
4 Team PROJECT. HoCar

4 TEAM MEMBER : 민경환, 김동진, 김지환, 소강석, 김주원

Presenter

Kim Ji Hwan



HOCAR CONTENTS

- 프로젝트 개요
- 프로그램 설계
- 주요 코드 시연 (JAVA)
- 프로젝트 진행 후기



프로젝트 개요



- 프로젝트 수행목적

- JAVA-DataBase 및 **CRUD** 연동 프로그램 개발
- **MVC** 모델 이해 및 구현



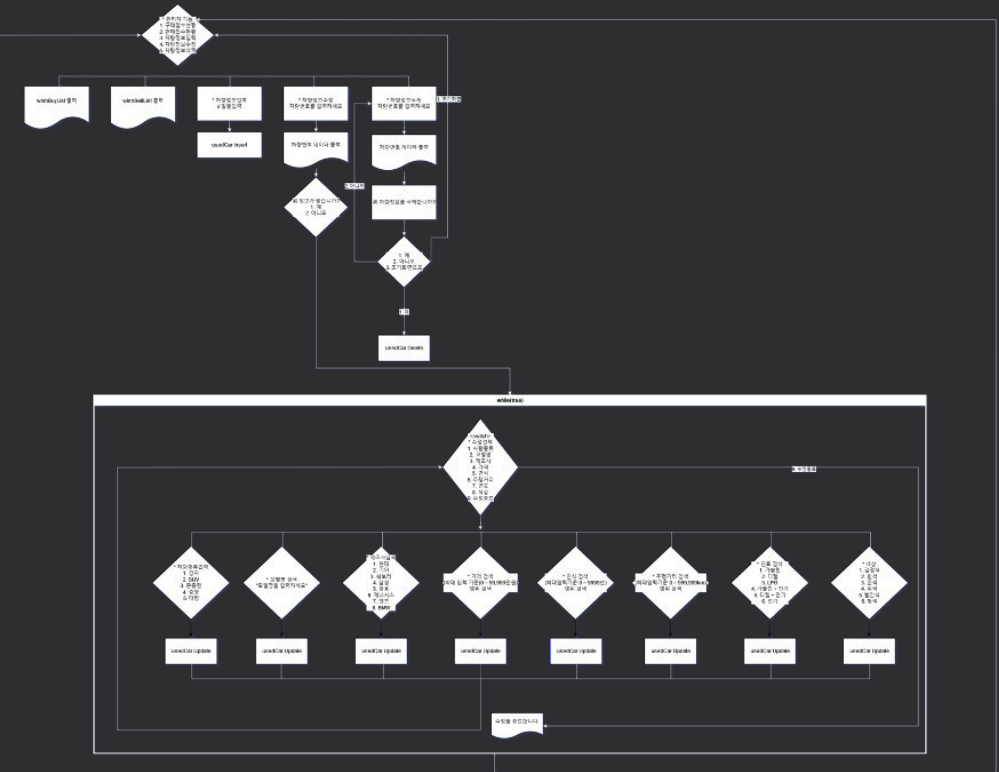
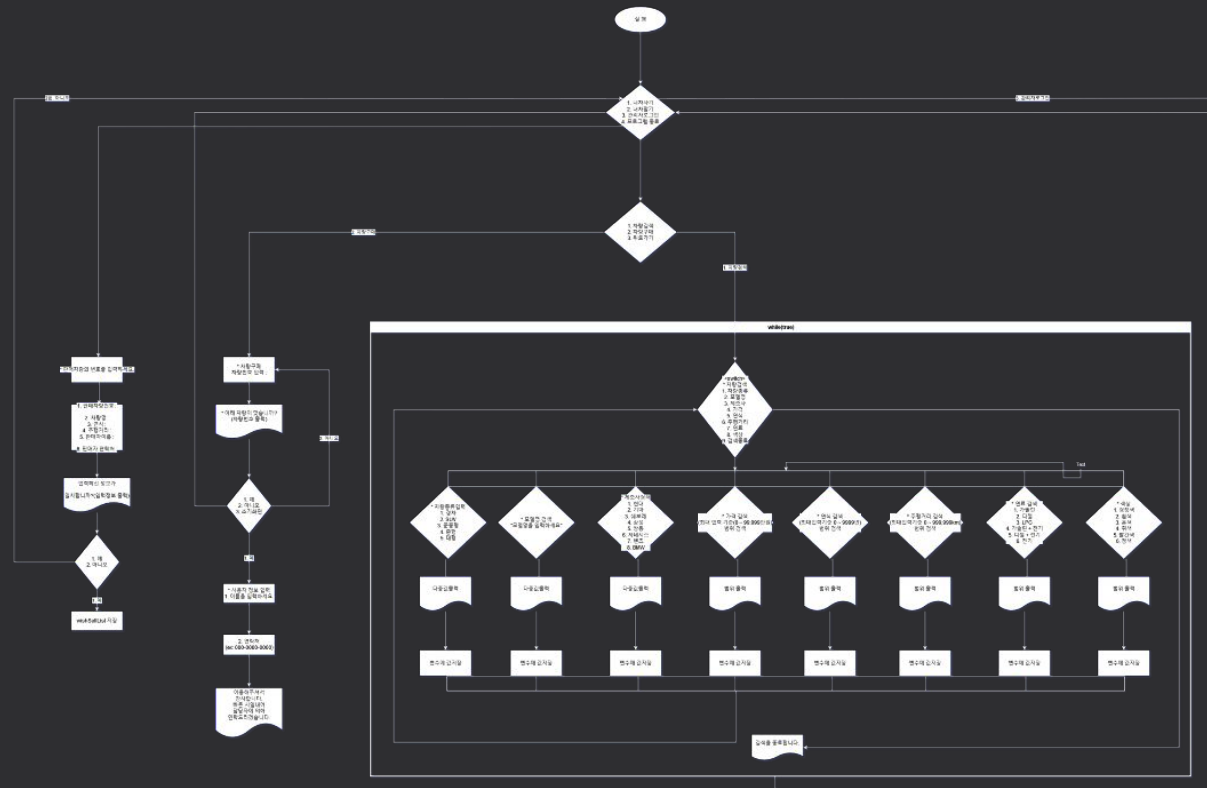
- 프로젝트 개발목표

- 사용자입장
 - 다중조건으로 차량을 검색하고, 구매희망차량에 대해서는 주문접수, 판매희망차량은 판매접수로 서비스 이용이 가능해야한다.
- 관리자입장
 - 차량데이터가 입력된 DB서버를 통해 사용자에게 명확하고 정확한 데이터 조회를 제공한다.
 - 주문 및 판매접수 현황을 조회하고, 저장된 데이터를 통해 현장직원에게 안내가 가능해야한다.

프로젝트 개요

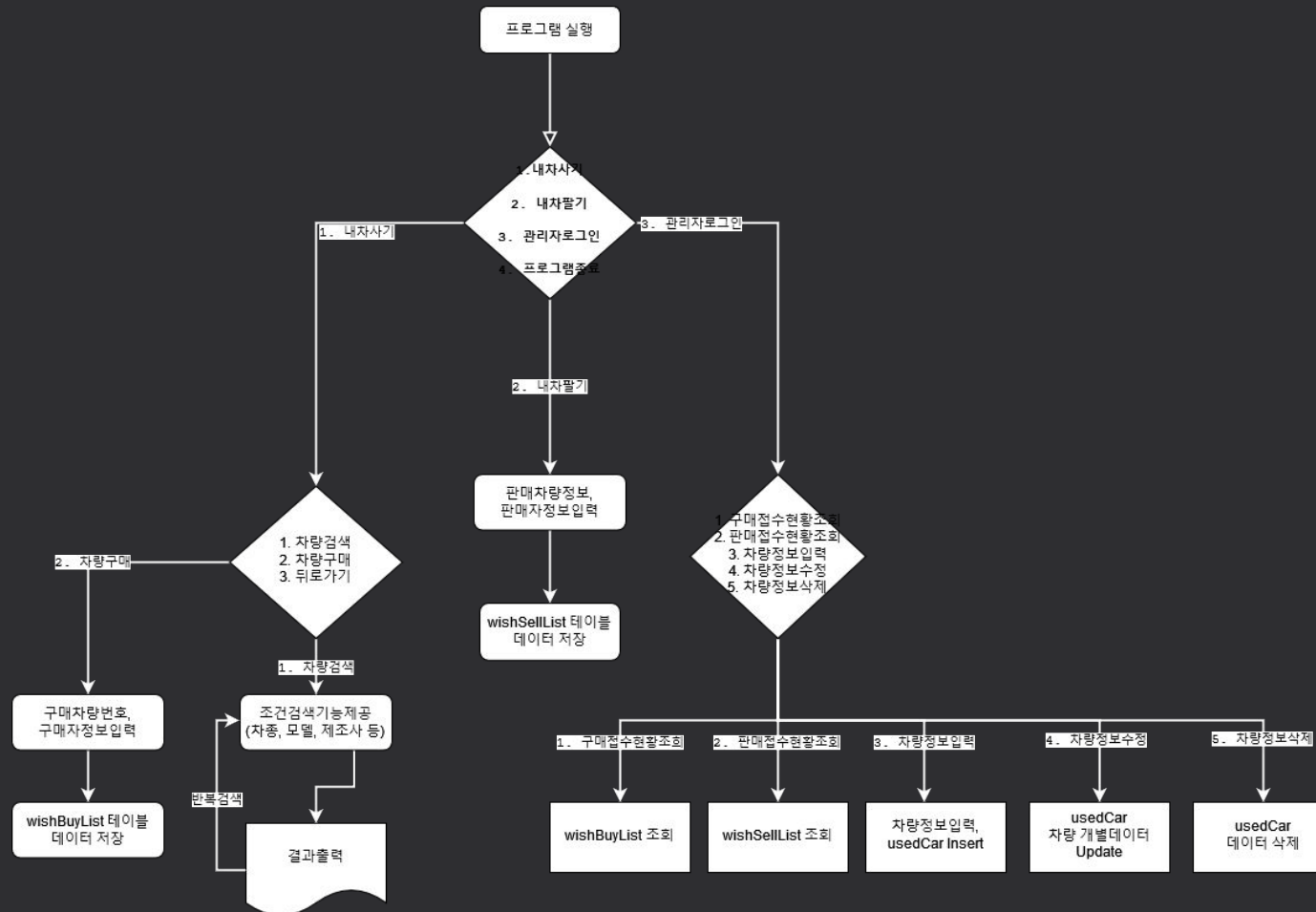
순서도

Fri Oct 22 2021 15:30:38 [4팀 자바프로젝트 (중고차 사이트 호카)] // 민경환 김지환 김동진 소강석 김주원 (93)



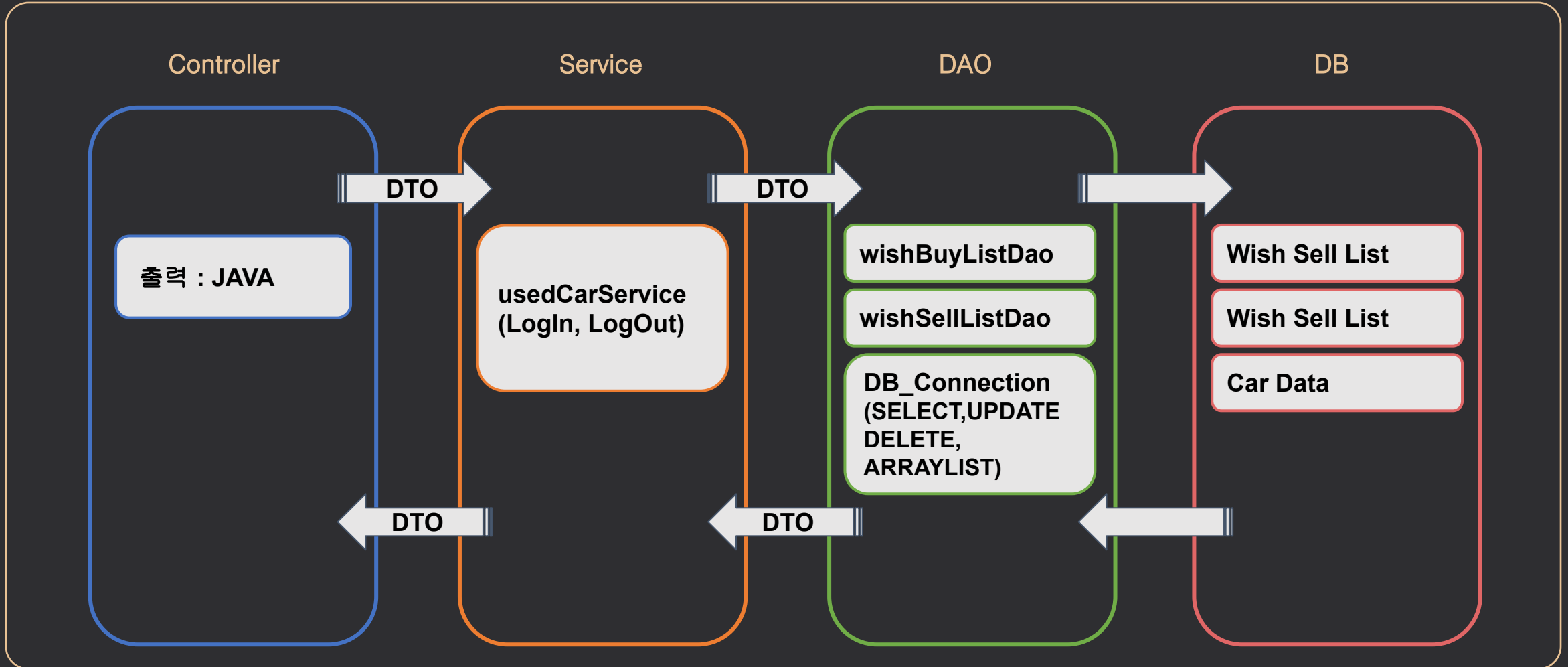
프로젝트 구성도

구성도



프로젝트 개요

MVC 패턴



프로그램 설계 (1)

사용자 시스템 설계

구매

```
1 // 메인에서 if문으로 선택시 구매 문구 출력
2 else if (choice == 2) {
3
4     System.out.print("에시)57부1234 \n");
5     System.out.print("구매하려는 차량 번호를 입력하세요: \n");
6     car_number=sc.next();
7     db.SearchCar(car_number);
8     System.out.print("\n");
9     System.out.print("구매하시려는 차량 정보가 맞습니까?");
10    System.out.print("1. 예 / 2. 아니오");
11
12    choice = sc.nextInt();
13    if(choice == 1){
14        System.out.print("이름: \n");
15        wishbuyuser_name=sc.next();
16        System.out.print("에시)010-0000-0000 \n");
17        System.out.print("전화번호: \n");
18        phone_num=sc.next();
19        db.insertwishBuyList(car_number,wishbuyuser_name,phone_num);
20
21        System.out.print("이용해 주셔서 감사합니다. 빠른 시일 내에 담당자가 확인 후 연락드리겠습니다
22 \n");
23        break;
24    }
25    else if(choice == 2){
26        menuWhile();
27    }
28 }
29 // 입력한 값이 wishBuyList로 이동
30
31 public void insertwishBuyList(String car_number, String wishbuyuser_name, String phone_num) {
32
33     String SQL = "Insert INTO wishBuyList(car_number,wishbuyuser_name,phone_num)VALUES(?,?,?)";
34
35     try {
36         pstmt = con.prepareStatement(SQL);
37         pstmt.setString(1, car_number);
38         pstmt.setString(2, wishbuyuser_name);
39         pstmt.setString(3, phone_num);
40         pstmt.executeUpdate();
41     } catch (SQLException e) {
42         e.printStackTrace();
43     }
44 }
```

- (6), (14), (17) 행 각 데이터 입력
- (29)행 데이터 > 테이블 입력

프로그램 설계 (2)

사용자 시스템 설계

판매

```
1 else if (choice == 2) {
2     System.out.print("판매차량의 정보를 입력하세요\n");
3     System.out.print("예시) 26주3441\n");
4     System.out.print("판매차량 번호: \n");
5     car_number=sc.next();
6     System.out.print("예시) 쏘나타, 아반떼 ... \n");
7     System.out.print("차량명: \n");
8     car_name=sc.next();
9     System.out.print("연식: \n");
10    made_year=sc.nextDouble();
11    System.out.print("예시) 10000(0) 1만km(X) 숫자만 입력하세요 \n");
12    System.out.print("주행거리: \n");
13    dis_driven=sc.nextInt();
14    System.out.print("판매자 이름: \n");
15    wishSellUser_name=sc.next();
16    System.out.print("예시) 010-0000-0000 \n");
17    System.out.print("판매자 연락처: \n");
18    phone_num=sc.next();
19
20    System.out.print("입력하신 정보가 일치할까요? \n");
21    System.out.print("1. 예 2. 아니요\n");
22    System.out.print("===== \n");
23    System.out.println(car_number);
24    System.out.println(car_name);
25    System.out.println(made_year+"년");
26    System.out.println(dis_driven+"km");
27    System.out.println(wishSellUser_name);
28    System.out.println(phone_num);
29    System.out.print("===== \n");
30    choice = sc.nextInt();
31    if(choice == 1){
32        db.insertwishSellList(car_number, car_name, made_year, dis_driven, wishSellUser_name, phone_num);
33        System.out.print("이동에 주셔서 감사합니다. 빠른 시일 내에 담당자가 확인 후 연락드리겠습니다\n");
34        break;
35    }
36    }
37    else if(choice == 2){
38        menuWhlie();
39    }
40    }
41    public void insertwishSellList(String car_number, String car_name, Double made_year, int dis_driven, String wishSellUser_name, String
42    phone_num){
43        String SQL = "Insert INTO wishSellList(car_number,car_name,made_year,dis_driven,wishSellUser_name,phone_num)VALUES(?,?,?,?,?,?)";
44        try {
45            pstmt = con.prepareStatement(SQL);
46            pstmt.setString(1, car_number);
47            pstmt.setString(2, car_name);
48            pstmt.setDouble(3, made_year);
49            pstmt.setInt(4, dis_driven);
50            pstmt.setString(5, wishSellUser_name);
51            pstmt.setString(6, phone_num);
52            pstmt.executeUpdate();
53        } catch (SQLException e) {
54            e.printStackTrace();
55        }
56    }
```

- (5), (8), (10), (13), (15), (18) 행
각 데이터 입력
- (42) 행
데이터 > 테이블
입력

프로그램 설계 (3)

사용자 시스템 설계

조회(1)

```
1 public void searchByCarType(String type) {
2     String SQL = "SELECT * FROM usedCar WHERE Car_type= " + "'" + type + "'";
3     try {
4         rs = st.executeQuery(SQL);
5         while (rs.next()) {
6             String car_number = rs.getString("car_number");
7             String car_type = rs.getString("car_type");
8             String car_name = rs.getString("car_name");
9             String made_company = rs.getString("made_company");
10            int price = rs.getInt("price");
11            double made_year = rs.getDouble("made_year");
12            int dis_driven = rs.getInt("dis_driven");
13            String fuel_type = rs.getString("fuel_type");
14            String color = rs.getString("color");
15
16            usedCarVo list = new usedCarVo(); // list 객체 생성
17            list.setcar_number(rs.getString("car_number"));
18            list.setcar_type(rs.getString("car_type"));
19            list.setcar_name(rs.getString("car_name"));
20            list.setmade_company(rs.getString("made_company"));
21            list.setprice(rs.getInt("price"));
22            list.setmade_year(rs.getDouble("made_year"));
23            list.setdis_driven(rs.getInt("dis_driven"));
24            list.setfuel_type(rs.getString("fuel_type"));
25            list.setcolor(rs.getString("color"));
26            arrayListCar.add(list); // list 객체를 이용해 ArrayList형식인 arrayListCar에 저장
```

- (16) list 객체생성
- (26) list 이용 ArrayList 형식인 arrayListCar에 검색 데이터 저장

프로그램 설계 (3)

사용자 시스템 설계

조회(2)

```
1 public void listByCarCompany(String type) {
2     for (int i = (arrayListCar.size()-1); i > -1; i--) {
3         String word = arrayListCar.get(i).getmade_company();
4         if (word.contains(type)) {
5             double j = arrayListCar.get(i).getmade_year();
6             System.out.printf("차량번호 : " + arrayListCar.get(i).getcar_number() + " / ");
7             System.out.printf("차량종류 : " + arrayListCar.get(i).getcar_type() + " / ");
8             System.out.printf("모델명 : " + arrayListCar.get(i).getcar_name() + " / ");
9             System.out.printf("제조사 : " + arrayListCar.get(i).getmade_company() + " / ");
10            System.out.printf("가격 : " + DRIVEN.format(arrayListCar.get(i).getprice()) + "만원 / ");
11            System.out.printf("연식 : %.0f년형 / ", j);
12            System.out.printf("주행거리 : " + DRIVEN.format(arrayListCar.get(i).getdis_driven());
13            System.out.printf("연료 : " + arrayListCar.get(i).getfuel_type() + " / ");
14            System.out.println("색상 : " + arrayListCar.get(i).getcolor());
15        } else {
16            arrayListCar.remove(i);
17        }
18    }
19    if(arrayListCar.isEmpty()) {
20        System.out.println("조회 결과가 없습니다. 검색종료 후 재검색하세요.");
21    }
22    System.out.println(" ");
23 }
```

- (1) ArrayList 형식 Loop문 내에 remove 메서드 이용시 오류발생가능
☞ 역순으로 검색시 오류 미발생
- (4) contains 메서드를 활용 String word에 저장된 값과 일치여부확인
☞ 일치 시 출력, 불일치 시 remove
- (19) isEmpty 메서드 활용
arrayListCar 내에 데이터가 없을 시
“조회결과가 없습니다.” 출력
- arrayListCar에 db서버에서 받아온
데이터가 저장되고, 이후
arrayListCar에 저장된 값에 검색
메서드를 활용해서 출력

프로그램 설계 (4)

관리자 시스템 설계

조회

```
1 public void SearchByNo(String input_car_number) {
2     String SQL = "SELECT * FROM usedCar WHERE Car_Number=" + "'" + input_car_number + "'";
3     try {
4         rs = st.executeQuery(SQL);
5
6         while (rs.next()) {
7
8             String car_number = rs.getString("Car_Number");
9             String car_type = rs.getString("Car_Type");
10            String car_name = rs.getString("Car_Name");
11            String made_company = rs.getString("Made_Company");
12            int price = rs.getInt("Price");
13            double made_year = rs.getDouble("Made_Year");
14            int dis_driven = rs.getInt("Dis_Driven");
15            String fuel_type = rs.getString("Fuel_Type");
16            String color = rs.getString("Color");
17
18            System.out.printf("차량번호 : %s / ", car_number);
19            System.out.printf("차량종류 : %s / ", car_type);
20            System.out.printf("모델명 : %s / ", car_name);
21            System.out.printf("제조사 : %s / ", made_company);
22            System.out.printf("가격 : %d만원 / ", price);
23            System.out.printf("연식 : %.0f년형 / ", made_year);
24            System.out.printf("주행거리 : %dkm / ", dis_driven);
25            System.out.printf("연료 : %s / ", fuel_type);
26            System.out.printf("색상 : %s \n", color);
27
28        }
29    } catch (SQLException e) {
30        e.printStackTrace();
31    }
32 }
```

execute

관리자 기능

1. 구매접수현황
2. 판매접수현황
3. 차량정보입력
4. 차량정보수정
5. 차량정보삭제
6. 최초화면으로 이동

4

차량 번호 입력: (출력값이 없으면 잘못된 입력값입니다.)

54나9016

차량번호 : 54나9016 / 차량종류 : SUV / 모델명 : 2시리즈 스포츠 투어링 / 제조사 : BMW /
가격 : 2470만원 / 연식 : 2019년형 / 주행거리 : 34828km / 연료 : 디젤 / 색상 : 은색

- (2) SQL 조건 검색 실행
- (6~26) 조건 검색 실행 데이터 출력
- 이후 판단된 데이터를 기준으로 데이터 수정을 진행

프로그램 설계 (5)

관리자 시스템 설계

수정



```
1 public void UpdateusedCarbyCartype(String car_number, String car_type) {
2     String SQL = "update usedCar set Car_Type=? where Car_Number=?";
3
4     try {
5         pstmt = con.prepareStatement(SQL);
6         pstmt.setString(1, car_type);
7         pstmt.setString(2, car_number);
8         pstmt.executeUpdate();
9     } catch (SQLException e) {
10        e.printStackTrace();
11    }
12 }
13
14 public void UpdateusedCarbyCarname(String car_number, String car_name) {
15     String SQL = "update usedCar set Car_Name=? where Car_Number=?";
16
17     try {
18         pstmt = con.prepareStatement(SQL);
19         pstmt.setString(1, car_name);
20         pstmt.setString(2, car_number);
21         pstmt.executeUpdate();
22     } catch (SQLException e) {
23        e.printStackTrace();
24    }
25 }
```

execute

54나9016

차량번호 : 54나9016 / 차량종류 : SUV / 모델명 : 2시리즈 스포츠 투어링 / 제조사 : BMW /
가격 : 2470만원 / 연식 : 2019년형 / 주행거리 : 34828km / 연료 : 디젤 / 색상 : 은색
위 정보가 맞습니까?

1. 네

2. 아니오

1

*수정할 항목을 선택하세요(번호를 입력)

1. 차량 종류

2. 모델명

3. 제조사

4. 가격

5. 연식

6. 주행거리

7. 연료 종류

8. 색상

9. 수정 종료

1

*수정할 종류 선택(번호를 입력)

1. 경차

2. SUV

3. 준중형

4. 중형

5. 대형

4

다음과 같이 수정되었습니다.

차량번호 : 54나9016 / 차량종류 : 중형 / 모델명 : 2시리즈 스포츠 투어링 / 제조사 : BMW /
가격 : 2470만원 / 연식 : 2019년형 / 주행거리 : 34828km / 연료 : 디젤 / 색상 : 은색

프로그램 설계 (6)

관리자 시스템 설계

입력/삭제

```
1 public void InsertusedCar(String car_number, String car_type, String car_name, String made_company, int price,
2   double made_year, int dis_driven, String fuel_type, String color) {
3   String SQL = "Insert INTO
4   usedCar(Car_Number,Car_Type,Car_Name,Made_Company,Price,Made_Year,Dis_Driven,Fuel_Type,Color)"
5   + "VALUES (?,?,,?,,?,,?,,?)";
6
7   try {
8     pstmt = con.prepareStatement(SQL);
9     pstmt.setString(1, car_number);
10    pstmt.setString(2, car_type);
11    pstmt.setString(3, car_name);
12    pstmt.setString(4, made_company);
13    pstmt.setInt(5, price);
14    pstmt.setDouble(6, made_year);
15    pstmt.setInt(7, dis_driven);
16    pstmt.setString(8, fuel_type);
17    pstmt.setString(9, color);
18    pstmt.executeUpdate();
19  } catch (SQLException e) {
20    e.printStackTrace();
21  }
```

execute

차량 번호 입력:12아1134
차량 종류 입력:
경차
차량 모델명 입력:소나타1시리즈
제조사 입력:현대
가격 입력(단위 만원):1530
연식 입력(년도 입력):2016
주행거리 입력(KM단위 입력):60
연료 종류 입력:가솔린
색상 입력:흰색
정보 입력이 완료되었습니다.

- SQL Create기능 구현
- SQL Delete기능 구현

```
1 public void DeleteusedCar(String car_number) {
2   String SQL = "delete from usedCar where Car_Number=?";
3
4   try {
5     pstmt = con.prepareStatement(SQL);
6     pstmt.setString(1, car_number);
7     pstmt.executeUpdate();
8   } catch (SQLException e) {
9     e.printStackTrace();
10  }
11 }
```

execute

차량 번호 입력:12아1134
차량번호 : 12아1134 / 차량종류 : 경차 / 모델명 : 소나타1시리즈 / 제조사 : 현대 /
가격 : 1530만원 / 연식 : 2016년형 / 주행거리 : 60km / 연료 : 가솔린 / 색상 : 흰색
위의 차량 정보를 삭제하겠습니까?
(출력값이 없으면 잘못된 입력값입니다.)
1. 네
2. 아니요
3. 초기화면으로

프로젝트 진행 후기

만족스러운 부분



- **CRUD** 구현이 잘 되었다.
- 사용자 친화적인 구조 구현
- 순서도를 통해 구현 목표 공유, 비즈니스 모델 정립
- 사용자와 관리자의 메뉴셋 구분

부족했던 부분



- **DB** 공유서버 연동 에러
- 조건 검색 시 서브 쿼리 활용 미흡
- 검색, 수정 모듈화 미흡
- service 클래스 활용 미흡

감사합니다.

4 TEAM MEMBER : 김동진, 김지환, 소강석, 김주원, 민경환

Presenter

Kim Ji Hwan

