

Parking Management System

주차 관리 시스템

김새샘

Parking Management System

Contents

/

- 01 프로젝트 개요
- 02 프로젝트 설명
- 03 화면 구현 및
상세 내용
- 04 결과

01

프로젝트 개요

Parking Management System

01 프로젝트 개요

주차 관리 시스템

- 주차 및 출차를 할 수 있다.
- 주차공간에 주차된 차가 있는지 조회한다.
- 공간번호에 지상, 층, 구역, 번호를 나눠 표시한다.
- 주차, 출차, 조회 시 주차 관리 기록을 남긴다.

02

프로젝트 설명

Parking Management System

02 프로젝트 설명

개발환경

- Visual Studio Community 2019
Windows Forms (.NET Framework)

개발기간

- 20. 04. 20 ~ 20. 04. 26

03

화면구현 및 상세내용

03 화면구현 및 상세내용

×

closeBtn

주차/출차

공간번호

지상11층 A구역11

차량번호

12가3456

차주이름

테스트

전화번호

01011111111

pAddBtn

주차

pRemoveBtn

출차

주차 공간 현황 관리

공간번호

조회

selLookUpBtn

주차 현황

	ParkingSpot	CarNumber	DriverName	PhoneNumber	ParkingTime
▶	111111	12가3456	테스트	01011111111	2020-06-12 오후 3...
	211111	23나4567	테스트2	01022222222	2020-06-14 오후 5...
	111113				2020-05-12 오후 2...
	111114				2020-05-12 오후 2:07
	111115				2020-06-12 오전 1...
					2020-04-14

dataGridView

주차 관리 기록

logPrint

현재 시간 : 2020-06-14 17:10:19

timer_DisplayNowtime

Form1.cs

```
private void button_parkingAdd_Click(object sender, EventArgs e)
{
    if (textBox_parkingSpot.Text.Trim() == "")
    {
        MessageBox.Show("주차공간을 입력해주세요.");
    }
    else if (textBox_carNumber.Text.Trim() == "")
    {
        MessageBox.Show("차량 번호를 입력해주세요.");
    }
    else
    {
        try
        {
            ParkingCar car = DataManager.Cars.Single((x) => x.ParkingSpot.ToString() == textBox_parkingSpot.Text.ToString());
            if (car.CarNumber.Trim() != "")
            {
                MessageBox.Show("해당 공간에는 이미 차가 존재합니다.");
            }
            else
            {
                car.ParkingSpot = int.Parse(textBox_parkingSpot.Text);
                car.CarNumber = textBox_carNumber.Text;
                car.DriverName = textBox_driverName.Text;
                car.PhoneNumber = textBox_phoneNumber.Text;
                car.ParkingTime = DateTime.Now;

                dataGridView_parkingManager.DataSource = null;
                dataGridView_parkingManager.DataSource = DataManager.Cars;
                DataManager.Save();

                string contents = $"주차 공간 {textBox_parkingSpot.Text}에 {textBox_carNumber.Text}차를 주차하였습니다.";
                WriteLog(contents);
            }
        }
        catch (Exception)
        {
            string contents = $"주차할 수 없습니다. 주차공간 {textBox_parkingSpot.Text} 은(는) 존재하지 않습니다.";
            MessageBox.Show(contents);
        }
    }
}
```

The screenshot shows a parking reservation interface. On the left, a form titled '주차/출차' (Parking/Departure) contains the following fields: '공간번호' (Space Number) with the value '지상11층 A구역14' (Ground 11th floor A Zone 14), '차량번호' (Vehicle Number) with '45라6789', '차주이름' (Driver Name) with '테스트4', and '전화번호' (Phone Number) with '01044444444'. To the right of these fields are two buttons: '주차' (Park) and '출차' (Departure). The '주차' button is highlighted with a red rectangle. An overlay dialog box is open on the right, displaying the message '주차 공간 지상11층 A구역14번에 45라6789차를 주차하였습니다.' (The car 45라6789 has been parked in space Ground 11th floor A Zone 14). The dialog also has a '확인' (Confirm) button highlighted with a blue rectangle.

- 공간번호 111114를 입력하고 '주차'했을 경우 [지상11층 A구역14번]으로 변환된다.

The screenshot shows the same parking reservation interface as above, but with empty input fields for '공간번호', '차량번호', '차주이름', and '전화번호'. The '주차' button is still highlighted with a red rectangle. An overlay dialog box is open, displaying the error message '주차공간을 입력해주세요.' (Please enter the parking space). The dialog has a '확인' (Confirm) button highlighted with a blue rectangle.

- 공간번호에 정보 입력을 하지 않고 '주차' 버튼을 클릭했을 때 '주차공간을 입력해주세요.' 팝업창이 열린다.

주차/출차

공간번호 111114

차량번호 45라6789

차주이름 테스트4

전화번호 01044444444

주차

출차

해당 공간에는 이미 차가 존재합니다.

확인

- 주차중인 공간을 입력하고 '주차' 버튼을 클릭하면
'해당 공간에는 이미 차가 존재합니다.' 팝업창이 열린다.

주차/출차

공간번호 111116

차량번호 56마7891

차주이름 테스트5

전화번호 01055555555

주차

출차

주차 공간 현황 관리

주차할 수 없습니다. 주차공간 111116 은(는) 존재하지 않습니다.

확인

- 존재하지 않는 공간을 입력하고 '주차' 버튼을 클릭하면
'주차할 수 없습니다. 주차공간 ~은(는) 존재하지 않습니다.'
팝업창이 열린다.

```

private void button_parkingRemove_Click(object sender, EventArgs e)
{
    if (textBox_parkingSpot.Text.Trim() == "")
    {
        MessageBox.Show("주차공간을 입력해주세요.");
    }
    else
    {
        try
        {
            ParkingCar car = DataManager.Cars.Single((x) => x.ParkingSpot.ToString() == textBox_parkingSpot.Text.ToString());
            if (car.CarNumber.Trim() == "")
            {
                MessageBox.Show("해당 공간에는 아직 주차된 차가 없습니다.");
            }
            else
            {
                string oldcar = car.CarNumber;
                car.ParkingSpot = int.Parse(textBox_parkingSpot.Text);
                car.CarNumber = "";
                car.DriverName = "";
                car.PhoneNumber = "";
                car.ParkingTime = DateTime.Now;

                dataGridView_parkingManager.DataSource = null;
                dataGridView_parkingManager.DataSource = DataManager.Cars;
                DataManager.Save();
                string contents = $"{oldcar}차를 주차 공간 {textBox_parkingSpot.Text}에서 출차하였습니다.";

                WriteLog(contents);
            }
        }
        catch (Exception)
        {
            string contents = $"출차할 수 없습니다. 주차공간 {textBox_parkingSpot.Text} 은(는) 존재하지 않습니다.";
            MessageBox.Show(contents);
        }
    }
}

```

주차/출차

공간번호

차량번호

차주이름

전화번호

45라6789차를 주차 공간 지상11층 A구역14번에서 출차하였습니다.

- 공간번호를 입력하고 ‘출차’했을 경우
‘~차를 주차 공간 ~에서 출차하였습니다.’ 팝업창이 열린다.

주차/출차

공간번호

차량번호

차주이름

전화번호

해당 공간에는 아직 주차된 차가 없습니다.

- 주차된 차가 없는 공간을 입력하고 ‘출차’ 버튼을 클릭하면
‘해당 공간에는 아직 주차된 차가 없습니다.’ 팝업창이 열린다.

```

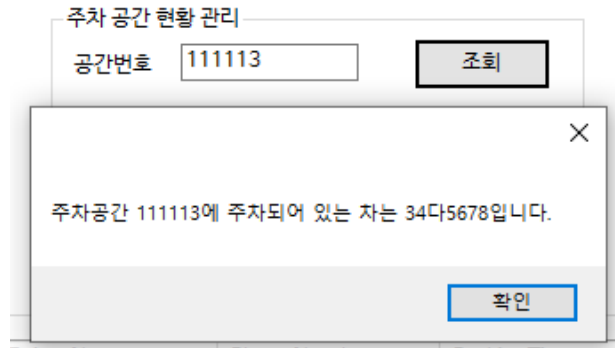
private void button_SelectedLookUp_Click(object sender, EventArgs e)
{
    try
    {
        int parkingSpot = int.Parse(textBox_parkingSpot_lookUp.Text);
        string ParkedCar = lookUpParkingFunction(parkingSpot);
        if (ParkedCar != "")
        {
            string contents = $"주차공간 {textBox_parkingSpot_lookUp.Text}에 주차되어 있는 차는 {ParkedCar}입니다.";

            WriteLog(contents);
        }
        else
        {
            string contents = $"주차공간 {textBox_parkingSpot_lookUp.Text}에 주차된 차가 없습니다.";

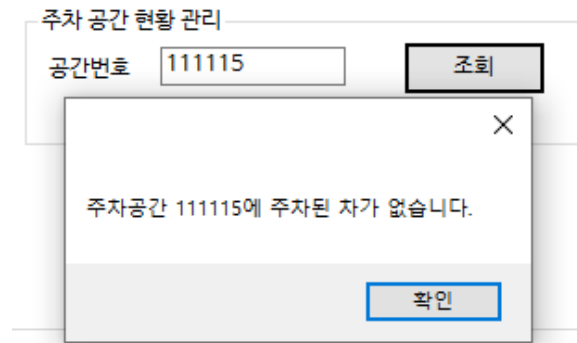
            WriteLog(contents);
        }
    }
    catch (Exception)
    {
        MessageBox.Show("주차공간번호가 유효하지 않습니다.");
    }
}

private string lookUpParkingFunction(int parkingSpot)
{
    try
    {
        string ParkedCarNumber = "";
        foreach (var item in DataManager.Cars)
        {
            if (item.ParkingSpot == parkingSpot)
                ParkedCarNumber = item.CarNumber;
        }
        return ParkedCarNumber;
    }
    catch (Exception e)
    {
        throw;
    }
}

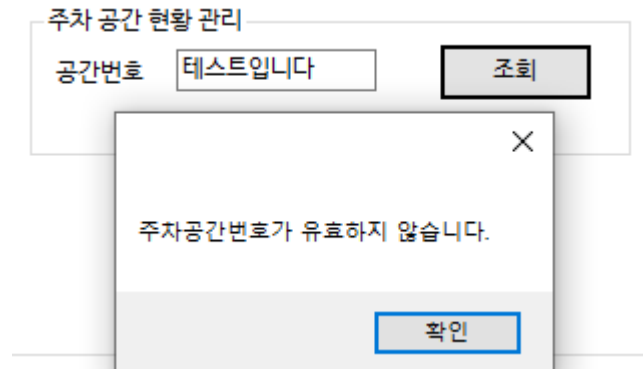
```



- 주차된 공간번호를 입력하고 '조회' 버튼을 클릭하면
'주차공간 ~에 주차되어 있는 차는 ~입니다.' 팝업창이 열린다.



- 주차되지 않은 공간번호를 입력하고 '조회' 버튼을 클릭하면
'주차공간 ~에 주차된 차가 없습니다.' 팝업창이 열린다.



- 유효하지 않은 공간번호를 입력하고 '조회' 버튼을 클릭하면 '주차공간번호가 유효하지 않습니다.' 팝업창이 열린다.


```

private void timer_DisplayNowtime_Tick(object sender, EventArgs e)
{
    toolStripStatusLabel_Now.Text = "현재 시간 : " + DateTime.Now.ToString("yyyy/MM/dd HH:mm:ss");
}

private void dataGridView_parkingManager_CurrentCellChanged(object sender, EventArgs e)
{
    try
    {
        ParkingCar car = dataGridView_parkingManager.CurrentRow.DataBoundItem as ParkingCar;

        textBox_parkingSpot.Text = Psc(car.ParkingSpot.ToString());
        textBox_carNumber.Text = car.CarNumber;
        textBox_driverName.Text = car.DriverName;
        textBox_phoneNumber.Text = car.PhoneNumber;
    }
    catch (Exception exception)
    {
    }
}

```

주차/출차

공간번호

차량번호

차주이름

전화번호

주차 공간 현황 관리

공간번호

주차 현황

	ParkingSpot	CarNumber	DriverName	PhoneNumber	ParkingTime
▶	111111	12가3456	테스트	01011111111	2020-06-12 오후 3...
	211111	23나4567	테스트2	01022222222	2020-06-14 오후 5...
	111113	34다5678	테스트3	01033333333	2020-06-14 오후 5...
	111114				2020-06-14 오후 8...
	111115				2020-04-14

- 주차 현황의 데이터 그리드 셀을 선택하면
주차/출차 텍스트박스에 내용이 출력된다.

DataManager.cs

```
public static void Load()
{
    try
    {
        string carsOutput = File.ReadAllText(@"./Cars.xml");
        XElement carsXElement = XElement.Parse(carsOutput);
        Cars = (from item in carsXElement.Descendants("car")
                select new ParkingCar()
                {
                    ParkingSpot = int.Parse(item.Element("parkingSpot").Value),
                    CarNumber = item.Element("carNumber").Value,
                    DriverName = item.Element("driverName").Value,
                    PhoneNumber = item.Element("phoneNumber").Value,
                    ParkingTime = item.Element("parkingTime").Value == "" ? DateTime.Now : DateTime.Parse(item.Element("parkingTime").Value)
                }).ToList<ParkingCar>();
    }
    catch (Exception exception)
    {
        MessageBox.Show("Cars 파일이 누락되었습니다!!!");
        CreateFile();
        Save();
        Load();
    }
}

public static void CreateFile()
{
    string fileName = @"./Cars.xml";
    StreamWriter textWrite = File.CreateText(fileName);
    textWrite.Dispose();
}
```

- Load()에서 Cars.xml 파일을 읽어온다.
Cars.xml 파일이 없을 경우 'Cars 파일이 누락되었습니다!!!' 를 출력하고, CreateFile()에서 Cars.xml 파일을 생성하고 읽어온다.

```

public static void Save()
{
    string carsOutput = "";
    carsOutput += "<cars>\n";
    if(Cars.Count > 0)
    {
        foreach (var item in Cars)
        {
            carsOutput += "<car>\n";
            carsOutput += "  <parkingSpot>" + item.ParkingSpot + "</parkingSpot>\n";
            carsOutput += "  <carNumber>" + item.CarNumber + "</carNumber>\n";
            carsOutput += "  <driverName>" + item.DriverName + "</driverName>\n";
            carsOutput += "  <phoneNumber>" + item.PhoneNumber + "</phoneNumber>\n";
            carsOutput += "  <parkingTime>" + item.ParkingTime + "</parkingTime>\n";
            carsOutput += "</car>\n";
        }
    }
    else
    {
        for(int i = 1; i <=5; i++)
        {
            carsOutput += "<car>\n";
            carsOutput += $"  <parkingSpot>{i}</parkingSpot>\n";
            carsOutput += "  <carNumber></carNumber>\n";
            carsOutput += "  <driverName></driverName>\n";
            carsOutput += "  <phoneNumber></phoneNumber>\n";
            carsOutput += "  <parkingTime></parkingTime>\n";
            carsOutput += "</car>\n";
        }
    }
    carsOutput += "</cars>";

    File.WriteAllText(@"./Cars.xml", carsOutput);
}

```

- Cars.xml에 텍스트를 저장한다.

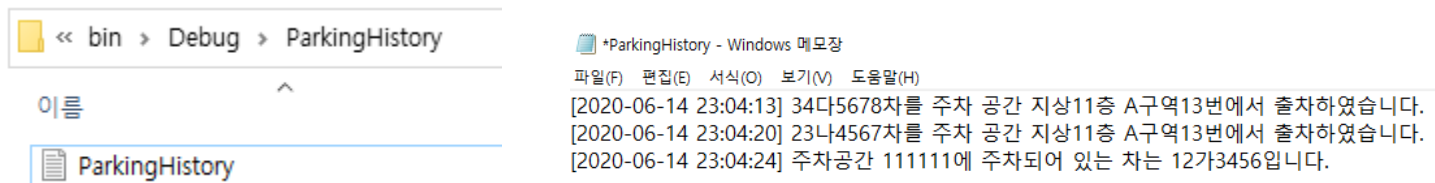
CreateFile에 의해 파일을 생성한 경우 else의 for문을 실행한다.

```

public static void PrintLog(string contents)
{
    DirectoryInfo di = new DirectoryInfo("ParkingHistory");
    if (!di.Exists)
    {
        di.Create();
    }

    using (StreamWriter writer = new StreamWriter("ParkingHistory" + "\\\" + "ParkingHistory" + ".txt", true))
    {
        writer.WriteLine(contents);
    }
}

```



- ParkingHistory 폴더 안의 ParkingHistory.txt 파일에 로그를 저장한다.

04

결과

04 결과

X

주차/출차

공간번호

차량번호

차주이름

전화번호

주차 공간 현황 관리

공간번호

주차 현황

	ParkingSpot	CarNumber	DriverName	PhoneNumber	ParkingTime
▶	111111	12가3456	테스트	01011111111	2020-06-12 오후 3...
	211111	23나4567	테스트2	01022222222	2020-06-14 오후 5...
	111113				2020-05-12 오후 2... 2020-05-12 오후 2:07
	111114				2020-06-12 오전 1...
	111115				2020-04-14

주차 관리 기록

Contact

/

Email kimss.river@gmail.com

Github <https://github.com/kimss-river>