



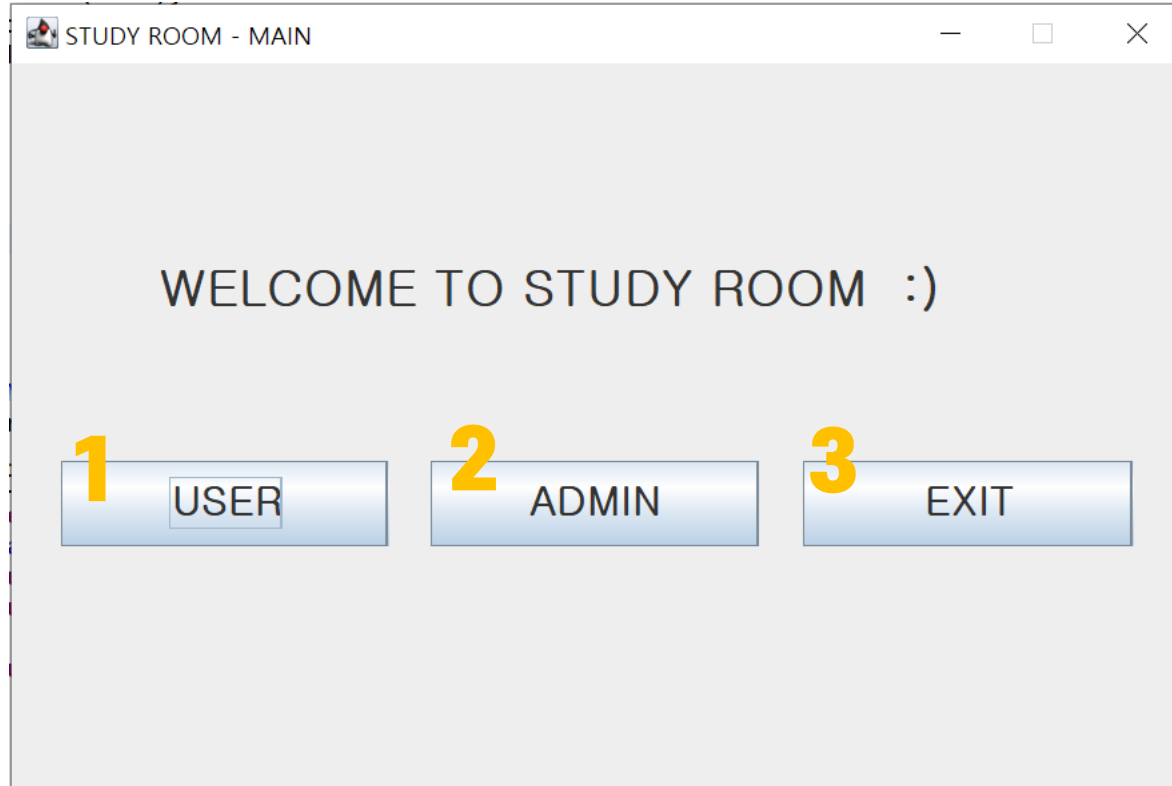
# **JAVA PROGRAMMING**

## **-STUDY ROOM GUI 적용-**

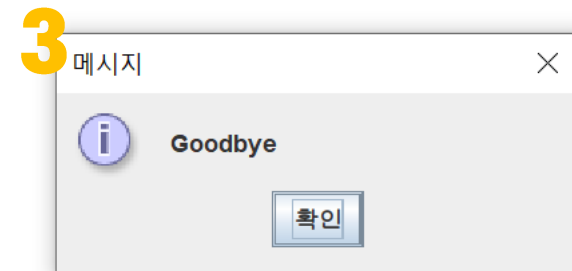
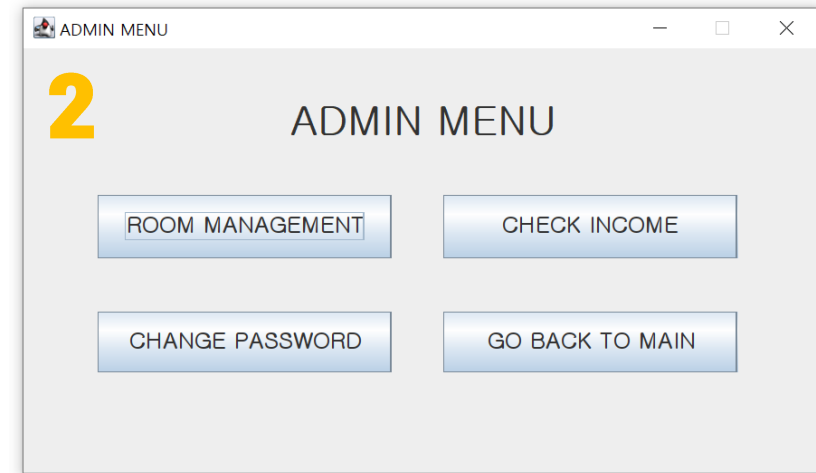
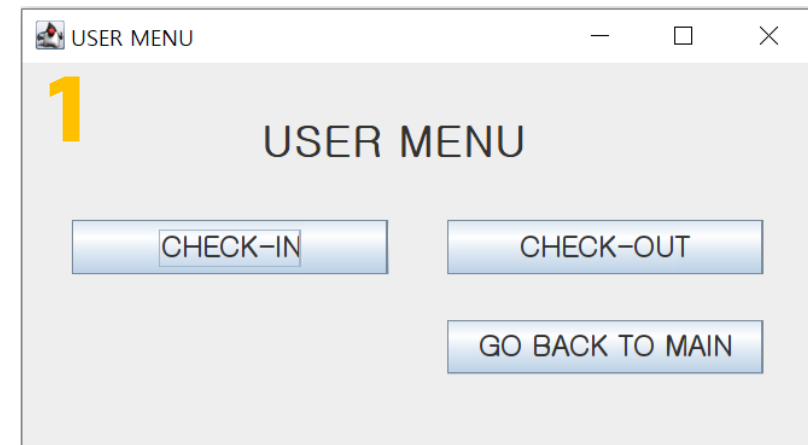


# 첫번째 입장 화면 [MainGUI 클래스]

Jframe과 JLabel, JButton 등을 이용하여 만든다.



버튼을 클릭하여 사용자 / 관리자 / 종료 중 하나를 선택한다.

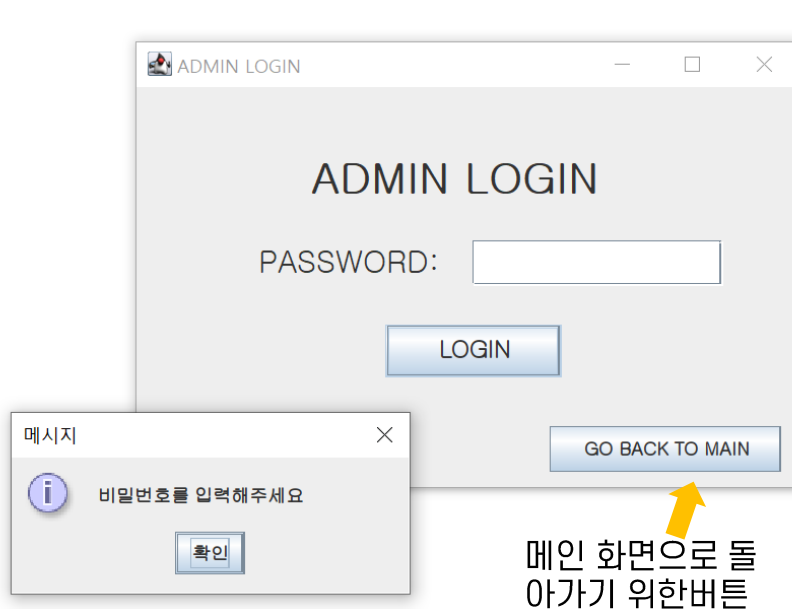


# 관리자 로그인 화면

## [AdminLogin 클래스]

<관리자>

- 패스워드 데이터는 RoomManagement클래스의 패스워드 필드를 파일에 저장해서 읽어온다.
- 텍스트필드의 값과 패스워드 데이터가 일치하는지 비교한다.



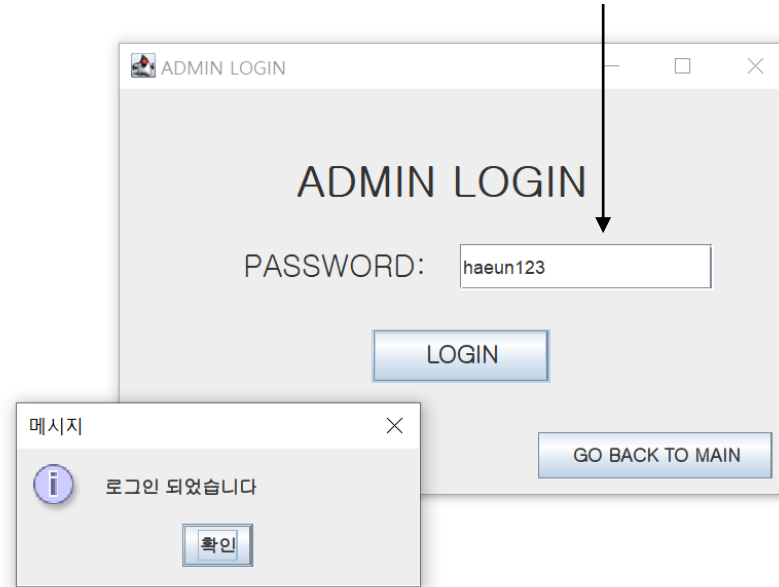
<비밀번호를 입력하지 않은 경우>



팝업창



로그인 실패



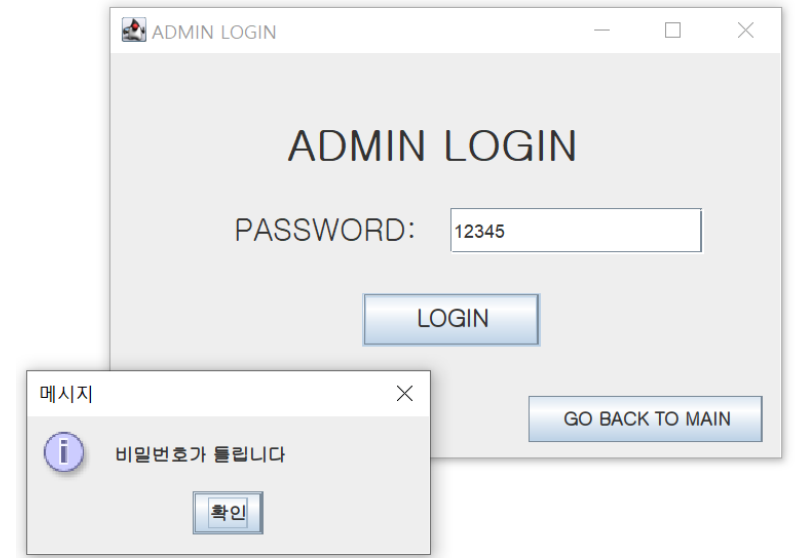
<올바른 비밀번호를 입력한 경우>



팝업창



로그인 성공



<잘못된 비밀번호를 입력한 경우>



팝업창



로그인 실패

Jframe과 JLabel, JButton 등을 이용하여 만든다.

방 생성/삭제/수정



ROOM MANAGEMENT

CHECK INCOME



수입 확인

비밀번호 바꾸기

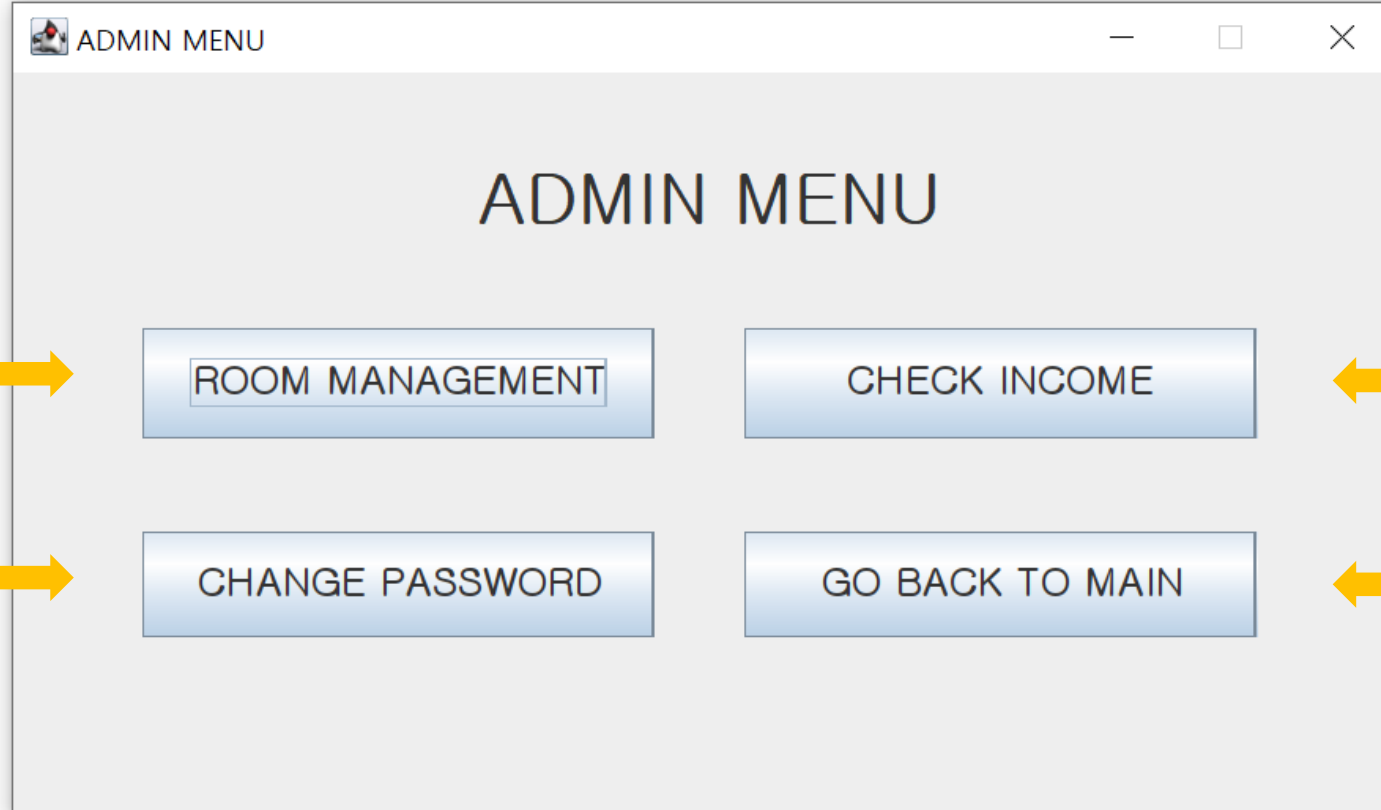


CHANGE PASSWORD

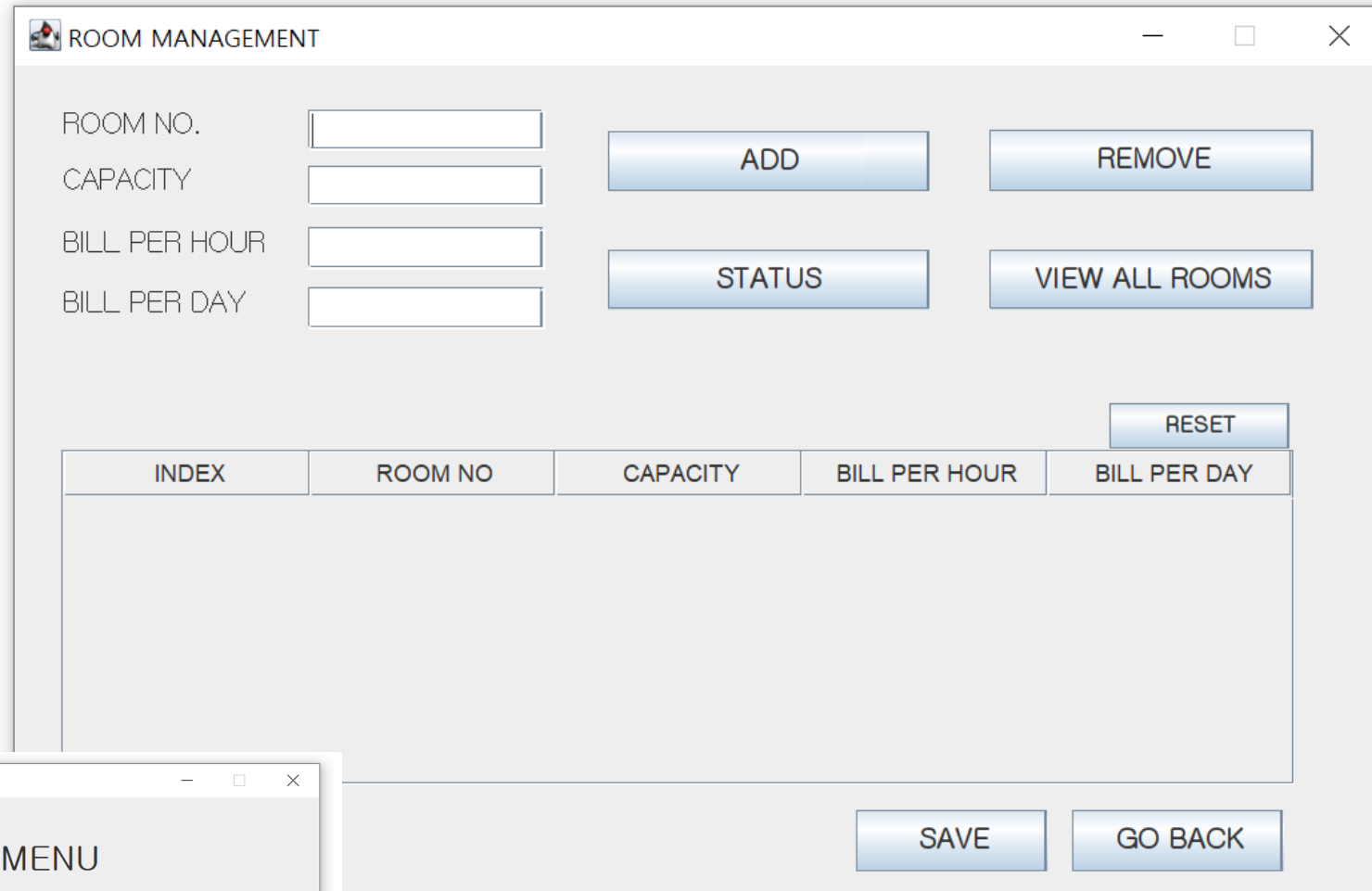
GO BACK TO MAIN



메인으로 돌아가기

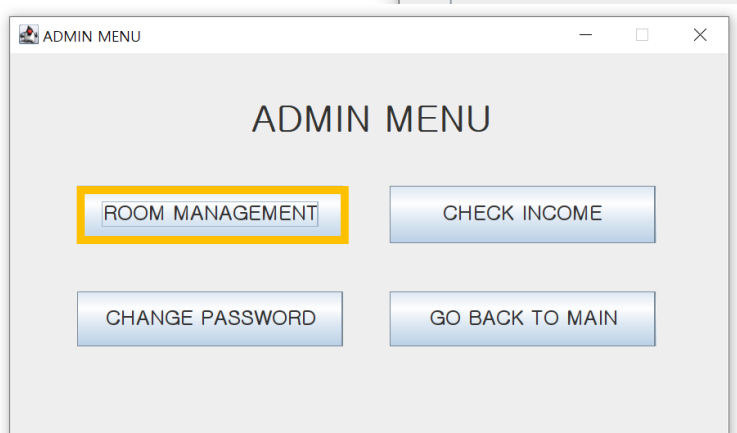


버튼을 클릭하여 원하는 메뉴를 선택한다.



A screenshot of a 'ROOM MANAGEMENT' window. It features four input fields on the left for 'ROOM NO.', 'CAPACITY', 'BILL PER HOUR', and 'BILL PER DAY'. To the right of these fields are four buttons: 'ADD', 'REMOVE', 'STATUS', and 'VIEW ALL ROOMS'. A 'RESET' button is located above a table. The table has five columns: 'INDEX', 'ROOM NO', 'CAPACITY', 'BILL PER HOUR', and 'BILL PER DAY'. At the bottom right of the window are 'SAVE' and 'GO BACK' buttons.

INDEX	ROOM NO	CAPACITY	BILL PER HOUR	BILL PER DAY
-------	---------	----------	---------------	--------------



A screenshot of an 'ADMIN MENU' window. It contains four buttons arranged in a 2x2 grid: 'ROOM MANAGEMENT' (highlighted with a yellow border), 'CHECK INCOME', 'CHANGE PASSWORD', and 'GO BACK TO MAIN'.

← 관리자 메뉴 화면에서  
ROOM MANAGEMENT 버튼 클릭 시 나타나는 화면.

# 방 관리 화면 – 기능1. 방 생성 [ManagementGUI 클래스]

<관리자>

1 각각의 텍스트필드에 기입된 정보를 Room클래스 생성자의 매개변수들에 대입하여 사용한다.

```
arr[0]=Integer.toString(table.getRowCount()+1);  
arr[1]=room.getRoomNum();  
arr[2]=room.getCapacity();  
arr[3]=room.getBillPerHour();  
arr[4]=room.getBillPerDay();  
model.addRow(arr);
```

3 JTable에 배열을 생성하여 텍스트필드에 기입된 정보들이 배열 형식으로 삽입되게 한다.

The screenshot shows a window titled "ROOM MANAGEMENT". It contains several input fields and buttons. A yellow box highlights the input fields for "ROOM NO.", "CAPACITY", "BILL PER HOUR", and "BILL PER DAY", which are labeled with a yellow "1". Another yellow box highlights the "ADD" button, labeled with a yellow "2". A third yellow box highlights the first row of a table, labeled with a yellow "3". The table has columns: INDEX, ROOM NO, CAPACITY, BILL PER HOUR, and BILL PER DAY. The first row contains the values: 1, 101, 2, 1000, and 2000. A fourth yellow box highlights the "SAVE" button, labeled with a yellow "4".

INDEX	ROOM NO	CAPACITY	BILL PER HOUR	BILL PER DAY
1	101	2	1000	2000

저장버튼을 누르면 동시에 파일에 수정된 내용이 저장된다.  
수정된 정보가 Room.dat파일에 저장된다.

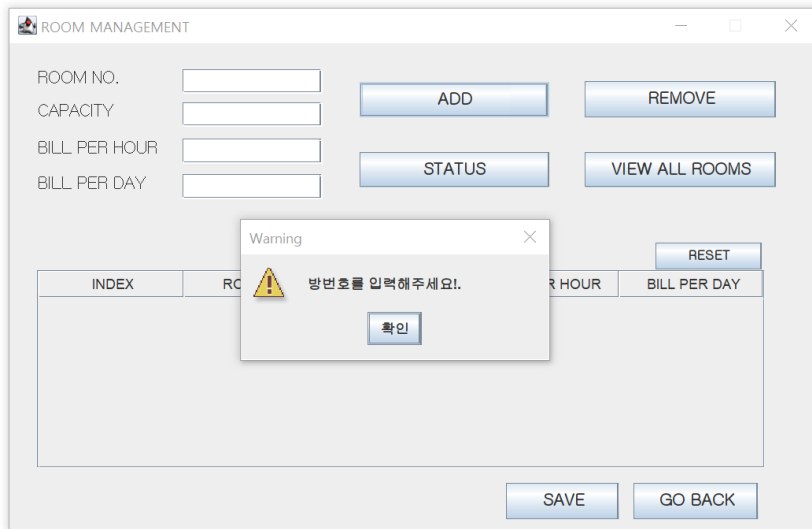
1. 방 번호, 방 수용 인원, 요금 등 방 정보를 기입한다.

2. 추가 버튼을 클릭한다.

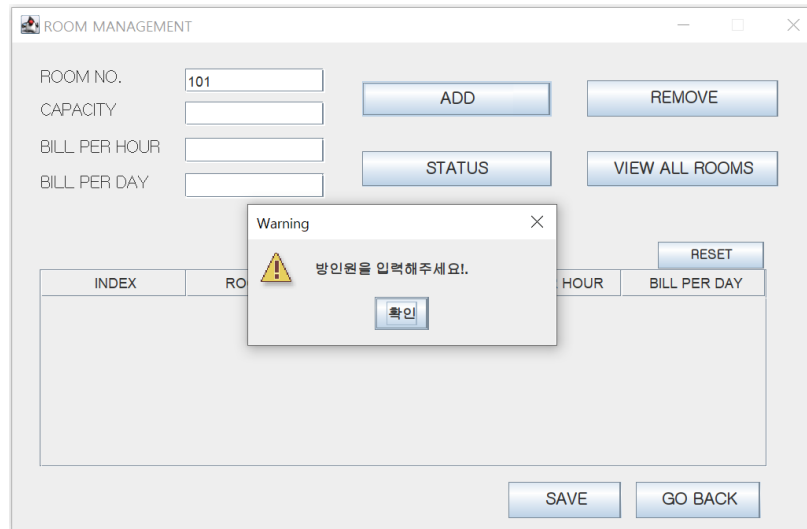
3. 테이블표에 만들어진 방이 추가된다.

4. SAVE 버튼을 클릭하여 저장한다.

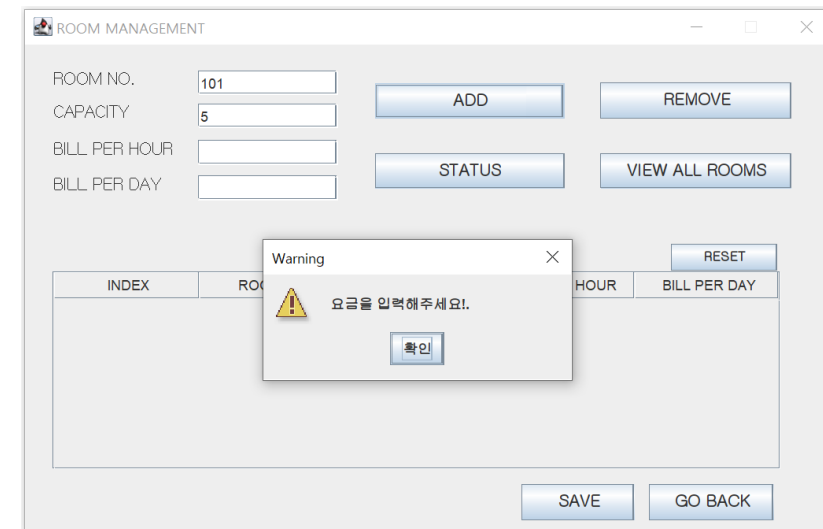
### < 방 생성 예외 처리 >



1. 방 번호 미 입력 후 ADD버튼 클릭 시



2. 방 인원 미 입력 후 ADD버튼 클릭 시



3. 방 요금 미 입력 후 ADD버튼 클릭 시

# 방 관리 화면 – 기능2. 전체 방 조회 [ManagementGUI 클래스]

<관리자>

The screenshot shows a Java Swing window titled "ROOM MANAGEMENT". It contains four text input fields for "ROOM NO.", "CAPACITY", "BILL PER HOUR", and "BILL PER DAY". To the right of these fields are four buttons: "ADD", "REMOVE", "STATUS", and "VIEW ALL ROOMS". The "VIEW ALL ROOMS" button is highlighted with a yellow border and a yellow number "1" next to it. Below the input fields is a "RESET" button. At the bottom of the window are "SAVE" and "GO BACK" buttons. In the center is a table with 5 rows and 5 columns. The columns are labeled "INDEX", "ROOM NO", "CAPACITY", "BILL PER HOUR", and "BILL PER DAY". The table contains the following data:

INDEX	ROOM NO	CAPACITY	BILL PER HOUR	BILL PER DAY
1	101	2	1000	2000
2	102	3	1500	3000
3	103	5	2000	4000
4	104	7	2500	5000
5	105	9	3000	6000

RoomUI 클래스의  
ViewAllRoom 함수와 연결  
한다.

```
arr[0]=Integer.toString(table.getRowCount()+1);  
arr[1]=room.getRoomNum();  
arr[2]=room.getCapacity();  
arr[3]=room.getBillPerHour();  
arr[4]=room.getBillPerDay();  
model.addRow(arr);
```

Jtable에 배열을 생성하여  
텍스트필드에 기입된 정보  
들이 배열 형식으로 삽입되  
게 한다.

1. VIEW ALL ROOMS 버튼을 클릭한다

2. 테이블표에 생성된 모든 방이 나온다.



# 방 관리 화면 - 기능3. 방 삭제 [ManagementGUI 클래스]

<관리자>

ROOM MANAGEMENT

ROOM NO.

CAPACITY

BILL PER HOUR

BILL PER DAY

ADD REMOVE

STATUS VIEW ALL ROOMS

RESET

INDEX	ROOM NO	CAPACITY	BILL PER HOUR	BILL PER DAY
1	101	2	1000	2000
2	102	3	1500	3000
3	103	5	2000	4000
4	104	7	2500	5000
5	105	9	3000	6000

메시지

방 104가 삭제되었습니다

확인

SAVE GO BACK

ROOM MANAGEMENT

ROOM NO.

CAPACITY

BILL PER HOUR

BILL PER DAY

ADD REMOVE

STATUS VIEW ALL ROOMS

RESET

INDEX	ROOM NO	CAPACITY	BILL PER HOUR	BILL PER DAY
1	101	2	1000	2000
2	102	3	1500	3000
3	103	5	2000	4000
5	105	9	3000	6000

SAVE GO BACK

## 주요 함수들

```
int row = table.getSelectedRow();  
int value = (int) model.getValueAt(row, 1);  
roomMan.removeRoom(value);
```

테이블의 row로 해당 열의 정보를 찾아서 RoomManagement 클래스의 removeRoom() 함수를 사용하여 삭제한다.

1. 테이블표에서 방을 클릭한다.
2. REMOVE 버튼을 누른다 -> 방이 제거된다.
3. 팝업창이 뜬다
4. 테이블표와 데이터에서 방이 제거된다.
5. SAVE 버튼을 클릭하여 저장 한다. 저장버튼을 누르면 동시에 파일에 수정된 내용이 저장된다. (수정된 정보가 Room.dat파일에 저장된다.)

## < 방 삭제 예외 처리 >

The screenshot shows a 'ROOM MANAGEMENT' window with a form for adding or updating room information. The form includes fields for 'ROOM NO.', 'CAPACITY', 'BILL PER HOUR', and 'BILL PER DAY'. There are buttons for 'ADD', 'REMOVE', 'STATUS', 'VIEW ALL ROOMS', 'RESET', 'SAVE', and 'GO BACK'. A table displays room data with columns for 'INDEX', 'ROOM NO.', 'HOUR', and 'BILL PER DAY'. A 'Warning' dialog box is overlaid on the table, displaying a warning icon and the message '삭제할 방을 선택해주세요' (Please select a room to delete). The dialog box has an '확인' (OK) button.

INDEX	ROOM NO.	HOUR	BILL PER DAY
1	101	1000	2000
2	102	1500	3000
3	103	2000	4000

방 선택 안하고 REMOVE 버튼 클릭 시

ROOM MANAGEMENT

ROOM NO.

CAPACITY

BILL PER HOUR

BILL PER DAY

ADD REMOVE

STATUS VIEW ALL ROOMS

RESET

INDEX	ROOM NO	CAPACITY	BILL PER HOUR	BILL PER DAY
-------	---------	----------	---------------	--------------

SAVE GO BACK

1. REMOVE 버튼을 누른다.

2. 테이블표에서 방이 모두 사라진다.

주요함수

```
DefaultTableModel model = (DefaultTableModel)table.getModel();  
model.setNumRows(0);
```

ROOM STATUS & OCCUPACY

INDEX	ROOM NO	CAPACITY	BILL PER HOUR	BILL PER DAY	USING	USER
1	101	2	1000	2000	false	
2	102	3	1500	3000	true	User@9dfd2e0
3	103	5	2000	4000	false	
4	104	7	2500	5000	true	User@55c88361
5	105	9	3000	6000	false	

체크인

GO BACK

ROOM STATUS & OCCUPACY

INDEX	ROOM NO	CAPACITY	BILL PER HOUR	BILL PER DAY	USING	USER
1	101	2	1000	2000	false	
2	102	3	1500	3000	true	User@59063fb1
3	103	5	2000	4000	false	
4	104	7	2500	5000	false	
5	105	9	3000	6000	false	

체크아웃

GO BACK

ROOM MANAGEMENT

ROOM NO.

CAPACITY

BILL PER HOUR

BILL PER DAY

ADD REMOVE

STATUS VIEW ALL ROOMS

RESET

INDEX	ROOM NO	CAPACITY	BILL PER HOUR	BILL PER DAY
-------	---------	----------	---------------	--------------

SAVE GO BACK

```
arr[0]=Integer.toString(table.getRowCount()+1);
arr[1]=room.getRoomNum();
arr[2]=room.getCapacity();
arr[3]=room.getBillPerHour();
arr[4]=room.getBillPerDay();
model.addRow(arr);
```

Jtable에 배열을 생성하여 텍스트필드에 기입된 정보들이 배열 형식으로 삽입되게 한다.

방의 정보와 사용여부 / 사용자를 알 수 있다 (번호가 아닌 것이 뜸 – 수정해야 함)



ManagementGUI 클래스에서 STATUS 버튼 클릭 시 나타나는 화면.

CHECK-INCOME

DAILY INCOME:  Month  Day

MONTHLY INCOME:

TOTAL INCOME:

TOTAL INCOME

총 수입은: 5500입니다.

CHECK-IN

Capacity:

INDEX	ROOM NO	CAPACITY	BILL PER HOUR	BILL PER DAY	USING
1	101	3	1000	2000	false
2	102	5	1500	3000	false
3	103	7	2000	4000	false

message

phoneNo 3225

User@3ace0f88님 103번 방에 체크인 되었습니다

CHECK-OUT

Phone Number:

message

체크아웃 되었습니다

CHECK-INCOME

DAILY INCOME:  Month  Day

MONTHLY INCOME:

TOTAL INCOME:

TOTAL INCOME

총 수입은: 7500입니다.

## 2000원 방 체크인/체크아웃

사용자가 체크아웃 시, 시간에 따라 요금을 계산한 후 수입 정보에 더하여 진다. -> ManagerIncome.dat 파일에 저장되고 불러와진다.

총 수입 정보만 파일에 저장, 갱신 됨 -> 월별/일별 수입 데이터 불러오기 구현 해야함

ADMIN MENU

관리자 메뉴 화면에서  
CHECK-INCOME 버튼 클릭 시 나타나는 화면.

<미 입력 시 예외처리>

Warning

새로운 비밀번호를 입력해주세요!

확인

CHANGE PASSWORD

CHANGE ADMIN'S PASSWORD

LOGIN ID: admin

NEW PASSWORD: 1 123

2 SAVE GO BACK TO MENU

3 메시지

비밀번호가 성공적으로 바뀌었습니다.

확인

ADMIN MENU

ROOM MANAGEMENT CHECK INCOME

CHANGE PASSWORD GO BACK TO MAIN

1. 새로운 비밀번호를 입력한다.

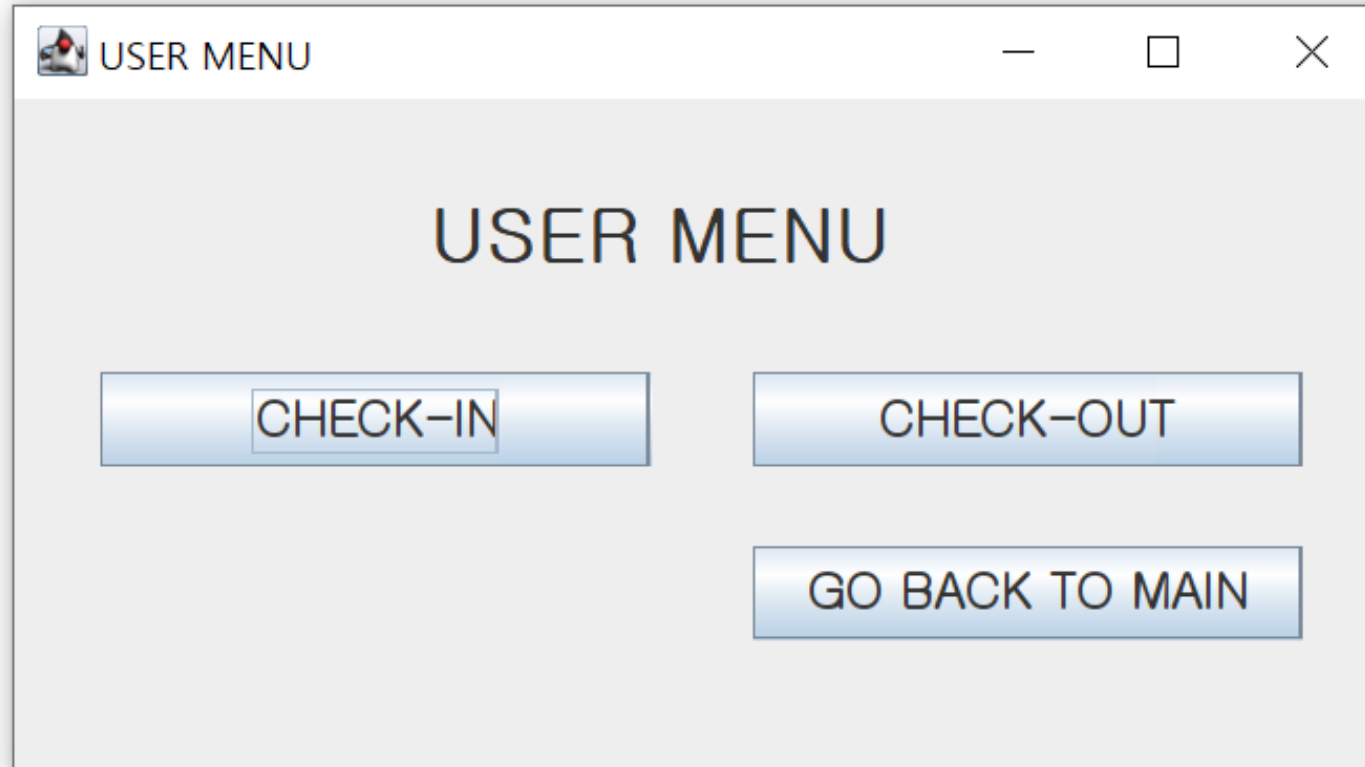
2. SAVE 버튼을 눌러 새로운 비밀번호를 저장한다.

3. 비밀번호가 성공적으로 바뀌면 팝업창이 나온다.

관리자 메뉴 화면에서  
CHANGE PASSWORD 버튼 클릭 시 나타나는 화면.

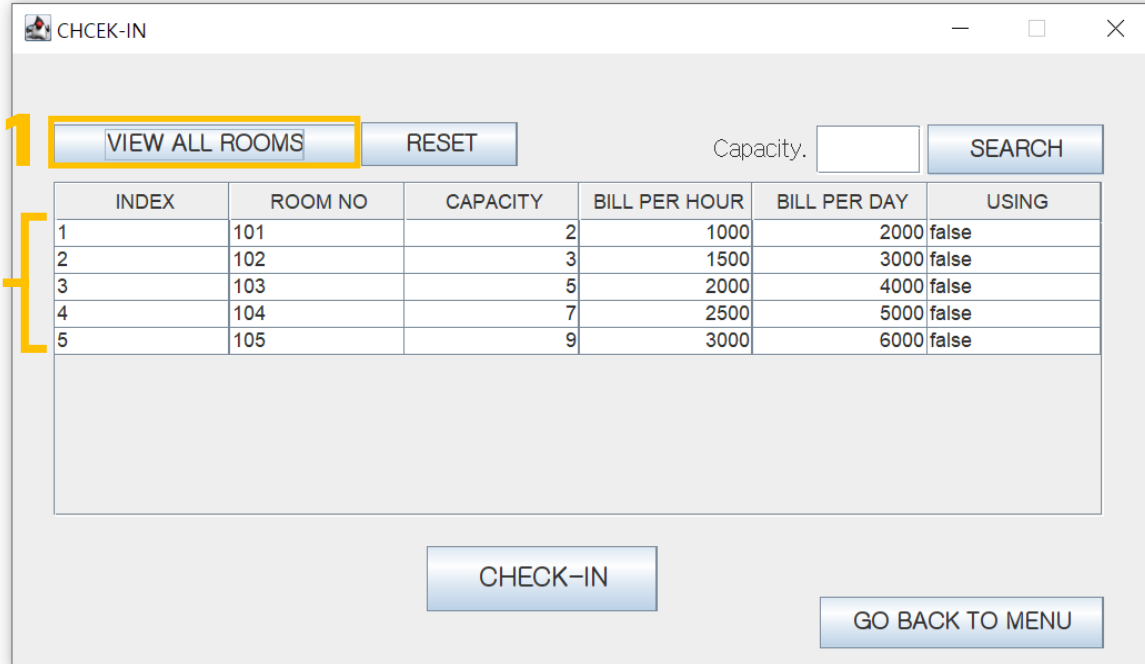
텍스트필드로 받은 패스워드 정보가 Manager.dat에 저장된 패스워드 정보를 덮어쓰고 재저장된다.

Jframe과 JLabel, JButton 등을 이용하여 만든다.



버튼을 클릭하여 원하는 메뉴를 선택한다.

#### <생성 된 모든 방 보기>



1 [VIEW ALL ROOMS] [RESET] Capacity. [ ] [SEARCH]

INDEX	ROOM NO	CAPACITY	BILL PER HOUR	BILL PER DAY	USING
1	101	2	1000	2000	false
2	102	3	1500	3000	false
3	103	5	2000	4000	false
4	104	7	2500	5000	false
5	105	9	3000	6000	false

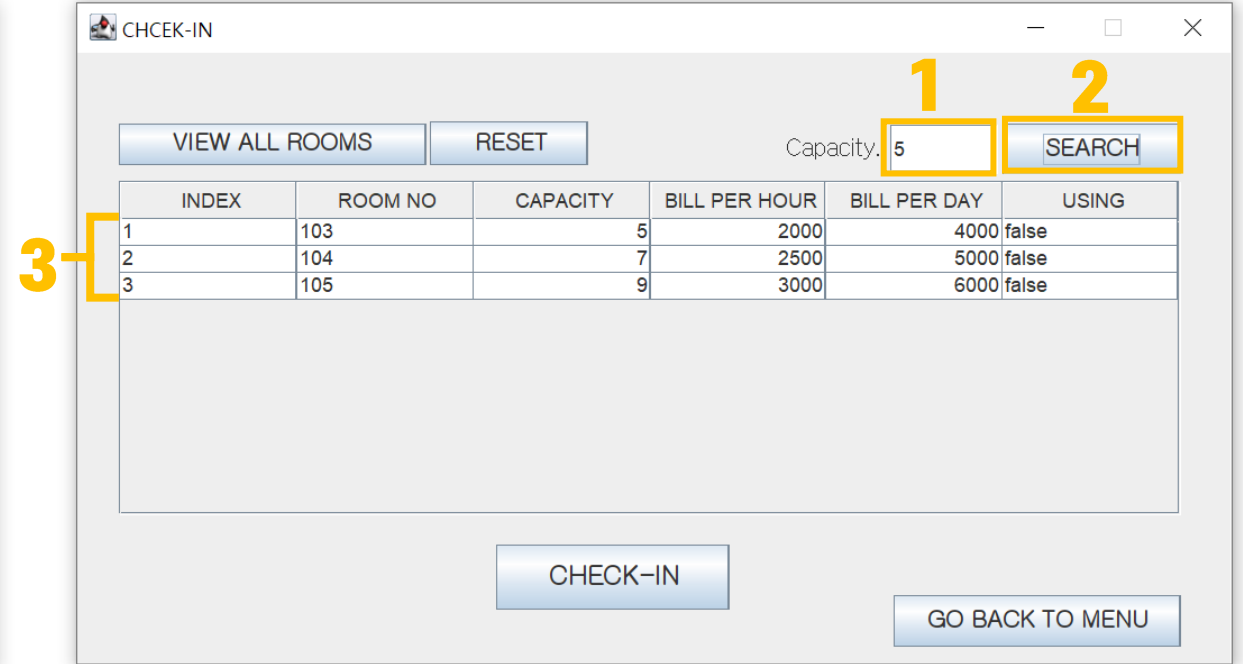
2 [ ]

[CHECK-IN] [GO BACK TO MENU]

1. 모든 방 보기 버튼을 클릭한다.
2. 테이블에 생성된 전체 방이 나온다.
3. 전체 리스트 중 원하는 방을 선택한다.

RoomManager에 SearchRoom() 함수를 불러와서 정보들을 JTable에 배열 형식으로 넣어서 보여준다.

#### <입력된 인원수 이상이고 비어 있는 방 보기>



1 [VIEW ALL ROOMS] [RESET] Capacity. 5 2 [SEARCH] 3

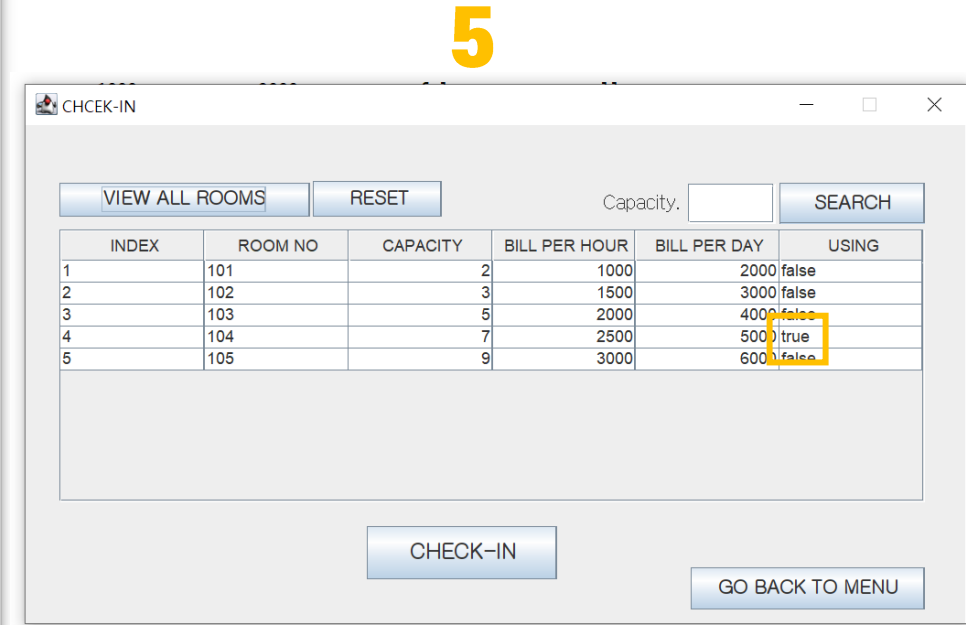
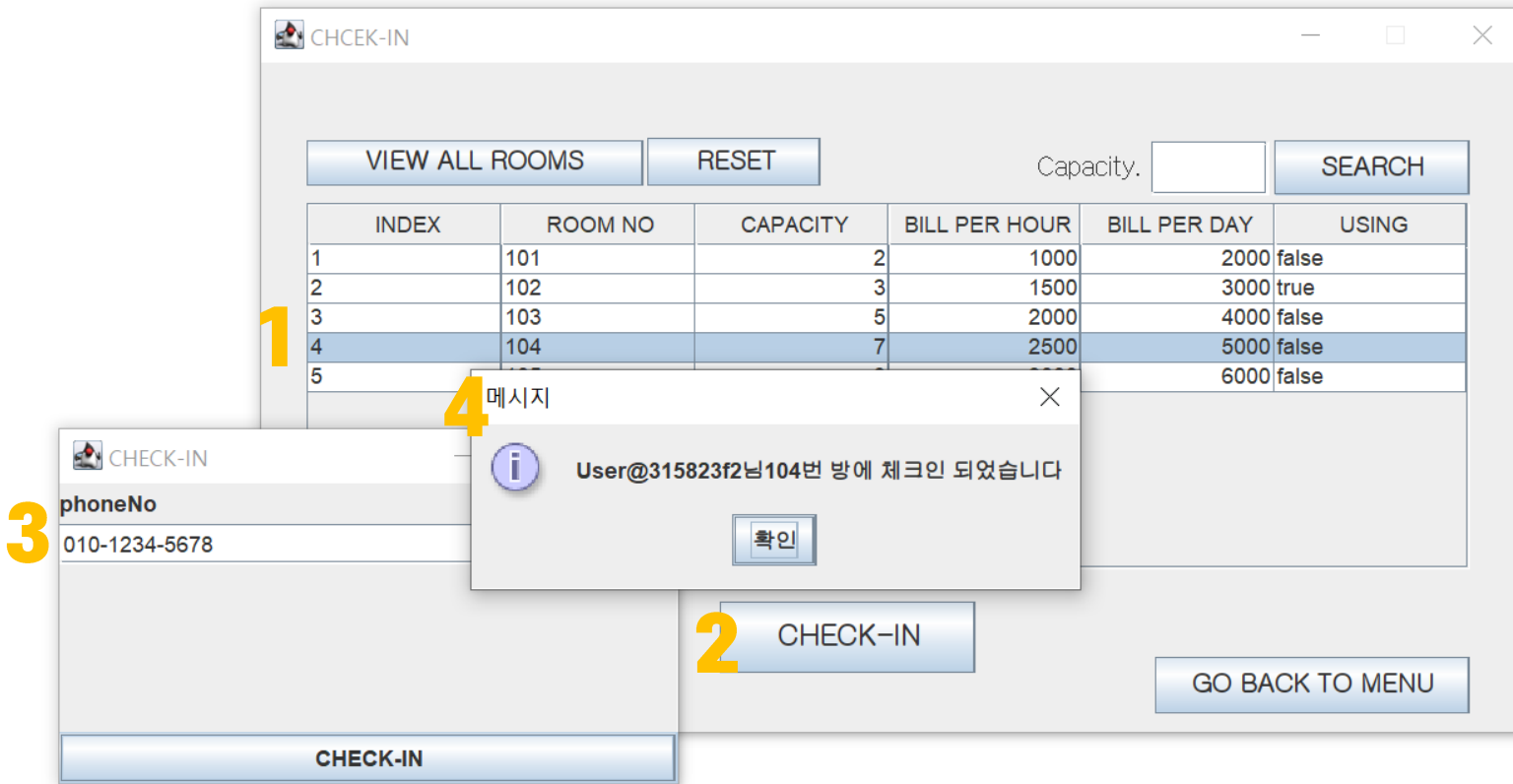
INDEX	ROOM NO	CAPACITY	BILL PER HOUR	BILL PER DAY	USING
1	103	5	2000	4000	false
2	104	7	2500	5000	false
3	105	9	3000	6000	false

[CHECK-IN] [GO BACK TO MENU]

1. 원하는 인원수를 입력한다.
2. 검색 버튼을 클릭한다.
3. 입력한 인원수 이상의 방 리스트를 나타낸다 .

RoomManager에 SearchEmptyRoom() 함수를 불러와서 정보들을 JTable에 배열 형식으로 넣어서 보여준다.



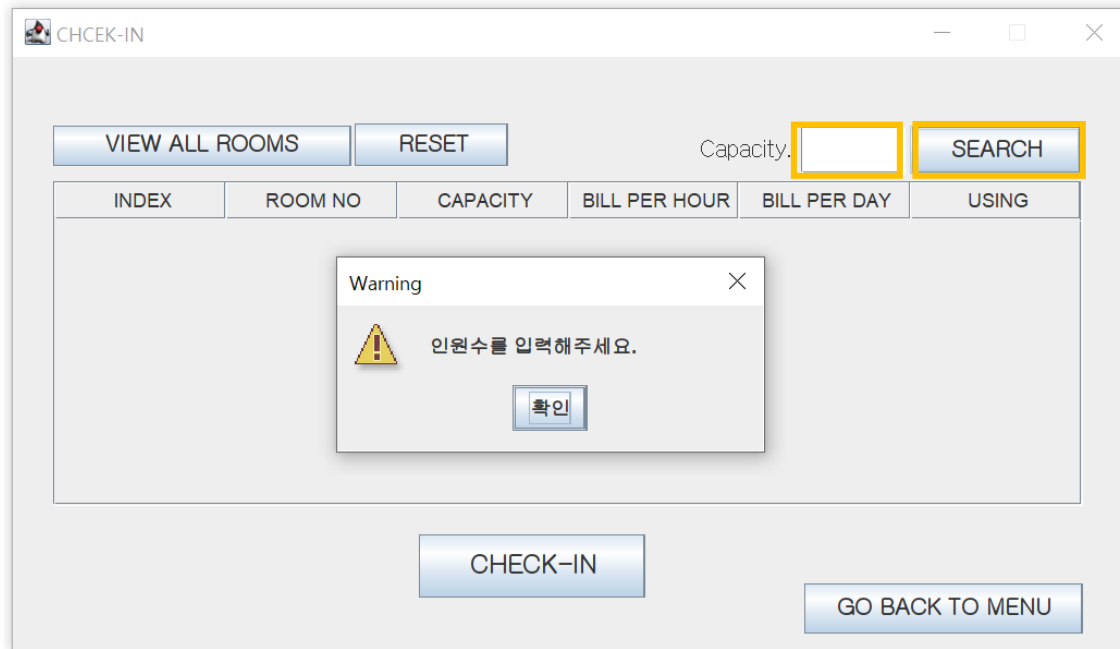


### <주요 함수들>

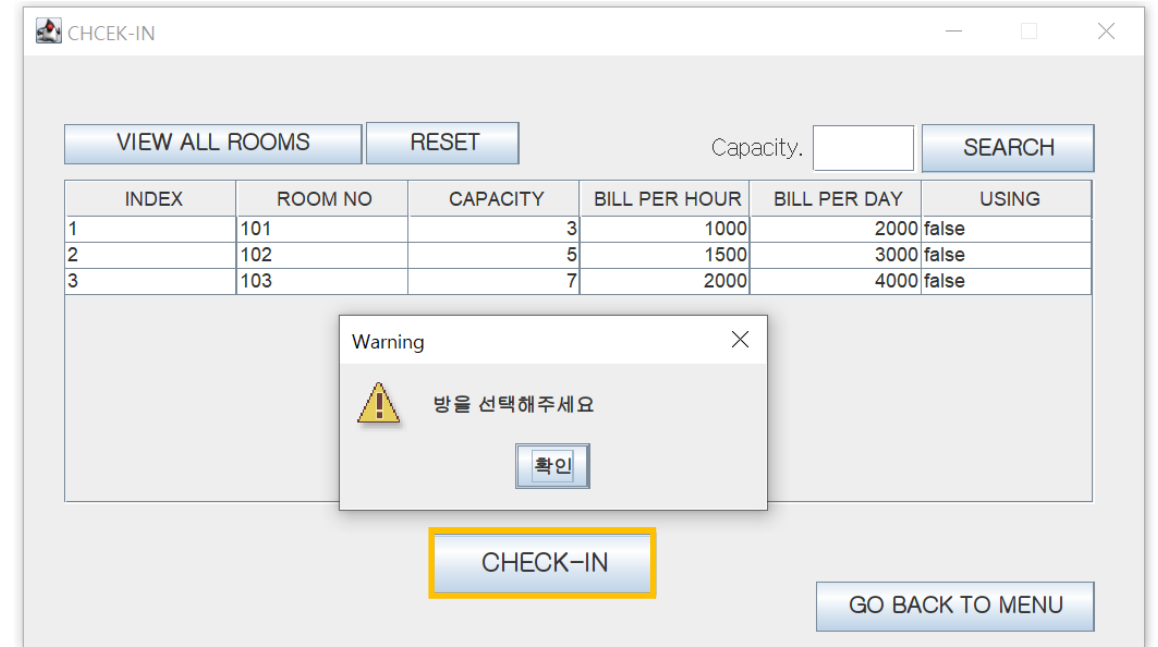
1. JTable에서 원하는 방의 row를 클릭한다.
2. 체크인 버튼을 클릭한다.
3. 사용자의 핸드폰 번호를 기입하는 창이 나온다.
4. Jtable에서 104번방의 USING이 True(사용중)가 되어 나타난다.

- `int row = table.getSelectedRow();`
- `DefaultTableModel model = (DefaultTableModel)table.getModel();`
- `int roomNum = (int) model.getValueAt(row, 1);`
- `String phoneNo = tPhoneNum.getText();`
- `User user = new User(phoneNo);`
- `roomMan.checkIn(roomNum, user);`

### < 체크인 화면 예외 처리 >

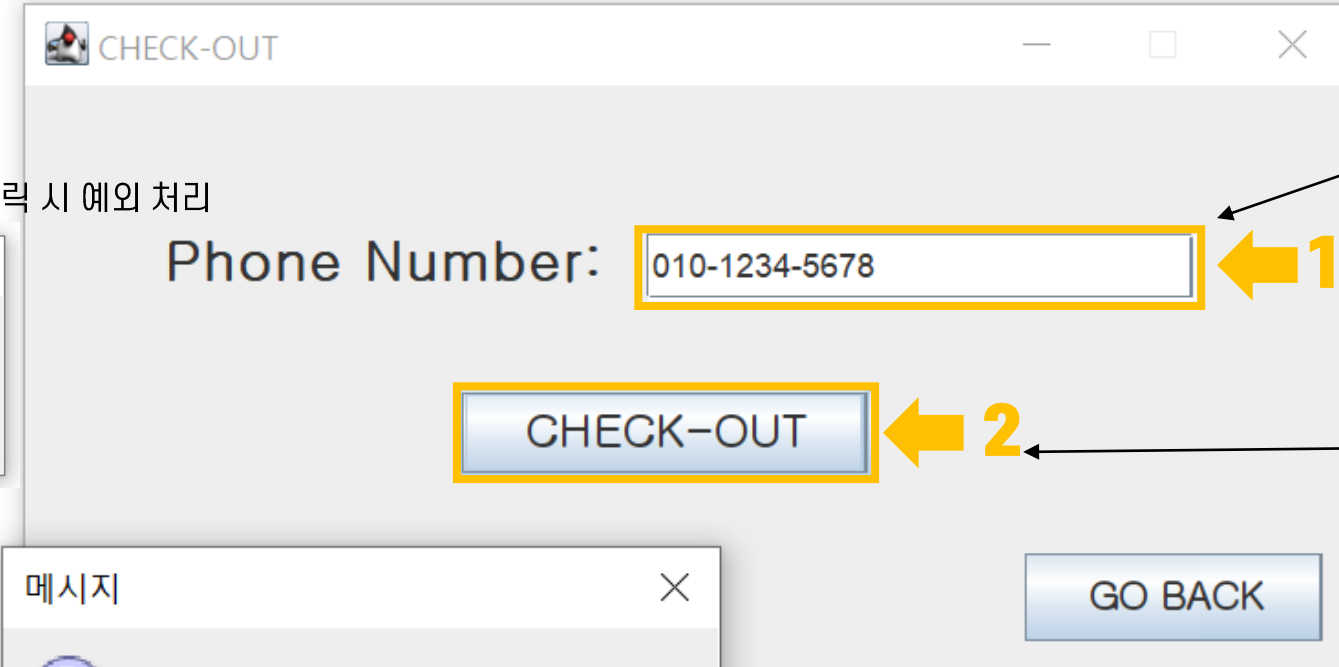
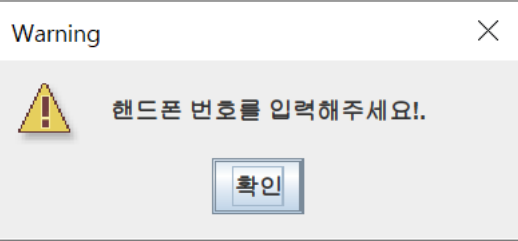


인원수 누락 후 SEARCH 버튼 클릭 시



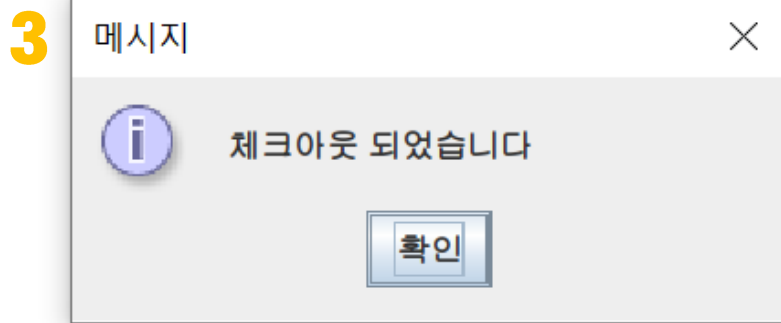
방 선택 안하고 CHECK-IN 버튼 클릭 시

번호 미 기입 후 CHECK-OUT 버튼 클릭 시 예외 처리



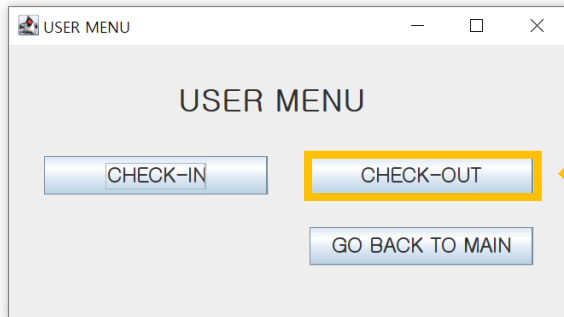
텍스트필드로 받은 번호 정보를 매개변수로 User 클래스 생성자를 생성한 후 RoomManagement 클래스의 CheckOut() 함수를 구현한다.

체크아웃 버튼을 누르면 수정된 정보가 Room.dat 파일에 저장된다.

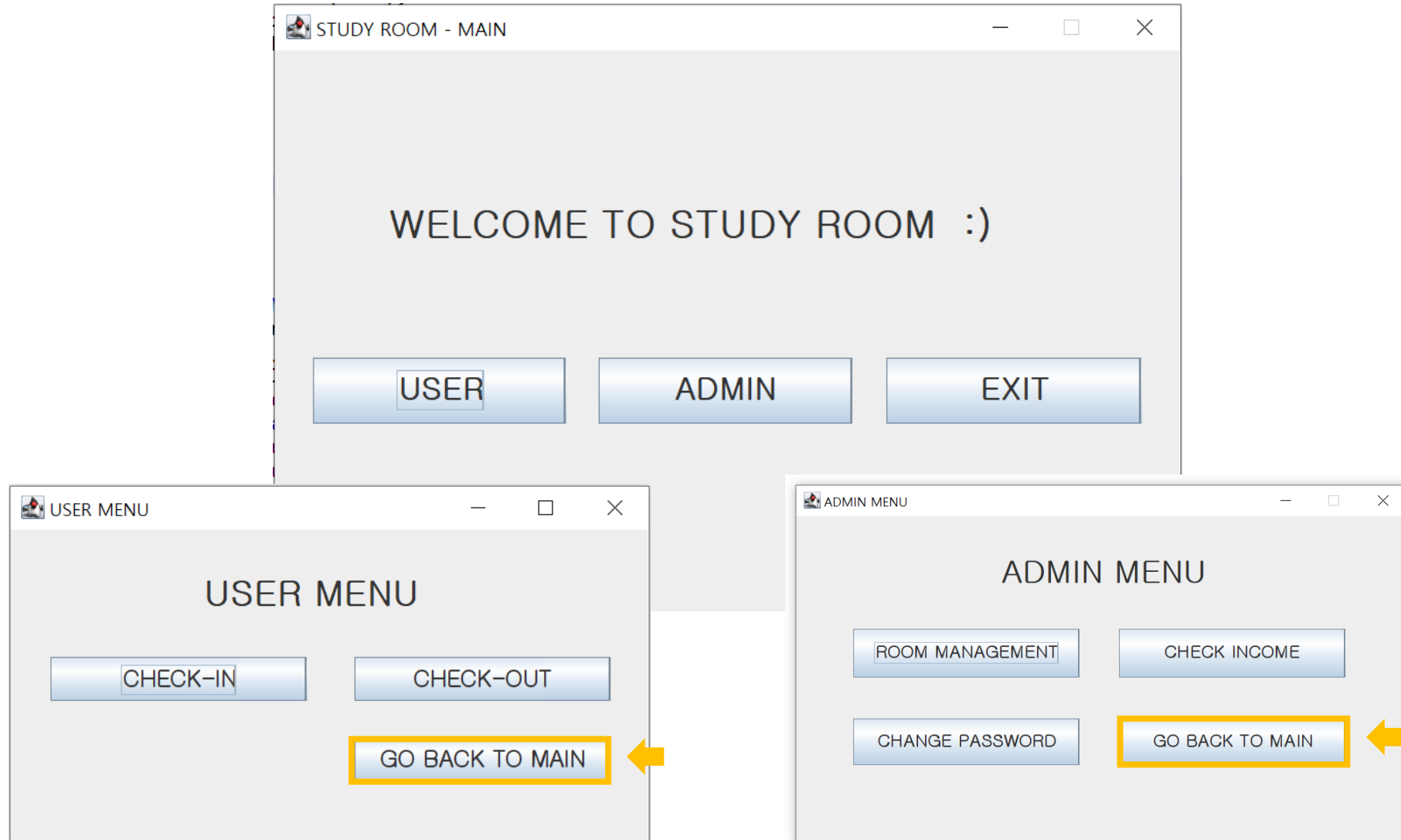


<체크아웃 버튼 클릭 시 나타나는 팝업창>

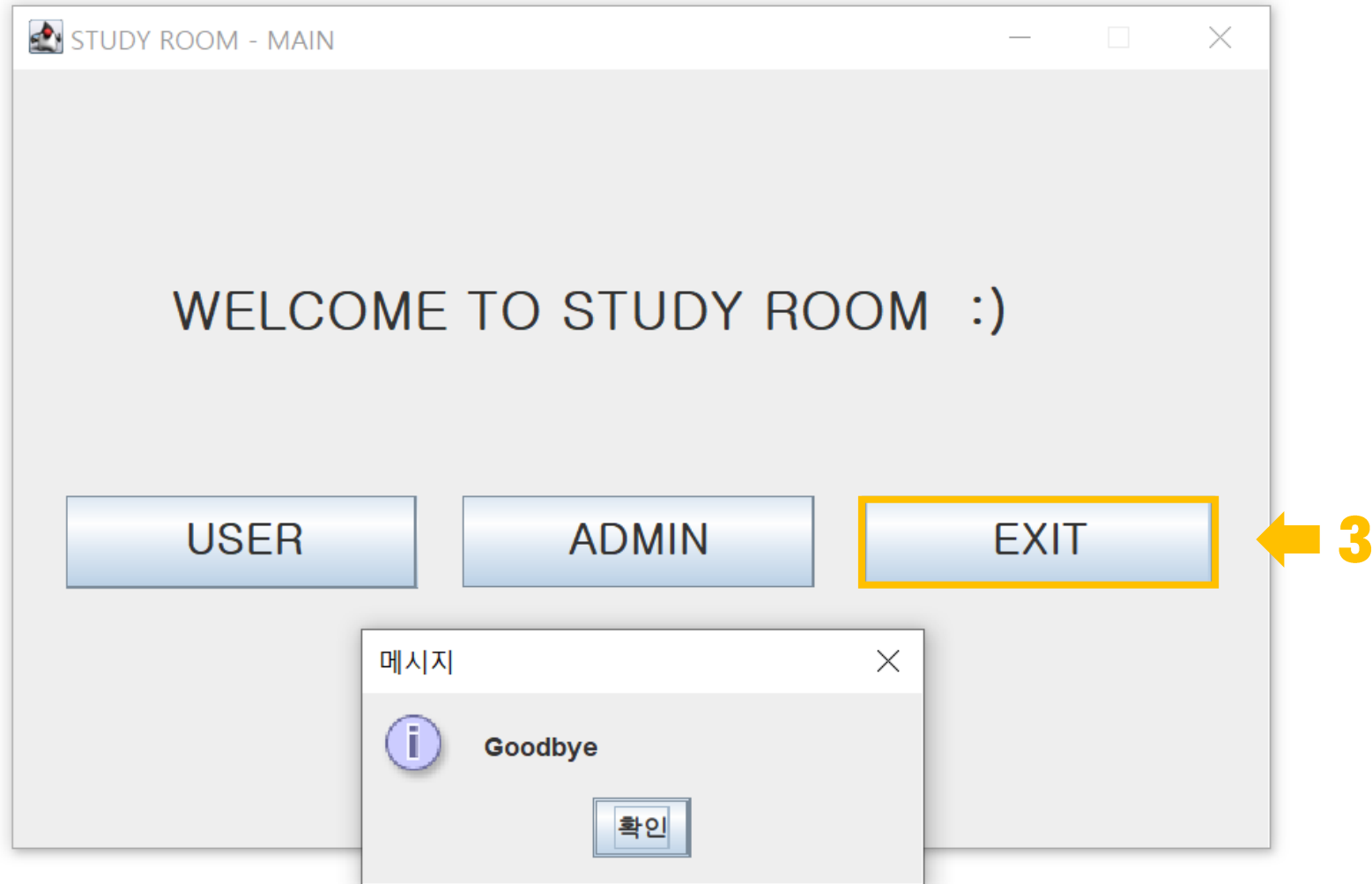
1. 이용자 핸드폰번호를 입력한다.
2. 체크아웃 버튼을 누른다.
3. 체크아웃 인사말 팝업창이 나온다.



사용자 메뉴 화면에서 CHECK - OUT 버튼 클릭 시 나타나는 화면.



관리자 메뉴 화면/ 이용자 메뉴 화면에서  
GO BACK TO MAIN 버튼 클릭 시 나타나는 화면



EXIT 버튼 클릭 시 팝업창이 나타나고 종료된다.

# 감사합니다

---