

독서실 관리 프로그램

Contents.

- 1 소개 및 목적
- 2 Window From 및 기본
- 3 로직 소개(DB,Class)
- 4 기타 내용

Part 1

소개 및 목적

Part 1, 프로그램 소개

프로그램 만든 목적 및 사용 정보

1

개인 지정석 독서실 관리를 위한 프로그램

2

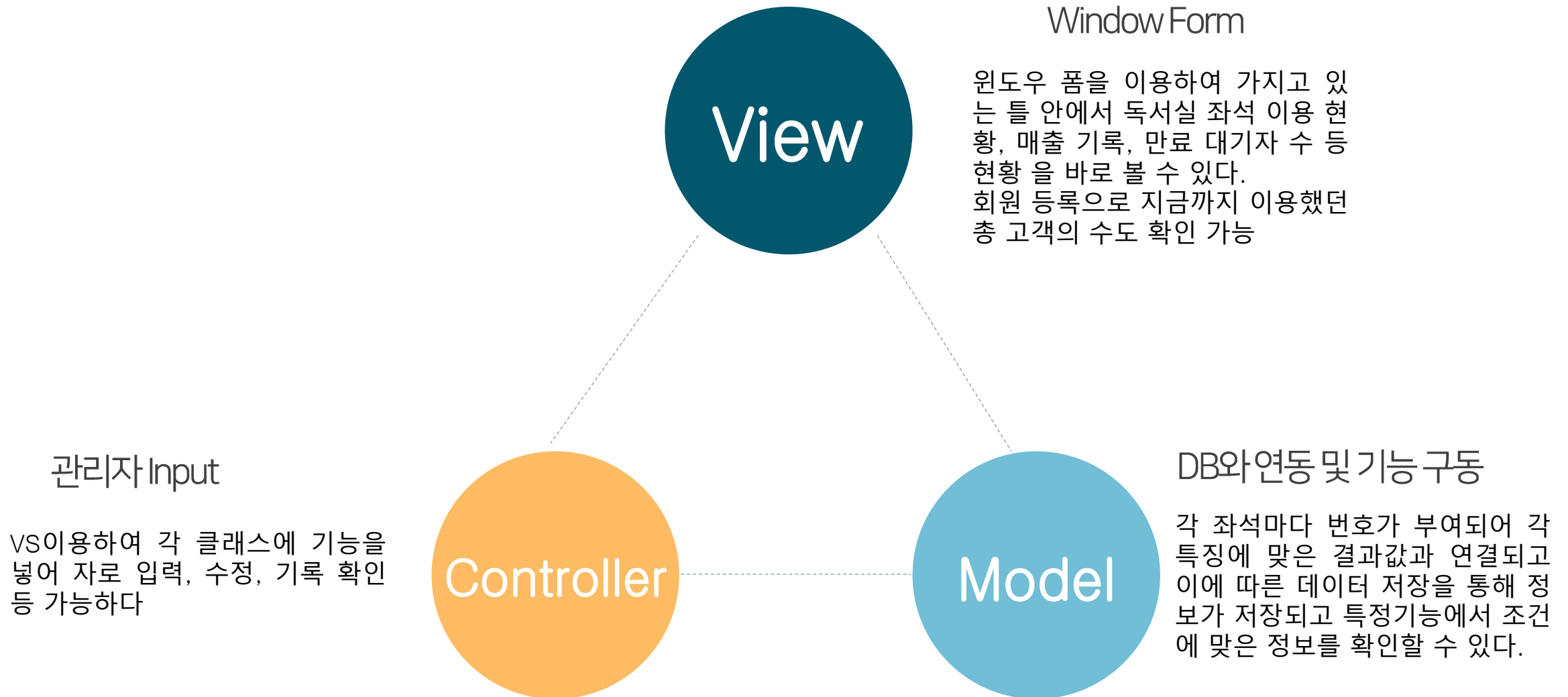
회원가입으로 중복가입 방지 및 실시간 독서실 등록 현황 확인

3

C# 언어를 이용한 Window Form 제작

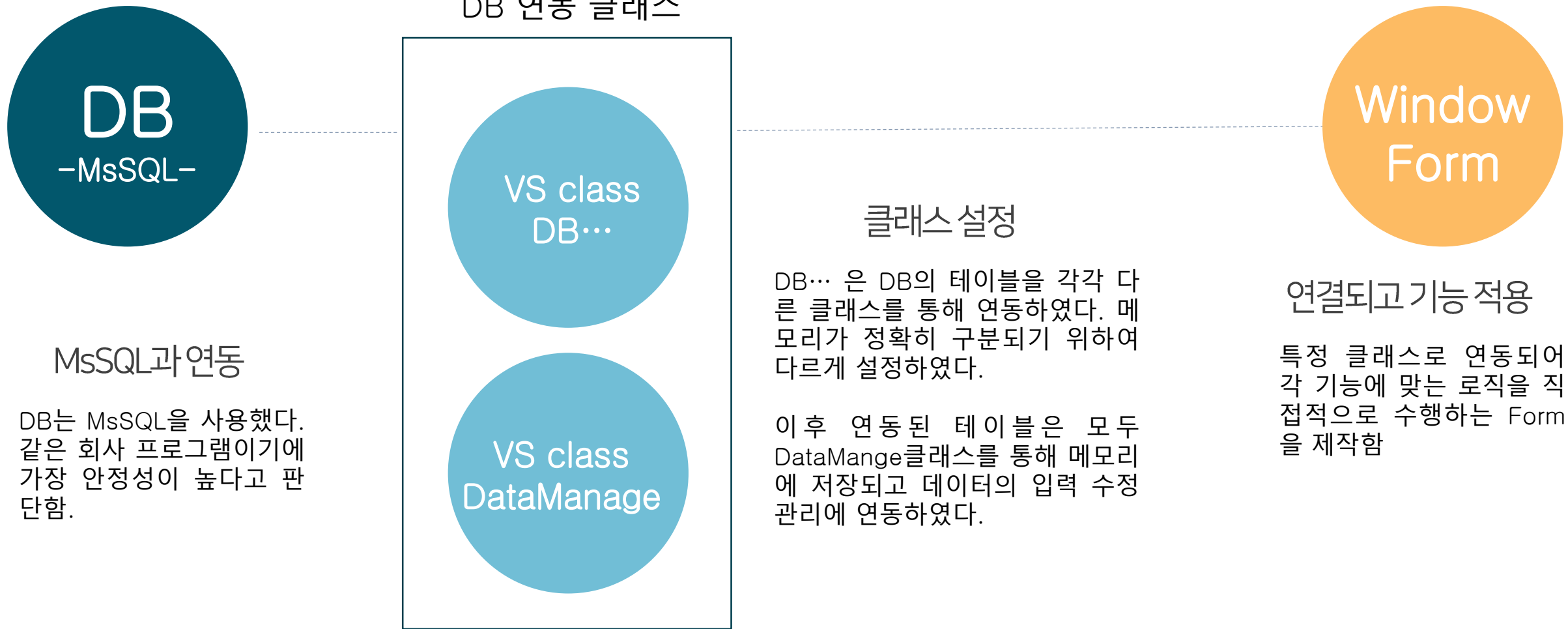
Part 1, 기본 프로그램 구조

DB와 연동하여 MVC패턴일 참고하여 구조를 설정



Part 1, 구체적인 프로그래밍 구조

Lorem Ipsum is simply dummy text of the printing and typesetting industry.



회원 관리

회원 가입을 통해 이용
객 확인이 가능하며 특
정 기능을 이용했던 사
람의 정보를 확인도 가
능

>>

좌석 실시간 확인

자리 이용내역 중 가장
중요한 현재상황을 실
시간으로 확인 가능
수치적, 시각적 모두
가능하며 언제 그 좌석
이 여석이 되는 시점도
확인 가능

>>

특정기간 매출확인

매출 확인을 통해 특정
기간에 자리에 등록된
연 건수, 금액, 환불금
등을 확인할 수 있다.
그 외 기능으로 가장
먼저 여석이 언제 발생
하는지 현황으로 확인
가능

>>

독서실 운영

실시간 운영이 가능하
고 좌석이나 구조에 따
라 변경이 유연하다.
필요 할 시 DB만 유지
된다면 다른 기능을 유
연하게 넣을 수 있다.

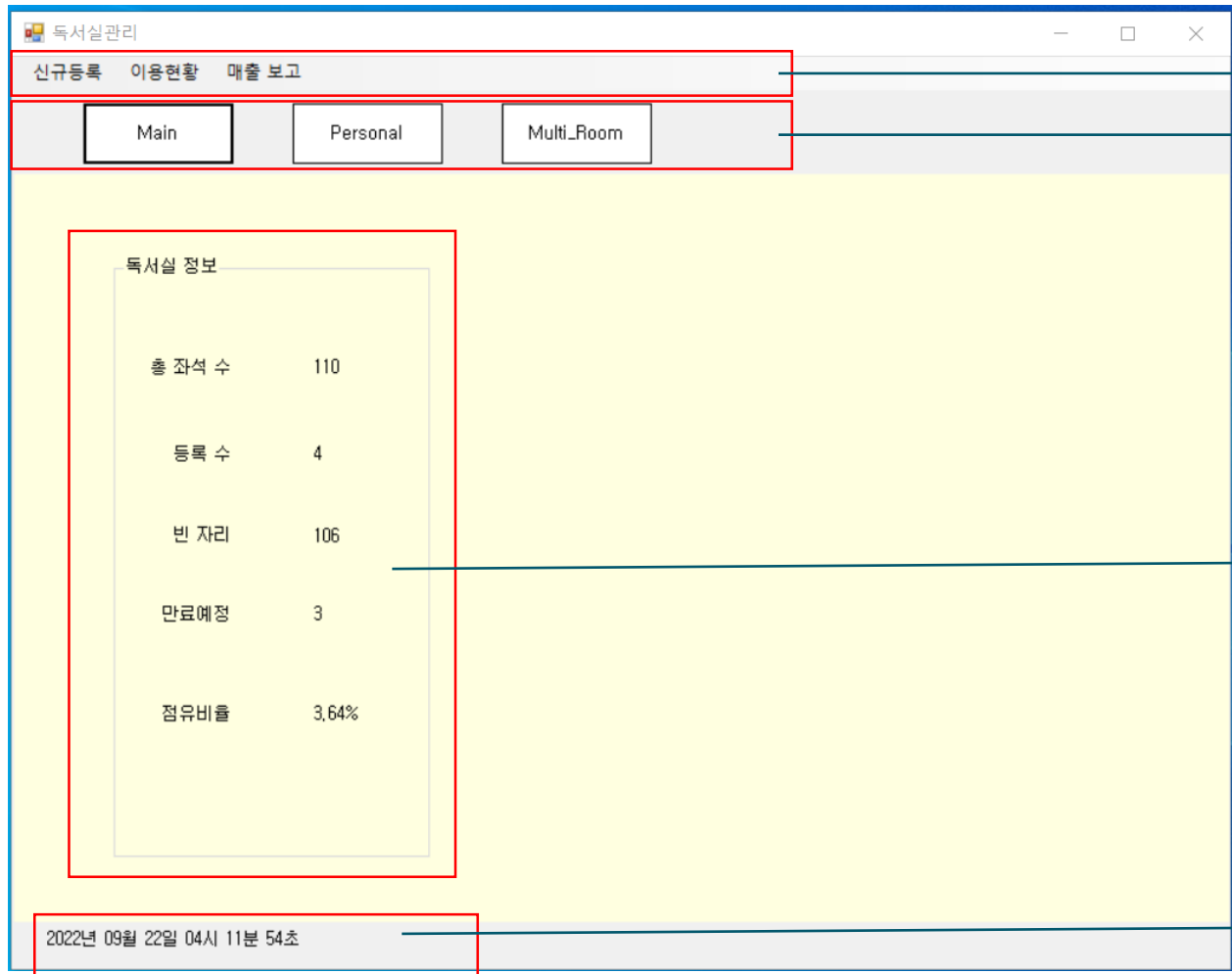
사용 기록이 저장되어
이슈가 발생해도 위치
를 확인할 수 있다.

Part 2

프로그램 이용 (Window Form)

Part 2, 메인창

프로그램 실행 후 첫 화면 및 기능 소개



실시간 좌석 정보가 아닌 기능을 연동

Usercontrol을 이용하여 실시간 으로 각 목적에 맞는 정보를 실시간으로 시각적자료로 확인

DB를 연동하여 현재 이용객 및 만료 예정 인원확인 (실시간)

실시간 시간 제공

Part 2, 도구모음기능소개

회원 가입 및 등록 된 정보 확인 가능

ID_manage

Id	Name
reddde	김정반
popo	포동이

신규정보

ID :

이름 :

연락처 :

생년월일: 2022년 9월 22일 ▼

※ 주의사항

ID는 중복불가
연락처는 필수확인입니다.
생년월일은 정확히 작성해주세요

회원 가입
(ID 중복불가)

등록확인

검색

ID : 방 : 자리 :

	roomNum	seatNum	userId	startday	endday
▶	201	2	popo	2022-09-22	2022-09-22
	201	1	popo	2022-09-22	2022-09-22
	201	6	reddde	2022-09-22	2022-09-23
	201	7	popo	2022-09-22	2022-10-07

현재 등록 인원 명단
(endday 기준으로 정렬)

매출관리

검색기간 : 2022년 9월 21일 수요일 ~ 2022년 9월 22일 목요일

검색 결과

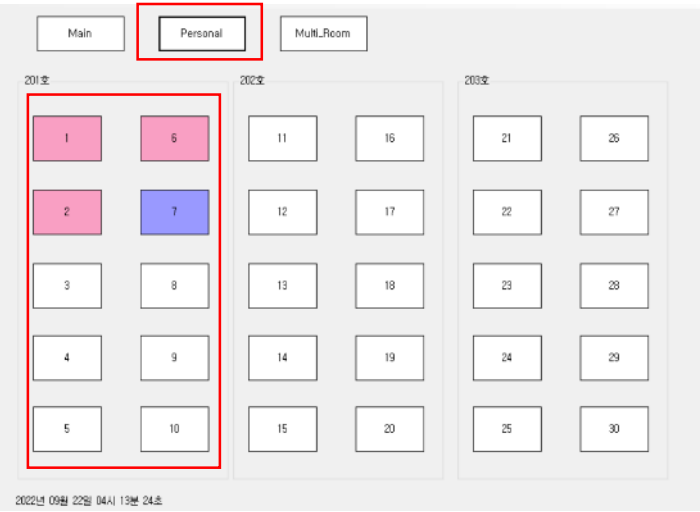
	등록번호	ID	방호수	자리번호	등록일	결제금액	시작일	만료일	비고
▶	2	popo	201	7	2022-09-22	85000	2022-09-22	2022-10-07	
	3	reddde	201	6	2022-09-22	8000	2022-09-22	2022-09-23	
	4	popo	201	2	2022-09-22	8000	2022-09-22	2022-09-22	
	5	popo	201	1	2022-09-22	8000	2022-09-22	2022-09-22	

총 건수 : 4건 매출 : 109000원 환불금 : 0원 총 매출 : 109000원

검색 기간을 특정하여 기간 내 등록된 내역 정보확인

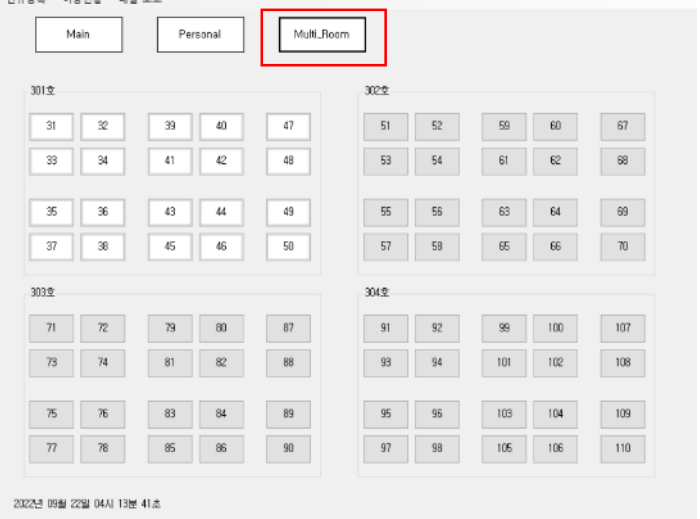
Part 2, 실시간 여석 확인 품(메인창이 변경됨)

다인실 개인실로 구별(50번좌석까지만 구동), 원하는 자석 클릭 시 결제 페이지 팝업됨

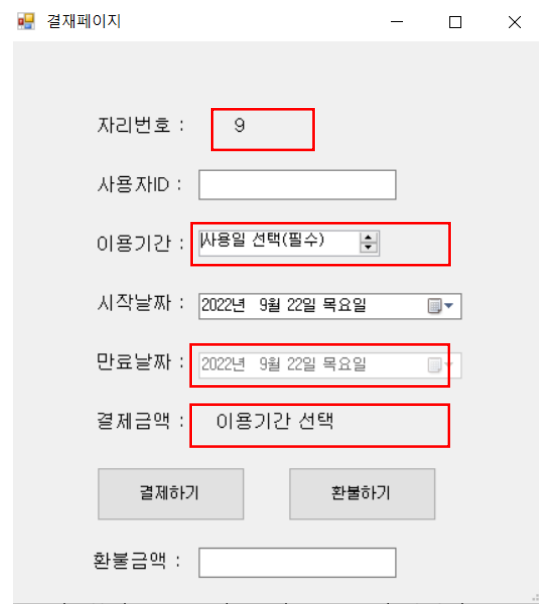


Personal 클릭시 화면

1버튼 = 1좌석으로 구성
흰색 - 빈자리
분홍색 - 1주일 이내 만료
군청색 - 이용중(1주일내 만료 아님)



Multi-Room 클릭시 화면



각 좌석 클릭시 연동

자리번호와 방번호는 자동으로 메모리 저장하여 output됨
이용기간 설정에 따라 만료 날짜 금액이 연동
호실(구분)에 따라 다른 가격 측정, 시작날짜 조정 가능

Part 2, 오류 및 수정 변경 저장

정상적인 등록 과 이슈 발생시 메세지창 연동(이용좌석 클릭시 메세지창 팝업후 결제페이지 이동)

결제페이지

미등록 회원

자리번호 : 9

사용자ID : sdfas

이용기간 : 7일(1주일)

등록된 회원이 아닙니다. 등록 후 다시 결제를 진행해주세요

확인

결제하기

환불하기

환불금액 :

결제페이지

정상 등록

자리번호 : 9

사용자ID : reddde

이용기간 : 7일(1주일)

ID : reddde님이 201호 9번에
2022-09-22 오후 4:13:58~2022-09-29 오후 4:13:58까지 이용하실입니다.
결제 금액은 52000원입니다.

확인

결제하기

환불하기

환불금액 :

이용중 인 좌석 클릭

금일 이용중인 좌석입니다.

확인

결제페이지

환불

자리번호 : 6

사용자ID : reddde

이용기간 : 7일(1주일)

등록번호 3 ID : reddde님이 201호 6번에
2022-09-22 오후 4:13:38환불하셨습니다.
환불 금액은 8000원입니다.

확인

결제하기

환불하기

환불금액 : 8000

검색기간 : 2022년 9월 21일 수요일 ~ 2022년 9월 22일 목요일

검색 결과

	등록번호	ID	방호수	자리번호	등록일	결제금액	시작일	만료일	비고
▶	2	popo	201	7	2022-09-22	85000	2022-09-22	2022-10-07	
	3	reddde	201	6	2022-09-22	8000	2022-09-22	2022-09-22	환불
	4	popo	201	2	2022-09-22	8000	2022-09-22	2022-09-22	
	5	popo	201	1	2022-09-22	8000	2022-09-22	2022-09-22	
	6	reddde	201	9	2022-09-22	52000	2022-09-22	2022-09-29	

환불 시 비고에 환불확인 만료일을 환불기록 당일로 저장
아래 통계에 환불금, 매출금 변경

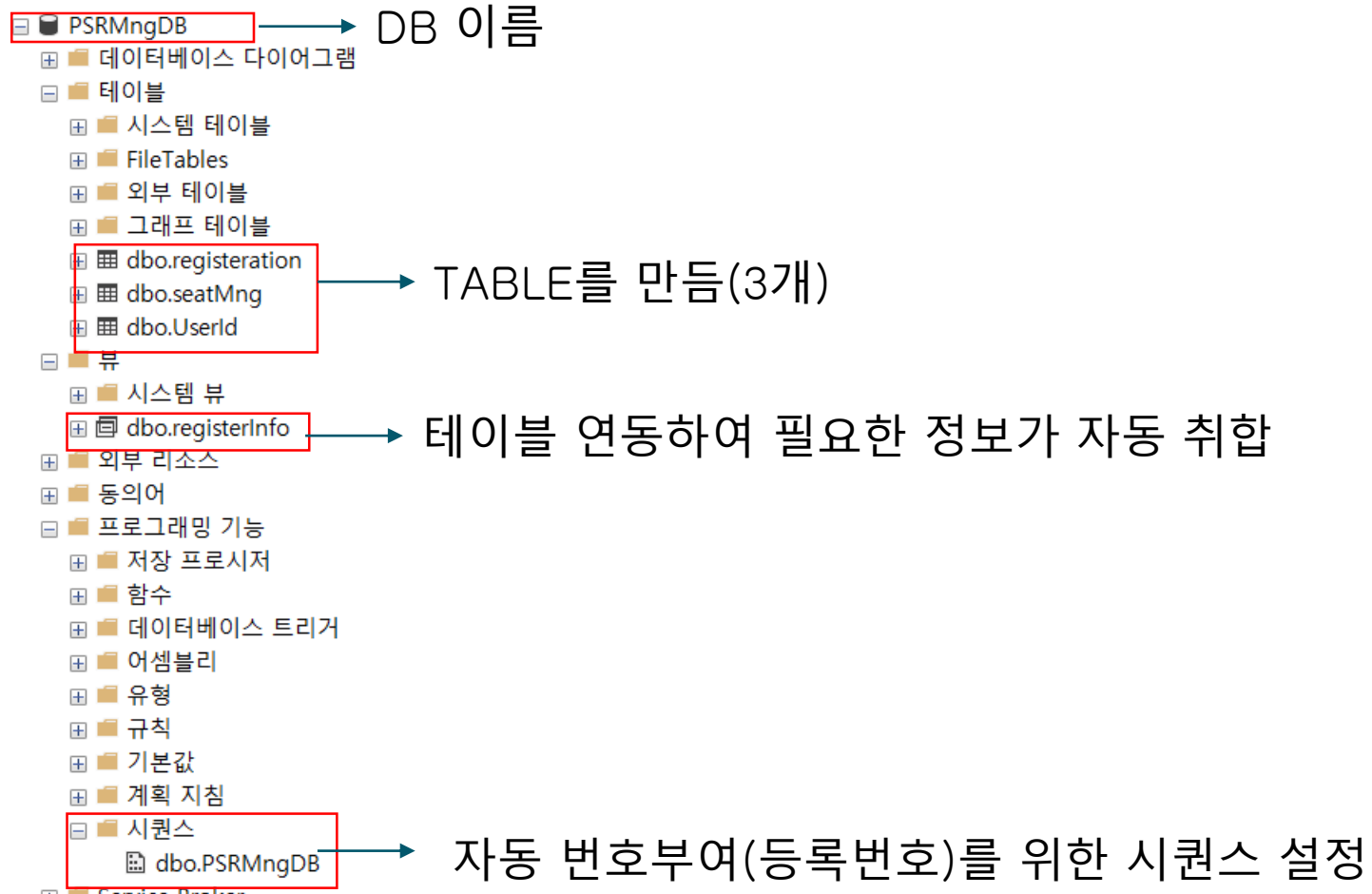
출 건수 : 5건 매출 : 161000원 환불금 : 8000원 총 매출 : 153000원

Part 3

로직 소개 (DB, Class)

Part 3, DB 제작

MSSQL에서 DB 테이블을 만들고 그외 기능들 제작



Part 3, DB 연동 및 DB테이블의 변환

MsSQL과 연동, 연동 및 검색은 SQL문으로 이용(암호화), 불러온

- ▶ C# DBconnect.cs
- ▶ C# DBregistration.cs
- ▶ C# DBseatMng.cs

→ TABLE의 수(3개) 만큼 연동 클래스 설정(각각 연동)

```
public class DBconnect
{
    private static SqlConnection conn = new SqlConnection();
    public static SqlDataAdapter da;
    public static DataSet ds;
    public static DataTable dt;

    참조 5개
    public static void ConnectDB()
    {
        conn.ConnectionString =
            string.Format("Data Source={0}); "
            + "initial Catalog = {1};" +
            "integrated Security = {2};" +
            "Timeout=3"
            "local", "PSRMngDB" "SSPI"); //학원에서는 PSRMngDB 로 연결해야함
        conn = new SqlConnection(conn.ConnectionString);
        conn.Open();
    }
}
```

DB연동

```
public static void newbeQuery(string userId, string name, string phone, DateTime birth, DateTime registrationDay)
{
    try
    {
        ConnectDB();
        SqlCommand cmd = new SqlCommand();
        cmd.Connection = conn;
        cmd.CommandType = CommandType.Text;
        string sqlcommand = "insert into UserId values(@p1, @p2, @p3, @p4, @p5)";
        cmd.Parameters.AddWithValue("@p1", userId);
        cmd.Parameters.AddWithValue("@p2", name);
        cmd.Parameters.AddWithValue("@p3", phone);
        cmd.Parameters.AddWithValue("@p4", birth);
        cmd.Parameters.AddWithValue("@p5", registrationDay);
        cmd.CommandText = sqlcommand;
        cmd.ExecuteNonQuery();
    }
    catch (Exception)
    {
    }
    finally
    {
        conn.Close();
    }
}
```

SQL문으로 필요기능
DB에서 설정

파라미터 암호로 변경하여
DB에서 구동

cmd.CommandText = "select * from UserId";

- ▶ C# Registration.cs
- ▶ C# seatMng.cs
- ▶ C# User.cs

→ 필요 게터세터 저장되는 DTO제작

Part 3, 리스트에 메모리 저장

MSSQL에서 SQL문으로 불러들인 정보를 리스트에 저장하여 메모리화

```
public static List<Registration> registrations = new List<Registration>();  
public static List<Registration> allregistration = new List<Registration>();  
public static List<Registration> expiredmembers = new List<Registration>();  
public static List<User> users = new List<User>();  
public static List<seatMng> nowseats = new List<seatMng>();  
... ..
```

→ 각 리스트에 필요와 목적에 따라 리스트 제작

```
public static void SeatLoad()  
{  
    try  
    {  
        DBregistration.seatMngselectQuery();  
  
        foreach (DataRow item in DBregistration.dt.Rows)  
        {  
            Registration seat = new Registration();  
  
            seat.registerNum = int.Parse(item["registerNum"].ToString());  
            seat.roomNum = int.Parse(item["roomNum"].ToString());  
            seat.seatNum = int.Parse(item["seatNum"].ToString());  
            seat.userId = item["userId"].ToString();  
            seat.startday = DateTime.Parse(item["startday"].ToString());  
            seat.endday = DateTime.Parse(item["endday"].ToString());  
            seat.pay = int.Parse(item["pay"].ToString());  
            registrations.Add(seat);  
        }  
    }  
    catch (Exception ex)  
    {  
        System.Windows.Forms.MessageBox.Show(ex.Message);  
        MessageBox.Show("자리테이블");  
    }  
}
```

→ Foreach문을 이용해 테이블을 행단위로 저장

→ Try catch 를 통해 이슈 발생 내역 확인 및 프로그램 폭파 방지

Part 3, 각기능로직

회원 등록 로직(이슈정리), 결제 페이지 가격자동 변경, 매출통계 정보 변경

회원 등록 로직

```
try
{
    if (DataManager.users.Exists((x) => x.Id == textBox_id.Text))
    {
        MessageBox.Show("이 ID는 이미 존재하는 ID입니다..");
    }
    else
    {
        User user = new User()
        {
            Id = textBox_id.Text,
            Name = textBox_name.Text,
            Phone = textBox_phone.Text,
            birth = dateTimePicker_birth.Value,
            registrationday = DateTime.Now
        };
        DataManager.users.Add(user);

        DataManager.Save
        (textBox_id.Text, textBox_name.Text,
        textBox_phone.Text, dateTimePicker_birth.Value, DateTime.Now);

        string contents = $"ID : {textBox_id.Text}에 " +
        $"{textBox_name.Text}님이 등록하셨습니다.";

        dataGridView1.DataSource = null;
        dataGridView1.DataSource = DataManager.users;
        WriteLog(contents);
        MessageBox.Show(contents);

        Close();
    }
}
catch (Exception)
{
    string contents= $"ID등록이 실패하셨습니다.";
    WriteLog(contents);
    MessageBox.Show(contents);
}
```

결제페이지

-이용 날짜에 따른 가격 변동

```
if (roomNum == 201 || roomNum == 202 || roomNum == 203)
{
    if (domaInUpDown_day.SelectedItem.Equals("1일"))
    {
        label_money.Text = "8,000원";
        billing = 8000;
        dateTimePicker_end.Value = dateTimePicker_start.Value;
    }
    else if (domaInUpDown_day.SelectedItem.Equals("7일(1주일)"))
    {
        label_money.Text = "52,000원";
        billing = 52000;
        dateTimePicker_end.Value = dateTimePicker_start.Value.AddDays(7);
    }
    else if (domaInUpDown_day.SelectedItem.Equals("15일"))
    {
        label_money.Text = "85,000원";
        billing = 85000;
        dateTimePicker_end.Value = dateTimePicker_start.Value.AddDays(15);
    }
    else if (domaInUpDown_day.SelectedItem.Equals("30일"))
    {
        label_money.Text = "150,000원";
        billing = 150000;
        dateTimePicker_end.Value = dateTimePicker_start.Value.AddDays(30);
    }
}

if (domaInUpDown_day.SelectedItem.Equals("1일"))
{
    label_money.Text = "6,000원";
    billing = 6000;
    dateTimePicker_end.Value = dateTimePicker_start.Value;
}
else if (domaInUpDown_day.SelectedItem.Equals("7일(1주일)"))
{
    label_money.Text = "40,000원";
    billing = 40000;
    dateTimePicker_end.Value = dateTimePicker_start.Value.AddDays(7);
}
else if (domaInUpDown_day.SelectedItem.Equals("15일"))
{
    label_money.Text = "75,000원";
    billing = 75000;
    dateTimePicker_end.Value = dateTimePicker_start.Value.AddDays(15);
}
else if (domaInUpDown_day.SelectedItem.Equals("30일"))
{
    label_money.Text = "120,000원";
    billing = 120000;
    dateTimePicker_end.Value = dateTimePicker_start.Value.AddDays(30);
}
```

매출 통계 쿼리

-통계 변경 로직

```
private void dateTimePicker_start_ValueChanged_1(object sender, EventArgs e)
{
    dataGridView_detail.DataSource = null;
    DataManager.registerLoad(dateTimePicker_start.Value.AddDays(-1), dateTimePicker_end.Value);
    dataGridView_detail.DataSource = DataManager.allregistration;
    sumsale();
    refund();
    label_totalcount.Text = DataManager.allregistration.Count.ToString() + "건";
    label_sale.Text = sales.ToString() + "원";
    label_refund.Text = refundcash.ToString() + "원";
    label_finalsale.Text = (sales - refundcash).ToString() + "원";
}

public int sumsale()
{
    foreach (var list in DataManager.allregistration)
    {
        sales += list.pay;
    }
    return sales;
}

//참조 2개
public int refund()
{
    foreach (var list in DataManager.allregistration)
    {
        refundcash += list.refundpay;
    }
    return refundcash;
}
```


Part 3, 결제 (좌석 등록) 로직

자리 버튼을 누른후 DB저장 및 이후 다시 클릭 시 연동

입력된 정보가 메모리를 통해 DB로 저장

```
else if (DataManager.users.Exists((x) => x.Id == textBox_id.Text))
{
    try
    {
        Registration registrations = new Registration()
        {
            userId = textBox_id.Text,
            roomNum = roomNum,
            seatNum = seatNum,
            pay = billing,
            startday = dateTimePicker_start.Value,
            endday = dateTimePicker_end.Value
        };
        DataManager.registrations.Add(registrations);

        DataManager.Save(textBox_id.Text, roomNum, seatNum, DateTime.Now, dateTimePicker_start.Value, dateTimePicker_end.Value, billing, "");
        DataManager.seatInfo(seatNum, textBox_id.Text, dateTimePicker_start.Value, dateTimePicker_end.Value);
        string contents = $"ID : {textBox_id.Text}님이 {roomNum}호 {seatNum}번에 \n" +
            $"{dateTimePicker_start.Value}~{dateTimePicker_end.Value}까지 이용하십니다. \n 결제 금액은 {billing}원입니다.";

        WriteLog(contents);
        MessageBox.Show(contents);
        Close();
    }
    catch (Exception)
    {
    }
}
```

DB에서 자리번호(1차)와 날짜(2차)를 검색후 모두 해당시 작동

```
if (DataManager.registrations.Exists((x) => x.seatNum == seatNum))
{
    if (DataManager.registrations.Exists((x) => x.endday >= dateTimePicker_start.Value) || DataManager.registrations.Exists((x) => x.startday >= dateTimePicker_end.Value))
    {
        MessageBox.Show("금일 이용중인 좌석입니다.");
        Registration refund= DataManager.registrations.Find((x) => x.seatNum == seatNum);
        string expay = refund.pay.ToString();
        textBox_id.Text = refund.userId;
        dateTimePicker_start.Value = refund.startday;
        dateTimePicker_end.Value = refund.endday;
        textBox_id.Enabled = dateTimePicker_start.Enabled = dateTimePicker_end.Enabled = domainUpDown_day.Enabled=false;
        label_money.Text = expay;
        registernumber = refund.registerNum;
        paycash = refund.pay;
    }
}
```

주인 있는 자리 등록 방지
환불 시 제 검색 방지

Part 3, 좌석에 부여된 로직

시각적 효과 부여 및 좌석 버튼 클릭시 일어나는 메모리 저장 기능

```
seat[0] = b201_1; seat[1] = b201_2; seat[2] = b201_3; seat[3] = b201_4; seat[4] = b201_5; seat[5] = b201_6; seat[6] = b201_7; seat[7] = b201_8; seat[8] = b201_9; seat[9] = b201_10;
seat[10] = b202_11; seat[11] = b202_12; seat[12] = b202_13; seat[13] = b202_14; seat[14] = b202_15; seat[15] = b202_16; seat[16] = b202_17; seat[17] = b202_18; seat[18] = b202_19; seat[19] = b202_20;
seat[20] = b203_21; seat[21] = b203_22; seat[22] = b203_23; seat[23] = b203_24; seat[24] = b203_25; seat[25] = b203_26; seat[26] = b203_27; seat[27] = b203_28; seat[28] = b203_29; seat[29] = b203_30;

for (int i = 0; i < seat.Length; i++)
{
    DataManager.seatcolorQuery(i + 1);
    color(seat[i], DataManager.nowseats[0].endday);
    DataManager.nowseats.Clear();
}
```

배열을 통해 각 버튼 이름 저장
For문으로 모든 버튼에 color메소드적용

```
private void b201_2_Click(object sender, EventArgs e)
{
    DataManager.registrations.Clear();
    registration.seatNum = seatNum = 2;
    registration.roomNum = roomNum = 201;
    Form_charge temp = new Form_charge(seatNum, roomNum);
    temp.ShowDialog();
}
```

참조 1개

```
private void b201_3_Click(object sender, EventArgs e)
{
    DataManager.registrations.Clear();
    registration.seatNum = seatNum = 3;
    registration.roomNum = roomNum = 201;
    Form_charge temp = new Form_charge(seatNum, roomNum);
    temp.ShowDialog();
}
```

참조 1개

```
private void b201_4_Click(object sender, EventArgs e)
{
    DataManager.registrations.Clear();
    registration.seatNum = seatNum = 4;
    registration.roomNum = roomNum = 201;
    Form_charge temp = new Form_charge(seatNum, roomNum);
    temp.ShowDialog();
}
```

자리버튼 클릭시
각 버튼마다 방, 번호 지정
그리고 결제창 팝업

```
private void color(Button I, DateTime endday)
{
    if (endday == new DateTime()) // 빈자리
    {
        I.BackColor = ColorTranslator.FromHtml("#ffffff");
    }
    else if (DateTime.Now.AddDays(7) >= endday) //7일전
    {
        I.BackColor = ColorTranslator.FromHtml("#ff9fc3");
    }
    else if (DateTime.Today == endday) // 만료일
    {
        I.BackColor = ColorTranslator.FromHtml("#ff0000");
    }
    else if (DateTime.Now.AddDays(7) < endday) //이용중
    {
        I.BackColor = ColorTranslator.FromHtml("#9999ff");
    }
}
```

Part 3, 프로그램 사용 기록 저장

MSSQL에서 DB 테이블을 만들고 그외 기능들 제작

```
public static void printLog(string contents)
{
    DirectoryInfo di = new DirectoryInfo("PSRMngHistory");
    if (di.Exists == false)
        di.Create();
    using (StreamWriter w =
        new StreamWriter("PSRMngHistory\PSRMngHistory.txt", true))
    {
        w.WriteLine(contents);
    }
}
```

특정 내용을 외부에 저장

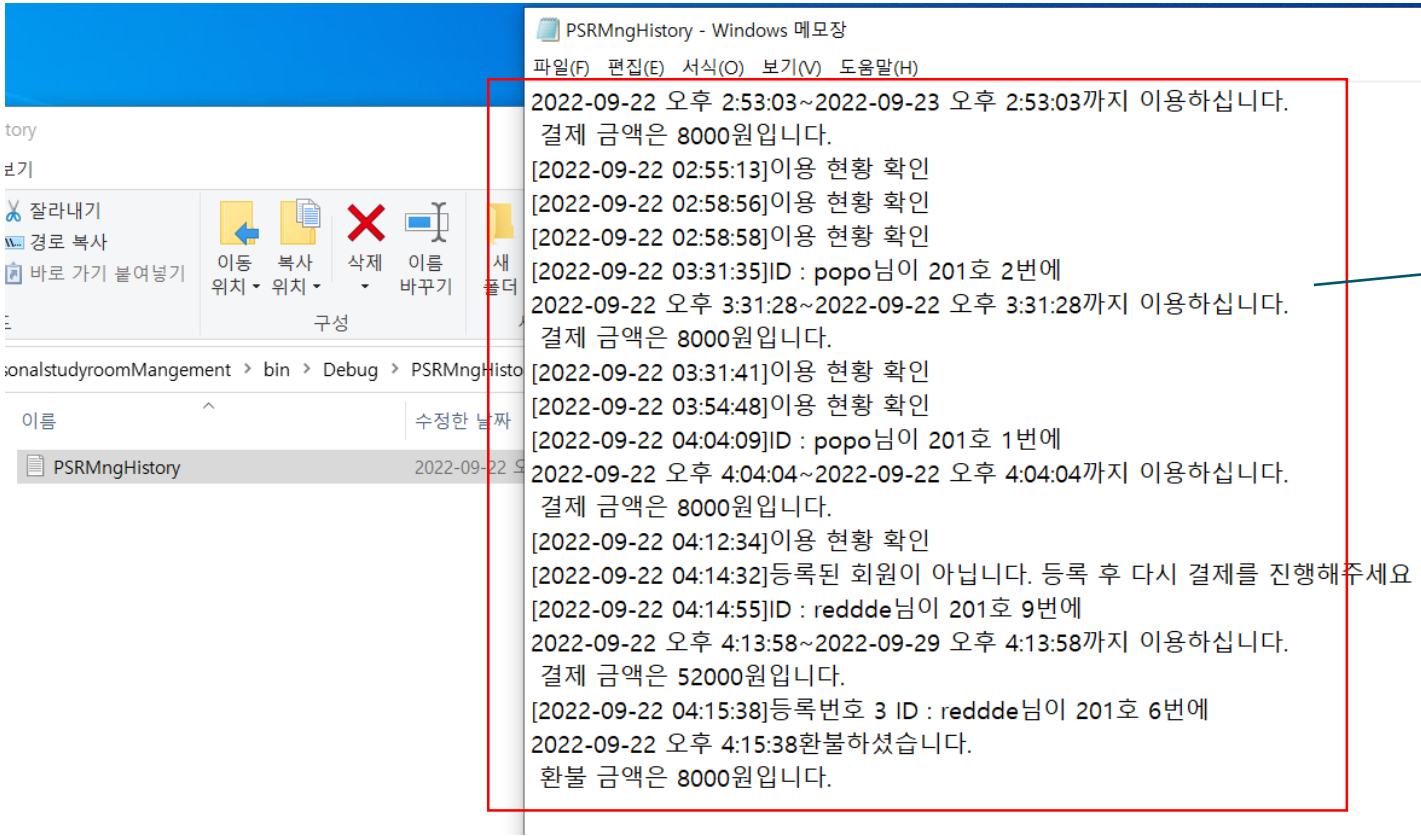
참조 1개

```
public static void Save(string userId, string name, string phone, DateTime birth, DateTime registrationDay)
{
    try
    {
        DBconnect.newbeQuery(userId, name, phone, birth, registrationDay);
    }
    catch (Exception)
    {
    }
}
```

VS > PersonalstudyroomMangement > PersonalstudyroomMangement > bin > Debug				
이름	수정한 날짜	유형	크기	
PSRMngHistory	2022-09-22 오후 2:43	파일 폴더		

Part 3, DB 제작

MSSQL에서 DB 테이블을 만들고 그외 기능들 제작



클릭 내용, 메시지 박스 내용
모두 저장

로그기록 저장 목적 : 이슈 발생시 어느 부분 이슈인지 확인가능

Part 4

기타 (추가)

Part 4, 추가필요 내역 및 아쉬운 내용

001

대부분의 기능이
버튼 클릭 시 발생

(Key Down 로직 필요)

002

C# 로직이 아닌
SQL문의 비중이 크다

003


자리 이동 쿼리가 없다.
환불-재등록 방식 강제

결제 금액 변경은
프로그래머가 변경해야만
한다
(관리자 불가)


004

좌석 개수 및 구조 변경
등 몇몇 부분에 존재
유연성이 떨어지고
프로그램 이용자의 자율
성이 높지 않다

(하지만 프로그래머의
경우 수정이 간편하다)



이 프로그램은 소규모로 지정석으로만 운영되는 독서실을 가정하고 만든 프로그램입니다.
관리자가 확인하며 현 상황을 볼 수 있도록 준비해보았습니다.





감사합니다

