fill(ds, "UserTable");

        dt_user = ds.Tables[0];
    }

    conn.Close();
}

```

DB헬퍼 클래스를 만들어 DB연결을 한 후에 book, user테이블 모두 DataAdapter과 DataSet으로 테이블 값들을 불러와서 데이터테이블에 값을 넣어준다.

```

public static void updateQuery(string isbn, int userId = 0, bool doBorrow = false)
{
    try
    {
        ConnectDB();
        SqlCommand cmd = new SqlCommand();
        cmd.Connection = conn;
        cmd.CommandType = CommandType.Text;

        string sqlcommand;
        if (doBorrow) //대출
        {
            string userName = dt_user.Rows[0]["Name"].ToString();
            sqlcommand = "Update BookTable set UserId = @p1, UserName = @p2, isBorrowed = 1, BorrowedAt = @p3 where Isbn = @p4";

            cmd.Parameters.AddWithValue("@p1", userId);
            cmd.Parameters.AddWithValue("@p2", userName);
            cmd.Parameters.AddWithValue("@p3", DateTime.Now.ToString("yyyy-MM-dd"));
            cmd.Parameters.AddWithValue("@p4", isbn);
        }
        else //반납
        {
            sqlcommand = "Update BookTable set UserId = null, UserName = null, isBorrowed = 0, BorrowedAt = null where Isbn = @p1";
            cmd.Parameters.AddWithValue("@p1", isbn);
        }
        cmd.CommandText = sqlcommand;
        cmd.ExecuteNonQuery(); //쿼리 실행
        conn.Close();
    }
    catch (Exception eee)
    {
    }
}

```

User Id를 이용하여 User 값들을 가지고 온다.

기능 테스트

BookManagerProgram

도서 추가/수정/삭제

isbn 1
도서 이름 html5,css북프로그래밍
출판사 한빛 아카데미
페이지 456

추가

수정

삭제

도서 현황

	isbn	Name	Publisher	Page	UserId	isBorrowed	Borro
▶	1	html5,css북프...	한빛 아카데미	456		<input type="checkbox"/>	
	2	java programing	한빛 아카데미	322		<input type="checkbox"/>	

BookManagerProgram

도서 추가/수정/삭제

isbn 1
도서 이름 html5,css북프로그래밍
출판사 한빛 아카데미
페이지 456

추가

수정

삭제

도서 현황

	isbn	Name	Publisher	Page	UserId	isBorrowed	Borro
▶	1	html5,css북프...	한빛 아카데미	456		<input type="checkbox"/>	

책을 등록 및 삭제

기능 테스트

BookManagerProgram

사용자 현황

	Id	Name
▶	1	이현덕
	2	응소기

사용자 추가/수정/삭제

사용자 ID

이름

추가

수정

삭제

BookManagerProgram

사용자 현황

	Id	Name
▶	1	이현덕2

사용자 추가/수정/삭제

사용자 ID

이름

추가

수정

삭제

유저 이름 등록 및 수정 및 삭제

기능 테스트

BookManagerProgram

도서관 현황

전체 도서 수: 1
사용자 수: 2
대출 중인 도서의 수: 0
연체 중인 도서의 수: 0

대여/반납

Isbn: 1
도서 이름: html5,css북프로그래밍
사용자 ID: 1

대여
반납

도서 현황

Isbn	Name	Publisher	Page	BorrowedAt	IsBorrowed	UserId
1	html5,css북프...	한빛 아카데미	456		<input checked="" type="checkbox"/>	1

사용자 현황

Id	Name
1	이현덕
2	윤소기

"html5,css북프로그래밍"이/가 "이현덕"님께 대여되었습니다.

확인

도서 현황

Isbn	Name	Publisher	Page	BorrowedAt	IsBorrowed	UserId
1	html5,css북프...	한빛 아카데미	456	2020-07-08	<input checked="" type="checkbox"/>	1

특정 사용자가 책을 대여

기능 테스트

BookManagerProgram

도서판 현황
현재 도서 수: 1
사용자 수: 2
대출 중인 도서의 수: 1
연체 중인 도서의 수: 0

대여/반납
Isbn: 1
도서 이름: html5,css북프로그래밍
사용자 ID: 1
대여
반납

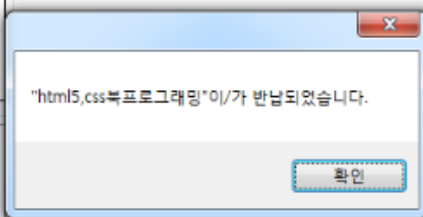
도서 현황

	Isbn	Name	Publisher	Page	BorrowedAt	IsBorrowed	UserId
▶	1	html5,css북프...	한빛 아카데미	456	2020-07-08	<input checked="" type="checkbox"/>	

사용자 현황

	Id	Name
▶	1	이현덕
	2	응소기

[2020-07-08 07:23:39] 이현덕 이/가 html5,css북프로그래밍을 대여



도서 현황

	Isbn	Name	Publisher	Page	BorrowedAt	IsBorrowed	UserId
▶	1	html5,css북프...	한빛 아카데미	456		<input type="checkbox"/>	

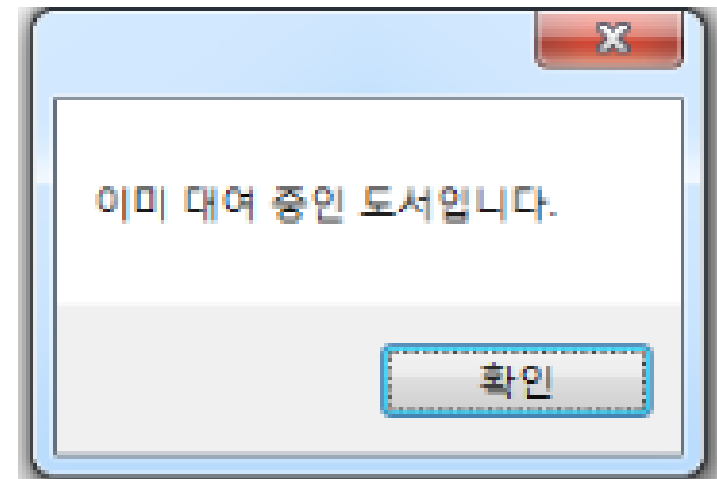
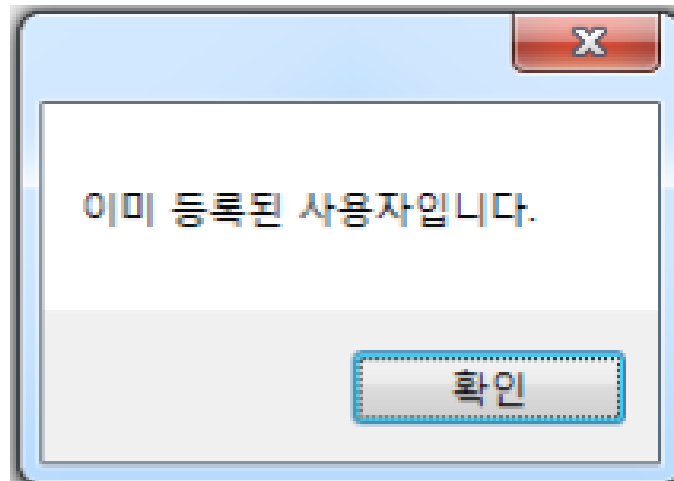
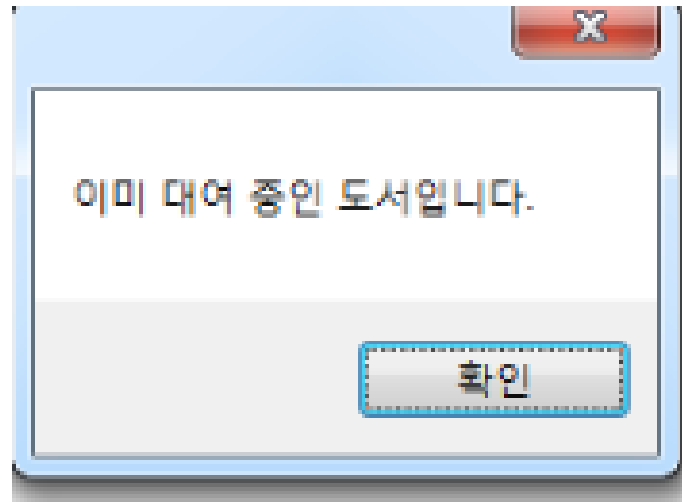
책을 대여한 사용자가 다시 반납

기능 테스트

[2020-07-08 07:25:07] html5,css북프로그래밍을 반납했습니다.
[2020-07-08 07:23:39] 이현덕 이/가 html5,css북프로그래밍을 대여

도서 대여 및 반납 실시간 현황 로그남기기

기능 테스트



도서나 유저 등록, 도서 대여, 반납 중복일 경우 안내문구

감사합니다.

H.P : 010 8607 9138

Mail : tig02230@naver.com