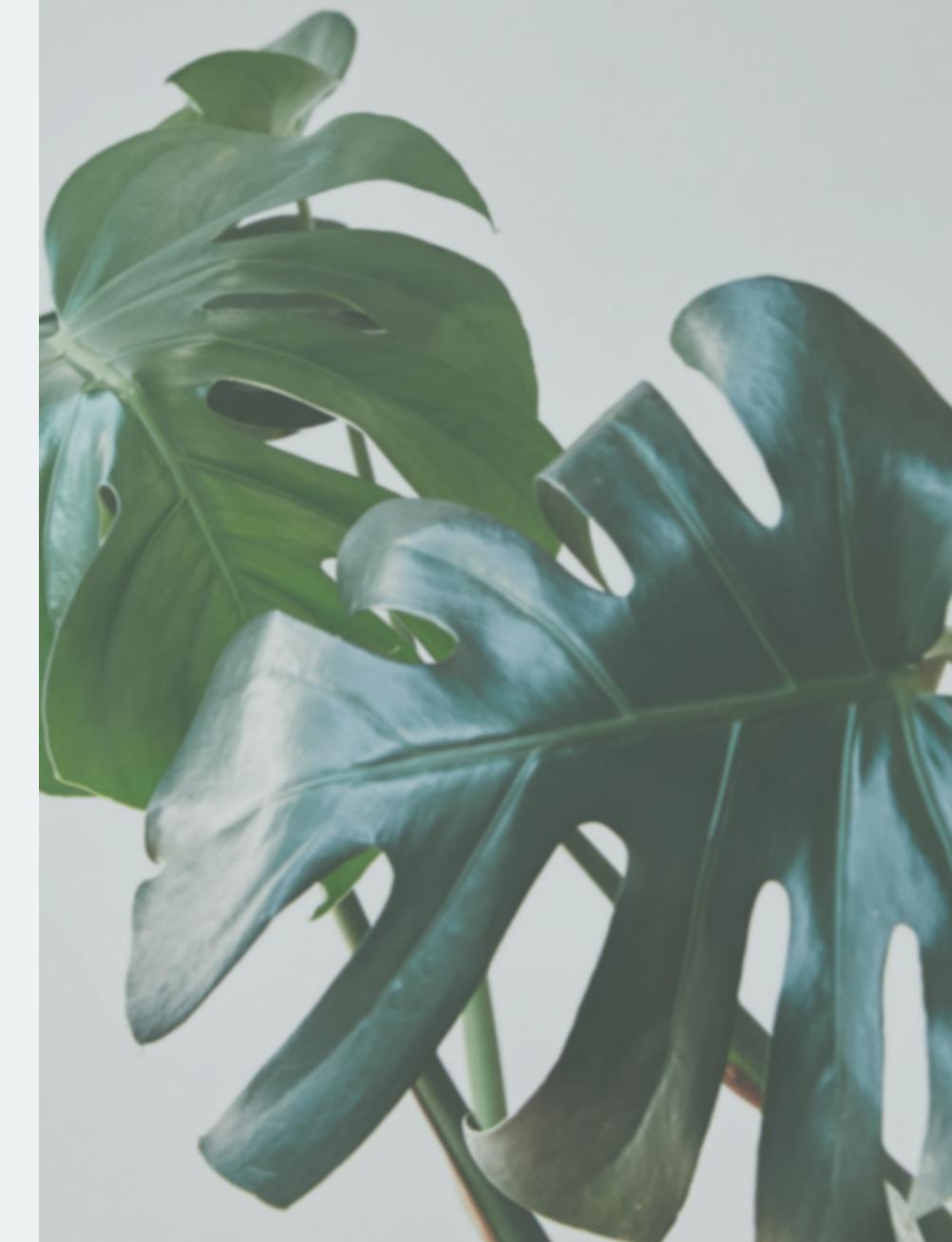


시간 지나면
견인하조

주차장 프로그램

- 김시우
- 문기연
- 박종휘
- 김강영
- 박지윤



□ 프로젝트 기간

Timeline

프로젝트 구상

2022.05.12 ~ 05.17

- 주제 선정 및 파트 분배
- 각 파트 설계

프로젝트 진행

2022.05.18 ~ 06.05

- 각 파트 코딩 및 취합
- 오류 수정
- 회의 및 테스트 시연

5월						
일	월	화	수	목	금	토
1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30	31	1	2	3	4

← 시작일

→ 마감일 06.05



01

목차

- 프로젝트 설명
- 프로젝트 시연
- 프로젝트 소감 및 QNA



□ 01 프로젝트 설명

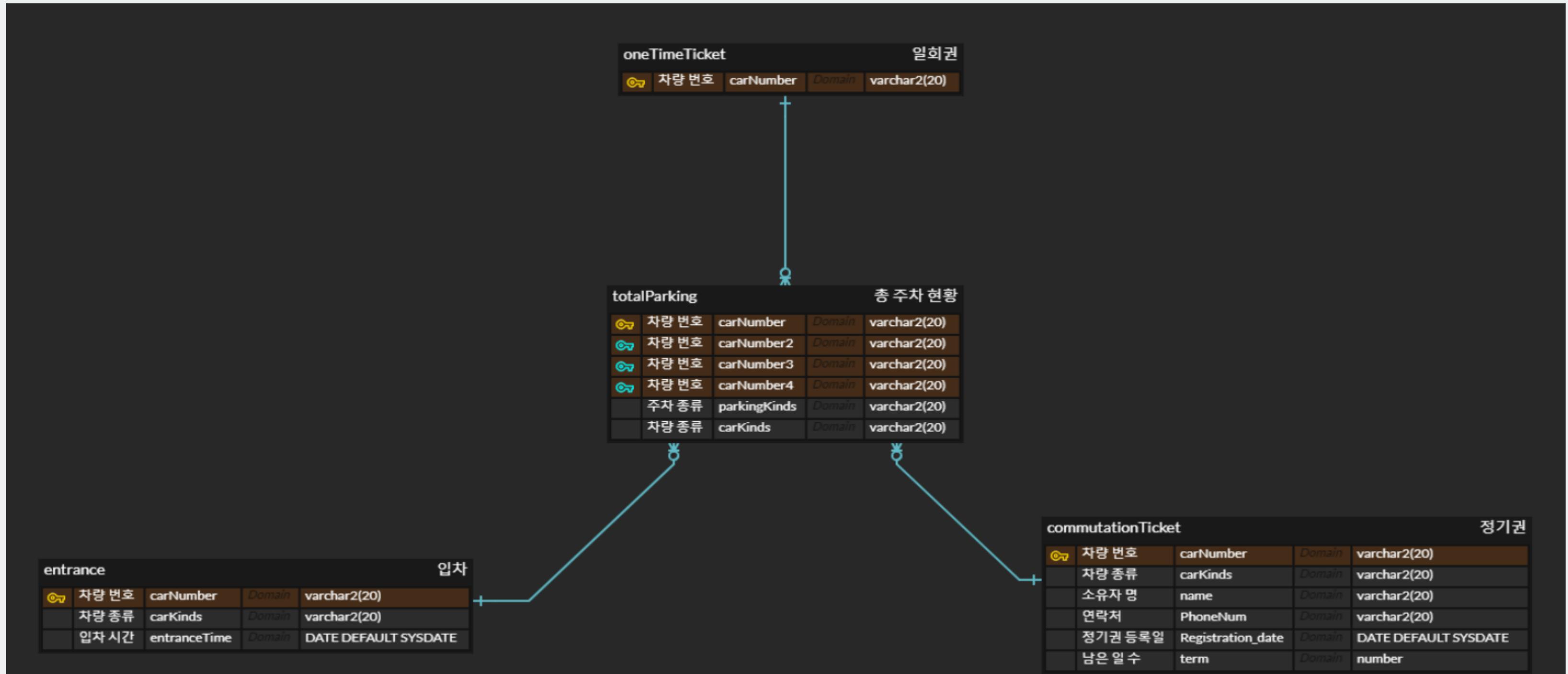
프로젝트 설명

시간이 지나면 견인하는 주차장 프로그램

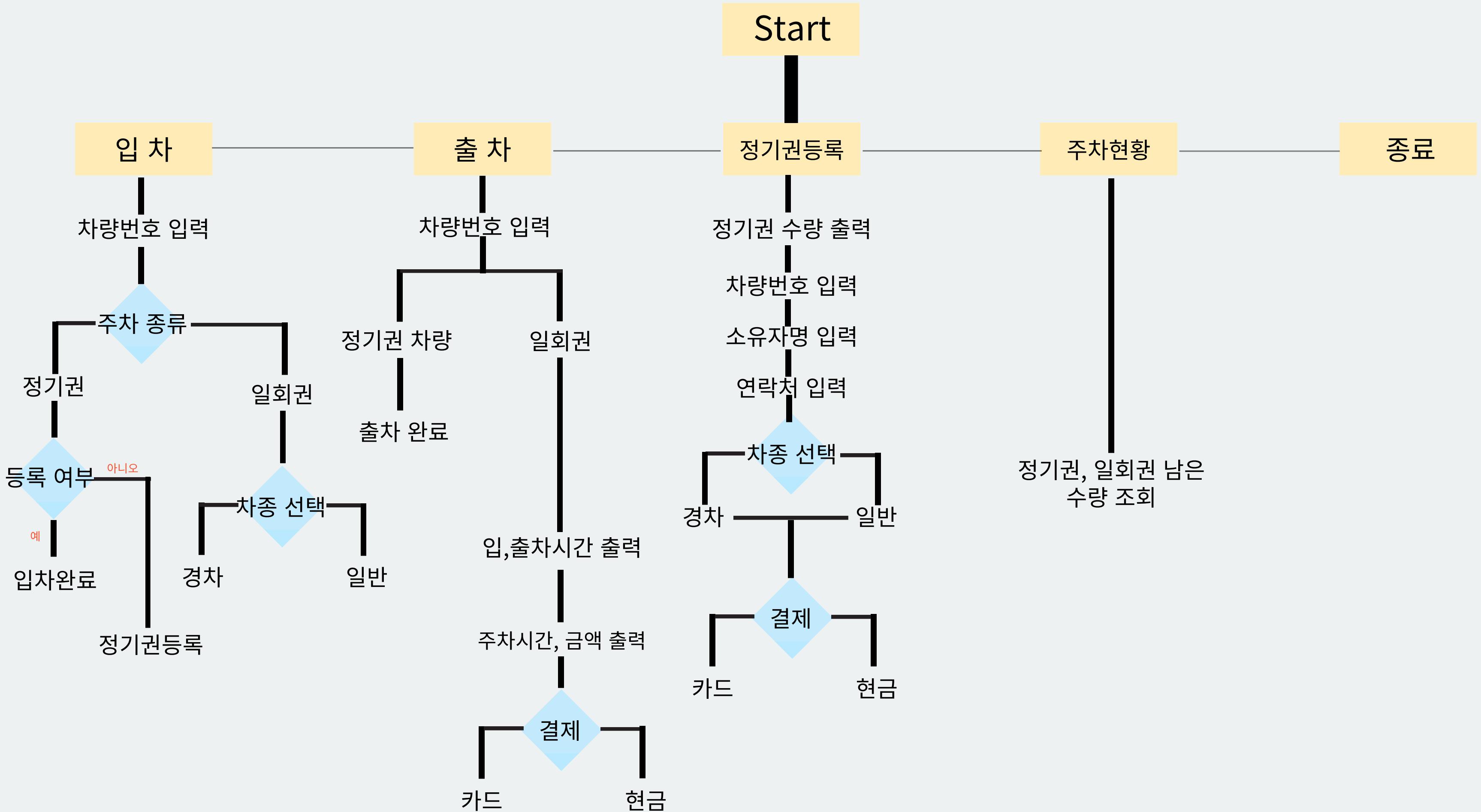
메인메뉴로 입차, 출차, 정기권 등록, 주차현황, 종료를 구현하여,
각각의 클래스를 만들어주어 실행이 가능하도록 프로그래밍

□ 프로젝트 설명

ERD



□ User Logic



● 통합개발Tool
Eclipse, DBeaver



● DBMS
Oracle



● Language
Java



● 소통
Discord, 카카오톡



□ 프로그램 설명

코드리뷰

Inout.java

주차장 입차와 출차를 하는 클래스

□ 프로그램 설명

```
public class Inout {
    static ParkingDTO dto = new ParkingDTO();
    static TotalParking tp = new TotalParking();
    public static void Enterance() throws AuthenException{
        Scanner sc = new Scanner(System.in);
        LocalTime now = LocalTime.now();
        DateTimeFormatter formatter = DateTimeFormatter.ofPattern("HH:mm:ss");
        String formatedNow = now.format(formatter);
        DAO dao = new DAO();
        while (true) {
            System.out.println("차량 번호를 입력하세요");
            dto.setCarNumber(sc.nextLine());
            boolean isNum = Pattern.matches("^([0-9가-힣]{3,4})\\s?[0-9]{4}", dto.getCarNumber());
            if (!isNum) {
                System.out.println("등록되지 않은 차량입니다.");
            } else {
                System.out.println("1. 정기권(20만원)" + "\t" + "2. 일회권(30분에 5000원, 이후 10분 단위로 1000원씩 부과됩니다)");
                String num = sc.next();
                switch (num) {
                    case "정기권":
                        if(isNum || dao.selectComm().contains(dto.getCarNumber())) {
                            System.out.println("등록된 차량입니다.");
                            System.out.println("입차완료");
                            dao.insertEntrance(dto.getCarNumber(), dto.getCarKinds(), dto.getEntranceTime());
                        }
                        break;
                    case "일회권":
                        System.out.println("차종을 입력해주세요(경차\ 일반)");
                        dto.setCarKinds(sc.next());
                        if (dto.getCarKinds().equals("경차")) {
                            dto.setEntranceTime(now.format(formatter));
                        } else if (dto.getCarKinds().equals("일반")) {
                            dto.setEntranceTime(now.format(formatter));
                        } else {
                            return;
                        }
                        dao.insertEntrance(dto.getCarNumber(), dto.getCarKinds(), dto.getEntranceTime());
                        dao.insertOneTime(dto.getCarNumber());
                        return;
                }
            }
        }
    }
}
```

```
public static void Exit() throws AuthenException{
    Scanner sc = new Scanner(System.in);
    CommutationTicket ct = new CommutationTicket();
    DAO dao = new DAO();
    while (true) {
        System.out.println("출차를 진행합니다.");
        System.out.println("차량 번호를 입력하세요");
        dto.setCarNumber(sc.next());
        boolean isNum = Pattern.matches("^([0-9가-힣]{3,4})\\s?[0-9]{4}", dto.getCarNumber());
        if (!isNum) {
            System.out.println("잘못된 차량번호입니다.");
            continue;
        } else {
            if( dao.selectTotalParking().contains(dto.getCarNumber())
                || dao.selectComm().contains(dto.getCarNumber()) ) {
                CommutationTicket.Term_Dday1();
                int result = dao.deleteExit(dto.getCarNumber());
                dao.deleteComm(dto.getCarNumber());
            } else {
                int result = dao.deleteExit(dto.getCarNumber());
                dao.deleteOneTime(dto.getCarNumber());
            }
        }
        break;
    }
}
```

□ 프로그램 설명

코드리뷰

CommutationTicket.java

정기권 구매를 위한 클래스

□ 프로그램 설명

```
public class CommutationTicket {
    public static Calendar inDay1; // public static 사용 이유
    public static void CommuTicket() throws AuthenException {

        DAO dao = new DAO();
        ParkingDTO dto = new ParkingDTO();
        Scanner sc = new Scanner(System.in);
        System.out.println("정기권을 선택하셨습니다. \n" + "정기권은 30일 단위이며, 자유롭게 입차 가능합니다.\n"
            + "다만 기간 내에 연장 또는 기간 초과 후 출차해주지 않으시면 견인되오니 주의 부탁드립니다.");
        for (int i = 0; i < 60; i++) {
            System.out.print("-");
        }
        System.out.println("\n정기권 수량 조회 중입니다.");
        System.out.println("...");

        ParkingMain pm = new ParkingMain();
        if (dao.selectComm().size() == 10) {
            System.out.println("정기권이 마감되었습니다.");
            return;
        } else {
            System.out.println("정기권이 " + (10 - dao.selectComm().size()) + "개 남았습니다.");
        }

        System.out.println("정기권을 등록할 수 있습니다.");

        boolean a = true;
        while (a) {
            System.out.println("차량 번호를 입력해주세요!");
            dto.setCarNumber(sc.nextLine());

            boolean isCarNumber = Pattern.matches("^[0-9가-힣]{3,4}\\s?[0-9]{4}", dto.getCarNumber());

            if (!isCarNumber) {
                System.out.println("차량 번호 형식이 맞지 않습니다.");
                System.out.println("다시 입력해 주세요.");
                continue;
            } else {
                if(dao.iscommu().containsKey(dto.getCarNumber())) {
                    System.out.println("중복된 차량번호입니다. 다시 입력해주세요.");
                    continue;
                } else {
                    break;
                }
            }
        }

        System.out.println("차종을 입력해주세요!");
        Payment pa = new Payment();
        boolean c = true;
        while (c) {
            System.out.println("1. 경차 " + " / " + " 2. 일반");
            String kind = sc.nextLine();
            dto.setCarKinds(kind); // DB등록
            if (dto.getCarKinds().contains("경차") || dto.getCarKinds().contains("1")) {
                System.out.println("10%가 할인됩니다.");
                for (int i = 0; i < 15; i++) {
                    System.out.print("-");
                }
            }

            System.out.println("\n감사합니다.\n" + "결제창으로 넘어갑니다.");
            dto.setWhatDate(1);
            pa.payment();
        } else {
            System.out.println("다시 입력해주세요!");
            continue;
        }
        break;
    }

    dao.insertCommuTicket(dto.getCarNumber(), dto.getCarKinds(), dto.getName(),
        dto.getPhoneNum(), dto.getRegistration_date(), dto.getTerm());
}

public static boolean already(String Car_Num) {
    ParkingDTO dto = new ParkingDTO();
    DAO dao = new DAO();
    return dao.selectComm().contains(dao.getCarNumber());
}
```

□ 프로그램 설명

코드리뷰

Payment.java

결제를 위한 클래스

□ 프로그램 설명

```
public class Payment{
    ParkingDTO dto = new ParkingDTO();
    DAO dao = new DAO();

    public void payment() throws AuthenException{
        Scanner sc = new Scanner(System.in);
        // 정기권이면 1, 일회권이면 2
        if(dto.getWhatDate() == 1) {
            dto.setAmount(200000);
        } else {
            dto.setAmount(5000);
            int a = (int)(dto.getMin()-30);
            if(a<0) {
                dto.setAmount(5000);
            } else {
                dto.setAmount(5000 + a/10*1000);
            }
            System.out.println(dto.getAmount()+"원");
        }

        if(dao.ent_carKinds().containsValue("경차")||dao.ent_carKinds().containsValue("1")
            || dto.getCarKinds().contains("경차")||dto.getCarKinds().contains("1")) {

            System.out.println("경차 할인 혜택(10%)이 적용됩니다.");
            dto.setAmount(dto.getAmount() - dto.getAmount()/10);
        }

        System.out.println(dto.getAmount() + "원 입니다. 결제 방법을 선택해주세요.");
        System.out.println("1. 카드결제 2. 현금결제");
        boolean flag = true;
        while(flag) {
            try {
                String choice = sc.next();
                switch(Integer.parseInt(choice)) {
                    case 1 :
                        System.out.println("카드를 꽂아주세요.(카드입력)");
                        dto.setPayment(sc.next());
                }
            }
        }
    }
}
```

```
if(dto.getPayment().equals("카드")) {
    System.out.println("감사합니다. 결제 완료되었습니다.");
    flag = false;
    break;
} else {
    System.out.println("카드가 올바르지 않습니다. 다시 입력해주세요.");
    continue;
}

case 2 :
    System.out.println("현금을 넣어주세요.(넣을 만큼 금액 입력. 최소단위 천원)");
    while(flag) {
        String check = sc.next();
        boolean ispay = Pattern.matches("(\\d{4})", check);
        if(ispay) {
            if(dto.getPayment() != null) {
                dto.setPayment(String.valueOf(Integer.parseInt(dto.getPayment())+Integer.parseInt(check)));
            } else {
                dto.setPayment(check);
            }
        } else {
            System.out.println("잘못 입력하셨습니다. 올바른 금액을 입력해주세요.");
            continue;
        }
        System.out.println(dto.getPayment() + "원 받았습니다.");
        int change = Integer.parseInt(dto.getPayment()) - dto.getAmount();
        int change_count = 0;
        int [] money = {10000, 5000, 1000, 500, 100};
        if(dto.getAmount() == Integer.parseInt(dto.getPayment())) {
            System.out.println("거스름돈이 없습니다. 감사합니다.");
            flag = false;
            break;
        } else if(dto.getAmount() > Integer.parseInt(dto.getPayment())) {
            System.out.println("금액이 부족합니다. 돈을 더 넣어주세요.");
            continue;
        } else {
            System.out.print("거스름돈 : ");
            for(int i = 0; i<money.length; i++){
                change_count = change / money[i];
                System.out.print(money[i] + "원 : " + change_count + "\t");
                change = change % money[i];
            }
            System.out.println("감사합니다.");
            flag = false;
            break;
        }
    }
} catch (NumberFormatException e) {
    System.out.println("올바른 번호를 입력해주세요.");
    continue;
} catch (Exception e) {
    System.out.println("[ERROR]알 수 없는 에러가 발생했습니다. 관리자에게 문의해주세요.");
    e.printStackTrace();
}
```

□ 프로그램 설명

코드리뷰

TotalParking.java

정기권, 일회권, 남은자리를 출력해주는 클래스

□ 프로그램 설명

```
public class TotalParking {  
  
    public static int comm() {  
        DAO dao = new DAO();  
        int a = (10-dao.selectComm().size());  
        return a;  
    }  
    public static int oneTime() {  
        DAO dao = new DAO();  
        int a = (20-dao.selectOnetime().size());  
        return a;  
    }  
    public static void TotalParking() {  
        System.out.println("현재 남은 자리");  
        System.out.println("정기권 : " + comm() + "일회권 : " + oneTime());  
    }  
}
```

□ 프로그램 설명

코드리뷰

ParkingMain.java

메인 클래스

□ 프로그램 설명

```
public class ParkingMain {  
    public static void main(String[] args) throws AuthenException {  
        DAO dao = new DAO();  
        Scanner sc = new Scanner(System.in);  
        int choice;  
        while(true) {  
            do {  
                System.out.println("-----");  
                System.out.println(" 시간지나면 견인하조 주차장 ");  
                System.out.println("1. 입차      2. 출차      3. 정기권등록      4.주차현황");  
                System.out.println("-----");  
                System.out.print(">>");  
                choice = sc.nextInt();  
            } while (choice < 1 || choice > 5);  
        }  
    }  
}
```

```
switch(choice) {  
    case 1:  
        try {  
            Inout.Enterance();  
        } catch (AuthenException e) {  
            e.printStackTrace();  
        }  
        break;  
    case 2:  
        5. 종료: Inout.Exit();  
        break;  
    case 3:  
        try {  
            CommutationTicket.CommuTicket();  
        } catch (Exception e) {  
  
            e.printStackTrace();  
        }  
        break;  
    case 4:  
        TotalParking.TotalParking();  
        break;  
    case 5:  
        // 종료  
        DBConn.close(); // DB연결 끊고 종료  
        System.exit(0); // 프로그램 강제 종료  
}
```

□ 프로그램 설명

코드리뷰
DAO.java
DB의 데이터를 불러오는 클래스

□ 프로그램 설명

```
public int insertEntrance(String carNumber, String carKinds, String entranceTime) {  
    int result = 0;  
    Connection con = DBConn.getConnection();  
    PreparedStatement pstmt = null;  
    String sql;  
    try {  
        sql = "insert into entrance(carNumber, carKinds, entranceTime) ";  
        sql += "values (?, ?, sysdate)";  
        pstmt = con.prepareStatement(sql);  
        pstmt.setString(1, carNumber);  
        pstmt.setString(2, carKinds);  
        result=pstmt.executeUpdate();  
    } catch (Exception e) {  
        System.out.println(e.toString());  
    }  
    return result;  
}  
  
public List<ParkingDTO> selectTotalParking() {  
    Connection conn = new DBConn().getConnection();  
    ArrayList<ParkingDTO> list_day = new ArrayList<ParkingDTO>();  
    Connection con = null;  
    PreparedStatement pstmt = null;  
    ResultSet rs = null;  
    int index = 1;  
    try {  
        con = DBConn.getConnection();  
        StringBuffer sql = new StringBuffer();  
        sql.append("select carNumber from entrance");  
        pstmt = con.prepareStatement(sql.toString());  
        rs = pstmt.executeQuery(); //쿼리 실행  
        while(rs.next()) { //데이터베이스형식과 java 형식이 다름.  
            index = 1; //1부터 시작  
            String carNumber = rs.getString(index++);  
            ParkingDTO dto = new ParkingDTO(carNumber);  
            list_day.add(dto);  
        }  
    } catch (SQLException e) {  
        e.printStackTrace();  
    }  
    return list_day;  
}
```

```
public int deleteExit(String carNumber) {  
    int result = 0;  
    Connection conn = DBConn.getConnection();  
    try {  
        String sql = "delete from entrance where carNumber=?";  
        PreparedStatement pstmt = conn.prepareStatement(sql);  
        pstmt.setString(1, carNumber);  
        result = pstmt.executeUpdate();  
        pstmt.close();  
        conn.commit();  
    } catch (Exception e) {  
        System.out.println(e.toString());  
    }  
    return result;  
}
```



02

프로젝트 시연

입차, 출차 확인

정기권 등록 확인

주차현황 확인

□ 프로젝트 소감

01

김시우

처음하는 팀프로젝트여서 파트분배나 어디서부터 시작을 해야 할지 막막하였는데, 좋은 팀원들과 함께하여 즐겁게 진행 할 수 있었던거 같습니다. 물론 중간중간 많은 오류에 힘이 들었으나, 팀원들과 같이 머리를 맞대어 오류를 해결했을땐 말로 표현 할 수 없을정도의 성취감을 느꼈습니다. 나혼자서는 어렵고 힘들어서 중간에 포기를 했을거 같지만, 팀원들 덕분에 포기하지 않고 끝까지 프로젝트를 완성할 수 있어서 좋았습니다.

02

문기연

혼자했던 자바 프로젝트 때와 달리, 디비 연결부터 팀원 간의 파트 분배까지, 시작부터 어려움이 많은 프로젝트였습니다. 분명 혼자 했다면 포기했을지도 모르겠어요. 하지만 팀원들과 으쌰으쌰 한 덕분에 끝까지 올 수 있었습니다. 물론 끝의 끝까지도 어려웠어요 하하. 그래도 끝을 내니 무척 뿌듯합니다. 이제 다른 프로젝트 시작해볼게요?

03

박종휘

팀플을 함에 있어서는 소통이 가장 중요하다는 것을 다시 한번 느꼈습니다! 서로 모여서 오류를 수정해나가는 과정에서 정말 많은 의견을 나눴고 모르는 부분은 구글링을 해가며 같이 어려움을 헤쳐나갔습니다. 그러다 보니 혼자 할 때보다 더 다양한 솔루션들이 나왔고 조금 더 완성도 높은 결과를 얻을 수 있었습니다. 팀플은 혼자하는 것이 아니기에 자기의 주장을 무조건적으로 밀고나가는 것이 아니라 다른 사람들과 의견을 공유하면 조금 더 최선의 솔루션을 찾을 수 있다는 것을 팀 프로젝트하면서 다시 한번 느끼게되었습니다. 정말 뜻깊은 시간이였습니다.

04

김강영

혼자했던 자바 프로젝트 때와 달리, 디비 연결부터 팀원 간의 파트 분배까지, 시작부터 어려움이 많은 프로젝트였습니다. 분명 혼자 했다면 포기했을지도 모르겠어요. 하지만 팀원들과 으쌰으쌰 한 덕분에 끝까지 올 수 있었습니다. 물론 끝의 끝까지도 어려웠어요 하하. 그래도 끝을 내니 무척 뿌듯합니다. 이제 다른 프로젝트 시작해볼게요?

05

박지윤

이번 팀프로젝트를 하면서 좋은사람들과 즐겁게 열심히 했던거 같습니다 여러가지 시행착오와 오류가 있었지만 그때마다 다같이 함께 수정하고 해결하며 나아가면서 더 재미있게 했던거 같고 파트분배를 했지만 나중에는 결국 다같이 달려들어 함께 손보며 진행했는데 힘들었지만 다음에도 하고 싶을 만큼 보람있는 프로젝트였습니당:D



발표를 들어주셔서
|
감사합니다 :)

질문이 있으시다면 언제든지 말씀해주세요!
감사합니다.