

〈SW 개발 문서〉

영화 예매 프로그램

팀 원 : 심무성 윤수한
오창현 권선미
김유창 이지태
임성수 장성혁
최윤진 하연서

〈윈도우 플랫폼 기반 IoT 시스템 개발자 양성과정〉

목 차

1. 개발 상세	1
1) 프로젝트 개요	1
2) 프로젝트 대상	1
3) 프로젝트 일정	1
4) 프로젝트 인원	3
5) 프로젝트 환경	4
2. 개발 기초 상세	5
1) DB Table	5
2) Data Process Flow Chart	6
3. 프로그램 상세 설명	8
1) 로그인	8
2) 회원가입	9
3) 메인메뉴	10
4) 예매	10
5) 인원 및 좌석 선택	11
6) 예매정보 확인	13
7) 매점	14
8) 결제	15
9) 예매 완료	15
10) 예매 조회	16
11) 매출	18
12) 창고재고	20
13) 매장재고	22
4. 결과	25
1) 문제리스트	25
2) 아쉬운 점	26
3) 배운 점	27
4) 느낀 점	27
5) 향후 공부 방향	27
5. 동영상	28

1 개발 상세

1) 프로젝트 개요

- (1) Visual Studio에서 제공하는 Windows form을 활용하여 프로그램 개발
- (2) MS-SQL과 연동하여 DB를 효율적으로 관리하는 방법을 이해
- (3) 지금까지 공부해온 C#과 SQL Query문을 이용해 프로젝트에 적용하며 직접 이해
- (4) 소켓 통신을 이용하여 간단한 TCP / IP 통신 구축하여 네트워크 이해도 향상 목표
- (5) Github를 이용해 팀 프로젝트 간 통합성 및 연계성 증진

2) 프로젝트 대상

- (1) C# Windows Form과 DataBase를 활용할 수 있는 프로그램 제작.
- (2) 실생활에서 접할 수 있는 영화 예매 프로그램을 개발하여 프로그램에 대한 이해를 높이기로 함.
- (3) 통신을 이용해 하나의 컴퓨터 SQL Server에 저장된 Data를 다른 Client PC에서 조회 및 추가 / 삭제가 가능한 프로그램 작성.
- (4) 단순한 예매 프로그램보다는 배워왔던 ERP와 연동하기 위해 관리자 쪽에 매출 같은 회계적인 부분과 재고, 발주 등의 생산관리 프로세스를 추가

3) 프로젝트 일정

(1) 프로젝트 전

- ① 각 팀원 C#과 SQL에 대한 이해도 향상을 위해 주교재, 부교재를 통한 학습
- ② 정규수업과정에서 필요시 보충수업을 통한 이해노력
- ③ 회의를 통해 구현 할 부분에 대해 요구사항 및 우선순위 부여
- ④ UI 화면과 DB 설계, 총합 프로세스, 디자인을 구분하여 개인에게 역할 부여
- ⑤ 기간이 짧기 때문에 단계별 개발을 진행하여 1단계 프로젝트 완성 후 동작 확인이 완료되면 2단계 기능을 구현

(2) 프로젝트 단계

- ① 5월 19 ~ 20일 : 전반적인 프로젝트 주제를 정하고 관련 UI 및 DB 설계
- ② 5월 21 ~ 26일 : 1단계 기능 구현, 프로젝트를 진행하면서 기초 UI, DB에서 부족했던 부분을 회의를 통해 보강.
- ③ 5월 26 ~ 27일 : GitHub를 통해 전반적인 프로젝트 통합, 개발 1단계를 끝내지 못한 인원을 제외하고 2단계 진행. 마무리 후 디자인, 문서, 동영상 작업진행.

(3) 개발 후

- ① 개발 프로그램에 관한 후속조치, 미흡했던 부분 보완
- ② 개인적인 공부 방향 설정, 차후 최종 프로젝트에 대한 토의

(3) 전체 개발 일정

개발 1단계	개발 2단계
<ul style="list-style-type: none"> · UI 구성 · 로그인 기능 · 예매정보 확인(예매번호) 기능 · 상영관 및 시간 선택 기능 · 좌석 및 인원 선택 기능 · 예매완료 화면 구현 · 결제 기능 · 재고 및 발주 기능 · 매출 확인 기능 	<ul style="list-style-type: none"> · 회원가입 기능 · 예매정보 확인(전화번호) 기능 · SQL - Client간 통신 구현 · 상영관 및 시간선택(Form4) / 좌석 및 인원 선택(Form5)에 관한 추가기능 구현 · 매출화면에 그래프표시 · 디자인

		5/19	5/20	5/23	5/24	5/25	5/26	5/27
개발 계획	계획수립							
	파트분배							
설계 및 구현	UI구성							
	DB구성							
	로그인화면							
	예매조회							
	상영관 / 시간 선택							
	좌석 / 인원 선택							
	결제화면							
	매출화면							
	재고화면							
	발주화면							
	디자인							
	문서작성							

4) 프로젝트 인원

	개발 1단계	개발 2단계
심무성(팀장)	전반적인 프로젝트 지휘 재고화면 구현 발주화면 구현	전반적인 프로젝트 지휘 발주화면 구현 동영상편집
윤수한	로그인 기능 구현 예매정보 확인(예매번호)	회원가입 기능 구현 예매정보 확인(전화번호) 문서 작성
오창현	매점기능 구현	매점기능 구현 디자인 설계
권선미	좌석 및 인원 선택화면 상영관 및 시간 선택화면 UI 구성	좌석 및 인원 선택화면 상영관 및 시간 선택화면 디자인 설계
김유창	SQL 관리	SQL 관리 디자인 설계
이지태	결제화면, 예매완료화면 구현 GitHub관리	결제화면, 예매완료화면 구현 SQL Server - Client간 통신 구현, GitHub 관리
임성수	매출화면 구현	
장성혁	매점기능 구현	매점기능 구현 디자인 설계
최윤진	총 프로세스 관리 UI 구성	총 프로세스 관리 디자인 설계
하연서	SQL 관리	좌석 및 인원 선택화면 상영관 및 시간 선택화면

5) 프로젝트 환경



Visual Studio 2010



Microsoft SQLServer Management



한글 2010



PhotoShop



GitHub



Guna FramWork

〈그림 1〉 개발 프로젝트 환경

2 개발 기초 상세

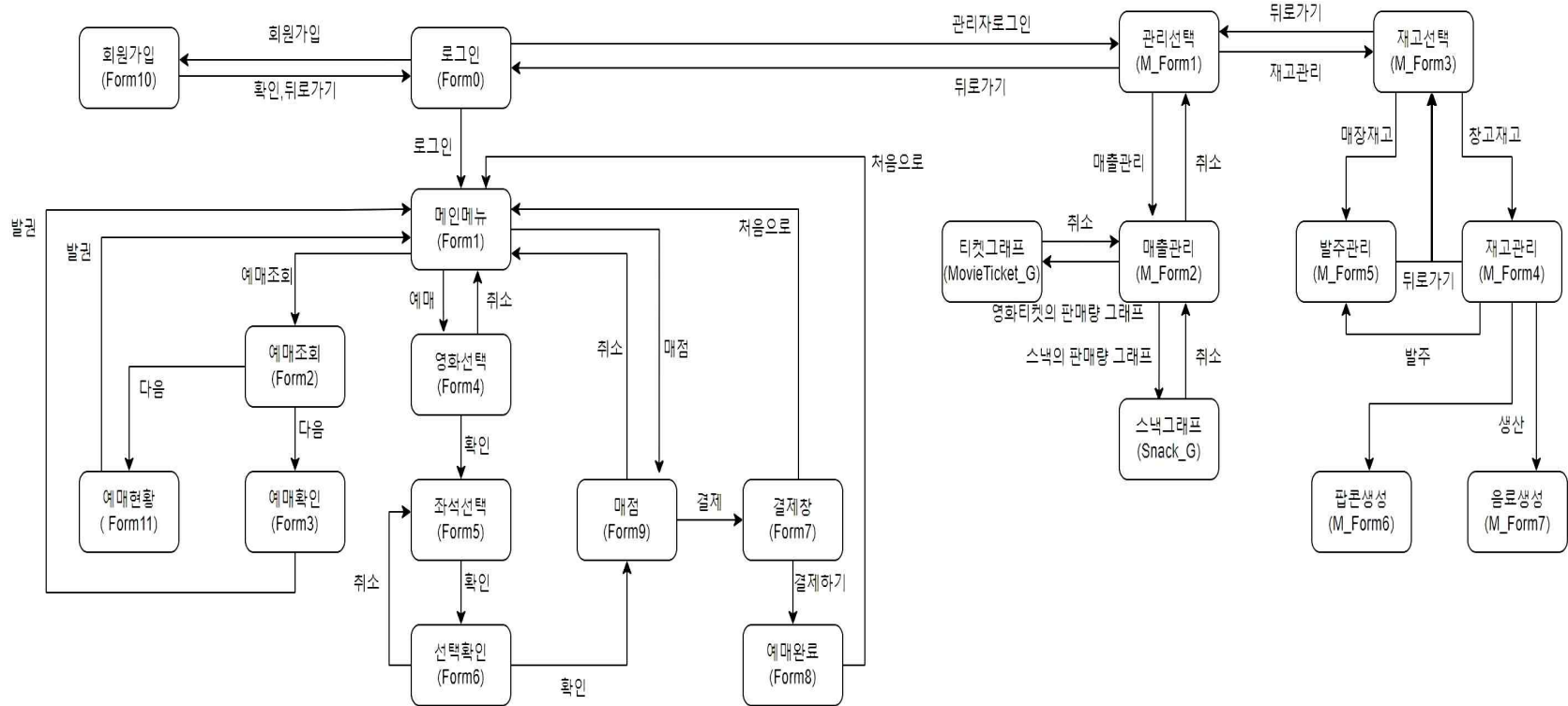
1) DB Table

<table><tr><th colspan="2">Movie Table</th></tr><tr><td>MvName</td><td>NVARCHAR(20)</td></tr><tr><td>Hall</td><td>INT</td></tr><tr><td>StartTime</td><td>TIME</td></tr><tr><td>RunningTime</td><td>TIME</td></tr></table> <p>영화 테이블</p>	Movie Table		MvName	NVARCHAR(20)	Hall	INT	StartTime	TIME	RunningTime	TIME	<table><tr><th colspan="2">Theater Table</th></tr><tr><td>ThNum</td><td>INT NOT NULL</td></tr><tr><td>SeatNum</td><td>VARCHAR(10)</td></tr><tr><td>Eempty</td><td>INT</td></tr></table> <p>상영관 테이블</p>	Theater Table		ThNum	INT NOT NULL	SeatNum	VARCHAR(10)	Eempty	INT	<table><tr><th colspan="2">Member Table</th></tr><tr><td>ID(PK)</td><td>VARCHAR(10) NOT NULL</td></tr><tr><td>PW</td><td>VARCHAR(20)</td></tr><tr><td>Phone</td><td>VARCHAR(20)</td></tr><tr><td>Who</td><td>INT</td></tr></table> <p>멤버 테이블</p>	Member Table		ID(PK)	VARCHAR(10) NOT NULL	PW	VARCHAR(20)	Phone	VARCHAR(20)	Who	INT																												
Movie Table																																																										
MvName	NVARCHAR(20)																																																									
Hall	INT																																																									
StartTime	TIME																																																									
RunningTime	TIME																																																									
Theater Table																																																										
ThNum	INT NOT NULL																																																									
SeatNum	VARCHAR(10)																																																									
Eempty	INT																																																									
Member Table																																																										
ID(PK)	VARCHAR(10) NOT NULL																																																									
PW	VARCHAR(20)																																																									
Phone	VARCHAR(20)																																																									
Who	INT																																																									
<table><tr><th colspan="2">Reservation Table</th></tr><tr><td>ID</td><td>VARCHAR(10)</td></tr><tr><td>RsvCode</td><td>VARCHAR(10)</td></tr><tr><td>MvName</td><td>NVARCHAR(20)</td></tr><tr><td>Hall</td><td>INT</td></tr><tr><td>SeatNum</td><td>VARCHAR(10)</td></tr><tr><td>StartTime</td><td>TIME</td></tr><tr><td>RunningTime</td><td>TIME</td></tr><tr><td>Ccount</td><td>INT</td></tr><tr><td>Mmoney</td><td>INT</td></tr></table> <p>예약 테이블</p>	Reservation Table		ID	VARCHAR(10)	RsvCode	VARCHAR(10)	MvName	NVARCHAR(20)	Hall	INT	SeatNum	VARCHAR(10)	StartTime	TIME	RunningTime	TIME	Ccount	INT	Mmoney	INT	<table><tr><th colspan="2">TmpReservation Table</th></tr><tr><td>ID</td><td>VARCHAR(10)</td></tr><tr><td>MvName</td><td>NVARCHAR(20)</td></tr><tr><td>Hall</td><td>INT</td></tr><tr><td>SeatNum</td><td>VARCHAR(10)</td></tr><tr><td>StartTime</td><td>TIME</td></tr><tr><td>RunningTime</td><td>TIME</td></tr><tr><td>Ccount</td><td>INT</td></tr><tr><td>Mmoney</td><td>INT</td></tr></table> <p>예약(임시) 테이블</p>	TmpReservation Table		ID	VARCHAR(10)	MvName	NVARCHAR(20)	Hall	INT	SeatNum	VARCHAR(10)	StartTime	TIME	RunningTime	TIME	Ccount	INT	Mmoney	INT	<table><tr><th colspan="2">Crjo Table</th></tr><tr><td>MvName</td><td>NVARCHAR(20)</td></tr><tr><td>Hall</td><td>INT</td></tr><tr><td>StartTime</td><td>Time(7)</td></tr><tr><td>RunningTime</td><td>Time(7)</td></tr><tr><td>MvNum</td><td>VARCHAR(10)</td></tr><tr><td>ThNum</td><td>INT</td></tr><tr><td>SeatNum</td><td>VARCHAR(20)</td></tr><tr><td>Eempty</td><td>INT</td></tr></table> <p>크로스조인 테이블</p>	Crjo Table		MvName	NVARCHAR(20)	Hall	INT	StartTime	Time(7)	RunningTime	Time(7)	MvNum	VARCHAR(10)	ThNum	INT	SeatNum	VARCHAR(20)	Eempty	INT
Reservation Table																																																										
ID	VARCHAR(10)																																																									
RsvCode	VARCHAR(10)																																																									
MvName	NVARCHAR(20)																																																									
Hall	INT																																																									
SeatNum	VARCHAR(10)																																																									
StartTime	TIME																																																									
RunningTime	TIME																																																									
Ccount	INT																																																									
Mmoney	INT																																																									
TmpReservation Table																																																										
ID	VARCHAR(10)																																																									
MvName	NVARCHAR(20)																																																									
Hall	INT																																																									
SeatNum	VARCHAR(10)																																																									
StartTime	TIME																																																									
RunningTime	TIME																																																									
Ccount	INT																																																									
Mmoney	INT																																																									
Crjo Table																																																										
MvName	NVARCHAR(20)																																																									
Hall	INT																																																									
StartTime	Time(7)																																																									
RunningTime	Time(7)																																																									
MvNum	VARCHAR(10)																																																									
ThNum	INT																																																									
SeatNum	VARCHAR(20)																																																									
Eempty	INT																																																									
<table><tr><th colspan="2">BuySnack</th></tr><tr><td>ID</td><td>VARCHAR(10)</td></tr><tr><td>SnackName</td><td>NVARCHAR(10)</td></tr><tr><td>SnackNum</td><td>VARCHAR(10)</td></tr><tr><td>BuyPrice</td><td>INT</td></tr><tr><td>BuyCount</td><td>INT</td></tr><tr><td>SnackPrice</td><td>INT</td></tr></table> <p>매점 구매내역 테이블</p>	BuySnack		ID	VARCHAR(10)	SnackName	NVARCHAR(10)	SnackNum	VARCHAR(10)	BuyPrice	INT	BuyCount	INT	SnackPrice	INT	<table><tr><th colspan="2">TmpBuySnack</th></tr><tr><td>ID</td><td>VARCHAR(10)</td></tr><tr><td>SnackName</td><td>NVARCHAR(10)</td></tr><tr><td>SnackNum</td><td>VARCHAR(10)</td></tr><tr><td>BuyPrice</td><td>INT</td></tr><tr><td>BuyCount</td><td>INT</td></tr><tr><td>SnackPrice</td><td>INT</td></tr></table> <p>매점 구매내역(임시) 테이블</p>	TmpBuySnack		ID	VARCHAR(10)	SnackName	NVARCHAR(10)	SnackNum	VARCHAR(10)	BuyPrice	INT	BuyCount	INT	SnackPrice	INT	<table><tr><th colspan="2">Snack Table</th></tr><tr><td>SnackName</td><td>NVARCHAR(10)</td></tr><tr><td>SnackNum</td><td>VARCHAR(10)</td></tr><tr><td>SnackPrice</td><td>INT</td></tr></table> <p>매점 테이블</p>	Snack Table		SnackName	NVARCHAR(10)	SnackNum	VARCHAR(10)	SnackPrice	INT																				
BuySnack																																																										
ID	VARCHAR(10)																																																									
SnackName	NVARCHAR(10)																																																									
SnackNum	VARCHAR(10)																																																									
BuyPrice	INT																																																									
BuyCount	INT																																																									
SnackPrice	INT																																																									
TmpBuySnack																																																										
ID	VARCHAR(10)																																																									
SnackName	NVARCHAR(10)																																																									
SnackNum	VARCHAR(10)																																																									
BuyPrice	INT																																																									
BuyCount	INT																																																									
SnackPrice	INT																																																									
Snack Table																																																										
SnackName	NVARCHAR(10)																																																									
SnackNum	VARCHAR(10)																																																									
SnackPrice	INT																																																									
<table><tr><th colspan="2">Jaego</th></tr><tr><td>SnackName</td><td>NVARCHAR(10)</td></tr><tr><td>SnackNum</td><td>VARCHAR(10)</td></tr><tr><td>SnackCount</td><td>INT</td></tr></table> <p>재고 테이블</p>	Jaego		SnackName	NVARCHAR(10)	SnackNum	VARCHAR(10)	SnackCount	INT	<table><tr><th colspan="2">Balju</th></tr><tr><td>SnackName</td><td>NVARCHAR(10)</td></tr><tr><td>SnackNum</td><td>VARCHAR(10)</td></tr><tr><td>SnackCount</td><td>INT</td></tr><tr><td>EdongNum</td><td>INT</td></tr></table> <p>발주 테이블</p>	Balju		SnackName	NVARCHAR(10)	SnackNum	VARCHAR(10)	SnackCount	INT	EdongNum	INT																																							
Jaego																																																										
SnackName	NVARCHAR(10)																																																									
SnackNum	VARCHAR(10)																																																									
SnackCount	INT																																																									
Balju																																																										
SnackName	NVARCHAR(10)																																																									
SnackNum	VARCHAR(10)																																																									
SnackCount	INT																																																									
EdongNum	INT																																																									

〈그림 2〉 SQL DB Table

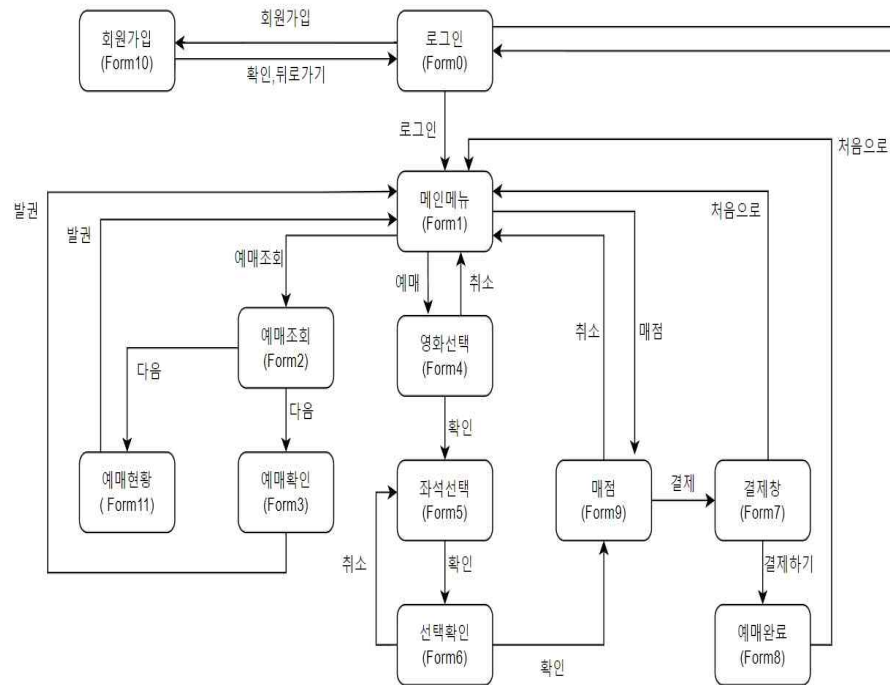
2) Data Process Flow Chart

① 전체 Data Process Flow Chart



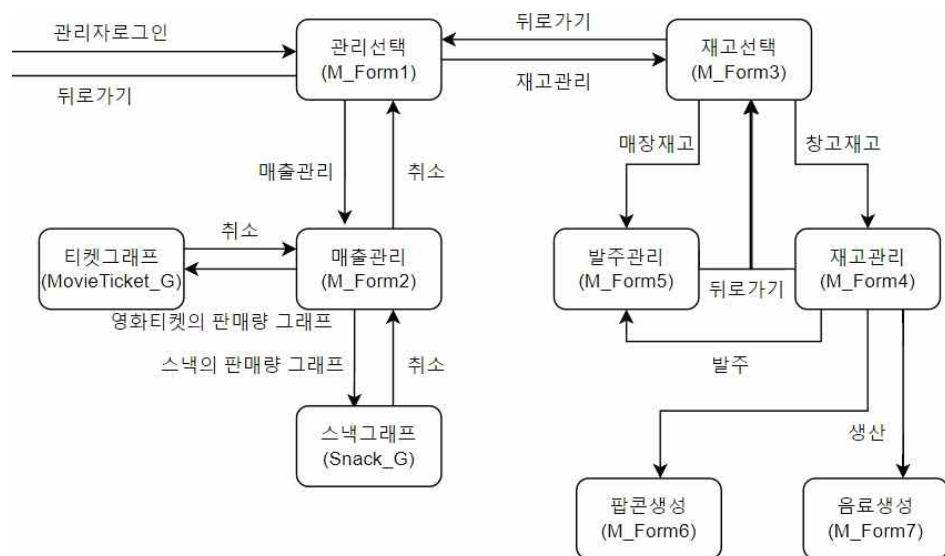
〈그림 3〉 전체 Data Process Flow Chart

② 소비자 Data Process Flow Chart



〈그림 4〉 소비자 Date Process Flow Chart

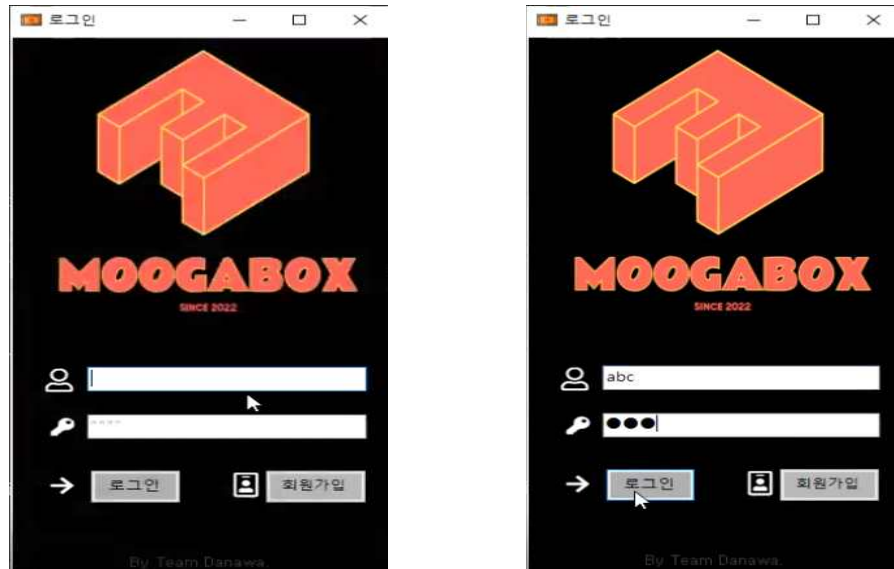
③ 생산자 Data Process Flow Chart



〈그림 5〉 생산자 Date Process Flow Chart

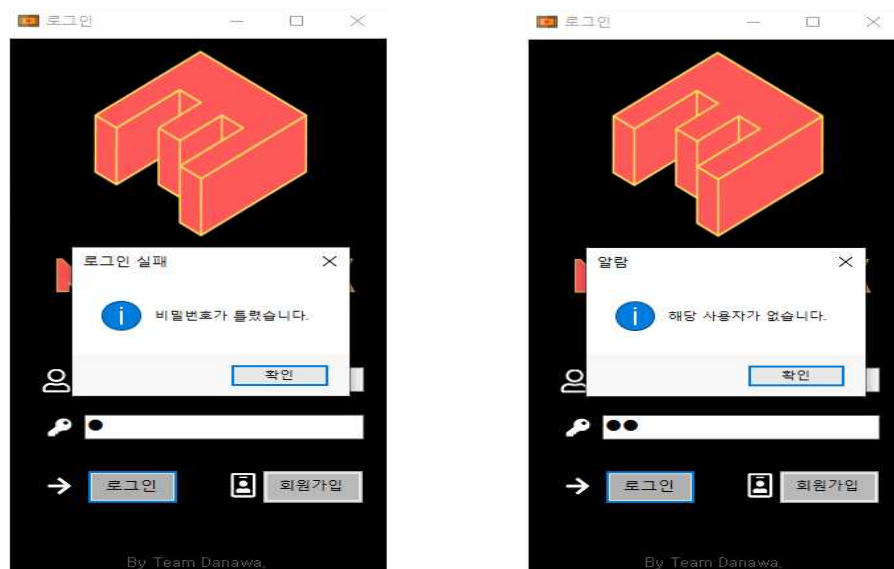
3 프로그램 상세 설명

1) 로그인



〈그림 6〉 로그인 화면 1

- 회원가입 된 ID가 있다면, ID와 Password를 치면 자동으로 로그인 됨.
- 로그인의 경우 관리자 / 일반 사용자의 2가지로 나뉘며 이걸 회원가입 메뉴에서 결정



〈그림 7〉 로그인 화면 2

- 사용자 ID가 있는 상태에서 비밀번호만 틀렸으면 '비밀번호가 틀렸습니다.' 문구 출력
- 아예 해당 ID가 없는 경우에는 '해당 사용자가 없습니다.'라는 문구가 출력

2) 회원가입

The image shows two side-by-side screenshots of a web application window titled '회원가입' (Sign Up). The window has a dark background with white text and red buttons. The left screenshot shows the registration form with empty input fields for ID, PW, and PhoneNum. Below the input fields are two radio buttons labeled '회원유형' (Member Type): '일반회원' (General Member) and '관리자' (Admin). The right screenshot shows the same form with pre-filled values: ID: abc, PW: ***, and PhoneNum: 010-2222-2222. The '관리자' radio button is selected. At the bottom of the form are two red buttons: '뒤로가기' (Go Back) and '확인' (Confirm). The MOOGABOX logo is visible at the bottom of the window.

〈그림 8〉 회원가입 화면 1

- 로그인 화면에서 회원가입을 클릭하면 나오는 ‘회원가입 폼’
- ID, PW, PhoneNum, 회원유형을 입력하고 확인을 클릭하면 회원가입이 완료

The image shows two side-by-side screenshots of the same web application window. The left screenshot shows an error message box titled '회원가입 실패' (Sign Up Failed) with a red 'X' icon and the text '중복되는 ID입니다' (ID is duplicated). The right screenshot shows a success message box titled '회원가입 성공' (Sign Up Success) with a blue 'i' icon and the text '회원가입이 정상적으로 처리되었습니다.' (Sign up was processed normally). Both screenshots show the registration form with pre-filled values: ID: lot (left) or abc (right), PW: ***, and PhoneNum: 010-2222-2222. The '관리자' radio button is selected. At the bottom of the form are two red buttons: '뒤로가기' (Go Back) and '확인' (Confirm). The MOOGABOX logo is visible at the bottom of the window.

〈그림 9〉 회원가입 화면 2

- 만약 중복되는 ID로 회원가입을 시도하면 ‘중복되는 ID입니다’라는 문구 출력
- 정상적으로 모든 입력을 완료하고 확인을 누르면 회원가입이 정상적으로 완료

3) 메인메뉴



〈그림 10〉 메인메뉴 화면

- 로그인을 통해 메인 메뉴(Form2)로 넘어오면 보이는 화면
- 각각의 버튼을 통해 예매조회, 예매, 매점 폼으로 넘어갈 수 있음.

4) 예매



〈그림 11〉 예매 화면 1

- 메인 화면에서 '예매' 버튼을 누르면 나오는 화면(Form4)
- 현재시간을 표시해 이전시간의 버튼들은 비활성화, 표시가 안 되도록 설계(빨간 부분)



〈그림 12〉예매 화면 2

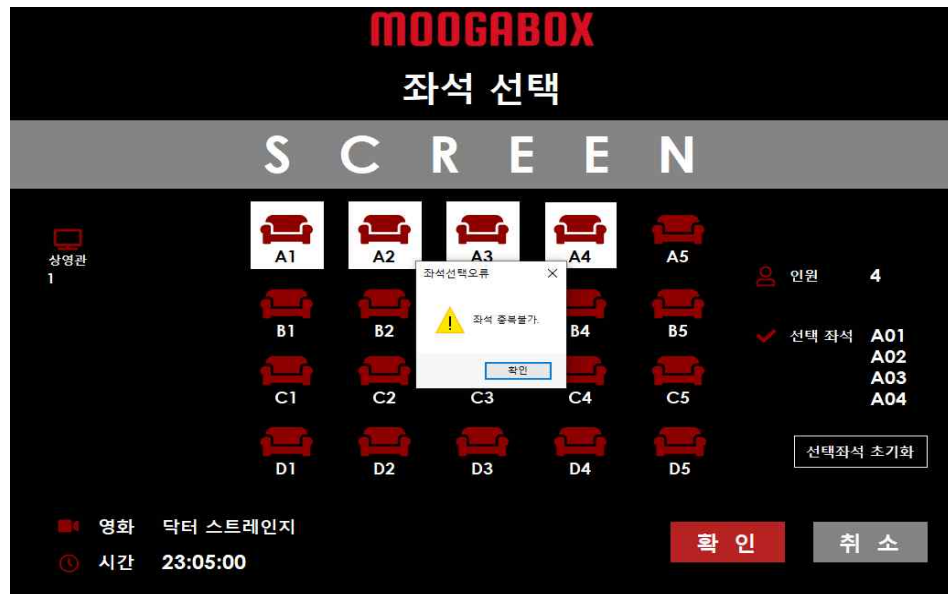
- 위의 〈그림 11〉에는 있었지만 실제 화면에선 삭제 된 상태로 보이게 구성.
- ‘확인’ 버튼을 누르면 인원 및 좌석 화면으로, 취소를 누르면 메인화면으로 넘어감.

5) 인원 및 좌석 선택



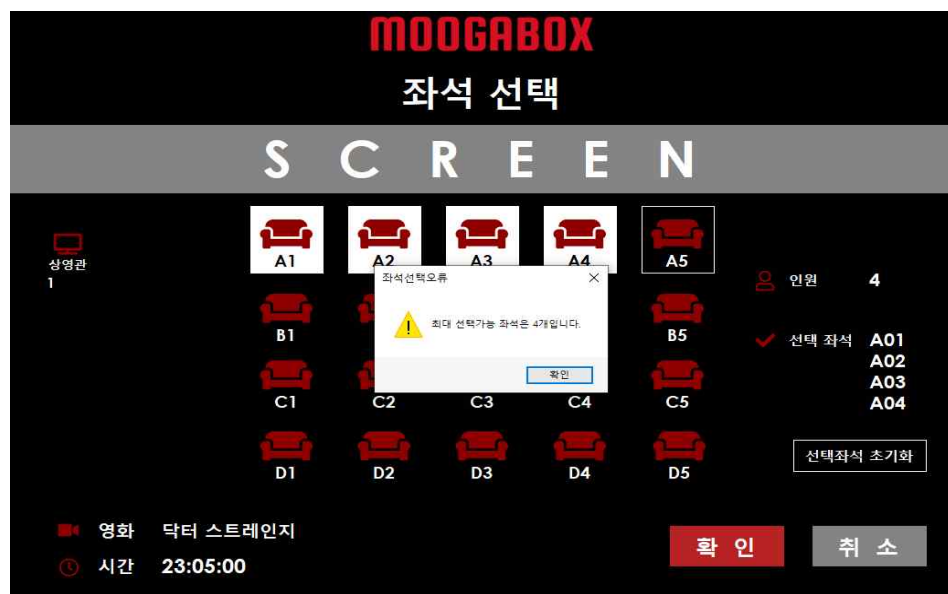
〈그림 13〉인원 및 좌석 선택 화면 1

- 인원 및 좌석 선택 화면으로 해당 좌석을 클릭하면 좌석이 클릭되며 오른쪽에 있는 인원이 클릭 횟수만큼 증가, 선택 좌석이 보이도록 설계하였음.
- ‘선택좌석 초기화’를 클릭 시 눌렀던 좌석, 인원, 선택좌석이 모두 초기화 되도록 설계.



〈그림 14〉 인원 및 좌석 선택 화면 2

- 중복된 좌석을 클릭한다면 Message Box로 중복되는 좌석을 클릭하지 못하도록 방지



〈그림 15〉 인원 및 좌석 선택 화면 3

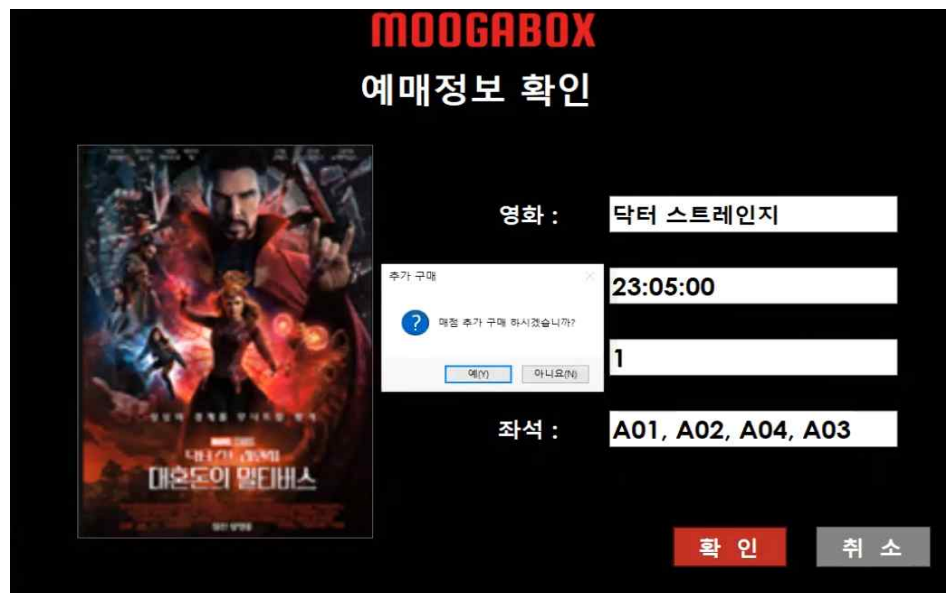
- 최대 선택가능 좌석을 4로 설정하여 그 이상을 선택하게 되면 Message Box로 '최대 선택가능 좌석은 4개입니다'가 출력됨.



〈그림 16〉인원 및 좌석 선택 화면 4

- 만약에 기존에 예매된 좌석이 있다면 Form_Load 될 때 해당 좌석은 이미 클릭된 좌석으로 표시하고 선택좌석 초기화를 클릭하더라도 초기화가 안 되도록 설계

6) 예매정보 확인



〈그림 17〉예매 내용 확인

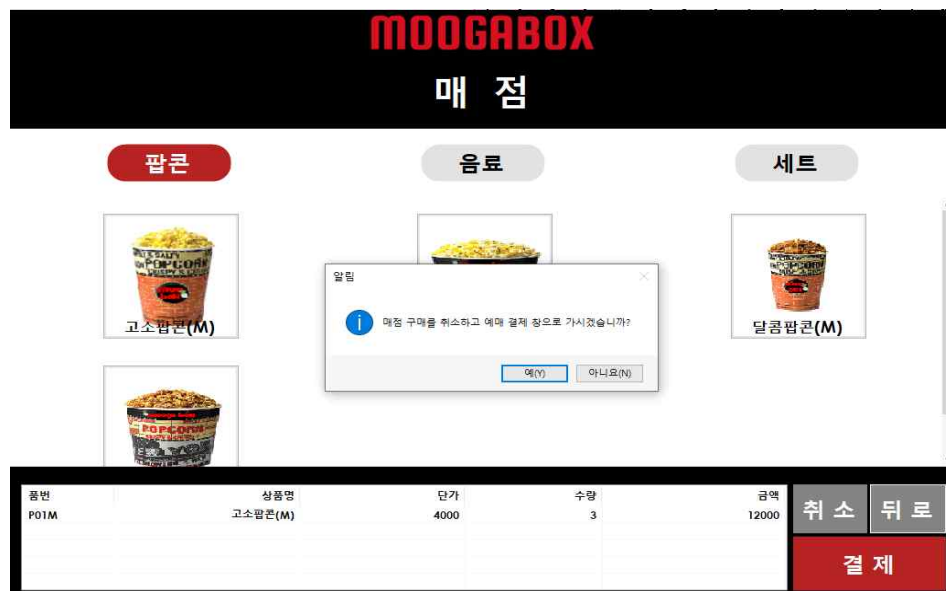
- 지금까지 올바르게 입력 했다면 지금까지 저장된 정보가 화면에 출력
- 올바르게 입력된지 확인하고 확인을 누르면 추가 구매 Message Box가 나오며 여기서 확인을 누르면 매점으로, 취소를 누르면 결제화면으로 갈 수 있도록 설계.

7) 매점



〈그림 18〉매점 1

- 매점화면으로 Tab Control을 이용해 팝콘, 음료, 세트를 클릭하면 넘어가도록 설계
- 해당 그림을 클릭하면 아래 ListView에 해당 항목이 등재되도록 설계



〈그림 19〉매점 2

- 취소 클릭 시 메뉴가 하나씩 삭제되도록 설계
- 뒤로가기 클릭 시 '매점 구매를 취소하고 예매결제 창으로 가겠습니까?'라는 Message Box가 출력되며 확인 시 결제 창으로 이동.

8) 결제



MOOGABOX
결제

예매 결제정보

영화이름	시간	상영관	좌석	수량	합계
닥터 스트레인지	23:05:00	1	A01, A02, A04, A03	4	56000

예매 금액 : **56000**

매점 결제정보

상품명	상품번호	단가	수량	합계
고소팔콘(M)	P01M	4000	10	40000
고소팔콘(L)	P01L	5000	11	55000
달콤팔콘(M)	P02M	5000	10	50000
더블콤보	C02D	12000	4	48000

매점 금액 : **193000**

총 결제금액 : **249000** 결제하기 처음으로

〈그림 20〉 결제

- 지금까지 TmpReservation Table과 TmpBuySnack Table에 저장되어있던 데이터들을 한 화면에 출력.
- 지금까지 티켓 구매했던 예매 금액과 매점 금액을 합산하여 총 결제금액을 표시

9) 예매 완료



MOOGABOX
예매 완료

영화 : 닥터 스트레인지

시간 : 23:05:00

상영관 : 1

좌석 : A01, A02, A04, A03

예매번호 : 20220527170027

처음으로

〈그림 21〉 예매 완료

- 지금까지 Tmp Table에 저장된 Data들 표시하며 실제 Reservation과 BuySnack Table로 옮김.

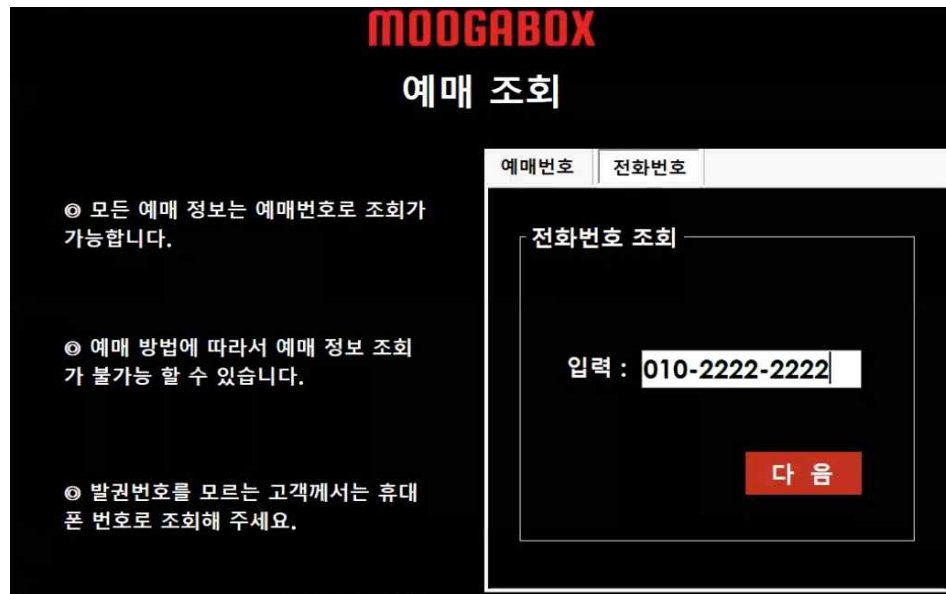
10) 예매 조회

〈그림 22〉예매번호 조회

- 메인메뉴에서 예매 조회버튼을 클릭하면 나오는 화면
- Tab Control을 이용하여 예매번호, 전화번호로 각각 조회가 가능하도록 설계

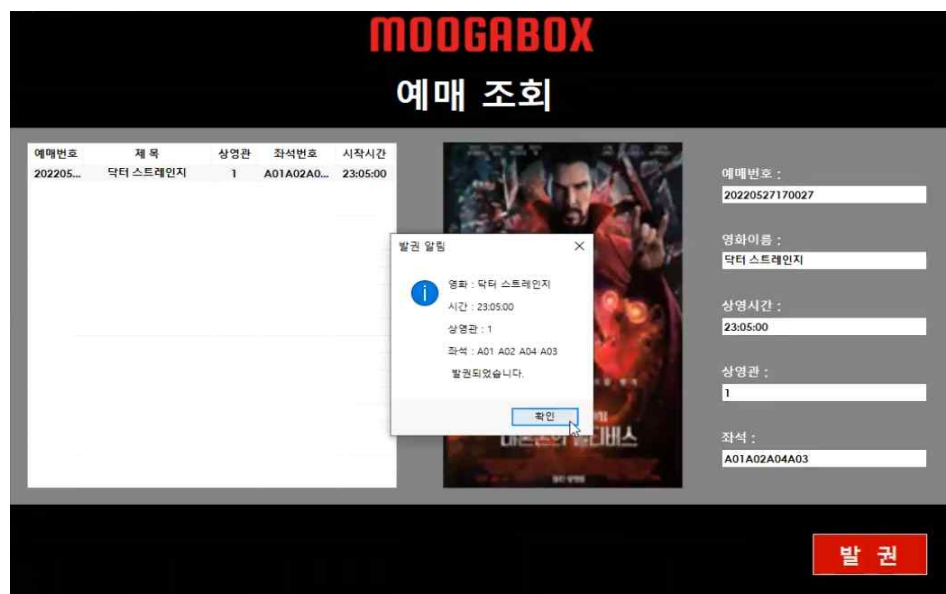
〈그림 23〉예매 확인

- 예매조회를 입력하고 다음을 누르면 기존에 예매했던 정보들이 표시
- 발권 클릭 시 해당 정보가 Message Box에 출력되며 확인을 클릭하면 메인메뉴로 다시 돌아감



〈그림 24〉 전화번호 조회

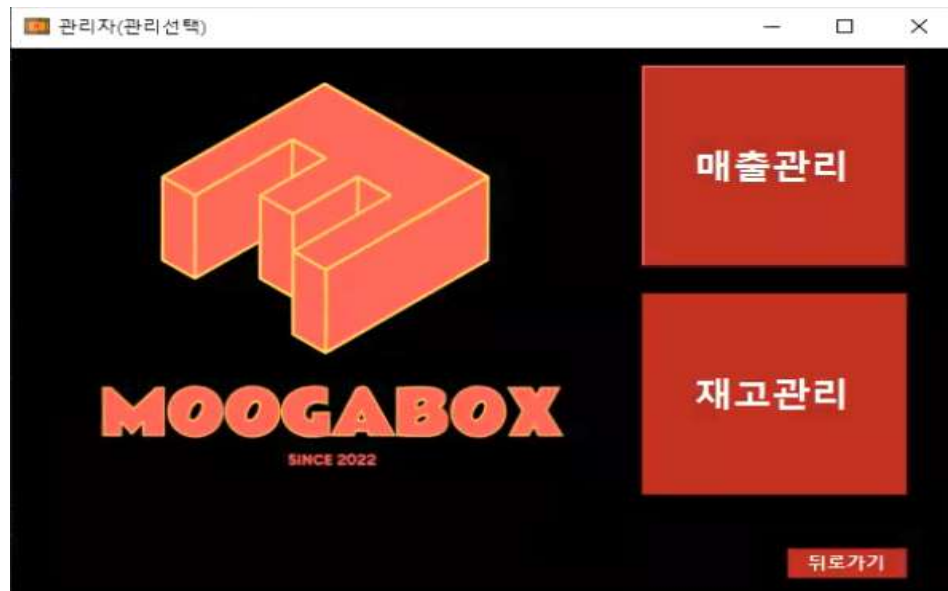
- 전화번호로 조회 시 가입에 이용했던 전화번호를 이용해서 조회가 가능함



〈그림 25〉 전화번호 조회 2

- 예매번호가 아니라 전화번호로 조회를 했기 때문에 예매를 여러번 했을 가능성이 있어 ListView로 해당 전화번호로 예매한 내역을 모두 불러오게 설계
- ListView에서 하나를 선택하면 선택 된 항목의 정보가 우측에 표시되며, 발권 버튼을 클릭 시 Message Box로 해당 정보가 출력
- 아무것도 클릭 없이 발권을 누르면 예외처리를 두어 선택된 데이터가 없다고 출력

11) 매출화면



〈그림 26〉 관리자 메인화면

- 로그인에서 관리자로 로그인하면 나오는 화면
- 각 각 매출관리 화면과 재고관리 화면으로 넘어갈 수 있음



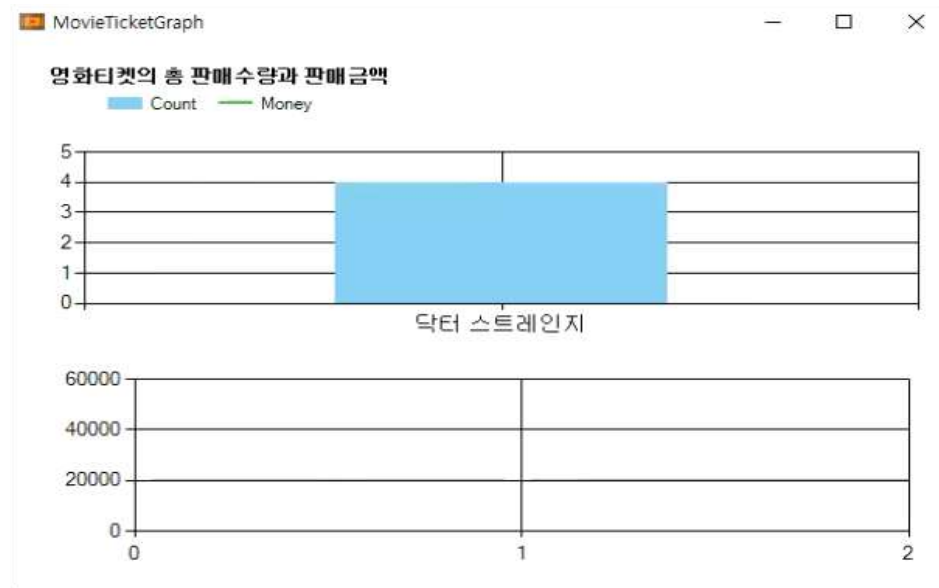
〈그림 27〉 총 판매수량

- 매출관리를 클릭하면 지금까지 매점 판매내역과 티켓 판매내역이 하나의 List로 출력
- 아래의 판매량 그래프를 클릭하면 매점과 티켓 각 각의 판매량을 확인할 수 있음.



〈그림 28〉 스낵의 판매량 그래프

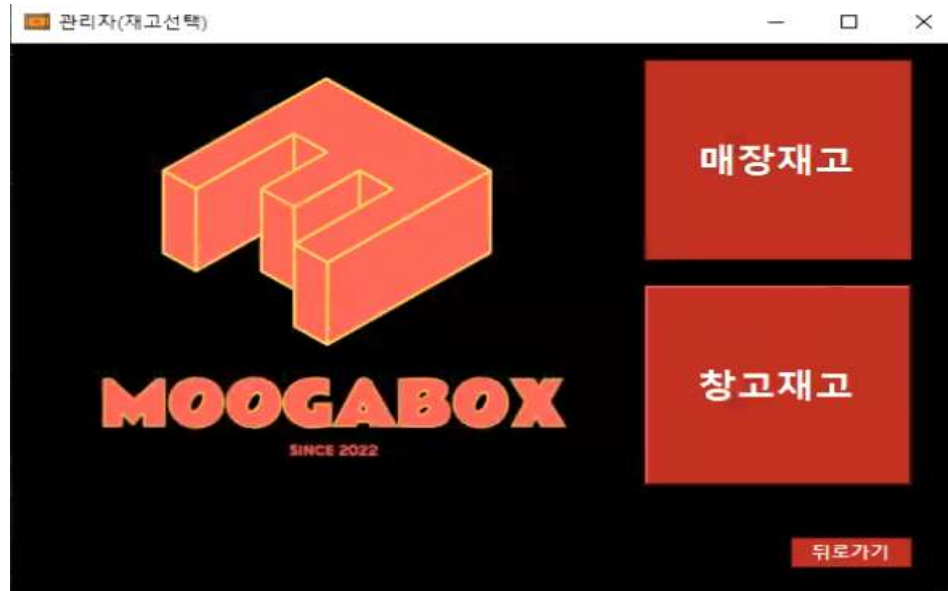
- 스낵의 판매량 그래프를 클릭하면 나오는 화면으로 지금까지 판매했던 매점메뉴들을 항목별 그래프로 확인할 수 있음.



〈그림 29〉 영화티켓 판매량 그래프

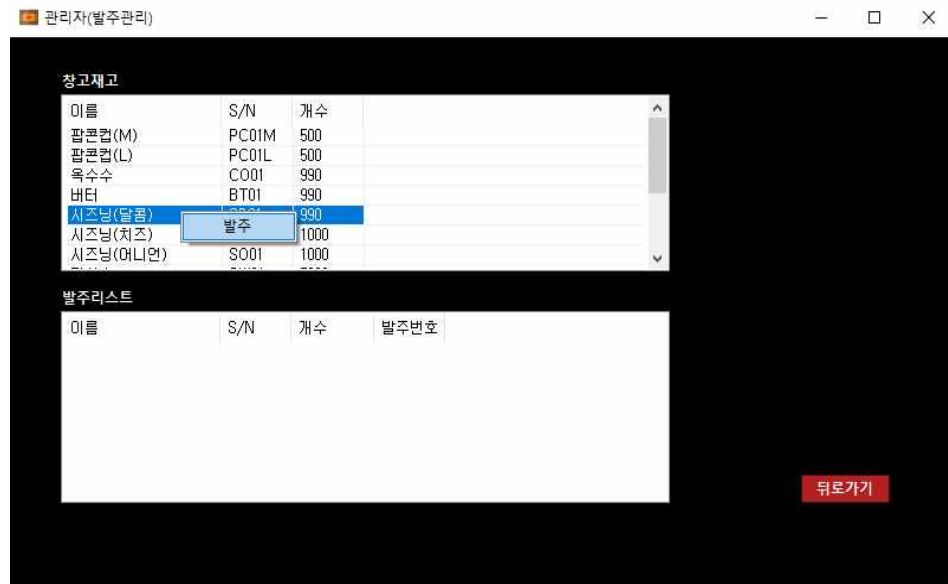
- 영화티켓 판매량 그래프를 클릭하면 나오는 화면. 판매한 티켓에 해당하는 영화의 수익을 확인할 수 있음.

12) 창고재고



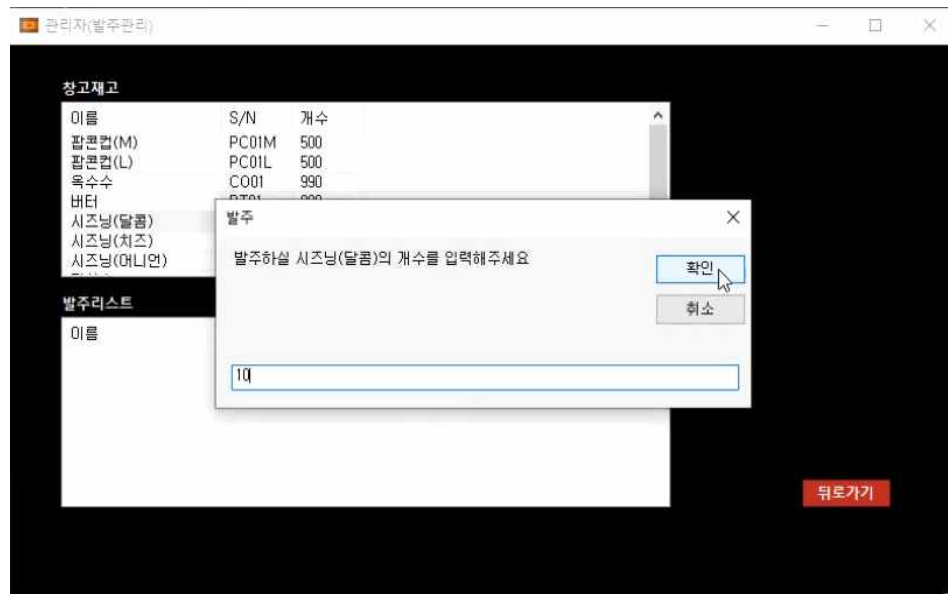
〈그림 30〉 재고 메인화면

- 기존 관리자 메인메뉴에서 재고관리를 클릭하면 나오는 화면
- 매장재고와 창고재고를 선택할 수 있으며 매장재고를 클릭하면 매점으로 생산 및 창고재고 쪽으로 발주를 할 수 있으며 창고재고를 클릭하면 거래처쪽으로 발주 및 입고처리를 할 수있음.



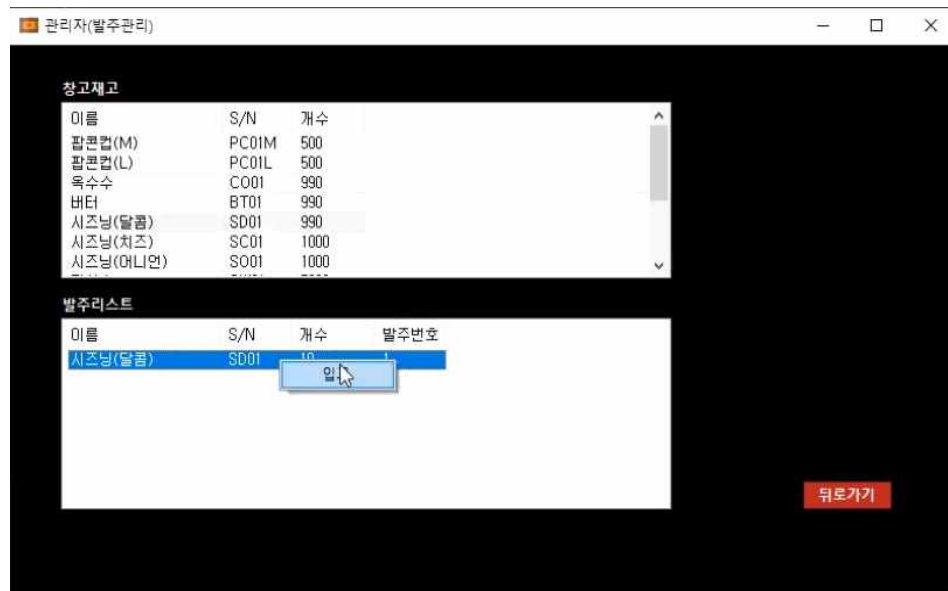
〈그림 31〉 창고재고

- 창고재고를 클릭하면 창고재고와 발주리스트가 출력
- 창고재고 List에서 마우스 오른쪽클릭을 하면 발주가 가능하도록 설계



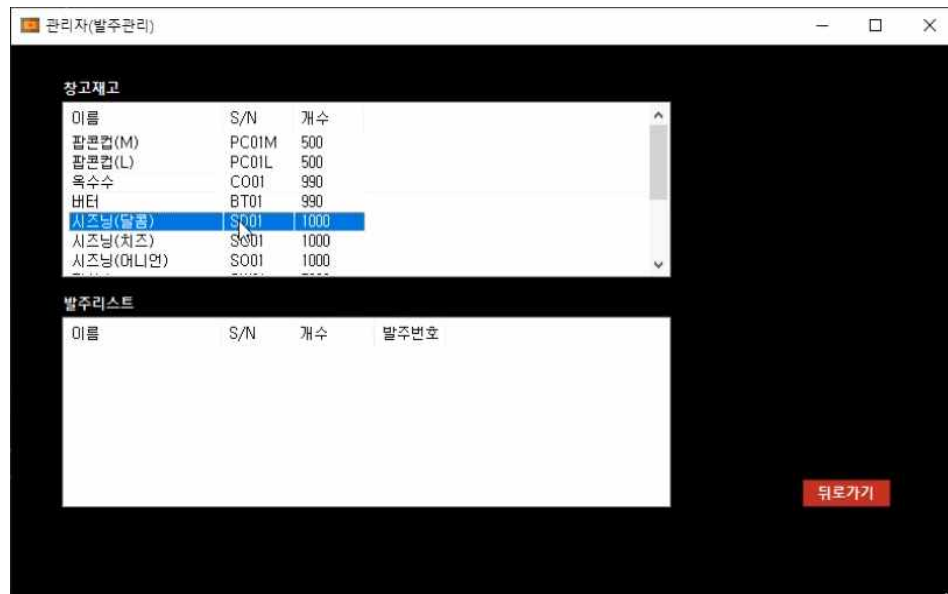
〈그림 32〉 창고재고(발주)

- 발주버튼을 클릭하면 해당화면이 나오며 개수를 입력하여 해당하는 항목을 원하는 만큼 발주할 수 있도록 설계



〈그림 33〉 창고재고(입고)

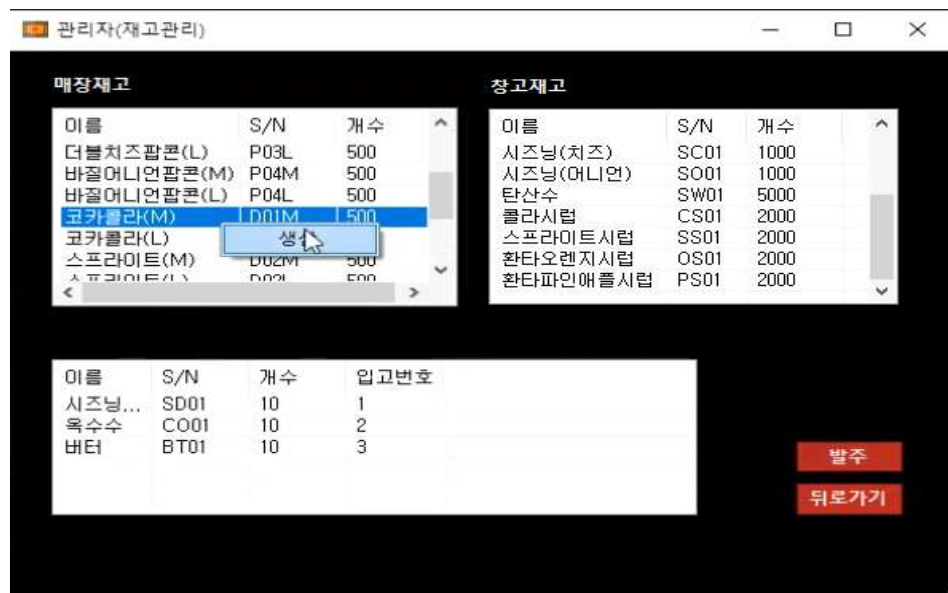
- 발주에서 확인을 클릭하면 해당하는 항목과 개수가 발주리스트에 등록되고 발주리스트에서 마우스 오른쪽 클릭을하게 되면 입고버튼이 활성화 됨.



〈그림 34〉 입고확인

- 입고버튼을 클릭하면 해당하는 창고재고 쪽으로 덧셈되며 화면에선 '시즈닝(달콤)'이 기존에 990개 있던 것이 10개가 입고되어 재고가 1000개 된 것을 확인할 수 있음.

13) 매장재고



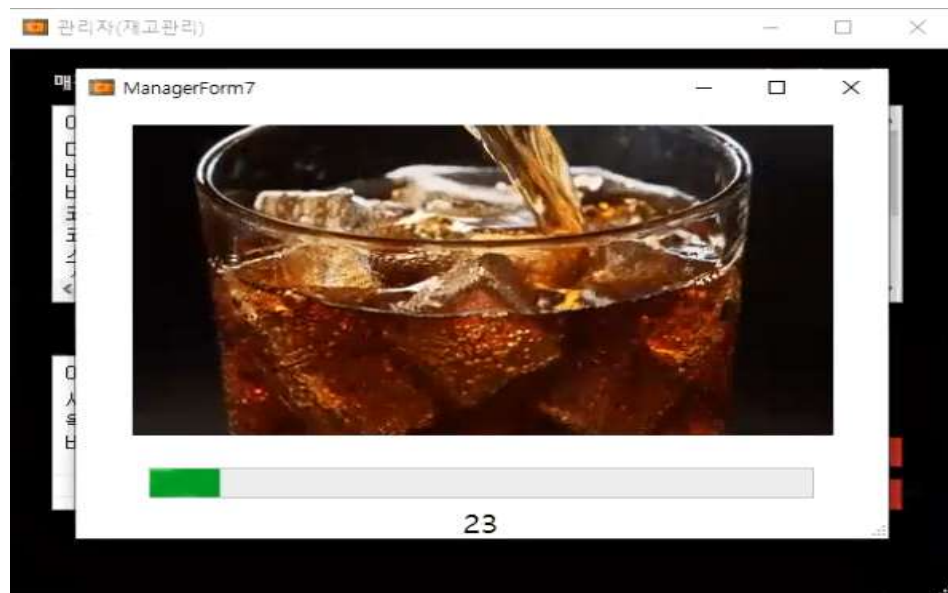
〈그림 35〉 매장재고

- 관리자 메뉴에서 매장재고를 클릭하면 나오는 화면
- 매장재고, 창고재고, 발주리스트를 통해 입고된 내용을 확인할 수 있음.
- 매장재고 List에서 마우스 오른쪽 클릭을 하게 되면 생산 버튼이 활성화.



〈그림 36〉 매장재고(생산)

- 생산버튼을 클릭하면 해당하는 항목을 얼마나 생산할지를 선택할 수 있는 화면이 나오며 개수를 입력하면 해당개수만큼 생산이 되며 매장재고에 덧셈이 되며 창고재고에서 해당하는 항목의 BOM에 따라 재료들이 차감됨.



〈그림 37〉 콜라생산 동영상

- 생산을 클릭하면 해당제품을 생산하는 동영상이 출력되도록 설계

관리자(재고관리)

매장재고			창고재고		
이름	S/N	개수	이름	S/N	개수
바질머니언팝콘(M)	P04M	500	시즈닝(치즈)	SC01	1000
바질머니언팝콘(L)	P04L	500	시즈닝(머니언)	SO01	1000
코카콜라(M)	D01M	530	탄산수	SW01	4970
코카콜라(L)	D01L	500	콜라시럽	CS01	1970
스프라이트(M)	D02M	500	스프라이트시럽	SS01	2000
스프라이트(L)	D02L	500	환타오렌지시럽	OS01	2000
환타오렌지(M)	D03M	500	환타파인애플시럽	PS01	2000

이름	S/N	개수	입고번호
시즈닝...	SD01	10	1
옥수수	CO01	10	2
버터	BT01	10	3

발주
뒤로가기

〈그림 38〉 매장재고 확인

- 코카콜라(M)을 30개 생산하여 매장재고에는 30개가 더해졌고 창고재고에서 탄산수 및 콜라시럽은 30개씩 차감된걸 확인할 수 있음.

4 결과

1) 문제 리스트

(1) 회원가입에서 ID 중복되는 문제 발생

열 이름	데이터 형식	Null 허용
ID	varchar(10)	<input type="checkbox"/>

```
try
{
    int insertRowCnt = Comm.ExecuteNonQuery();
    Conn.Close();

    if (insertRowCnt == 1)
    {
        MessageBox.Show("회원가입이 정상적으로 처리되었습니다.", "회원가입 성공",
            MessageBoxButtons.OK, MessageBoxIcon.Information);

        this.DialogResult = DialogResult.OK;
    }

    else
    {
        MessageBox.Show("회원가입이 실패하였습니다.", "회원가입 실패",
            MessageBoxButtons.OK, MessageBoxIcon.Error);
    }
}
catch
{
    MessageBox.Show("중복되는 ID입니다", "회원가입 실패",
        MessageBoxButtons.OK, MessageBoxIcon.Error);
    TxtClear();
    this.ActiveControl = txtID;
}
```

- SQL의 Member Table의 ID를 PRIMARY KEY로 설정하여 중복을 방지하고 Try~Catch 문을 이용해서 Message Box로 표시해 줌.

(2) 좌석 선택화면에서 취소 클릭 시 기존 예매됐던 좌석까지 취소

```
Button[] Button = { btnA1, btnA2, btnA3, btnA4, btnA5,
    btnB1, btnB2, btnB3, btnB4, btnB5,
    btnC1, btnC2, btnC3, btnC4, btnC5,
    btnD1, btnD2, btnD3, btnD4, btnD5};
var Conn = new SqlConnection(Constr);
Conn.Open();

var Comm1 = new SqlCommand("SELECT Empty FROM Crjo where MvNum = '" + Passvalue + "'", Conn);
var MyRead1 = Comm1.ExecuteReader();

int[] Empty = new int[20];

int i = 0;
while (MyRead1.Read())
{
    Empty[i] = Convert.ToInt32(MyRead1[0]);
    i++;
}

MyRead1.Close();

for (int j = 0; j < 20; j++)
{
    if (Empty[j] == 0)
    {
        Button[j].BackColor = Color.Black;
        Button[j].ForeColor = Color.White;
    }
}

Conn.Close();
```

- 취소를 버튼 누를 때 SQL문을 이용해서 기존에 예매된 좌석을 불러와서 예매가 안 됐던(Empty[j] == 0)부분만 색상을 변경

- (3) Form1에서 예매확인(Form3, Form11)쪽으로 넘어갔다 다시 발권버튼 이후 돌아오면 Form1이 종료되지 않은 상태에서 Form1가 다시 실행되는 문제 발생

```
if (reader.Read())
{
    Reservation_Num = txtNum.Text;

    Form3 frm3 = new Form3();
    frm3.Reservation_Num = Reservation_Num;
    this.Visible = false;
    frm3.ShowDialog();
    Application.Exit();
}
```

- This.Visible = false;를 이용해 Form3(예매조회)이 실행될 때부터 기존 Form1(메인메뉴)는 안보이게 설정하고 Form3이 종료되면 새로운 Form1가 실행되도록 함.

- (4) 동영상 재생 시 Windows Media Player를 사용했으나 GitHub로 데이터 전송 시 참조파일 차이로 인해 재생이 불가

```
public ManagerForm7()
{
    InitializeComponent();

    timer2.Interval = 40;
    timer2.Tick += timer1_Tick;
    progressBar1.Minimum = 0;
    progressBar1.Maximum = 100;
    progressBar1.Value = 0;
    timer2.Start();

    ///////////////////////////////////
    var CurrentDirectory = Directory.GetCurrentDirectory();
    string newPath = Path.GetFullPath(Path.Combine(CurrentDirectory, @"..\\..\\Media\\Drink.gif"));
    pictureBox1.Load(newPath);
    pictureBox1.SizeMode = PictureBoxSizeMode.CenterImage;
}
```

- PictureBox에 gif형식으로 변환하여 재생하도록 변경

- (5) Button이 여러개 있는 경우가 많아 버튼을 하나씩 IF문으로 설정하다보니 코드가 길고 가독성이 안 좋아짐.

```
Button[] button = {btnTime07, btnTime08, btnTime09, btnTime10, btnTime11,
    btnTime12, btnTime01, btnTime02, btnTime03, btnTime04,
    btnTime05, btnTime06, btnTime13, btnTime15, btnTime16,
    btnTime17, btnTime18, btnTime19, btnTime20};

for (int j = 0; j < 19; j++)
{
    if (!Btn_enabled(time[j]))
    {
        button[j].Enabled = false;
    }
}
```

- Button 배열, For 반복문을 이용해서 간단하게 표현

2) 아쉬운 점

- (1) 기능 미구현
 - ① 결제 취소, 환불, 발권 시 영수증 출력 같은 기능 구현 실패
 - ② 공부해온 ERP를 구현하기 위해 관리자 기능을 넣었지만 다소 미흡하게 구현되어 추가적인 수정 및 보완이 필요
- (2) Form을 하나의 컴퓨터에서 제작하지 않고 여러 컴퓨터에서 분산하여 제작하다보니 Form이름이 겹쳐 통합 시 문제가 생김.
- (3) 이번 프로젝트에서 GitHub를 사용했지만 아직 미숙한 인원이 많았음.
- (4) DB에서 Table 간의 참조, PK, FK 사용이 부족하여 DB의 완성도가 떨어짐.
- (5) Log 생성, 예외처리등의 미구현

3) 배운 점

- (1) 지금까지 배워왔던 C#, Windows Form, SQL을 한 번에 연동하여 프로그램을 작성하다보니 각 프로그램 간 연동방식, 데이터 흐름을 이해할 수 있었음.
- (2) 저번 C, C++ 미니프로젝트를 진행하면서 GitHub를 미사용하고 메신저로 연락하여 업무 연동성이 낮았는데, 이번에는 GitHub를 활용하면서 업무 연동성을 높일 수 있다는 것을 배웠음.
- (3) 팀 프로젝트 시 회의를 자주하여 상호간 업무보고를 하여 전반적인 프로젝트 조율, 임무 재분배가 필요하다는 것을 배움.
- (4) Form 통합 시 이름 같은 부분에서 문제가 생겼기 때문에, 초반 회의에서 사소한 이름 같은 것도 미리 정하고 프로젝트를 진행 하는 것이 좋음.
- (5) SQL DB를 Project 시작단계에서 미리 설계를 하고 시작을 했지만 DB 설계에 미숙하여 프로젝트 진행도중 수정소요가 많았기 때문에 앞으로는 기존 설계단계에서 DB 설계 비중을 늘려 시간을 가지고 자세하게 설계해야함을 배움.

4) 느낀 점

- (1) 주석의 필요성을 다시 한 번 느낌.
- (2) 이전 프로젝트에서 GitHub의 필요성을 느껴 이번에 사용했지만 미숙한 부분도 많고, Commit 할 때 자세한 추가설명을 자세히 적거나 회의 할 때마다 GitHub의 Issue Page에 꾸준히 기록하여 팀원끼리 공유되도록 하는 것이 필요하다고 느낌.
- (3) WinForm을 사용하여 작업 시 기능구현도 중요하지만 사용자 눈에 보이는 디자인도 중요한 것을 느낌.
- (4) SQL서버 연동과 TCP 소켓통신등의 원격 연결에 대해 더 공부하여 더욱 효율적인 방법으로 서버와 클라이언트를 연결하는 방법을 알아야 함을 느낌.

5) 향후 공부 방향

- (1) 전체 인원들의 GitHub에 관한 추가적인 공부
- (2) 완성도 높은 DB 설계와 PK(개인키), FK(외래키) 활용을 위해 MY-SQL 공부
- (3) 생산자 측면에서 매출, 재고 관리 외에도 생산 프로세스를 이해하기 위해 ERP,

- MES을 공부하고 I-CUBE 핵심 ERP를 연계해 생산관리 시스템 구축 고려
- (4) 단순한 SQL 서버에서 제공하는 통신 외에도 차후에 배울 TCP / IP를 이용해서 효율적인 Server – Client 간의 통신체계 구축방법 공부
 - (5) 프로젝트 기간이 짧아서 구현하지 못했던 결제 취소, 환불, 발권 시 영수증 출력 기능 구현
 - (6) 차후에 있을 C# 보충 수업을 통해 학습 후 추가기능 추가, 보완
 - (7) 보충 수업 외에 개인적으로도 WinForm 컨트롤 속성, 기능, 사용자 정의 폼을 공부하여 프로그램 퀄리티를 향상시키기 위해 노력

5 동영상

시연 영상 : <https://youtu.be/UQH7eHv1bzl>