

# 스터디룸 예약 프로그램

## 2차 검사 보고서

### A04 팀

201910184 김민기

201911286 최민호

201911294 홍승택

202111294 박성준

202111309 손 본

202111332 유의진

해당 문서에서 입력 또는 출력 예시에 나오는  
“는 실제로 입력하지 않아야 하고, 출력도 되지 않습니다.

## 7. 명령창 이동 및 공통 처리

- 목표: 어느 명령창에서든 아무 문자열 입력 없이 엔터(개행)를 입력하면 명령어를 다시 입력 받음

- ID: 7-1
- 입력: ""
- 예상 결과: 명령어를 다시 입력 받음
- 실제 결과: (상동)
- 화면:

```
user >  
  
user >
```

- 목표: 어느 명령창에서든 공백 1개 이상으로만 이루어진 문자열을 입력하면, 잘못된 명령어로 처리

- ID: 7-2
- 입력: " ", " ", " "
- 예상 결과: "명령어 은(는) 존재하지 않는 명령어입니다."
- 실제 결과: (상동)
- 화면:

```
user >   
[오류] 21WrongCommandException 예외가 발생하였습니다.  
명령어 은(는) 존재하지 않는 명령어입니다.  
도움말을 출력합니다.  
[생략]
```

- 목표: back, logout을 제외한 명령창 이동 명령어를 입력할 경우, 대소문자 구분없이 동작함

- ID: 7-3
- 입력: "book", "Book", "bOOk", "search", "Search"
- 예상 결과: 모두 정상적으로 명령창 이동
- 실제 결과: (상동)
- 화면:

```
user > Book  
  
user/Book >
```

- 목표: 명령창 이동 명령어는 인자가 반드시 0개 이어야 함

- ID: 7-4
- 입력: "book 1", "Book "

- 예상 결과: “[오류] 명령어가 가질 수 있는 인자의 개수와 다릅니다.”
- 실제 결과: (상동)
- 화면:

```
user > book 1
[오류] 25WrongNumArgumentException 예외가 발생하였습니다.
명령어 book(이)가 가질 수 있는 인자의 개수와 다릅니다.
```

## 9. 공통 명령

### 9.1 뒤로 가기: back

- 목표: 대소문자가 혼합된 명령어 입력에 대한 반응 확인 (2절 용어의 명령어)

- ID: 9.1-1
- 입력: “BaCk”
- 예상 결과: 'back' 명령어를 입력한 것과 동일하게 작동합니다.
- 실제 결과: (상동)
- 화면:

```
user/Book > BaCk

user >
```

- 목표: 명령어 앞 공백이 1개 이상인 경우에 대한 반응 확인

- ID: 9.1-2
- 입력: “\_back”
- 예상 결과: “[오류] 명령어 은(는) 존재하지 않는 명령어입니다. 도움말을 출력합니다.”
- 실제 결과: (상동)
- 화면:

```
user/Book > back
[오류] class WrongCommandException 예외가 발생하였습니다.
명령어 은(는) 존재하지 않는 명령어입니다.
도움말을 출력합니다.

-----
명령어   |      기능
-----
help     |  도움말 출력
back     |  이전 명령창으로 되돌아감
book     |  예약하기
list     |  스터디룸 예약 현황 출력

user/Book >
```

■ 목표: 메인 명령창에서 뒤로가기 명령에 대한 반응 확인

- ID: 9.1-3
- 입력: "back"
- 예상 결과: 이전 명령창이 없기 때문에 동작하지 않습니다.  
"[오류] 현재 명령창 Main에서 back은(는) 사용할 수 없는 명령어입니다. 도움말을 출력합니다." 를 출력
- 실제 결과: (상동)
- 화면:

```
Main > back
[오류] class UnableCommandException 예외가 발생하였습니다.
현재 명령창 Main에서 back은(는) 사용할 수 없는 명령어입니다.
도움말을 출력합니다.
-----
명령어   |      기능
-----
help     |  도움말 출력
signup   |  회원가입
signin   |  로그인
exit     |  프로그램 종료

Main >
```

■ 목표: 사용자 명령창에서 뒤로가기 명령에 대한 반응 확인

- ID: 9.1-4
- 입력: "back"
- 예상 결과: logout 명령어를 사용해야 메인명령창으로 돌아갈 수 있기 때문에 동작하지 않습니다.  
"[오류] 현재 명령창 User에서 back은(는) 사용할 수 없는 명령어입니다. 도움말을 출력합니다." 를 출력
- 실제 결과: (상동)
- 화면:

```
user > back
[오류] class UnableCommandException 예외가 발생하였습니다.
현재 명령창 User에서 back은(는) 사용할 수 없는 명령어입니다.
도움말을 출력합니다.
-----
명령어   |      기능
-----
help     |   도움말 출력
logout   |   로그아웃 후 메인 명령창으로 이동
book     |   예약 명령창으로 이동
search   |   예약 확인 명령창으로 이동

user >
```

■ 목표: 사용자 명령창 내부 메뉴에서 뒤로가기 명령에 대한 반응 확인

- ID: 9.1-5
- 입력: "back"
- 예상 결과: 이전 명령창인 사용자 명령창으로 돌아갑니다.
- 실제 결과: (상동)
- 화면:

```
user/Search > back

user >
```

■ 목표: 관리자 명령창에서 뒤로가기 명령에 대한 반응 확인

- ID: 9.1-6
- 입력: "back"
- 예상 결과: logout 명령어를 사용해야 메인명령창으로 돌아갈 수 있기 때문에 동작하지 않습니다.  
"[오류] 현재 명령창 Admin에서 back은(는) 사용할 수 없는 명령어입니다. 도움말을 출력합니다." 를 출력
- 실제 결과: (상동)
- 화면:

```
Admin > back
[오류] class UnableCommandException 예외가 발생하였습니다.
현재 명령창 Admin에서 back은(는) 사용할 수 없는 명령어입니다.
도움말을 출력합니다.

-----
명령어 | 기능
-----
help   | 도움말 출력
logout | 로그아웃 후 메인 명령창으로 이동
search | 회원 검색 명령창으로 이동

Admin >
```

■ 목표: 관리자 명령창 내부 메뉴에서 뒤로가기 명령에 대한 반응 확인

- ID: 9.1-7
- 입력: "back"
- 예상 결과: 이전 명령창인 관리자 명령창으로 돌아갑니다.
- 실제 결과: (상동)
- 화면:

```
Admin/Search > back

Admin >
```

## 9.2 도움말: help

■ 목표: 인자 없이, 올바른 명령어에 대한 반응 확인 (9.2절 정상 동작 1)

- ID: 9.2-1
- 입력: "help"
- 예상 결과: 도움말을 출력합니다.
- 실제 결과: (상동)
- 화면:

```
Main > help
-----
명령어 | 기능
-----
help   | 도움말 출력
signup | 회원가입
signin | 로그인
exit   | 프로그램 종료

Main >
```

■ 목표: 인자 1개를 가지는 명령어에 대한 반응 확인 (9.2절 정상 동작 2)

- ID: 9.2-2
- 입력: "help\_signup"
- 예상 결과: 입력한 인자(명령어)의 설명과 입력한 인자(명령어)의 가능한 인자의 형태와 각 인자의 규칙을 출력합니다.
- 실제 결과: (상동)
- 화면:

```
Main > help signup
명령어

      signup : 도움말을 출력합니다.
가능한 인자의 형태
      signup [인자 1] [인자 2]
각 인자의 규칙
      [인자 1] : 사용자 이름을 입력합니다.
      [인자 2] : 사용자 전화번호를 입력합니다.

Main >
```

■ 목표: 대소문자가 혼합된 명령어 입력에 대한 반응 확인 (2절 용어의 명령어)

- ID: 9.2-3
- 입력: "HeLP"
- 예상 결과: "help" 명령어를 입력한 것과 동일하게 작동합니다.
- 실제 결과: (상동)
- 화면:

```
Main > HeLP
-----
명령어 | 기능
-----
help   | 도움말 출력
signup | 회원가입
signin | 로그인
exit   | 프로그램 종료

Main >
```

■ 목표: 명령어 앞 공백이 1개 이상인 경우에 대한 반응 확인

- ID: 9.2-4
- 입력: "\_help"
- 예상 결과: "[오류] 명령어 은(는) 존재하지 않는 명령어입니다. 도움말을 출력합니다."
- 실제 결과: (상동)

- 화면:

```
Main > help
[오류] class WrongCommandException 예외가 발생하였습니다.
명령어 은(는) 존재하지 않는 명령어입니다.
도움말을 출력합니다.
```

- 목표: 인자 없이, 명령어 뒤 공백이 1개 이상인 경우에 대한 반응 확인

- ID: 9.2-5
- 입력: "help\_"
- 예상 결과: "[오류] 현재 명령창 command에서 은(는) 사용할 수 없는 명령어입니다. 도움말을 출력합니다."
- 실제 결과: (상동)
- 화면:

```
Main > help
[오류] class UnableCommandException 예외가 발생하였습니다.
현재 명령창 command에서 은(는) 사용할 수 없는 명령어입니다.
도움말을 출력합니다.

-----
명령어 |      기능
-----
help   |  도움말 출력
signup |  회원가입
signin |  로그인
exit   |  프로그램 종료
```

- 목표: 명령어 마지막 인자 뒤 공백이 1개 이상인 경우에 대한 반응 확인 (8.1.2절 인자 문법 오류)

- ID: 9.2-6
- 입력: "help\_signup\_"
- 예상 결과: "[오류] 명령어 help이(가) 가질 수 있는 인자의 개수와 다릅니다."
- 실제 결과: (상동)
- 화면:

```
Main > help signup
[오류] class WrongNumArgumentException 예외가 발생하였습니다.
명령어 help(이)가 가질 수 있는 인자의 개수와 다릅니다.
```



## 10. 메인 명령

### 10.1 회원가입

- 목표: 등록되지 않은 사용자에게 대한 정상 동작 확인(정상 동작 1)

- ID: 10.1-1
- 입력: "signup kang 010-8888-2222"
- 예상 결과: 회원가입이 정상적으로 됨
- 실제 결과: (상동)
- 화면:

```
Main > signup kang 010-8888-2222
회원가입이 완료되었습니다.
사용자 이름: kang
사용자 전화번호: 01088882222
```

- 목표: 이미 회원가입된 전화번호인 경우 동작 확인 (비정상 동작 1)

- ID: 10.1-2
- 입력: "signup kim 7777777"
- 예상 결과: 이미 회원가입된 전화번호라고 출력, 회원가입 안됨
- 실제 결과: (상동)
- 화면:

```
Main > signup kim 7777777
[오류] class WrongRuleArgumentException 예외가 발생하였습니다.
입력 인자 7777777(이)가 의미규칙을 만족하지 못합니다.
규칙 : 이미 회원가입이 되어있는 전화번호입니다.
```

- 목표: 이름에 대소문자가 섞여있는 경우 & 전화번호에 '-'가 있는 경우 동작 확인

- ID: 10.1-3
- 입력: "signup SeoUnG 010-5253-8250"
- 예상 결과: 이름은 전부 소문자로 변환하고 전화번호는 숫자만 인식해서 회원가입 진행
- 실제 결과: (상동)
- 화면:

```
Main > signup SeoUnG 010-5253-8250
회원가입이 완료되었습니다.
사용자 이름: seoung
사용자 전화번호: 01052538250
```

- 목표: 동일한 이름 다른 전화번호로 가입하는 경우 정상 동작 확인

- ID: 10.1-4

[A04] 2차 검사 보고서

- 입력: "signup choi 2244556688"
- 예상 결과: 정상적으로 회원가입 됨
- 실제 결과: (상동)
- 화면:

```
Main > signup choi 2244556688
회원가입이 완료되었습니다.
사용자 이름: choi
사용자 전화번호: 2244556688
```

■ 목표: 관리자와 동일한 전화번호로 가입하는 경우 동작 확인

- ID: 10.1-5
- 입력: "signup choi 01012345678"
- 예상 결과: 이미 회원가입된 전화번호라고 출력 & 회원가입 안됨
- 실제 결과: (상동)
- 화면:

```
Main > signup choi 01012345678
[오류] class WrongRuleArgumentException 예외가 발생하였습니다.
입력 인자 01012345678(이)가 의미규칙을 만족하지 못합니다.
규칙 : 이미 회원가입이 되어있는 전화번호입니다.
```

■ 목표: 전화번호와 이름의 위치를 바꿔서 입력하는 경우 오류 발생 확인

- ID: 10.1-6
- 입력: "signup 01012345678 choi"
- 예상 결과: 인자 규칙을 만족하지 않는다고 출력 & 회원가입 안됨
- 실제 결과: (상동)
- 화면:

```
Main > signup 01012345678 choi
[오류] class WrongCharArgumentException 예외가 발생하였습니다.
입력 인자 01012345678에 사용할 수 없는 문자 '0'(이)가 존재합니다.
```

■ 목표: 이름의 문법 규칙을 따르지 않는 경우 오류 발생 확인

- ID: 10.1-7
- 입력: "signup CCo!!mp 0105678975"
- 예상 결과: 사용할 수 없는 문자로 인한 오류 발생
- 실제 결과: (상동)
- 화면:

```
Main > signup CCo!!mp 0105678975
[오류] class WrongCharArgumentException 예외가 발생하였습니다.
입력 인자 CCo!!mp에 사용할 수 없는 문자 '!' (이)가 존재합니다.
```

■ 목표: 전화번호의 문법 규칙을 따르지 않는 경우 오류 발생 확인

- ID: 10.1-8
- 입력: "signup Comput 010----a--5345"
- 예상 결과: 사용할 수 없는 문자로 인한 오류 발생
- 실제 결과: (상동)
- 화면:

```
Main > signup Comput 010----a--5345
[오류] class WrongCharArgumentException 예외가 발생하였습니다.
입력 인자 010----a--5345에 사용할 수 없는 문자 'a' (이)가 존재합니다.
```

## 10.2 로그인

■ 목표: 입력한 계정이 사용자 계정인 경우에 대한 반응 확인 (정상 동작 1)

- ID: 10.2-1
- 입력: "signin choi 88888888"
- 예상 결과: 사용자로 로그인되었다고 출력, Main -> choi로 변경
- 실제 결과: (상동)
- 화면:

```
Main > signin choi 88888888
사용자로 로그인 되었습니다.
```

■ 목표: 입력한 계정이 관리자 계정인 경우에 대한 반응 확인 (정상 동작 2)

- ID: 10.2-2
- 입력: "signin admin 01012345678"
- 예상 결과: 관리자로 로그인 되었다고 출력, Main -> Admin으로 변경
- 실제 결과: (상동)
- 화면:

```
Main > signin admin 01012345678
관리자로 로그인 되었습니다.
```

■ 목표: 존재하지 않는 이용자를 입력한 경우 반응 확인 (비정상 동작 1)

- ID: 10.2-3

[A04] 2차 검사 보고서

- 입력: "signin kang 01011112222"
- 예상 결과: 존재하지 않는 회원이라고 출력
- 실제 결과: (상동)
- 화면:

```
Main > signin kang 01011112222
[오류] class WrongRuleArgumentException 예외가 발생하였습니다.
입력 인자 회원정보(이)가 의미규칙을 만족하지 못합니다.
규칙 : 존재하지 않는 회원입니다.
```

■ 목표: 이름의 문법 규칙을 따르지 않는 경우 (10.1 인자 문법 규칙 오류)

- ID: 10.2-4
- 입력: "signin KI-M 7777777"
- 예상 결과: 사용할 수 없는 문자로 인한 오류 발생
- 실제 결과: (상동)
- 화면:

```
Main > signin KI-M 7777777
[오류] class WrongCharArgumentException 예외가 발생하였습니다.
입력 인자 KI-M에 사용할 수 없는 문자 '-'(이)가 존재합니다.
```

■ 목표: 전화번호의 문법 규칙을 따르지 않는 경우 (10.1 인자 문법 규칙 오류)

- ID: 10.2-5
- 입력: "signin choi 8888-88/88"
- 예상 결과: 사용할 수 없는 문자로 인한 오류 발생
- 실제 결과: (상동)
- 화면:

```
Main > signin choi 8888-88/88
[오류] class WrongCharArgumentException 예외가 발생하였습니다.
입력 인자 8888-88/88에 사용할 수 없는 문자 '/'(이)가 존재합니다.
```

■ 목표: 전화번호의 문법 규칙을 따르지 않는 경우 (10.1 인자 의미 규칙 오류)

- ID: 10.2-6
- 입력: "signin choi 8888-----8888"
- 예상 결과: 전화번호 길이로 인한 오류 발생
- 실제 결과: (상동)
- 화면:

```
Main > signin choi 8888-----8888
[오류] class WrongLengthArgumentException 예외가 발생하였습니다.
입력 인자 8888-----가 허용 가능한 길이 범위 [7-20] (을)를 벗어났습니다.
```

■ 목표: 전화번호의 문법 규칙을 따르지 않는 경우 (10.1 인자 문법 규칙 오류)

- ID: 10.2-7
- 입력: "signin choi -88888888"
- 예상 결과: 맨 앞의 하이픈으로 인한 오류 발생
- 실제 결과: (상동)
- 화면:

```
Main > signin choi -88888888
[오류] class ArgumentException 예외가 발생하였습니다.
인자 -88888888(이)가 잘못되었습니다.
맨 앞이나 맨 뒤에 하이픈(-)이 존재합니다.
```

### 10.3 프로그램 종료: exit

■ 목표: 올바른 명령어에 대한 반응 확인(10.3절 정상 동작)

- ID: 10.3-1
- 입력: "exit"
- 예상 결과: "프로그램을 종료합니다."
- 실제 결과: (상동)
- 화면:

```
Main > exit
프로그램을 종료합니다.
```

■ 목표: 대소문자가 혼합된 명령어 입력에 대한 반응 확인 (2절 용어의 명령어)

- ID: 10.3-2
- 입력: "Exit"
- 예상 결과: "exit" 명령어를 입력한 것과 동일하게 작동
- 실제 결과: (상동)
- 화면:

```
Main > Exit
프로그램을 종료합니다.
```

■ 목표: 명령어 앞 공백이 1개 이상인 경우에 대한 반응 확인

## [A04] 2차 검사 보고서

- ID: 10.3-3
- 입력: “\_exit”
- 예상 결과: “[오류] 명령어 은(는) 존재하지 않는 명령어입니다. 도움말을 출력합니다.
- 실제 결과: (상동)
- 화면:

```
Main > exit
[오류] class WrongCommandException 예외가 발생하였습니다.
명령어 은(는) 존재하지 않는 명령어입니다.
도움말을 출력합니다.
[생략]
```

- 목표: 명령어 뒤 공백이 1개 이상인 경우에 대한 반응 확인

- ID: 10.3-4
- 입력: “exit\_”
- 예상 결과: “명령어 exit(이)가 가질 수 있는 인자의 개수와 다릅니다.”
- 실제 결과: (상동)
- 화면:

```
Main > exit
[오류] 25WrongNumArgumentException 예외가 발생하였습니다.
명령어 exit(이)가 가질 수 있는 인자의 개수와 다릅니다.
```

## 11. 사용자 명령

### 11.1 예약 하기

- 목표: 올바른 규칙을 모두 만족하는 명령어에 대한 동작 확인 (11.1절 정상 동작)

- ID: 11.1-1
- 입력: “book 221209 1 0900 1100 4”
- 예상 결과: “2022-12-09 1번 스터디룸 09:00 ~ 11:00 로 정상 예약되었습니다.”(정상적인 예약 확인메시지)
- 실제 결과: (상동)
- 화면:

```
User/Book > book 221209 1 0900 1100 4
2022-12-09 1번 스터디룸 09:00 ~ 11:00 로 정상 예약되었습니다.
```

- 목표: 선택한 스터디룸 방의 최대 예약 가능 인원 수보다 큰 수를 입력했을 경우에 대한 반응 확인 (11.1절 비정상 동작 2)

- ID: 11.1-2

[A04] 2차 검사 보고서

- 입력: "book 221209 1 0900 1100 50"
- 예상 결과: "[오류] 스터디룸의 최대 예약 가능 인원수를 초과하였습니다.."
- 실제 결과: (상동)
- 화면:

```
user/Book > book 221209 1 0900 1100 50
[오류] class WrongRuleArgumentException 예외가 발생하였습니다.
입력 인자 50(이)가 의미규칙을 만족하지 못합니다.
규칙 : 스터디룸의 최대 예약 가능 인원수를 초과하였습니다.
```

■ 목표: 인자의 개수가 문법에서 정의한 5개와 다를 때 동작 확인

- ID: 11.1-2-1
- 입력: "book 221024 2 0900"
- 예상 결과: 잘못된 인자 개수로 인한 오류 발생
- 실제 결과: (상동)
- 화면:

```
User/Book > book 221024 2 0900
[오류] class WrongNumArgumentException 예외가 발생하였습니다.
명령어 book(이)가 가질 수 있는 인자의 개수와 다릅니다.
```

■ 목표: 첫번째 인자 "날짜"에 대한 문법 오류:잘못된 구분자 사용 확인

- ID: 11.1-2-2-1
- 입력: "book 202-21/024 2 0900 1100 4"
- 예상 결과: 인자 '202-21/024'에 의한 문법 오류 발생
- 실제 결과: (상동)
- 화면:

```
User/Book > book 202-21/024 2 0900 1100 4
[오류] 17ArgumentException 예외가 발생하였습니다.
인자 202-21/024(이)가 잘못되었습니다.
(구분자 사용시) 연도(year)는 2개 또는 4개의 숫자로 표현되어야 합니다.
```

■ 목표: 첫번째 인자 "날짜"에 대한 문법 오류:인자 제한길이 초과 확인

- ID: 11.1-2-2-2
- 입력: "book 200000000000021024 2 0900 1100 4"
- 예상 결과: 인자 '200000000000021024'에 의한 문법 오류 발생
- 실제 결과: (상동)
- 화면:

```
User/Book > book 20000000000021024 2 0900 1100 4
```

[오류] class WrongLengthArgumentException 예외가 발생하였습니다.

입력 인자 2000000000가 허용 가능한 길이 범위 [6-10] (을)를 벗어났습니다.

■ 목표: 첫번째 인자 “날짜”에 대한 문법 오류:숫자가 아닌 다른 요소 사용 확인

- ID: 11.1-2-2-3
- 입력: “book year/mm/dd 2 0900 1100 4”
- 예상 결과: 인자 ‘year/mm/dd’에 의한 문법 오류 발생
- 실제 결과: (상동)
- 화면:

```
User/Book > book year/mm/dd 2 0900 1100 4
```

[오류] class WrongCharArgumentException 예외가 발생하였습니다.

입력 인자 year/mm/dd에 사용할 수 없는 문자 'y'(이)가 존재합니다.

■ 목표: 두번째 인자 “방번호”에 대한 문법 오류:인자 제한길이 초과 확인

- ID: 11.1-2-3-1
- 입력: “book 221024 22 0900 1100 4”
- 예상 결과: 인자 ‘22’ 에 의한 문법 오류 발생
- 실제 결과: (상동)
- 화면:

```
User/Book > book 221024 22 0900 1100 4
```

[오류] class WrongLengthArgumentException 예외가 발생하였습니다.

입력 인자 22가 허용 가능한 길이 범위 [1] (을)를 벗어났습니다.

■ 목표: 두번째 인자 “방번호”에 대한 문법 오류:숫자가 아닌 다른 요소 사용 확인

- ID: 11.1-2-3-2
- 입력: “book 221024 c 0900 1100 4”
- 예상 결과: 인자 ‘c’ 에 의한 문법 오류 발생
- 실제 결과: (상동)
- 화면:

```
User/Book > book 221024 c 0900 1100 4
```

[오류] class WrongCharArgumentException 예외가 발생하였습니다.

입력 인자 c에 사용할 수 없는 문자 'c'(이)가 존재합니다.



[A04] 2차 검사 보고서

■ 목표: 세번째/네번째 인자 “시간”에 대한 문법 오류:잘못된 구분자 사용 확인

- ID: 11.1-2-4-1
- 입력: “book 221024 2 9/0/0 1100 4”
- 예상 결과: 인자 '09/0/0'에 의한 문법 오류 발생
- 실제 결과: (상동)
- 화면:

```
User/Book> book 221024 2 9/0/0 1100 4
[오류] class ArgumentException 예외가 발생하였습니다.
인자 9/0/0(이)가 잘못되었습니다.
구분자 사용이 잘못되었습니다.
```

■ 목표: 세번째/네번째 인자 “시간”에 대한 문법 오류:인자 제한길이 초과 확인

- ID: 11.1-2-4-2
- 입력: “book 221024 2 00:000900 1100 4”
- 예상 결과: 인자 '00:000900'에 의한 문법 오류 발생
- 실제 결과: (상동)
- 화면:

```
User/Book > book 221024 2 00:000900 1100 4
[오류] class WrongLengthArgumentException 예외가 발생하였습니다.
입력 인자 00:000900가 허용 가능한 길이 범위 [1-5] (을)를 벗어났습니다.
```

■ 목표: 세번째/네번째 인자 “시간”에 대한 문법 오류:숫자가 아닌 다른 요소 사용 확인

- ID: 11.1-2-4-3
- 입력: “book 221024 2 0900 end\_t 4”
- 예상 결과: 인자 'end\_t'에 의한 문법 오류 발생
- 실제 결과: (상동)
- 화면:

```
User/Book> book 221024 2 0900 end_t 4
[오류] 26WrongCharArgumentException 예외가 발생하였습니다.
입력 인자 end_t에 사용할 수 없는 문자 'e'(이)가 존재합니다.
```

■ 목표: 다섯 번째 인자 “예약인원”에 대한 문법 오류:인자 제한길이 초과 확인

- ID: 11.1-2-5-1
- 입력: “book 221209 1 0900 1100 004”
- 예상 결과: 인자 '004'에 의한 문법 오류 발생
- 실제 결과: (상동)

[A04] 2차 검사 보고서

- 화면:

```
User/Book> book 221209 1 0900 1100 004
[오류] class WrongLengthArgumentException 예외가 발생하였습니다.
입력 인자 004가 허용 가능한 길이 범위 [1-2] (을)를 벗어났습니다.
```

- 목표: 다섯 번째 인자 “예약인원”에 대한 의미규칙 오류: “예약인원”은 0이 될 수 없다

- ID: 11.1-2-5-2
- 입력: “book 221209 1 0900 1100 0”
- 예상 결과: 인자 '0'에 의한 의미규칙 오류 발생
- 실제 결과: (상동)
- 화면:

```
User/Book> book 221209 1 0900 1100 0
[오류] class WrongRuleArgumentException 예외가 발생하였습니다.

입력 인자 0(이)가 의미규칙을 만족하지 못합니다.

규칙 : 예약 인원수는 0보다 커야합니다
```

- 목표: 자동시간 보정에 대한 확인

- ID: 11.1-2-4-4
- 입력: “book 221024 2 09:04 10:54 4”
- 예상 결과: 09:00 ~ 11:00 으로 정상적으로 예약완료
- 실제 결과: (상동)
- 화면:

```
User/Book > book 221024 2 09:04 10:54 4
2022-10-24 2번 스터디룸 09:00 ~ 11:00 로 정상 예약되었습니다.
```

- 목표: 예약 날짜가 그레고리력 상에서 실제로 존재하는 날짜인지에 대한 반응 확인

- ID: 11.1-3-1
- 입력: “book 221099 3 0900 1100 4”
- 예상 결과: 존재하지 않는 날짜에 대한 의미규칙 예외 발생
- 실제 결과: (상동)
- 화면:

```
User/Book > book 221099 3 0900 1100 4
[오류] 26WrongRuleArgumentException 예외가 발생하였습니다.
입력 인자 2022-10-99(이)가 의미규칙을 만족하지 못합니다.
규칙 : 존재하지 않는 일(day)입니다.
```

■ 목표: 예약 날짜가 지금으로부터 90일 이후인 입력에 대한 반응 확인

- ID: 11.1-3-2
- 입력: "book 230410 3 0900 1100 4"
- 예상 결과: 90일 이내에만 예약이 가능하다는 안내문과 함께 예외 발생
- 실제 결과: (상동)
- 화면:

```
User/Book > book 230410 3 0900 1100 4
[오류] class WrongRuleArgumentException 예외가 발생하였습니다.
입력 인자 2023-04-10(이)가 의미규칙을 만족하지 못합니다.
규칙 : 최대 90일 후까지 예약할 수 있습니다.
```

■ 목표: 예약 시간이 운영시간이 아닌 입력에 대한 반응 확인

- ID: 11.1-3-3
- 입력: "book 221024 3 0400 1100 4"
- 예상 결과: 운영시간안에만 예약이 가능하다는 안내문과 함께 예외 발생
- 실제 결과: (상동)
- 화면:

```
User/Book > book 221024 3 0400 1100 4
[오류] class WrongRuleArgumentException 예외가 발생하였습니다.
입력 인자 04:00(이)가 의미규칙을 만족하지 못합니다.
규칙 : 스터디룸은 09:00 ~ 20:00까지 운영합니다.
```

■ 목표: 현재시간보다 이전시간에 예약하는 경우에 대한 반응 확인

- ID: 11.1-3-4
- 입력: "book 220924 3 0900 1100 4"
- 예상 결과: "[오류] 입력 인자 "시간"이 의미규칙을 만족하지 못합니다."
- 실제 결과: (상동)
- 화면:

```
User/Book > book 220924 3 0900 1100 4
```

[오류] class WrongRuleArgumentException 예외가 발생하였습니다.  
입력 인자 2022-09-24(이)가 의미규칙을 만족하지 못합니다.  
규칙 : 입력 인자 “시간”이 의미규칙을 만족하지 못합니다.

■ 목표: 예약종료시간이 예약 시작시간보다 빠른 경우에 대한 반응 확인

- ID: 11.1-3-5
- 입력: “book 221024 3 1100 0900 4”
- 예상 결과: 예약 종료시간이 시작시간보다 빠를 수는 없다는 안내문과 함께 예외 발생
- 실제 결과: (상동)
- 화면:

```
User/Book > book 221024 3 1100 0900 4
```

[오류] class WrongRuleArgumentException 예외가 발생하였습니다.  
입력 인자 09:00(이)가 의미규칙을 만족하지 못합니다.  
규칙 : 예약 끝나는 시각은 시작 시각 이후여야 합니다.

■ 목표: 존재하지 않는 방번호(0번)에 대한 반응 확인

- ID: 11.1-3-6
- 입력: “book 221024 0 0900 1100 4”
- 예상 결과: 존재하지 않는 방번호라는 안내문과 함께 예외 발생
- 실제 결과: (상동)
- 화면:

```
User/Book > book 221024 0 0900 1100 4
```

[오류] 26WrongRuleArgumentException 예외가 발생하였습니다.  
입력 인자 0(이)가 의미규칙을 만족하지 못합니다.  
규칙 : 스터디룸 번호는 1부터 9까지 존재합니다.

■ 목표: 대소문자가 혼합된 명령어 입력에 대한 반응 확인 (2절 용어의 명령어)

- ID: 11.1-4-1
- 입력: “BoOK 221024 4 0900 1100 4”
- 예상 결과: “book” 명령어를 입력한 것과 동일하게 정상적으로 작동
- 실제 결과: (상동)
- 화면:

```
User/Book > BoOK 221024 4 0900 1100 4
```

2022-10-24 4번 스터디룸 09:00 ~ 11:00 로 정상 예약되었습니다.

■ 목표: 명령어 앞 공백이 1개 이상인 경우에 대한 반응 확인

- ID: 11.1-4-2
- 입력: “\_book 221024 4 0900 1100 4”
- 예상 결과: “[오류] 명령어 은(는) 존재하지 않는 명령어입니다. 도움말을 출력합니다.”
- 실제 결과: (상동)
- 화면:

```
User/Book > book 221024 0 0900 1100 4
[오류] class WrongCommandException 예외가 발생하였습니다.
명령어 은(는) 존재하지 않는 명령어입니다.
도움말을 출력합니다.
```

명령어	기능
help	도움말 출력
back	이전 명령창으로 되돌아감
book	예약하기
list	스터디룸 예약 현황 출력

■ 목표: 명령어 뒤, 인자 앞의 공백이 1개 이상인 경우에 대한 반응 확인

- ID: 11.1-4-3
- 입력: “book \_ \_ 221024 4 0900 1100 4”
- 예상 결과: “[오류] 명령어 help이(가) 가질 수 있는 인자의 개수와 다릅니다.”
- 실제 결과: (상동)
- 화면 :

```
User/Book > book 221024 0 0900 1100 4
[오류] class WrongNumArgumentException 예외가 발생하였습니다.
명령어 book(이)가 가질 수 있는 인자의 개수와 다릅니다.
```

■ 목표: 인자 사이의 공백이 1개 이상인 경우에 대한 반응 확인

- ID: 11.1-4-4
- 입력: “book 221024 \_ \_ 4 0900 1100 4”
- 예상 결과: “[오류] 명령어 book이(가) 가질 수 있는 인자의 개수와 다릅니다.”
- 실제 결과: (상동)
- 화면 :

```
User/Book > book 221024 4 0900 1100 4
[오류] class WrongNumArgumentException 예외가 발생하였습니다.
명령어 book(이)가 가질 수 있는 인자의 개수와 다릅니다.
```

## 11.2 예약 확인

### 11.2.1 스터디룸 예약 상태 확인

- 목표: 올바른 형식의 인자 4개를 가지는 명령어에 대한 반응 확인 (11.2.1절 정상 동작 1)

- ID: 11.2.1-1
- 입력: "list\_4\_2022-12-09"
- 예상 결과: 입력 받은 예약 인원을 고려한, 입력 날짜의 예약 결과 출력
- 실제 결과: (상동)
- 화면:

```
kim/Book > list 4 2022-12-09
2022-12-09 스터디룸 상태입니다.
스터디룸 09:00 09:30 10:00 10:30 11:00 11:30 [생략]
1(6인실) 가능 가능 가능 가능 가능 가능
[생략]
```

- 목표: 인자 없이, 인자가 1개일 경우, 올바른 명령어에 대한 반응 확인 (11.2.1절 정상 동작 2)

- ID: 11.2.1-2
- 입력: "list\_4"
- 예상 결과: 입력 받은 예약 인원을 고려한, 예약 당일의 예약 결과 출력
- 실제 결과: (상동)
- 화면:

```
kim/Book > list 4
2022-10-24 스터디룸 상태입니다.
스터디룸 09:00 09:30 10:00 10:30 11:00 11:30 [생략]
1(6인실) 가능 가능 가능 가능 가능 가능
[생략]
```

- 목표: 인자가 없는 명령어에 대한 반응 확인

- ID: 11.2.1-3
- 입력: "list"
- 예상 결과: [오류] 명령어 list(이)가 가질 수 있는 인자의 개수와 다릅니다.
- 실제 결과: (상동)
- 화면:

```
kim/Book > list
[오류] class WrongNumArgumentException 예외가 발생하였습니다.
명령어 list(이)가 가질 수 있는 인자의 개수와 다릅니다.
```

■ 목표: 대소문자가 혼합된 명령어 입력에 대한 반응 확인 (2절 용어의 명령어)

- ID: 11.2.1-4
- 입력: "LisT 4"
- 예상 결과: "list" 명령어를 입력한 것과 동일하게 작동
- 실제 결과: (상동)
- 화면:

```
kim/Book > LisT 4
2022-10-24 스터디룸 상태입니다.
스터디룸 09:00 09:30 10:00 10:30 11:00 11:30 [생략]
1(6인실) 가능 가능 가능 가능 가능 가능
[생략]
```

■ 목표: 명령어에 인자 23개 이상이 입력될 경우에 대한 반응 확인 (8.1.2절 인자 문법 오류)

- ID: 11.2.1-5
- 입력: "list\_4\_2022-10-30\_abcd"
- 예상 결과: "[오류] 명령어 list이(가) 가질 수 있는 인자의 개수와 다릅니다."
- 실제 결과: (상동)
- 화면:

```
kim/Book > list 4 2022-10-30 abcd
[오류] 25WrongNumArgumentException 예외가 발생하였습니다.
명령어 list(이)가 가질 수 있는 인자의 개수와 다릅니다.
```

■ 목표: 인자1에 대한 문법 오류:인자 제한길이 초과 확인

- ID: 11.2.1-6
- 입력: "list\_004\_221209"
- 예상 결과: "[오류] 입력 인자 004가 허용 가능한 길이 범위 [1-2](을)를 벗어났습니다."
- 실제 결과: (상동)
- 화면:

```
kim/Book > list 004 221209
[오류] class WrongLengthArgumentException 예외가 발생하였습니다.
입력 인자 004가 허용 가능한 길이 범위 [1-2] (을)를 벗어났습니다.
```

■ 목표: 인자 1개가 인자2가 낱자 형식에 맞지 않게 입력된 경우에 대한 반응 확인 (8.1.2절 인자 문법 오류)

- ID: 11.2.1-7
- 입력: "list\_4\_abcd12"

[A04] 2차 검사 보고서

- 예상 결과: “[오류] 입력 인자 abcd12에 사용할 수 없는 문자 ‘a’(이)가 존재합니다.”
- 실제 결과: (상동)
- 화면:

```
kim/Book > list 4 abcd12
[오류] 26WrongCharArgumentException 예외가 발생하였습니다.
입력 인자 abcd12에 사용할 수 없는 문자 'a'(이)가 존재합니다.
```

- 목표: 그레고리력으로 올바르지 않은 날짜가 1개의 인자 인자2로 입력된 경우에 대한 반응 확인 (8.1.3절 인자 의미 오류)

- ID: 11.2.1-8
- 입력: “list\_4\_2022-10-34”
- 예상 결과: “[오류] 입력 인자 2022-10-34(이)가 의미 규칙을 만족하지 못합니다.”
- 실제 결과: (상동)
- 화면:

```
kim/Book > list 4 2022-10-34
[오류] 26WrongRuleArgumentException 예외가 발생하였습니다.
입력 인자 2022-10-34(이)가 의미규칙을 만족하지 못합니다.
규칙 : 존재하지 않는 일(day)입니다.
```

- 목표: 예약일 기준 90일을 초과한 날짜가 1개의 인자로 입력된 경우에 대한 반응 확인 (11.2.1절 비정상 동작 1)

- ID: 11.2.1-9
- 입력: “list\_4\_2023-02-04”
- 예상 결과: “[오류] 최대 90일 후까지 확인할 수 있습니다.”
- 실제 결과: (상동)
- 화면:

```
kim/Book > list 4 2023-02-04
[오류] 26WrongRuleArgumentException 예외가 발생하였습니다.
입력 인자 2023-02-04(이)가 의미규칙을 만족하지 못합니다.
규칙 : 최대 90일 후까지 확인할 수 있습니다.
```

- 목표: 예약일 기준 과거 날짜가 1개의 인자로 입력된 경우에 대한 반응 확인 (11.2.1절 비정상 동작 2)

- ID: 11.2.1-10
- 입력: “list\_4\_2022-05-05”
- 예상 결과: “[오류] 과거 날짜는 조회할 수 없습니다.”
- 실제 결과: (상동)



- 화면 :

```
kim/Book > list 4 2022-05-05
[오류] 26WrongRuleArgumentException 예외가 발생하였습니다.
입력 인자 2022-05-05(이)가 의미규칙을 만족하지 못합니다.
규칙 : 과거 날짜는 조회할 수 없습니다.
```

- 목표: 명령어와 인자 사이의 공백이 2개 이상인 경우에 대한 반응 확인 (8.1.2절 인자 문법 오류)

- ID: 11.2.1-11
- 입력: "list\_ 4\_2022-10-30"
- 예상 결과: "[오류] 명령어 list이(가) 가질 수 있는 인자의 개수와 다릅니다."
- 실제 결과: (상동)
- 화면:

```
kim/Book > list 4 2022-10-30
[오류] 25WrongNumArgumentException 예외가 발생하였습니다.
명령어 list(이)가 가질 수 있는 인자의 개수와 다릅니다.
```

- 목표: 명령어 앞 또는 마지막 인자 뒤 공백이 1개 이상인 경우에 대한 반응 확인

- ID: 11.2.1-12
- 입력: "\_list\_ 4\_2022-10-30\_"
- 예상 결과: "[오류] 명령어 은(는) 존재하지 않는 명령어입니다. 도움말을 출력합니다."
- 실제 결과: (상동)
- 화면:

```
kim/Book > list 4 2022-10-30
[오류] 21WrongCommandException 예외가 발생하였습니다.
명령어 은(는) 존재하지 않는 명령어입니다.
도움말을 출력합니다.
[생략]
```

### 11.2.2 사용자 예약 내역 확인: check

- 목표: 올바른 명령어에 대한 반응 확인(11.2.2절 정상 동작)

- ID: 11.2-1
- 입력: "check"
- 예상 결과: 사용자의 예약된 스터디룸을 나열 우선 순위에 맞춰 나열합니다.
- 실제 결과: (상동)
- 화면:

kim/Search > check

예약번호	예약날짜	인원	스터디룸 번호	시작시간	종료시간
32	2022-10-24	4명	1번 스터디룸	16:00	17:00
26	2022-10-25	3명	1번 스터디룸	09:00	10:00
23	2022-10-27	3명	7번 스터디룸	13:00	17:00

[생략]

■ 목표: 대소문자가 혼합된 명령어 입력에 대한 반응 확인 (2절 용어의 명령어)

- ID: 11.2-2
- 입력: "CheCK"
- 예상 결과: "check" 명령어를 입력한 것과 동일하게 작동
- 실제 결과: (상동)
- 화면:

kim/Search > CheCK

예약번호	예약날짜	인원	스터디룸 번호	시작시간	종료시간
32	2022-10-24	4명	1번 스터디룸	16:00	17:00
26	2022-10-25	3명	1번 스터디룸	09:00	10:00
23	2022-10-27	3명	7번 스터디룸	13:00	17:00

[생략]

■ 목표: 명령어 앞 공백이 1개 이상인 경우에 대한 반응 확인

- ID: 11.2-3
- 입력: "\_check"
- 예상 결과: "[오류] 명령어 은(는) 존재하지 않는 명령어입니다. 도움말을 출력합니다."
- 실제 결과: (상동)
- 화면:

kim/Search > check

[오류] class WrongCommandException 예외가 발생하였습니다.

명령어 은(는) 존재하지 않는 명령어입니다.

도움말을 출력합니다.

[생략]

■ 목표: 명령어 뒤 인자가 1개 이상인 경우에 대한 반응 확인

- ID: 11.2-4
- 입력: "check\_check"
- 예상 결과: [오류] 명령어 check(이)가 가질 수 있는 인자의 개수와 다릅니다.
- 실제 결과: (상동)

- 화면 :

```
kim/Search > check check
```

[오류] class WrongNumArgumentException 예외가 발생하였습니다.  
명령어 check(이)가 가질 수 있는 인자의 개수와 다릅니다.

### 11.3 로그아웃: logout

- 목표: 올바른 명령어에 대한 반응 확인(11.3절 정상 동작)

- ID: 11.3-1
- 입력: "logout"
- 예상 결과: 메인 명령창으로 이동합니다.
- 실제 결과: (상동)
- 화면:

```
user > logout
```

```
Main >
```

- 목표: 대소문자가 혼합된 명령어 입력에 대한 반응 확인 (2절 용어의 명령어)

- ID: 11.3-2
- 입력: "LogOuT"
- 예상 결과: 'logout' 명령어를 입력한 것과 동일하게 작동
- 실제 결과: (상동)
- 화면:

```
user > LogOuT
```

```
Main >
```

- 목표: 명령어 앞 공백이 1개 이상인 경우에 대한 반응 확인

- ID: 11.3-3
- 입력: " \_logout"
- 예상 결과: "[오류] 명령어 은(는) 존재하지 않는 명령어입니다. 도움말을 출력합니다."
- 실제 결과: (상동)
- 화면:

```
user > logout
[오류] class WrongCommandException 예외가 발생하였습니다.
명령어 은(는) 존재하지 않는 명령어입니다.
도움말을 출력합니다.
-----
명령어   |   기능
-----
help     |   도움말 출력
logout   |   로그아웃 후 메인 명령창으로 이동
book     |   예약 명령창으로 이동
search   |   예약 확인 명령창으로 이동

user >
```

## 12. 관리자 명령

### 12.1 회원 정보 검색

- 목표: 인자가 없는 경우에 대한 반응 확인 (정상 동작 1)

- ID: 12.1-1
- 입력: "ask"
- 예상 결과: 모든 회원 리스트를 출력합니다.
- 실제 결과: (상동)
- 화면:

```
Admin/Search > ask
이름   전화번호
minho  01026350303
minho  01055884412
minho  01023213123
tester 112233445
testertwo      11223245
```

- 목표: 인자가 1개이고, 인자가 이름인 경우에 대한 반응 확인 (정상 동작 2-1)

- ID: 12.1-2
- 입력: "ask minho"
- 예상 결과: 해당 이름과 일치하는 사용자의 목록을 출력합니다.
- 실제 결과: (상동)
- 화면:

Admin/Search > ask minho

이름	전화번호
minho	01026350303
minho	01055884412
minho	01023213123

■ 목표: 인자가 1개이고, 인자가 이름인 경우에 대한 반응 확인 (정상 동작 2-2)

- ID: 12.1-3
- 입력: "ask hhh"
- 예상 결과: 해당 이름과 일치하는 사용자의 목록을 출력합니다. 일치하지 않을 경우 빈 리스트를 출력합니다.
- 실제 결과: (상동)
- 화면:

Admin/Search > ask hhh

이름	전화번호
----	------

■ 목표: 인자가 1개이고, 인자가 전화번호인 경우에 대한 반응 확인 (정상 동작 3-1)

- ID: 12.1-4
- 입력: "ask 01026350303"
- 예상 결과: 해당 사용자의 이름, 전화번호와 예약 현황을 출력합니다.
- 실제 결과: (상동)
- 화면:

Admin/Search > ask 01026350303

이름	전화번호
minho	01026350303

예약 현황

2022-12-12 | 10:00 ~ 18:00 |     스터디 룸 번호 : 1     |     예약번호 : 5

과거 이용 내역

■ 목표: 인자가 1개이고, 인자가 전화번호인 경우에 대한 반응 확인 (정상 동작 3-2)

- ID: 12.1-5
- 입력: "ask 123456789"
- 예상 결과: 해당 사용자의 이름, 전화번호와 예약 현황을 출력합니다. 존재하지 않는 전화번호일 경우 해당 전화번호와 일치하는 사용자가 없다는 안내문구를 출력합니다.
- 실제 결과: (상동)

- 화면:

```
Admin/Search > ask 123456789
[오류] class WrongRuleArgumentException 예외가 발생하였습니다.
입력 인자 전화번호(이)가 의미규칙을 만족하지 못합니다.
규칙 : 해당 전화번호와 일치하는 사용자를 찾을 수 없습니다.
```

- 목표: 인자가 1개이고, 이름과 전화번호를 공백없이 입력한 경우에 대한 반응 확인 (정상 동작 4)

- ID: 12.1-6
- 입력: "ask minho01026350303"
- 예상 결과: 인자에 맞는 데이터가 있는 경우 예약현황과 과거 이용 내역을 출력합니다.
- 실제 결과: (상동)
- 화면:

```
Admin/Search > ask minho01026350303
이름                전화번호
minho                01026350303

예약 현황

2022-12-12 | 10:00 ~ 18:00 |   스터디 룸 번호 : 1   |   예약번호 : 5

과거 이용 내역
```

- 목표: 인자가 2개이고, 이름과 전화번호를 입력한 경우에 대한 반응 확인 (정상 동작 5)

- ID: 12.1-7
- 입력: "ask minho 01026350303"
- 예상 결과: 인자 두개를 만족하는 유일한 데이터의 예약현황과 과거 이용 내역을 출력합니다.
- 실제 결과: (상동)
- 화면:

Admin/Search > ask minho 01026350303

이름	전화번호
minho	01026350303

예약 현황

2022-12-12 | 10:00 ~ 18:00 |     스터디 룸 번호 : 1     |     예약번호 : 5

과거 이용 내역

- 목표: 인자가 1개이고, 이름과 전화번호를 공백없이 혼합하여 입력한 경우에 대한 반응 확인 (비정상 동작 1)

- ID: 12.1-8
- 입력: "ask min0102635ho0303"
- 예상 결과: 숫자와 영문자를 문자열 2개로 나눌 수 없기때문에 에러를 발생시킵니다.
- 실제 결과: (상동)
- 화면:

Admin/Search > ask min0102635ho0303  
[오류] class ArgumentException 예외가 발생하였습니다.  
인자 min0102635ho0303(이)가 잘못되었습니다.

- 목표: 인자가 1개이고, 전화번호를 입력한 경우에 일치하는 사용자가 없는 경우에 대한 반응 확인 (비정상 동작 2)

- ID: 12.1-9
- 입력: "ask 0101122556"
- 예상 결과: 일치하는 사용자가 없을 경우 찾을 수 없다는 안내문구를 출력합니다.
- 실제 결과: (상동)
- 화면:

Admin/Search > ask 0101122556  
[오류] class WrongRuleArgumentException 예외가 발생하였습니다.  
입력 인자 전화번호(이)가 의미규칙을 만족하지 못합니다.

- 목표: 인자가 1개이고, 이름 전화번호를 붙여서 입력한 경우에 일치하는 사용자가 없는 경우에 대한 반응 확인 (비정상 동작 2)

- ID: 12.1-10
- 입력: "ask aaadd01026350303"
- 예상 결과: 전화번호와 사용자가 서로 일치하지 않는다는 안내문구를 출력합니다.
- 실제 결과: (상동)
- 화면:

```
Admin/Search > ask asdf01026350303
[오류] class WrongRuleArgumentException 예외가 발생하였습니다.
입력 인자 전화번호와 이름(이)가 의미규칙을 만족하지 못합니다.
규칙 : 해당 전화번호와 일치하는 사용자 이름이 없습니다.
```

- 목표: 인자가 1개이고, 전화번호 이름을 붙여서 입력한 경우에 대한 반응 확인
  - ID: 12.1-11
  - 입력: "ask 01026350303minho"
  - 예상결과: 전화번호 이름은 지원하지 않는 형식이기 때문에 예외가 발생한다는 안내문구를 출력합니다.
  - 실제 결과: (상동)
  - 화면:

```
Admin/Search > ask 01026350303minho
[오류] class ArgumentException 예외가 발생하였습니다.
인자 01026350303minho(이)가 잘못되었습니다.
```

## 12.2 로그아웃: logout

- 목표: 올바른 명령어에 대한 반응 확인(12.2절 정상 동작)
  - ID: 12.2-1
  - 입력: "logout"
  - 예상 결과: 메인 명령창으로 이동합니다.
  - 실제 결과: (상동)
  - 화면:

```
Admin > logout

Main >
```

- 목표: 대소문자가 혼합된 명령어 입력에 대한 반응 확인 (2절 용어의 명령어)
  - ID: 12.2-2
  - 입력: "LogOuT"
  - 예상 결과: 'logout' 명령어를 입력한 것과 동일하게 작동
  - 실제 결과: (상동)
  - 화면:

```
Admin > LogOuT

Main >
```



■ 목표: 명령어 앞 공백이 1개 이상인 경우에 대한 반응 확인

- ID: 12.2-3
- 입력: “\_logout”
- 예상 결과: “[오류] 명령어 은(는) 존재하지 않는 명령어입니다. 도움말을 출력합니다.”
- 실제 결과: (상동)
- 화면:

```
Admin > logout
[오류] class WrongCommandException 예외가 발생하였습니다.
명령어 은(는) 존재하지 않는 명령어입니다.
도움말을 출력합니다.
-----
명령어   |      기능
-----
help     |   도움말 출력
logout   |   로그아웃 후 메인 명령창으로 이동
search   |   회원 검색 명령창으로 이동

Admin >
```

## 13. 예약 최적화

### 13.1 스터디룸 자동 변경 알고리즘 - 여러 시나리오에 대한 최적화 반응 확인

■ 목표: 스터디룸 예약 시나리오에 대한 최적화 반응 확인 1

- ID: 13.1-1
- 전제: 1번 방(6인실), 2번 방(4인실), 3번 방(4인실)을 제외한 모든 방은 1인실
- 기존 예약: 1번 방(6인실)=  
2번 방(4인실)= a팀(4명, 12시~13시) / b팀(2명, 15시~17시)  
3번 방(4인실)=
- 새로운 예약: user(1번 방, 6명, 10시~18시)
- 예상 결과: 1번 방(6인실)= user(6명, 10시~18시)  
2번 방(4인실)= a팀(4명, 12시~13시) / b팀(2명, 15시~17시)  
3번 방(4인실)=
- 실제 결과: (상동)
- 화면:

aa/Search > check					
예약번호	예약날짜	인원	스터디룸 번호	시작시간	종료시간
1	2022-12-11	4명	2번 스터디룸	12:00	13:00
bb/Search > check					
예약번호	예약날짜	인원	스터디룸 번호	시작시간	종료시간
2	2022-12-11	2명	2번 스터디룸	15:00	17:00

user/Book > book 221211 1 1000 1800 6					
2022-12-11 1번 스터디룸 10:00 ~ 18:00 로 정상 예약되었습니다.					
user/Search > check					
예약번호	예약날짜	인원	스터디룸 번호	시작시간	종료시간
4	2022-12-11	6명	1번 스터디룸	10:00	18:00

aa/Search > check					
예약번호	예약날짜	인원	스터디룸 번호	시작시간	종료시간
1	2022-12-11	4명	2번 스터디룸	12:00	13:00
bb/Search > check					
예약번호	예약날짜	인원	스터디룸 번호	시작시간	종료시간
2	2022-12-11	2명	2번 스터디룸	15:00	17:00

- 목표: 스터디룸 예약 시나리오에 대한 최적화 반응 확인 2
  - ID: 13.1-2
  - 전제: 1번 방(6인실), 2번 방(4인실), 3번 방(4인실)을 제외한 모든 방은 1인실
  - 기존 예약: 1번 방(6인실)= a팀(2명, 9시~19시)
    - 2번 방(4인실)= b팀(4명, 12시~13시) / c팀(2명, 15시~17시)
    - 3번 방(4인실)=
  - 새로운 예약: user(1번 방, 6명, 10시~18시)
  - 예상 결과: 1번 방(6인실)= a팀(2명, 9시~10시) / user(6명, 10시~18시) / a팀(2명, 18시~19시)
    - 2번 방(4인실)= a팀(2명, 10시~12시) / b팀(4명, 12시~13시) / a팀(2명, 13시~15시) / b팀(2명, 15시~17시) / a팀(2명, 17시~18시)
    - 3번 방(4인실)= a팀(2명, 12시~13시, 15시~17시)
  - 실제 결과: (상동)
  - 화면:

aa/Search > check

예약번호	예약날짜	인원	스터디룸 번호	시작시간	종료시간
1	2022-12-11	2명	1번 스터디룸	09:00	19:00

bb/Search > check

예약번호	예약날짜	인원	스터디룸 번호	시작시간	종료시간
2	2022-12-11	4명	2번 스터디룸	12:00	13:00

cc/Search > check

예약번호	예약날짜	인원	스터디룸 번호	시작시간	종료시간
3	2022-12-11	2명	2번 스터디룸	15:00	17:00

user/Book > book 221211 1 1000 1800 6

2022-12-11 1번 스터디룸 10:00 ~ 18:00 로 정상 예약되었습니다.

user/Search > check

예약번호	예약날짜	인원	스터디룸 번호	시작시간	종료시간
4	2022-12-11	6명	1번 스터디룸	10:00	18:00

aa/Search > check

예약번호	예약날짜	인원	스터디룸 번호	시작시간	종료시간
1	2022-12-11	2명	1번 스터디룸	09:00	10:00
	2022-12-11	2명	2번 스터디룸	10:00	12:00
	2022-12-11	2명	3번 스터디룸	12:00	13:00
	2022-12-11	2명	2번 스터디룸	13:00	15:00
	2022-12-11	2명	3번 스터디룸	15:00	17:00
	2022-12-11	2명	2번 스터디룸	17:00	18:00
	2022-12-11	2명	1번 스터디룸	18:00	19:00

bb/Search > check

예약번호	예약날짜	인원	스터디룸 번호	시작시간	종료시간
2	2022-12-11	4명	2번 스터디룸	12:00	13:00

cc/Search > check

예약번호	예약날짜	인원	스터디룸 번호	시작시간	종료시간
3	2022-12-11	2명	2번 스터디룸	15:00	17:00

■ 목표: 스터디룸 예약 시나리오에 대한 최적화 반응 확인 3

- ID: 13.1-3
- 전제: 1번 방(6인실), 2번 방(4인실), 3번 방(4인실)을 제외한 모든 방은 1인실
- 기존 예약: 1번 방(6인실)=a팀(2명, 9시~19시)  
2번 방(4인실)=

[A04] 2차 검사 보고서

3번 방(4인실)= b팀(4명, 12시~13시) / c팀(2명, 15시~17시)

- 새로운 예약: user(1번 방, 6명, 10시~18시)
- 예상 결과: 1번 방(6인실)= a팀(2명, 9시~10시) / user(6명, 10시~18시) / a팀(2명, 18시~19시)

2번 방(4인실)= a팀(2명, 10시~18시)

3번 방(4인실)= b팀(4명, 12시~13시) / c팀(2명, 15시~17시)

- 실제 결과: (상동)
- 화면:

aa/Search > check

예약번호	예약날짜	인원	스터디룸 번호	시작시간	종료시간
1	2022-12-13	2명	1번 스테디룸	09:00	19:00

bb/Search > check

예약번호	예약날짜	인원	스터디룸 번호	시작시간	종료시간
2	2022-12-13	4명	3번 스테디룸	12:00	13:00

cc/Search > check

예약번호	예약날짜	인원	스터디룸 번호	시작시간	종료시간
3	2022-12-13	2명	3번 스테디룸	15:00	17:00

user/Book > book 221213 1 1000 1800 6

2022-12-13 1번 스테디룸 10:00 ~ 18:00 로 정상 예약되었습니다.

user/Search > check

예약번호	예약날짜	인원	스터디룸 번호	시작시간	종료시간
4	2022-12-13	6명	1번 스테디룸	10:00	18:00

aa/Search > check

예약번호	예약날짜	인원	스터디룸 번호	시작시간	종료시간
1	2022-12-13	2명	1번 스테디룸	09:00	10:00
	2022-12-13	2명	2번 스테디룸	10:00	18:00
	2022-12-13	2명	1번 스테디룸	18:00	19:00

bb/Search > check

예약번호	예약날짜	인원	스터디룸 번호	시작시간	종료시간
2	2022-12-13	4명	3번 스테디룸	12:00	13:00

cc/Search > check

예약번호	예약날짜	인원	스터디룸 번호	시작시간	종료시간
3	2022-12-13	2명	3번 스테디룸	15:00	17:00

[A04] 2차 검사 보고서

■ 목표: 스터디룸 예약 시나리오에 대한 최적화 반응 확인 4

- ID: 13.1-4
- 전제: 1번 방(6인실), 2번 방(4인실), 3번 방(2인실)을 제외한 모든 방은 1인실
- 기존 예약: 1번 방(6인실)= a팀(3명, 9시~19시)  
2번 방(4인실)= b팀(1명, 12시~13시) / c팀(2명, 15시~17시)  
3번 방(2인실)=
- 새로운 예약: user(1번 방, 6명, 10시~18시)
- 예상 결과: 1번 방(6인실)= a팀(3명, 9시~10시) / user(6명, 10시~18시) / a팀(3명, 18시~19시)  
2번 방(4인실)= a팀(3명, 10시~18시)  
3번 방(2인실)= b팀(1명, 12시~13시) / c팀(2명, 15시~17시)
- 실제 결과: (상동)
- 화면:

aa/Search > check

예약번호	예약날짜	인원	스터디룸 번호	시작시간	종료시간
1	2022-12-13	3명	1번 스터디룸	09:00	19:00

bb/Search > check

예약번호	예약날짜	인원	스터디룸 번호	시작시간	종료시간
2	2022-12-13	1명	2번 스터디룸	12:00	13:00

cc/Search > check

예약번호	예약날짜	인원	스터디룸 번호	시작시간	종료시간
3	2022-12-13	2명	2번 스터디룸	15:00	17:00

user/Book > book 221213 1 1000 1800 6

2022-12-13 1번 스터디룸 10:00 ~ 18:00 로 정상 예약되었습니다.

user/Search > check

예약번호	예약날짜	인원	스터디룸 번호	시작시간	종료시간
4	2022-12-13	6명	1번 스터디룸	10:00	18:00

aa/Search &gt; check

예약번호	예약날짜	인원	스터디룸 번호	시작시간	종료시간
1	2022-12-13	3명	1번 스터디룸	09:00	10:00
	2022-12-13	3명	2번 스터디룸	10:00	18:00
	2022-12-13	3명	1번 스터디룸	18:00	19:00

bb/Search &gt; check

예약번호	예약날짜	인원	스터디룸 번호	시작시간	종료시간
2	2022-12-13	1명	3번 스터디룸	12:00	13:00

cc/Search &gt; check

예약번호	예약날짜	인원	스터디룸 번호	시작시간	종료시간
3	2022-12-13	2명	3번 스터디룸	15:00	17:00

■ 목표: 스터디룸 예약 시나리오에 대한 최적화 반응 확인 5

- ID: 13.1-5
- 전제: 1번 방(6인실), 2번 방(4인실), 3번 방(2인실)을 제외한 모든 방은 1인실
- 기존 예약: 1번 방(6인실)= a팀(2명, 9시~19시) / b팀(1명, 12시~13시) / c팀(3명, 15시~17시)  
2번 방(4인실)= d팀(4명, 12시~13시) / e팀(1명, 15시~17시)  
3번 방(2인실)=
- 새로운 예약: user(1번 방, 6명, 10시~18시)
- 예상 결과: 1번 방(6인실)= a팀(2명, 9시~10시) / user(6명, 10시~18시)  
2번 방(4인실)= a팀(2명, 10시~12시) / d팀(4명, 12시~13시) / c팀(3명, 15시~17시)  
3번 방(2인실)= b팀(1명, 12시~13시) / e팀(1명, 15시~17시)
- 실제 결과: (상동)
- 화면:

aa/Search &gt; check

예약번호	예약날짜	인원	스터디룸 번호	시작시간	종료시간
5	2022-12-14	2명	1번 스터디룸	09:00	12:00

bb/Search &gt; check

예약번호	예약날짜	인원	스터디룸 번호	시작시간	종료시간
6	2022-12-14	1명	1번 스터디룸	12:00	13:00

cc/Search &gt; check

예약번호	예약날짜	인원	스터디룸 번호	시작시간	종료시간
7	2022-12-14	3명	1번 스터디룸	15:00	17:00

dd/Search &gt; check

예약번호	예약날짜	인원	스터디룸 번호	시작시간	종료시간
8	2022-12-14	4명	2번 스터디룸	12:00	13:00

ee/Search &gt; check

예약번호	예약날짜	인원	스터디룸 번호	시작시간	종료시간
9	2022-12-14	1명	2번 스터디룸	15:00	17:00

```
user/Book > book 221214 1 1000 1800 6
2022-12-14 1번 스터디룸 10:00 ~ 18:00 로 정상 예약되었습니다.
```

```
user/Search > check
```

예약번호	예약날짜	인원	스터디룸 번호	시작시간	종료시간
10	2022-12-14	6명	1번 스터디룸	10:00	18:00

```
aa/Search > check
```

예약번호	예약날짜	인원	스터디룸 번호	시작시간	종료시간
5	2022-12-14	2명	1번 스터디룸	09:00	10:00
	2022-12-14	2명	2번 스터디룸	10:00	12:00

```
bb/Search > check
```

예약번호	예약날짜	인원	스터디룸 번호	시작시간	종료시간
6	2022-12-14	1명	3번 스터디룸	12:00	13:00

```
cc/Search > check
```

예약번호	예약날짜	인원	스터디룸 번호	시작시간	종료시간
7	2022-12-14	3명	2번 스터디룸	15:00	17:00

```
dd/Search > check
```

예약번호	예약날짜	인원	스터디룸 번호	시작시간	종료시간
8	2022-12-14	4명	2번 스터디룸	12:00	13:00

```
ee/Search > check
```

예약번호	예약날짜	인원	스터디룸 번호	시작시간	종료시간
9	2022-12-14	1명	3번 스터디룸	15:00	17:00

- 목표: 스터디룸 예약 시나리오에 대한 최적화 반응 확인 6
  - ID: 13.1-6
  - 전제: 1번 방(6인실), 2번 방(4인실), 3번 방(4인실)을 제외한 모든 방은 1인실
  - 기존 예약: 1번 방(6인실)= a팀(2명, 9시~12시) / b팀(1명, 12시~13시)
    - 2번 방(4인실)= c팀(4명, 12시~13시) / d팀(1명, 15시~17시)
    - 3번 방(4인실)=
  - 새로운 예약: user(1번 방, 6명, 10시~18시)
  - 예상 결과: 1번 방(6인실)= a팀(2명, 9시~10시) / user(6명, 10시~18시)
    - 2번 방(4인실)= a팀(2명, 10시~12시) / c팀(4명, 12시~13시) / d팀(1명, 15시~17시)
    - 3번 방(4인실)= b팀(1명, 12시~13시)
  - 실제 결과: (상동)
  - 화면:

aa/Search > check

예약번호	예약날짜	인원	스터디룸 번호	시작시간	종료시간
1	2022-12-12	2명	1번 스터디룸	09:00	12:00

bb/Search > check

예약번호	예약날짜	인원	스터디룸 번호	시작시간	종료시간
2	2022-12-12	1명	1번 스터디룸	12:00	13:00

cc/Search > check

예약번호	예약날짜	인원	스터디룸 번호	시작시간	종료시간
3	2022-12-12	4명	2번 스터디룸	12:00	13:00

dd/Search > check

예약번호	예약날짜	인원	스터디룸 번호	시작시간	종료시간
4	2022-12-12	1명	2번 스터디룸	15:00	17:00

user/Book > book 221212 1 1000 1800 6

2022-12-12 1번 스터디룸 10:00 ~ 18:00 로 정상 예약되었습니다.

user/Search > check

예약번호	예약날짜	인원	스터디룸 번호	시작시간	종료시간
5	2022-12-12	6명	1번 스터디룸	10:00	18:00

aa/Search > check

예약번호	예약날짜	인원	스터디룸 번호	시작시간	종료시간
1	2022-12-12	2명	1번 스터디룸	09:00	10:00
	2022-12-12	2명	2번 스터디룸	10:00	12:00

bb/Search > check

예약번호	예약날짜	인원	스터디룸 번호	시작시간	종료시간
2	2022-12-12	1명	3번 스터디룸	12:00	13:00

cc/Search > check

예약번호	예약날짜	인원	스터디룸 번호	시작시간	종료시간
3	2022-12-12	4명	2번 스터디룸	12:00	13:00

dd/Search > check

예약번호	예약날짜	인원	스터디룸 번호	시작시간	종료시간
4	2022-12-12	1명	2번 스터디룸	15:00	17:00

■ 목표: 스터디룸 예약 시나리오에 대한 최적화 반응 확인 7

- ID: 13.1-7
- 전제: 1번 방(6인실), 2번 방(4인실), 3번 방(2인실)을 제외한 모든 방은 1인실
- 기존 예약: 1번 방(6인실)= a팀(2명, 9시~12시) / b팀(1명, 12시~13시) / c팀(3명, 14시~16시) / d팀(2명, 16시30분~19시)



2번 방(4인실)= e팀(4명, 12시~13시) / f팀(1명, 15시~17시) / g팀(1명, 18시~19시)

3번 방(2인실)=

- 새로운 예약: user(1번 방, 6명, 10시~18시)
- 예상 결과: 1번 방(6인실)= a팀(2명, 9시~12시) / user(6명, 10시~18시)/ d팀(2명, 18분~19시)
- 2번 방(4인실)= a팀(2명, 10시~12시) / e팀(4명, 12시~13시) / c팀(3명, 14시~16시)) / f팀(1명, 16시~17시) / d팀(2명, 17분~18시) / g팀(1명, 18시~19시)
- 3번 방(2인실)= b팀(1명, 12시~13시) / f팀(1명, 15시~16시) / d팀(2명, 16시30분~17시)
- 실제 결과: (상동)
- 화면:

aa/Search > check					
예약번호	예약날짜	인원	스터디룸 번호	시작시간	종료시간
11	2022-12-15	2명	1번 스터디룸	09:00	12:00
bb/Search > check					
예약번호	예약날짜	인원	스터디룸 번호	시작시간	종료시간
12	2022-12-15	1명	1번 스터디룸	12:00	13:00
cc/Search > check					
예약번호	예약날짜	인원	스터디룸 번호	시작시간	종료시간
13	2022-12-15	3명	1번 스터디룸	14:00	16:00
dd/Search > check					
예약번호	예약날짜	인원	스터디룸 번호	시작시간	종료시간
14	2022-12-15	2명	1번 스터디룸	16:30	19:00
cc/Search > check					
예약번호	예약날짜	인원	스터디룸 번호	시작시간	종료시간
13	2022-12-15	3명	1번 스터디룸	14:00	16:00
dd/Search > check					
예약번호	예약날짜	인원	스터디룸 번호	시작시간	종료시간
14	2022-12-15	2명	1번 스터디룸	16:30	19:00
ee/Search > check					
예약번호	예약날짜	인원	스터디룸 번호	시작시간	종료시간
15	2022-12-15	4명	2번 스터디룸	12:00	13:00
ff/Search > check					
예약번호	예약날짜	인원	스터디룸 번호	시작시간	종료시간
16	2022-12-15	1명	2번 스터디룸	15:00	17:00
gg/Search > check					
예약번호	예약날짜	인원	스터디룸 번호	시작시간	종료시간
17	2022-12-15	1명	2번 스터디룸	18:00	19:00

user/Book > book 221215 1 1000 1800 6

2022-12-15 1번 스터디룸 10:00 ~ 18:00 로 정상 예약되었습니다.

user/Search > check

예약번호	예약날짜	인원	스터디룸 번호	시작시간	종료시간
18	2022-12-15	6명	1번 스터디룸	10:00	18:00

aa/Search > check

예약번호	예약날짜	인원	스터디룸 번호	시작시간	종료시간
11	2022-12-15	2명	1번 스터디룸	09:00	10:00
	2022-12-15	2명	2번 스터디룸	10:00	12:00

bb/Search > check

예약번호	예약날짜	인원	스터디룸 번호	시작시간	종료시간
12	2022-12-15	1명	3번 스터디룸	12:00	13:00

cc/Search > check

예약번호	예약날짜	인원	스터디룸 번호	시작시간	종료시간
13	2022-12-15	3명	2번 스터디룸	14:00	16:00

dd/Search > check

예약번호	예약날짜	인원	스터디룸 번호	시작시간	종료시간
14	2022-12-15	2명	3번 스터디룸	16:30	17:00
	2022-12-15	2명	2번 스터디룸	17:00	18:00
	2022-12-15	2명	1번 스터디룸	18:00	19:00

cc/Search > check

예약번호	예약날짜	인원	스터디룸 번호	시작시간	종료시간
13	2022-12-15	3명	2번 스터디룸	14:00	16:00

dd/Search > check

예약번호	예약날짜	인원	스터디룸 번호	시작시간	종료시간
14	2022-12-15	2명	3번 스터디룸	16:30	17:00
	2022-12-15	2명	2번 스터디룸	17:00	18:00
	2022-12-15	2명	1번 스터디룸	18:00	19:00

ee/Search > check

예약번호	예약날짜	인원	스터디룸 번호	시작시간	종료시간
15	2022-12-15	4명	2번 스터디룸	12:00	13:00

ff/Search > check

예약번호	예약날짜	인원	스터디룸 번호	시작시간	종료시간
16	2022-12-15	1명	3번 스터디룸	15:00	16:00
	2022-12-15	1명	2번 스터디룸	16:00	17:00

gg/Search > check

예약번호	예약날짜	인원	스터디룸 번호	시작시간	종료시간
17	2022-12-15	1명	2번 스터디룸	18:00	19:00