

# 객체 지향 설계 프로젝트

Team 4

구성인원 : 박병주(122482)

황세웅(122466)

김상수(120100)

김가영(142661)

## 목표 어플리케이션 이름

: 전컴 도서관

## 개발동기

: 도서관을 방문 할 때에 도서관 중앙에 자신이 원하는 책을 찾을 수 있도록 컴퓨터가 대부분 놓여있습니다. 대부분의 도서관이 운영하는 컴퓨터에는 단순히 검색하는 기능으로 그 책의 위치만을 알 수 있게끔 되어있습니다. 저희 4조는 대부분 책을 좋아하는 학생들로, 도서관의 역할을 더욱 확대하고 책을 읽으려는 사용자들과 소통을 할 수 있는 도서 관리 시스템을 구축하기로 했습니다. 도서관을 이용하면서 불만사항이나 요구사항들을 들음으로써 서로가 발전할 수 있는 도서관을 만들려고 합니다.

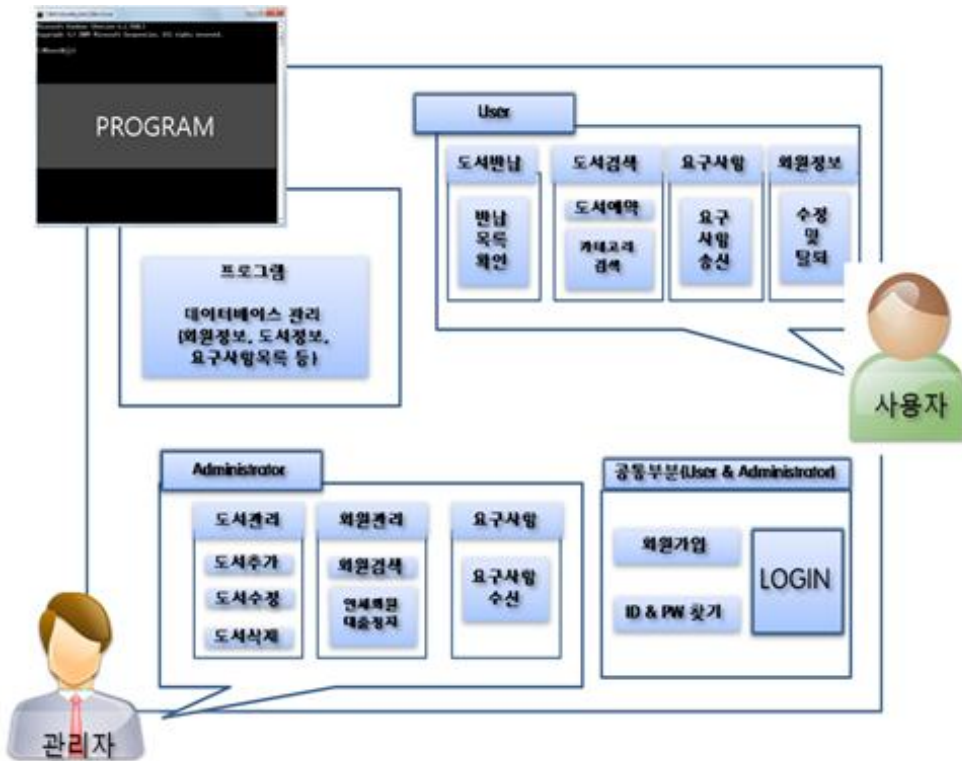
## 제공할 기능

로그인화면	로그인	아이디/패스워드 입력 후 사용자 화면 전환
	아이디/비번 찾기	분실한 아이디 및 패스워드 찾기
	회원가입	회원가입
관리자화면	도서관리	도서 대출/반납 시스템 운용
	회원관리	회원 정보 확인
	요구사항	회원들의 요구사항을 확인
회원화면	도서검색	도서예약
		도서목록확인
		카테고리 별 검색
		리뷰 확인 및 작성
		대출 여부 확인
		추천도서
	회원정보	회원정보 수정 및 탈퇴
	요구사항	개선사항/불만사항 입력

## 요구사항 리스트

- 1.로그인 실패 시 오류문구를 출력한다.
- 2.회원가입 시 아이디 중복여부를 확인한다.
- 3.회원가입 시 비밀번호 힌트문항을 설정한다.
- 4.분실한 아이디를 찾을 때 없으면 오류문구를 출력한다.
- 5.회원은 도서 대출 목록 및 반납 기한을 확인한다.
- 6.회원은 도서 대출예약이 가능하다.
- 7.회원은 도서를 카테고리 형식(문자열, 장르별)으로 쉽게 볼 수 있다.
- 8.회원은 자신의 정보를 확인 및 수정이 가능하다.
- 9.회원은 자신의 불편사항이나 요구사항을 관리자에게 문의한다.
- 10.회원은 자신이 찾는 도서에 관해 리뷰를 작성한다.
- 11.관리자는 회원들의 독서량과 인기도서 확인이 가능하다.
- 12.관리자는 연체회원을 관리한다.
- 13.관리자는 회원들의 요구사항을 수신한다.

## 유즈케이스 다이어그램



## 세부 유즈케이스

### 로그인창

#### 1. 회원가입

1. 메인화면에서 회원가입 버튼을 누른다.
2. 아이디, 패스워드, 이름, 성별, 나이, 주소, 연락처, 비밀번호 분실 시 힌트문항 입력  
(아이디는 중복확인 버튼을 추가한다. 가입 시 아이디 중복을 확인한다.)
3. 가입 버튼을 누른다.
4. 회원가입 완료 알림 팝업창을 띄운다.

#### 2. 아이디 및 패스워드 찾기

1. 아이디 찾기
  1. 아이디 찾기 버튼을 누른다.
  2. 이름을 입력한 후, 확인 버튼을 누른다.
  3. 이름과 일치하는 회원의 아이디를 출력한다.
2. 패스워드 찾기
  1. 패스워드 찾기 버튼을 누른다.
  2. 이름과 아이디, 힌트문항을 선택하고 답변을 입력한 뒤에 확인버튼을 누른다.
  3. 이름과 아이디, 힌트문항이 일치하면 패스워드를 출력한다. 일치하지 않을 시에는 '일치하는 결과가 없습니다.'라고 출력한다.

#### 3. 관리자 계정 로그인

1. 관리자 아이디 비밀번호 입력
2. 로그인 버튼을 누른다.

### 관리자 use case

#### 1. 회원관리

1. 회원관리 버튼 클릭-> 회원의 수, 회원 목록을 팝업창으로 출력한다.
2. 연체 회원 관리 버튼을 클릭한다-> 연체회원목록을 출력.  
이 때 탭으로 연체일, 연체 횟 수 또한 같이 출력한다.  
(회원을 누르면 팝업창이 뜨고,  
팝업창 내에 대출 정지 일수를 입력한 뒤, 정지버튼을 누른다.)  
->그 회원의 대출이 제한된다.
3. 버튼없이 회원들의 독서량 순위 또는 인기 도서를 확인 할 수 있다.(화면에 바로 출력)

#### 2. 도서관리

1. 도서목록 버튼을 누른다-> 도서목록을 확인한다.  
2. 도서목록 버튼을 누른다-> 도서목록을 확인 한다 ->  
<도서추가>버튼을 누른다->  
팝업창이 뜬다-> 도서의 기본 정보(저자, 출판사, 개정판 수 등) 입력이 완료되면 저장버튼을 누른다.  
->저장 버튼을 클릭->새로운 도서 추가를 완료한다.

3. 수정할 도서를 검색란에 검색->키워드를 입력-> 키워드와 일치하는 도서가 있을 시 해당 도서목록에서 해당 도서를 선택 / 도서가 없을 시 일치하는 도서가 없습니다. 팝업창 출력->해당 도서 있을 시 팝업창이 뜬다-> 도서의 기본 정보 출력-> 수정이 필요한 항목의 내용을 수정한다-> 수정버튼<도서수정>클릭->도서정보수정을 완료한다.

4. 제거할 도서를 검색란에 검색->키워드를 입력-> 키워드와 일치하는 도서가 있을 시 해당 도서목록에서 해당 도서를 선택 / 도서가 없을 시 일치하는 도서가 없습니다. 팝업창 출력->제거하려는 도서를 선택->도서의 기본 정보 출력-> 제거버튼을 클릭-><도서제거> 도서를 제거한다.

**3. 요구사항** : 메인화면에서 수신리스트 버튼 클릭-> 수신리스트 팝업 창 출력-> 수신리스트에서 요구사항을 확인한다.( 수신리스트에서 요구사항 목록이 출력되면 해당 요구사항을 클릭할 시 내용을 출력.

## 사용자 use case

### 0. 메인화면

신간도서

도서를 관리자가 새로 등록한 순서 기준으로 신간도서를 보여준다.  
(메인화면의 한 공간에 따로 출력)

### 1. 도서검색

1. 메인화면의 도서검색란에 도서명(키워드)를 검색한다.  
(키워드에 일치하는 도서가 있으면 관련도서 목록을 표기하고, 일치하는 도서가 없을 시 일치하는 도서가 없다고 출력한다.)  
목록을 표기할 때, 책이름, 작가, 출판사를 출력한다.
2. 도서 대출예약  
표시된 관련 도서목록 중 자신이 원하는 도서 클릭  
팝업창이 뜨면서 도서 정보(책표지, 책이름, 작가, 출판사)를 출력한다.  
아래에 <대출예약> 버튼을 추가한다.  
<대출예약을 하시겠습니까?>  
<예->대출예약->완료 / 아니오-> 클릭 전 화면으로 되돌아가기>

### 3. 도서 반납.

그 페이지 자체에 사용자 회원이 빌린 도서 목록을 표기, 이때 사용자가 반납할 때 필요한 제목과 대출기한을 표기한다.

### 4. 카테고리를 통한 검색

<카테고리별 검색> 버튼을 누른다.

<<카테고리별 검색에 따른 팝업창 제시>>

카테고리 목록을 문자열 또는 장르별로 나타내어 준다.

(예: 대분류- 에세이/경제/문학/과학/자연)

자신이 원하는 카테고리를 더블클릭으로 선택한다.

해당 카테고리에 맞는 책을 출력해준다.

출력된 책들을 많이 대여한 순으로 정렬하여 사용자에게 보여준다.(화면에 그대로 출력)

## 2. 회원정보

우측상단에 사용자에게 대한 아이디와 환영 메시지 출력

\*\*\*\*\*님 환영합니다!

그 아래 부분에 회원정보수정 버튼 삽입.

<회원정보수정>버튼 클릭-> 팝업창 출력

-> 기존 회원가입 시 입력했던 정보란을 표시

1. 회원정보 수정 : 회원가입시 입력했던 정보에서 원하는 정보를 수정하고

<저장>버튼을 누른다.(수정완료)

2. 회원탈퇴 : <저장>버튼 옆 <탈퇴>버튼 삽입.

<탈퇴> 버튼을 누르면 팝업창을 출력한다.

<정말 탈퇴하시겠습니까?>

예-> 탈퇴가 완료되었습니다,

아니오-> 팝업창 꺼진 후 원래 화면으로 되돌아감.

## 3. 요구사항

<요구사항>버튼을 누른다.

버튼을 누르면 요구사항을 입력할 수 있는 팝업창이 뜨고

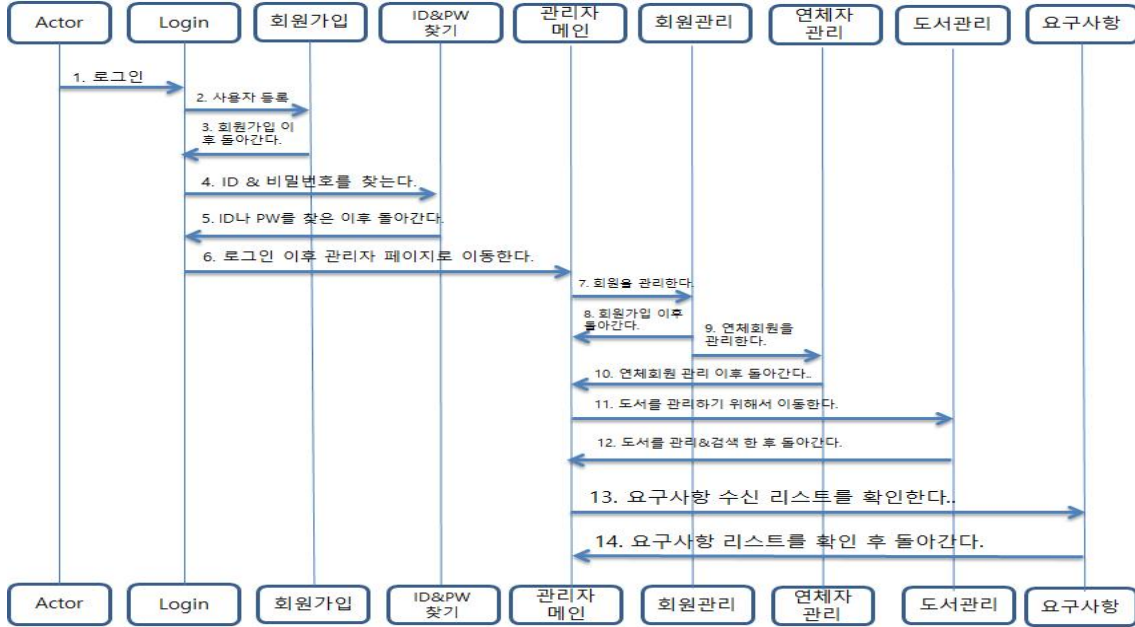
제목과 내용을 입력할 수 있는 공란을 만든다.

공란에 제목과 내용을 각각 입력한 후 보내기 버튼을 누른다.

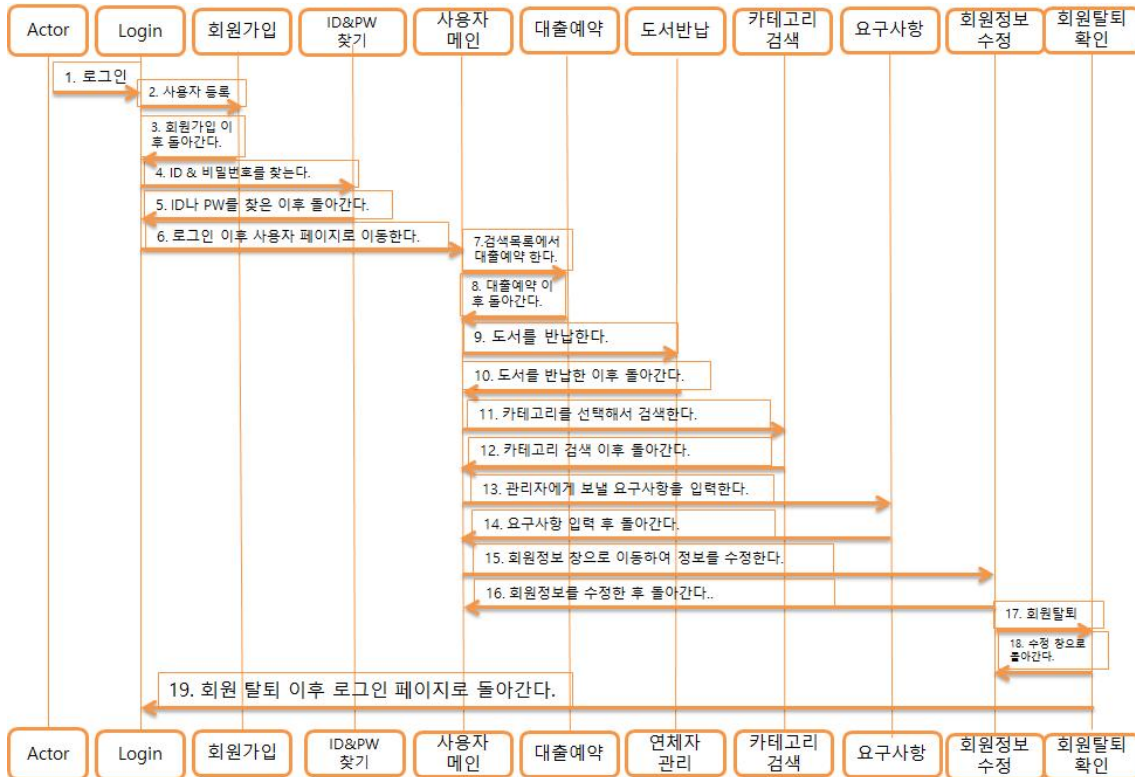
(관리자 페이지에서 리드가능)

# 시퀀스 다이어그램

## 1. 관리자 페이지



## 2. 사용자 페이지



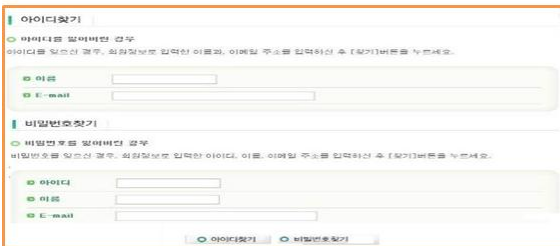
# 화면 정의서



1. 메인화면



2. 회원가입



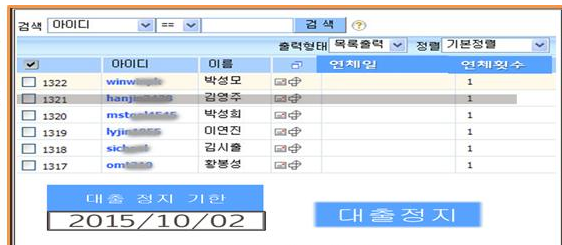
3. 아이디, 비밀번호 찾기



4. 관리자 메인페이지



5. 회원관리



6. 연체회원 관리



7. 도서관리



8. 요구사항리스트



회원 도서 관리 페이지

2015/09/24

도서반납

카테고리 검색

요구사항

도서명 검색

총 6건의 검색 결과가 있습니다.

도서명	제목	저자	출판사	대여가능
의자놀이	의자놀이	박준	문학동네	X
당신의 이름을 지어다가 먹었다	당신의 이름을 지어다가 먹었다	박준	문학동네	X
내 말에 있는 사랑	내 말에 있는 사랑	이영률	달	O
미움 받을 용기	미움 받을 용기	기시미 이치코	연봉로앤북	O
7분 공기 코포팅	7분 공기 코포팅	와카구치 에리	외즈일 북우즈	O
오베라는 남자	오베라는 남자	프렌드릭 베크만	다산책방	X

9. 회원 도서 관리 페이지

알림

대출예약을 하시겠습니까?

예 아니오

10. 대출예약 확인 페이지

도서반납

총 2건의 대여 중인 도서가 있습니다.

제목	저자	출판사	대출기한
의자놀이	박준	문학동네	15/09/27
당신의 이름을 지어다가 먹었다	박준	문학동네	15/10/02

11. 도서 반납 페이지

카테고리 별 검색

대표: 소설 소분류: 현대문학 Search

제목	저자	출판사	대여횟수
의자놀이	박준	문학동네	73
당신의 이름을 지어다가 먹었다	박준	문학동네	52
오베라는 남자	프렌드릭 베크만	다산책방	36

12. 카테고리 별 검색 페이지

요구사항/건의사항

제목 프로그램이 자주만 려이 걸려요.

프로그램이 원활하게 작동할 수 있도록 개선해 주시면 고맙겠습니다.

관리자에게 보내기

13. 요구사항 전달 페이지

회원정보관리

회원정보

ID: ema1130

비밀번호:

비밀번호 확인:

이름: 황세웅

성명:

성별:

전화번호: 010-3021-4850

이메일: ema1130@naver.com

회원탈퇴 정보수정

14. 회원정보 수정 페이지

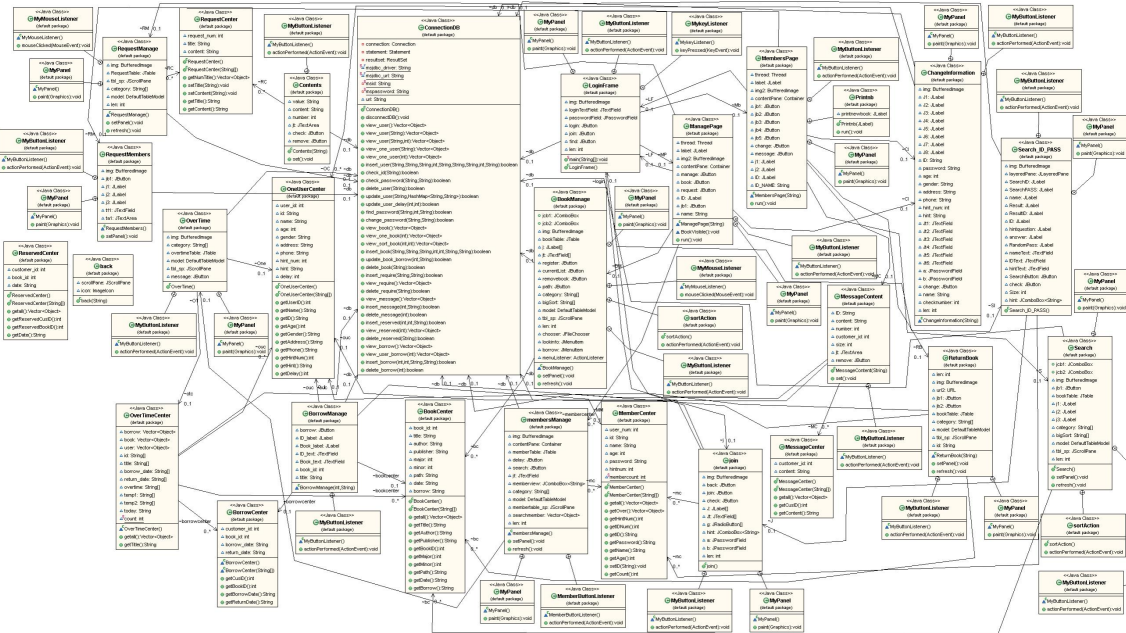
알림

정말 탈퇴하시겠습니까?

예 아니오

15. 회원탈퇴 확인 페이지

# 클래스 다이어그램



클래스 다이어그램 파일 추가

## API 매뉴얼(Javadoc 사용) 초기페이지 스크린샷 :



## 기능 구현 상황

### class : ManagePage 관리자 페이지

- 연체시간을 정확하게 저장하고 시간에 맞게 연체 여부를 확인 할 수 있도록 시계를 쓰레드로 구현하였다.(우측상단 확인 가능)
- 현재 어떤 사람들이 가입을 했는지 아이디를 확인할 수 있도록 구현하였다.
- 관리자는 크게 도서관리 회원관리 그리고 요구사항 창으로 구성하였다.
- logout을 구현하여 다른 사용자가 이용할 수 있도록 구현하였다.

**class : BookManage 도서관리 페이지**

- 책제목, 저자, 출판사를 입력한후 등록 버튼을 누르면 DB에 책의 정보가 저장되도록 했다.
- 저장 후 현재목록을 누르면 새로이 등록된 책의 정보를 확인할 수 있도록 구현했다.
- 삭제 버튼을 누르고 책 제목을 입력하면 책의 정보(DB에서)를 지울 수 있도록 구성했다.

**class : membersManage 회원관리 페이지**

- 이름, 나이, 오름차순, 내림차순으로 회원의 정보를 볼수 있게 구현했다.
- 회원의 정보는 아이디, 이름, 나이, 독서량을 알 수 있게 구현했다.
- 연체회원 버튼을 통해 지금까지 책을 반납하지 않은 회원과 기한을 확인 하려고 한다.(2차 발표 이후 구현 예정)

**class : RequestManage 요구사항 리스트**

- 2차발표 이후 구현 예정

**class : MembersPage 사용자 도서관리 페이지**

- Manage페이지와 동일하게 연체 시간을 확인하기 위해 시계를 쓰레드로 구현하였다.
- 현재 누가 가입을 하였는지 가입한 회원의 아이디와 회원들의 수를 보여줄 수 있도록 구현하였다.
- 사용자페이지는 크게 도서반납, 카테고리 검색, 회원정보수정, 요구사항 창으로 구성하였다.
- logout을 구현하여 다른 사용자가 이용할 수 있도록 구현하였다.

**class : ReturnBook 도서반납 페이지**

- 각 회원이 빌린 도서를 확인하고 반납 기한을 확인할 수 있는 창이다.
- 반납 기한 확인 기능은 2차 발표 이후 구현할 예정이다.

**class : Search 카테고리 검색 페이지**

- 대분류 소분류로 책을 검색할 수 있는 기능이다.
- 창 구성을 완료했으며 기능은 2차 발표 이후 카테고리 별로 검색할 수 있는 부분을 구현할 예정

**class : informationChange 회원정보수정 페이지**

- 회원정보를 수정할 수 있는 창이다. 회원정보 수정 창에 회원이 입력했던 정보를 TextField에 보여주고, 회원이 자신의 정보를 수정할 수 있도록 하는 부분은 2차 발표 이후 구현할 예정이다.

**class : RequestMembers 요구사항 페이지**

- 회원이 요구사항이 있을 경우 요구사항 버튼을 클릭 한 후 제목과 내용을 입력한 후 보내기버튼을 누른다. 관리자페이지에서 관리자가 요구사항수신리스트 버튼 클릭할 경우 확인

할 수 있다. 회원이 글을 남기면 데이터베이스에서 입력 한 내용을 받아서 관리자 페이지에서 보여주도록 구현하는 것이 목적이다.

## 작업 진행 상황 - ALPHA

### 황세웅

저는 로그인페이지의 구성과 전체 프로그램 내부 디자인 및 버튼과 패널위치의 설정을 맡게 되었습니다.

먼저 로그인페이지의 기능 설명입니다.

외부패널에는 배경화면이미지가 삽입되어 있습니다.

버튼은 로그인, 아이디와 비밀번호 찾기, 회원가입 세개의 버튼이 있고,

텍스트필드는 아이디와 비밀번호를 입력하는 두 군데에 사용되었습니다.

로그인을 시도할 경우입니다.

만약 계정이 없을 시에는 입력란 아래의 회원가입 버튼을 클릭한 뒤 개인에 대한 정보를 입력하여 계정을 만들게 됩니다.

개인의 정보에 대한 입력란을 입력하게되며, 이는 회원가입 시 개개의 아이디 정보로 데이터베이스에 저장, 이후 활용됩니다.

이 때, 아이디 중복확인인 경우 버튼 클릭 시, 기존 데이터베이스에 저장된 아이디의 정보를 찾아서 없을경우, "사용가능합니다"라는 팝업창이 출력됩니다.

중복 될 경우에는 "사용할 수 없습니다"라는 팝업 창이 출력됩니다.

중복확인 버튼을 클릭하지 않거나 필수항목에 정보를 입력하지 않았을 시에는 회원가입을 누를 경우,

완료되지 않은 항목이 있다는 팝업창을 출력합니다.

다음으로 아이디/비밀번호를 찾고 싶을 경우입니다.

데이터베이스에서 본인이 가입할 때 입력했던 핸드폰 번호를 입력 한 뒤, 해당하는 정보가 있을경우 아이디를 출력합니다.

비밀번호는 본인의 아이디와 가입 시 입력했던 비밀번호 힌트를 통해서 찾을 수 있습니다.

마찬가지로 해당하는 정보가 있을경우 비밀번호를 출력합니다.

두 경우 모두 일치하는 정보가 없거나 바르게 입력되지 않았을 경우에는 "입력하신 정보가 없습니다" 라는 팝업창을 출력됩니다.

하지만 아이디와 비밀번호를 찾는 페이지는 따로 구체적으로 설계되지않았으며, 추후에 구성할 예정입니다.

현재 데이터베이스에 연동되어 활용은 되지 않고있으나, 회원가입페이지와 아이디 및 비밀

번호 찾기 페이지는 표면적인 부분이 구성되어 있습니다.

추후에, 데이터베이스를 연동시켜 회원가입 및 아이디 및 비밀번호 찾기 페이지를 원활하게 활성화 시킬 예정입니다.

그리고 로그인 버튼을 누를 경우, 아이디와 비밀번호를 다 입력했을 경우에만 프로그램 자체에 관리자로 등록된 계정일 경우 로그인 이후 관리자 페이지로, 일반적인 가입일 경우에는 회원 페이지로 이동하게됩니다. 입력값이 올바르지 않거나 없는 정보일 경우에는 "해당 정보를 찾을수 없습니다" 라는 팝업창이 출력됩니다.

그 다음이자 마지막으로 전체 프로그램 내부 디자인작업입니다. 로그인 페이지에서는 최대한 프로그램이 다루기 쉽고 친숙하게 다가올 수 있도록 책과 관련된 배경을 사용하였습니다. 또한 회원가입 페이지에도 배경 패널에 이미지가 삽입되어있습니다.

로그인페이지 이후 에는, 프로그램은 전체적으로 외부패널과 내부패널을 이용해서 겉패널 안에서 내부 패널이 유동적으로 변환될 수 있도록 구성하였습니다.

외부패널 이미지와 내부패널 이미지는 각각 서로 다른 이미지를 사용하였으며, 버튼 역시 기능을 대표하는 버튼아이콘을 제작해 각 버튼마다 삽입하였습니다.

각 페이지마다의 기능을 대표하는 버튼은 프로그램 왼쪽을 기준하여 각 3개씩 제공하였습니다.

페이지 내부에도 기능을 설명하는 배너 이미지와 버튼 이미지를 따로 사용하였습니다. 현재 아직 회원정보 수정 페이지가 아직 구성되어 있지 않은데, 구성된 이후에 회원정보를 수정하는 부분에도 회원가입 페이지와 비슷하게 이미지가 삽입될 예정입니다.

전체적인 디자인은 연두색 색감을 주로 활용하였으며, 최대한 사용자가 헛갈리지 않고 쉽게 버튼을 찾을 수 있도록 구성하였습니다.

추후에 모든 페이지가 완료된 이후 능률적인 프로그램을 구성하기위해 버튼과 전체적인 구성을 다듬을 예정입니다. 그리하여 프로그램을 이용하는 사용자가 올드하거나 불편함 없이 프로그램을 쉽고 기분 좋게 이용할 수 있게 하는 것이 저의 디자인 및 버튼/패널 구성 목적입니다.

## 박병주

패널연결

로그인 창

로그인 버튼을 누를때 회원의 아이디,비밀번호를 DB로부터 받아와 관리자창(ID:admin) 또는 사용자창(ID:admin을제외한 모든 ID)을 띄우도록한다.(일치하지 않을시 일치하지 않다는 팝업창을 띄우도록한다.)

회원가입 버튼을 누를때 회원가입창이 띄어진다.

아이디,비밀번호 찾기 버튼의 이벤트는 2차 발표 이후 작업할 예정이다.

#### 관리자창

로그인창에서 관리자창으로 이동할때 name값을 넘겨주어 사용자의 이름또는 관리자의 이름이 드러나게끔 하였다.

시간을 thread를 이용하여 추가하였다. 이것은 현재 시간을 확인할수 있고 연체 회원의 연체 기한을 확인 할때 유용하게 이용될 예정이다

로그아웃버튼을 삽입하여 회원간에 자유자재로 이동이 가능하도록 했다.

도서관리 : 도서관리 버튼을 누를때 도서관리의 setPanel()을 호출하고 회원관리,요구사항의 객체는 null 값을 지정한다(hash코드의 중복으로 패널연결을 해도 값을 새로이 지정할수 없음.null지정으로 새로운 hash코드를 받도록하였다.

이로서 패널의 삭제 이동이 편리해졌다.)

현재 책제목, 저자, 출판사를 입력하고 등록을 누를 때 DB의 inset\_book()메서드를 호출하여 세가지 입력정보를 추가하도록 한다.

의 카테고리별로 책의 정보를 추가하도록 할예정이다(2차발표 이후)

현재목록버튼을 누르면 DB의 view\_book()메소드를 호출하여 현재 책목록을 갱신할수 있도록 하였다.

책 제거 버튼을 누르면 제거할 책을 입력하라는 팝업창을 표시하고 책의 이름을 입력하면 DB의 remove\_book()메소드를 호출하여

책의 목록에서 제거하도록한다.

JTable메서드를 이용하여 DB의 view\_book()클래스를 가져와 현재 책의 정보를 보이도록 하였다.

회원관리 : 김가영 팀원이 구축했다. 2차 발표때 연체회원버튼의 리스너를 달아주려고 한다.

요구사항확인 : JTable 클래스를 이용하여 요구사항 수신리스트를 볼수 있도록 했다. DB 의 view\_require()메서드를 이용하여

RequestCenter 에서 request\_num(수신번호),title(제목),content(내용)을 받아오도록 한다.

JTable의 너비와 행의 길이를 조절하여 보기 편하게 구성하였다.

JTable의 각 요소에 리스너를 달아 요구사항의 내용을 볼수 있게끔할려고 한다.

(2차발표 이후구현)

#### 사용자창

(도서반납버튼,카테고리검색버튼,요구사항버튼)에 패널을 연결하였다. 관리자 창과 동일하게 로그아웃, 시간쓰레드를 구성하였다.

이후 기본적인 팬 구성은 김상수 팀원이 하였다.

각 버튼에 대한 리스너와 기능은 2차발표 이후 달아줄 예정이다.

관리자창과 사용자창의 팬 구성은 기본 팬 위에 layeredPane팬을 달아 그 위에 배경과 기본 버튼, 라벨, 텍스트를 구성하였다. 새로운 팬이 연결될 때 배경이 새로운 팬을 먹어버리는 에러 사항으로 새로운 팬이 나오는 위치에 null값을 구성하여 새로운 팬이 드러나게 했다. 팬 위에 두개의 팬이(겹치지 않게)구성하였다

## 김가영

담당 작업은 클래스에 필요한 세부 메소드 만들기, 이벤트를 구현하는 부분을 맡았습니다. 또한, 데이터베이스와 연동을 하는 클래스를 담당하고, 데이터베이스 클래스의 메소드 추가 및 수정, 삭제를 맡았습니다.

현재 저희가 사용하는 데이터베이스는 MS SQL로 LIBRARY라는 이름을 가진 데이터베이스로 CUSTOMER, BOOK, RESERVED, REQUIRE, BORROW의 테이블이 있습니다. CUSTOMER 테이블은 회원이 가입할 때 정보를 저장해주고, 회원이 자신의 정보를 수정할 때 정보가 수정되는 공간으로, 각 회원의 number를 자동으로 부여해주는 customer\_id가 있습니다. 회원이 가입할 때 입력하는 id, 이름, 나이, 성별, 주소, 핸드폰 번호, 비밀번호 힌트, 힌트 정답으로 구성이 되어 있습니다. 회원이 입력한 password는 암호화를 거쳐 저장되며 관리자인 저희 또한 회원의 password를 알 수가 없도록 했습니다. 또한 각 회원별로 대출한 책의 권수를 축적하는 count를 만들어 줄 예정입니다. BOOK 테이블은 관리자가 입력해주는 책에 대한 정보를 저장하는 공간으로 책에도 고유번호를 자동으로 부여해주는 book\_id가 있습니다. 관리자가 추가한 책의 제목, 저자, 출판사, 책 표지 경로, 판수를 저장할 수 있도록 구성이 되어 있습니다. 아직까진 책 표지 경로를 추가해주는 부분은 사용하고 있지 않으며, BOOK 테이블에서 책에 대한 카테고리를 지정해 줄 수 있는 부분을 구현할 예정입니다. RESERVED 테이블은 책을 예약한 정보를 저장하는 공간으로 REQUIRE 테이블은 회원들이 보낸 요구사항을 저장하고, 관리자에게 보여주는 곳으로 익명성을 위해 회원의 아이디는 저장하지 않도록 요구사항의 번호, 요구사항의 제목과 내용만을 담고 있습니다. BORROW 테이블은 각 회원이 빌린 책의 빌린 날짜와 반납해야 될 날짜를 저장하는 곳입니다. 회원의 아이디를 저장하고, 빌린 날짜와 반납 날짜를 저장하는 공간을 만들어 두었습니다.

현재 구현한 데이터베이스 클래스는 ConnectionDB로 데이터베이스와 연동을 해주는 역할을 합니다. 회원의 정보를 보여주는 view\_user()는 관리자가 회원목록을 확인할 때 보여주는 메소드로 회원의 user\_id, id, 이름, 나이 등을 출력합니다. 또 다른 방식으로 회원의 정보를 보여주는 메소드인 view\_user(String name)는 회원의 이름을 받아 입력받은 name이 포함되는 이름을 가진 회원들의 정보를 출력해줍니다. combobox를 통해 정렬방식과 이름을 받아 회원정보를 출력해주는 메소드인 view\_user(String name, int sort)는 combobox의 인덱스에 해당되는 정렬 방식으로 입력받은 name이 포함되는 회원 정보를 출력해하도록 switch문을 통해 정렬을 해줍니다. view\_one\_user(String customer\_id)는 자신의 정보를 확인하고자 하는 회원의 번호를 받아서 해당하는 회원의 정보를 출력해줍니다.

insert\_user(String id, String password, String name, String age, String gender, String address, String phone, int hint\_num, String hint) 메소드는 회원이 가입할 때 입력한 정보를 데이터베이스에 저장해주는 메소드입니다. password는 암호화를 통해 저장되어야하므로 sql에 pwdencrypt를 통해 암호화하여 데이터베이스 저장을 해주도록 하였습니다. 회원이 자신의 정보를 수정할 수 있는 창인 update\_user(String id, HashMap) 메소드는 회원의 id에 해당되는 정보 중에서 수정된 정보를 가지고 데이터베이스에 저장된 해당 회원의 정보를 수정해줍니다. delete\_user(String id) 메소드는 회원이 탈퇴를 하고자 할 때, 회원의 id에 해당되는 회원의 정보를 데이터베이스에서 삭제를 해주는 역할을 합니다. check\_id(String id) 메소드는 데이터베이스에 있는 id인지를 확인하여 사용가능한지의 여부를 반환해주는 메소드로 회원이 가입할 때, 중복확인을 통해 확인을 할 수 있으며, 로그인을 할 때 id가 존재하는지 여부를 확인해줍니다.

check\_password(String id, String password) 메소드는 id와 password가 데이터베이스에 저장된 정보와 일치하는지를 확인하는 메소드입니다. password가 암호화되어 있기 때문에 pwcompare를 통해 입력된 비밀번호와 일치하는지를 확인해줍니다. view\_book() 메소드는 현재 데이터베이스에 저장된 책에 대한 정보를 출력해주는 공간으로 책의 번호, 책의 제목, 저자, 출판사를 출력해줍니다. update\_book(String title, String author, String publisher, String path, int edition) 메소드는 관리자가 추가한 책의 정보인 제목, 저자, 출판사, 사진경로, 판수를 입력받아 데이터베이스에 책의 번호와 함께 추가를 해줍니다. delete\_book(String title) 메소드는 관리자가 삭제하고자 하는 책의 제목을 입력받아 책의 제목에 해당되는 책의 정보를 데이터베이스에서 삭제해줍니다. view\_require()은 관리자에게 사용자가 보낸 요구사항을 보여주는 메소드로 이 메소드에서는 요구사항의 번호와 제목만을 출력해줍니다.

아직 구현되지 않은 메소드인 view\_require(int require\_num)은 관리자가 보고자 하는 요구사항을 선택했을 때 요구사항의 번호에 해당되는 내용을 출력해줄 수 있도록 구현할 예정입니다. insert\_require(String title, String content) 메소드는 회원이 요구사항의 제목과 내용을 입력하면 데이터베이스의 REQUIRE 테이블에 제목과 내용을 저장해줍니다.

delete\_require(int require\_num)은 아직 구현되지 않은 메소드로 이 메소드에서는 관리자가 삭제하고자 하는 요구사항을 데이터베이스에서 삭제를 해줄 수 있도록 구현을 할 예정입니다.

## 김상수

저는 사용자페이지를 구성하고 세부적인 기능을 추가하는 부분을 맡았습니다.

사용자로 로그인 되면 바로 도서명으로 검색을 할 수 있는 패널이 뜨고 검색을 할 수 있습니다.

텍스트박스과 버튼을 추가하여 구성하였습니다. 아직 이부분은 DB 작업이 되지않아서 패널 구현만 되어있습니다.

DB작업이 끝난 후 검색버튼의 이벤트를 추가하여 실제로 검색할 수 있게 작업할 것입니다. 도서반납 버튼을 누르면 빌린도서목록이 출력될 수 있게 테이블이 있고 반납 버튼이 있습니다.

이부분도 DB작업이 되지않아 출력되지않고, 반납버튼의 이벤트처리는 DB작업이 된 이후 하려고 합니다.



카테고리검색 버튼을 누르면 combobox 두개(대분류,소분류)가 있고 대분류에 맞는 소분류 item을 띄우게 작업하였고,  
 검색결과가 출력되는 테이블을 만들어놓았습니다. combobox사용시 다른 창에 이동하였다 오면 제대로 작동되지 않는 문제가  
 있었는데 다른 패널들을 이동할 때 다른 객체를 null값으로 지정하여 새로운 hash코드를 받아오는 방법으로 문제해결을 해결했습니다.  
 요구사항버튼을 누르면 아직 흰 패널만 띄웠고 발표 이후 작업하여 요구사항을 입력할 수 있는 텍스트박스를 만들고  
 관리자에게 보낼 수 있게 보내기 버튼을 만들 것입니다.  
 또한 신규도서 목록을 한 귀퉁이에 출력시키는 기능도 추가할 예정입니다.

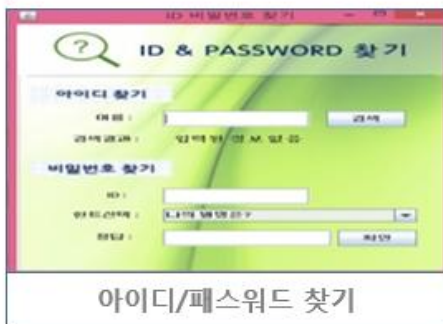
## 프로그램 구동 스크린샷



로그인 화면



회원가입



아이디/패스워드 찾기



관리자MAIN



도서관리



회원관리



요구사항 수신리스트



회원MAIN



도서예약



카테고리 별 검색



요구사항 보내기



회원정보수정

## 기능 구현 상황

### -로그인페이지-

#### <Motivation>

1. 사용자가 쉽게 로그인을 할 수 있도록 하여야 한다.
2. 계정 정보가 없을 시 가입을 할 수 있어야한다.
3. 계정 정보를 잊었을 경우 계정정보를 찾을 수 있어야한다.
4. 접속자가 관리자인지 사용자인지에 따라 로그인 이후 접속화면이 달라야한다.

#### <기능구현 및 화면구성>

##### 로그인화면

1. 사용자는 아이디와 패스워드를 입력할 수 있다.
2. 등록된 계정정보를 입력 시 로그인 된다.
3. 등록 되지 않은 계정정보이거나 아이디와 패스워드를 입력하지 않은 경우 로그인을 할 수 없다는 메시지가 출력된다.

##### 회원가입

1. 사용자는 가입 시 필요한 사용자의 이름, 나이, 주소, 성별, 비밀번호, 아이디 등 필수 입력정보를 입력하여야 한다.
2. 필수 정보 입력 시 사용자는 회원가입을 할 수가 있다.
3. 필수 정보 중 완전히 입력되지 않은 정보가 있을 경우 해당하는 정보를 입력하라는 메시지를 출력한다.
4. 아이디를 입력할 시 기존에 있는 아이디인지를 확인할 수 있는 기능을 구현하였다.
5. 비밀번호와 비밀번호확인란의 정보가 일치하여야 회원가입을 할 수 있다.
6. 모든 필수 정보를 입력했을 시 사용자는 회원가입을 할 수 있다.

##### 아이디 패스워드 찾기

1. 사용자는 자신의 계정(아이디, 패스워드) 정보를 잊어버렸을 경우 찾을 수 있다.
2. 회원가입 시 입력했던 정보 중 이름을 입력할 경우 아이디를 찾을 수 있다.
3. 계정정보가 담긴 데이터베이스에 입력한 정보가 없을 경우 아이디가 없다는 메시지를 출력한다.
4. 비밀번호는 비밀번호를 찾으려는 아이디와 회원가입 시 입력했던 비밀번호 힌트를 선택한 후 답을 입력하면 찾을 수 있다.
5. 이 경우에도 계정정보와 비밀번호 힌트와 비밀번호가 일치하지 않을 시 일치하는 정보가 없다는 메시지를 출력한다.
6. 모든 경우에 회원가입 시 입력했던 정보와 맞는 올바른 정보를 입력할 경우 아이디와 패스워드를 찾을 수 있다.

## -관리자페이지-

### <Motivation>

관리자는 도서와 회원을 관리할 수 있어야한다.

관리자는 회원의 요구사항을 해결할 수 있도록 요구사항을 읽을 수 있어야한다.

관리자는 회원의 요구사항을 읽고 이를 피드백 결과를 알려줄 수 있도록 요구사항을 보낸 회원에게 답장 메시지를 보낼 수 있어야한다.

관리자는 연체회원을 확인하고 연체된 회원에게 도서를 대출할 수 없는 연체기한을 정할 수 있어야한다.

관리자는 연체회원에게 개인적인 경고 메시지를 보낼 수 있어야한다.

관리자는 신규 도서가 있을 시 이를 기존 도서목록에 추가할 수 있어야한다.

관리자는 도서의 정보를 추가와 동시에 수정 및 삭제를 할 수 있어야 한다.

### 도서관리

1. 관리자가 도서의 목록을 확인하고 갱신할 수 있도록 하였다.
2. 관리자는 원하는 도서의 정보를 수정 및 삭제 할 수 있도록 하였다.
3. 신간 도서가 있을 경우 새로운 도서의 정보를 등록할 수 있다.

### 회원관리

1. 프로그램 내에 등록된 사용자에 대한 정보를 확인 할 수 있다.
2. 사용자를 선택적인 기준에 따라서 오름차순, 내림차순을 할 수 있다.
3. 회원 중 대출기한을 넘긴 회원 즉 연체회원을 일반회원과 구분할 수 있다.
4. 연체된 회원에 대해 대출기한을 제재하고 개인적인 경고 메시지를 보낼 수 있다.

### 요구사항

1. 회원이 보낸 요구사항 메시지 목록을 확인할 수 있다.
2. 메시지 목록에서 메시지를 확인하고 그에대한 개별적인 답장을 할 수 있다.
3. 메시지 목록을 갱신 및 읽은 메시지에 대해서 삭제 할 수 있다.

## -회원페이지-

### <Motivation>

1. 사용자(회원)은 자신이 원하는 도서를 예약할 수 있어야한다.
2. 사용자는 원하는 도서가 목록에 있는지 검색할 수 있어야한다.
3. 사용자가 보다 쉽게 검색할 수 있도록, 도서의 분류기준에 따른 카테고리 별 검색 기능을 제공한다.
4. 사용자는 출력 정보를 확인하고 해당 도서의 정보(도서명, 출판사, 도서이미지 등)를 확인할 수 있어야한다.
5. 회원은 프로그램 이용 시 불편한 사항을 관리자에게 전달할 수 있어야한다.
6. 회원은 불편한 사항에 대한 피드백을 관리자로부터 받고 확인할 수 있어야한다.
7. 연체회원일 경우 관리자로부터 메시지를 수신 받고 읽을 수 있어야한다.
8. 회원은 회원가입 시 입력했던 정보를 수정할 수 있어야한다.

9. 회원은 원하는 경우 회원탈퇴를 할 수 있어야한다.
10. 프로그램은 회원에게 신간도서에 대한 정보를 출력시켜 주어야한다.

### 도서예약

1. 사용자는 원하는 도서를 도서예약버튼 클릭 후 검색탭에서 검색하여 찾은 뒤 예약할 수 있다.
2. 이 때 도서검색란을 통해서 다른 등록된 도서와 그에 대한 정보를 확인할 수 있다.

### 도서검색

1. 도서검색 버튼 클릭 후 카테고리를 선택하면 회원은 그에 맞는 책의 목록을 확인할 수 있다.
2. 회원은 오른쪽 상단 개인정보 탭 아래에서 신간도서의 목록을 확인할 수 있다.
3. 책의 정보에 따라 도서의 목록을 오름차순, 내림차순으로 정렬할 수 있다.

### 요구사항

1. 회원은 자신이 원하는 정보를 관리자에게 요구사항수신리스트 버튼 클릭 후 메시지를 입력하여 보낼 수 있다.
2. 이에 대한 피드백 메시지를 오른쪽 상단 개인정보 탭에 있는 메시지 함 버튼으로 관리자에게 수신 받을 수 있다.
3. 수신 받은 메시지는 원하는 경우 삭제 가능하다.

### 회원정보수정

1. 사용자는 오른쪽 상단의 정보수정 버튼을 이용하여 정보수정 페이지로 이동할 수 있다.
2. 정보수정 페이지에서 회원가입 시 입력했던 정보를 아이디를 제외하고 모두 수정할 수 있다.
3. 수정한 정보를 변경버튼을 통해서 기존의 데이터를 수정/저장할 수 있다.
4. 회원정보 수정 창에서 원하는 경우 회원탈퇴를 할 수 있다.
5. 회원은 필요한 경우 로그아웃 버튼을 이용해서 로그인페이지로 로그아웃할 수 있다.

## 개인별 작업

### 김상수

저는 사용자페이지를 구성하고 세부적인 기능을 추가하는 부분을 맡았습니다.

### 도서예약

도서예약 버튼을 누르면 전체 도서목록이 출력될 수 있게 table이 있고 예약 버튼이 있습니다.

이 table에는 스크롤기능이 추가되어 목록이 table창을 넘어갈 경우 스크롤을 이용하여 확인

할 수 있습니다.

도서 예약은 DB에서 값을 불러와 출력되며, 전체 도서의 정보를 출력해주고 예약을 원하는 도서를 클릭하여 예약버튼을 누르면 예약처리 되도록 하였습니다.

이벤트처리는 DB가 사용되는데 그 부분은 김가영, 박병주 조원이 기능을 추가하였습니다.

### 카테고리 검색

카테고리 검색 버튼을 누르면 combobox 두 개(대분류, 소분류)가 있고 대분류에 맞는 소분류 item을 띄우게 작업하였습니다, 그리고 검색결과가 출력되는 table을 만들어놓았습니다.

combobox 사용 시 다른 창에 이동하였다 오면 제대로 작동되지 않는 문제가 있었는데 다른 패널들을 이동할 때 다른 객체를 null값으로 지정하여 새로운 hash코드를 받아오는 방법으로 문제를 해결했습니다.

combobox는 대분류에 맞는 소분류 item들이 출력되게 기능을 완성하였습니다.

검색기능에서는 DB기능이 추가되어 테이블에 검색결과를 table에 검색된 도서의 책 이름, 저자, 출판사와 대출가능여부를 출력하게 됩니다.

이 기능 역시 DB를 이용하여 구현하여야 했기 때문에 김가영, 박병주 조원이 기능을 구현해주었습니다.

검색된 도서의 목록이 출력되는 table 역시 스크롤기능이 추가되어 있습니다.

table에 출력된 검색결과를 클릭 시 도서의 정보가 출력되고, 도서예약을 할 수 있는 기능 역시 DB를 이용하기 때문에 김가영, 박병주 조원이 구현하였고,

저는 도서의 정보를 수집하는데 도움을 주었습니다.

### 요구사항

요구사항 버튼을 누르면 제목을 작성할 수 있는 textfield와 요구사항을 작성할 수 있는 textarea를 삽입했고, 관리자에게 전송하는 버튼을 만들었습니다.

요구사항부분도 DB기능을 사용하여 세부적인 기능은 김가영, 박병주 조원이 구현하였고, 요구사항을 작성할 수 있는 textarea부분에 스크롤기능을 추가하여 작성 시 창을 넘어갈 경우 문제가 없도록 할 예정이었으나, 구현하지 못하였습니다.

사용자로 로그인 했을 시 프레임에 신규도서목록을 출력하는 기능을 구현하였습니다.

쓰레드를 이용하여 신규도서가 하나씩 출력되게 구현하였습니다.

하지만 DB기능을 추가하여 신규로 추가된 목록만을 출력하는 것은 시간이 부족하여 구현하지 못하였고,

대신 배열을 이용하여 그 배열에 추가된 모든 도서를 입력하여 추가된 모든 도서가 출력되게 하였습니다.

제가 총 코딩한 라인은 362이고, DB에서 추가로 100라인(책 정보)이 있습니다.

## 박병주

저는 패널을 연결하여 조원들에게 전달한 후 각자 분담한 패널에 기능추가와 수정을 하였습니다.

### 관리자 전용 페이지

첫 화면에 관리자의 아이디를 확인할 수 있게 하였고(사용자 전용페이지와 통일성을 염두하여 구성하였습니다.) 시간 쓰레드를 이용하여 실시간으로 시간을 입력받게 했습니다. 로그아웃 버튼을 추가하여 로그아웃 후 사용자가 로그인할 수 있게 구성했습니다. 추가되는 기능은 세 개의 버튼(도서관리, 회원관리, 요구사항 수신리스트)으로 구성했습니다.

### 도서관리 페이지

책제목에 콤보박스(회원페이지에서 카테고리 검색을 유용하게 하도록)를 추가하여 대분류 소분류 그리고 책제목을 입력받게 구성하였습니다. 저자와 출판사 그리고 경로를 가져와 책의 이미지를 데이터베이스에 저장하여 불러올 수 있게끔 구성하였습니다. 쓰레드로 도서관리 페이지를 구성하지 않아서 등록버튼을 누르고 나서 현재목록을 눌러야 데이터베이스에서 값을 새롭게 불러옵니다. 책 제거 버튼을 추가하여 책제목을 입력받고 데이터베이스에서 책을 제거 하도록 하였습니다. 테이블에 이벤트 리스너를 달아서 책의 정보를 누르면 책의 이미지를 띄우도록 구성하였고 대출하기기능과 반납기능은 김 가영 학생이 구성하였습니다.

### 회원관리 페이지

회원검색을 콤보박스(오름차순, 내림차순, 이름순, 나이순)으로 구성하여 보고 싶은 회원을 한눈에 보기 쉽게 구성하였고 아래 테이블에 그 목록이 출력됩니다. 연체회원관리 버튼 리스너를 달았습니다. 그 내용은 아이디, 빌린 책, 빌리날, 연체 일을 계산하여 어떤 회원이 연체를 어떻게 했는지 한눈에 보기 쉽게 확인 할 수 있습니다. 연체회원에게 메시지를 보내면 연체회원은 자신이 연체를 한지 확인 할 수 있게 됩니다.

### 요구사항 수신리스트

회원들의 불만사항을 볼 수 있는 페이지입니다. 테이블을 이용하여 데이터베이스에서 요구사항을 출력하게 만들었고 내용을 확인 후 삭제, 확인을 할 수 있게 구성했습니다.

### 사용자 페이지 구성

관리자 페이지와 동일하게 자신의 아이디를 확인하고 시간을 실시간으로 확인 할 수 있게 구성하였습니다. 메시지 함을 만들어 자신이 연체를 하였는지 확인 할 수 있게 구성하였습니다. 위와 동일하게 로그아웃 버튼도 구성하였습니다.

### 요구사항 보내기

사용자가 요구사항을 입력하고 메시지 보내기 버튼에 리스너를 달아 그 내용이 데이터베이스로 전달되도록 구성하였습니다.

제가 총 코딩한 라인 수는 약 1400줄입니다.

## 김가영

저는 조원들이 만든 소스에 대한 기능을 더 추가하고, 데이터베이스의 전체적인 부분을 맡게 되었습니다.

### 대출기능

회원이 마음대로 대출을 하는 것이 아닌 관리자를 통해 대출이 가능하도록 만들었습니다. 원하는 책을 선택하고 마우스 오른쪽 버튼을 누르면 선택할 수 있는 메뉴에서 '대출하기'를 대출을 할 수 있도록 프레임을 띄울 수 있도록 이벤트를 구현하였습니다.

프레임에는 아이디 입력란과 선택한 책의 도서명이 출력됩니다.

아이디 입력이 없을 시에는 아이디를 입력하라는 명령창이 나오고, 아이디를 입력하면 데이터베이스에 있는 아이디인지를 확인해줍니다.

데이터베이스에 저장된 아이디가 아니라면 없는 아이디라는 명령 창을 출력해줍니다.

또한 연체된 회원인지를 확인하고 연체된 회원이 아니고, 대출한 책의 권수가 2권 이하라면 대출에 성공을 합니다.

대출에 성공을 하면 대출가능여부가 'N'으로 수정되도록 하였습니다.

## 도서반납

회원이 마음대로 반납을 하는 것이 아닌 관리자를 통해 반납이 가능하도록 만들었습니다.

반납하고자 하는 책을 선택했을 때, 대출여부가 'N'일 때 popup메뉴에 반납하기 메뉴가 추가될 수 있도록 하였습니다. '반납하기'를 누르게 되면 borrow테이블에서 빌린 내용이 사라지고 대출여부가 'Y'로 변경되도록 하였습니다.

## 연체회원센터

연체회원센터에서는 빌린 회원들의 정보를 가져와서 반납기한이 지났는지를 확인하여 대출한 회원들의 정보를 모두 각 문자열 배열에 저장을 합니다.

count변수를 static으로 두어 다른 클래스에서 이 변수를 이용하여 값을 얻어올 수 있도록 했습니다.

getAll에서 vector에 저장을 해서 넘겨줄 수 있도록 했습니다.

## connectionDB

데이터베이스에 있는 회원정보, 책, 메세지, 요구사항, 책 예약, 책 대출에 관련된 테이블의 정보를 가져올 수 있는 메소드를 모두 추가하였습니다.

그리고 세부적으로 필요한 삭제, 수정, 삽입할 수 있는 메소드 또한 모두 구현을 하였습니다.

내용을 보여줄 때는 vector에 저장하여 값을 넘겨주고, 삭제, 수정, 삽입에서는 boolean을 이용하여 성공적으로 이루어졌는지를 확인할 수 있도록 하였습니다.

## 가입

가입을 할 때 일어날 수 있는 모든 경우에 대한 명령창을 추가해두었습니다. 아이디입력, 비밀번호, 비밀번호 체크 등의 가입시 필요한 정보에 대한 내용이 입력되지 않았을 경우에 그에 대한 경고창을 출력하도록 하였습니다.

## 책예약

책을 예약할 수 있는 부분에 추가적인 명령창 및 예약제한을 만들어 두었습니다. 예약 권수가 3권 이하일 때 예약이 가능하고, 다른 사람이 예약하지 않았을 경우에만 예약이 가능합니다.

## 데이터베이스

데이터베이스에는 회원정보를 저장하는 테이블, 책 정보를 저장하는 테이블, 요구사항정보를



저장하는 테이블, 메시지 정보를 저장하는 테이블, 예약정보를 저장하는 테이블, 대출정보를 저장하는 테이블을 구현하였습니다.

사용하는 변수만을 놔두고 필요 없는 변수는 두지 않았고, 회원의 비밀번호는 암호화를 통해 알 수 없도록 하였습니다.

데이터베이스에 INSERT문을 이용하여 기본적으로 보여줄 수 있는 내용을 추가하였고, 많은 양의 책의 내용 또한 저장을 해주었습니다.

제가 총 코딩한 라인 수는 약 **2000**줄이고, 데이터베이스 **69**줄입니다.

## 항세응

담당한 파트 : 로그인화면, 프로그램 디자인

### 목표 및 동기의식 :

로그인화면// 로그인을 하는데 타 프로그램보다 못하지 않고 필요한 기능이 주어진 부족함 없는 로그인 화면 구성을 하는 게 목적이였다.

프로그램디자인// 누구나 사용하기 쉽도록 도서프로그램을 디자인 하고 프로그램을 사용할 때 오래되고 낡은 느낌이 아닌 세련되고 사용자가 밝은 환경에서 이용할 수 있도록 디자인 하는 것이 목적이였다.

### 작업 진행 정도:

처음에 하려고했던,

관리자, 사용자로 구분되는 화면으로 이동하는 과정

아이디 패스워드 찾기, 회원가입 클래스는 모두 완성 되었다.

로그인화면 프레임만 구성하고 DB연결 및 관련 소스는 박병주 학생과 김가영 학생이 작업 하였다.

디자인경우도 필요한 버튼 및 배경의 이미지를 모두 삽입하였다.

팝업창에 필요한 이미지를 삽입하려고 마음 먹은 것도 모두 완료하였다.

코딩라인 : 로그인프레임 138line. + 기타 라벨 사이즈 설정

setbound() 메소드, setForeground,background 및 new imageicon() 메소드를 통한 이미지 삽입 라인까지 **200**line 정도입니다.

## 전체적인 프로그램 디자인 구성 예시

