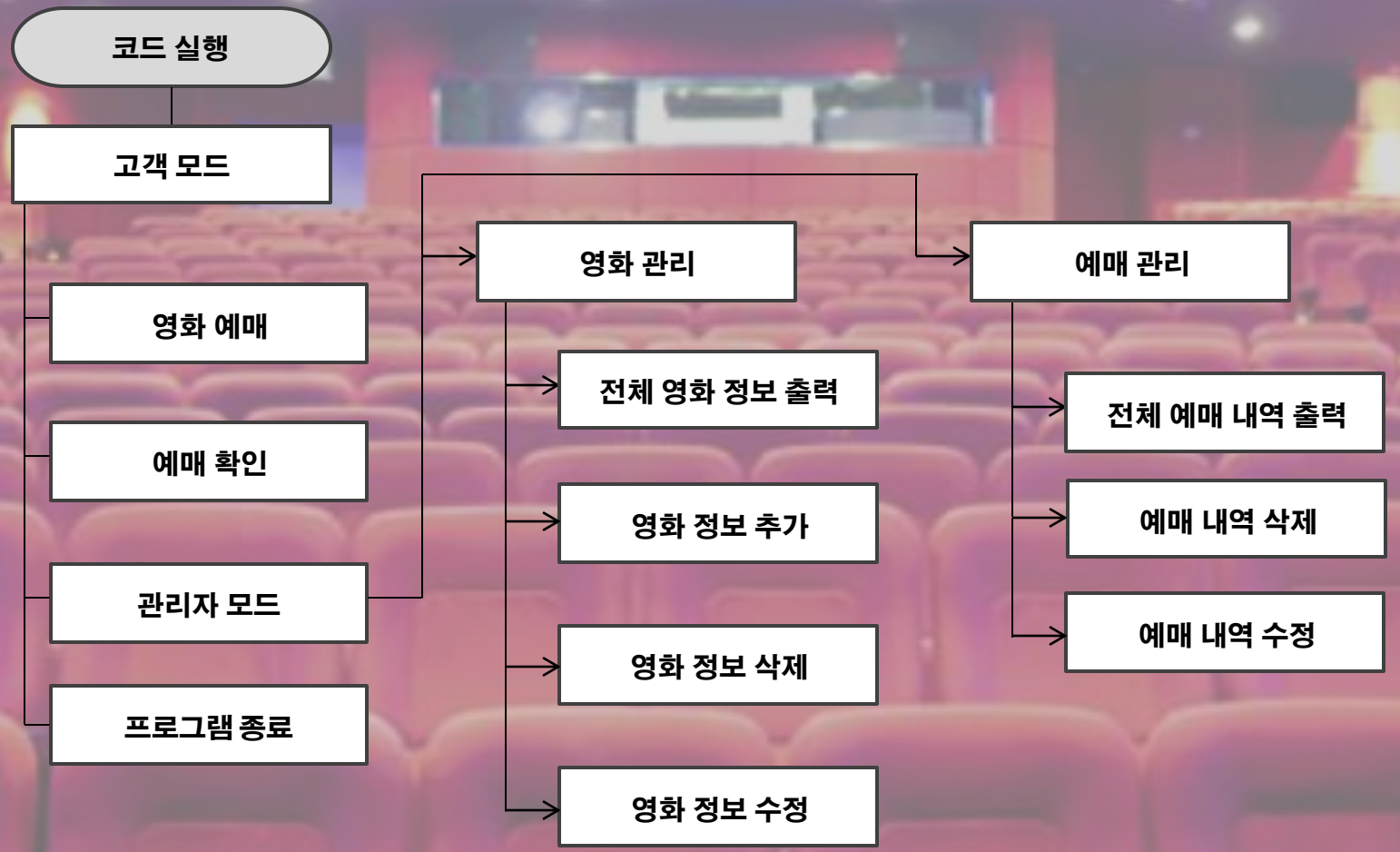
영화 좌석 예매 프로그램
화면 설계서

전장훈

2023.03.27

User flow


코드 실행
고객 모드
영화 예매
예매 확인
관리자 모드
프로그램 종료
영화 관리
전체 영화 정보 출력
영화 정보 추가
영화 정보 삭제
영화 정보 수정
예매 관리
전체 예매 내역 출력
예매 내역 삭제
예매 내역 수정

SQL Table

<Movie Table>

Field	Type	Null	Key	Default	Extra
mv_num	int	NO	PRI	NULL	auto_increment
mv_name	varchar(50)	NO		NULL	
mv_genre	varchar(30)	NO		NULL	
mv_time	varchar(20)	NO		NULL	
mv_theater	varchar(30)	NO		NULL	

<Reservation Table>

Field	Type	Null	Key	Default	Extra
res_num	int	NO	PRI	NULL	auto_increment
res_hp	int	YES		NULL	
res_mv_name	varchar(50)	YES		NULL	
res_mv_time	varchar(10)	YES		NULL	
res_mv_theater	varchar(30)	YES		NULL	
res_seat	varchar(10)	YES		NULL	
res_time	timestamp	YES		CURRENT_TIMESTAMP	DEFAULT_GENERATED

DTO 클래스 – Movie, Reservation

```python
class Movie:
    def __init__(self, name, genre, time, theater):
        self.mv_name = name
        self.mv_genre = genre
        self.mv_time = time
        self.mv_theater = theater

    # setter
    def setMv_name(self, name):
        self.mv_name = name

    def setMv_genre(self, genre):
        self.mv_genre = genre

    def setMv_time(self, time):
        self.time = time

    def setMv_theater(self, theater):
        self.theater = theater

    # getter
    def getMv_name(self):
        return self.mv_name

    def getMv_genre(self):
        return self.mv_genre

    def getMv_time(self):
        return self.mv_time

    def getMv_theater(self):
        return self.mv_theater
```

```python
class Reservation:
    def __init__(self, hp, name, time, theater, seat):
        self.res_hp = hp
        self.res_mv_name = name
        self.res_mv_time = time
        self.res_mv_theater = theater
        self.res_seat = seat

    # setter
    def setRes_hp(self, hp):
        self.res_hp = hp

    def setRes_mv_name(self, name):
        self.res_mv_name = name

    def setRes_mv_time(self, time):
        self.res_mv_time = time

    def setRes_mv_theater(self, theater):
        self.res_mv_theater = theater

    def setRes_seat(self, seat):
        self.res_seat = seat
```

```python
    # getter
    def getRes_hp(self):
        return self.res_hp

    def getRes_mv_name(self):
        return self.res_mv_name

    def getRes_mv_time(self):
        return self.res_mv_time

    def getRes_mv_theater(self):
        return self.res_mv_theater

    def getRes_seat(self):
        return self.res_seat
```

DAO 클래스1 - movieList

```python
import MySQLdb
class movieList:
    def __init__(self):
        self.db = None

    def connect(self):
        self.db = MySQLdb.connect('localhost', 'root', '1234', 'hw')

    def disconnect(self):
        self.db.close()

    def printAll(self):
        self.connect()
        cur = self.db.cursor(MySQLdb.cursors.DictCursor)
        sql = 'select * from movie'
        cur.execute(sql)
        while 1:
            mv = cur.fetchone()
            if mv:
                print(f"영화 번호: {mv['mv_num']}, 영화명: {mv['mv_name']}, 장르: {mv['mv_genre']}, 시간: {
            else:
                break
        self.disconnect()

    def insert(self, mv):
        self.connect()
        cur = self.db.cursor()
        sql = "insert into movie (mv_name, mv_genre, mv_time, mv_theater) values (%s, %s, %s, %s)"
        data = (mv.getMv_name(), mv.getMv_genre(), mv.getMv_time(), mv.getMv_theater())
        cur.execute(sql, data)
        self.db.commit()
        self.disconnect()
```

```python
    def search(self, num):
        self.connect()
        cur = self.db.cursor()
        sql = 'select * from movie where mv_num=%s'
        cur.execute(sql, (num,))
        row = cur.fetchone()
        self.disconnect()
        return row

    def delete(self, row):
        self.connect()
        cur = self.db.cursor()
        sql = 'delete from movie where mv_num=%s'
        result = cur.execute(sql, (row,))
        self.db.commit()
        if result > 0:
            print('삭제되었습니다')
        else:
            print('해당 영화 정보가 없습니다')
        self.disconnect()

    def update(self, num, col, new_data):
        self.connect()
        cur = self.db.cursor()
        sql = 'update movie set ' + col + '=%s where mv_num=%s'
        data = (new_data, num)
        result = cur.execute(sql, data)
        self.db.commit()
        if result > 0:
            print('수정되었습니다')
        else:
            print('해당 영화 정보가 없습니다')
        self.disconnect()
```

DAO 클래스2 - reserveList

```
class reserveList:
    def __init__(self):
        self.db = None

    def connect(self):
        self.db = MySQLdb.connect('localhost', 'root', '1234', 'hw')

    def disconnect(self):
        self.db.close()

    def printAll(self):
        self.connect()
        cur = self.db.cursor(MySQLdb.cursors.DictCursor)
        sql = 'select * from reservation'
        cur.execute(sql)
        while 1:
            rs = cur.fetchone()
            if rs:
                print(f"예매 번호: {rs['res_num']}, 영화명: {rs['res_mv_name']}, 
            else:
                break
        self.disconnect()

    def insert(self, hp, seat, mv):
        self.connect()
        cur = self.db.cursor(MySQLdb.cursors.DictCursor)
        sql = "insert into reservation (res_hp, res_mv_name, res_mv_time, res_mv_t
        data = (hp, mv[1], mv[3], mv[4], seat)
        result = cur.execute(sql, data)
        self.db.commit()
```

```
def search(self, row):
    self.connect()
    cur = self.db.cursor(MySQLdb.cursors.DictCursor)
    sql = 'select * from reservation where res_hp=%s'
    cur.execute(sql, (row,))
    while 1:
        rs = cur.fetchone()
        if rs:
            print(f"예매 번호: {rs['res_num']}, 영화명: {rs['res_mv_name']}, 시간: {rs['res_mv_time']},
        else:
            break
    self.disconnect()
    return row

def delete(self, row):
    self.connect()
    cur = self.db.cursor()
    sql = 'delete from reservation where res_num=%s'
    result = cur.execute(sql, (row,))
    self.db.commit()
    if result > 0:
        print('삭제되었습니다')
    else:
        print('해당 예매 내역이 없습니다')
    self.disconnect()

def update(self, num, new_data):
    self.connect()
    cur = self.db.cursor()
    sql = 'update reservation set res_seat=%s where res_num=%s'
    data = (new_data, num)
    result = cur.execute(sql, data)
    self.db.commit()
    if result > 0:
        print('수정되었습니다')
    else:
        print('해당 영화 정보가 없습니다')
    self.disconnect()
```

Service 클래스1 : 고객 모드

```
class csMode():
    def __init__(self):
        self.ml = movieList() # 영화 리스트
        self.rl = reserveList() # 예매 리스트

    # 영화 예매 함수
    def selectMovie(self):
        hp = input('예매 확인 번호(전화번호 뒷 4자리)를 입력하세요: ')

        # 4자리인지 확인
        if len(hp) != 4:
            print('4자리가 아닙니다. 다시 입력하세요.')
            self.selectMovie()

        print('상영 영화 리스트')
        self.ml.printAll()
        num = input('예매할 영화 번호를 입력하세요: ')
        mv = list(self.ml.search(num))
        people = int(input('예매 인원을 입력하세요(최대 3명, 숫자만 입력): '))
        for i in range(0,people):
            print(str(people) + '명 중 ' + str(i+1) + '번째 예매입니다.₩n좌석은 [A1 ~ M10]을 선택할 수 있습니다.')
            seat = input('예매할 좌석을 입력하세요 (E5): ')
            seat = seat.upper()
            self.rl.insert(hp, seat, mv)
        print('예매가 완료되었습니다.')


    # 예매 내역 확인
    def printReserve(self):
        num = input('예매 확인 번호(전화번호 뒷 4자리)를 입력하세요: ')
        if len(num) != 4:
            print('4자리가 아닙니다. 다시 입력하세요.')
            self.printReserve()
        self.rl.search(num)
```

Service 클래스2 : 매니저 모드

```
class mnMode:
    def __init__(self):
        self.ml = movieList() # 영화 리스트
        self.rl = reserveList() # 예매 리스트

    # 전체 영화 정보 출력
    def printAllMovie(self):
        datas = self.ml.printAll()

    # 영화 정보 추가
    def insertMovie(self):
        name = input('영화명을 입력하세요: ')
        genre = input('영화장르를 입력하세요: ')
        time = input('상영시간을 입력하세요: ')
        theater = input('상영관를 입력하세요: ')
        movie = Movie(name, genre, time, theater)
        self.ml.insert(movie)

    # 영화 정보 삭제
    def delMovie(self, num):
        row = self.ml.search(num)
        if row == None:
            print('삭제할 영화가 없습니다.')
        else:
            self.ml.delete(num)
            print(num + '번 영화 정보가 삭제되었습니다.')

    # 영화 정보 수정
    def updateMovie(self, num):
        row = self.ml.search(num)
        if row == None:
            print('수정할 영화가 없습니다.')
        else:
            col = {1:"mv_name", 2:"mv_genre", 3:"mv_time", 4:"mv_theater"}
            select = int(input('수정할 정보를 입력하세요(1.영화명 2.장르 3.상영시간 4.상영관): '))
            new_data = input('새로운 정보를 입력하세요: ')
            self.ml.update(num, col[select], new_data)
```

```
    # 전체 예매 내역 출력
    def printAllReserve(self):
        datas = self.rl.printAll()

    # 예매 내역 삭제
    def delReserve(self, num):
        row = self.rl.search(num)
        if row == None:
            print('삭제할 예매 내역이 없습니다.')
        else:
            self.rl.delete(row)
            print('예매 내역이 삭제되었습니다.')

    # 예매 내역 수정
    def updateReserve(self, num):
        row = self.rl.search(num)
        if row == None:
            print('수정할 예매 내역이 없습니다.')
        else:
            new_data = input('변경할 좌석을 입력하세요: ')
            new_data = new_data.upper()
            self.rl.update(num, new_data)
```

Main 클래스 [1]

```
class Main:
    def __init__(self):
        self.mn = mnMode()  # 관리자모드
        self.cs = csMode()  # 고객모드

    def run(self):
        try:
            print('********** 메뉴 **********')
            print('영화 예매 프로그램입니다. 영화 예매를 할 수 있습니다.')
            # 고객 모드
            while 1:
                menu = int(input('메뉴를 선택하세요₩n1. 영화 예매₩n2. 예매 확인₩n9. 관리자 모드₩n0. 프로그램 종료₩n'))
                print()
                # 고객 모드 > 영화 예매
                if menu == 1:
                    self.cs.selectMovie()

                # 고객 모드 > 예매 확인
                elif menu == 2:
                    self.cs.printReserve()

                # 관리자 모드
                elif menu == 9:
                    while 1:
                        # 관리자 로그인
                        id = input('관리자 아이디를 입력하세요: ')
                        pw = input('관리자 비밀번호를 입력하세요: ')
                        # 로그인 성공
                        if id == 'admin' and pw == 'admin':
                            break
```

Main 클래스 [2]

```
while 1:
    mn_menu = int(input('관리자 모드입니다.₩n1. 영화 관리₩n2. 예매 관리₩n0. 뒤로 가기₩n'))

    # 관리자 모드 > 영화 관리
    if mn_menu == 1:
        while 1:
            mn_mv_menu = int(input('메뉴를 선택하세요.₩n1. 전체 영화 정보 출력₩n2. 영화 정보 추가₩n3. 영화 정보 삭제₩n4. 영화 정보 수정
            print()
            # 관리자 모드 > 영화 관리 > 전체 영화 정보 출력
            if mn_mv_menu == 1:
                self.mn.printAllMovie()

            # 관리자 모드 > 영화 관리 > 영화 정보 추가
            elif mn_mv_menu == 2:
                self.mn.insertMovie()

            # 관리자 모드 > 영화 관리 > 영화 정보 삭제
            elif mn_mv_menu == 3:
                num = input('삭제할 영화 번호를 입력하세요: ')
                self.mn.delMovie(num)

            # 관리자 모드 > 영화 관리 > 영화 정보 수정
            elif mn_mv_menu == 4:
                num = input('수정할 영화 번호를 입력하세요: ')
                self.mn.updateMovie(num)

            # 관리자 모드
            elif mn_mv_menu == 0:
                break
```

Main 클래스 [3]

```
            # 관리자 모드 > 예매 관리
            elif mn_menu == 2:
                while 1:
                    mn_rs_menu = int(input('메뉴를 선택하세요.₩n1. 전체 예매 내역 출력₩n2. 예매 내역 삭제₩n3. 예매 내역 수정₩n0. 뒤로 가기₩n'))
                    print()
                    # 관리자 모드 > 예매 관리 > 전체 예매 내역 출력
                    if mn_rs_menu == 1:
                        self.mn.printAllReserve()

                    # 관리자 모드 > 예매 관리 > 예매 내역 삭제
                    elif mn_rs_menu == 2:
                        num = input('삭제할 예매 번호를 입력하세요: ')
                        self.mn.delReserve(num)

                    # 관리자 모드 > 예매 관리 > 예매 내역 수정
                    elif mn_rs_menu == 3:
                        num = input('수정할 예매 번호를 입력하세요: ')
                        self.mn.updateReserve(num)

                    # 관리자 모드
                    elif mn_rs_menu == 0:
                        break

            # 고객 모드
            elif mn_menu == 0:
                print('관리자 모드 해제')
                break
            print()

        elif menu == 0: # 프로그램 종료
            print('프로그램 종료')
            break
        print()
except:
    print('다시 입력하세요')
```