

---

# 프로젝트 실습(NFL)2 결과 보고서

---

연구과제명 | PolyDormi

---



---

학과 | 컴퓨터정보과

---

담당교수 | 김은아

---

제작자 | 정인환

---

작성년월 | 2019년 06월 25일

---

# 작 품 설 명 서

작 품 명	PolyDormi		
학 과	컴퓨터정보과	지 도 교 수	김은아
제 작 자	정인환		
작 품 개 요	한국폴리텍대학 인천캠퍼스의 기숙사인 '한리관'과 '백두관'에 입사해 있는 학생들의 정보를 관리할 수 있다. 각 호실별로, 또는 학과별로 학생들의 정보를 조회 및 수정하여 효율적으로 학생정보를 관리하기 위해 개발하게 되었다.		
동 작 기 능	Form	DB	
	- C#을 이용한 폼 - 각 컨트롤을 이용한 이벤트	- 학생 정보 저장 - 시설 정보 저장 - 쿼리문을 이용한 정보 관리	
활 용 방 법	각 생활관 에 입사해 있는 학생정보 및 시설관리 정보를 입력할 수 있으며, 입력된 정보들을 수정, 관리를 하며 관리할 수 있다.		
제 작 기 술	C#, MySQL		

# 목 차

## I. 서 론

- 가. 세부 계획표
- 나. 프로젝트 제작 동기

## II. 동작원리 및 관계이론

- 가. DB 설계
- 나. DB 연동
- 다. 쿼리문
- 라. Windows Form

## III. 작품구성

- 가. PolyDormi DataBase
  - 1. 한라관에서 사용하는 테이블
  - 2. 백두관에서 사용하는 테이블
- 나. Windows Form 구성 및 주요기능
  - 1. 시작화면
  - 2. 메인화면(한라관)
  - 3. 생성화면(시설)
  - 4. 생성화면(학생)
  - 5. 호실정보화면
  - 6. 학생검색기능
  - 7. 현황화면(통계)

## IV. 기타

- 가. 기대효과 및 활용방안
- 나. 보완해야 할 점
- 다. 참조 문헌

# I. 서 론

## 가. 세부 계획표

1. 기존 기숙사 관리 프로그램 조사 및 분석을 통한 좀 더 세부적인 디자인 확립
  - 관리자가 쉽게 정보들을 관리하기 위해 필요한 메뉴들을 생각하여 설계하였다. Visual Studio2017 이용하여 C#, Windows Form으로 UI를 설계하였으며, 각 메뉴에 맞게 컨트롤을 생성하였다.
2. 시간을 분배를 통한 기간선정 및 효율성 추구
  - 정해진 기간에 맞게 적절한 분배를 하고, 설계하여 주어진 과제를 효율적으로 진행할 수 있도록 하였다.
3. 언어 환경, 버전 확인 및 DB 설계
  - 개발자가 사용하고 있는 .NET Framework 및 SQL 등 환경과 버전에 따라 오류가 발생할 수 있다. 따라서 개발 전 환경에 맞게 준비하고, 프로그램에 필요한 DB를 설계하였다.
4. 해당 작품 Windows Form설계
  - Windows Form을 설계하기 위한 환경으로는 Visual Studio 2017을 사용하였다.
5. 디자인 완성 및 소스 입력
  - PolyDormi 프로젝트에 맞는 디자인, 이미지 등을 고려하여 한국폴리텍대학의 로고를 사용하였다.

## 나. 프로젝트 제작 동기

현재 한국폴리텍대학 인천캠퍼스의 기숙사에 있는 학생들의 정보들을 관리하기 위해 DataBase를 이용해 프로그램을 제작하여 현재 입사해 있는 학생들의 기본정보, 배정되어 있는 호실정보, 원하는 학생의 정보, 입사했던 모든 학생들의 입사 및 퇴사날짜 등의 정보를 파악하기 위해 제작하게 되었다. 뿐만 아니라 각 층에 존재하는 시설에 대한 정보 관리 기능도 추가하여 시설이 고장 나게 되면 시설에 대한 정보와 고장날짜를 입력하고, 수리가 완료되면 수리날짜를 입력하여 시설 관리를 좀 더 원활하게 할 수 있게 하기 위해 제작하였다.

## Ⅱ. 동작원리 및 관계이론

### 가. DB 설계

원하는 정보를 DB를 사용할 프로그램에 맞게 DB를 설계하고, 생성하는 작업이다.

Query 1

halla\_stu

+

테이블 이름

halla\_stu

엔진

InnoDB

데이터베이스

polydormi

문자 인코딩

utf8

클레이션

utf8\_general\_ci

1 새로 행

2 인덱스

3 외래 키

4 Check Constraint

5 Advanced

6 SQL Preview

☐ 언어 옵션 숨기기

<input type="checkbox"/> 행 이름	데이터 타입	길이	디폴트 (초기화)	PK?	Null이 아님?	부호 없이?	자동 증가?	Zerofill?
<input type="checkbox"/> room	varchar	10		<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/> stu_name	varchar	20		<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/> major	varchar	50		<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/> stu_num	varchar	11		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/> grade	varchar	11		<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/> sex	varchar	10		<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/> jointime	varchar	10		<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/> outtime	varchar	10	~	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/>				<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

그림 1. 한라관의 학생정보를 저장하는 테이블

### 나. DB 연동

C#에서 MySQL을 사용하기 위해서는 C#과 MySQL의 커넥터인 'MySQL Connector/Net'을 설치해야 한다. 그리고 연결을 위해 Mysql.Data항목을 참조 추가한다. 아래는 DB연동을 위한 코드이다.

```

public static String url = "SERVER=LOCALHOST; USER=root; DATABASE=polydormi;" +
    "PORT=3306; PASSWORD=0159; CHARSET=utf8";
    // DB접속 URL 설정 - SERVER : DB주소, USER : ID명, DATABASE : DB명, PORT : TCP 포트번호
    // PASSWORD : 비밀번호, 언어설정

private MySqlConnection mConnection; // DB접속
private MySqlCommand mCommand; // 쿼리문
private MySqlDataReader mDataReader; // 실행문
    
```

그림 2. DB접속코드

## 다. 퀴리문

데이터 관리를 위해서 INSERT, SELECT, DELETE, UPDATE 등과 같은 쿼리문을 이용해서 정보들을 입력, 수정, 삭제, 불러오기 등의 기능을 수행한다.

```

MySQLCommand sel = new MySqlCommand("SELECT * FROM halla_stu WHERE outtime= '" + "" + "'ORDER BY room", mConnection);
MySQLCommand upd = new MySqlCommand("UPDATE halla_stat SET stuA = '" + "" + "'WHERE room='" + room + "'", mConnection);
MySQLCommand del = new MySqlCommand("DELETE FROM halla_stu WHERE stu_num = '" + num + "'", mConnection);
MySQLCommand ins = new MySqlCommand("INSERT INTO halla_stu values('" + room + "','"
                                + name + "','"
                                + major + "','"
                                + num + "','"
                                + grade + "','"
                                + sex + "','"
                                + date + "','"
                                + "" + "')", mConnection);

```

그림 3. 쿼리문 예시

## 라. Windows Form

사용자가 컨트롤을 사용해서 원하는 화면을 구성하고 이벤트 발생시 처리하고자 하는 작업을 이벤트 처리기에 기술하는 방식으로 프로그래밍이 진행된다.

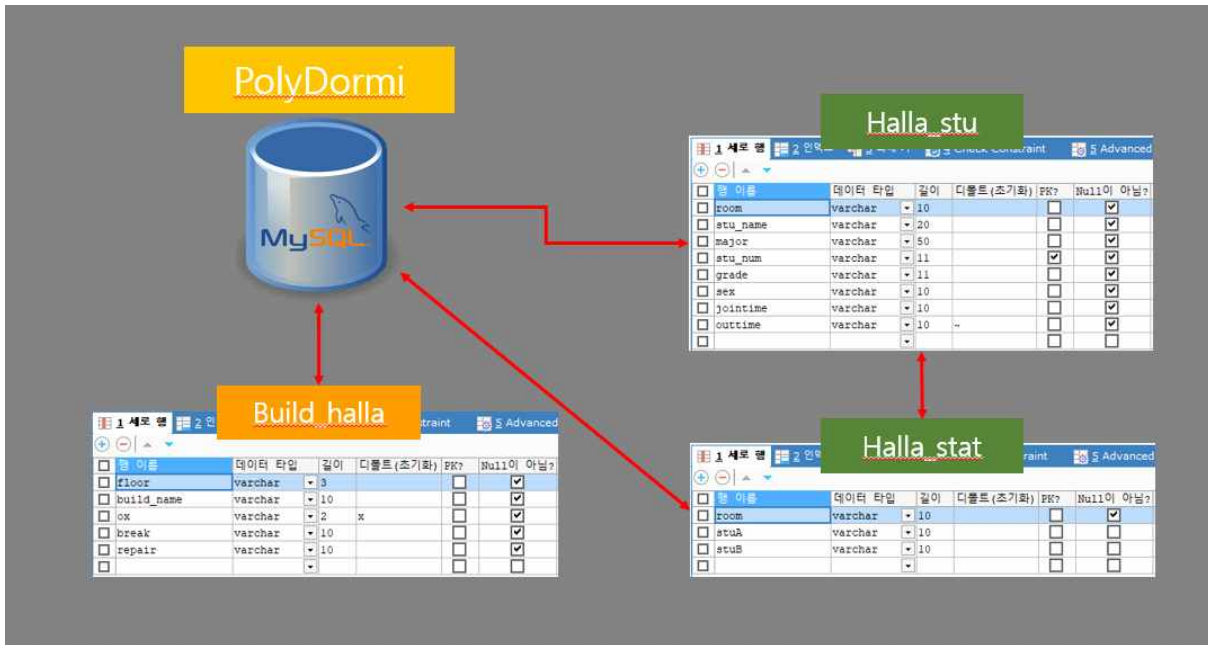
[illegible]

그림 4. 한라관의 기본정보들이 출력되는 폼

### Ⅲ. 작품구성

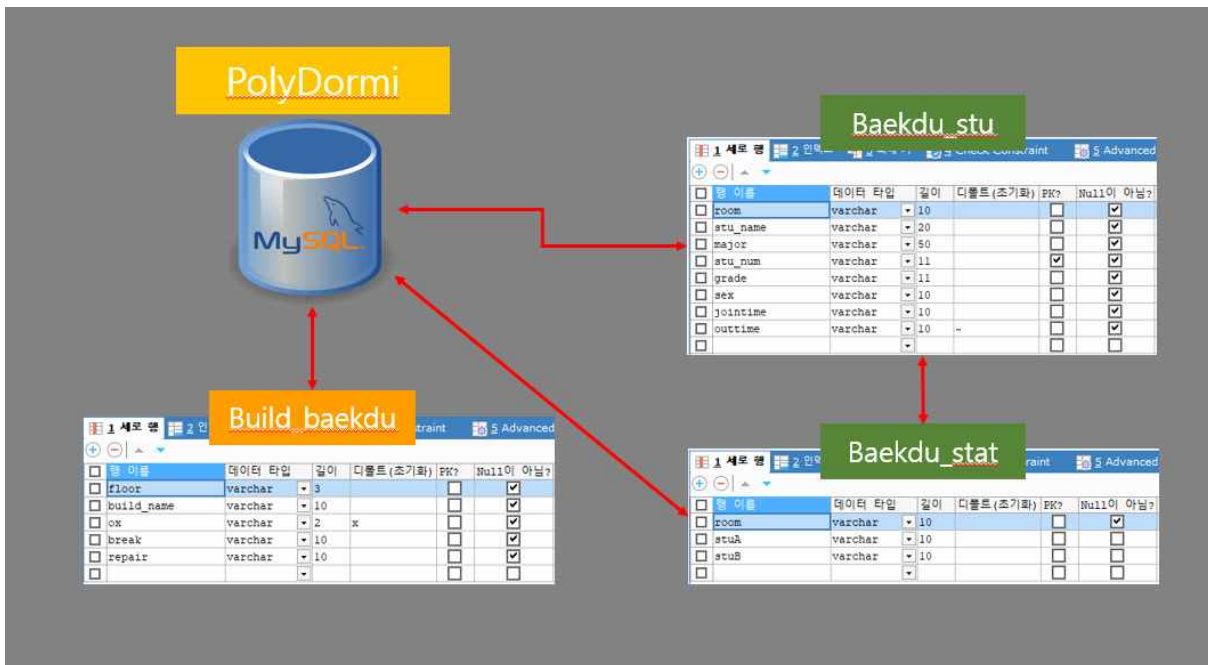
#### 가. PolyDormi DataBase

1)한라관에서 사용하는 테이블



Halla\_stu : 학생정보가 들어있는 테이블 Halla\_stat : 호실정보가 들어있는 테이블

Build\_halla : 시설정보가 들어있는 테이블



2)백두관에서 사용하는 테이블

Halla\_stu : 학생정보가 들어있는 테이블 Halla\_stat : 호실정보가 들어있는 테이블

Build\_halla : 시설정보가 들어있는 테이블

## 나. Windows Form 구성 및 주요기능

### 1. 시작화면

#### \*프로젝트 실행 후 시작화면

원하는 기숙사의 버튼을 누르면 선택된 기숙사의 폼이 열린다.



#### \*핵심 코드

```
private void button1_Click(object sender, EventArgs e)
{
    Halla h = new Halla();
    h.Show();
}

private void button2_Click(object sender, EventArgs e)
{
    Baekdu b = new Baekdu();
    b.Show();
}
```

### 2. 메인화면(한라관)

#### \*선택한 기숙사의 학생 및 시설정보 화면

(고장 시설정보, 입사 학생정보, 각 호실정보, 학생 검색 기능)

DB에 저장된 학생정보와 시설정보를 SELECT문을 이용해서 불러와 해당하는 리스트뷰에 추가한다.

호실	이름	학번	학과	학년	성별
101	정인환	1502100484	컴퓨터정보	2학년	남
101	아무개	1903400241	전기에너지시스템	1학년	남



### \*핵심 코드

```
private void selectdb()
{
    mConnection = new MySqlConnection(url);
    Student_View.Items.Clear();
    Build_View.Items.Clear();
    mConnection.Open();
    //입사학생 정보
    MySqlCommand cmd = new MySqlCommand("SELECT * FROM halla_stu WHERE outtime= '" + "" + "'ORDER BY room", mConnection);

    MySqlDataReader dr = cmd.ExecuteReader();
    while (dr.Read() == true)
    {
        list = new ListViewItem(dr[0].ToString());
        list.SubItems.Add(dr[1].ToString());
        list.SubItems.Add(dr[3].ToString());
        list.SubItems.Add(dr[2].ToString());
        list.SubItems.Add(dr[4].ToString());
        list.SubItems.Add(dr[5].ToString());
        Student_View.Items.Add(list);
    }
    dr.Close();
    //시설정보
    MySqlCommand bl = new MySqlCommand("select * from build_halla WHERE repair= '" + "" + "'ORDER BY floor", mConnection);

    MySqlDataReader drbl = bl.ExecuteReader();
    while (drbl.Read() == true)
    {
        list = new ListViewItem(drbl[0].ToString());
        list.SubItems.Add(drbl[1].ToString());
        Build_View.Items.Add(list);
    }
    drbl.Close();
    mConnection.Close();
}
```

## 3.생성화면(시설)

### \*고장난 시설의 정보를 입력하는 화면

각 컨트롤에 입력된 값들을 INSERT문을 이용해서 build\_halla 테이블에 저장한다.

### \*핵심 코드

```
mConnection.Open();
MySqlCommand cmd = new MySqlCommand("INSERT INTO build_halla values('"+ floor + "','"+ //층
+ build_name.Text + "','"+ //시설 이름
+ "x" + "','"+ //수리여부
+ date + "','"+ //고장난 날짜
+ "" + "')", mConnection); //수리된 날짜

cmd.Connection = mConnection;
cmd.ExecuteNonQuery();
mConnection.Close();
```

#### 4.생성화면(학생)

**\*새로 입사하는 학생의 정보를 입력하는 화면**

각 컨트롤에 입력된 값들을 INSERT문을 이용해서 Halla\_stu 테이블에 저장한다. 그리고 호실정보에 학생이 있는지 검사하고 호실에 빈 자리가 없으면 오류발생

한라관...

입사 학생 정보

호실 :

이름 :

학과 :

학번 :

학년 : ☐ 1학년 ☐ 2학년

성별 : ☐ 남 ☐ 여

<<입사날짜>>

2019년 06월 26일

생성      달기

### \*핵심 코드

```
MySqlCommand cmd1 = new MySqlCommand("INSERT INTO halla_stu values('" + new_Room.Text + "','" //호실
+ new_Name.Text + "','" //이름
+ new_Class.Text + "','" //학과
+ new_Num.Text + "','" //학번
+ grade + "','" //학년
+ sex + "','" //성별
+ date + "','" //입사날짜
+ "'" + "'" + ")", mConnection);
```

```
cmd1.Connection = mConnection;  
cmd1.ExecuteNonQuery();
```

```
MySqlCommand same = new MySqlCommand("SELECT * FROM halla_stat where room = '" + newRoom.Text + "'", mConnection);
MySqlDataReader dr = same.ExecuteReader();
```

```
while (dr.Read() == true)
{
    dr.Close();
}
```

```
if (!Equals(stuB, ""))
```

```
MessageBox.Show("해당 호실에 이미 2명의 학생이 있습니다!", "경고", MessageBoxButtons.OK, MessageBoxIcon.Error);
return;
```

```
if (Equals(stuA, ""))
```

```
MySqlCommand cmd2 = new MySqlCommand("UPDATE halla_stat SET stuA='" + new_Name.Text +  
                                     " WHERE room = '" + new_Room.Text + "'", mConnect ion);
```

```
cmd2.ExecuteNonQuery();
```

```
InsertDB();
```

```
up = false;
```

```
if (!Equals(stuA, ""))
```

```
MySQLCommand same2 = new MySQLCommand("SELECT * FROM halla_stu WHERE stu_num=" + new_Num.Text + "", mConnection);
MySQLDataReader dr1 = same2.ExecuteReader();
```

```
while (dr1.Read() == true)
```

```
stunum = drl[3].ToString();
```

```
dr1.Close();
```

```
up = true;
```

## 5.호실정보화면

\*선택한 호실에 배정된 학생의 정보를 보여주는 화면

호실을 지정하면 SELECT문을 이용하여 해당되는 호실의 학생에 대한 정보를 불러온다.

Figure 1-1 shows two forms side-by-side. The left form has a yellow background and a thin border. It contains the following fields: 호실 (Room) with value 101, 이름 (Name) with value 정민환, 학과 (Department) with value 컴퓨터정보, 학년 (Grade) with radio buttons for 1학년 and 2학년 (2학년 is selected), and 성별 (Gender) with radio buttons for 남 and 여 (남 is selected). The right form has a white background and a thin border. It contains the following fields: 호실 (Room) with value 101, 이름 (Name) with value 아무개, 학과 (Department) with value 전기에너지시스템, 학년 (Grade) with radio buttons for 1학년 and 2학년 (1학년 is selected), and 성별 (Gender) with radio buttons for 남 and 여 (남 is selected).

### \*핵심 코드

```
mConnection = new MySqlConnection(url);
mConnection.Open();
MySqlCommand cmd = new MySqlCommand("SELECT * FROM halla_stu WHERE room=" + roomvalue +
                                     "AND outtime=" + "" + "", mConnection);
MySqlDataReader dr = cmd.ExecuteReader();

while (dr.Read() == true) {
    mConnection.Close();
}
```

## 6. 학생 검색 기능

\*입사해 있는 학생들을 학번을 기준으로 검색하는 기능

각 기숙사의 메인화면에 있는 기능인 학생 검색 기능이다.  
학생의 학번을 입력하면 해당되는 학생의 정보가 출력된다.

학생 검색

학번 : 1502100484

검색

정인환

학생의 정보

학번 : 1502100484

학년 : 2학년

학과 : 컴퓨터정보

호실 : 101

성별 : 남

입사 : 2019-06-25

### \*핵심 코드

```
//학생 검색
private void search_Click(object sender, EventArgs e)
{
    string searchnum = stu_search.Text;
    if (Equals(searchnum, ""))
    {
        MessageBox.Show("학번을 입력해주세요!", "오류", MessageBoxButtons.OK, MessageBoxIcon.Warning);
        return;
    }
    MySqlConnection con = new MySqlConnection("server=localhost;database=halla;user=root;password=12345678");
    con.Open();
    MySqlCommand ser = new MySqlCommand("SELECT * FROM halla_stu WHERE stu_num = '" + searchnum + "' AND outtime = '' + '' + ''", con);
    MySqlDataReader dser = ser.ExecuteReader();
    int i = 0;
    while (dser.Read() == true)
    {
        info_name.Text = (dser[1].ToString());
        info_num.Text = (dser[3].ToString());
        info_grade.Text = (dser[4].ToString());
        info_major.Text = (dser[2].ToString());
        info_room.Text = (dser[0].ToString());
        info_sex.Text = (dser[5].ToString());
        info_join.Text = (dser[6].ToString());
        i++;
    }
    dser.Close();
    ser.Connection = con;
    ser.ExecuteNonQuery();
    con.Close();
    if (i == 0)
    {
        MessageBox.Show("학번을 확인해주세요!", "오류", MessageBoxButtons.OK, MessageBoxIcon.Warning);
    }
}
```

## 7. 현황화면(통계)

\*학생 및 시설에대한 현황을 통계로 보여주는 기능

(기숙사 현황, 입사/퇴사 현황, 시설 고장/수리 현황)

한라관 현황

한라관 학생 및 시설 현황

데이터 초기화

나가기

입사/퇴사 현황

이름	학번	학과	호실	입사 날짜	퇴사 날짜
마무개	1903400241	전기에너지시스템	101	2019-06-25	~
정인환	1502100484	컴퓨터정보	101	2019-06-25	~

학생 삭제

시설 고장/수리 현황

층	시설 정보	수리	고장 날짜	수리 날짜
1F	세탁기	x	2019-06-25	
1F	전자레인지	x	2019-06-25	
3F	TV	x	2019-06-25	
4F	403호 샤워호스	x	2019-06-25	
4F	408호 세면대	x	2019-06-25	

시설 삭제

기숙사 현황

호실

A

B

101

정인환

마무개

### \*핵심 코드

```

//입.퇴사 현황
MySQLCommand cmd1 = new MySqlCommand("select * from halla_stu ORDER BY outtime, stu_name", mConnection);
MySQLDataReader dr1 = cmd1.ExecuteReader();

while (dr1.Read() == true)
{
    list = new ListViewItem(dr1[1].ToString());
    list.SubItems.Add(dr1[3].ToString());
    list.SubItems.Add(dr1[2].ToString());
    list.SubItems.Add(dr1[0].ToString());
    list.SubItems.Add(dr1[6].ToString());
    list.SubItems.Add("-");
    list.SubItems.Add(dr1[7].ToString());
    Student_View.Items.Add(list);
}
dr1.Close();

//거숙사 현황
MySQLCommand cmd2 = new MySqlCommand("select * from halla_stat ORDER BY room", mConnection);
MySQLDataReader dr2 = cmd2.ExecuteReader();

while (dr2.Read() == true)
{
    list = new ListViewItem(dr2[0].ToString());
    list.SubItems.Add(dr2[1].ToString());
    list.SubItems.Add(dr2[2].ToString());
    Now_View.Items.Add(list);
}
dr2.Close();

//시설 현황
MySQLCommand cmd3 = new MySqlCommand("select * from build_halla ORDER BY ox desc, floor :", mConnection);
MySQLDataReader dr3 = cmd3.ExecuteReader();

while (dr3.Read() == true)
{
    list = new ListViewItem(dr3[0].ToString());
    list.SubItems.Add(dr3[1].ToString());
    list.SubItems.Add(dr3[2].ToString());
    list.SubItems.Add(dr3[3].ToString());
    Build_View.Items.Add(list);
}
dr3.Close();
mConnection.Close();

```

## IV. 기타

### 가. 기대효과 및 활용방안

PolyDormi를 이용하여 학생들의 정보를 좀 더 수월하게 관리하고, 어떤 시설들이 많이 고장 나는지 빈도를 파악하여 시설 수리를 더 원활하게 할 수 있다.

현재 기숙사는 개방과 폐쇄를 하는 시스템이다. PolyDormi와 카드단말기를 연동시켜 카드에 각 학생들의 정보를 저장하고, 각 기숙사 출입문에 장치를 달아서 카드를 이용해 출입을 가능하게 한다면, 따로 개방과 폐쇄를 하지 않고도 외부인출입이나 학생이 현재 기숙사에 있는지도 알 수 있는 시스템으로 발전할 수 있다.

### 나. 보완해야 할 점

디자인 : 컨트롤간의 밸런스와, 로고의 크기, 글씨체

기능 : 현재 기숙사에 존재하는 별점제도 추가, 일 단위로 학생의 외박여부를 파악하는 기능 추가

### 다. 참조문헌

MySQL DB생성 및 테이블 관리 : 은노기의 JSP 웹 프로그래밍 입문

C#과 MySQL연동 : <https://m.blog.naver.com/lorch/220049159180>

ListView와 MySQL연동 코드(쿼리문) : <https://blog.naver.com/idhhf123/120201686548>

C# inputbox : <https://www.c-sharpcorner.com/blogs/inputbox-in-c-sharpnet1>