

최종보고서

18011529 김효경

(팀원: 18011583 김성훈)

1.요구사항명세서

2020 년 1 학기 데이터베이스	요구사항 명세서	문서번호	
		페이지번호	1/5 페이지
		작 성 자(학번/이름)	18011583/김성훈 18011529/김효경
		작성일자	2020. 6 . 15 .

구 분	요 구 사 항	상 세 내 용	우 선 순 위
1	관리자/사용자를 구분하여 사용할 수 있다.	-프로그램 시작 시 관리자로 시작 -화면에 관리자/사용자 전환이 가능한 버튼 출력 관리자로 사용 중일 때는 버튼에 '관리자' 출력, 사용자로 사용 중일 때는 버튼에 '사용자' 출력	
2	관리자는 캠핑카회사(Camping_Company)를 등록, 삭제, 수정할 수 있다.	-관리자 화면의 'Camping_Company'버튼을 누르면 'Camping_Company'테이블과 입력필드, 입력, 삭제, 변경 버튼 화면출력 -입력 및 변경시, 데이터 타입에 맞는 데이터 입력 -compid 입력시, 테이블에 중복되지 않는 값을 입력 -삭제 및 변경시, 해당 데이터 행(row)를 선택 변경시, 새로 대입한 compid 는 선택한 행의 compid 와 같도록 입력 -입력값이 잘못됐을 경우 오류메세지 출력	

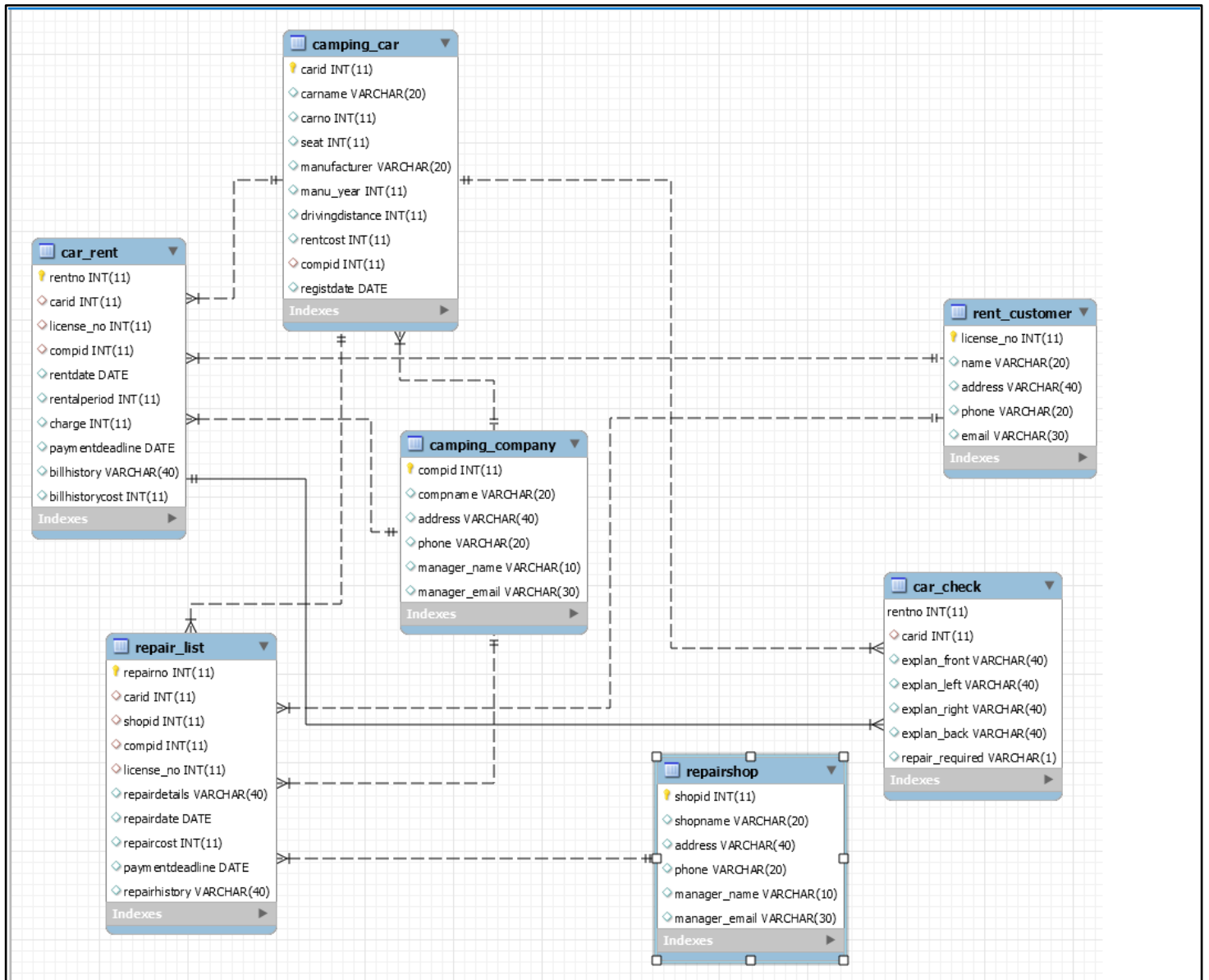
3	관리자는 캠핑카(Camping_Car)를 등록, 삭제, 수정할 수 있다.	<ul style="list-style-type: none"> - 관리자 화면의 'Camping_Car' 버튼을 누르면 'Camping_Car' 테이블과 입력필드, 입력, 삭제, 변경 버튼 화면출력 - 입력 및 변경시, 데이터 타입에 맞는 데이터 입력 - carid 입력시, 테이블에 중복되지 않는 값을 입력 - compid 입력시, Camping_Company 테이블에 존재하는 compid를 입력 - 삭제 및 변경시, 해당 데이터 행(row)를 선택 - 변경시, 새로 대입한 carid는 선택한 행의 carid와 같도록 입력 - 입력값이 잘못됐을 경우 오류메세지 출력 	
4	관리자는 고객(Rent_Customer)을 등록, 삭제, 수정할 수 있다.	<ul style="list-style-type: none"> - 관리자 화면의 'Rent_Customer' 버튼을 누르면 'Rent_Customer' 테이블과 입력필드, 입력, 삭제, 변경 버튼 화면출력 - 입력 및 변경시, 데이터 타입에 맞는 데이터 입력 - license_no 입력시, 테이블에 중복되지 않는 값을 입력 - 삭제 및 변경시, 해당 데이터 행(row)를 선택 - 변경시, 새로 대입한 license_no는 선택한 행의 license_no와 같도록 입력 - 입력값이 잘못됐을 경우 오류메세지 출력 	
5	관리자는 정비소(Repairshop)를 등록, 삭제, 수정할 수 있다.	<ul style="list-style-type: none"> - 관리자 화면의 'Repairshop' 버튼을 누르면 'Repairshop' 테이블과 입력필드, 입력, 삭제, 변경 버튼 화면출력 - 입력 및 변경시, 데이터 타입에 맞는 데이터 입력 - shopid 입력시, 테이블에 중복되지 않는 값을 입력 - 삭제 및 변경시, 해당 데이터 행(row)를 선택 - 변경시, 새로 대입한 shopid는 선택한 행의 shopid와 같도록 입력 - 입력값이 잘못됐을 경우 오류메세지 출력 	
6	관리자는 캠핑카를 반환할 수 있다.	<ul style="list-style-type: none"> - 관리자 화면의 'Rent_Car' 버튼을 누르면 'Car_Rent' 테이블과 입력필드, 반환 버튼 화면출력 - 반환시, 해당 데이터 행(row)를 선택 - 입력값이 잘못됐을 경우 오류메세지 출력 - 반환할 행을 선택하면 그 행의 rentno, carid가 입력필드에 자동으로 들어간다. - 이 때 rentno와 carid는 직접입력해 변경할 수 없다. - repair_required는 Y 또는 N으로 입력한다. 	

		'반환'버튼을 누르면 입력값들이 점검내역(Car_Check)에 insert 된다.	
7	관리자는 수리가 필요한 캠핑카의 정비정보를 입력할 수 있다.	<ul style="list-style-type: none"> - 관리자 화면의 'Car_Check' 버튼을 누르면 'Car_Check' 테이블과 입력필드, 수리요청 버튼 화면출력 -수리요청시, 해당 데이터 행(row)를 선택 -수리요청은 수리필요여부(repair_required)가 Y 일 때만 가능 -입력값이 잘못됐을 경우 오류메세지 출력 <p>수리요청할 행을 선택하면 그 행의 carid, compid, license_no 가 입력필드에 자동으로 들어간다.(license_no 는 rentno 를 이용해 Car_Rent 테이블에서 구한다.) 이 때 carid와 compid, license_no 는 직접입력해 변경할 수 없다. repair_required 가 N 인 행을 수리요청하면 오류메세지가 출력된다. '수리요청'버튼을 누르면 입력값들이 정비정보(Repair_List)에 insert 된다.</p>	
8	관리자는 캠핑카 정비정보를 삭제, 변경 할 수 있다.	<ul style="list-style-type: none"> - 관리자 화면의 'Repair_List' 버튼을 누르면 'Repair_List' 테이블과 입력필드, 삭제, 변경 버튼 화면출력 -삭제 및 변경시, 해당 데이터 행(row)를 선택 -입력값이 잘못됐을 경우 오류메세지 출력 <p>삭제/변경할 행을 선택하면 그 행의 repairno, carid, compid, license_no 가 입력필드에 자동으로 들어간다. 이 때 repairno, carid, compid, license_no 는 직접입력해 변경할 수 없다.</p>	
9	관리자는 반환하지 않은 캠핑카 중 렌탈기간이 지난 차를 검색할 수 있다.	<ul style="list-style-type: none"> - 관리자 화면의 '검색 1' 버튼을 누르면 현재날짜로 설정할 값을 입력하는 창 출력 - 값을 입력하고 'OK'버튼을 누르면 Car_Rent 테이블에서 반환되지 않았으면서 rentdate + rentalperiod 의 값이 입력한 날짜보다 작은 리스트들이 출력 -입력값이 잘못됐을 경우 오류메세지 출력 	
10	관리자는 차 대여로 얻은 수익이 가장 큰 캠핑회사 top10 을 검색할 수 있다.	<ul style="list-style-type: none"> - 관리자 화면의 '검색 2' 버튼을 누르면 rentcost 의 합이 가장 큰 회사 10 개를 내림차순으로 출력 <p>Camping_Company 와 Camping_Car 테이블을 join 하고 각 회사별 sum(rentcost)를 계산해 합이 큰 순서대로 10 개를 출력하도록 한다.</p>	
	관리자는 캠핑카를 반환 할 때 수리가 필요한 경우가 많았던 순으로 고객을 검색할 수 있다.	<ul style="list-style-type: none"> - 관리자 화면의 '검색 3'버튼을 누르면 반환시 점검내역에 repair_required 가 'Y'가 많았던 고객 순으로 고객이름, 운전면허증번호, 전체 대여 횟수, 수리필요 횟수를 출력 	

		Car_Rent, Car_Check, Rent_Customer 테이블을 join 하여 각 손님별 대여 횟수를 계산하고 그 중 repair_required 가 Y 인 경우를 선택 후 출력한다. 출력은 수리필요횟수 기준 내림차순이다.	
1 1	관리자는 정비소를 차 수리로 얻은 수익 순으로 검색할 수 있다.	- 관리자 화면의 '검색 4' 버튼을 누르면 repaircost 의 합이 가장 큰 정비소를 내림차순으로 출력 Repairshop 과 Repair_List 테이블을 join 하여 각 정비소별 sum(repaircost)를 계산해 합이 큰 순서대로 출력하도록 한다	
1 2	관리자는 데이터를 초기화 할 수 있다.	- 관리자 화면의 '초기화' 버튼을 누르면 설정된 데이터베이스로 초기화	
1 3	사용자는 캠핑카를 대여할 수 있다.	- 사용자 화면의 'Camping_Car(대여가능)' 버튼을 누르면 Camping_Car 테이블에서 대여가능한 차 목록과 입력필드, 대여 버튼 화면 출력 -대여시, 해당 데이터 행(row)를 선택 -입력값이 잘못됐을 경우 오류메세지 출력 대여할 행을 선택하면 그 행의 carid, compid 가 입력필드에 자동으로 들어간다. 이 때 carid, compid 는 직접입력해 변경할 수 없다. '대여'버튼을 누르면 입력값들이 캠핑카대여(Car_Rent)에 insert 된다.	
1 4	사용자는 원하는 금액 내의 대여 가능한 캠핑카 리스트를 검색 할 수 있다.	- 사용자 화면의 '검색 1' 버튼을 누르면 원하는 최대 금액 값을 입력하는 창 출력 -입력시, 입력 데이터는 Int 형을 입력 '검색 1' 버튼을 누르고, 최대 가격(max_price)을 입력한다. 최대 가격을 입력한 후, 'OK'를 누를 때, 대여가능한 캠핑카의 목록 중, 가격이 최대 가격보다 적은 차들이 테이블에 나타난다.	
1 5	사용자는 원하는 년도 이후에 만들어진 대여가능한 캠핑카 리스트를 검색할 수 있다.	- 사용자 화면의 '검색 2' 버튼을 누르면 원하는 최소 제조년도 값을 입력하는 창 출력 -입력시, 입력 데이터는 Int 형을 입력 '검색 2' 버튼을 누르고, 최소 연식을 입력한다. 최소 연식을 입력한 후, 'OK'를 누를 때, 대여가능한 캠핑카의 목록 중, 제조년도가 최소 연식보다 큰 차들이 테이블에 나타난다.	
1 6	사용자는 일정거리 이하로 달린 대여가능한 캠핑카 리스트를 검색할 수 있다.	- 사용자 화면의 '검색 3' 버튼을 누르면 원하는 최대 주행거리 값을 입력하는 창 출력	

		<p>-입력시, 입력 데이터는 Int 형을 입력</p> <p>'검색 3' 버튼을 누르고, 최대 주행거리를 입력하다. 최대 주행거리를 입력한 후, 'OK'를 누를 때, 대여가능한 캠핑카의 목록 중, 주행거리가 최대 주행거리보다 적은 차들이 테이블에 나타난다.</p>	
--	--	---	--

2.스키마 정의



3. 프로그램 구조

*프로그램 시작 시 init 함수호출로 GUI 레이아웃 생성

- 패널, 버튼, 텍스트필드, 테이블모델 등 생성 및 크기 조정, ActionListener 부착

*connDB함수호출로 madang DB와 연결

*initDB 함수 호출로 데이터베이스 초기화 및 테이블 생성, 값 insert

- resetDB 함수로 table drop, createTable 함수로 table create

*각 버튼 기능은 actionPerformed 함수에서 정의

- 버튼에 따라 다른 알맞은 데이터를 테이블 모델에 불러옴

- 버튼에 따라 다른 updatePanel(입력 텍스트필드)을 부착

- 버튼에 따라 하단에 다른 buttonPanel(입력/삭제/변경/대여/반환/수리요청 등의 버튼)을 부착

*원하는 행을 선택하여 저장하기 위해 mouseClicked함수 정의

- 보여지고 있는 테이블에 따라 행을 클릭했을 때 하단 입력필드에 대입되는 값이 다름

- 선택한 행 번호를 저장해 삭제/변경 등이 일어났을 때 그에 맞는 데이터가 삭제/변경 되도록 함

*UpdatePanel/ButtonPanel

- 메인 패널의 각 버튼(테이블 이름이 적혀있는 버튼)을 눌렀을 때마다 다른 함수를 호출하여 텍스트필드와 버튼을 다르게 배치

4. 테스트 방법

-관리자 버튼

:Camping_Company

compid 는 int 형식, 나머지(compname, address, phone, manager_num, manager_email)는 String 형식이다.

'입력'시에 compid 는 primary key 이므로

테이블에서 중복되지 않는 값을 입력필드에 넣어준다.

'삭제'시에는, 삭제할 행(row)을 선택(click)하고 삭제버튼을 눌러준다. 또한, '변경'시에도, 변경할 행(row)을 선택하고,

입력필드에 값을 대입하고 변경버튼을 눌러준다.

이때, 선택한 행의 compid 와 대입한 compid 가 같아야 변경이 가능하다.

입력예시)

compid: 16 , compname: 주리네, address: 인천광역시 연수구 동춘동, phone: 010-9382-3543,
manager_name: 이두림, manager_email: duri@naver.com

:Camping_Car

carid, carno, seat, manu_year, drivingdistance, rentcost, compid 에는 int 형식, carname, manufacturer 에는 String 형식, registdate 에는 Date 형식(e x. 2011-3-4)이다.

'입력'시에, carid 는 primary key 이므로

테이블에서 중복되지 않는 값을 입력필드에 넣어준다.

또한 compid 는 foreign key 이므로, Camping_Company 테이블에 존재하는 값을 대입한다.

'삭제'시에는, 삭제할 행(row)을 선택(click)하고 삭제버튼을 눌러준다. 또한, '변경'시에도, 변경할 행(row)을 선택하고,

입력필드에 값을 대입하고 변경버튼을 눌러준다.

이때, 선택한 행의 carid 와 대입한 carid 가 같아야 변경이 가능하다.

입력예시)

carid: 16, carname: 부릉부릉이, carno: 2343, seat: 6, manufacturer: 기아, manu_year: 2011,
drivingdistance: 30000, rentcost: 70000, compid: 3, registdate: 2012-3-4

:Rent_Customer

license_no 는 Int 형식, 나머지(name, address, phone, email)는 String 형식 이다.

‘입력’시에, license_no 는 primary key 이므로

테이블에서 중복되지 않는 값을 입력필드에 넣어준다.

‘삭제’시에는, 삭제할 행(row)을 선택(click)하고 삭제버튼을 눌러준다. 또한, ‘변경’시에도, 변경할 행(row)을 선택하고,

입력필드에 값을 대입하고 변경버튼을 눌러준다.

이때, 선택한 행의 license_no 와 대입한 license_no 가 같아야 변경이 가능 하다.

입력예시)

license_no: 16, name: 김부릉, address: 경기도 분당구, phone: 010-1234-7643, email: booreung@naver.com

: Repairshop

shopid 는 Int 형식, 나머지(shopname, address, phone, manager_name, manager_email)는 String 형식이다.

‘입력’시에, shopid 는 primary key 이므로

테이블에서 중복되지 않는 값을 입력필드에 넣어준다.

‘삭제’시에는, 삭제할 행(row)을 선택(click)하고 삭제버튼을 눌러준다. 또한, ‘변경’시에도, 변경할 행(row)을 선택하고,

입력필드에 값을 대입하고 변경버튼을 눌러준다.

이때, 선택한 행의 shopid 와 대입한 shopid 가 같아야 변경이 가능하다.

입력예시)

shopid: 16, shopname: 우아한브로들, address: 서울특별시 강남구, phone: 02-3433-4634,
manager_name: 준성훈, manager_email: junes@naver.com

: Rent_Car

rentno, carid 에는 Int 형식, 나머지(explain_front, explain_left, explain_right, explain_back, repair_required)에는 String 형식이다.

반환할 행을 선택하면 그 행의 rentno, carid 가 입력필드에 자동으로 들어간다.

이 때 rentno와 carid는 직접입력해 변경할 수 없다. repair_required는 Y또는 N으로 입력한다.

'반환'버튼을 누르면 입력값들이 점검내역(Car_Check)에 insert된다.

입력예시)

(Car_Check에 이미 rentno가 16번까지 채워져있으므로, 즉 rentno가 16번 이하의 차는 이미 반환 되어있으므로 16번 이하의 차를 반환을 시도할 경우, 에러가 발생)

(rentno **17번 이상**의 데이터 클릭 후)

rentno: (클릭시 자동) , carid: (클릭시 자동), explain_front: X, explain_left: X, explain_right: X, explain_back: X, repair_required: N

: Car_Check

repairno, carid, shopid, compid, license_no, repaircost, 에는 Int 형식, repairedetails, repairhistory 는

String 형식, repairedate, paymentdeadline 은 Date 형식(ex. 2011-3-4)이다.

수리요청시에, repairno 는 primary key 이므로

테이블에서 중복되지 않는 값을 입력필드에 넣어준다.

수리요청할 행을 선택하면 그 행의 carid, compid, license_no 가 입력필드에 자동으로 들어간다.(license_no는 rentno를 이용해 Car_Rent테이블에서 구한다.)

이 때 carid 와 compid, license_no 는 직접입력해 변경할 수 없다. repair_required가 N인 행을 수리요청하면 오류메세지가 출력된다. '수리요청'버튼을 누르면 입력값들이 정비정보(Repair_List)에 insert된다.

입력 예시)

(Repair_requested가 Y인 데이터 클릭 후)

repairno: 17, carid: (클릭시 자동), shopid: 3, compid: (클릭시 자동), license_no: (클릭시 자동),
repairdetails: bumper change, repairedate: 2017-3-4, repaircost: 1000000, paymentdeadline: 2017-4-6, repairhistory: Nothing

: Repair_List

repairno, carid, shopid, compid, license_no, repaircost, 에는 Int 형식, repairdetails, repairhistory 는 String 형식, repairedate, paymentdeadline 은 Date 형식(ex. 2011-3-4)이다.

삭제/변경할 행을 선택하면 그 행의 repairno, carid, compid, license_no가 입력필드에 자동으로 들어간다.

이 때 repairno, carid, compid, license_no 는 직접입력해 변경할 수 없다.

입력예시

(변경시, 데이터 클릭 후)

repairno: (클릭시 자동), carid: (클릭시 자동), shopid: (클릭시 자동), compid: (클릭시 자동),
license_no: (클릭시 자동), repairdetails: Bumper change, repairedate: 2020-3-4, repaircost: 1200000,

paymentdeadline: 2020-4-5, repairhistory: Nothing

: 검색1

입력값은 Date형식(yyyy-mm-dd)으로 입력한다.

이때 현재 존재하는 테이블에서 rentdate 는 2019~2020 년도의 날짜이므로 2019 년도의 날짜를 입력했을 때 가장 값 확인이 쉽다.

입력예시)

(검색1 버튼을 누른 후)

2019-12-20

-사용자 버튼

: Camping_Car(대여가능)

rentno, carid, license_no, compid, retalperiod, charge, billhistorycost 에는 Int 형식, billhistory 는 String 형식, rent_date, paymentdeadline 은 Date 형식(ex. 2011-3-4)이다.

대여할 행을 선택하면 그 행의 carid, compid 가 입력필드에 자동으로 들어간다.

이 때 carid, compid 는 직접입력해 변경할 수 없다.

'대여'버튼을 누르면 입력값들이 캠핑카대여(Car_Rent)에 insert된다.

이때 rentno는 Car_Rent에 존재하는 데이터의 rentno와 중복되어서는 안된다.

입력예시)

(대여할 데이터 선택 후)

rentno: 22, carid: (클릭시 자동), license_no: 3, compid: (클릭시 자동), rent_date: 2020-6-12, rentalperiod: 4, charge: 200000, paymentdeadline: 2020-6-30, billhistory: X, billhistorycost: 0

: 검색1

'검색 1' 버튼을 누르고, 최대 가격(max_price)을 입력한다.

이때, 최대 가격은 Int 형식이다.

최대 가격을 입력한 후, 'OK'를 누를 때,

대여가능한 캠핑카의 목록 중, 가격이 최대 가격보다 적은 차들이 테이블에 나타난다.

이때 현재 존재하는 테이블에서 최소가격은 60000 이므로 60000 이상을 입력해야 데이터들을 확인할 수 있다.

입력예시)

(검색1 클릭 후)

80000

: 검색2

'검색 2' 버튼을 누르고, 최소 연식을 입력한다.

이때 최소 연식은 Int 형식이다.

최소 연식을 입력한 후, 'OK'를 누를 때,

대여가능한 캠핑카의 목록 중, 제조연도가 최소 연식보다 큰 차들이 테이블에 나타난다.

이때 현재 존재하는 테이블에서 최대 연식은 2016 이므로, 2016 이하를 입력해야 데이터들을 확

인 할 수 있다.

입력예시)

(검색2 클릭 후)

2011

: 검색3

'검색 3' 버튼을 누르고, 최대 주행거리를 입력하다.

이때 최대 주행거리는 Int 형식이다.

최대 주행거리를 입력한 후, 'OK'를 누를 때,

대여가능한 캠핑카의 목록 중, 주행거리가 최대 주행거리보다 적은 차들이 테이블에 나타난다.

이때 현재 존재하는 테이블에서 최소 주행거리는 3000 이므로, 3000 이하를 입력해야 데이터들을 확인 할 수 있다.

입력 예시)

(검색3 클릭 후)

11000