

데이터베이스 프로젝트



18011501 김대선

I. 학번, 이름

본 프로젝트는 컴퓨터 공학과 18학번 18011501 김대선이 1인으로 진행하였습니다.

II. 요구사항 분석내용

요구사항에 대하여 분석한 내용은 다음과 같습니다.

1. 영화에 대해서는 영화번호, 영화명, 상영시간, 상영등급, 감독명, 배우명, 장르, 영화소개 및 개봉일 정보를 저장한다. 모든 영화는 1개 이상에서 4개 이하의 상영일정을 가진다.

→ 영화 테이블 : 영화번호 , 영화명, 상영시간, 상영등급, 감독명, 배우명, 장르, 영화소개 및 개봉일 정보 (단, 같은 영화에 1~4개의 상영일정이 연결될 수 있다.)

2. 상영일정에 대해서는 상영일정번호, 영화번호, 상영관번호, 상영시작일, 상영요일, 상영회차 및 상영시작시간 정보를 저장한다.

→ 상영일정 테이블 : 상영일정번호, 영화번호, 상영관번호, 상영시작일, 상영요일, 상영회차 및 상영시작시간

3. 상영관에 대해서는 상영관번호, 좌석수 및 상영관사용여부를 저장한다. 각 상영관에는 1개 이상의 상영일정을 배정한다.

→ 상영관 테이블 : 상영관번호, 좌석수 , 상영관사용여부 (단, 상영관별로 1개 이상의 상영일정이 연결되어야 한다.)

4. 티켓에 대해서는 티켓번호, 상영일정번호, 상영관번호, 좌석번호, 예매번호, 발권여부, 표준가격 및 판매가격 정보를 저장한다. 각 티켓은 1개의 좌석과 연결된다.

→ 티켓 테이블 : 티켓번호, 상영일정번호, 상영관번호, 좌석번호, 예매번호, 발권여부, 표준가격, 판매가격 (단, 각 티켓은 1개의 좌석과 연결되어야 함)

5. 좌석에 대해서는 좌석번호, 상영관번호 및 좌석사용여부를 저장한다.

→ 좌석 테이블 : 좌석번호, 상영관번호 , 좌석사용여부

6. 회원고객에 대해서는 회원아이디, 고객명, 휴대폰번호 및 전자메일주소를 저장한다.

→ 회원 테이블 : 회원아이디, 고객명, 휴대폰번호, 전자메일주소

7. 예매정보에 대해서는 예매번호, 결제방법, 결제상태, 결제금액, 회원아이디 및 결제일자를 저장한다.

→ 예매 테이블 : 예매번호, 결제방법, 결제상태, 결제금액, 회원아이디, 결제

일자

8. 영화예매 사이트는 관리자 혹은 회원으로서 이용이 가능하다.

→ 회원으로 이용하는 방식이 있고 관리자로 이용하는 방식이 있다.

요구사항 이외에 추가적으로 적용한 부분은 다음과 같습니다.

1. 상영일정 정보에 들어가는 상영 시작일에 대해서 다음과 같이 해석하였습니다. 상영일정 테이블에는 말 그대로의 상영일정에 대한 정보가 들어가야 하는데 이 정보를 상영 시작일 정보에 담았습니다. 예를 들어, 2022년 6월 6일 밤 23시에 시작하는 120분짜리 영화가 있다면 영화가 시작한 날짜는 6월 6일 이지만 영화가 끝나는 날짜는 6월 7일이므로 상영시작일을 해당 영화가 상영을 시작한 상영날짜로 해석하였습니다.

2. 문제에서 주어진 조건의 모순을 해결하기 위하여 다음과 같은 방법을 사용하였습니다. ‘각 티켓은 1개의 좌석과 연결된다’라는 조건과 ‘모든 영화는 1개 이상에서 4개 이하의 상영일정을 가진다.’라는 조건과 ‘각 상영관에는 1개 이상의 상영일정을 배정한다.’라는 조건을 현재 가지고 있는 정보만으로 모두 만족시키기에 모순되는 부분이 있었습니다.

예를 들어, A라는 영화가 상영관 1에서만 상영을 하는 상황인데 상영관 1에서의 좌석은 좌석번호(1~5)까지 5개가 있고 A라는 영화는 2022년 6월 5일과 2022년 6월 6일이라는 2개의 상영일정을 가진다고 가정하겠습니다. 각 티켓이 1개의 좌석과 연결되기 때문에, 다음과 같은 상황이 발생할 수 있습니다. 2022년 6월 5일 상영관1의 1번자리와 2번자리가 예매가 되어있고, 2022년 6월 6일 상영관1의 3번자리와 4번자리와 5번자리가 모두 예매가 되어있다면, 6월 5일에 남은 3,4,5번 자리와 6월 6일에 남은 1,2번 자리가 모두 예매가 가능해야 하지만 각 티켓과 각 좌석이 1:1로 매칭되어 있으므로 좌석사용여부는 모두O가 될 수가 있습니다. 6월 5일에는 사용하는 좌석이지만 6월 6일에는 사용하지 않는 좌석일 수 있는데, 이러한 경우에 좌석 사용여부를 O로 해야 하는지, X로 해야 하는지에 대한 치명적인 모순이 발생합니다.

이러한 모순을 해결해주기 위하여, 좌석 테이블에 상영일정번호를 추가하였고 주키(PK)를 좌석번호만 설정하는 것이 아니라 상영일정번호와 좌석번호를 동시에 묶었습니다. 따라서 각 티켓은 1개의 좌석과 연결된다는 점도 해결하였고, 각 상영일정에 맞게 좌석사용여부가 표시되므로 모순되는 부분 또한 해소되었습니다.

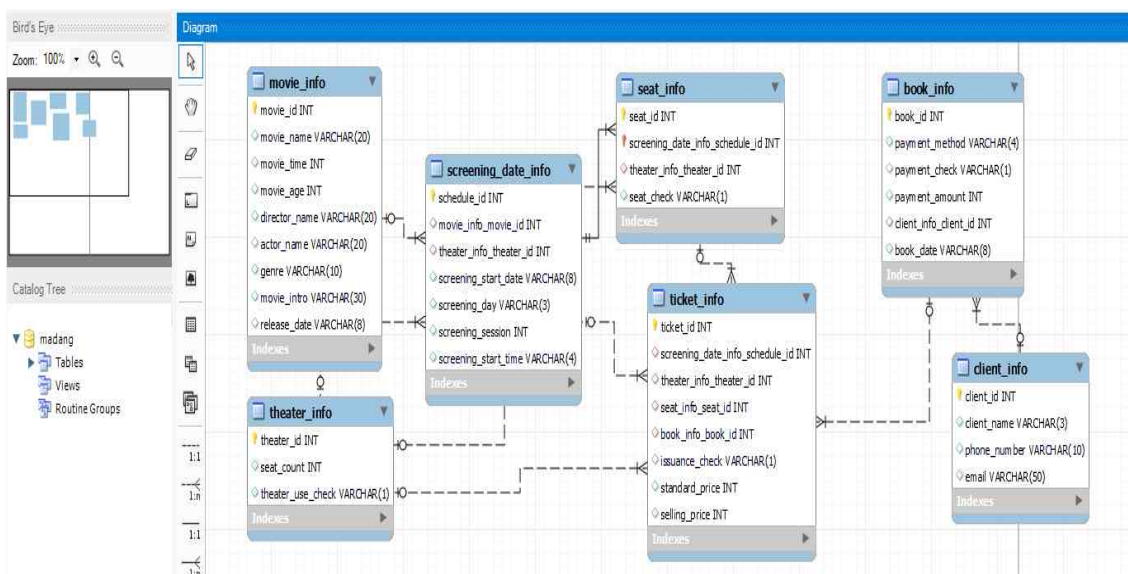
Ⅲ. 요구사항 명세서

프로그램의 원활한 실행을 위해, 다음과 요구사항을 가진다.

1. 모든 상영관은 사용가능한 상영관이다.
2. 수월한 예매를 위해 예매번호인 book_id는 AUTO INCREMENT이다.
3. 서로 묶여있는 정보들이 많으므로 관리자 모드에서 무분별한 delete와 update는 불가능하다. SET foreign_key_checks = 1이므로 선부른 삭제 접근은 애초에 예외처리에 걸려서 불가능하지만 UPDATE는 가능하기 때문에, 다른 테이블에도 영향을 주는 UPDATE는 해당 항목까지 같이 올바르게 수정하게 하거나, 시도해서는 안 된다.
4. 프로그램 편의성을 위하여 프로그램을 사용하는 오늘의 날짜는 20220401로 고정시켜 주었다. 따라서 예매를 하게 되면 날짜는 20220401로 들어간다.
5. 여석이 남아있지 않으면 예매를 하는 기능은 불가능하다.
6. 회원 기능을 사용할 때 처음에 먼저 ID를 입력하여 로그인 해주지 않으면, 여러 기능들의 이용에 있어서 제한이 된다.
7. II. 요구사항 분석파트에서 요구사항에 대하여 분석한 내용과 요구사항 이외에 추가적으로 적용한 부분을 반드시 고려해주어야 한다.
8. 회원 테이블에 정보가 저장된 회원만 회원기능을 이용할 수 있다. INSERT 문을 이용하여 회원ID는 관리자에서 추가할 수 있다.

IV. 스키마 정의

Mysql의 EER Diagram 도구를 활용하여 구현한 스키마는 다음과 같습니다.



2의 요구사항 분석내용에서 분석 후 추가적으로 고려해준 부분들을 추가하여 스키마를 다음과 같이 구현하였습니다.

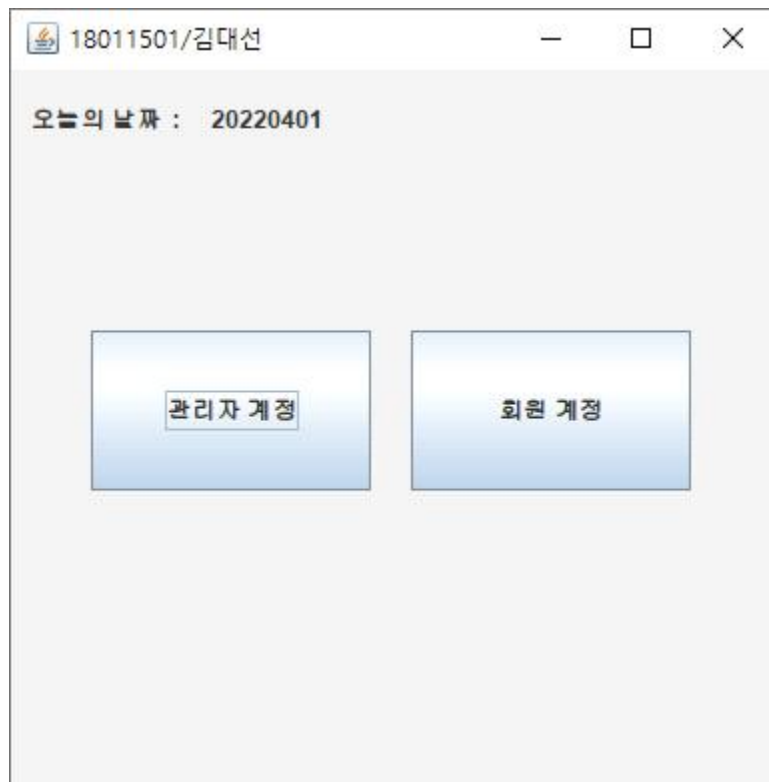
Mysql의 EER Diagram 도구를 활용하여 구현한 스키마에 넣을 샘플들의 INSERT문은 압축파일에 첨부해주었습니다. 또한 EER Diagram 도구를 통하여 구현한 스키마 파일 또한 압축파일에 첨부해주었습니다.

V. 프로그램 구조 및 프로그램 테스트

메인 메소드를 포함한 모든 메소드는 처음 생성해준 DBproject라는 클래스 내부에 포함되어 있습니다. DBproject라는 클래스의 생성자에서는 자바와 SQL을 연동해주는 코드를 실행해줍니다. 해당 코드는 중간고사때 나온 코드와 동일한 코드입니다. 코드의 제일 마지막에 나오는 메인 메소드를 통하여 MyFrame이라는 메소드를 실행시켜줍니다.

```
1901 public static void main(String args[]) { // 메인함수
1902     DBproject dbp = new DBproject();
1903     dbp.MyFrame();
```

MyFrame을 실행하면 다음과 같은 window창을 나타나게 됩니다.

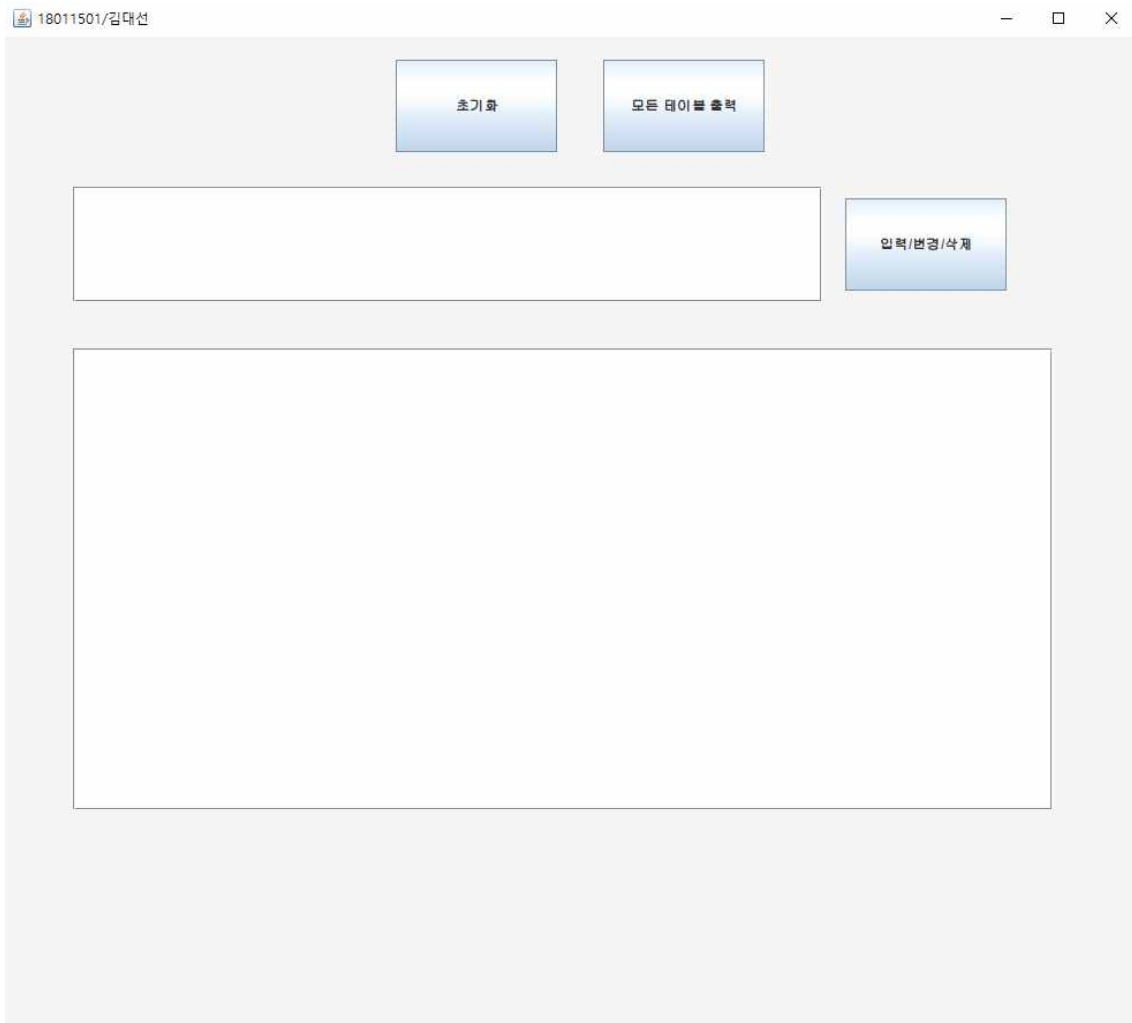


위 창에는 문자열 today에 저장된 오늘의 날짜 정보와 관리자 모드로 실행을

시킬지 , 회원모드로 실행시킬지를 고를 수 있는 버튼이 나옵니다.

관리자 계정버튼을 클릭하게 되면 리스너가 해당 이벤트를 받아서 MyManagerFrame이라는 메소드를 실행시켜주고, 회원 계정 버튼을 클릭하게 되면 리스너가 해당 이벤트를 받아서 MyClientFrame을 실행시켜주게 됩니다.

MyMangerFrame메소드를 실행하면 다음과 같은 윈도우창이 나타납니다.



초기화 버튼을 클릭하면 drop table if exists create table insert into를 통하여 구현된 SQL 문장들이 실행되어서 테이블이 모두 삭제되고 다시 테이블이 만들어지고 다시 값들이 삽입되어서 프로그램의 데이터베이스가 초기상태로 되돌려지게 됩니다. 초기화에 성공하면 하단의 큰 JTextarea창에 초기화 성공 안내메세지가 나타납니다.

입력/변경/삭제 버튼 옆에 있는 JTextField창에 ~한 조건식을 포함하는 쿼리문을 입력하고 입력/변경/삭제 버튼을 눌러주면 해당 쿼리문이 실행되어 원하는 입력/변경/삭제가 이루어지게 됩니다. 그러나 잘못된 내용을 입력하여주게

되면, 예외처리가 발생하고 하단의 큰 JTextarea창에 경고메세지가 나옵니다. 초기화 버튼 옆에 있는 모든 테이블 출력을 누르면 '전체 테이블 보기' 기능을 이용할 수 있습니다. 하단의 큰 JTextarea창에 모든 테이블에 담긴 모든 내용이 출력됩니다.

The screenshot shows a web application interface. At the top, there are two buttons: '초기화' (Initialize) and '모든 테이블 출력' (Output all tables). Below these buttons is a large text area (JTextArea) and a button labeled '입력/변경/삭제' (Input/Modify/Delete). The JTextArea contains the following message: '데이터 베이스를 초기화하였습니다. 초기화된 내용을 확인해보려면 모든 테이블 출력버튼을 눌러주세요.' (The database has been initialized. To check the initialized content, please click the output all tables button.)

초기화 성공 시 JTextArea에 메시지가 나타난 모습이다. 오류가 나서 예외처리가 되면 다른 메시지가 나타난다.

The screenshot shows the same web application interface. The '입력/변경/삭제' button has been clicked, and the JTextArea now displays the message: '적절하지 않은 입력값입니다.' (Invalid input value.).

TextField에 올바르지 않은 문장을 입력하고 입력/변경/삭제 버튼을 누른 모습이다. JTextArea에 경고메세지가 나타난다.

The screenshot shows the same web application interface. The '입력/변경/삭제' button has been clicked, and the JTextArea now displays the SQL update statement: 'UPDATE movie_info SET movie_name = "범죄도시1" where movie_id = 1;'. This indicates a successful update operation.

해당 변경문을 입력해준다.

초기화
모든 테이블 출력

UPDATE movie_info SET movie_name = "범죄도시1" where movie_id = 1;

입력/변경/삭제

movie_info table							
1	범죄도시1	106	15	이상용	마동석		
2	닥터스트레인지2		126	12	샘 레이미	베네딕트	
3	스파이더맨	148	12	존 왓츠	톨 홀랜드		
4	블랙위도우	134	12	케이트 쇼트랜드	스칼릿 조핸슨		
5	귀멸의칼날	117	15	소토자키 하루오	이경태		
6	배놓2	97	15	앤디 서키스	톨 하디		
7	크루엘라	133	12	크레이그 질레스피		엠마 스	
8	반도	116	15	연상호	강동원		
9	국한직업	111	15	이병헌	류승룡		
10	겨울왕국2	103	0	크리스 벅	크리스틴 벨		

모든 테이블 출력 시 영화 범죄도시2가 범죄도시 1로 바뀌어 있다.

초기화
모든 테이블 출력

DELETE from movie_info where movie_id = 1;

입력/변경/삭제

적절하지 않은 입력값입니다.

TextField에 movie_id=1인 영화 정보를 삭제하는 쿼리문을 입력하고 입력/변경/삭제버튼을 입력한 모습이다. 아래키로 상영일정테이블과 묶여있기 때문에 삭제되지 않고 경고메세지가 나타나게 된다.

초기화
모든 테이블 출력

into movie_info values(11, "겨울왕국", 108, 0, "크리스 벅", "크리스틴 벨", "애니메이션", "디즈니 애니메이션", "20140116");

입력/변경/삭제

movie_info table							
1	범죄도시1	106	15	이상용	마동석		
2	닥터스트레인지2		126	12	샘 레이미	베네딕트	
3	스파이더맨	148	12	존 왓츠	톨 홀랜드		
4	블랙위도우	134	12	케이트 쇼트랜드	스칼릿 조핸슨		
5	귀멸의칼날	117	15	소토자키 하루오	이경태		
6	배놓2	97	15	앤디 서키스	톨 하디		
7	크루엘라	133	12	크레이그 질레스피		엠마 스	
8	반도	116	15	연상호	강동원		
9	국한직업	111	15	이병헌	류승룡		
10	겨울왕국2	103	0	크리스 벅	크리스틴 벨		
11	겨울왕국	108	0	크리스 벅	크리스틴 벨		

INSERT문을 TextField에 입력하고 입력/변경/삭제 버튼을 입력한 모습이다.

MyClientFrame메소드를 실행하면 다음과 같은 윈도우창이 실행된다.

우선 회원아이디에 회원의 아이디를 입력하고 로그인을 하여야 한다. 로그인 버튼 클릭 시 이벤트가 발생하여 TextField에 쓰여진 내용을 clientID라는 스트링에 저장한다. 예를들어 회원아이디 1로 로그인을 하면 다음과 같은 메시지가 우측의 JTextArea에 나타난다.

영화 검색 기능은 각 영화명, 감독명, 배우명, 장르 정보가 쿼리문에서 and로 묶여 정보를 받아 JTable로 나타내어 주었다. 아무것도 입력하지 않고 검색하면 모든 영화가 조회된다. 원하는 정보를 입력한 후 영화조회하기버튼을 누르면 movietable메소드가 실행된다. movietable 메소드가 실행되면 다음과 같은 윈도우창이 나타난다. 다음은 아무 정보도 입력하지 않고 영화조회하기를 했을 때 창이 나타난 모습이다.

[illegible]

만약에 장르에 판타지만 입력하고 영화를 검색하면 다음과 같이 조회된다.

영화명 : 감독명 : 배우명 : 장르 : 판타지

movie_id	movie_na...	movie_time	movie_age	director_na...	actor_name	genre	movie_intro	release_da...	booking?
2	닥터스트레...	126	12	샘 레이미	배네딕트...	판타지	마블영화	20210204	예매
3	스파이더맨	148	12	조 핫츠	톰 홀랜드	판타지	마블영화	20210329	예매
6	배틀2	97	15	앤디 서키스	토 하디	판타지	마블영화	20210624	예매

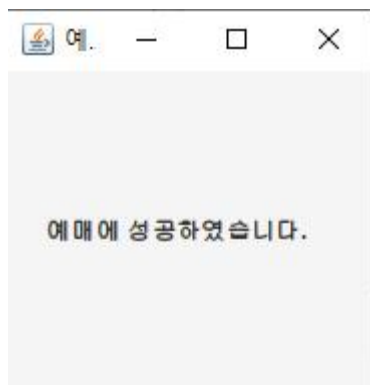
만약에 장르에 판타지 배우명에 톰 홀랜드를 입력하면 다음과 같이 조회된다.

영화명 : 감독명 : 배우명 : 톰 홀랜드 장르 : 판타지

movie_id	movie_na...	movie_time	movie_age	director_na...	actor_name	genre	movie_intro	release_da...	booking?
3	스파이더맨	148	12	조 핫츠	톰 홀랜드	판타지	마블영화	20210329	예매

movietable메소드로 나타난 윈도우창에서 booking?이라는 항목 밑에 나온 예매버튼을 누르면 해당 영화를 예매할 수 있다. TableCell이라는 클래스를 만들어 `jt.getColumnModel().getColumn(9).setCellRenderer(new TableCell());`

`jt.getColumnModel().getColumn(9).setCellEditor(new TableCell());`라는 코드를 넣어서 해당 버튼을 추가해주었다. TableCell이라는 클래스의 내용의 일부를 보면, 생성자 내부에 JButton jb;를 통하여 jb라는 필드를 선언해주고 jb에 리스너를 달아주어서 이벤트 발생시에, 해당 영화의 정보를 가져와서 예매할 수 있도록 만들어 주었다. `int row = jt.getSelectedRow(); movienum = data[row][0];` 라는 코드를 통하여 영화의 정보를 가져올 수 있도록 하였다. 해당영화의 정보로 해당 상영관의 정보를 찾고 해당 상영관의 좌석이 남아있는지 확인해준 후에, if문을 걸어 좌석이 있으면 예매정보를 추가하고 배정해준 좌석의 사용여부를 0로 바꿔주고 배정받은 티켓의 발권여부를 0로 바꿔주고 새로운 윈도우 창을 통하여 예매 성공을 알린다. 만약에 좌석이 없으면 예매할 수 없다는 경고메세지를 새로운 윈도우창을 통해 나타낸다.



예매에 성공하면 다음과 같은 메시지가 뜬다. 예매가 잘 되었는지는 바로 하단의 예매한 영화 정보 조회 버튼을 클릭하여 확인해줄 수 있다. 해당 버튼을 클릭하면 bookingchecking메소드를 실행시켜 윈도우창을 나타나게 한다. 해당

[illegible][illegible]

이때 자세한 정보를 누르면 ‘위에서 표시된 예매 정보 중에서 하나를 “클릭”하면 해당 예매에 대해서 모든 상영일정, 상영관, 티켓에 대한 정보를 보여주는 기능’을 사용할 수 있다. 자세한 정보 버튼을 클릭하면 MoreinformationFrame

이라는 메소드를 실행하게 되는데 이 메소드는 새로운 윈도우창을 띄워 해당 영화의 자세한 정보들을 표시하여 준다. 예매해서 했던 원리와 비슷하게 해당 정보를 받아서 상영일정 상영관 티켓 테이블에 관한 정보를 쿼리문을 통해 가져오고 해당 정보를 새로운 윈도우창의 JTextArea에 나타나게한다.



극한직업 영화의 자세한 정보를 클릭하여 나타난 창이다.



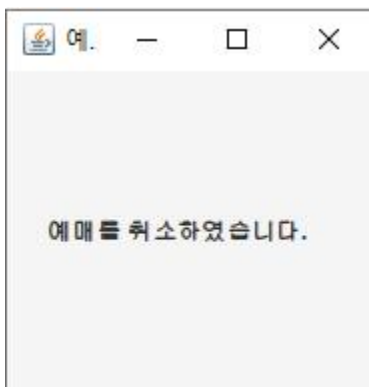
베놈2 영화의 자세한 정보를 클릭하여 나타난 창이다.

그 다음은 예매영화 조회 후 예매 취소 버튼에 대해서이다. 이 버튼을 클릭시 deletebookingmovie라는 메소드가 실행된다. 이 메소드는 예매한 영화정보라는 윈도우창을 실행시키는데, 위에서 구현한 예매한 영화정보 윈도우와 동일한 내용으로 구성하였다. 대신 제일 우측의 Column을 자세한 정보가 아니라 예매 정보삭제 버튼으로 바꾸고 기능을 바꾸어 삭제기능을 구현하였다.

movie_name	screening schedule	theater_id	seat_id	selling_price	delete booking?
극한직업	111	1	2	14000	예매정보 삭제
범죄도시2	106	7	19	14000	예매정보 삭제
블랙위도우	134	6	17	14000	예매정보 삭제
베놈2	97	8	22	14000	예매정보 삭제
					예매정보 삭제
					예매정보 삭제
					예매정보 삭제
					예매정보 삭제
					예매정보 삭제
					예매정보 삭제
					예매정보 삭제
					예매정보 삭제
					예매정보 삭제
					예매정보 삭제
					예매정보 삭제
					예매정보 삭제

예매 영화 조회 후 예매 취소 버튼을 클릭하면 다음과 같은 창이 나타난다. 예매정보 삭제버튼을 누르면 해당영화가 가지고 있던 티켓id 정보를 활용하여 삭제를 진행한다. 우선 해당 좌석테이블의 좌석 사용여부를 X로 바꿔주고 외래키의 원활한 삭제를 위하여 SET foreign_key_checks = 0를 이용한다. 해당 예매정보를 지워준다. 그리고 SET foreign_key_checks = 1을 통하여 다시 원래 상태로 돌려주고, 해당 티켓에 가서 발권 여부를 X로 예매번호를 NULL 바꿔준다. 여기서 묶여있는 해당 내용들을 모두 정상적으로 처리하였으므로 SET foreign_key_checks = 0 문장을 사용해주어도 전혀 문제가 되지 않는다.

예매정보삭제 버튼을 누르면 다음과 같은 윈도우 창이 나타나서 삭제가 되었는지 알려준다.



범죄도시2를 삭제하여준 모습이다. 이 때 창을 닫고 다시 예매영화 조회 후 예매 취소 버튼을 눌러주면 범죄도시2라는 영화가 예매 목록에서 삭제되었는지 확인해줄 수 있다.

예매한 영화 정보

movie_name	screening schedule	theater_id	seat_id	selling_price	delete booking?
극한직업	111	1	2	14000	예매정보 삭제
블랙위도우	134	6	17	14000	예매정보 삭제
베놈2	97	8	22	14000	예매정보 삭제
					예매정보 삭제
					예매정보 삭제
					예매정보 삭제
					예매정보 삭제
					예매정보 삭제
					예매정보 삭제
					예매정보 삭제
					예매정보 삭제
					예매정보 삭제
					예매정보 삭제
					예매정보 삭제
					예매정보 삭제
					예매정보 삭제

깔끔하게 삭제되어 있는 모습이다.

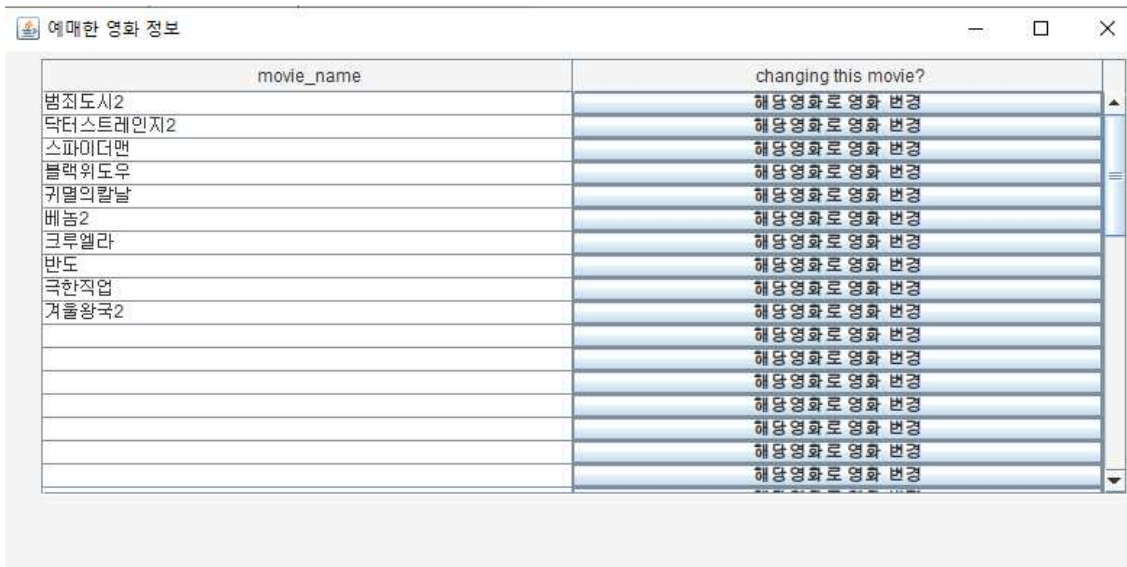
예매영화 조회 후 다른영화로 예매 변경 버튼을 클릭하면 changingbookingmovie 메소드를 실행시킨다. 이 메소드를 실행시키면 예매한 영화정보들을 조회할 수 있다. 편의성을 위해서 위의 3번기능(예매한 영화 정보 조회)과 5번기능(예매영화 조회 후 예매취소)에서와 같은 창으로 구현하고 버튼만 예매정보변경으로 바꾸어주었다.

예매한 영화 정보

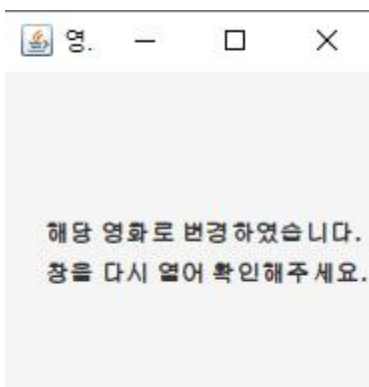
movie_name	screening schedule	theater_id	seat_id	selling_price	changing booking?
극한직업	111	1	2	14000	예매정보 변경
블랙위도우	134	6	17	14000	예매정보 변경
베놈2	97	8	22	14000	예매정보 변경
					예매정보 변경
					예매정보 변경
					예매정보 변경
					예매정보 변경
					예매정보 변경
					예매정보 변경
					예매정보 변경
					예매정보 변경
					예매정보 변경
					예매정보 변경
					예매정보 변경
					예매정보 변경
					예매정보 변경

예매정보 변경버튼을 클릭하면 moviecategory 메소드를 실행시키고 해당 메소드는 다음과 같은 윈도우창을 나타낸다. 영화 테이블에 저장되어있는 모든 영화를 나타내고 해당영화로 변경할 지에 대한 버튼이 나타난다.

다음은 블랙위도우 영화의 예매정보 변경 버튼을 클릭하여 나타난 창이다.

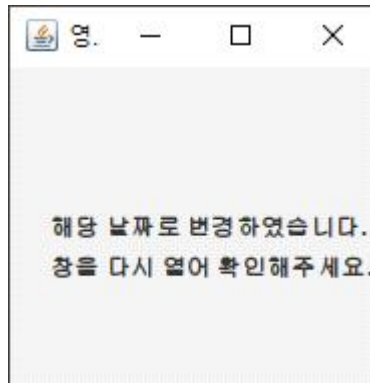


이 때 해당영화로 영화 변경 버튼을 누르면 해당영화의 정보를 가져와서 해당 영화의 정보를 가진 상영관의 좌석중에서 좌석 사용 여부가 X인 좌석 정보를 찾아와준다. 이 때 만약에 해당영화와 연결된 상영관의 좌석중에서 좌석 사용 여부가 X인 좌석이 한 개도 없다면 해당영화로 예매를 변경할 수 없기 때문에, 새로운 윈도우창에서 해당영화는 여석이 없습니다라는 경고메세지를 출력시켜주고, 그 다음 쿼리문들은 실행하지 않는다. 먼저 해당좌석의 사용여부를 O로 바꾸어주고 기존 티켓이 가지고 있던 좌석의 사용여부를 X로 바꾸어준다. 그리고 새로 찾은 좌석과 연결된 티켓 정보를 가져와서 기존 티켓의 상영일정 번호, 상영관번호, 좌석번호와 새로 찾은 좌석과 연결된 티켓의 해당 값들을 따로 저장해두었다가 서로 값들을 바꾸어준다. 그러면 해당영화로 영화가 변경된다. 다음은 스파이더맨으로 영화를 변경하였을 때에 예매영화 조회 후 다른 영화로 예매변경 버튼을 클릭하였을 때 나타난 화면이다.

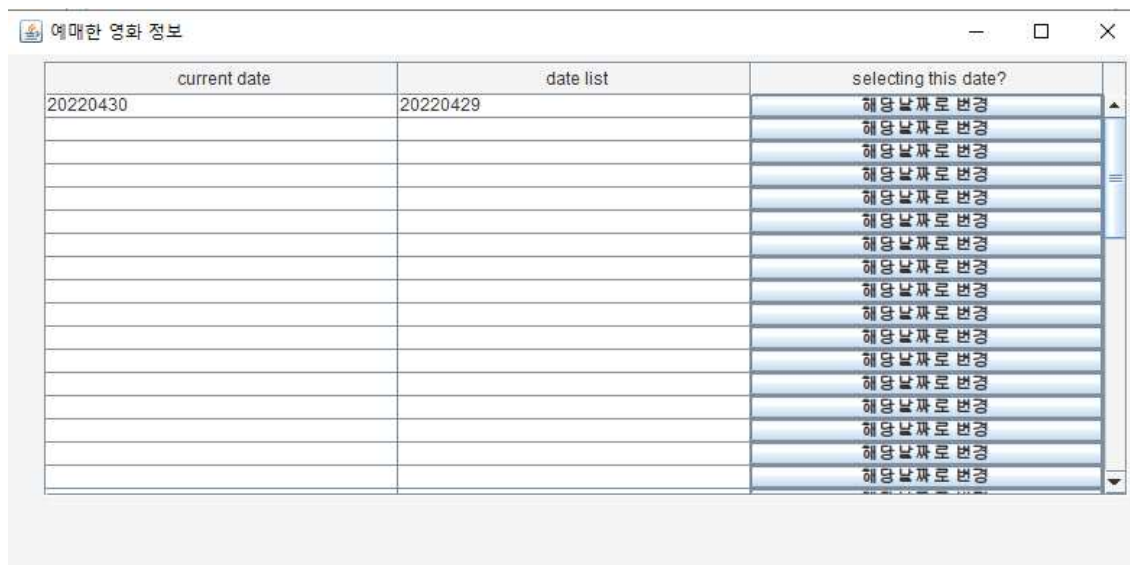


먼저 해당영화로 변경하는 일이 성공하면, 해당영화로 변경하였다는 새로운 윈도우창을 통하여 메시지를 나타낸다. 기존의 창을 닫고 다시 확인하여주면, 제대로 바뀌어 있는 모습을 볼 수 있다.

X인 좌석을 가져온다. 해당 좌석과 연결되어있는 티켓도 가져온다. 그리고 해당 티켓과 연결되어 있는 상영일정번호, 상영관번호, 좌석번호를 가져온다. 기존 좌석은 좌석사용여부를 X로 바꿔주고 새로 받을 좌석의 좌석사용여부를 O로 만들어준다. 티켓정보에서 두 티켓의 저장해두었던 서로의 좌석번호, 상영일정번호를 서로 바꿔준다. 그러면 날짜 변경이 완료된다.



해당 날짜로 변경 버튼을 눌렀을 때 메시지가 나타난 모습이다.



창을 닫고 다시 상영일정변경 버튼을 눌러주면 현재날짜가 바뀌어 있고 바꿀 수 있는 날짜 또한 바뀌어 있다.