

The background of the slide is a stylized, painterly illustration of the Grand Budapest Hotel. The hotel is a large, multi-story building with a pinkish-red facade and white window frames. It features a central entrance with a grand staircase and a sign that reads "GRAND BUDAPEST HOTEL". The building is set against a backdrop of snow-covered trees and a mountain. In the foreground, there is a stone bridge with arches and a small building with a blue roof.

# Hotel Reservation System

## - JAVA Console Project -

3조

팀원 : 장용범, 신우현, 김태호

2021.10.27



# 목차

---

1. 개요 (Hotel Reservation System )

2. 코드설계

3. Console 화면구현

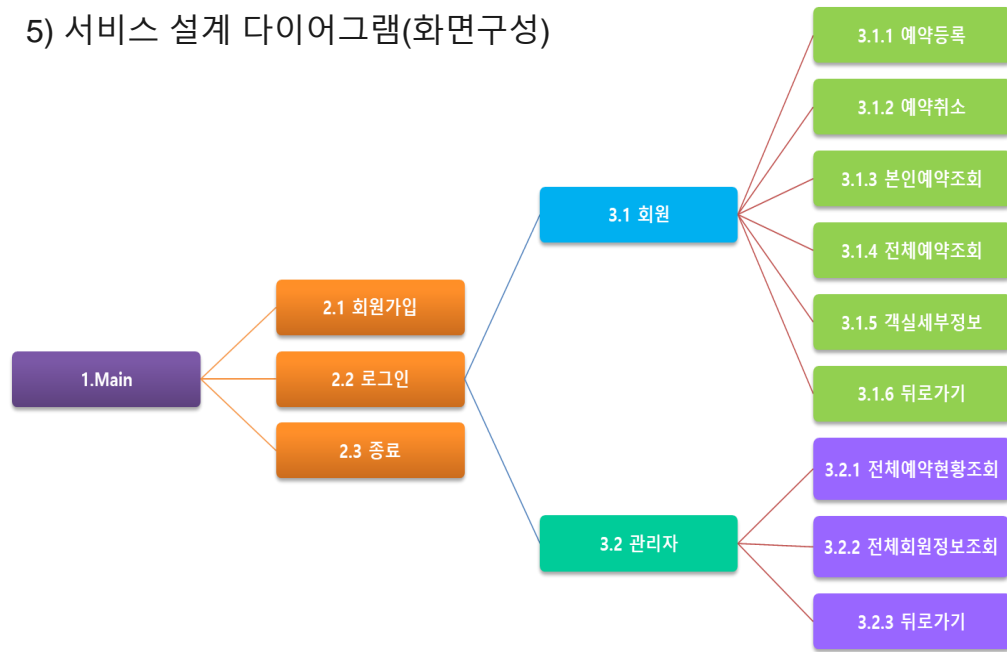
4. 평가 ( 역할 및 느낀점 / 개선방향)



# 1. 개요 (Hotel Reservation System)

- 1) 주제 : 호텔 예약 시스템
- 2) 목적 : 호텔 이용을 편리하게 하기 위해 예약 서비스를 제공하여 고객들의 만족도를 높임
- 3) 고객층 : 호텔 이용 예정중인 고객과, 호텔예약프로그램을 쓰고자 하는 호텔회사
- 4) 환경 : JDK11 / Eclipse / Window10 / Git
- 5) 프로젝트 기간 : 2021.10.19 ~ 2021.10.27

## 5) 서비스 설계 다이어그램(화면구성)

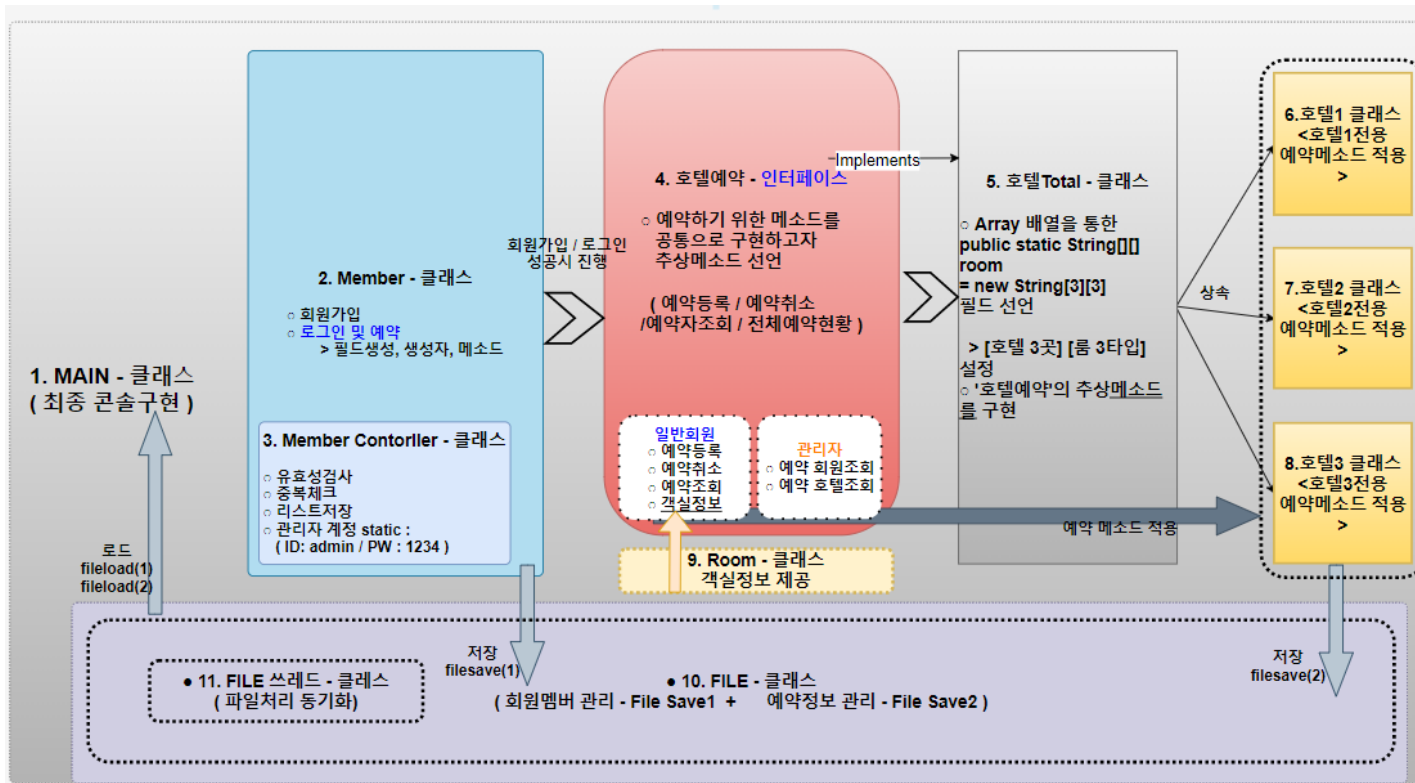
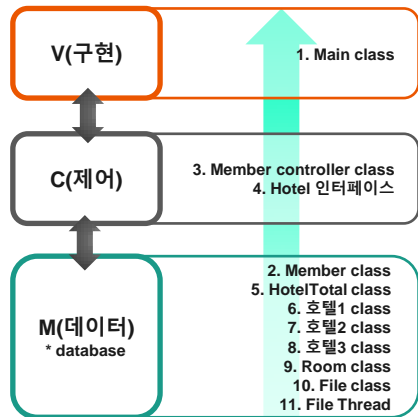




## 2. 코드설계

### 1) 전체구조 설계

- 총 : 1 패키지 11 자바파일  
(10 클래스 / 1 인터페이스)
- > Main : 동작구현 (Menu)
- > Interface : 추상메소드 제공
- > Hotel : 배열 > Hotel1 / Hotel2 / Hotel3 상속





## 2. 코드설계

### 2-1) Main Class - 메뉴

- 1. 입력객체 생성
- 2. 파일Thread 및 파일Road
- 3. 메뉴선택 > While 무한반복

while (true) ( ch == 1/2/3 )

적용 및 메인메뉴 출력

[ 1.회원가입페이지 / 2.로그인 / 3.종료]

> 회원 가입 시 유효성 검사를 통하여 가입제한

: MemberController 내

public static boolean signup() { } 메소드 적용

- 4. 조건 예외값 입력시 입력오류 처리 : try{} catch{}

```

1 package 김태호;
2
3 import java.util.ArrayList;
4
5
6 public class TestMain {
7
8
9     // 입력객체 생성
10    public static Scanner scanner = new Scanner(System.in);
11
12    public static void main(String[] args) {
13
14        // 스레드 시작
15        FileThread fileThread = new FileThread();
16        fileThread.start();
17
18        File fileload();
19
20        // 1. 인터페이스 생성
21        호텔예약 호텔예약;
22
23        while (true) {
24            try {
25                System.out.println(" ===== 호텔예약시스템 ===== ");
26                System.out.println(" ===== main menu ===== ");
27                System.out.println("\t1.회원가입\t\t2.로그인\t\t3.종료");
28                System.out.println(" ===== ");
29                System.out.print(">>> 선택 : ");
30
31                int ch = scanner.nextInt();
32                if (ch == 1) { // 회원가입
33
34                    System.out.println(" ===== 회원가입 페이지 ===== ");
35                    MemberController.signup();
36                }
37            } catch (Exception e) {
38                System.out.println(" ===== 회원가입 메소드 ===== ");
39            }
40        }
41    }
42 }
  
```

□ Console 구현  
[회원가입]

```

===== 호텔예약시스템 =====
===== main menu =====
1.회원가입 2.로그인 3.종료
>>> 선택 : 1
  
```

```

===== 호텔예약시스템 =====
===== 회원가입 페이지 =====
Id[4글자이상] : ezen
Password[4글자만] : ezen
name[2글자이상] : ezen
HP Number : 010-1234-1234
[유효] : 회원가입 성공
===== 호텔예약시스템 =====
===== main menu =====
1.회원가입 2.로그인 3.종료
>>> 선택 :
  
```

## 2. 코드설계

### 2-1) Main Class - 회원

#### - 4. 메뉴선택을 위한 While 무한반복

적용 및 메인메뉴 출력

[ 1.회원가입페이지 / 2.로그인 / 3.종료]

#### - 5. 로그인 [ch1 = 회원 / ch2 = 관리자]

> 회원 로그인 성공 시 (유효성검사)

예약을 위한 입력값 별 구성

While (true) { if (hotelch == 1~5) }

1. 예약등록 2. 예약취소 3. 본인예약조회
4. 전체예약조회 5. 객실세부정보 6. 뒤로가기

```

else if (ch == 2) {
    System.out.println("----- 로그인 페이지 -----");
    로그인();
    System.out.println("\t\t1.회원 \t\t2.관리자");
    System.out.print("선택 : ");
    int ch1 = scanner.nextInt();

    if (ch1 == 1) { // 회원
        회원();
        String h_id = scanner.next();
        System.out.print(" Password : ");
        String h_pw = scanner.next();

        boolean result = MemberController.login(h_id, h_pw);
        if (result) { // 로그인 성공시
            System.err.println(" [알림] : 로그인 성공.");

            while (true) {
                System.out.println("1.예약등록 2.예약취소 3.본인예약조회\n4.전체예약조회 5.객실세부정보 6.뒤로가기");
                System.out.print(">>> 선택 : ");
                int ch2 = scanner.nextInt();

                if (ch2 == 1) { // 예약등록
                    System.out.println(">>>호텔지점 선택<<");
                    System.out.println("1.신라호텔 2.라마다호텔 3.그랜드호텔");
                    int hotelch = TestMain.scanner.nextInt();
                    System.out.println(">>>방 타입 선택<<");
                    System.out.println("1.스위트 2.디럭스 3.스탠다드");
                    int roomch = TestMain.scanner.nextInt();

                    if (hotelch == 1) {
                        호텔예약 = new 호텔1();
                        for (int i = 1; i <= 3; i++) {
                            if (roomch == 1) {
                                호텔예약.예약등록(hotelch, roomch, h_id);
                            }
                        }
                    } else if (hotelch == 2) {
                        호텔예약 = new 호텔2();
                        for (int i = 1; i <= 3; i++) {
                            if (roomch == 1) {
                                호텔예약.예약등록(hotelch, roomch, h_id);
                            }
                        }
                    } else if (hotelch == 3) {
                        호텔예약 = new 호텔3();
                        for (int i = 1; i <= 3; i++) {
                            if (roomch == 1) {
                                호텔예약.예약등록(hotelch, roomch, h_id);
                            }
                        }
                    } else {
                        System.err.println(" 잘못된 입력번호입니다. 번호를 다시 선택해주세요. ");
                    }
                }
            }
        }
    }
}

else if (ch2 == 2) { // 예약취소
    System.out.println(">>>호텔지점 선택<<");
    System.out.println("1.신라호텔 2.라마다호텔 3.그랜드호텔");
    int hotelch = TestMain.scanner.nextInt();
    System.out.println(">>>방 타입 선택<<");
    System.out.println("1.스위트 2.디럭스 3.스탠다드");
    int roomch = TestMain.scanner.nextInt();

    if (hotelch == 1) {
        호텔예약 = new 호텔1();
        for (int i = 1; i <= 3; i++) {
            if (roomch == 1) {
                호텔예약.예약취소(hotelch, roomch, h_id);
            }
        }
    } else if (hotelch == 2) {
        호텔예약 = new 호텔2();
        for (int i = 1; i <= 3; i++) {
            if (roomch == 1) {
                호텔예약.예약취소(hotelch, roomch, h_id);
            }
        }
    } else if (hotelch == 3) {
        호텔예약 = new 호텔3();
        for (int i = 1; i <= 3; i++) {
            if (roomch == 1) {
                호텔예약.예약취소(hotelch, roomch, h_id);
            }
        }
    } else {
        System.err.println(" 잘못된 입력번호입니다. 번호를 다시 선택해주세요. ");
    }
}

else if (ch2 == 3) {
    호텔예약 = new 호텔3();
    for (int i = 1; i <= 3; i++) {
        if (roomch == 1) {
            호텔예약.예약취소(hotelch, roomch, h_id);
        }
    }
}

else {
    System.err.println(" 잘못된 입력번호입니다. 번호를 다시 선택해주세요. ");
}

System.out.println("-----");
System.out.println("\t\t본인예약조회");
System.out.println("-----");
호텔예약 = new 호텔2();
호텔예약.본인예약조회(h_id);
호텔예약 = new 호텔2();
호텔예약.본인예약조회(h_id);
호텔예약 = new 호텔3();
호텔예약.본인예약조회(h_id);

} else if (ch2 == 4) { // 전체예약조회
    System.out.println("-----");
    System.out.println("\t\t전체예약조회");
    System.out.println("-----");
    호텔예약 = new 호텔2();
    호텔예약.전체예약조회(h_id);
    호텔예약 = new 호텔2();
    호텔예약.전체예약조회(h_id);
    호텔예약 = new 호텔3();
    호텔예약.전체예약조회(h_id);

} else if (ch2 == 5) { // 객실정보
    System.out.println("-----");
    System.out.println("\t\t객실정보");
    System.out.println("-----");
    호텔예약 = new 호텔2();
    호텔예약.객실정보(h_id);
    호텔예약 = new 호텔2();
    호텔예약.객실정보(h_id);
    호텔예약 = new 호텔3();
    호텔예약.객실정보(h_id);

} else if (ch2 == 6) { // 뒤로가기
    System.out.println("-----");
    System.out.println("\t\t뒤로가기");
    System.out.println("-----");
    호텔예약 = new 호텔2();
    호텔예약.뒤로가기(h_id);
    호텔예약 = new 호텔2();
    호텔예약.뒤로가기(h_id);
    호텔예약 = new 호텔3();
    호텔예약.뒤로가기(h_id);

} else {
    System.err.println(" 잘못된 입력번호입니다. 번호를 다시 선택해주세요. ");
}

} else { // 로그인 실패시
    System.err.println(" 아이디를 다시 입력해주세요. ");
}
}
    
```

```

else if (ch2 == 2) { // 예약취소
    System.out.println(">>>호텔지점 선택<<");
    System.out.println("1.신라호텔 2.라마다호텔 3.그랜드호텔");
    int hotelch = TestMain.scanner.nextInt();
    System.out.println(">>>방 타입 선택<<");
    System.out.println("1.스위트 2.디럭스 3.스탠다드");
    int roomch = TestMain.scanner.nextInt();

    if (hotelch == 1) {
        호텔예약 = new 호텔1();
        for (int i = 1; i <= 3; i++) {
            if (roomch == 1) {
                호텔예약.예약취소(hotelch, roomch, h_id);
            }
        }
    } else if (hotelch == 2) {
        호텔예약 = new 호텔2();
        for (int i = 1; i <= 3; i++) {
            if (roomch == 1) {
                호텔예약.예약취소(hotelch, roomch, h_id);
            }
        }
    } else if (hotelch == 3) {
        호텔예약 = new 호텔3();
        for (int i = 1; i <= 3; i++) {
            if (roomch == 1) {
                호텔예약.예약취소(hotelch, roomch, h_id);
            }
        }
    } else {
        System.err.println(" 잘못된 입력번호입니다. 번호를 다시 선택해주세요. ");
    }
}

else if (ch2 == 3) {
    호텔예약 = new 호텔3();
    for (int i = 1; i <= 3; i++) {
        if (roomch == 1) {
            호텔예약.예약취소(hotelch, roomch, h_id);
        }
    }
}

else {
    System.err.println(" 잘못된 입력번호입니다. 번호를 다시 선택해주세요. ");
}

System.out.println("-----");
System.out.println("\t\t본인예약조회");
System.out.println("-----");
호텔예약 = new 호텔2();
호텔예약.본인예약조회(h_id);
호텔예약 = new 호텔2();
호텔예약.본인예약조회(h_id);
호텔예약 = new 호텔3();
호텔예약.본인예약조회(h_id);

} else if (ch2 == 4) { // 전체예약조회
    System.out.println("-----");
    System.out.println("\t\t전체예약조회");
    System.out.println("-----");
    호텔예약 = new 호텔2();
    호텔예약.전체예약조회(h_id);
    호텔예약 = new 호텔2();
    호텔예약.전체예약조회(h_id);
    호텔예약 = new 호텔3();
    호텔예약.전체예약조회(h_id);

} else if (ch2 == 5) { // 객실정보
    System.out.println("-----");
    System.out.println("\t\t객실정보");
    System.out.println("-----");
    호텔예약 = new 호텔2();
    호텔예약.객실정보(h_id);
    호텔예약 = new 호텔2();
    호텔예약.객실정보(h_id);
    호텔예약 = new 호텔3();
    호텔예약.객실정보(h_id);

} else if (ch2 == 6) { // 뒤로가기
    System.out.println("-----");
    System.out.println("\t\t뒤로가기");
    System.out.println("-----");
    호텔예약 = new 호텔2();
    호텔예약.뒤로가기(h_id);
    호텔예약 = new 호텔2();
    호텔예약.뒤로가기(h_id);
    호텔예약 = new 호텔3();
    호텔예약.뒤로가기(h_id);

} else {
    System.err.println(" 잘못된 입력번호입니다. 번호를 다시 선택해주세요. ");
}

} else { // 로그인 실패시
    System.err.println(" 아이디를 다시 입력해주세요. ");
}
}
    
```



## 2. 코드설계

### 2-1) Main Class - 관리자

#### - 6. 로그인 관리자메뉴

> 관리자 로그인

: (ID:admin , PW:1234) 설정값 고정

및 그 외의 입력값에 대한 예외처리

> 전제메뉴 while (true) { if (ch1 == 1~3) }

1. 예약현황 2.회원정보 3.뒤로가기

```

else if (ch1 == 2) { // 관리자
    System.out.println(" ID : ");
    String h_id = scanner.next();
    System.out.println(" Password : ");
    String h_pw = scanner.next();
    if (h_id.equals("admin")) {
        System.err.println(" 관리자가 로그인하셨습니다. ");
        while (true) {
            System.out.println("\t1.예약현황\t2.회원정보\t3.뒤로가기");
            System.out.print(">>> 선택 : ");
            int ch3 = scanner.nextInt();
            if (ch3 == 1) {
                System.out.println("호텔\ts우루름\ts디럭스룸\ts스탠다드룸");
                호텔예약 = new 호텔1();
                호텔예약.전체예약현황.관리자();
                호텔예약 = new 호텔2();
                호텔예약.전체예약현황.관리자();
                호텔예약 = new 호텔3();
                호텔예약.전체예약현황.관리자();
            } else if (ch3 == 2) {
                Member member = new Member(h_id, h_pw, h_id, h_pw);
                for (Member temp : MemberController.memberlist) {
                    if (temp.getId().equals("admin")) {
                        System.out.println("ID : " + temp.getId() + " / 이름 : " + temp.getName()
                            + " / 연락처 : " + temp.getPhone());
                    }
                }
            } else if (ch3 == 3) {
                System.err.println(" 관리자가 로그아웃하셨습니다. ");
                break;
            } else { // 관리자 선택
                System.err.println(" 다시 입력해주세요.");
            }
        }
    } else {
        System.err.println(" 잘못입력하셨습니다. 초기화면으로 돌아갑니다. ");
    }
} else if (ch == 3) {
    System.err.println(" 종료 ");
    break;
} else {
    System.err.println(" 다시 입력해주세요. ");
}
} catch (Exception e) {
    System.err.println(" [입력] : 메뉴 입력번호 오류 [ 관리자용의 ] ");
    scanner = new Scanner(System.in);
} // 입력력제 초기화
}
}
    
```



## 2. 코드설계

□ Console 구현  
[로그인]

```
===== 호텔예약시스템 =====
===== main menu =====
      1.회원가입      2.로그인      3.종료
>>> 선택 : 2
===== 로그인 페이지 =====
      1.회원      2.관리자
선택 : 1|
```



1. [회원 로그인]

2. [관리자 로그인]

```
===== 호텔예약시스템 =====
===== main menu =====
      1.회원가입      2.로그인      3.종료
>>> 선택 : 2
===== 로그인 페이지 =====
      1.회원      2.관리자
선택 : 1
Id : ezen
Password : ezen
[알림] : 로그인 성공
1. 예약등록 2. 예약취소 3. 본인예약조회
4. 전체예약조회 5. 객실세부정보 6. 뒤로가기
>>> 선택 :
```

```
===== 호텔예약시스템 =====
===== main menu =====
      1.회원가입      2.로그인      3.종료
>>> 선택 : 2
===== 로그인 페이지 =====
      1.회원      2.관리자
선택 : 2
Id : admin
Password : 1234
관리자가 로그인하셨습니다.
1. 예약현황 2. 회원정보 3. 뒤로가기
>>> 선택 :
```

□ Console 구현

[로그인] : 성공시

1.회원

< 1. 예약등록 2. 예약취소 3. 본인예약조회  
4. 전체예약조회 5. 객실세부정보 6.  
뒤로가기 >

2. 관리자

< 1. 예약현황 2. 회원정보 3. 뒤로가기 >





## 2. 코드설계

### 2-2) Member controller class

#### - 1. Member 클래스의 필드를 get호출로

메소드하여 유효성 검사 논리제어

#### - 2. MainTest 에서 member 필드 내

변수를 객체화 하여 관리.

```
// 4개 변수를 따로따로 저장하면 힘들어짐 => 4개 변수를 묶어서 객체화
Member member = new Member(h_id, h_pw, h_name, h_phone);
```

#### - 3. 유효성 검사 / 로그인 중복체크

/ Admin 관리자 static 설정(id:admin pw:1234)

-> MainTest 에서 호출

```
// Controller 전달 [ 다른 클래스나 메소드 호출하는 방법 : 1. 2.
boolean result = MemberController.signup(member);
if (result) {
    System.err.println(" [알림] : 회원가입 성공 ");
} else {
    System.err.println(" [알림] : 회원가입 실패 ");
}
```

#### - 4. 리스트저장으로 회원 가입자 add

File클래스 내 저장 filesave(int type)연계

```
1 package 김태호;
2
3 import java.util.ArrayList;
4
5
6 public class MemberController {
7     // File 내 멤버리스트 동기화하기 위해 Vector 사용
8
9     // 1. 회원가입, 회원가입, 회원가입, 회원가입, 회원가입, 회원가입
10    public static Vector<Member> memberlist = new Vector<>();
11    public static Scanner scanner = new Scanner(System.in);
12
13    // 2. 메소드
14
15    // 1. 회원가입
16    public static boolean signup() {
17        // 반환타입 :
18        // * 입력받기 -> 변수 저장 -> 변수가 여러개 -> 객체 -> 객체 여러개 -> 배열/컬렉션
19        System.out.print(" Id[4글자이상] : "); String h_id = scanner.next();
20        System.out.print(" Password[4글자만] : "); String h_pw = scanner.next();
21        System.out.print(" name[2글자이상] : "); String h_name = scanner.next();
22        System.out.print(" HP Number : "); String h_phone = scanner.next();
23
24        // 4개 변수를 따로따로 저장하면 힘들어짐 => 4개 변수를 묶어서 객체화
25        Member member = new Member(h_id, h_pw, h_name, h_phone);
26
27        // 1. 유효성검사
28        if (member.getH_id().length() < 4) {
29            System.err.println(" [알림] : ID는 4글자이상 가능 합니다. ");
30            return false;
31        }
32        if (member.getH_id().equals("admin")) {
33            System.err.println(" 사용할 수 없는 아이디입니다. ");
34            return false;
35        }
36        if (member.getH_pw().length() != 4) {
37            System.err.println(" [알림] : Password는 4글자만 입니다. ");
38            return false;
39        }
40        if (member.getH_name().length() < 2) {
41            System.err.println(" [알림] : Name는 2글자이상 가능 합니다. ");
42            return false;
43        }
44    }
```

```
// 2. ID 중복체크 [ '리스트'에 '중첩'된 '아이디'가 '있는지' 확인 ]
for (Member temp : memberlist) {
    if (temp.getH_id().equals(member.getH_id())) {
        System.err.println(" [알림] : 이미 사용 중인 아이디 입니다. ");
        return false;
    }
}
```

```
// 3. 리스트 저장
memberlist.add(member);

File.filesave(1);

return true; // 회원가입 성공시
```

```
// 2. 로그인
public static boolean login(String h_id, String h_pw) {
    for (Member member : memberlist) {
        if (h_id.equals("admin")) {
            if (h_id.equals("admin")) {
                System.err.println(" 사용할 수 없는 아이디입니다. ");
                return false;
            } else if (member.getH_id().equals(h_id)
                && member.getH_pw().equals(h_pw)) {
                return true; // 로그인 성공시
            }
        }
    }
    return false; // 로그인 실패시
}
```



## 2. 코드설계

### 2-3) Hotel interface - 주요코드

#### - 1. 추상 메소드 선언

(예약등록 / 예약취소 / 본인예약조회 /

전체예약현황(회원) / 전체예약현황(관리자) )

#### - 2. 호텔예약 (Interface )

> 호텔TOTAL(Implements)

> 호텔1,2,3 (extends)

```
1 package 김태호;
2
3 import java.io.IOException;
4
5 public interface 호텔예약 {
6
7     // 추상메소드
8     public void 예약등록(int hotelch, int roomch, String h_id);
9
10    public void 예약취소(int hotelch, int roomch, String h_id);
11
12    public void 본인예약조회(String h_id);
13
14    public void 전체예약현황_회원();
15
16    public void 전체예약현황_관리자();
17
18 }
```

## 2. 코드설계

### 2-4) Member class

- 1. 호텔 예약을 위한 회원관리 구성

- (1) 필드
- (2) 생성자
- (3) 메소드

```

1 package 김태호;
2
3 import java.io.FileInputStream;
4
5
6 public class Member {
7
8     // 1. 필드
9     private String h_id;
10    private String h_pw;
11    private String h_name;
12    private String h_phone;
13
14    // 2. 생성자
15    public Member() {
16
17    }
18
19    public Member(String h_id, String h_pw, String h_name, String h_phone) {
20
21        super();
22        this.h_id = h_id;
23        this.h_pw = h_pw;
24        this.h_name = h_name;
25        this.h_phone = h_phone;
26    }
27
28    // 3. 메소드
29
30    public String getH_id() {
31        return h_id;
32    }
33
34    public void setH_id(String h_id) {
35        this.h_id = h_id;
36    }
37
38    public String getH_pw() {
39        return h_pw;
40    }
41
42    public void setH_pw(String h_pw) {
43        this.h_pw = h_pw;
44    }
45
46    public String getH_name() {
47        return h_name;
48    }
49
50    public void setH_name(String h_name) {
51        this.h_name = h_name;
52    }
53
54    public String getH_phone() {
55        return h_phone;
56    }
57
58 }

```



## 2. 코드설계

### 2-5) 호텔TOTAL class

- 1. 호텔 예약을 위한 주요 메소드 설계

- 2. 3종류의 호텔 3type의 방을 구현하기 위해

=> String room [hotelch][roomch] 배열

=> 출력 시 예약 전/ 후의 상황을 표시하기 위해

빈공간을 허용하는 배열로 설정.

=> Hotel (1~3) 각 객체로 상속을 주기위함

```

1 package 김태호;
2
3 import java.util.ArrayList;
4
5 public class 호텔Total implements 호텔예약 {
6
7     // 필드
8     public static String[][] room = new String[3][3];
9
10    // 메소드
11    @Override
12    public void 예약등록(int hotelch, int roomch, String h_id) {
13
14        if (room[hotelch - 1][roomch - 1] == null) {
15            System.err.println(" *** 호텔 / ** 룸 예약 완료 *** ");
16            room[hotelch - 1][roomch - 1] = h_id;
17        }
18    }
19
20    @Override
21    public void 예약취소(int hotelch, int roomch, String h_id) {
22        if (room[hotelch - 1][roomch - 1] == null) {
23            System.out.println(" *** 호텔 / ** 룸 예약취소 완료 *** ");
24            room[hotelch - 1][roomch - 1] = h_id;
25        }
26        System.out.println(" *** 예약 취소 완료 *** ");
27    }
28
29    @Override
30    public void 본인예약조회(String h_id) {
31        Member member = new Member();
32        for (int i = 0; i < 3; i++) {
33            for (int j = 0; j < 3; j++) {
34                if (room[i][j] != null) {
35                    System.err.println(room[0][i]);
36                }
37            }
38        }
39    }
40
41    @Override
42    public void 전체예약현황_회원() {
43        System.err.print("호텔 **\t");
44        for (int i = 0; i < 3; i++) {
45            if (room[0][i] == null) {
46                System.out.print("□\t");
47            } else {
48                System.out.print("■\t");
49            }
50        }
51        System.out.println();
52    }
53
54    @Override
55    public void 전체예약현황_관리자() {
56        System.out.print("호텔 **\t");
57        for (int i = 0; i < 3; i++) {
58            if (room[0][i] == null) {
59                System.out.print("□\t");
60            } else {
61                System.out.print(room[0][i]);
62            }
63        }
64    }
65
66    }
67
68 }

```



## 2. 코드설계

### 2-6~8' 호텔1~3 class

- 1. 상속받은 호텔Total 의 room 변수의

room [hotelch][roomch] 에서

roomch의 3가지 방타입

< 스위트룸 / 디럭스룸 / 스탠다드룸 > 을

호텔(1, 2, 3)별로 적용

호텔1 : room [0][i=1,i=2,i=3]

호텔2 : room [1][i=1,i=2,i=3]

호텔3 : room [2][i=1,i=2,i=3]

M(데이터)  
\*database

6. Hotel1 class  
7. Hotel1 class  
8. Hotel1 class

```
1 package 김태우;
2
3 import java.util.ArrayList;
4
5 public class 호텔1 extends 호텔Total {
6
7     // 메소드
8     @Override
9     public void 예약등록(int hotelch, int roomch, String h_id) {
10         if (roomch == 1) {
11             if (room[hotelch - 1][roomch - 1] == null) {
12                 System.err.println(" [[예약 완료]] : 신라호텔 / 스위트룸 ");
13                 room[hotelch - 1][roomch - 1] = h_id;
14             } else {
15                 System.err.println(" 이미 예약된 객실입니다. ");
16             }
17         } else if (roomch == 2) {
18             if (room[hotelch - 1][roomch - 1] == null) {
19                 System.err.println(" [[예약 완료]] : 신라호텔 / 디럭스룸 ");
20                 room[hotelch - 1][roomch - 1] = h_id;
21             } else {
22                 System.err.println(" 이미 예약된 객실입니다. ");
23             }
24         } else if (roomch == 3) {
25             if (room[hotelch - 1][roomch - 1] == null) {
26                 System.err.println(" [[예약 완료]] : 신라호텔 / 스탠다드룸 ");
27                 room[hotelch - 1][roomch - 1] = h_id;
28             } else {
29                 System.err.println(" 이미 예약된 객실입니다. ");
30             }
31         }
32     }
33
34     @Override
35     public void 예약취소(int hotelch, int roomch, String h_id) {
36         if (roomch == 1) {
37             if (room[hotelch - 1][roomch - 1].equals(h_id)) {
38                 System.err.println(" [[예약 취소]] : 신라호텔 / 스위트룸 ");
39                 room[hotelch - 1][roomch - 1] = null;
40             } else {
41                 System.err.println(" 다시 입력바랍니다. ");
42             }
43         } else if (roomch == 2) {
44             if (room[hotelch - 1][roomch - 1].equals(h_id)) {
45                 System.err.println(" [[예약 취소]] : 신라호텔 / 디럭스룸 ");
46                 room[hotelch - 1][roomch - 1] = null;
47             } else {
48                 System.err.println(" 다시 입력바랍니다. ");
49             }
50         } else if (roomch == 3) {
51             if (room[hotelch - 1][roomch - 1].equals(h_id)) {
52                 System.err.println(" [[예약 취소]] : 신라호텔 / 스탠다드룸 ");
53                 room[hotelch - 1][roomch - 1] = null;
54             } else {
55                 System.err.println(" 다시 입력바랍니다. ");
56             }
57         }
58     }
59 }
60 }
```

```
69 @Override
70 public void 본인예약조회(String h_id) {
71     Member member = new Member();
72     for (int i = 0; i < 3; i++) {
73         if (room[0][i] != null && room[0][i].equals(h_id)) {
74             if (i == 0) {
75                 System.out.println(" *** 신라호텔 / 스위트룸 "
76                     + room[0][i] + " 님 예약 *** ");
77             } else if (i == 1) {
78                 System.out.println(" *** 신라호텔 / 디럭스룸 "
79                     + room[0][i] + " 님 예약 *** ");
80             } else if (i == 2) {
81                 System.out.println(" *** 신라호텔 / 스탠다드룸 "
82                     + room[0][i] + " 님 예약 *** ");
83             }
84         }
85     }
86 }
87
88 }
89
90 @Override
91 public void 전체예약현황_회원() {
92     System.out.print("신라호텔\t");
93     for (int i = 0; i < 3; i++) {
94         if (room[0][i] == null) {
95             System.out.print("\t\t");
96         } else {
97             System.out.print("\t\t\t");
98         }
99     }
100     System.out.println();
101 }
102
103 @Override
104 public void 전체예약현황_관리자() {
105     System.out.print("신라호텔\t");
106     for (int i = 0; i < 3; i++) {
107         if (room[0][i] == null) {
108             System.out.print("\t\t\t");
109         } else {
110             System.out.print("\t\t+room[0][i] + "\t");
111         }
112     }
113     System.out.println();
114 }
115 }
116
117 }
118 }
```



## 2. 코드설계



### 2-9) Room class

- 1. MainTest 에서 회원 로그인후 메뉴에서

5번 메뉴 객실정보 보기를 위한 클래스

- Room 클래스에서 room 타입별 변수 저장

String[] 스탠다드 = new String[3]{"방타입", "객실면적", "부가혜택" }

String[] 디럭스 = new String[3]{"방타입", "객실면적", "부가혜택" }

String[] 스위트 = new String[3]{"방타입", "객실면적", "부가혜택" }

#### ■ TestMain (ch2 == 5)

```

164         } else if (ch2 == 5) { // 방정보
165             System.out.println(" 방정보 : 1.스탠다드 2.디럭스 3.스위트 ");
166             System.out.println(">>> 선택 : ");
167             int ch3 = scanner.nextInt();
168
169             if (ch3 == 1) {
170                 Room.스탠다드();
171             } else if (ch3 == 2) {
172                 Room.디럭스();
173             } else if (ch3 == 3) {
174                 Room.스위트();
175             } else {
176                 System.err.println(" 방을 다시 선택해주세요. ");
177                 ch3 = scanner.nextInt();
178             }
179         }
180     }
181 }

```

#### ■ Room. 스탠다드

```

1 package 실패도;
2
3 public class Room {
4     // 룸타입
5     public static String[] 카테고리 = { "방타입", "객실면적", "부가혜택" };
6
7     // 메소드
8     public static void 스탠다드() {
9         String[] 스탠다드 = new String[3];
10
11         // 스탠다드
12         스탠다드[0] = "스탠다드";
13         스탠다드[1] = "12평";
14         스탠다드[2] = "32 인치 이상의 TV, 별 난방 조절기, 미니바, 생수, 에스프레소 커피 머신, "
15             + "보스 사운드링크 미니 블루투스 스피커, 다리미와 다리미 받, 객실 내 금고, 조강 신은 (요청 시 제공), "
16             + "유아용품대 (요청 시 제공), 르 라보 (Le Labo) 향수용품, 욕조가운과 슬리퍼";
17
18         System.out.println("[[ 객실세부정보 ]]");
19         for (int i = 0; i < 3; i++) {
20             System.out.print((i + 1) + " + " + 카테고리[i] + " : ");
21             if (i != 2) {
22                 System.out.println(스탠다드[i]);
23             } else {
24                 String[] strArray = 스탠다드[i].split(",");
25                 int k = 1;
26                 System.out.println();
27                 for (String temp : strArray) {
28                     System.out.println(" " + k + " + " + temp);
29                     k++;
30                 }
31             }
32         }
33     }
34 }

```

#### ■ Room. 디럭스룸

```

340 static void 디럭스() {
341     String[] 디럭스 = new String[3];
342
343     // 디럭스
344     디럭스[0] = "디럭스룸";
345     디럭스[1] = "15평";
346     디럭스[2] = "32 인치 이상의 TV, 별 난방 조절기, 미니바, 생수, 에스프레소 커피 머신, "
347         + "보스 사운드링크 미니 블루투스 스피커, 다리미와 다리미 받, 객실 내 금고, 조강 신은 (요청 시 제공), "
348         + "유아용품대 (요청 시 제공), 르 라보 (Le Labo) 향수용품, 욕조가운과 슬리퍼";
349
350     System.out.println("[[ 객실세부정보 ]]");
351     for (int i = 0; i < 3; i++) {
352         System.out.print((i + 1) + " + " + 카테고리[i] + " : ");
353         if (i != 2) {
354             System.out.println(디럭스[i]);
355         } else {
356             String[] strArray = 디럭스[i].split(",");
357             int k = 1;
358             System.out.println();
359             for (String temp : strArray) {
360                 System.out.println(" " + k + " + " + temp);
361                 k++;
362             }
363         }
364     }
365 }

```

#### ■ Room. 스위트룸

```

610 static void 스위트() {
611     String[] 스위트 = new String[3];
612
613     // 스위트
614     스위트[0] = "스위트룸";
615     스위트[1] = "34평";
616     스위트[2] = "구분형 식자공간과 미니 주방, 대형 테이블과 사무공간, 침실, 욕장의 이층 아메니티, "
617         + "LED TV (화상: 65인치 / 해상: 55인치), 가전 난방 조리기, 미니바와 생수, 에스프레소 커피 머신, "
618         + "보스 사운드링크 미니 블루투스 스피커, 링 & 슬립스 오디오 시스템, 다리미와 다리미 받, 객실 내 금고, "
619         + "유아용품대 (요청 시 제공), 다이슨 슈퍼소닉(Dyson Supersonic) 헤어드라이어";
620
621     System.out.println("[[ 객실세부정보 ]]");
622     for (int i = 0; i < 3; i++) {
623         System.out.print((i + 1) + " + " + 카테고리[i] + " : ");
624         if (i != 2) {
625             System.out.println(스위트[i]);
626         } else {
627             String[] strArray = 스위트[i].split(",");
628             int k = 1;
629             System.out.println();
630             for (String temp : strArray) {
631                 System.out.println(" " + k + " + " + temp);
632                 k++;
633             }
634         }
635     }
636 }

```



### - 3. 불러오기 (데이터 유지)

```

66 // 3. 클라이언트 요청
67 public synchronized boolean fileLoad(int type) {
68     try {
69         fileInputStream = fileInputStream == null;
70         if (type == 1) {
71             // 1. 입력스트림 경로 설정
72             fileInputStream = new FileInputStream(memberpath);
73             // 2. 스트림(파일) 바이트로 읽어들이기 선언
74             byte[] bytes = new byte[10000]; // 10kb 정도
75             // 3. 입력스트림에서 바이트 읽어들이기 배열에 저장
76             fileInputStream.read(bytes);
77             // 4. 바이트배열 -> 문자열 변환
78             String instr = new String(bytes);
79             // 5. 회원 분리하기 \n
80             String[] members = instr.split("\n"); // \n 구분시 공백회원 추가
81             // 6. 반복문을 이용한 회원별 필드 분리하기
82             for (int i = 0; i < members.length - 1; i++) { // -1 : 공백회원 제외
83                 // 7. 회원별 필드 분리
84                 String[] field = members[i].split(",");
85                 // 8. 분리된 필드를 객체로 ( point필드는 \t값으로 변환 : String -> \t Integer.parseInt
86                 Member member = new Member(field[0], field[1], field[2], field[3]);
87                 // 9. 객체들 리스트에 저장
88                 MemberController.memberList.add(member);
89             }
90             fileInputStream.close(); // 스트림 닫기
91             return true; // 파일 불러오기 성공
92         }
93         if (type == 2) {
94             // 1. 입력스트림 경로 설정
95             fileInputStream = new FileInputStream(roompath);
96             // 2. 스트림(파일) 바이트로 읽어들이기 선언
97             byte[] bytes = new byte[10000]; // 10kb 정도
98             // 3. 입력스트림에서 바이트 읽어들이기 배열에 저장
99             fileInputStream.read(bytes);
100             // 4. 바이트배열 -> 문자열 변환
101             String instr = new String(bytes);
102             // 5. 회원 분리하기 \n
103             String[] rooms = instr.split("\n"); // \n 구분시 공백회원 추가
104             // 6. 반복문을 이용한 회원별 필드 분리하기
105             for (int i = 0; i < rooms.length - 1; i++) { // -1 : 공백회원 제외
106                 // 7. 회원별 필드 분리
107                 String[] room = rooms[i].split(",");
108                 for (int j = 0; j < room.length; j++) {
109                     if (room[j].equals("null")) {
110                         총합Total.room[i][j] = room[j];
111                     }
112                 }
113             }
114             fileInputStream.close(); // 스트림 닫기
115             return true; // 파일 불러오기 성공
116         }
117     } catch (Exception e) {
118         System.out.println(" [입력] : 파일 불러오기 오류 발생 [ 관리자에서 문의 ]");
119     }
120     return false; // 파일 불러오기 실패시
121 }
122 }

```



## 2. 코드설계

### 2-11) Thread class

- 1. 예약정보 파일을 처리하는 속도가 빨라서 **오류**  
(roomlist의 배열이 밀리는 현상 및 roomlist의 내용을  
읽어오거나 저장하는데 문제가 생김) 를  
해결하고자 인터벌(1s) 간격을 두어 순차적 실행

ex. 오류 출력화면 > 보완

```

1 null,null,null
2 null,null,null
3 null,null,null
4 |

```

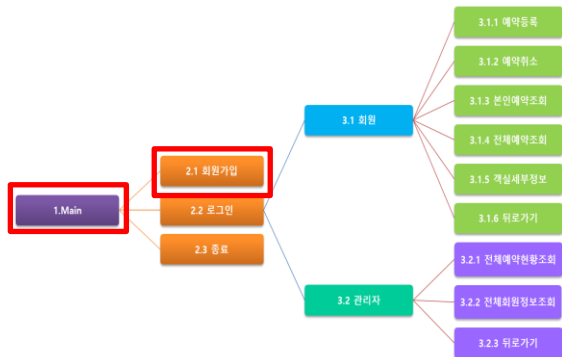
```

1 package 김태호;
2
3 public class FileThread extends Thread {
4
5     @Override
6     public void run() {
7         while (true) {
8
9             File.fileload(2);
10            try {
11                Thread.sleep(1000);
12            } catch (Exception e) {
13                System.err.println("처리중");
14            }
15
16            File.filesave(2);
17            try {
18                Thread.sleep(1000);
19            } catch (Exception e) {
20                System.err.println("처리중");
21            }
22        }
23    }
24
25 }
26

```

### 3. Console 화면구현

#### 3-1) 메뉴 및 회원가입



#### ■ 1. Main 메뉴

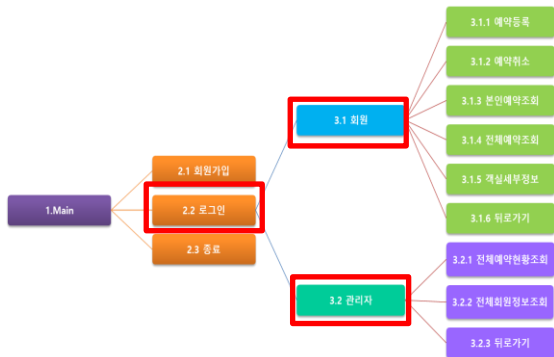
```
===== 호텔예약시스템 =====
===== main menu =====
1.회원가입      2.로그인      3.종료
>>> 선택 : 1
```

#### ■ 2.1 회원가입 (일반회원) : 유효성검사

```
===== 호텔예약시스템 =====
===== main menu =====
1.회원가입      2.로그인      3.종료
>>> 선택 : 1
===== 회원가입 페이지 =====
Id[4글자이상] : ezen
Password[4글자만] : ezen
name[2글자이상] : ezen
HP Number : 010-1234-1234
[입력] : 회원가입 성공
===== 호텔예약시스템 =====
===== main menu =====
1.회원가입      2.로그인      3.종료
>>> 선택 :
```

### 3. Console 화면구현

#### 3-2) 로그인메뉴 (회원, 관리자)



#### ■ 2.2 로그인

```
===== 로그인 페이지 =====
                        1. 회원      2. 관리자
=====
선택 :
```

#### ■ 3.1 로그인(회원)

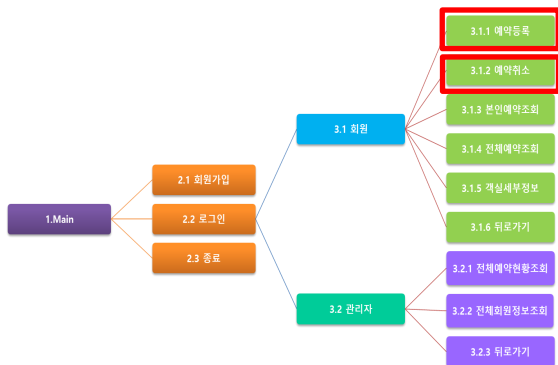
```
===== 로그인 페이지 =====
                        1. 회원      2. 관리자
=====
선택 : 1
=====
Id : ezen
Password : ezen
[알림] : 로그인 성공
=====
1.예약등록 2.예약취소 3.본인예약조회
4. 전체예약조회 5. 객실세부정보 6. 뒤로가기
=====
>>> 선택 :
```

#### ■ 3.2 로그인(관리자) : admin / 1234 고정

```
===== 호텔예약시스템 =====
===== main menu =====
                        1.회원가입      2.로그인      3.종료
>>> 선택 : 2
===== 로그인 페이지 =====
                        1.회원      2. 관리자
선택 : 2
Id : admin
Password : 1234
관리자가 로그인하셨습니다.
1.예약현황 2.회원정보 3. 뒤로가기
>>> 선택 :
```

### 3. Console 화면구현

#### 3-3) 예약 등록 및 취소 (일반회원)



##### ■ 3.1.1 예약등록 (ex:신라호텔 / 스위트룸)

```

=====
1.예약등록 2.예약취소 3.본인예약조회
4.전체예약조회 5.객실세부정보 6.뒤로가기
=====
>>> 선택 : 1
=====
>>호텔지점 선택<<
1.신라호텔 2.라마다호텔 3.그랜드호텔
1
=====
>>방 타입 선택<<
1.스위트 2.디럭스 3.스탠다드
1
| [[예약 완료]] : 신라호텔 / 스위트룸
=====
1.예약등록 2.예약취소 3.본인예약조회
4.전체예약조회 5.객실세부정보 6.뒤로가기
=====
>>> 선택 :
  
```

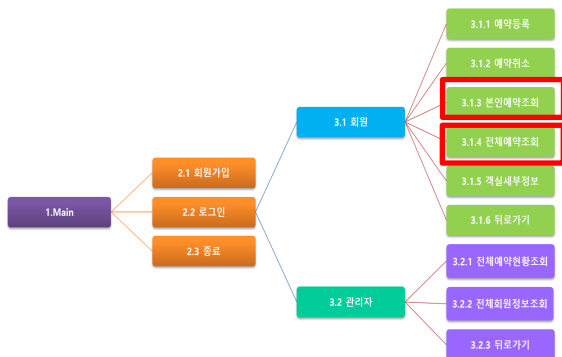
##### ■ 3.1.1 예약취소 (ex:신라호텔 / 스위트룸)

```

=====
1.예약등록 2.예약취소 3.본인예약조회
4.전체예약조회 5.객실세부정보 6.뒤로가기
=====
>>> 선택 : 2
=====
>>호텔지점 선택<<
1.신라호텔 2.라마다호텔 3.그랜드호텔
1
=====
>>방 타입 선택<<
1.스위트 2.디럭스 3.스탠다드
1
| [[예약 취소]] : 신라호텔 / 스위트룸
=====
1.예약등록 2.예약취소 3.본인예약조회
4.전체예약조회 5.객실세부정보 6.뒤로가기
=====
>>> 선택 :
  
```

### 3. Console 화면구현

#### 3-4) 예약 조회 (일반회원)



##### ■ 3.1.3 본인예약조회

```

=====
                          본인예약현황조회
=====
*** 신라호텔 / 스위트룸 ezen 님 예약 ***
=====

1.예약등록 2.예약취소 3.본인예약조회
4.전체예약조회 5.객실세부정보 6.뒤로가기

>>> 선택 :
  
```

##### ■ 3.1.4 전체예약조회

```

=====
                          전체예약현황조회
=====
호텔          스위트룸          디럭스룸          스탠다드룸
=====
신라호텔      ■                  □                  □
라마다호텔    □                  □                  □
그랜드호텔    □                  □                  □
=====

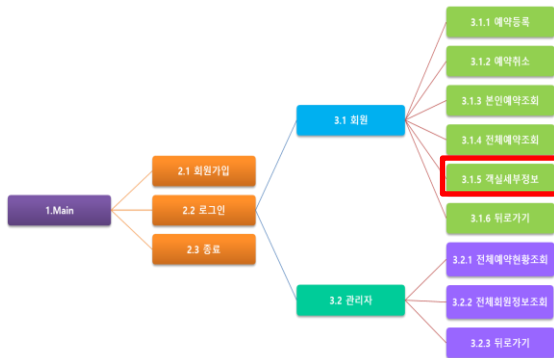
1.예약등록 2.예약취소 3.본인예약조회
4.전체예약조회 5.객실세부정보 6.뒤로가기

>>> 선택 :
  
```



# 3. Console 화면구현

## 3-5) 객실정보 (일반회원)



### ■ 3.1.5 객실세부정보 [메뉴]

```

=====
1.예약등록 2.예약취소 3.본인예약조회
4.전체예약조회 5.객실세부정보 6.뒤로가기
=====
>>> 선택 : 5
=====
방정보 : 1.스탠다드 2.디럭스 3.스위트
>>> 선택 :
  
```

### ■ 3.1.5 - 1 : 스탠다드

```

=====
1.예약등록 2.예약취소 3.본인예약조회
4.전체예약조회 5.객실세부정보 6.뒤로가기
=====
>>> 선택 : 5
=====
방정보 : 1.스탠다드 2.디럭스 3.스위트
>>> 선택 :
2
=====
[[ 객실세부정보 ]]
1.방타입 : 디럭스룸
2.객실면적 : 15평
3.부가혜택 :
  1)32인치 이상의 TV
  2)별 난방 조절기
  3)미니바
  4)생수
  5)에스프레소 커피 머신
  6)보스 사운드필드 미니 블루투스 스피커
  7)카리마와 다리미 판
  8)객실 내 금고
  9)조간 신문 (요청 시 제공)
  10)유아용품다 (요청 시 제공)
  11)트 라브 (Le Labo™) 욕품용품
  12)욕품가운과 슬리퍼
  
```

```

=====
1.예약등록 2.예약취소 3.본인예약조회
4.전체예약조회 5.객실세부정보 6.뒤로가기
=====
>>> 선택 : 5
=====
방정보 : 1.스탠다드 2.디럭스 3.스위트
>>> 선택 :
1
=====
[[ 객실세부정보 ]]
1.방타입 : 스탠다드룸
2.객실면적 : 12평
3.부가혜택 :
  1)32인치 이상의 TV
  2)별 난방 조절기
  3)미니바
  4)생수
  5)에스프레소 커피 머신
  6)보스 사운드필드 미니 블루투스 스피커
  7)카리마와 다리미 판
  8)객실 내 금고
  9)조간 신문 (요청 시 제공)
  10)유아용품다 (요청 시 제공)
  11)트 라브 (Le Labo™) 욕품용품
  
```

```

=====
1.예약등록 2.예약취소 3.본인예약조회
4.전체예약조회 5.객실세부정보 6.뒤로가기
=====
>>> 선택 : 5
=====
방정보 : 1.스탠다드 2.디럭스 3.스위트
>>> 선택 :
3
=====
[[ 객실세부정보 ]]
1.방타입 : 스위트룸
2.객실면적 : 34평
3.부가혜택 :
  1)구분된 식사공간과 미니 주방
  2)직접 테이블과 사용공간
  3)정물 출품의 이층 아메니티
  4)LED TV (거실: 65인치 / 침실: 55인치)
  5)가별 난방 조절기
  6)미니바와 생수
  7)에스프레소 커피 머신
  8)보스 사운드필드 미니 블루투스 스피커
  9)왕 & 풀루른 오디오 시스템
  10)다리미와 다리미 판
  11)객실 내 금고
  12)유카타 실내 가운 및 슬리퍼
  13)다이슨 유피소닉(Dyson Supersonic™) 헤어드라이어
  
```

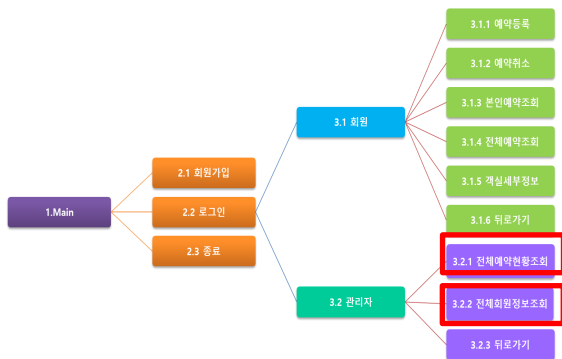
### ■ 3.1.5 - 3 : 스위트룸

```

=====
1.예약등록 2.예약취소 3.본인예약조회
4.전체예약조회 5.객실세부정보 6.뒤로가기
=====
>>> 선택 :
  
```

### 3. Console 화면구현

#### 3-6) 예약 조회 (관리자)



#### ■ 3.2.1 관리자 – 전체예약현황조회 ( ex : 예약회원 1 : ezen, 예약회원2 : java)

```

=====
1.예약현황 2.회원정보 3.뒤로가기
=====
>>> 선택 : 1
|
=====
전체예약현황조회
=====
호텔          스위트룸      디럭스룸      스탠다드룸
=====
신라호텔      ezen              □              □
라마다호텔    □                □              □
그랜드호텔    □                □              java
=====
1.예약현황 2.회원정보 3.뒤로가기
>>> 선택 :
  
```

#### ■ 3.2.2 회원정보

```

=====
1.예약현황 2.회원정보 3.뒤로가기
=====
>>> 선택 : 2
|
===== 회원 정보 =====
회원ID      이름      연락처
=====
ID : ezen/  이름 : ezen/  연락처 : ezen
=====
ID : java/   이름 : java/   연락처 : java
=====
1.예약현황 2.회원정보 3.뒤로가기
>>> 선택 :
  
```



## 3. 평가 및 개선방향

### 3-1. 역할 및 소감

3 조	업무분담	느낀점
조장 : 장용범	<ul style="list-style-type: none"> <li>· 인터페이스 · 주요 메소드 설계</li> <li>· 스프레드 처리</li> </ul>	<ul style="list-style-type: none"> <li>- 설계기준을 Java 문법의 다양한 사용에 맞춰 코딩에 굳이 필요 없는 불필요한 문법을 사용하게 되고 코드가 상당히 복잡하게 됨</li> <li>- 또한 중간 오류 수정 시 문법의 다양성을 그대로 살리려다 보니 제한적 사항이 많았음.</li> </ul>
신우현	<ul style="list-style-type: none"> <li>· Main 클래스 취합 ·파일처리</li> <li>· 예외처리</li> </ul>	<ul style="list-style-type: none"> <li>- 설계기준이 정해져 있어 배운 것들을 토대로 만들려고 한 결과 사용하지 않아도 되는 코드들을 사용하게 되어 코드가 많이 복잡하고 길어짐.</li> <li>- 현재는 배열로 코드를 만들었지만 앞으로 배울 것들을 활용하여 더 간결하고 효과적인 코드를 만들고 싶음.</li> </ul>
김태호	<ul style="list-style-type: none"> <li>· 메인메뉴 작성 · 취합보조 · 파일처리</li> <li>· 발표 및 자료 작성</li> </ul>	<ul style="list-style-type: none"> <li>- 문법을 실질적으로 코드를 작성하는데 어려움이 있었지만, 팀원들의 도움으로 습득하는데 좋은 기회가 된 것 같습니다.</li> <li>- 상황에 맞게 수정하는 과정이 쉽지 않았고, 처음 구상할 시에 기준설계가 비중이 크고 중요하다는 것을 한번 더 실감 할 수 있었습니다.</li> </ul>

### 3. 평가 및 개선방향

#### 3-2. 개선방향

현행 (문제점)	개선방향
<ul style="list-style-type: none"> <li>예외사항에 대한 메시지를 가독성을 위해 <code>System.err.println()</code> 을 사용. 이로 인해 출력 순서가 뒤바뀔 가능성이 있음.</li> </ul>	<ul style="list-style-type: none"> <li>멀티스레드를 이용하여 출력순서 조절 (1차 보완하였으나 여전히 스레드 처리속도가 빠름)</li> </ul>
<ul style="list-style-type: none"> <li>현재는 배열로 예약리스트를 사용하고 있지만 <u>관리가 용이하지 못하다고</u> 생각됨.</li> </ul>	<ul style="list-style-type: none"> <li>추후 ArrayList를 사용하여 관리를 더 편리하게 할 수 있도록 할 예정.</li> </ul>
<ul style="list-style-type: none"> <li>현재 멀티스레드를 통해 실시간으로 예약리스트 (배열 사용중) 업데이트를 하고 있습니다.</li> <li>저희 팀이 배열을 사용한 이유는 호텔의 수와 방 수가 정해져 있었기 때문에 배열이 메모리할당을 효율적으로 사용할 수 있을 것으로 생각하고 만들었습니다</li> </ul>	<ul style="list-style-type: none"> <li>호텔 수와 방 수가 적어 멀티스레드를 사용하면 멀티스레드 처리 속도가 더 빨라 메모리를 계속 잡아 먹게 되고 예약리스트 (배열)를 한번씩 읽는 속도가 너무 빨라 동기화를 시켰음에도 처리속도를 잡지 못하고 있습니다. 이 문제는 추후 호텔 수, 방의 수를 증가시키면 해결 할 수 있을 것으로 사료됩니다.</li> </ul>
<ul style="list-style-type: none"> <li>정해진 조건 (interface 및 extends) 를 활용하려고 먼저 선정하고 설계를 진행 하다보니, 간편화 될 수 있는 설계가 다소 코딩도 늘어진 것 같습니다.</li> <li>추후 호텔 및 방 조건등의 수정사항 시 관리의 불편.</li> </ul>	<ul style="list-style-type: none"> <li>객체를 ArrayList (호텔/룸)하여 묶어줌으로써 보다 더 관리하기 용이한 코드설계가 필요함</li> </ul>

감사합니다

