

# 2020

경북산업직업훈련학교

백한별

## [주차관리 프로그램]

Window Form 을 통해 주차관리 프로그램을 구현하였다.

## 목차

프로젝트 개요.....	2
프로젝트 설명	
기능정의서 .....	3
개발환경 .....	4
개발기간.....	4
화면 설계	
UI 스타일 가이드.....	5
유즈케이스 .....	6
데이터 베이스	
Xml .....	7
Json .....	7
Ms-sql.....	8
기능 설계	
파일 구조 .....	9
인터페이스 정의서	
주차기능 .....	10
출차기능 .....	10
조회기능 .....	10
그 밖의 주요 코드 .....	11
기능테스트.....	14
배포환경 .....	17
개선사항 .....	18
Contact Us.....	19

## [ 프로젝트 개요 ]

사용자를 기반으로 한 주차관리를 할 수 있도록 xml, json, ms-sql을 통해 데이터를 관리하고 txt 파일로 프로그램 구동 현황을 백업하는 프로그램을 제작하였다.

- 지정한 경로에 xml 또는 json 파일이 있다면 그 파일을 바탕으로 주차장의 차량들을 관리하도록 한다. 만약 파일이 없다면 프로그램 구동 시 5개의 공간을 가진 파일을 생성한다. DB를 통해서 관리를 한다면 ms-sql에 관련 table을 생성한다.
- 차량의 주차와 출차, 주차된 차량 조회는 button과 textBox에 입력하는 내용을 바탕으로 구동한다.
- 주차된 차량의 관련 정보는 dataGridView를 통해서 확인할 수 있다.
- Button을 눌러서 구동한 내용을 MessageBox로 확인할 수 있다.
- Button을 눌러서 구동한 내용을 바탕으로 관련 오류 및 차량 관리 정보는 지정한 위치에 txt 파일로 관리한다. 만약 txt 파일이 없다면 txt 파일을 생성하고 백업한다.
- Txt 파일로 남는 로그 내용은 ListBox에서도 확인할 수 있다.

## [ 프로그램 설명 ]

### 1. 기능정의서

주차 관리		
No	기능	설명
C-M-001	주차	공간번호, 차량번호, 차주이름, 전화번호를 textBox에 입력하고 주차버튼을 통해 입력한 내용과 동작한 날짜와 시간 정보를 추가한다.
C-M-002	출차	dataGridView에 입력된 내용을 선택하거나 textBox에 공간번호, 차량번호, 차주이름, 전화번호를 입력하고 출차버튼을 통해서 주차되어있던 차량의 정보를 지운다.
C-M-003	조회	공간번호를 textBox에 입력하고 조회버튼을 통해 해당 공간번호에 차량의 유무 또는 차량의 번호를 확인한다.

## 2. 개발환경

A. Visual Studio 2019

B. DBMS : Microsoft SQL Server 2012

- i. Microsoft SQL Server Management Studio 11.0.2100.60
- ii. Microsoft Data Access Components(MDAC) 6.1.7601.17514
- iii. Microsoft MSXML 3.0.6.0
- iv. Microsoft Internet Explorer 9.11.9600.17843
- v. Microsoft .NET Framework 4.0.30319.42000
- vi. 운영체제 6.1.7601

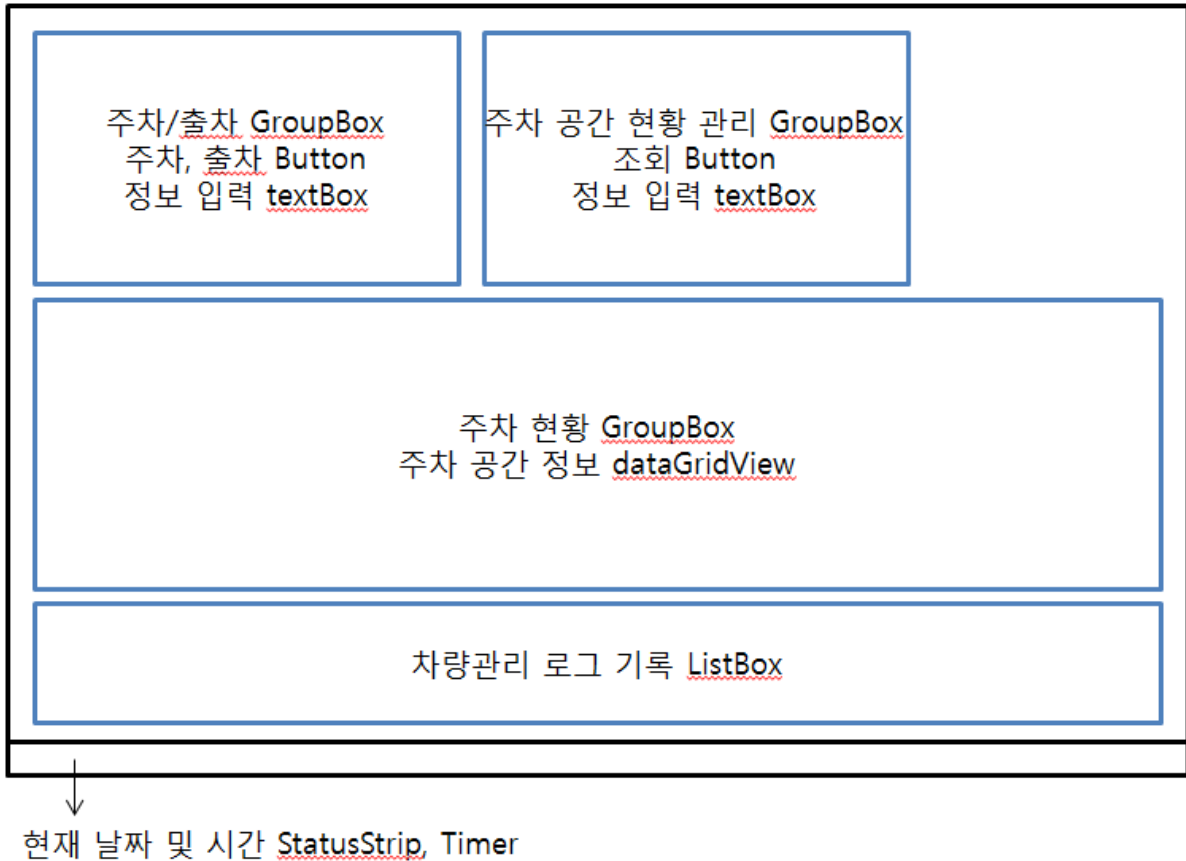
## 3. 개발 기간

2020.04.03 ~ 2020.04.10

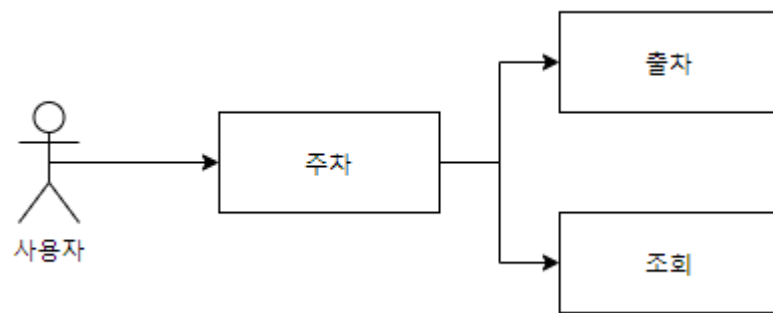
## [ 화면 설계 ]

### 1. UI 스타일 가이드

A. 해상도 : 816 x 603 이상 최적화



## 2. 유즈케이스



## [ 데이터 베이스 ]

### 1. Xml 파일 구조

속성	주요업무
parkingSpot	주차 공간
carNumber	차량 번호
driverName	차주 이름
phoneNumber	차주 전화번호
parkingTime	주차 시간

```
<?xml version="1.0"?>
- <cars>
  - <car>
    <parkingSpot>1</parkingSpot>
    <carNumber>11가1111</carNumber>
    <driverName>김일병</driverName>
    <phoneNumber>010111111111</phoneNumber>
    <parkingTime>2020-04-23 오전 10:58:40</parkingTime>
  </car>
```

### 2. Json 파일구조

```
{
  "cars": {
    "car": [
      {
        "parkingSpot": 1,
        "carNumber": "11가1111",
        "driverName": "김일순",
        "phoneNumber": "010111111111",
        "parkingTime": "2020년 4월 23일 목요일"
      }
    ]
  }
}
```



### 3. Ms-sql 테이블

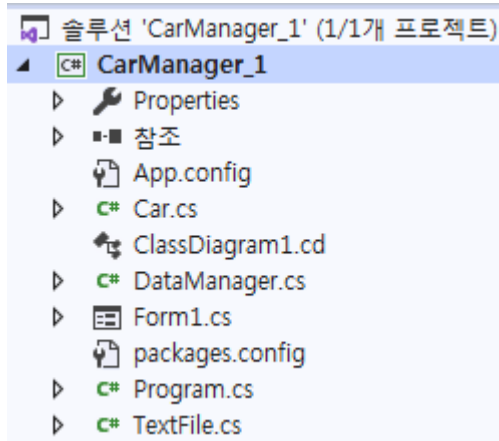
	열 이름	데이터 형식	Null 허용
🔑	Parking_Spot	int	<input type="checkbox"/>
	Car_Number	varchar(50)	<input type="checkbox"/>
	Driver_Name	varchar(50)	<input type="checkbox"/>
	Phone_Number	varchar(50)	<input type="checkbox"/>
	Parking_Time	datetime	<input type="checkbox"/>

[테이블] dbo.Car_Manager	
<div> <div> <div></div> <div></div> <div></div> </div> <div> <div></div> <div></div> <div></div> </div> </div>	
<div> <div>(ID)</div> <div> <div>(이름)</div> <div>Car_Manager</div> </div> <div> <div>데이터베이스 이름</div> <div>MYDB1</div> </div> <div> <div>서버 이름</div> <div>kb-pc#sqlexpress</div> </div> <div> <div>설명</div> <div></div> </div> <div> <div>스키마</div> <div>dbo</div> </div> </div>	
<div> <div>테이블 디자이너</div> <div> <div>ID 열</div> <div> <div>기본 데이터 공간 사양</div> <div>PRIMARY</div> </div> <div> <div>복제 됨</div> <div>아니요</div> </div> <div> <div>인덱싱 가능</div> <div>예</div> </div> <div> <div>잠금 에스컬레이션</div> <div>테이블</div> </div> <div> <div>텍스트/이미지 파일 그룹</div> <div>PRIMARY</div> </div> <div> <div>행 GUID 열</div> <div></div> </div> </div> </div>	

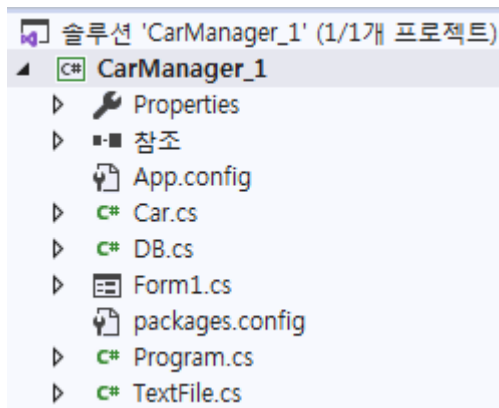
## [ 기능 설계 ]

### 1. 파일 구조

#### A. Xml / Json



#### B. Ms-sql



## 2. 인터페이스 정의서

### A. 주차기능

환경	개발환경	서비스 그룹	기능구현
서비스 명	C-M-001		
설명	주차관리 화면에서 '주차' 버튼을 누르면 Car 타입의 변수에 DataManager 클래스에서 Load 된 xml 이나 json 또는 DB 에 저장된 공간번호 'ParkingSpot' 의 값에 있는 데이터들을 담고 받아온 값 중 'CarNumber'로 차량이 이미 주차되어있는지 판별한다. 만약 주차된 차량이 없다면 입력해야 하는 값들을 입력하고 주차한다.		
주요 코드	<pre>Car car = DataManager.Cars.Single((x) =&gt; x.ParkingSpot.ToString() == textBox_ParkingSpot.Text);</pre> <p>textBox 에 있는 'ParkingSpot' 번호를 바탕으로 데이터 베이스에 저장되어있는 정보를 받아와 Car 타입의 변수 'car'에 저장한다.</p>		

### B. 출차기능

환경	개발환경	서비스 그룹	기능구현
서비스 명	C-M-002		
설명	주차관리 화면에서 '출차' 버튼을 누르면 Car 타입의 변수에 DataManager 클래스에서 Load 된 xml 이나 json 또는 DB 에 저장된 공간번호 'ParkingSpot'의 값에 있는 데이터들을 담고 받아온 값들에 공백을 넣어 출차시킨다.		

### C. 조회기능

환경	개발환경	서비스 그룹	기능구현
서비스 명	C-M-003		
설명	주차관리 화면에서 textBox 에 공간번호를 입력하고 '조회' 버튼을 누르면 Car 타입의 변수에 DataManager 클래스에서 Load 된 자료들 중 'ParkingNumber'가 일치하는 자료들을 담고 textBox 에서의 공간번호와 변수 'car'의 공간번호가 일치하고 'CarNumber'가 공백이 아닐 경우 주차된 것으로 판단하여 저장된 'CarNumber'를 MessageBox 로 보여준다.		
주요 코드	<pre>if(car.ParkingSpot.ToString() == textBox_Search.Text &amp;&amp; car.CarNumber != "")</pre> <p>Car 타입의 변수 'car'에 담은 정보를 바탕으로 주차여부를 판별한다.</p>		

#### D. 그 밖의 주요 코드

##### i. DataManager

```
public static void Load()
{
    string jsonFile = "./Cars.json";
    FileInfo fileInfo = new FileInfo(jsonFile);
    if (fileInfo.Exists)
    {
        string stCarValueJson = File.ReadAllText(@jsonFile);
        JObject jsonObjectCar = JObject.Parse(stCarValueJson);
        Cars = (from item in jsonObjectCar["cars"]["car"]
                select new Car()
                {
                    ParkingSpot = int.Parse(item["parkingSpot"].ToString()),
                    CarNumber = item["carNumber"].ToString(),
                    DriverName = item["driverName"].ToString(),
                    PhoneNumber = item["phoneNumber"].ToString(),
                    ParkingTime = DateTime.Parse(item["parkingTime"].ToString())
                }).ToList<Car>();
    }
    else
    {
        var jCarArray = new JArray();
        for (int i = 1; i < 6; i++)
        {
            var jCarObject = new JObject();
            jCarObject.Add("parkingSpot", i);
            jCarObject.Add("carNumber", "");
            jCarObject.Add("driverName", "");
            jCarObject.Add("phoneNumber", "");
            jCarObject.Add("parkingTime", DateTime.Now.ToString("yyyy-MM-dd hh:mm:ss"));
            jCarArray.Add(jCarObject);
        }

        var jCarArrayObject = new JObject();
        jCarArrayObject.Add("car", jCarArray);

        var jCarsObject = new JObject();
        jCarsObject.Add("cars", jCarArrayObject);

        File.WriteAllText(@"./Cars.json", jCarsObject.ToString());
    }
}
```

자료 내용을 백업시키는 파일의 존재여부를 확인하여 파일이 존재한다면 안의 자료들을 Load 하고 존재하지 않는다면 반복문을 통해 주차공간 5 개의 파일을 생성한다. (코드 예시 : Json)

ii. DB

```
class DB
{
    public static List<Car> Cars = new List<Car>();
    public static SqlConnection conn = new SqlConnection();

    참조 3개
    public static void ConnectDB()
    {
        conn.ConnectionString = string.Format("Data Source={0};" +
            "Initial Catalog = {1};" +
            "Integrated Security = {2};" +
            "Timeout = 3",
            "local", "MYDB1", "SSPI");
        conn = new SqlConnection(conn.ConnectionString);
        conn.Open(); //Open하면 DB연결
    }

    참조 1개
    public static void SelectDB()
    {
        ConnectDB();

        SqlCommand cmd = new SqlCommand();
        cmd.Connection = DB.conn;
        cmd.CommandText = "select * from Car_Manager";

        SqlDataAdapter da = new SqlDataAdapter(cmd);
        DataSet ds = new DataSet();
        da.Fill(ds, "Car_Manager");

        //db값을 list에 넣는 방법
        foreach (DataRow item in ds.Tables[0].Rows)
        {
            Car car = new Car();
            car.ParkingSpot = int.Parse(item["Parking_Spot"].ToString());
            car.CarNumber = item["Car_Number"].ToString();
            car.DriverName = item["Driver_Name"].ToString();
            car.PhoneNumber = item["Phone_Number"].ToString();
            car.ParkingTime = DateTime.Parse(item["parking_Time"].ToString());
            Cars.Add(car);
        }
    }
}
```

Ms-sql DB 를 이용하는 경우 DB 연결을 선 진행 후 DB 에 있는 자료들을 모두 선언되어있는 Car 타입의 리스트에 담아두고 기능구현 시 사용한다.

### iii. TextFile

```
public static void FileLoad(string str, string btn)
{
    DirectoryInfo di = new DirectoryInfo(@"./");
    if (!di.Exists) //Exists bool 타입 ! 없으면 true
    {
        di.Create();
    }
    using (StreamWriter writer = new StreamWriter(@"./ParkingHistroy.txt", true))
    {
        writer.WriteLine($"{DateTime.Now.ToString("yyyy/MM/dd hh:mm:ss")} '{str}' {btn} 클릭");
    }
}
```

사용자가 프로그램을 사용 중 정상작동을 하였는지, 오류의 여부 등을 txt 파일 형태로 저장한다.

만약 history 파일이 없다면 해당경로에 새로운 파일을 생성하고 로그를 남긴다.

## [ 기능테스트 ]

주차 관리

주차/출차

공간번호

차량번호

차주이름

전화번호

주차 공간 현황 관리

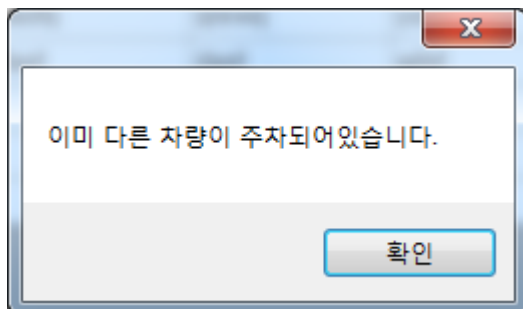
공간번호

주차 현황

ParkingSpot	CarNumber	DriverName	PhoneNumber	ParkingTime
1	dsafe	ewqrrwq	dgfdg	2020-04-23 오...
2	dasfq	qrewq	yutiu	2020-04-23 오...
3	dasf	dasf	adsf	2020-04-24 오...
4				2020-04-23 오...
5				2020-04-23 오...
6				2020-05-11 오...

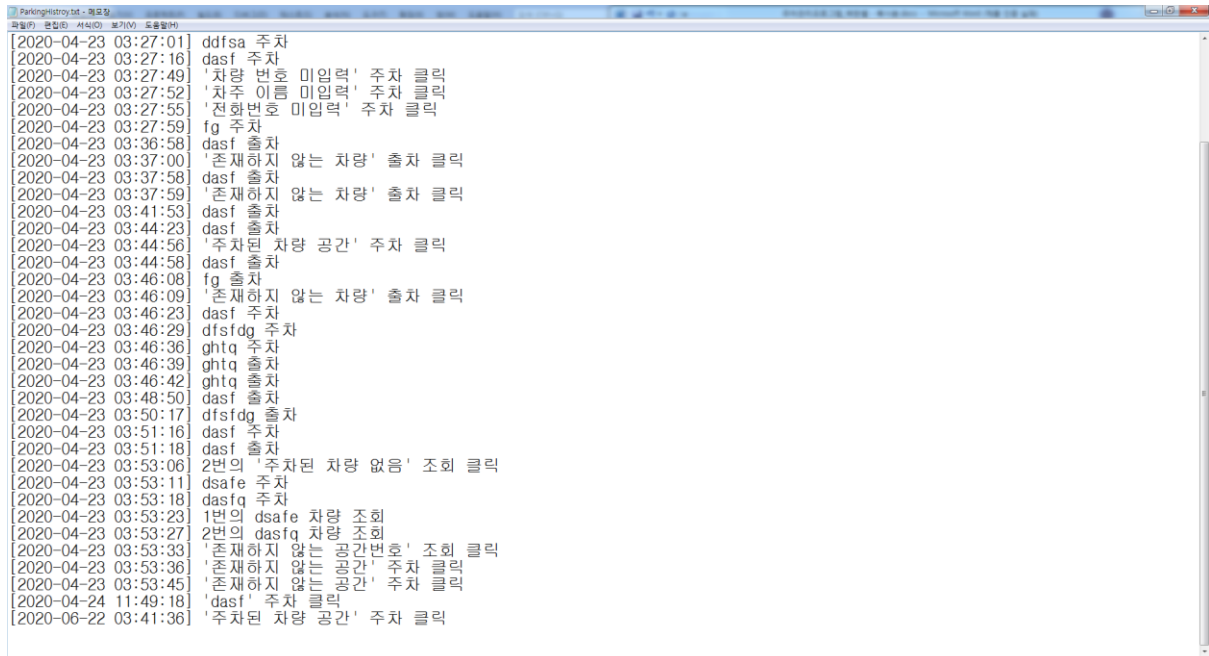
현재시간 : 2020년 06월 22일 15시 39분 25초

- textBox 의 내용이 dataGridView 를 선택하면 자동 변경된다.

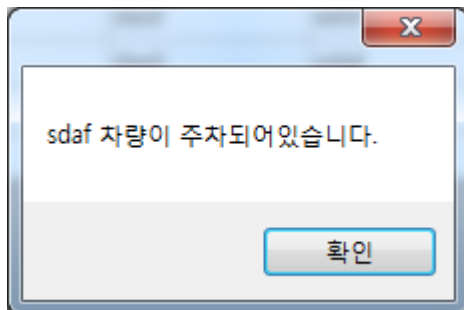


- 이미 차량이 존재하는 경우의 주차

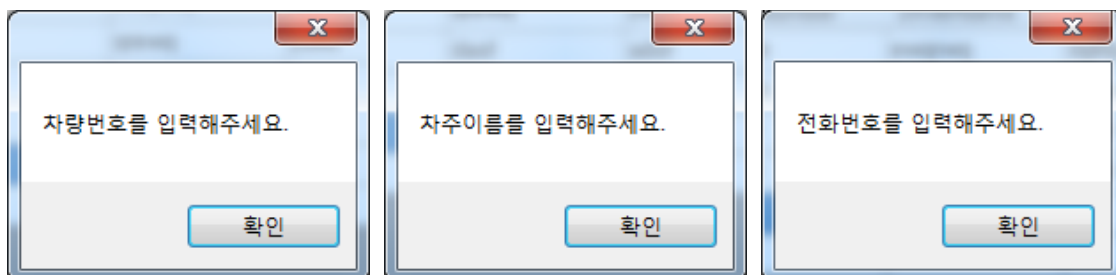
[2020-06-22 03:41:36] '주차된 차량 공간' 주차 클릭



- 프로그램을 실행하고 동작할때마다 동작 로그를 listBox 와 history 파일에 남긴다.

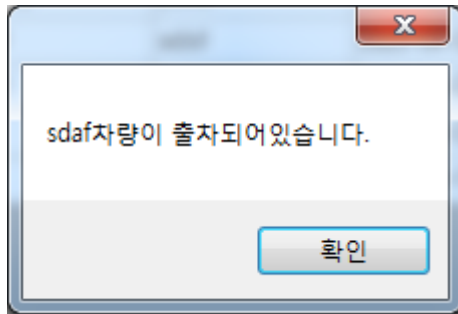


- 주차가 정상적으로 완료되었을 경우

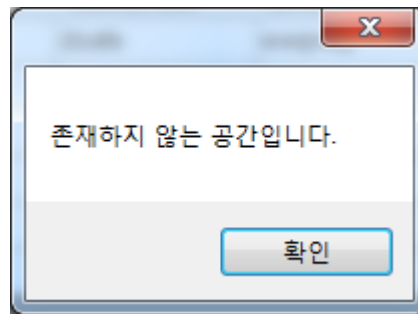
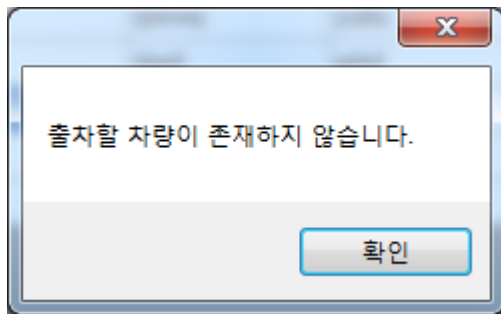


- 정보를 입력하지 않고 '주차' 버튼 클릭

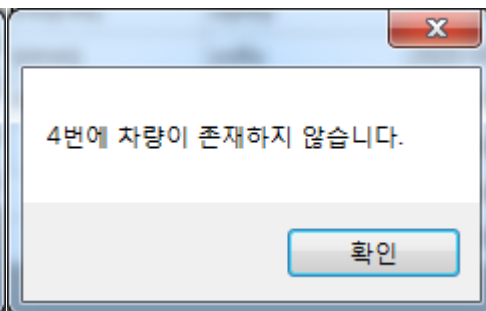
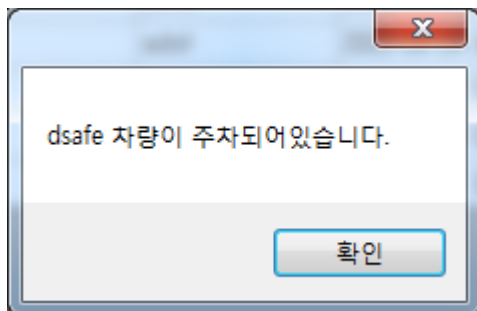




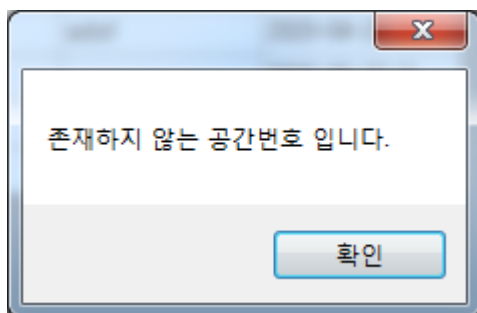
- 출차가 정상적으로 완료되었을 경우



- 출차 오류 (차량이 주차되어 있지 않을 때 / 존재하지 않는 공간번호 입력)



- 조회가 정상적으로 완료되었을 경우



- 조회 오류 (존재하지 않는 공간번호 입력)

## [ 배포환경 ]

IDE : Visual Studio 2019 이상

DBMS : Microsoft SQL Server 2012 이상

Microsoft SQL Server Management Studio 11.0 이상

Microsoft Data Access Components(MDAC) 6.1 이상

Microsoft MSXML 3.0 이상

Microsoft Internet Explorer 9.11 이상

Microsoft .NET Framework 4.0 이상

운영체제 6.1 이상

형상관리 : git, github <https://github.com/hanbyeolbaek/Parking>

## [ 개선사항 ]

1. 주차공간 숫자가 아닌 문자형태로 지하/지상/구역 나눠서 동작할 수 있도록 공간 늘려보기

## [ Contact Us ]

Name : 백한별 ( Baek Han Byeol )

Email : [yeongeon0765@gmail.com](mailto:yeongeon0765@gmail.com)

Tel : 010-2944-6756