

2014년도 1학기 프로그래밍 프로젝트

KU_NYAM 팀프로젝트 개발 계획서(2)

201111360 손준익

201311268 김예현

201311269 김제현

목 차

1. 전체 시스템 구조	5-1
1) 데이터베이스 내부	
2) 서버구조	
3) 클라이언트 구조	
4) Use-Case Diagram	
5) Sequence Diagram	
6) Class Diagram	
2. 각 모듈별 상세 설명	5-3
1) 회원가입	
2) 로그인	
3) 구역선택	
4) 식당검색	
5) 식당선택	
6) 마이페이지	
7) 로그아웃/프로그램 종료	
3. 데이터베이스 상세 설명	5-3
1) ER-Diagram	
2) 테이블 상세 정의	

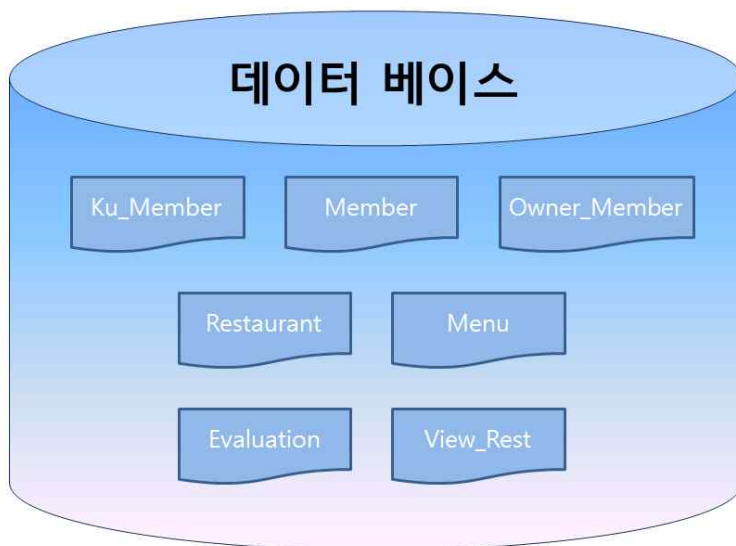
1. 전체 시스템 구조

System은 크게 Server와 Client로 구성되어 있다. 기본 구조는 아래와 같다.



기본적으로는, 안드로이드를 OS로 탑재하고 있는 스마트폰이 클라이언트가 되며, 앱을 실행 시에 필요할 때마다 서버를 경유해서 데이터베이스에 저장되어 있는 데이터를 이용하고, 데이터의 등록, 수정, 삭제 등의 갱신작업이 이루어진다.

1) 데이터베이스 내부 구조



데이터베이스의 내부구조로써, 사용하는 테이블의 목록이다.

- a) 회원을 건국대학교 학생회원, 일반회원, 사업자회원 3가지로 분류하였으므로 테이블 역시 Ku_Member, Member, Owner_Member의 3가지로 분할되었다.
- b) 이번 프로젝트에서 핵심인 식당의 정보와 관련해서는, 식당정보와 해당식당의 메뉴로 분류해서, 테이블로는 Restaurant, Menu의 2가지로 분할되었다.

- c) 부가적으로, 각 테이블간의 1:N, N:M의 관계를 연결시켜주기 위해 파생된, Evaluation, View_Rest의 테이블이 생성되었다.

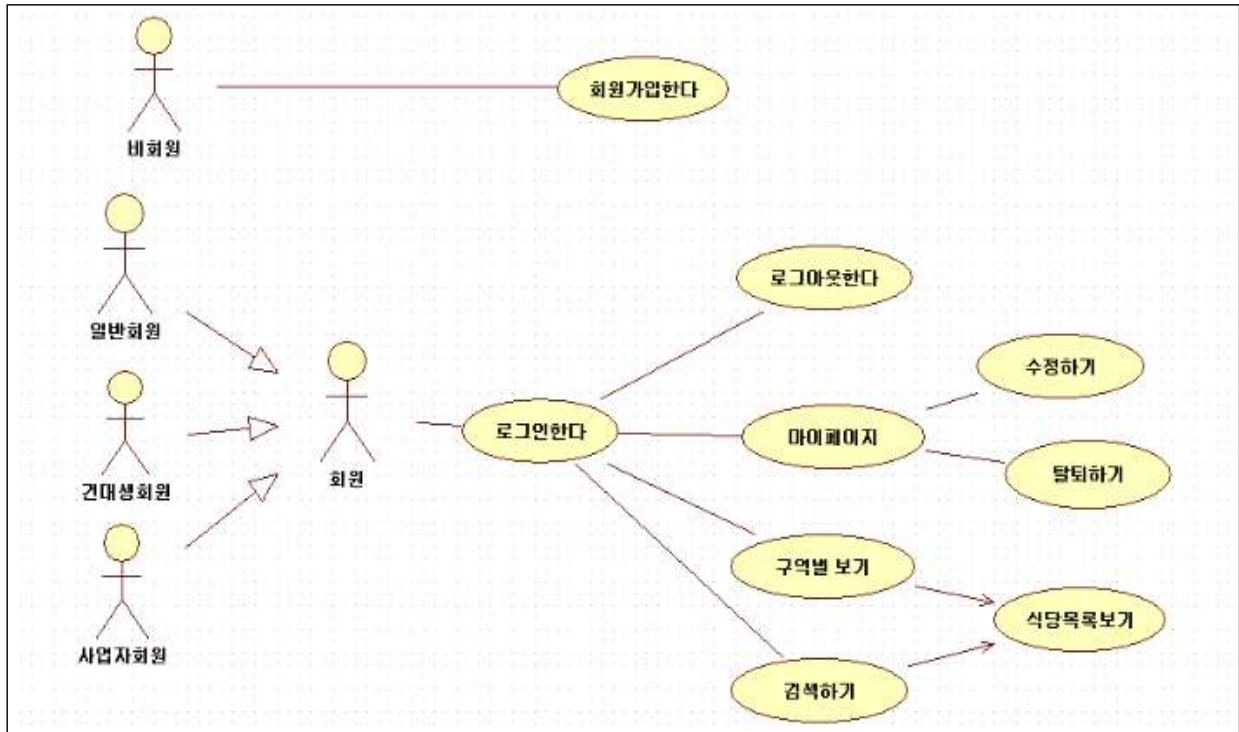
2) 서버 구조



3) 클라이언트 구조



4) Use-Case Diagram



사용자의 관점에서 바라본 프로그램의 전체 Use-Case Diagram이다.
사용자가 어떤 기능을 이용할 수 있는냐에 초점을 두고 작성을 했다.

a) 만약 사용자가 회원이 아닐 경우, 회원가입밖에 할 수 없다.

(a) 앱을 실행시킬 때 로딩화면 후의 첫 화면[로그인전 메인화면]에서는 회원가입이나 로그인 외에도, ‘건대맛집 구역별보기’ 나 ‘건대 식당 검색의 내용도 선택이 가능하나, 이는 로그인 후에 이용할 수 있는 기능들을 먼저 보여주는 것일 뿐이다.

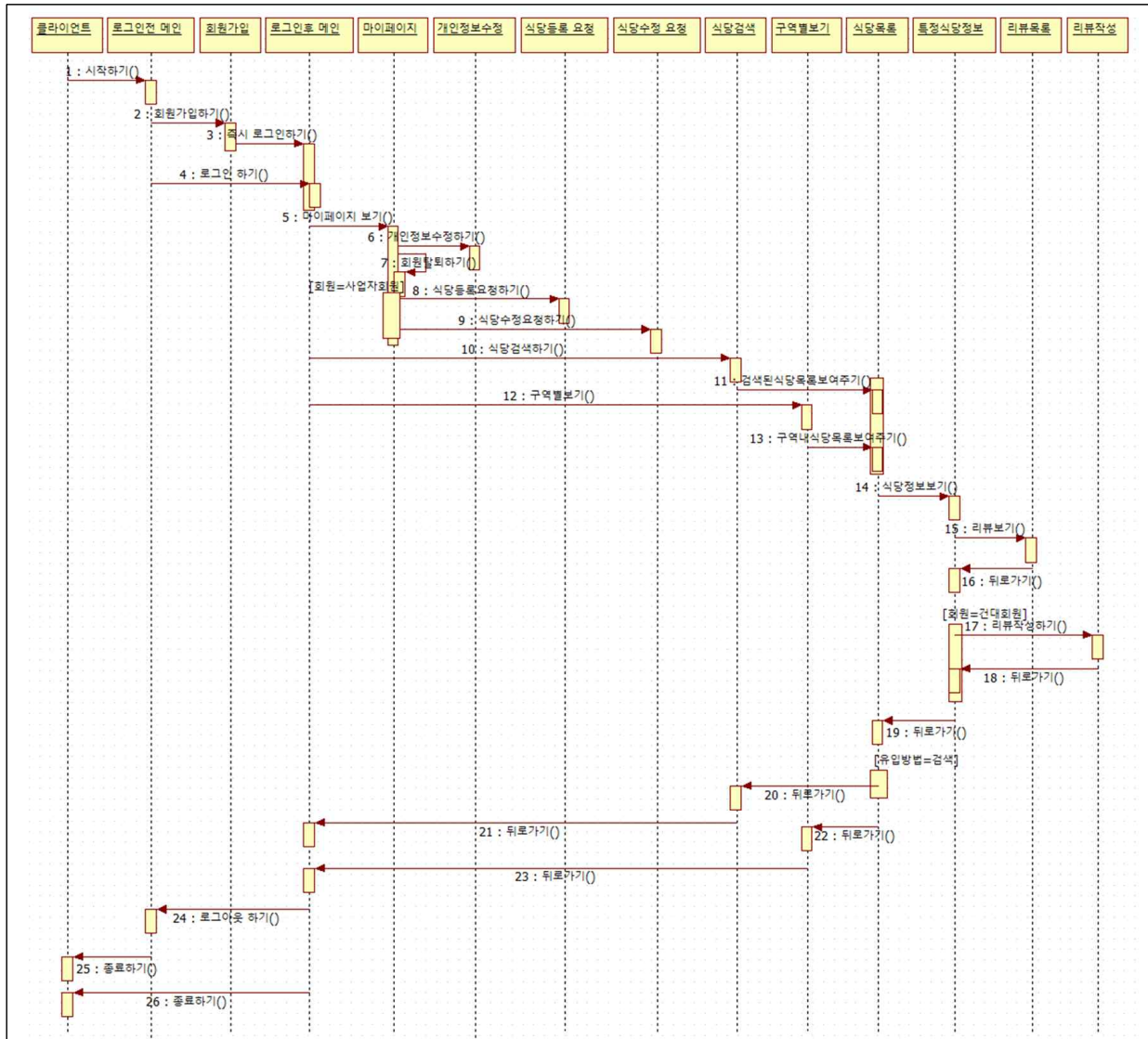
(b) 만약 로그인 전에 이러한 기능들을 선택할 경우, ‘로그인 후에 이용하세요!’라는 팝업창이 뜨고, 아무 일도 일어나지 않는다.

b) 만약 사용자가 회원이 맞을 경우, 우선 로그인을 통해 로그인후 메인화면으로 이동할 수가 있다. 이후에는 다음과 같은 기능을 이용할 수가 있다.

(a) 공통적으로는 마이페이지보기, 식당검색하기, 구역별보기, 식당선택하기, 리뷰보기의 기능을 이용할 수 있다.

(b) 이외에도 회원에 따라, 건대 회원인 경우는 리뷰보기에서의 리뷰작성하기, 사업자회원인 경우는 마이페이지에서의 식당등록/수정하기로 추가적인 기능을 이용할 수 있으나, ‘사용자’가 선택 가능한 기능 중 공통된 기능만 표현했다.

5) Sequence Diagram



개발자의 관점에서 바라본 프로그램의 전체 Sequence Diagram이다.

a) 전체적인 앱의 흐름을 모듈별로 분류해서 표현했다.

(a) 해당 화면에서 실행할 수 있는 기능의 흐름을 표현했다.

(b) 회원에 따라, 할 수 있는 기능도 표현했다. [건대회원인 경우 리뷰작성 가능 등]

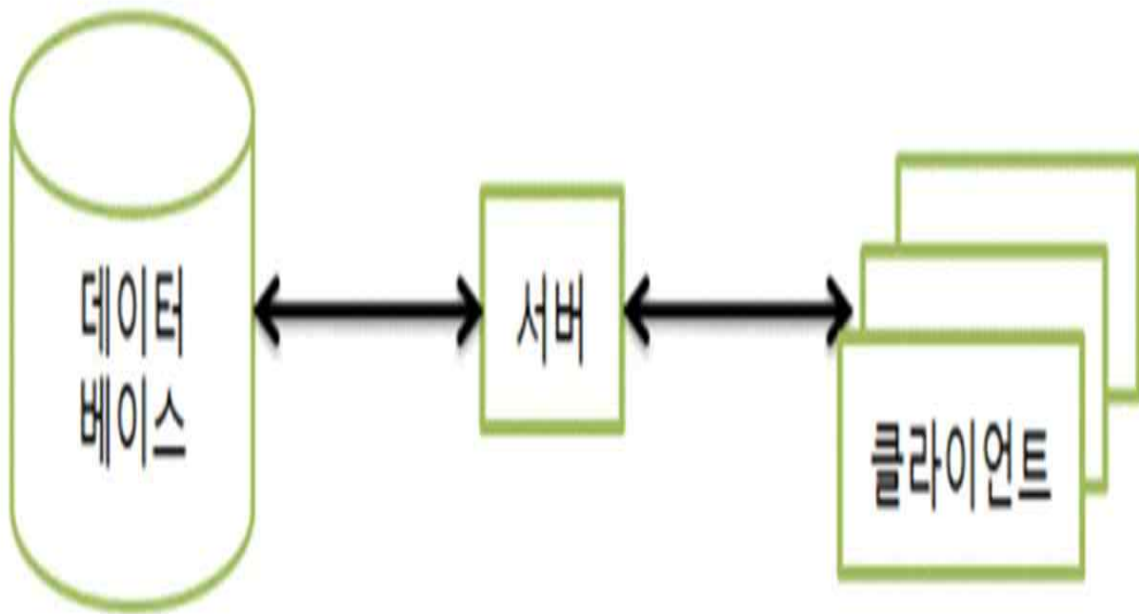
(c) 이 앱이 정상대로 실행된다는 가정 하에서 예외처리는 표시하지 않았다.

b) 세부적인 절차는 표현하지 않았다.

(a) 해당 화면에서 어떤 과정을 거쳐서 요구한 기능을 수행할지는 각 모듈별로의 Sequence Diagram에서 표현했기에, 세부적인 절차를 표현하지 않은 것이다.

c) 전체적인 관점이므로 Process-Model이라고도 불리기도 한다.

6) Class Diagram



2. 각 모듈별 상세 설계

1) 회원가입

a. 모듈기능설명

회원가입을 할 수 있는 모듈이다.

- 회원가입의 종류는 건국대학교 학생 회원, 일반회원, 사업자회원 이렇게 총 3가지로 나뉘어져 있으며, 각 회원별로 이용 할 수 있는 기능이 조금씩 다르다.

- 로그인전 메인화면에서 회원가입 버튼을 누르게 되면, 회원가입 화면이 뜨면서 회원 3종류중 하나를 택함으로써 회원가입 종류를 선택 할 수 있고, 이 후 그 안의 가입란에 세부내용을 입력 할 수 있다.

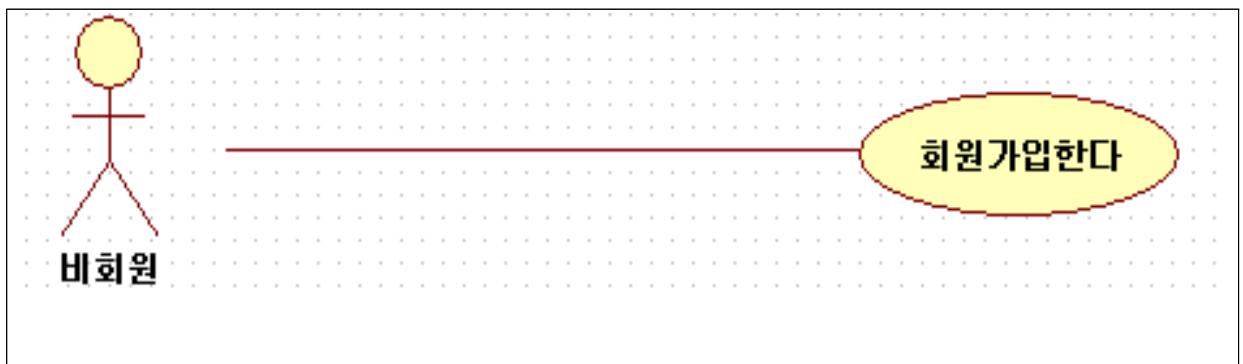
- 건국대학교 학생 회원인 경우는 학번, 학과, 이름, 핸드폰번호, 성별을 기입하고, 일반회원인 경우는 이름, 핸드폰번호, 성별을 기입하며, 사업자 회원인 경우는 사업자 이름, 사업자 번호, 나이, 사업자 경력을 기입해야 가입할 수 있게 했다. 참고로 사업자 회원은. 이번 앱 안에 들어갈 맛집 식당의 사장님이라고 볼 수 있다.

- 확인버튼을 클릭시, 우선 로그인 정보가 누락되어 있는지부터 확인하고, 누락되어 있지 않다면, 기존의 회원명단중 로그인 정보가 중복되는 회원이 존재하는지 확인하게 되며, 중복되지 않을 경우에만 회원가입이 진행된다.

- 회원가입이 진행될 경우 회원번호가 지정이 되는데, 회원번호는 건국대학교학생 회원, 일반회원, 사업자회원 따로따로 가입한 순서대로 지정이 된다. 이는 나중에 각 회원을 확인하는 기본키(Primary Key)로 사용된다.

- 본래 학번이나 사업자번호 같은 경우, 정확한 인증을 해야 하나, 이러한 구현은 능력 밖의 내용이라서 생략하게 됐다. 양심껏 본인인증을 했다고 가정하고 진행하기로 했다.

b. Use-Case Diagram



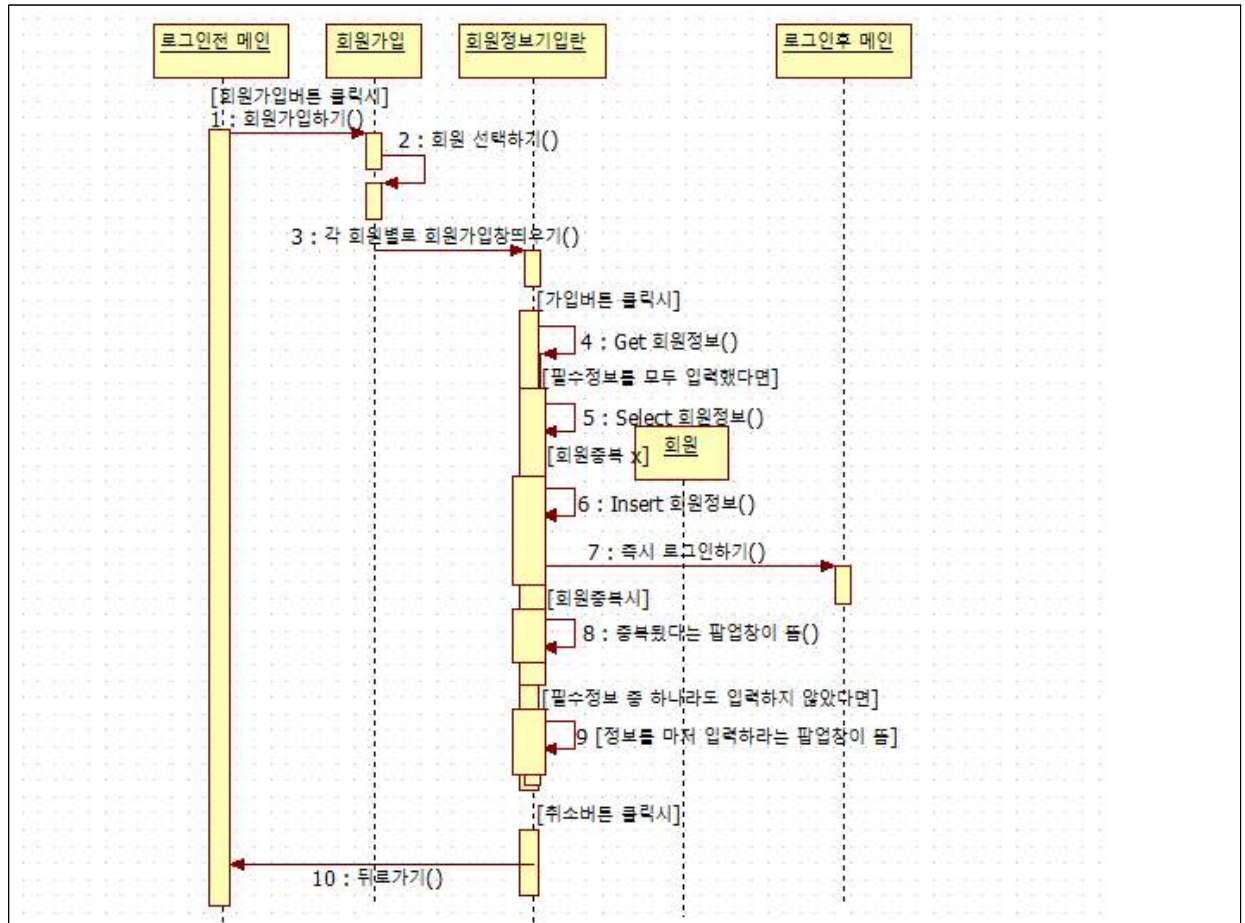
- ### c. Class Diagram

--

- | | |
|--|--|
| | |
| | |
| | |
| | |
| | |

- | | |
|--|--|
| | |
| | |
| | |
| | |
| | |

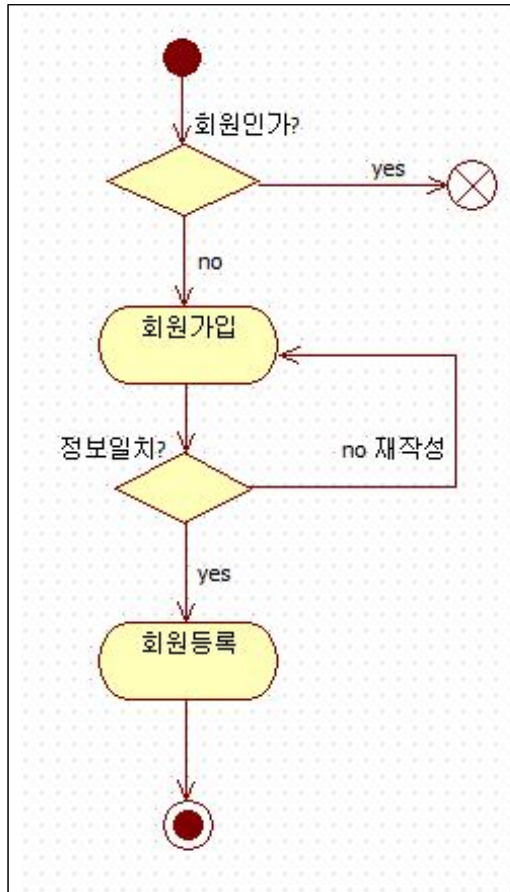
d. Sequence-Diagram



- a) 로그인전의 메인화면에서 회원가입버튼을 클릭시에 실행된다.
- (a) 회원가입 화면이 뜨면서 건국대학교 학생 회원, 일반회원, 사업자회원의 3종류중 하나를 택함으로써 회원가입 종류를 선택 할 수 있다.
 - (b) 회원을 선택 시, 각 회원별로 회원가입화면이 뜨게 된다. 각각 별개의 화면이 뜨는 것이나, 회원가입의 기능적인 측면에서 볼 때, 이러한 분류를 표시하는 것 보다는 결국 공통적인 흐름을 표시하는 것이 좋다고 생각해서 해당 그림처럼 표현했다.
 - (c) 각 회원가입화면의 기입란에 입력을 한 후에 가입버튼을 클릭 시, 우선 기입란의 정보를 받아오고, 필수정보[로그인정보]의 누락이 있는지 확인하며, 누락이 있을시 ‘로그인정보 기입란이 누락되어있습니다.’ 라는 팝업창이 뜨며 취소된다.
 - (d) 이후, 기존회원의 로그인 정보와 중복되는지를 확인하며, 회원정보가 중복 시에는 ‘이미 회원가입되어 있습니다.’ 라는 팝업창이 뜨면서 취소된다.

(e) 중복되지 않았을 경우에 회원정보를 저장하게 되며, 이후 즉시 로그인후 메인화면으로 이동하게 된다.

e. Activity-Diagram



a) 순서도(Flow Chart)처럼 표현해 본 Activity-Diagram이다.

(a) 사용자가 회원일 경우, 로그인을 이용하면 되므로 바로 빠져나간다.

(b) 사용자가 회원이 아닐 경우, 회원가입을 해야만 하는데, 가입란에 내용을 작성한 후 가입버튼을 클릭하였을시, 필수정보의 누락 및 중복이 되어있는 경우 [건국대학교 학생의 경우 학번과 이름], 다시 재작성을 해야 하는 방식으로 반복된다.

(c) 필수정보의 누락 및 중복이 없는 경우, 회원등록이 이루어진다.

2) 로그인

a. 모듈기능설명

로그인을 할 수 있는 모듈이다.

- 로그인의 종류 역시 회원가입과 마찬가지로 건국대학교 학생 회원, 일반회원, 사업자회원 이렇게 총 3가지로 나뉘어져 있다.

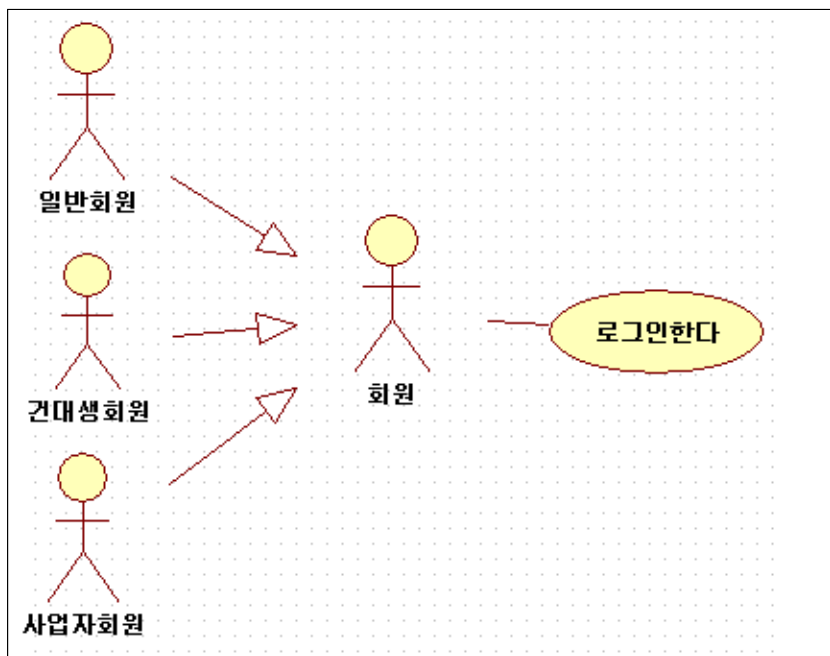
- 로그인전 메인화면에서 로그인 버튼을 누르게 되면, 로그인 화면이 뜨면서 회원 3종류중 하나를 택함으로써 로그인방식을 선택 할 수 있고, 이 후 그 안의 입력란에 세부내용을 입력 할 수 있다.

- 건국대학교 학생 회원인 경우는 로그인시 학번이랑 이름을 입력하고, 일반회원인 경우는 핸드폰번호, 이름을 입력하며, 마지막으로 사업자 회원인 경우는 사업자 번호, 이름을 입력해야 로그인이 가능하도록 했다.

로그인시, 회원이 맞는지 확인하는 절차가 있으며, 회원이 맞으면 로그인후 마이페이지로 이동하고, 맞지 않으면 로그인실패와 관련된 팝업창이 뜨게 된다.

- 로그인 후 메인화면은 로그인 전 메인화면이랑 다르게 밑에 회원가입, 로그인 버튼에서 마이페이지라는 버튼으로 바뀐다.

b. Use-Case Diagram



a) 사용자가 로그인을 진행할 시, 해당 회원을 선택한 후 로그인정보를 입력함으로써 로그인이 진행된다.

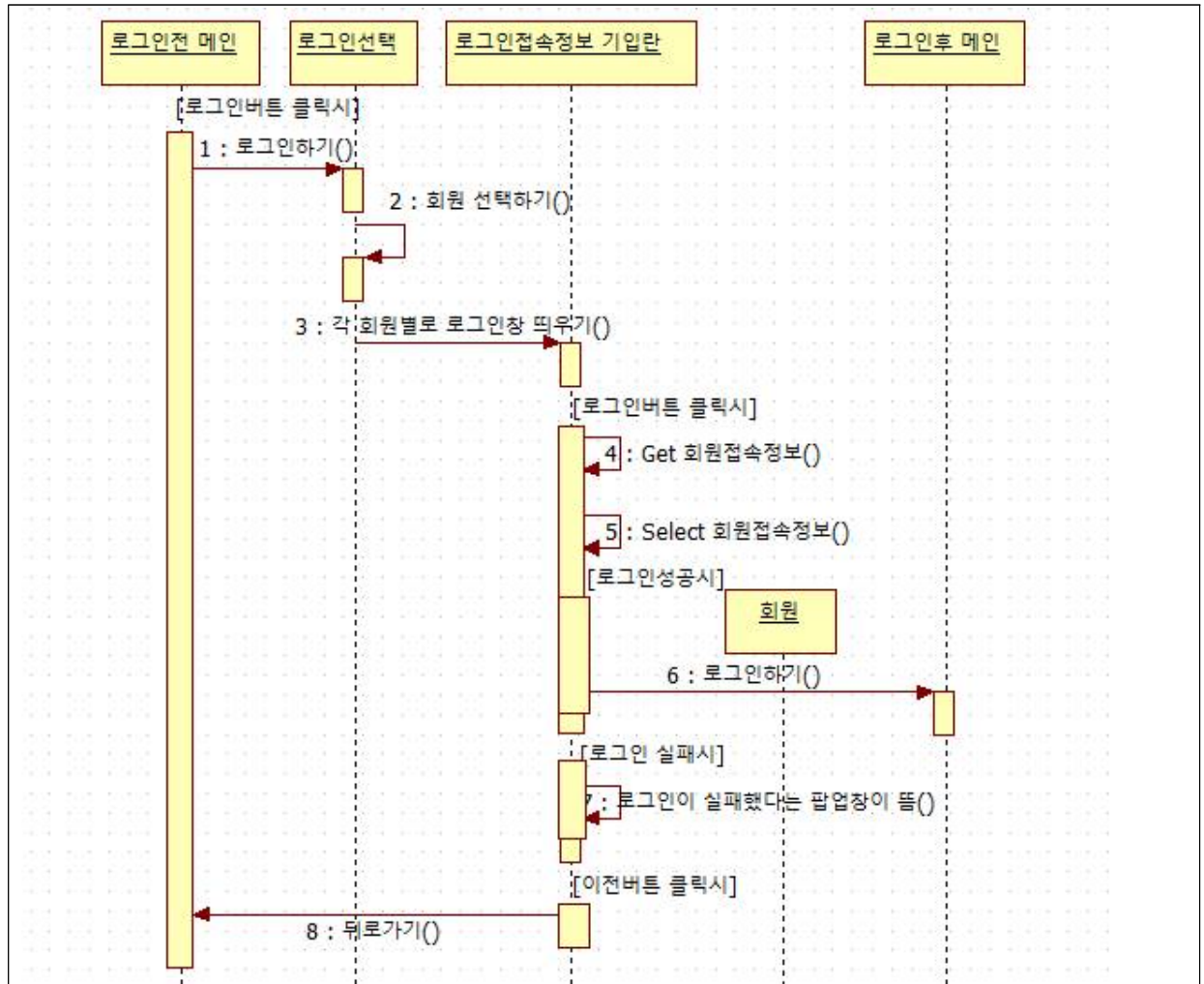
c. Class Diagram

--

a) 전체 메소드 기능

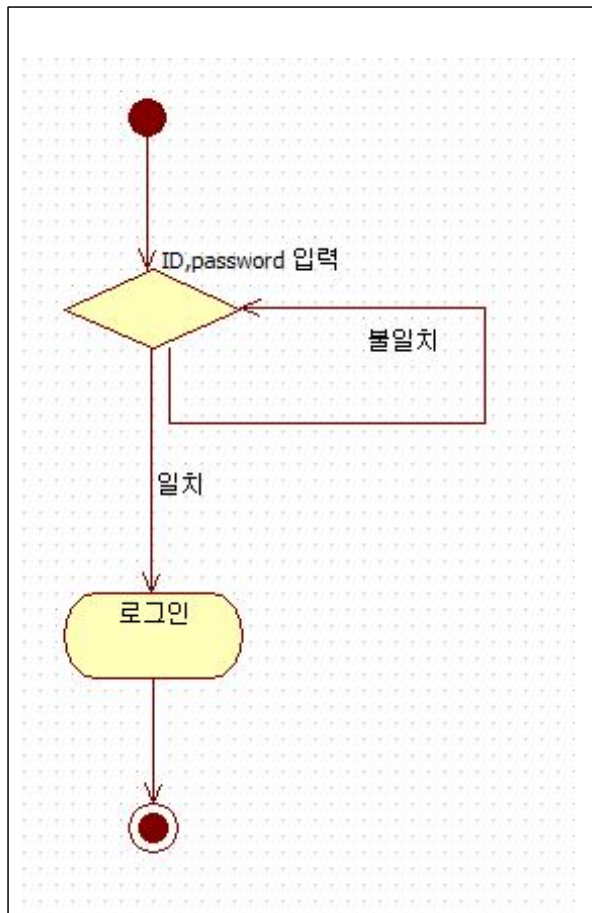
b) 일부 중요 변수

d. Sequence-Diagram



- a) 로그인전의 메인화면에서 로그인버튼을 클릭시에 실행된다.
- (a) 회원가입 화면이 뜨면서 건국대학교 학생 회원, 일반회원, 사업자회원의 3종류중 하나를 택함으로써 로그인의 방식을 선택 할 수 있다.
- (b) 회원을 선택 시, 각 회원별로 로그인화면이 뜨게 된다. 각각 별개의 화면이 뜨는 것이나, 이 역시 로그인의 기능적인 측면에서 볼 때, 공통적인 흐름을 표시하는 것이 좋다고 생각해서 해당 그림처럼 표현했다.
- (c) 각 로그인화면의 입력란에 입력을 한 후에 로그인버튼을 클릭 시, 우선 입력란의 정보를 받아오고, 회원이 맞는지 확인하는 절차가 있다.
- (d) 회원임이 확인되어 로그인이 성공되게 될 경우 마이페이지로 이동하게 되며, 맞지 않으면 ‘등록되어있는 회원이 아닙니다.’ 라는 팝업창이 뜨게 된다.

e. Activity-Diagram



a) 순서도(Flow Chart)처럼 표현해 본 Activity-Diagram이다.

(a) 로그인을 시도할 때, 각 회원에 맞는 필수정보를 입력하게 된다.

[건국대학교 학생 회원인 경우는 학번이랑 이름을, 일반회원인 경우는 핸드폰번호, 이름을, 사업자 회원인 경우는 사업자 번호, 이름을 입력해야 한다.]

(b) 회원이 맞을 경우 로그인이 성공한 것이므로, 로그인후 메인화면으로 이동하나, 회원이 틀릴 경우 팝업창이 뜨면서, 다시 재작성을 해야 하는 방식으로 반복된다.

(c) 이전 버튼을 클릭할 경우에는 로그인 전 메인화면으로 이동하게 된다.

3) 구역선택

a. 모듈기능설명

구역선택을 할 수 있는 모듈이다.

- 기본적으로 로그인후의 메인화면에서 구역별보기를 선택했다는 가정하에서 출발한다. [로그인 전에는 화면에서 클릭을 해도 실행되지 않는다.]

- 메인화면에서 구역별보기 버튼을 누르게 되면 구역선택 화면이 뜨게 된다.

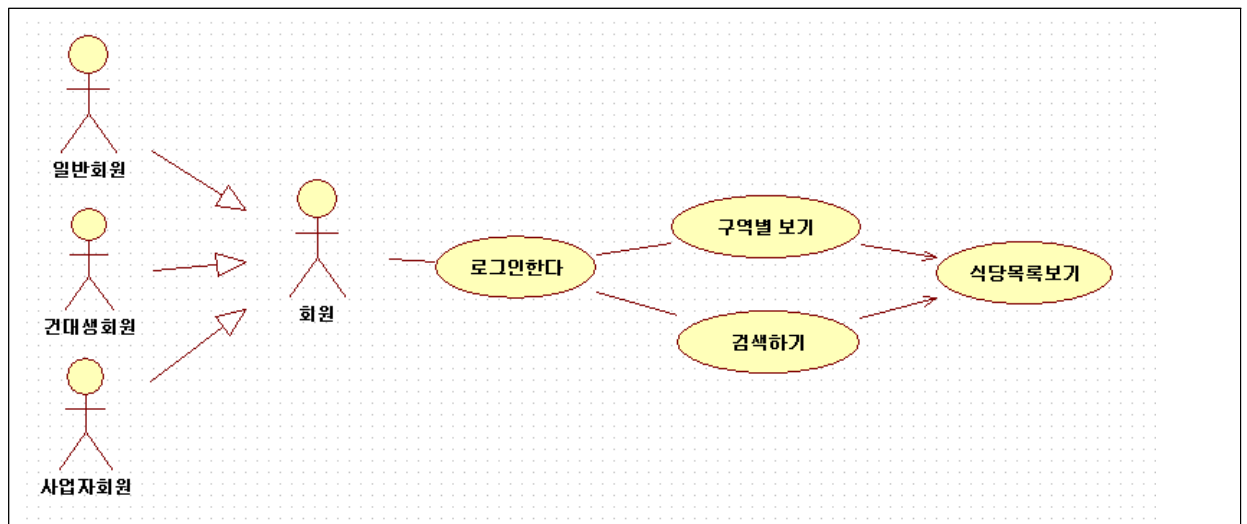
기본적인 구조는 구역선택 화면에서 전체 바탕이 지도로 표현되며, 건국대학교 내부[학생식당], 정문, 중문, 후문으로 총 4가지로 나뉘어져 있다.

- 각 구역별로 색깔을 다르게 지정해서 구분하기 쉽게 했다.

건대 구역에서는 건국대학교 학생식당 식단 정보에 관한 것이 포함되고, 정문에는 롯데백화점, 스타시티, 로데오거리 그쪽 주변 맛집들이 포함되며, 중문은 먹자골목과 그 주변 맛집들이 포함된다. 마지막으로 후문은 건국대학교 후문 주변과 세종대 주변 맛집이 포함되도록 분류시킬 예정이다.

- 그래서 원하는 구역을 선택할 시, 그 구역 맛집 식당들에 대한 정보가 리스트로 쭉 나타나서 선택할수 있도록 할 예정이다.

b. Use-Case Diagram



a) 우선 기본적으로 사용자가 회원일 경우, 로그인한 후에 구역별보기 통해 건국대학교 주변의 지도를 볼 수가 있고, 분할된 4개의 구역중 하나의 구역을 선택 함으로써, 해당 구역에 포함되어 있는 맛집 식당의 목록들을 볼 수가 있다.

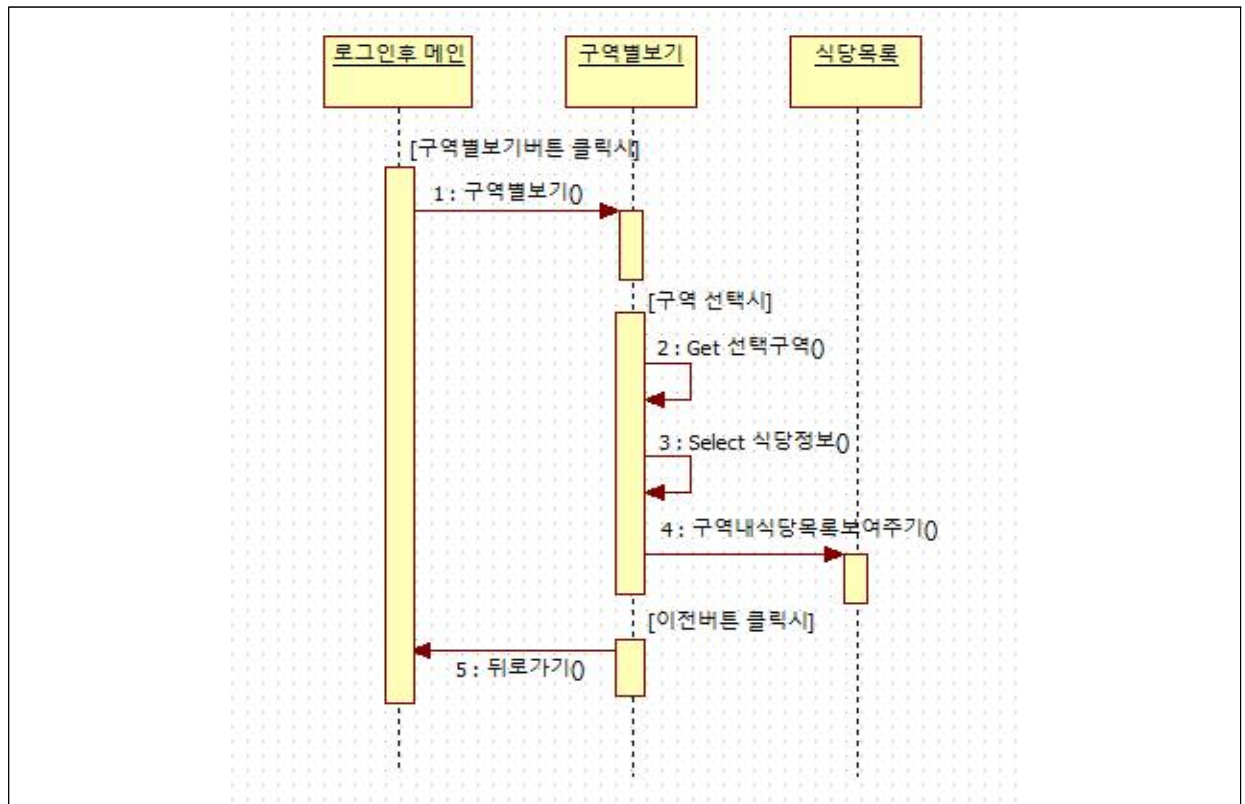
c. Class Diagram

--

a) 전체 메소드 기능

b) 일부 중요 변수

d. Sequence-Diagram



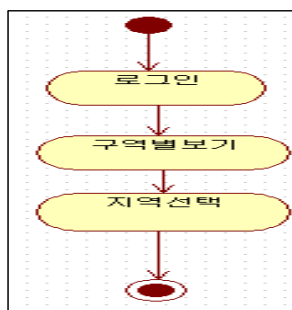
a) 로그인후의 메인화면에서 구역별보기버튼을 클릭시에 실행된다.

(a) 구역선택 화면이 뜨게 되며, 각 구역별로 색깔을 다르게 지정된 건국대학교 주변 지도가 전체 바탕으로 표현된다.

(b) 건국대학교 내부[학생식당], 정문, 중문, 후문의 4가지중 원하는 구역 한가지를 선택하게 되면, 어떤 구역을 선택했는지 읽어온다.

(c) 해당 구역의 맛집 식당들에 대한 정보를 찾는 작업으로 이어지며 이후, 식당목록에서 리스트로 쭉 나타게 된다.

e. Activity-Diagram



a) 로그인후의 메인화면에서 구역별보기를 선택한 후에, 구역을 선택하면 된다.

4) 식당검색

a. 모듈기능설명

식당검색을 할 수 있는 모듈이다.

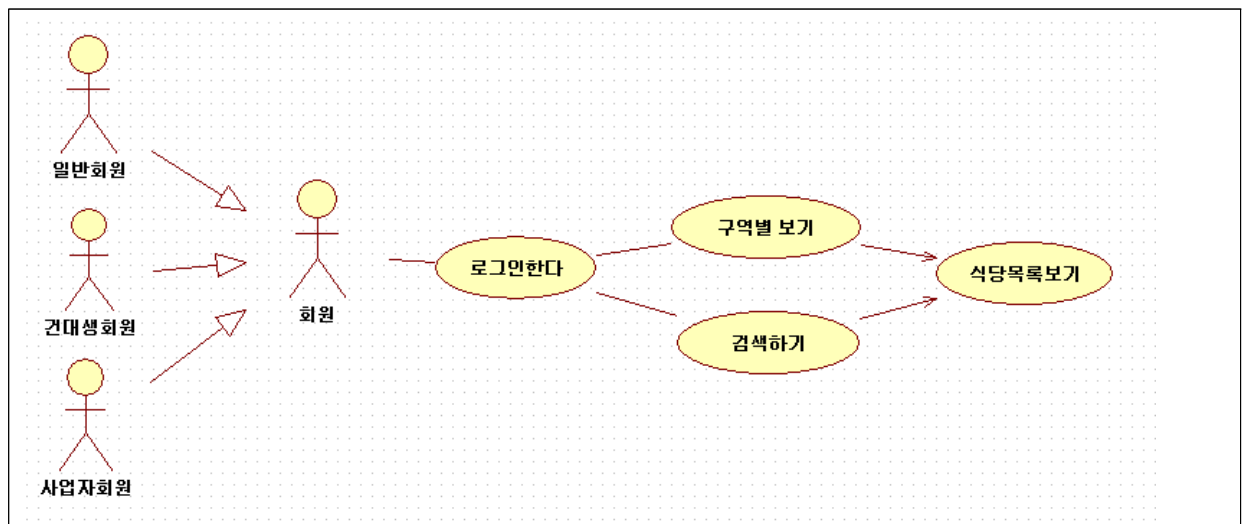
- 이 역시 기본적으로 로그인후의 메인화면에서 구역별보기를 선택했다는 가정하에서 출발한다. [로그인 전에는 화면에서 클릭을 해도 실행되지 않는다.]

- 메인화면에서 구역별보기 버튼을 누르게 되면 식당검색 화면이 뜨게 된다. 간단하게 식당이름을 입력할수 있는 검색란이 있다.

- 검색되기를 원하는 식당의 업종, 상호명 등을 입력한 후에 검색버튼을 클릭하면, DB에 저장되어 있는 식당 목록 중 검색요청 된 식당의 이름이 들어간 식당들을 찾아서, 해당 맛집 식당들에 대한 정보가 리스트로 쪽 나타나서 선택할 수 있도록 할 예정이다.

검색 결과가 없을 시에는 ‘검색 결과가 존재 하지 않습니다.’ 라는 팝업창이 뜬다.

b. Use-Case Diagram



a) 우선 기본적으로 사용자가 회원일 경우, 로그인한 후에 검색하기를 통해 검색화면으로 이동할수 있고, 식당이름을 입력하고 검색버튼을 누르면, 관련된 맛집 식당의 목록들을 볼 수가 있다.

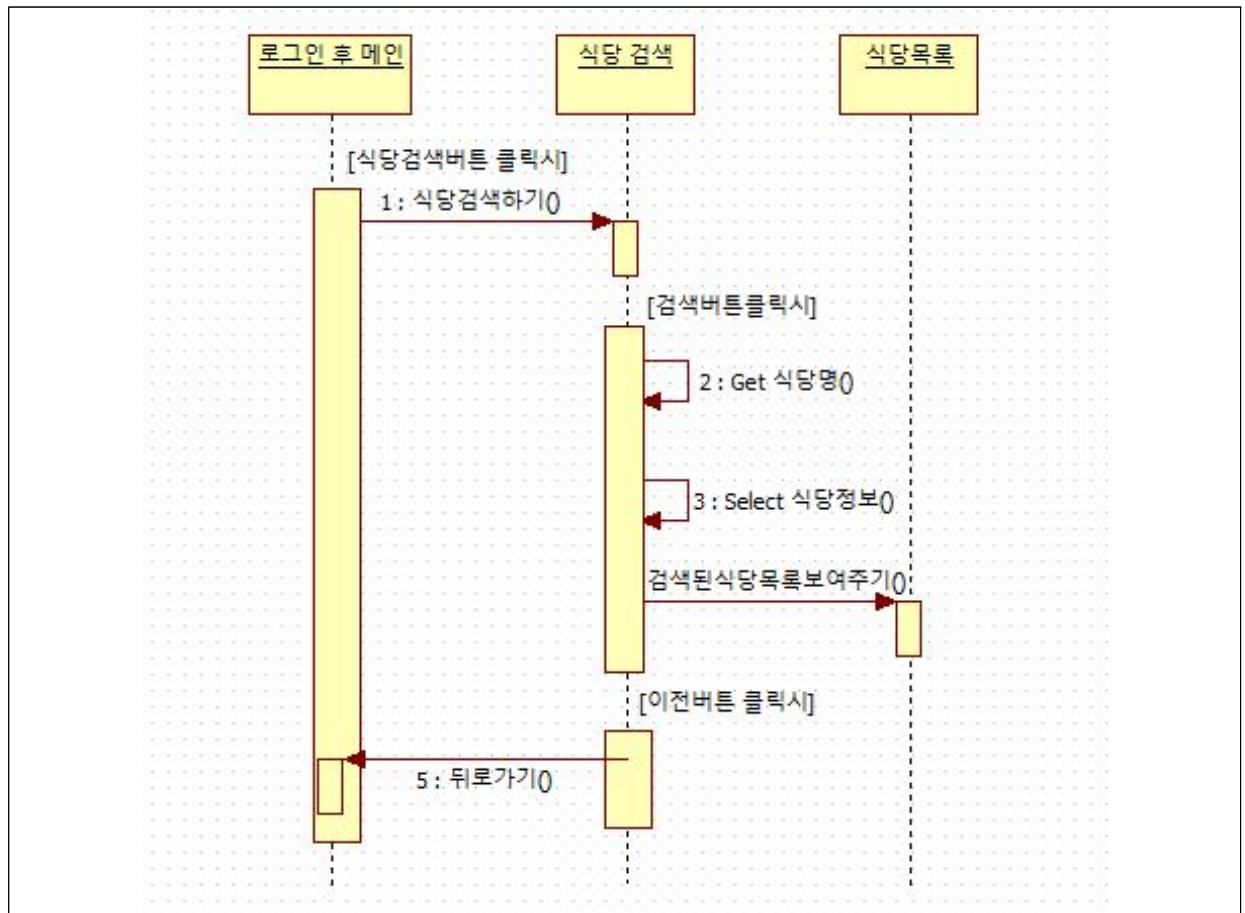
c. Class Diagram

--

a) 전체 메소드 기능

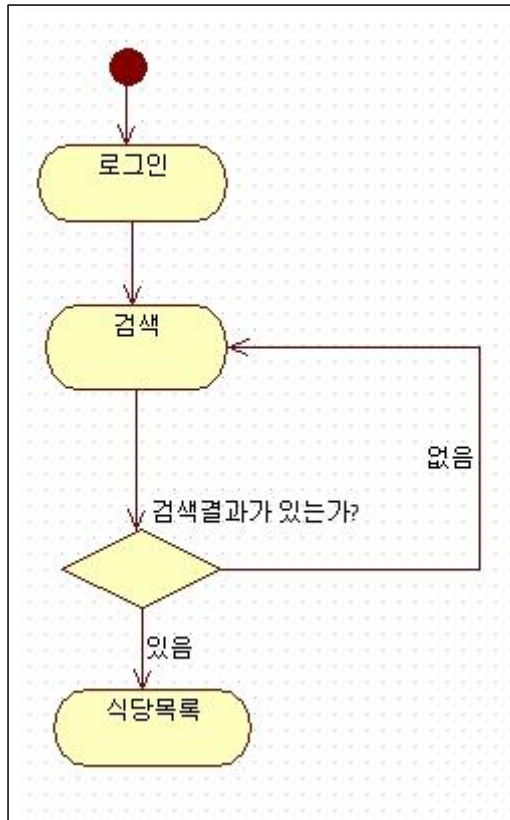
b) 일부 중요 변수

d. Sequence-Diagram



- a) 로그인후의 메인화면에서 식당검색버튼을 클릭시에 실행된다.
- (a) 식당검색 화면이 뜨게 되며, 검색버튼을 클릭하면, 입력한 식당명을 가져온다.
 - (b) 해당 식당명의 맛집 식당들에 대한 정보를 찾는 작업으로 이어지며 이후, 식당목록에서 리스트로 쪽 나타게 된다.

e. Activity-Diagram



a) 순서도(Flow Chart)처럼 표현해 본 Activity-Diagram이다.

- (a) 로그인 후, 검색버튼을 통해 검색화면으로 이동하고 검색을 한다.]
- (b) 검색결과가 있을 경우 바로 식당목록으로 이동하게 되며, 만약 전혀 없을 경우, ‘검색 결과가 존재 하지 않습니다.’ 는 팝업창이 뜨게 된다.
- (c) 이전 버튼을 클릭할 경우에는 로그인 후 메인화면으로 이동하게 된다.

5) 식당선택

a. 모듈기능설명

식당선택을 할 수 있는 모듈이다.

- 기본적으로 구역별보기나, 식당검색을 진행했다는 가정 하에서 출발한다.

- 어떤 유입방식이던간에 조건을 맞춘 맛집들이 나타나게 될 텐데, 목록에 식당이 쪽 나열되는 방식이고, 스크롤을 통해 나열된 식당정보들을 볼 수가 있다.

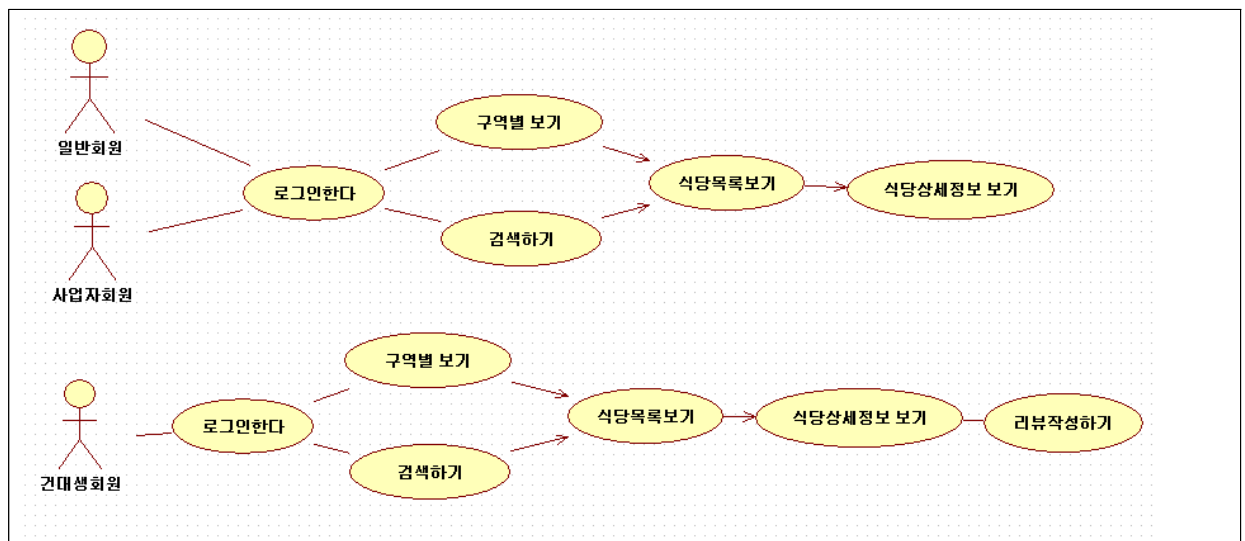
- 식당 목록에는 식당이름, 사진, 별점 등의 간단한 정보만 나타나며, 이중 식당을 하나 선택했을 시에 상세화면이 나타나게 된다.

- 앞서 적은 내용 이외에도 사진 또는 지도상의 위치, 주소, 전화번호, 메뉴목록과 가격도 같이 표시된다.

- 리뷰보기를 클릭 시, 작성되어 있던 리뷰를 확인할수 있으며, 여기서 리뷰 작성은 건국대학교 학생 회원인 경우만 가능하고, 리뷰 삭제는 불가능하다.

- 리뷰 작성시 별점을 함께 입력하고, 식당의 별점은 작성된 리뷰들의 별점들의 평균으로 구한다.

b. Use-Case Diagram



a) 일반 사용자 입장에서는 식당목록에서 식당을 선택함으로써 식당의 상세정보를 볼수가 있고, 작성되어있는 리뷰목록을 볼수가 있다.

b) 여기서 건국대학교 학생회원일 경우, 별점을 매기면서 리뷰를 작성할 수 있다.

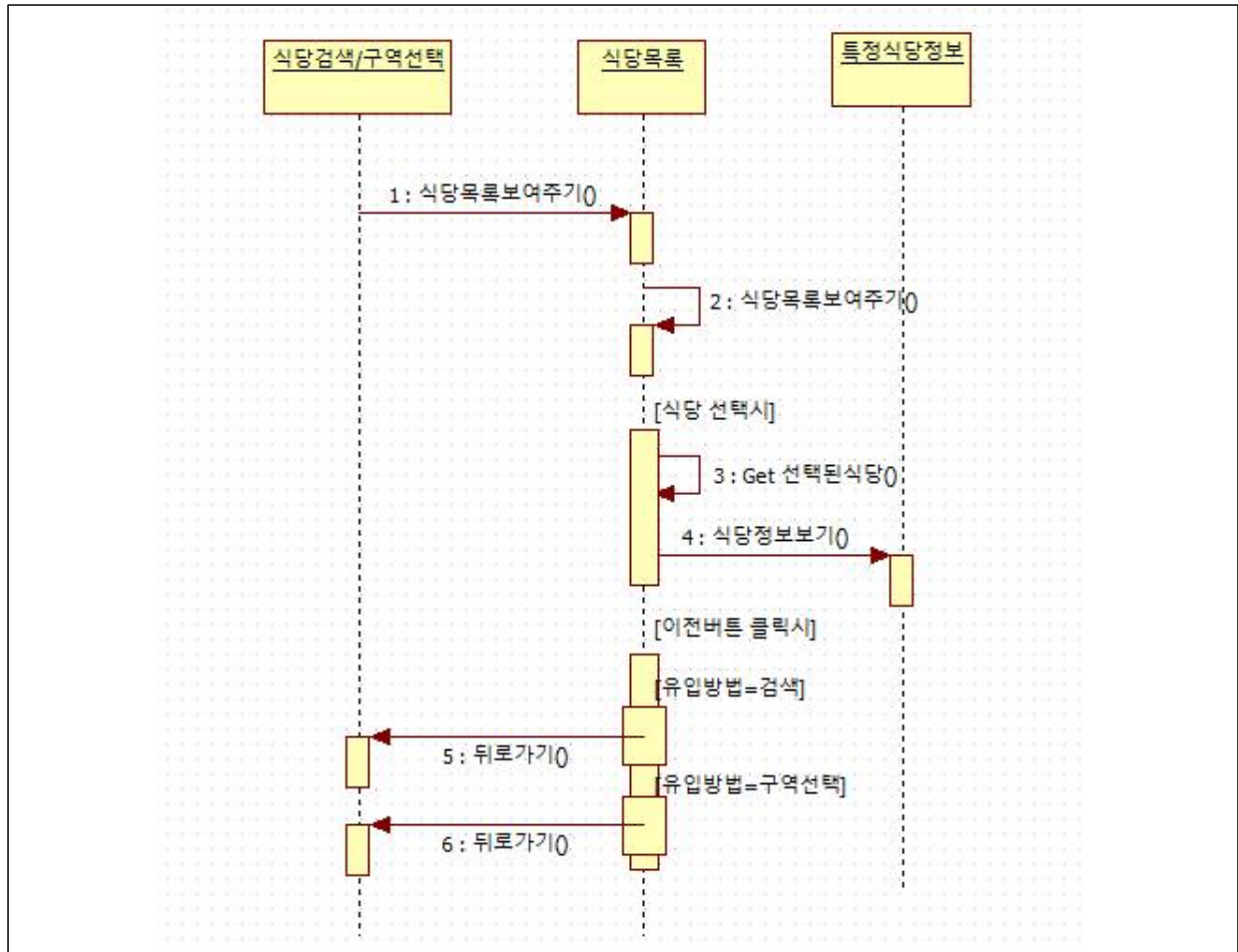
c. Class Diagram

--

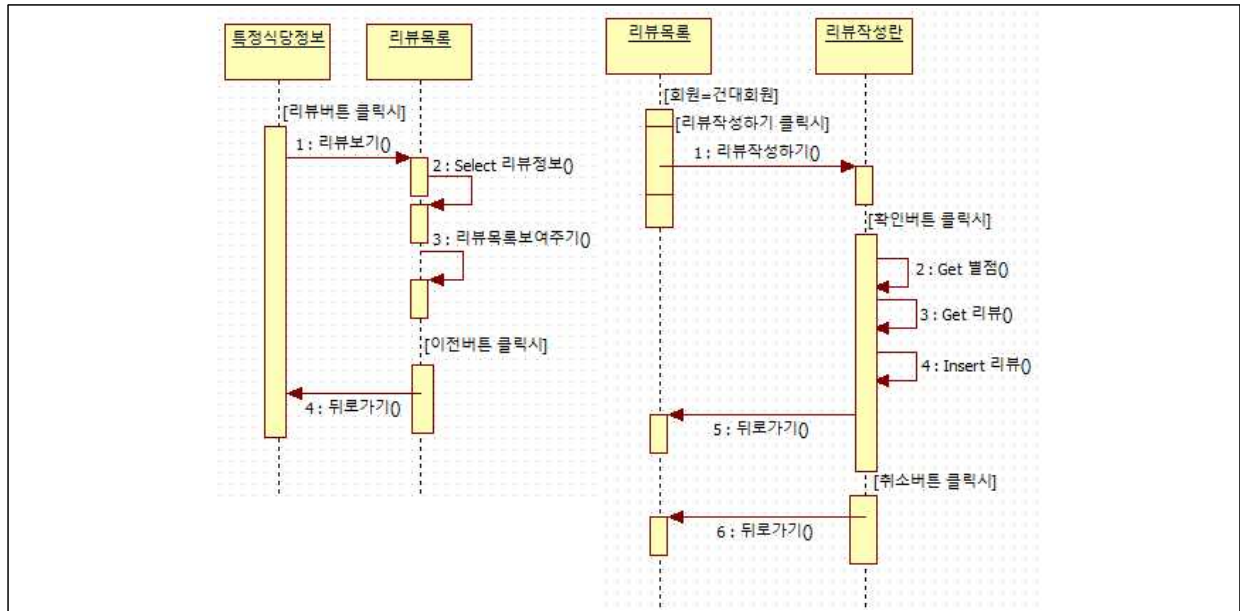
a) 전체 메소드 기능

b) 일부 중요 변수

d. Sequence-Diagram



- a) 식당검색 또는 구역 선택을 통해 진행 됐다는 가정 하에서 출발한다.
 - (a) 검색된 결과인 식당목록을 화면에 쭉 보여준다.
 - (b) 그중의 식당을 하나 선택 시, 선택된 식당이 몇 번째 값인지를 얻어와서 선택된 식당이 어떤 식당인지를 파악하고, 식당상세정보 화면으로 옮겨지게 된다.
 - (c) 식당상세정보에서는 식당이름, 사진 또는 지도상의 위치, 별점(총점), 주소, 전화번호, 메뉴목록과 가격이 표시된다.
 - (d) 여기서 이전버튼을 클릭 시 ,유입방식이 무엇이냐에 따라 검색을 통해 들어왔으면 검색화면으로 돌아가고, 구역선택을 통해 들어왔으면 구역선택화면으로 돌아가게 된다.



b) 특정식당의 식당정보화면에서 리뷰버튼을 클릭시, 해당 식당의 리뷰목록을 볼 수가 있다.

(a) 특정식당의 식당번호를 통해 DB에 검색을 요청해서, 리뷰의 정보를 가져오고, 화면에 꼭 보여준다.

(b) 이전버튼을 클릭 시, 식당상세정보 화면으로 돌아오게 된다.

c) 리뷰목록 화면에서 건대회원일 경우 리뷰를 작성할 수가 있다.

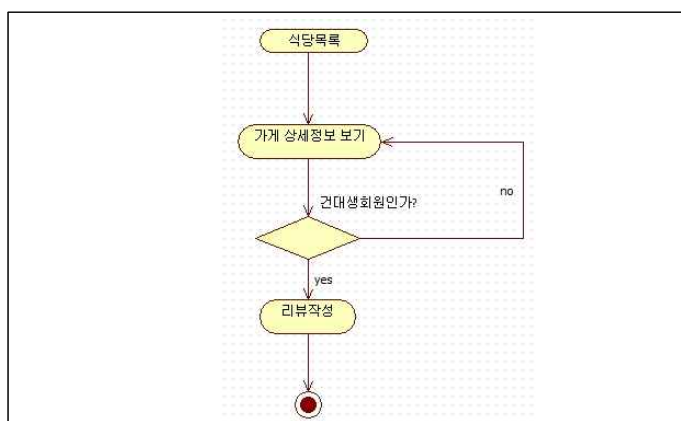
(a) 리뷰작성버튼을 클릭 시, 리뷰작성 란이 뜨게 된다.

(b) 확인버튼을 클릭시, 별점과 리뷰를 얻어오게 되고 DB에 등록하게 된다.

(c) 이후 리뷰목록 화면으로 돌아오게 된다.

(d) 작성도중 취소버튼을 클릭시, 리뷰목록 화면으로 돌아오게 된다.

e. Activity-Diagram



a) 식당 상세정보보기에서 리뷰보기로 넘어왔을 때, 건대생회원일 경우 리뷰작성도 가능하다.

6) 마이페이지

a. 모듈기능설명

마이페이지를 관리 할 수 있는 모듈이다.

- 기본적으로 로그인후의 메인화면에서 구역별보기를 선택했다는 가정 하에서 출발한다. [로그인 전에는 메인화면에 존재하지도 않았다.]

- 메인화면에서 마이페이지 버튼을 누르게 되면 마이페이지 화면이 뜨게 된다.

회원의 종류가 3가지이기 때문에 마이페이지 역시 총 3가지로 나뉘며 각각의 화면과 이용 가능한 기능도 조금씩 다르다.

- 건국대학교 학생 회원인 경우의 마이페이지는 학번, 학과, 이름, 핸드폰번호, 성별을 보는 것과 수정이 가능하고, 일반회원인 경우도 건국대학교 학생 회원인 경우와 마찬가지로 회원 가입 시 입력한 사항들을 보는 것과 수정이 가능하다. 또 회원 탈퇴도 가능하다.

- 마지막으로 사업자 회원인 경우는 앞에 두 경우와는 좀 다른데, 사업자 회원이 마이페이지를 클릭했을 시에는 총 3가지 버튼이 나타나게 된다.

첫 번째 버튼은 내 식당 등록 요청하기이고 두 번째 버튼은 내 식당 수정 요청하기이며, 마지막의 세 번째 버튼은 개인 정보 수정 이다.

- 세 번째 버튼은 앞에 두 경우와 마찬가지로 입력한 사항들을 보는 것과 수정이 가능하고 회원 탈퇴도 가능하다.

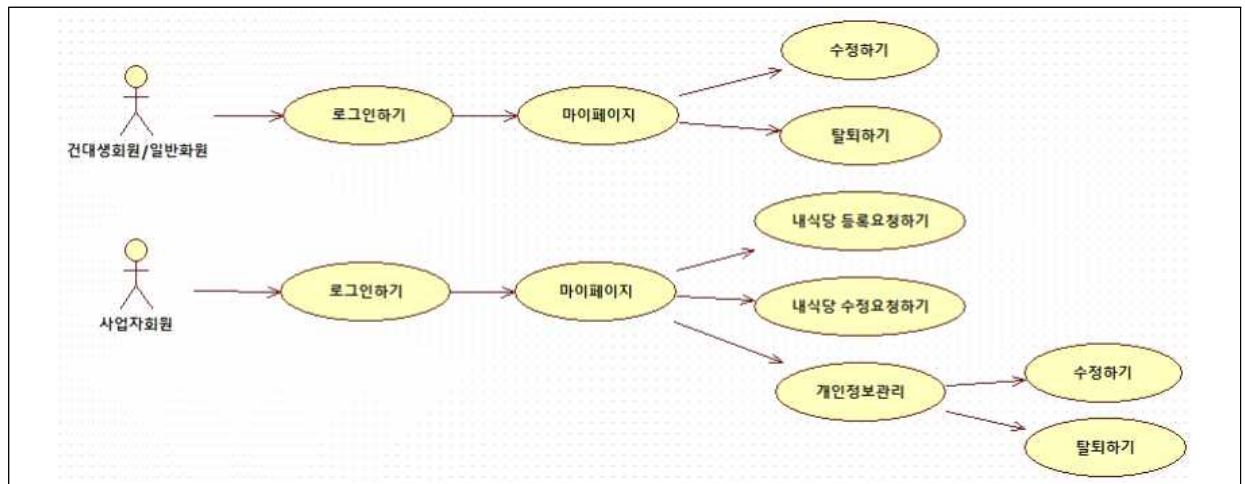
- 앞의 두 경우와 다른 첫 번째 버튼 같은 경우는 버튼을 눌렀을 시 맛집에 대한 화면이 나오게 되는데 그 안에는 내용이 아무 것도 없는 빈 양식이 나오게 된다.

빈 양식이 나왔을 때 사업자가 그 안에 내용들을 다 입력해서 관리자한테 보내게 되면 관리자는 그 안에 내용들을 확인하게 된다. 여기서 내용이 적당하다 판단하면 어플리케이션에 등록을 하게 되고, 적당하지 않다고 판단을 하면 어플리케이션에 등록을 하지 않는다.

- 두 번째 내 가게 수정 요청하기를 누르면 사업자 회원의 가게에 대한 화면이 나오게 되는데 거기서 수정할 사항들을 수정해서 관리자한테 보내게 된다.

관리자는 그 수정 내용을 확인하고 수정내용이 맞는다고 판단이 되면 그 수정사항을 어플리케이션에 반영하게 된다.

b. Use-Case Diagram



- a) 사용자가 건대생회원/일반회원이나 사업자회원이나에 따라 2가지로 분류했다.
- (a) 건대생회원 또는 일반회원일 경우, 로그인을 통해 로그인후화면에서 마이페이지를 선택 시, 자신이 회원가입 시에 입력했던 정보들을 볼 수 있다.
 - (b) 이를 수정하기 버튼을 통해 개인정보의 수정을 할 수 있다.
 - (c) 이 외에도 탈퇴하기 버튼을 통해 회원탈퇴가 가능하다.
- b) 만약 사용자가 사업자회원일 경우, 내가게 등록요청하기, 내가게 수정요청하기, 개인정보관리 3가지의 버튼이 나오게 돼서 선택을 할 수 있다.
- (a) 내 가게 등록 요청하기 버튼을 눌렀을 시 ,내용이 아무 것도 없는 빈 양식이 나오게 된다. 사업자가 그 안에 내용들을 다 입력한 후에 관리자에게 식당 등록요청을 보내게 된다.
 - (b) 내 가게 수정 요청하기 버튼을 누르면 사업자 회원이 기존에 등록했던 가게에 대한 정보가 채워진 화면이 나오게 되며, 이중 일부를 수정해서 관리자에게 수정요청을 보내게 된다.
 - (c) 개인정보관리 버튼을 누르면 건대생회원/일반회원과 같이 로그인시의 개인정보가 나타나게 되며, 이러한 정보의 수정을 할 수 있고, 탈퇴도 가능하다.

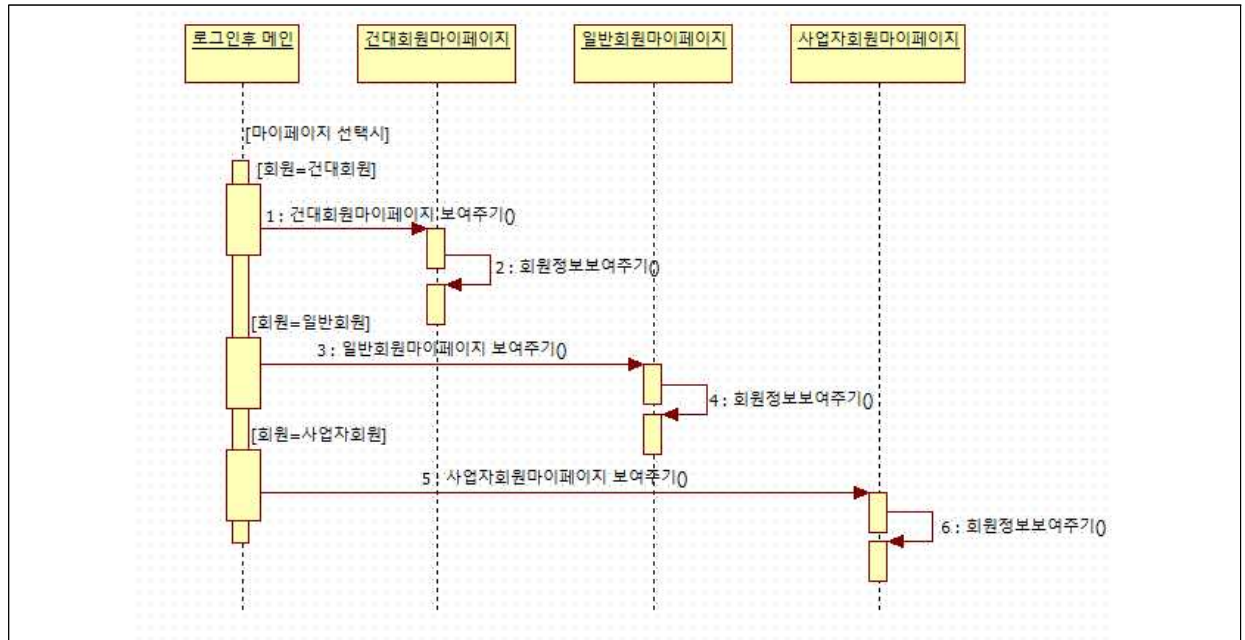
c. Class Diagram

--

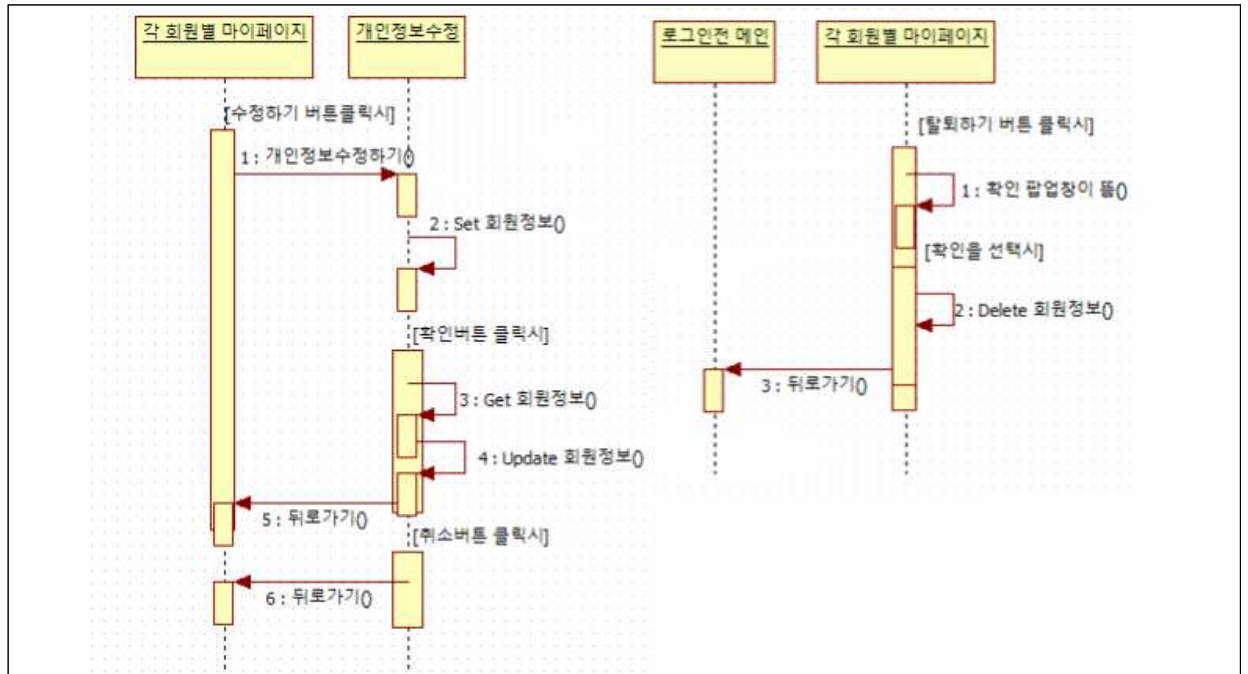
a) 전체 메소드 기능

b) 일부 중요 변수

d. Sequence-Diagram



- a) 로그인전의 메인화면에서 마이페이지버튼을 선택 시에 실행된다.
- (a) 해당 회원이 어떤 회원이냐에 따라, 각 회원에 맞는 마이페이지화면을 띄워주며, 이에 따라 회원의 정보를 보여준다.
 - (b) 각 페이지에서는 개인정보의 수정과 회원탈퇴버튼이 있어서, 해당기능을 수행할 수가 있다.
 - (c) 사업자회원의 경우 본래 3가지 버튼만 뜨며, 개인정보관리 버튼을 클릭 시에만 이와 같은 화면이 뜨나, 일반화시킨다는 가정 하에서 일단은 이렇게 표현했다.



b) 마이페이지화면에서 수정하기 버튼을 선택 시에 실행된다.

[사업자회원의 경우 개인정보관리버튼을 클릭한 이후가 해당된다.]

(a) 우선 회원의 정보가 표시된다.

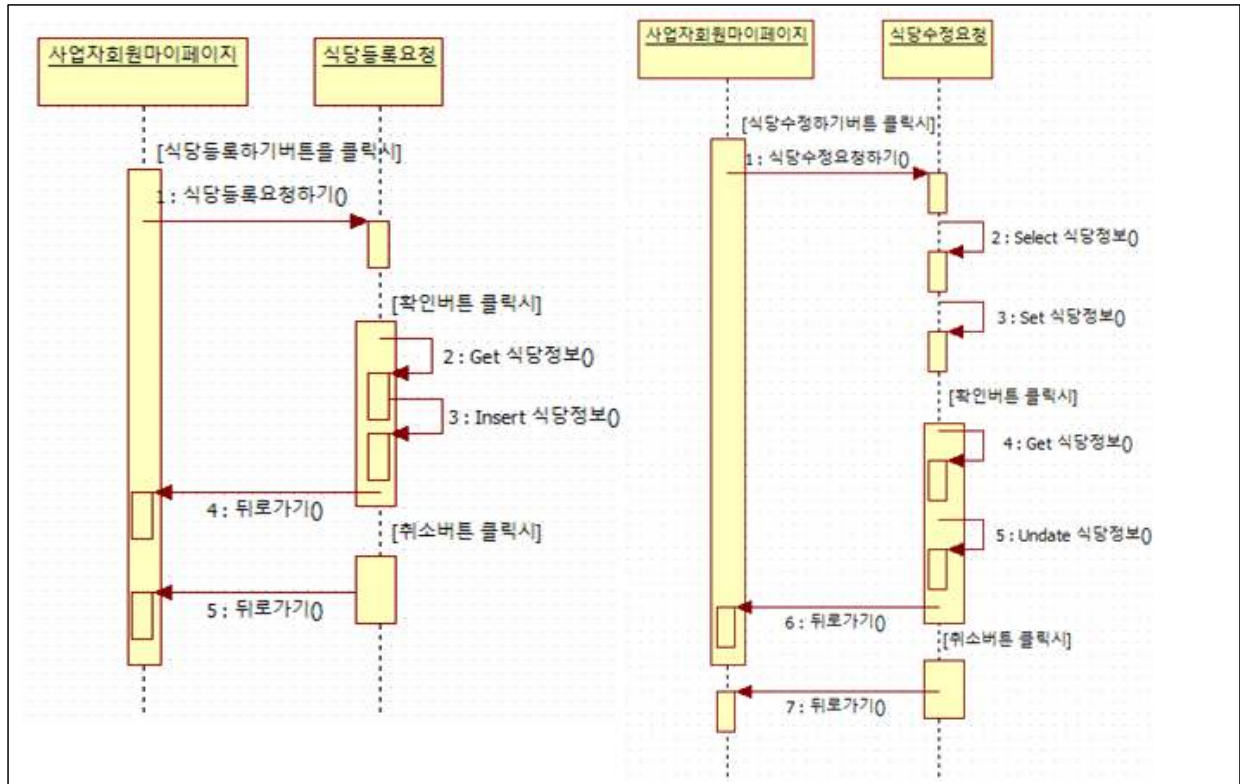
(b) 이후 확인버튼을 클릭 시, 회원정보를 읽어오고 DB를 수정 시킨다.

c) 마이페이지화면에서 탈퇴하기 버튼을 선택 시에 실행된다.

[사업자회원의 경우 개인정보관리버튼을 클릭한 이후가 해당된다.]

(a) ‘정말로 탈퇴하실 겁니까?’ 라는 팝업창이 뜬다.

(b) 이후 확인버튼을 클릭 시, DB에서 회원정보를 삭제 시킨다.



d) 사업자회원의 마이페이지화면에서 식당등록요청하기 버튼을 클릭 시에 실행된다.

(a) 식당과 관련된 기입란이 표시된다.

(b) 내용을 입력하고 확인버튼을 클릭하면, 식당정보를 읽어온다.

(c) 이후 DB에 등록시킨다. [이후에 이걸 관리자가 살펴보게 된다.]

e) 사업자회원의 마이페이지화면에서 식당수정요청하기 버튼을 클릭 시에 실행된다.

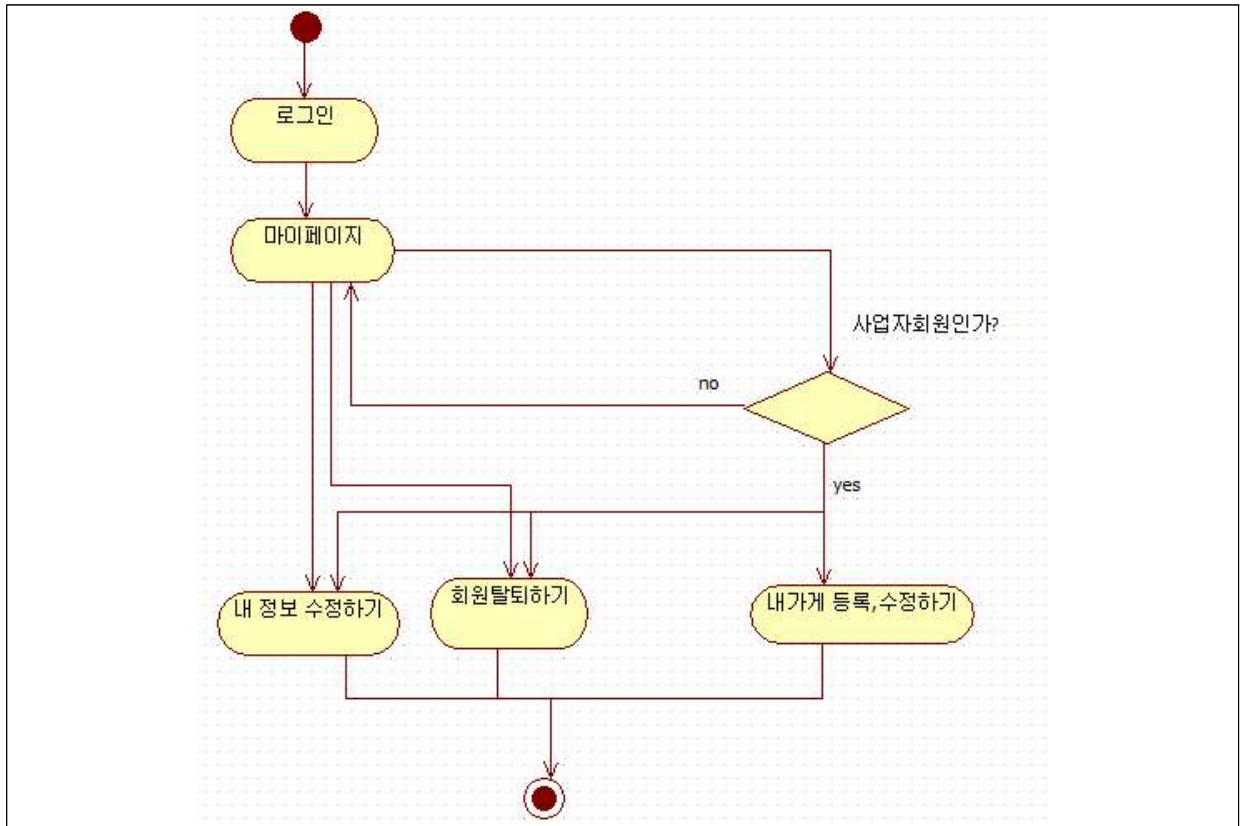
(a) DB에서 사업자가 등록했던 식당의 정보를 찾는다.

(b) 식당과 관련된 기입란에 찾은 정보 값을 표시한다.

(b) 내용을 수정하고 확인버튼을 클릭하면, 식당정보를 읽어온다.

(c) 이후 DB를 수정시킨다. [이후에 이걸 관리자가 살펴보게 된다.]

e. Activity-Diagram



a) 순서도(Flow Chart)처럼 표현해 본 Activity-Diagram이다.

(a) 사업자회원이 아닐 경우, 내 정보를 확인하면서, 내 정보 수정하기, 회원탈퇴하기만 가능하다.

(b) 사용자가 로그인을 해서 마이페이지에 들어온 이후에 사업자회원일 경우 내가게 등록하기, 수정하기의 추가적인 기능을 이용할 수가 있다.

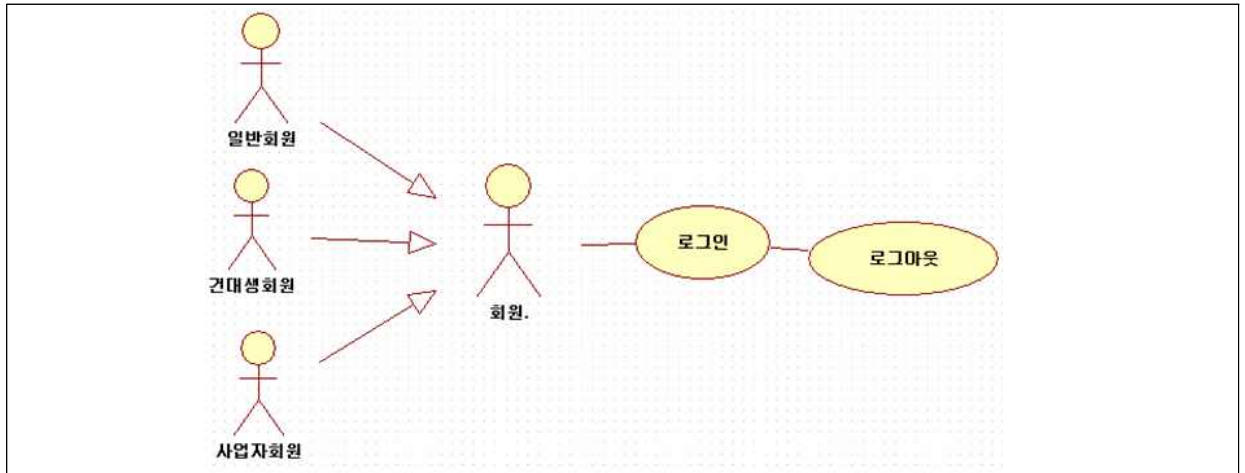
7) 로그아웃/프로그램 종료

a. 모듈기능설명

말 그대로, 로그아웃을 하거나 프로그램을 종료하는 모듈이다.

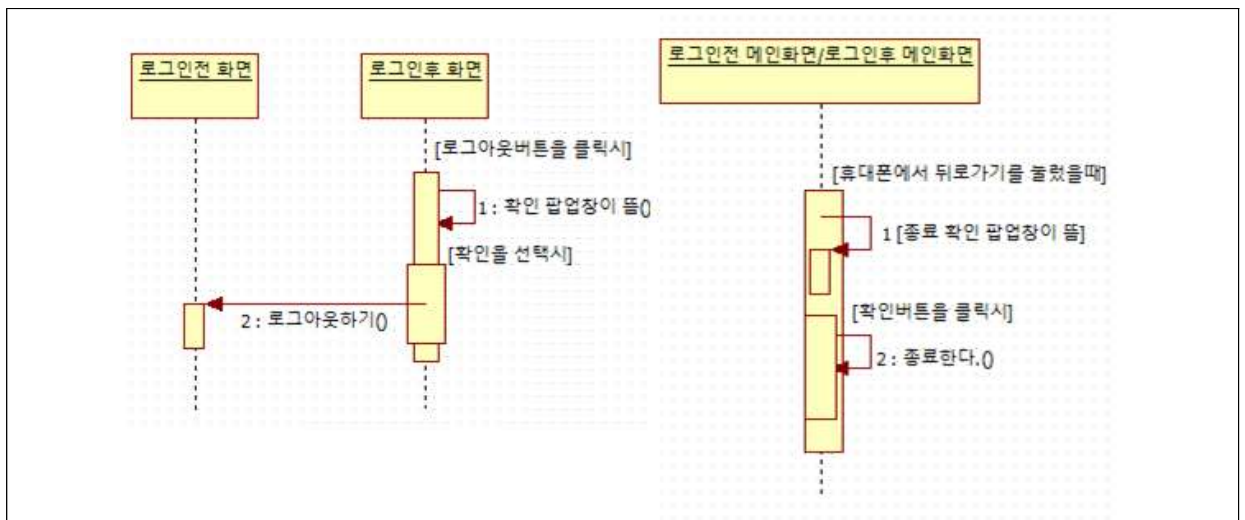
- 로그인 후의 메인화면에서 아랫부분에 위치한 로그아웃 버튼을 누르게 되면 로그아웃이 돼서 로그인 전의 메인화면으로 이동하게 된다.
- 로그인 전의 메인화면이나 로그인 후의 메인화면에서 뒤로가기를 선택하게 되면 프로그램을 종료할건지에 대해 물어보는 팝업창이 뜨게 되고, 확인을 선택하게 되면 프로그램이 종료하게 된다.

b. Use-Case Diagram



- a) 사용자가 어떤 회원이던간에, 로그인이 되어있는 상황이다.
- b) 로그인후 메인화면에서 로그아웃버튼을 클릭하게 되면 이루어진다.

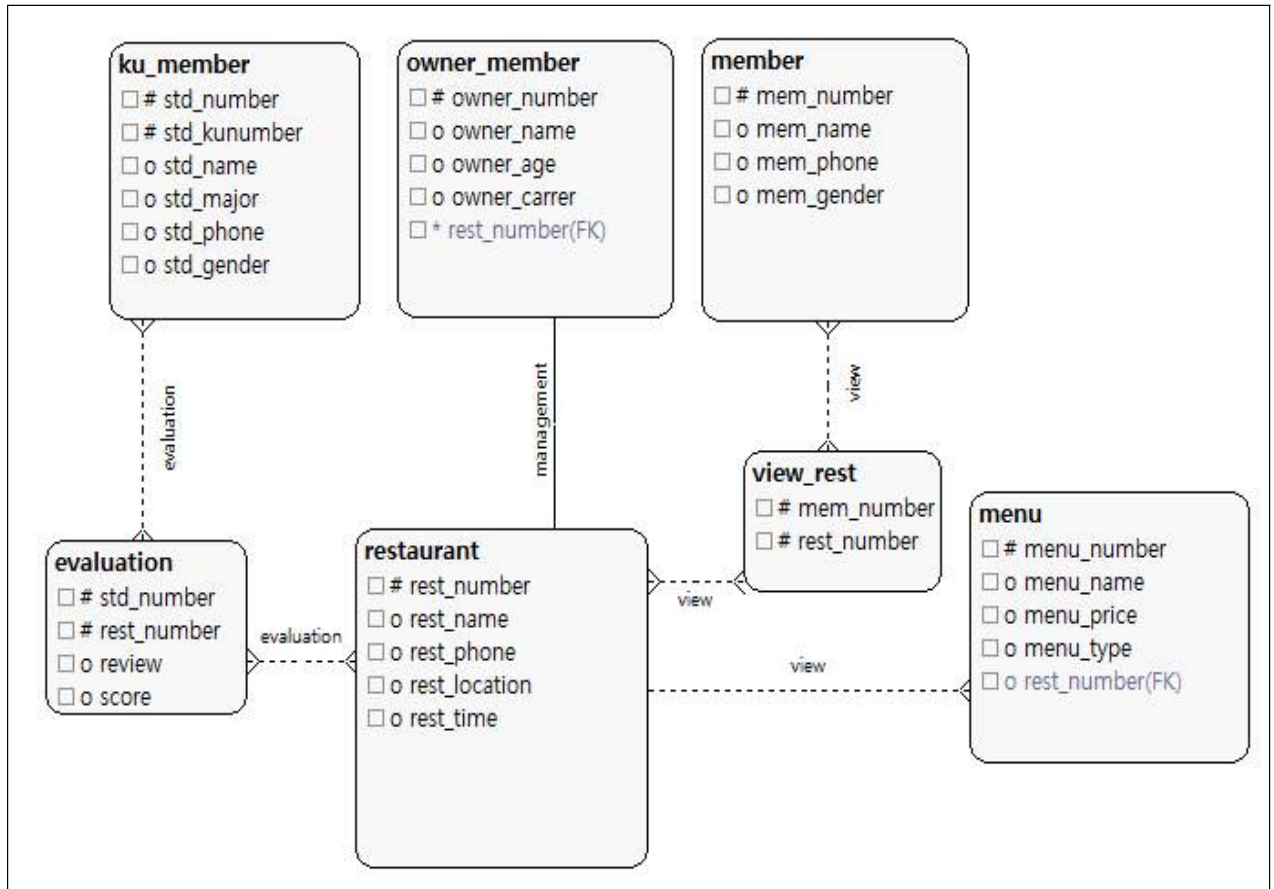
d. Sequence-Diagram



- a) 로그인후의 메인화면에서 로그아웃버튼을 선택 시에 실행된다.
 - (a) 정말로 로그아웃할것인냐고 확인하는 팝업창이 뜬 후에, 확인을 선택시 로그아웃이 이루어지면서, 로그인전 메인화면으로 이동하게 된다.
[로그아웃에서 팝업창이 뜨게 할지 말지는 고민중이다.]
- b) 로그인전의 메인화면이던, 로그인후의 메인화면이던간에 뒤로가기를 눌렀을 때 실행된다.
 - (a) ‘프로그램을 종료하시겠습니까?’ 라는 팝업창이 뜨며, ‘예’ 를 선택시에 프로그램이 종료되게 된다.

3. 데이터베이스 상세 설명

1) ER-Diagram



사용하는 데이터베이스의 ER-Diagram이다.

a) 맨 처음에 소개하였듯, 총 7가지의 테이블이 존재한다.

(a) 3가지로 분류한 회원 테이블인 Ku_Member, Member, Owner_Member,

(b) 식당정보와 메뉴로 테이블인 Restaurant, Menu

(c) 부가적으로 파생된, Evaluation, View_Rest의 테이블이 사용된다.

b) 각 테이블에는 속성값들이 존재하게 된다.

(a) 기본키(Primary Key)의 경우, 처음에 등록되게 되는 순서를 기준으로 삼았다.

(b) 외래키(Foreign Key)의 경우, 주로 Restaurant의 기본키인 rest_number를 참고있는 경우가 많으며 Owner_Member와 Menu에서 이를 외래키로 참고 있다.

2) 테이블 상세 정의

a. Ku_Member 테이블

a) 테이블 소개 : 건대생회원(학번, 이름, 학과, 핸드폰번호, 성별)

b) ER_Diagram의 릴레이션 사상의 결과

Ku_Member(Std_number, Std_kunumber, Std_name, Std_major, Std_phone number, Std_gender)

c) 작성된 SQL 코드

```
CREATE TABLE Ku_Member
(
  Std_number number not null,
  Std_kunumber number not null,
  Std_name varchar(20),
  Std_major varchar(20),
  Std_phone number,
  Std_gender varchar(10)
  PRIMARY KEY(Std_number)
);
```

b. Member 테이블

a) 테이블 소개 : 일반인회원(이름, 회원번호, 핸드폰번호, 성별)

b) ER_Diagram의 릴레이션 사상의 결과

Member(Mem_number, Mem_name, Mem_phone, Mem_gender)

c) 작성된 SQL 코드

```
CREATE TABLE Member
(
  Mem_number number not null,
  Mem_name varchar(20),
  Mem_phone number,
  Mem_gender varchar(10)
```

PRIMARY KEY(Mem_number)

);

c. Restaurant 테이블

a) 테이블 소개 : 식당(이름, 전화번호, 위치, 영업시간)

b) ER_Diagram의 릴레이션 사상의 결과

Restauant(Rest_number, Rest_name, Rest_phone, Rest_location, Rest_time)

c) 작성된 SQL 코드

CREATE TABLE Restaurant

(

Rest_number number not null,

Rest_name varchar(20),

Rest_phone number,

Rest_location varchar(50),

Rest_time date,

PRIMARY KEY(Rest_number)

);

d. Owner_Member 테이블

a) 테이블 소개 : 음식점주인회원(사업자번호, 이름, 나이, 경력)

b) ER_Diagram의 릴레이션 사상의 결과

Owner_Member(Owner_number, Owner_name, Owner_age, Owner_career, [Rest_number](#))

c) 작성된 SQL 코드

CREATE TABLE Owner_Member

(

Owner_number number not null,

Owner_name varchar(20),

Owner_age number,

Owner_career number,

```
Rest_number number,
PRIMARY KEY(Onwer_number),
FOREIGN KEY(Rest_number) REFERENCES Restaurant(Rest_number)
);
```

e. Menu 테이블

a) 테이블 소개 : 음식점메뉴(가격, 이름, 종류[양식, 한식, 일식, 분식, 중식])

b) ER_Diagram의 릴레이션 사상의 결과

Menu(Menu_number, Menu_name, Menu_price, Menu_type, Rest_number)

c) 작성된 SQL 코드

```
CREATE TABLE Menu
(
Menu_number number not null;
Menu_name varchar(20),
Menu_price number,
Menu_type varchar(20),
Rest_number number,
PRIMARY KEY(Menu_number),
FOREIGN KEY(Rest_number) REFERENCES Restaurant(Rest_number)
);
```

f. Evaluation 테이블

a) 테이블 소개 : 평가(리뷰, 별점)

b) ER_Diagram의 릴레이션 사상의 결과

Evaluation(Std_number, Rest_number, review, score number)

c) 작성된 SQL 코드

```
CREATE TABLE Evaluation
(
Std_number number not null,
```



```

Rest_number number not null,
review varchar(200),
score number,
PRIMARY KEY(Std_number, Rest_number),
FOREIGN KEY(Std_number) REFERENCES Ku_Member(Std_number),
FOREIGN KEY(Rest_number) REFERENCES Restaurant(Rest_number)
);

```

g. View_Rest 테이블

a) 테이블 소개 : 식당보기0. 순전히 N:M관계를 만족시키기 위해서 파생된 테이블이다.

b) ER_Diagram의 릴레이션 사상의 결과

View_Rest([Mem_number](#), [Rest_number](#))

c) 작성된 SQL 코드

```

CREATE TABLE View_Rest
(
Mem_number number not null,
Rest_number number not null,
PRIMARY KEY(Mem_number, Rest_number),
FOREIGN KEY(Mem_number) REFERENCES Member(Mem_number),
FOREIGN KEY(Rest_number) REFERENCES Restaurant(Rest_number)
);

```