C# PROJECT

4 TEAM MEMBER : 김동진 김지환 민경환 소강석 채예진

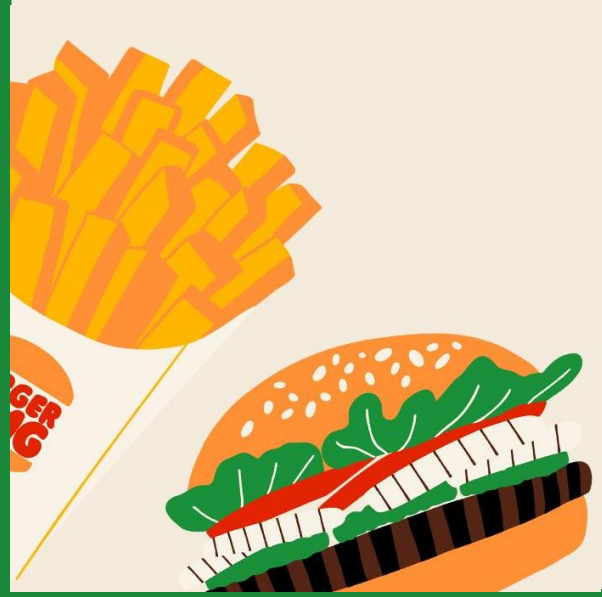
BURGERKINGCONTENTS

01. 우리가 계획한 여정

02. 우리가 생각했던 점

03. 우리가 해낸것들

04. 잘한 점과 못한 점들



01. 우리가 계획한 여정

- 프로젝트 수행 목적
  - API를 통한 데이터 확보 및 편리한 UI&UX를 적용한 프로그램 제작
  - C# window forms와 MySQL DB를 연동하여 데이터 저장 및 불러오기

- 프로젝트 개발 목표
  - 버거 킹 앱에서 사용되는 주요기능 구현

로그인
가까운 매장 찾기
햄버거 주문
고객 정보 확인
주문 확인

02. 우리가 생각했던 점

- 사용자가 편리하게 이용할 수 있는 프로그램을 만들어 보자 !
  - 사용자에게 가까운 매장 위치 제공을 위한 지도 API 활용
  - 직관적이고, 앙증맞은 UI 제작
  - 주문확인이 쉽도록 홈화면에 주문 확인서 추가

- 회원정보, 주문, 메뉴는 DB로 관리하자 !
  - MySQL C# 연동을 통한 데이터 관리
  - 회원정보, 주문, 메뉴 테이블 생성

03. 해낸것들(1) - C#/SQL 연동

DB접속 함수를 만들어 DB에서 데이터를 가지고 오는 코드

```
public static void ConnectDB()
{
    try
    {
    string pwd = "1234";
    string strConn = "Server=localhost;port=2421;Database=burgerking;Uid=root;Pwd=" + pwd + ";Charset=utf8";
    Myconn = new MySqlConnection(strConn);
    Myconn.Open();
    }
    catch (Exception e)
    {
        throw new Exception(e.Message);
    }
}
```

03. 해낸것들(2) - 로그인

```
try
{
    connection.Open();
    // insertQuery = $"SELECT pw FROM member where id='{id}'";
    var command = new MySqlCommand(insertQuery, connection);
    command.ExecuteNonQuery();
    MySqlDataReader reader = command.ExecuteReader();
    if (reader.HasRows)
        {
        while (reader.Read())
        {
            dbpw = reader["pw"].ToString();
        }
        if (dbpw == pw)
        {
            MessageBox.Show("로그인 성공!");
            Order_Form3 oder3 = new Order_Form3();
            Hide();
            // Home_Form1 : 메인화면
            Home_Form1 home = new Home_Form1();
            DataManager.session = id;
            home.SetText(id);
            home.ShowDialog();
            Close();
        ...
```

- 사용자가 입력한 ID 데이터와 DB 데이터의 일치여부를 확인
- 로그인에 성공할 경우 `home.SetText(id)`를 통해 `id` 값 유지

03. 해낸것들(3) - 햄버거 주문

```
public Menu mymenu
public List<Menu> selectMenu = new List<Menu>();
private void select_menu(object sender, EventArgs e)
{
    mymenu = new Menu();
    UserControl1 mySelect = (sender as UserControl1);
    mymenu.rownum = int.Parse(mySelect.Name.Split('_')[2]);

    Form1 form1 = new Form1(mymenu);

    if(mymenu.rownum ≤ 12)
        form1.ShowDialog();

    else
    {
        mymenu.name = DataManager.menulist[mymenu.rownum-1].name;
        mymenu.price = DataManager.menulist[mymenu.rownum - 1].price;
        mymenu.category = DataManager.menulist[mymenu.rownum - 1].category;
    }

    DataManager.myList.Add(mymenu);
    guna2DataGridView1.DataSource = null;

    if (DataManager.myList.Count > 0)
        guna2DataGridView1.DataSource = DataManager.myList;
}
```

- 사용자가 선택한 pictureBox의 Name 데이터에서 식별코드(rownum)를 가져와서 mymenu.rownum에 담는다.

- 식별코드(rownum)에 해당하는 데이터를 리스트(menulist)에서 찾고 데이터를 객체(mymenu)에 담는다.

03. 해낸것들(4) - 주문고객정보

```
1 private void callInfo()
2 {
3     string id = session;
4     string dbname = "";
5     string dbphone_num = "";
6     string dbaddress = "";
7     var connectionString = "server=localhost;port=2421;user=root;database=burgerking;password=1234";
8     var connection = new MySqlConnection(connectionString);
9     string insertQuery = $"SELECT * FROM member where id='{id}'";
10
11    try
12    {
13        connection.Open();
14        var command = new MySqlCommand(insertQuery, connection);
15        command.ExecuteNonQuery();
16
17        MySqlDataReader reader = command.ExecuteReader();
18        if (reader.HasRows)
19        {
20            while (reader.Read())
21        {
22            dbname = reader["name"].ToString();
23            dbphone_num = reader["phone_num"].ToString();
24            dbaddress = reader["address"].ToString();
25        }
26        gunaLabel8.Text = dbname;
27        gunaLabel9.Text = dbphone_num;
28        gunaLabel10.Text = dbaddress;
29        ...
```

- DB(member)에 접속해서 주문 고객 정보(name, phone_num, address)를 가져옴
- 각 정보로 Label의 Text 속성을 바꿔준다.(gunaLabel8.Text, gunaLabel9.Text, gunaLabel10.Text)

03. 해낸것들(5) - 지도 API

```
<body>
    <div id="map" style="width:750px;height:350px;"></div>
    <script src="//dapi.kakao.com/v2/maps/sdk.js?appkey=10c466959bfcce2c464bf99a4988acb3"></script>
    <script>
    // 카카오 지도 API 내에서 버거킹마커 생성 (총 21개)
    var positions = [{
                        title: '대구율하점', // 1
                        latlng: new kakao.maps.LatLng(35.866346, 128.696278)
                    },
                    {
                        title: '대구만촌점', // 2
                        latlng: new kakao.maps.LatLng(35.857015, 128.648724)
                    },
                    {
                        title: '대구삼성창조캠퍼스', // 3
                        latlng: new kakao.maps.LatLng(35.884415, 128.594601)
                    },
                    ...
```

- 카카오 지도 API를 불러오는 HTML 파일에서 `position`을 통해 지도 `marker` 좌표를 설정한다.

03. 해낸것들(5) - 지도 API

```
1 public static List<Locale> Search(string qstr = null)
2 {
3     List<Locale> mls = new List<Locale>();
4     String distance = "10000";//qstr;
5     // 키워드가 "대구 버거킹"이고, 설정된 좌표와 가까운 순으로 정렬시키는 쿼리
6     string query = $"https://dapi.kakao.com/v2/local/search/keyword.json?page=1&query=%EB%B2%84%EA
  %B1%B0%ED%82%B9&y=35.877015&x=128.626125&sort=distance&radius={distance}&size=5";
7
8     ...
9
10    for (int i = 0; i < length; i++)
11    {
12        string lname = docs[i]["place_name"];
13        double x = double.Parse(docs[i]["x"]);
14        double y = double.Parse(docs[i]["y"]);
15        int dis = int.Parse(docs[i]["distance"]);
16        string phone = docs[i]["phone"];
17        string address = docs[i]["road_address_name"];
18
19        mls.Add(new Locale(lname, y, x, dis, phone, address));
20    }
21    return mls;
22    }
23 }
```

- “대구 버거킹”, 설정된 좌표와 가까운 순으로 정렬시키는 쿼리

- 카카오 API에서 반환된 조건과 일치되는 데이터를 `docs`에 담고 각 데이터를 `mls` 객체에 담아 `return` 한다.

04. 잘한 점과 못한 점

잘한 점

- 앙증맞고, 직관적이며 사용자에게 친화적인 UI 제공
- 수업시간에 배운 지도 API를 활용하여 사용자 마커 추가, 가까운 매장 찾기 등을 구현
- DB 연동 및 데이터를 활용해서 회원관리와 주문하기 구현
- 활발하고 열정적인 팀워크

못한 점

- 회의를 통해 세심한 방향(파트분배, DB구성, 프로그램 구조) 설정 미흡
- 비효율적인 코드구조(코드가독성, MVC 패턴 등 활용 미흡)

BURGER KING

BURGER KING
BURGER KING
BURGER KING

ABCDEFGHI
JKLMNOPQRS
TUVWXYZ

HOME OF THE WHOPPER

HEARTY MELTY