Yeungnam University

예약의 민족

- SDS(Software Design Specification) -

21413500	손준우	sjw3957@gmail.com
21413503	이우빈	lub1158@naver.com
21413506	최수연	qjtmzj99@gmail.com
21413507	최인자	ingja@ynu.ac.kr

Yeungnam University

[Revision history]

Revision date	Version #	Description	Author
17/11/10	1.1	Describe the main points.	all member
17/12/02	1.2	Revise changing point from realistic possibility	all member
17/12/08	1.3	Final checked of sds	all member

Yeungnam University

= Contents =

1. Introduction 1

2. Use case analysis 2

3. Class diagram 23

4. Sequence diagram 48

5. State machine diagram 62

6. Implementation requirements 63

7. Glossary 64

8. References 65

1. Introduction

- Summarize the contents of this document. - Describe the important points of your design. - 12pt, 160%.

현대인들은 바쁜 하루를 살면서 불필요한 시간의 낭비를 최소화 하고자 한다. 가장 흔하게 생각할 수 있는 예는 점심시간에 밥을 먹을 때이다. 점심시간이 되면 갈 음식점을 정해야하고 식당에 가서도 자리가 만석이면 어쩔 수 없이 기다리거나 다른 음식점을 찾아 헤매야 한다. 또한 자리를 잡고도 메뉴를 고르고 주문한 음식을 기다리는데도 많은 시간이 소요된다.

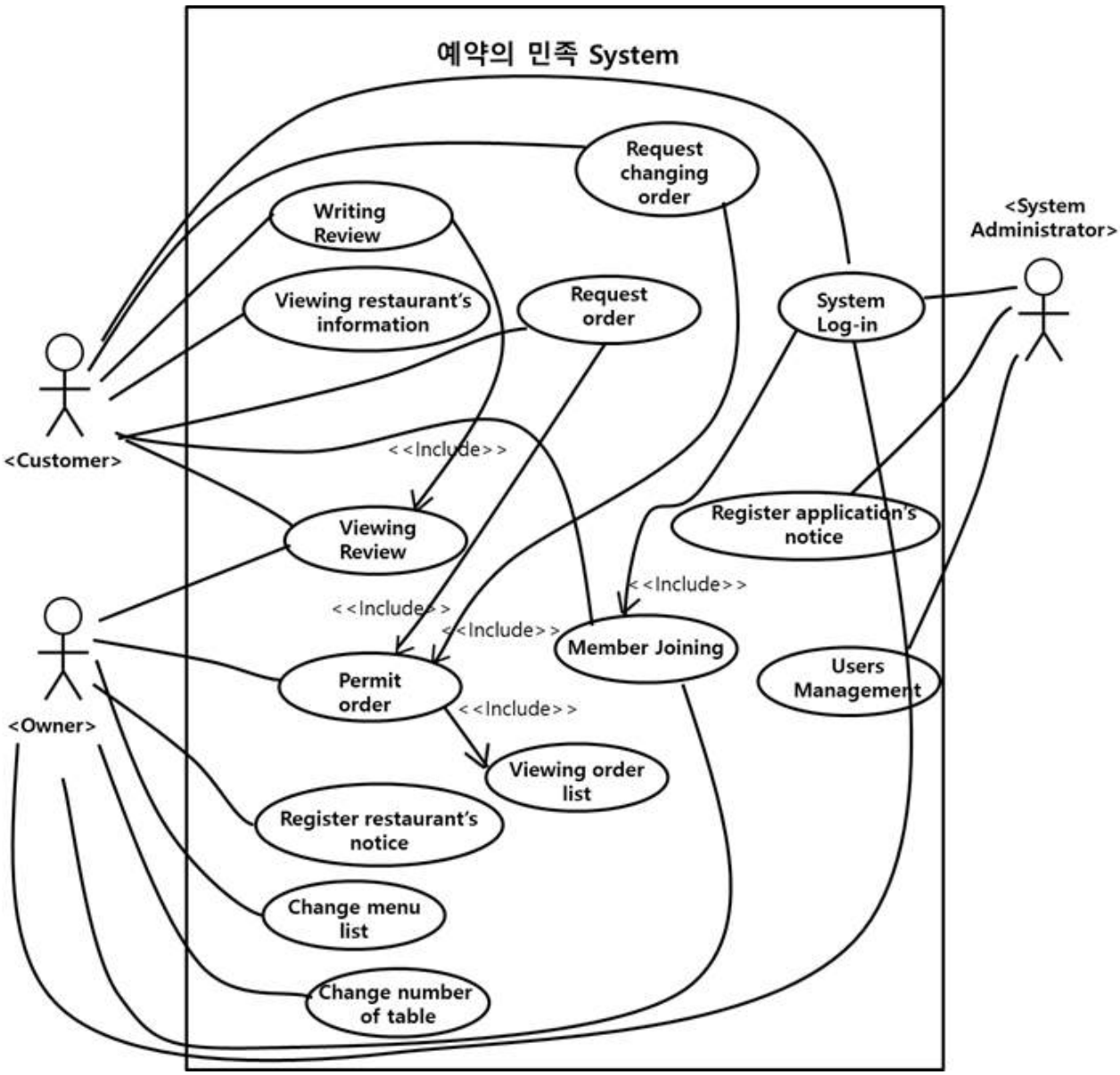
예약의 민족은 이러한 불편함을 해소해 보고자 기획된 프로젝트이다. 이는 음식점을 이용하는 고객과 음식점의 점주를 대상으로 하는데, 고객이 미리 어플을 통해 음식점에 예약을 한다는 것에 중점을 둔다.

먼저 고객이 현재 자신의 지역과 음식점의 종류를 선택하면 이 조건에 해당하는 음식점의 리스트가 보여 진다. 이때 원하는 음식점을 클릭하면 해당음식점의 내부사진, 리뷰 등을 볼 수 있으며 이를 고려해 음식점을 선택하고 현재 식당의 예약현황에 따라 원하는 시간에 미리 예약을 할 수 있다. 이때 메뉴의 이미지 등을 고려해 주문할 메뉴를 미리 선택하고 결제할 수 있기 때문에 점주는 이를 확인한 후 해당 시간에 맞춰 음식 준비가 가능하다. 이로써 고객은 예약한 시간에 식당에 방문해 기다리지 않고 바로 음식을 먹을 수 있다. 예약의 민족은 점주에게는 효율적인 식당운영을 고객에게는 식당에 대한 알권리 제공과 시간절약을 목표로 한다.

2. Use case analysis

- Build a use case diagram. - Make detailed description for each use case (Use case description) - 12pt, 160%.

2.1) Use Case Diagram



2.2) Use Case Description

#1. Log-in

Use case #1 : Log-in	
GENERAL CHARACTERISTICS	
Summary	사용자는 각종 기능을 활용하기 위해 예약의 민족 시스템에 로그인 할 수 있다.
Scope	예약의 민족
Level	User level
Author	all member
Last Update	2017. 10. 30.
Status	Analysis (Finalize)
Primary Actor	음식점 이용 고객, 음식점 점주,
Preconditions	예약의 민족 시스템에 회원가입이 되어 있는 상태여야 한다.
Trigger	예약의 민족 시스템에 로그인하여 사용자 인증을 받는다.
Success Post Condition	예약의 민족 시스템 사용 허가를 받고 서비스 이용이 가능하다.
Failed Post Condition	음식점 정보 보기 등 제한된 서비스만 이용가능하다.

MAIN SUCCESS SCENARIO	
Step	Action
S	회원(음식점 점주, 음식점 이용고객)이 예약의 민족에 로그인 한다 .
1	이 Use case는 회원이 예약의 민족에 로그인하기 위해 시스템을 시작했을 때 시작 된다 .
2	회원은 예약의 민족 첫 화면에서 'Login' 을 누른다.
3	ID 와 Password를 입력하고 로그인 버튼을 클릭 한다 .
4	예약의 민족에 등록된 회원인지 체크해보고 등록된 회원이라면 로그인에 성공한다 .
5	이 Use case는 로그인이 성공하면 끝난다.

EXTENSION SCENARIOS	
Step	Branching Action
3	3a. 아이디입력에 실패한다. 3a1. 아이디가 저장된 데이터와 불일치 할 시 에러 메세지를 출력한다. 3a2. 아이디가 저장된 데이터와 불일치 할 시 이전 페이지로 돌아가 재입력을 받도록 한다.

	3b. 비밀번호 입력에 실패한다. 3b1. 비밀번호가 저장된 데이터와 불일치 할 시 에러메세지를 출력한다. 3b2. 비밀번호가 저장된 데이터와 불일치 할 시 이전 페이지로 돌아가 재입력을 받도록 한다. 3c. 아이디입력란에 15자 이상 입력을 시도한다. 3c1. 입력자체를 불가능 하게 한다. 3d. 비밀번호입력란에 20자 이상 입력을 시도한다. 3d1. 입력자체를 불가능 하게 한다.

RELATED INFORMATION	
Performance	≤ 2 seconds
Frequency	음식점 점주 회원 : 하루에 평균 3번 음식점 이용 고객 회원 : 하루에 평균 1번
<Concurrency>	제한 없음
Due Date	2017.10.30.

#2. Member Joining

Use case #2 : Member Joining	
GENERAL CHARACTERISTICS	
Summary	사용자는 예약의 민족 시스템에 로그인하기 위해 회원 등록이 되지 않았을 시 회원에 등록하는 기능이다.
Scope	예약의 민족
Level	User level
Author	all member
Last Update	2017. 10. 30.
Status	Analysis (Finalize)
Primary Actor	음식점 이용 고객, 음식점 점주
Preconditions	예약의 민족 어플을 설치한 상태이다.
Trigger	예약의 민족 시스템에서 요구하는 사용자 정보를 입력 후 모두 검증받는다.
Success Post Condition	예약의 민족 시스템 회원으로 등록되어 로그인이 가능하다.
Failed Post Condition	회원으로 등록되지 못한다.

MAIN SUCCESS SCENARIO	
Step	Action
S	회원(음식점 점주, 음식점 이용고객)이 로그인을 하여 시스템을 이용하기 위해 회원가입을 한다.
1	이 Use case 는 회원이 예약의 민족시스템에 회원가입을 하기 위해 'JOIN US' 버튼을 눌렀을 때 시작 된다 .
2	회원(음식점의 점주, 음식점 이용 고객)은 회원가입 버튼을 누르고 점주용/고객용을 선택 한다 .
3	사용 할 아이디를 입력하고 중복확인을 한다.
4	사용 할 비밀번호를 입력한다.
5	예약의 민족에서 요구하는 필수 입력사항을 입력한다. (-음식점 이용 고객 : 이름, 이메일, 주소 등 -음식점 점주 : 음식점 이용고객 요소에서 가게이름, 사업자번호 등 추가사항)
6	회원가입 완료 버튼을 누른다.
7	회원가입 하시겠습니까? 메시지에 확인 버튼을 누른다.
8	회원가입이 완료된다.

EXTENSION SCENARIOS	
Step	Branching Action
3	3a. 아이디를 15자 이상 입력한다. 3a1. 입력자체를 불가능하게 한다. 3b. 중복되는 아이디가 있을 경우. 3b1. 경고문을 출력한다. 3b2. 재입력을 받는다. 4a. 비밀번호를 20자 이상 입력한다. 3b1. 입력자체를 불가능하게 한다. 7a. 아이디 중복확인을 하지 않고 완료 버튼을 누른다. 7a1. 경고문 출력 한다.

RELATED INFORMATION	
Performance	≤ 2 seconds
Frequency	처음 한번 가입 시만 이용(이후에는 0번)
<Concurrency>	제한 없음
Due Date	2017.10.30.

#3. Uploading picture

Use case #3 : Uploading picture	
GENERAL CHARACTERISTICS	
Summary	사용자는 식당의 썸네일이나 메뉴 사진을 업로드 하고 싶을 시 본인이 원하는 이미지를 업로드 시킬 수 있다.
Scope	예약의 민족
Level	User level
Author	all member
Last Update	2017. 10. 30.
Status	Analysis (Finalize)
Primary Actor	음식점 점주
Preconditions	사용자는 로그인 되어있는 상태여야 한다.
Trigger	사용자가 'restaurant image upload'나 'menu image upload'를 누르고 원하는 사진을 선택한다.
Success Post Condition	업로드한 이미지가 바뀐다.
Failed Post Condition	이미지 업로드에 실패한다.

MAIN SUCCESS SCENARIO	
Step	Action
S	사용자는 식당의 썸네일이나 메뉴 사진을 업로드 하고 싶을 시 본인이 원하는 이미지를 업로드 시킬 수 있다.
1	이 usecase는 음식점 점주가 식당의 메뉴나 썸네일 이미지를 변경하고 싶어 업로드 버튼을 누를 때 시작된다.
2	restaurant image upload나 menu image upload를 선택한다.
3	찾아보기를 클릭하여 이미지를 선택한다.
4	file upload를 클릭한다.
5	이미지 정보가 변경된다.

EXTENSION SCENARIOS	
Step	Branching Action
	4a. 이미지가를 선택하지 않고 업로드 버튼을 눌렀을 경우 4a1. 사진이 업로드 되지않고 홈 화면으로 돌아간다.

RELATED INFORMATION	
Performance	≤ 2 seconds
Frequency	1번이내
<Concurrency>	제한 없음
Due Date	2017.10.30.

#4. Rejection order

Use case #4 : Rejection Order	
GENERAL CHARACTERISTICS	
Summary	예약요청이 들어왔을 때 식당 상황에 따라서 예약 거절을 한다.
Scope	예약의 민족
Level	User level
Author	all member
Last Update	2017. 10. 30.
Status	Analysis (Finalize)
Primary Actor	음식점 점주
Preconditions	사용자가 로그인 되어있고 음식점 이용고객으로부터 주문(예약요청)이 들어온 상태이다.
Trigger	이용 고객으로부터 주문 요청이 들어와 음식점의 사정에 따라 예약 거절을 누르면
Success Post Condition	거절 처리가 되고 음식점 고객에게 결과가 전달된다.
Failed Post Condition	

MAIN SUCCESS SCENARIO	
Step	Action
S	음식점 이용 고객으로부터 들어온 예약요청을 거절한다.
1	이 usecase는 음식점 이용고객이 음식점 예약을 위해 예약 신청을 했을 때 시작된다.
2	음식점 점주가 음식점 이용고객으로부터 들어온 주문이 들어오면 프로그램을 통해 확인한다.
3	음식점 점주가 상황에 따라 승인할지 거절할지 판단 후 거절 버튼을 선택한다.
4	결과가 저장되고 음식점 이용고객에게 결과가 전송된다.

EXTENSION SCENARIOS	
Step	Branching Action
	.

RELATED INFORMATION

Performance	≤ 2 seconds
Frequency	20번 이상
<Concurrency>	제한 없음
Due Date	2017.10.30.

#5. Change menu list

Use case #5 : Change Menu list	
GENERAL CHARACTERISTICS	
Summary	음식점의 점주가 음식점의 메뉴를 변경한다.
Scope	예약의 민족
Level	User level
Author	all member
Last Update	2017. 10. 30.
Status	Analysis (Finalize)
Primary Actor	음식점 점주
Preconditions	사용자가 로그인 되어있는 상태이다.
Trigger	메뉴를 추가 하거나 삭제하면
Success Post Condition	메뉴 정보가 변경된다.
Failed Post Condition	변경내용은 저장되지 않는다.

MAIN SUCCESS SCENARIO	
Step	Action
S	음식점의 정보 중 메뉴 리스트를 추가하거나 삭제한다.
1	이 usecase는 음식점 점주가 음식점의 메뉴를 변경하기 위해 시스템에 로그인한 후 view menu 클릭했을 때 시작된다.
3	메뉴 추가 혹은 삭제를 누른다.
4	필수사항(메뉴이름, 사진 등)을 입력한 후 확인버튼을 누른다.(수정) 삭제할 메뉴에 체크한 후 확인 버튼을 누른다.(삭제)
5	'확인을 누른다.

EXTENSION SCENARIOS	
Step	Branching Action
	4a. 필수 입력사항을 모두 입력하지 않고 확인버튼을 누를 경우. 4a1. '필수 사항을 모두 입력해 주세요.' 메시지를 출력 후 이전 화면으로 복귀 4b. 입력 도중 뒤로가기 버튼을 누를 경우. 4b1. '입력한 사항이 모두 삭제됩니다.'라는 메시지를 출력한다. 4c. 삭제할 메뉴를 하나도 선택하지 않고 확인버튼을 누를 경우 4c1. '선택하신 메뉴가 없습니다.'라는 메시지를 출력한다.

RELATED INFORMATION	
Performance	≤ 2 seconds
Frequency	1번
<Concurrency>	제한 없음
Due Date	2017.10.30.

#6. Register application's notice

Use case #6 : Register application's notice	
GENERAL CHARACTERISTICS	
Summary	시스템 관리자가 어플의 공지사항을 수정한다.
Scope	예약의 민족
Level	System level
Author	all member
Last Update	2017. 10. 30.
Status	Analysis (Finalize)
Primary Actor	시스템 관리자
Preconditions	시스템 관리자가 로그인 되어있는 상태이다.
Trigger	관리자가 공지사항을 수정하면
Success Post Condition	디스플레이 되는 어플의 공지사항의 내용이 변경된다.
Failed Post Condition	

MAIN SUCCESS SCENARIO	
Step	Action
S	어플의 공지사항을 수정한다.
1	이 usecase는 시스템 관리자가 어플의 공지사항을 변경하기 위해 시스템에 로그인 한 후 공지사항 변경을 누를 때 시작된다.
2	공지사항 텍스트를 변경한다.
3	확인을 누른다.
4	변경된 공지사항 상태가 저장된다.

EXTENSION SCENARIOS	
Step	Branching Action
3	.

RELATED INFORMATION	
Performance	≤ 2 seconds
Frequency	하루에 한번
<Concurrency>	제한 없음
Due Date	2017.10.30.

#7. Register restaurant's notice

Use case #7 : Register restaurant's notice	
GENERAL CHARACTERISTICS	
Summary	음식점의 점주가 음식점의 공지사항을 수정한다.
Scope	예약의 민족
Level	User level
Author	all member
Last Update	2017. 10. 30.
Status	Analysis (Finalize)
Primary Actor	음식점 점주
Preconditions	사용자가 로그인 되어있는 상태이다.
Trigger	사용자가 공지사항을 수정하면
Success Post Condition	디스플레이 되는 공지사항의 내용이 변경된다.
Failed Post Condition	

MAIN SUCCESS SCENARIO	
Step	Action
S	음식점의 공지사항을 수정한다.
1	이 usecase는 음식점 점주가 공지사항을 변경하기 위해 시스템에 로그인 한 후 공지사항 변경을 누를 때 시작된다.
2	공지사항 텍스트를 변경한다.
3	확인을 누른다.
4	변경된 공지사항 상태가 저장된다.

EXTENSION SCENARIOS	
Step	Branching Action
	2a. 변경 중 확인을 누르지 않은 상태에서 뒤로가기 버튼을 누를 경우 2a1. '작성중인 내용이 삭제됩니다.'메세지를 출력한다.

RELATED INFORMATION	
Performance	≤ 2 seconds
Frequency	1번 이하
<Concurrency>	제한 없음
Due Date	2017.10.30.

#8. Viewing Review

Use case #8 : Viewing Review	
GENERAL CHARACTERISTICS	
Summary	사용자가 음식점의 고객이 등록한 리뷰를 본다.
Scope	예약의 민족
Level	User level
Author	all member
Last Update	2017. 10. 30.
Status	Analysis (Finalize)
Primary Actor	음식점 점주, 음식점 이용고객
Preconditions	사용자가 로그인 되어있는 상태이다.
Trigger	음식점 홈의 리뷰보기 메뉴에서 보고 싶은 리뷰를 클릭하면
Success Post Condition	전체 리뷰의 내용이 펼쳐진다.
Failed Post Condition	

MAIN SUCCESS SCENARIO	
Step	Action
S	음식점의 고객들이 작성한 리뷰를 본다.
1	이 usecase는 음식점 점주와 음식점 이용고객이 다른 이용고객들이 작성한 리뷰를 보기 위해 시스템에 로그인 한 후 리뷰 보기의 원하는 리뷰제목을 누를 때 시작된다.
2	원하는 리뷰의 전체 내용이 펼쳐지고 내용을 확인 할 수 있다.

EXTENSION SCENARIOS	
Step	Branching Action
	.

RELATED INFORMATION	
Performance	≤ 2 seconds
Frequency	50번 이상
<Concurrency>	제한 없음
Due Date	2017.10.30.

#9. Writing Review

Use case #9 : Writing Review	
GENERAL CHARACTERISTICS	
Summary	음식점의 이용고객이 식당에 대한 리뷰를 작성한다.
Scope	예약의 민족
Level	User level
Author	all member
Last Update	2017. 10. 30.
Status	Analysis (Finalize)
Primary Actor	음식점 이용고객
Preconditions	고객이 예약의 민족에서 로그인 되어 있는 상태이다.
Trigger	음식점의 홈 화면에서 리뷰쓰기를 클릭한 후 리뷰를 작성하면
Success Post Condition	사용자가 등록한 리뷰가 등록된다.
Failed Post Condition	작성한 리뷰가 정상적으로 등록되지 않는다.

MAIN SUCCESS SCENARIO	
Step	Action
S	음식점의 고객들이 이용했던 음식점의 리뷰를 쓴다.
1	이 usecase는 음식점 이용고객이 이용한 음식점의 리뷰를 쓰기 위해 시스템에 로그인 한 후 음식점 홈 화면에서 리뷰 등록 버튼을 누를 때 시작된다.
2	리뷰를 작성 할 수 있는 텍스트 화면에서 원하는 내용을 작성한다.
3	원하는 내용을 모두 작성 후 확인 버튼을 누른다.
4	리뷰 작성이 완료되고 이는 리뷰보기화면에서 확인 가능하다.

EXTENSION SCENARIOS	
Step	Branching Action
	3a. 변경 중 확인을 누르지 않은 상태에서 뒤로가기 버튼을 누를 경우 3a1. '작성중인 내용이 삭제됩니다.'메세지를 출력한다.

RELATED INFORMATION	
Performance	≤ 2 seconds
Frequency	50번이상
<Concurrency>	제한 없음
Due Date	2017.10.30.

#10. Request order

Use case #10 : Request order	
GENERAL CHARACTERISTICS	
Summary	음식점 이용 고객이 이용할 음식점을 고르고 이용시간과 자리를 선택해 예약 하는 과정이다.
Scope	예약의 민족
Level	User level
Author	all member
Last Update	2017. 10. 30.
Status	Analysis (Finalize)
Primary Actor	음식점 이용고객
Preconditions	사용자가 예약의 민족에 로그인 되어 있는 상태이다.
Trigger	요구하는 조건을 모두 작성 한 후 주문하기 버튼을 누른다.
Success Post Condition	음식점 점주가 주문승인을 하면 주문이 완료된다.
Failed Post Condition	음식점 점주가 주문거절을 하면 주문이 취소된다.

MAIN SUCCESS SCENARIO	
Step	Action
S	음식점 이용 고객이 이용할 음식점을 고르고 이용시간과 자리를 선택해 예약 하는 과정이다.
1	이 usecase는 음식점 이용고객이 음식점 예약을 하기 위해 시스템에 로그인 한 후 원하는 식당을 누를 때 시작된다.
2	원하는 메뉴를 장바구니에 넣는다.
3	주문하기 버튼을 누른다.
4	원하는 시간과 자리를 선택한다.
7	모든 과정이 완료되면 음식점 예약 요청이 완료된다.
8	음식점 점주가 주문 승인하면 예약이 완료되고 음식점 이용고객에게 안내 메시지가 출력된다.
9	음식점 점주가 주문을 거절하면 예약이 취소되고 음식점 이용고객에게 안내 메시지가 출력된다.

EXTENSION SCENARIOS	
Step	Branching Action
	5a. 필수 입력사항을 모두 입력하지 않았을 시. 5a1. 누락된 사항을 메시지로 출력해준다. 5a2. 확인 버튼을 누르면 누락된 첫 사항의 위치로 화면을 이동한다. 6a. 결제가 정상적으로 완료되지 않았을 시. 6a1. '결제가 정상적으로 완료되지 않았습니다.'라는 메시지를 출력한다. 6a2. 확인버튼을 누르면 사용자가 주문사항을 모두 입력했던 상태로 다시 복귀한다.

RELATED INFORMATION	
Performance	≤ 2 seconds
Frequency	1번
<Concurrency>	제한 없음
Due Date	2017.10.30.

#11. Viewing order list

Use case #11 : Viewing order list	
GENERAL CHARACTERISTICS	
Summary	음식점 점주가 현재 예약 완료 된 주문의 리스트를 본다.
Scope	예약의 민족
Level	User level
Author	all member
Last Update	2017. 10. 30.
Status	Analysis (Finalize)
Primary Actor	음식점 점주
Preconditions	사용자가 로그인 되어있는 상태이다.
Trigger	음식점 점주가 현재 'reservation check'를 클릭한다.
Success Post Condition	현재 주문접수된 내역이 모두 출력된다.
Failed Post Condition	

MAIN SUCCESS SCENARIO	
Step	Action
S	음식점 점주가 현재 예약 완료 된 주문의 리스트를 본다.
1	이 usecase는 음식점 점주가 현재 주문접수사항을 보기위해 'reservation check'를 클릭할 때 시작된다.
2	현재 예약 접수 된 목록이 시간 순서대로 출력된다.

EXTENSION SCENARIOS	
Step	Branching Action
	.

RELATED INFORMATION	
Performance	≤ 2 seconds
Frequency	50번 이상

<Concurrency>	제한 없음
Due Date	2017.10.30.

#12. Viewing restaurant's information

Use case #12 : Viewing restaurant's information	
GENERAL CHARACTERISTICS	
Summary	음식점 이용고객이 예약할 식당을 고르기 위해 음식점의 정보를 열람한다.
Scope	예약의 민족
Level	User level
Author	all member
Last Update	2017. 10. 30.
Status	Analysis (Finalize)
Primary Actor	음식점 이용 고객
Preconditions	사용자가 로그인 되어있는 상태이다.
Trigger	음식점 이용고객이 음식점의 리스트 중 원하는 음식점을 클릭하면
Success Post Condition	음식점의 제공 정보가 모두 출력된다.
Failed Post Condition	

MAIN SUCCESS SCENARIO	
Step	Action
S	음식점 이용고객이 음식점의 정보를 열람한다.
1	이 usecase는 음식점 이용고객이 예약할 음식점을 선택하기 위해 시스템에 로그인 했을 때 시작된다.
2	자신이 식사를 할 예정인 음식점 종류를 선택한다.
3	선택한 종류의 음식점이 출력된다.
4	원하는 음식점을 클릭한다.
5	음식점의 기본정보와 메뉴종류와 가격(음식 사진 포함)등의 정보가 제공된다.

EXTENSION SCENARIOS	
Step	Branching Action
	.

RELATED INFORMATION	
Performance	≤ 2 seconds
Frequency	20번
<Concurrency>	제한 없음
Due Date	2017.10.30.

#13. Register Restaurant

Use case #13 : Register Restaurant	
GENERAL CHARACTERISTICS	
Summary	시스템 관리자가 임의로 식당 목록을 추가 할 수 있다.
Scope	예약의 민족
Level	System level
Author	all member
Last Update	2017. 10. 30.
Status	Analysis (Finalize)
Primary Actor	시스템 관리자
Preconditions	시스템 관리자가 시스템에 로그인 되어 있어야 한다.
Trigger	시스템 관리자가 음식점을 목록으로 등록하거나 등록된 음식점을 삭제한다.
Success Post Condition	restaurantDB목록에서 식당이 추가되거나 삭제된다.
Failed Post Condition	

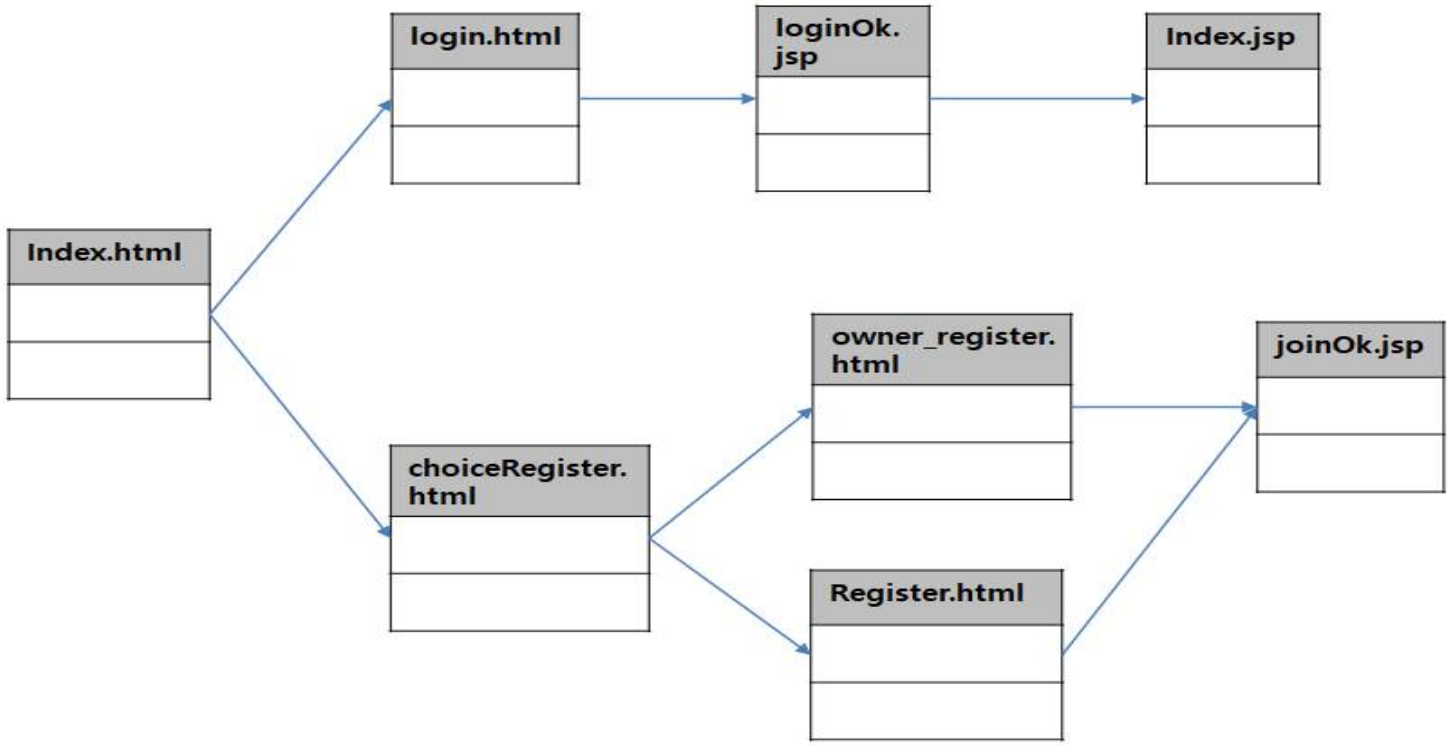
MAIN SUCCESS SCENARIO	
Step	Action
S	시스템 관리자가 임의로 음식점 목록을 업데이트 하거나 삭제 시킬 수 있다.
1	이 usecase는 시스템 관리자가 회원목록을 관리하기 위해 시스템에 로그인 하면 시작된다.
2	-시스템 관리자가 시스템의 데이터베이스에서 등록할 음식점 정보를 입력하고 음식점을 등록한다.(음식점등록) -시스템 관리자가 시스템의 데이터베이스에서 삭제할 음식점을 삭제한다.(음식점탈퇴)
3	시스템 관리자가 수정한 대로 데이터베이스의 음식점 목록이 수정된다.

EXTENSION SCENARIOS	
Step	Branching Action
	.

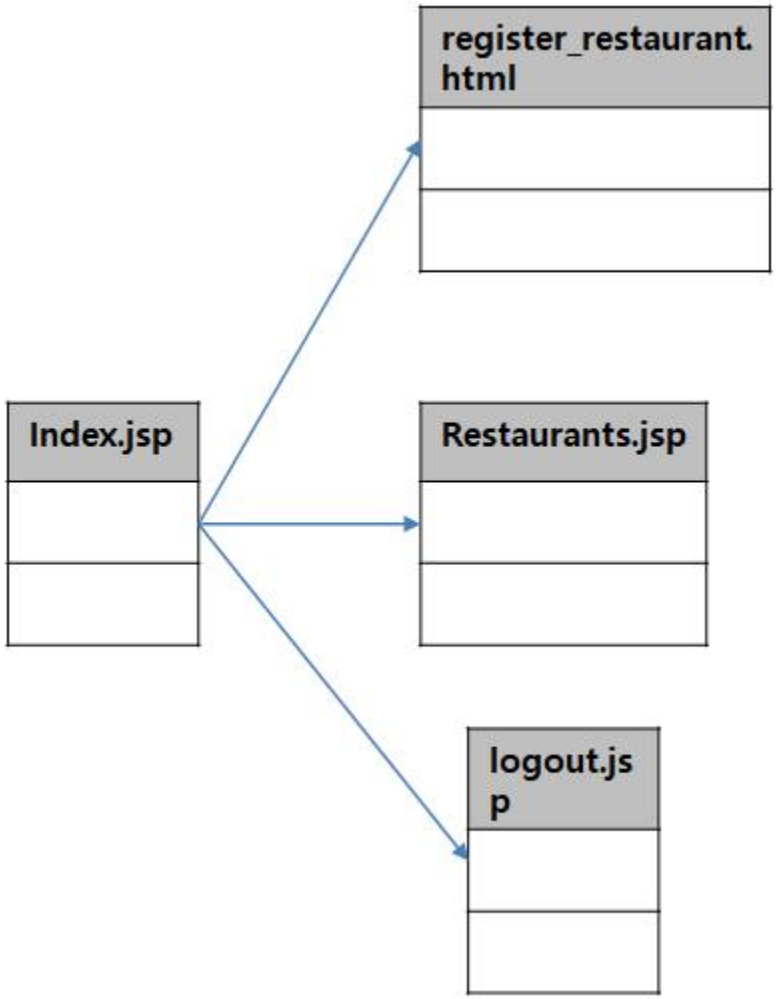
RELATED INFORMATION	
Performance	≤ 2 seconds
Frequency	5번
<Concurrency>	제한 없음
Due Date	2017.10.30.

3. Class diagram

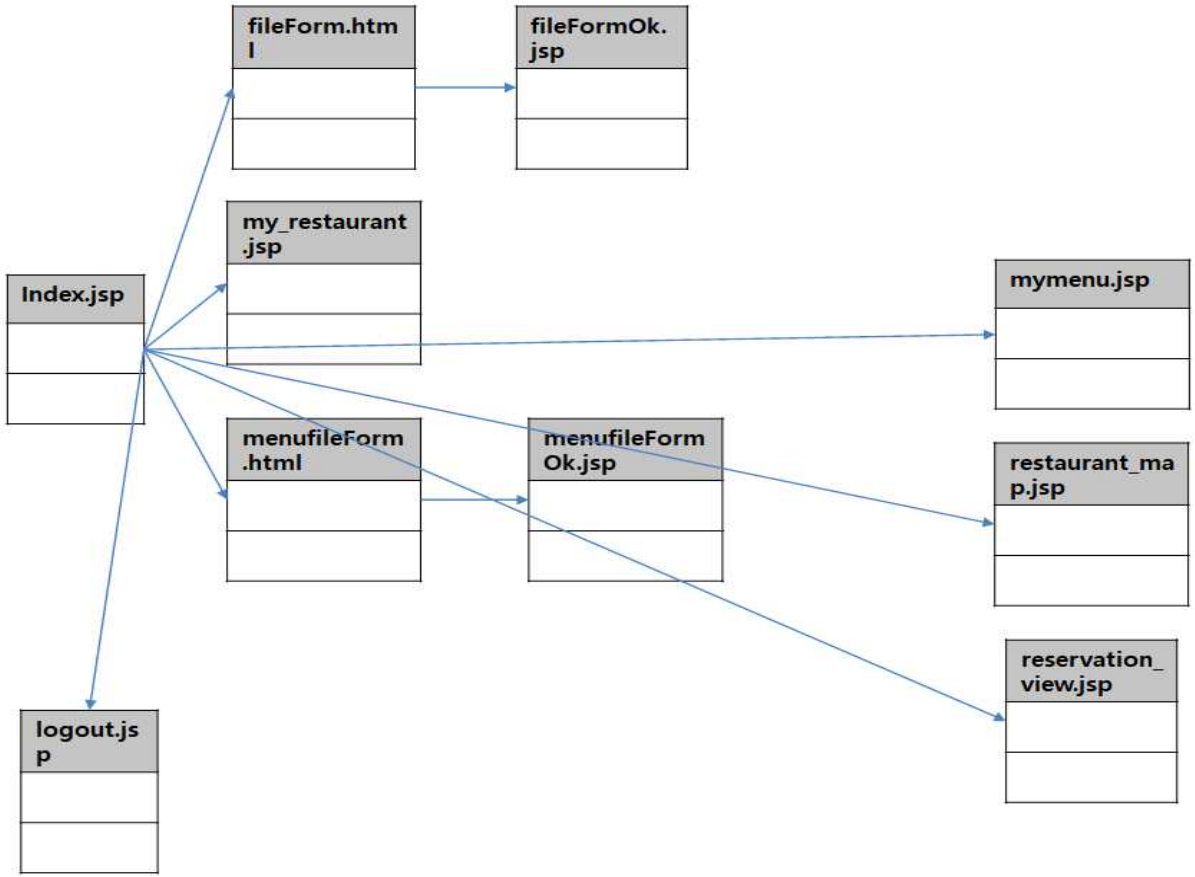
<로그인뷰>



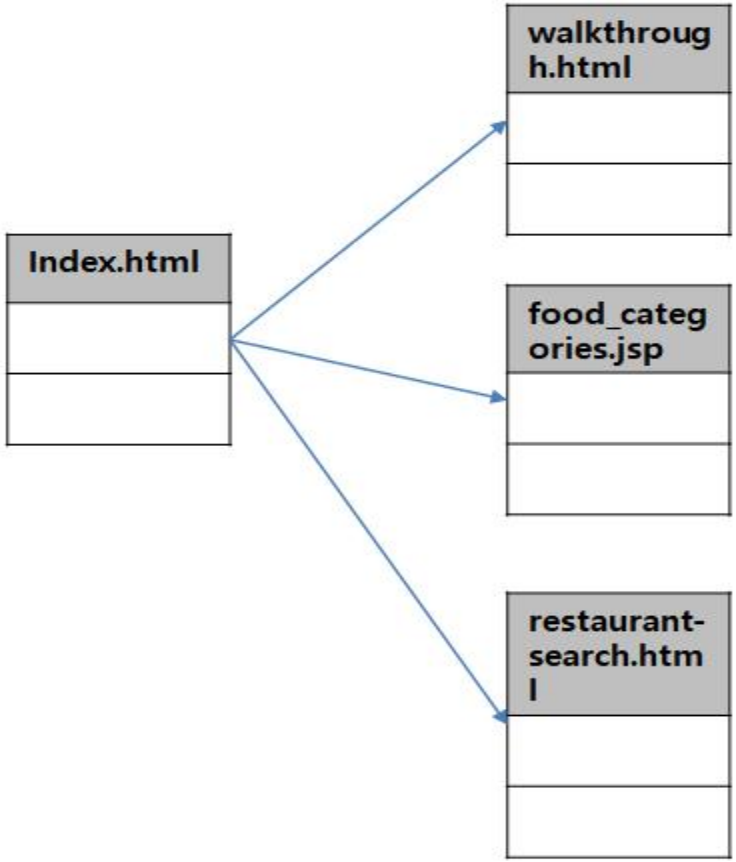
<루트 권한 기능 뷰>


register_restaurant.html
Index.jsp
Restaurants.jsp
logout.jsp

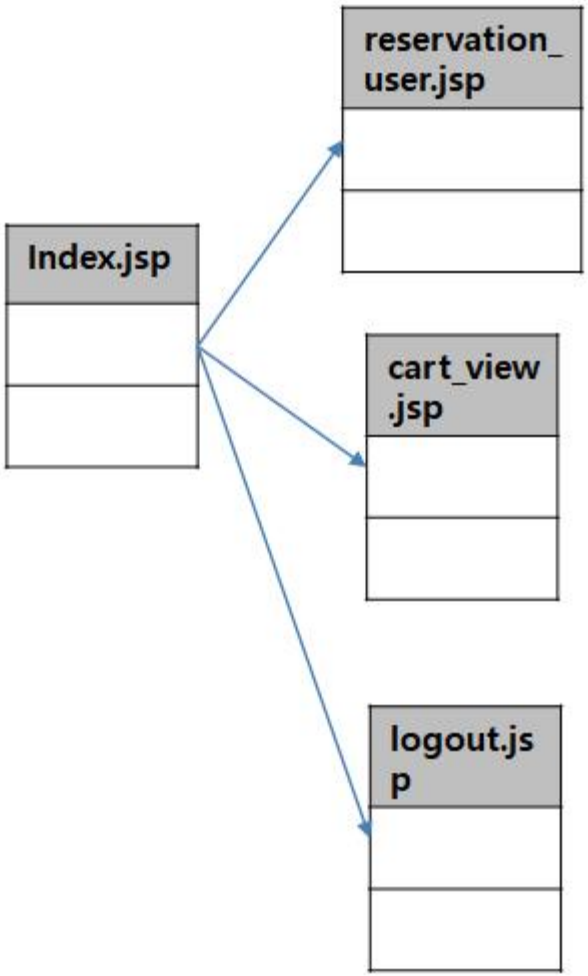
<점주권한 기능 뷰>


fileForm.html
fileFormOk.jsp
my_restaurant.jsp
Index.jsp
mymenu.jsp
menufileForm.html
menufileFormOk.jsp
restaurant_map.jsp
reservation_view.jsp
logout.jsp

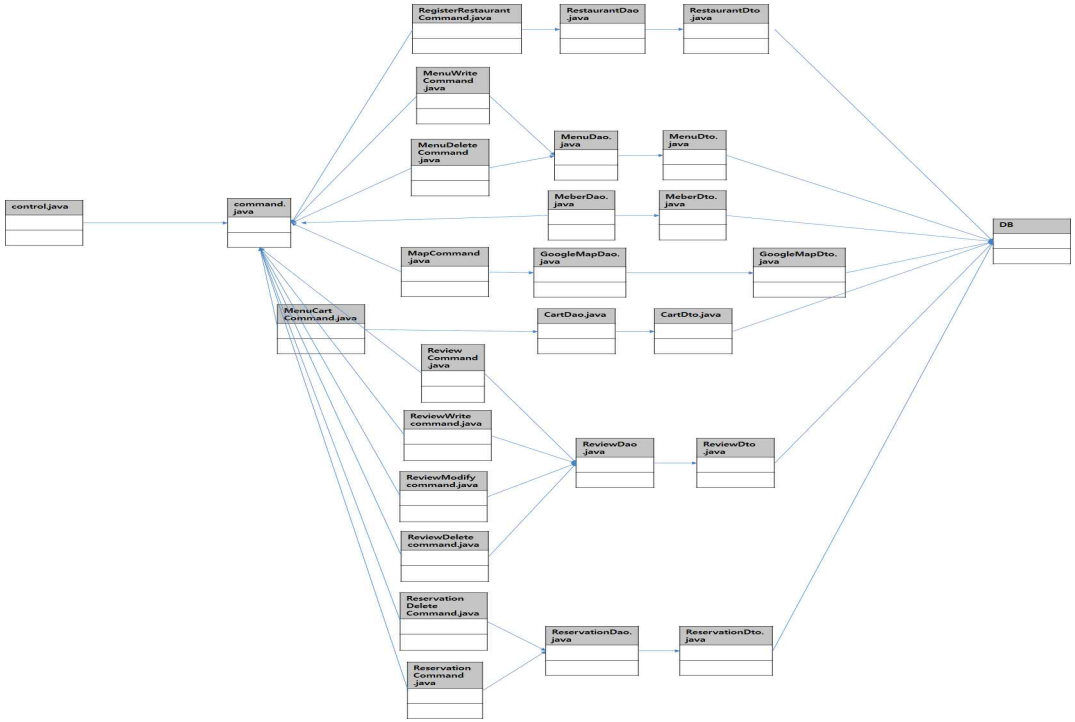
<기본 index.html 뷰>



<일반유저 기능 뷰>



<DB 컨트롤&모델>


control.java
command.java
RegisterRestaurant Command.java
RestaurantDao .java
RestaurantDto .java
MenuWrite Command .java
MenuDelete Command .java
MenuDao. java
MenuDto. java
MeberDao. java
MeberDto. java
MapCommand .java
GoogleMapDao. java
GoogleMapDto. java
MenuCart Command.java
CartDao.java
CartDto.java
Review Command .java
ReviewWrite command.java
ReviewModify command.java
ReviewDelete command.java
ReviewDao .java
ReviewDto .java
Reservation Delete Command.java
Reservation Command .java
ReservationDao. java
ReservationDto. java
DB

3.1) MVC 중 Control

3.1.1) control

Control
#doGet(request : HttpServletRequest, response : HttpServletResponse) : void #doPost(request : HttpServletRequest, response : HttpServletResponse) : void -actionDo(request : HttpServletRequest, response : HttpServletResponse) : void

Servlet:control		
Attributes		
Methods	#doGet(request : HttpServletRequest, response : HttpServletResponse) : void	Get방식으로 요청이 되었을 때 호출되는 메서드, doGet 메서드는 다시 actionDo메서드를 호출한다.
	#doPost(request : HttpServletRequest, response : HttpServletResponse) : void	Post방식으로 요청이 되었을 때 호출되는 메서드, doPost 메서드는 다시 actionDo메서드를 호출한다.
	-actionDo(request : HttpServletRequest, response : HttpServletResponse) : void	doGet메서드와 doPost메서드에서 호출되는 메서드, Get방식과 Post 방식중 어느 방식으로 요청이와도 actionDo 메서드에서 요청이 처리가 된다.
Others	서블릿으로 구현	

3.2) MVC 중 Model

3.2.1) Command (인터페이스)

Command (인터페이스)
+execute(request : HttpServletRequest, response : HttpServletResponse) : void

Interface:Command		
Attributes		
Methods	+execute(request : HttpServletRequest,	MVC중 Model에서의 기본이 되는 규칙을 정의해놓은 인

	response : HttpServletResponse) : void	터페이스, 서로 다른 기능의 class들이 동일한 execute메서드를 호출하여 각각의 기능을 수행한다.
Others	java로 구현	

3.2.2) command 인터페이스를 상속받은 클래스들

(MapCommand, MenuCartCommand, MenuDeleteCommand, MenuWriteCommand, RegisterRestaurantCommand, ReservationCommand, ReservationDeleteCommand, ReviewModifyCommand, ReviewWriteCommand)

MapCommand, MenuCartCommand, MenuDeleteCommand, MenuWriteCommand, RegisterRestaurantCommand, ReservationCommand, ReservationDeleteCommand, ReviewModifyCommand, ReviewWriteCommand
+execute(request : HttpServletRequest, response : HttpServletResponse) : void

Class:(괄호안의 클래스들)		
Attributes		
Methods	+execute(request : HttpServletRequest, response : HttpServletResponse) : void	Command 인터페이스의 execute를 위의 클래스들이 각각 상속받아 각각의 다른 기능들로 오버라이딩 하여 수행되는 메서드
Others	java로 구현	

3.2.3) CartDao

CartDao
+MENU_JOIN_SUCCESS : int -instance : CartDao
-CartDao() : void +getInstance() : CartDao +insertCart(dto : CartDto) : int +lists() : ArrayList<CartDto> +list(userId : String) : ArrayList<CartDto> -getConnection() : Connection

Class:CartDao		
Attributes	+MENU_JOIN_SUCCESS : int	메뉴가 데이터베이스에 성공적으로 insert 되었을 때 사용되는 변수이다. 값은 1이다. 이 값을 반환한다.
	-instance : CartDao	CartDao객체를 생성하여 instance 에 저장한다.
Methods	-CartDao() : void	CartDao의 생성자이다.
	+getInstance() : CartDao	instance를 반환하는 메서드이다. instance에는 CartDao 객체가 생성되어 있다.
	+insertCart(dto : CartDto) : int	매개변수로 받은 CartDto객체인 dto의 값을 데이터베이스에 insert하는 메서드이다. 데이터베이스에 성공적으로 insert가 되면 1을, 그렇지않다면 0을 반환한다.
	+lists() : ArrayList<CartDto>	데이터베이스에 있는 레코드를 가져와서 레코드의 리스트를 만들어 반환하는 메서드이다.
	+list(userId : String) : ArrayList<CartDto>	매개변수로받은 userId에 해당하는 레코드의 리스트를 가져와서 CartDao 객체의 리스트에 레코드의 각각의 데이터를 저장하여 반환하는 메서드이다.
	-getConnection() : Connection	데이터베이스를 연결할 때 호출하는 메서드이다.
Others	java로구현	

3.2.4) GoogleMapDao

GoogleMapDao
-instance : GoogleMapDao
-GoogleMapDao() : void +getInstance() : GoogleMapDao +getBusinessNumGoogle(businessNumber : String) : GoogleMapDto +insertMap(dto : GoogleMapDto) : int -getConnection() : Connection

Class:GoogleMapDao		
Attributes	-instance : GoogleMapDao	GoogleMapDao객체를 생성

		하여 instance 에 저장한다.
Methods	-GoogleMapDao() : void	GoogleMapDao의 생성자이다.
	+getInstance() : GoogleMapDao	instance를 반환하는 메서드이다. instance에는 GoogleMapDao객체가 생성되어 있다.
	+insertMap(dto : GoogleMapDto) : int	매개변수로 받은 GoogleMapDto객체인 dto의 값을 데이터베이스에 insert하는 메서드이다. 데이터베이스에 성공적으로 insert가 되면 1을, 그렇지않다면 0을 반환한다.
	+getBusinessNumGoogle(businessNumber : String) : GoogleMapDto	매개변수로 받은 businessNumber에 해당하는 레코드를 가져와서 GoogleMapDto 객체에 레코드의 데이터를 저장하여 그 객체를 반환하는 메서드이다.
	-getConnection() : Connection	데이터베이스를 연결할 때 호출하는 메서드이다.
Others	java로 구현	

3.2.5) MemberDao

MemberDao
+MEMBER_NONEXISTENT : int +MEMBER_EXISTENT : int +MEMBER_JOIN_FAIL : int +MEMBER_JOIN_SUCCESS : int +MEMBER_LOGIN_PW_NO_GOOD : int +MEMBER_LOGIN_SUCCESS : int +MEMBER_LOGIN_IS_NOT : int +MEMBER_USER_IS_SUCCESS : int +MEMBER_OWNER_IS_SUCCESS : int -instance : MemberDao
-MemberDao() : void +getInstance() : MemberDao +insertMember(dto : MemberDto) : int +confirmId(id : String) : int +userCheck(id : String, pw : String) : int +businessNumberCheck(id : String) : int +getMember(id : String) : MemberDto +getBusinessNumMember(num : String) : MemberDto +updateMember(dto : MemberDto) : int -getConnection() : Connection

Class:MemberDao		
Attributes	+MEMBER_NONEXISTENT : int	회원이 없을시 사용되는 변수이다.
	+MEMBER_EXISTENT : int	회원이 존재할 시 사용되는 변수이다.
	+MEMBER_JOIN_FAIL : int	회원의 접속이 실패할 시 사용되는 변수이다.
	+MEMBER_JOIN_SUCCESS : int	회원의 접속이 성공할 시 사용되는 변수이다.
	+MEMBER_LOGIN_PW_NO_GOOD : int	회원의 비밀번호가 맞지 않을시 사용되는 변수이다.
	+MEMBER_LOGIN_SUCCESS : int	회원의 로그인이 성공할 시 사용되는 변수이다.
	+MEMBER_LOGIN_IS_NOT : int	회원의 로그인이 실패할 시 사용되는 변수이다.
	+MEMBER_USER_IS_SUCCESS : int	회원이 손님인지를 판별하는 변수이다.
	+MEMBER_OWNER_IS_SUCCESS : int	회원이 점주인지를 판별하는 변수이다.
	-instance : MemberDao	MemberDao객체를 생성하여 instance 에 저장한다.

Methods	-MemberDao() : void	MemberDao의 생성자이다.
	+getInstance() : MemberDao	instance를 반환하는 메서드이다. instance에는 MemberDao객체가 생성되어 있다.
	+insertMember(dto : MemberDto) : int	매개변수로 받은 MemberDto객체인 dto의 값을 데이터베이스에 insert하는 메서드이다. 데이터베이스에 성공적으로 insert가 되면 1을, 그렇지않다면 0을 반환한다.
	+confirmId(id : String) : int	매개변수로 받은 id값에 해당하는 레코드가 데이터베이스에 존재하는지 확인하는 메서드이다. 데이터베이스에 존재한다면 1을 없다면 0을 반환한다.
	+userCheck(id : String, pw : String) : int	매개변수로 받은 id값에 해당하는 레코드를 데이터베이스에서 가져오고, 가져온 레코드의 패스워드와 매개변수로 받은 패스워드의 값이 같다면 1을 반환하고, 같지않다면 -1을 id값에 해당하는 레코드는 존재하지만 매개변수로받은 패스워드의 값과 다른 경우 0을 반환한다.
	+businessNumberCheck(id : String) : int	매개변수로 받은 id에 해당하는 레코드를 데이터베이스에서 가져와서 레코드의 businessNumber가 -1 이면 점주가아닌 일반 손님이므로, 일반손님이면 1을 반환하고, businessNumber가 -1 이아니라면 점주이므로 점주라면 0을 반환하고, 매개변수로받은 아이디에 해당하는 레코드가 데이터베이스에 존재하지 않는다면 -1을 반환한다.
	+getMember(id : String) :	매개변수로 받은 id값에 해

	MemberDto	당하는 레코드를 MemberDto 객체에 각각의 데이터를 set을 하여 그 객체를 반환하는 메서드이다.
	+getBusinessNumMember(num : String) : MemberDto	매개변수로 받은 businessNumber값에 해당하는 레코드를 MemberDto 객체에 각각의 데이터를 set을 하여 그 객체를 반환하는 메서드이다.
	+updateMember(dto : MemberDto) : int	매개변수로받은 MemberDto 객체인 dto의 값으로 데이터베이스의 레코드를 update 하는 메서드이다.
	-getConnection() : Connection	데이터베이스를 연결할 때 호출하는 메서드이다.
Others	java로 구현	

3.2.6) MenuDao

MenuDao
+MENU_JOIN_SUCCESS : int +instance : MenuDao
-MenuDao() : void +getInstance() : MenuDao +insertMenu(dto : MenuDto) : int +lists() : ArrayList<MenuDto> +MenuCheck(businessNumber : String, menuNumber : String) : MenuDto +MenuCnt(businessNumber : String) : int +deleteMenu(businessNumber : String, menuNumber : String) : int -getConnection() : Connection

Class:MenuDao		
Attributes	+MENU_JOIN_SUCCESS : int	메뉴가 데이터베이스에 성공적으로 insert되었을 시 사용되는 변수이다.
	+instance : MenuDao	MenuDao객체를 생성하여 instance 에 저장한다.
Methods	-MenuDao() : void	MenuDao의 생성자이다.
	+getInstance() : MenuDao	instance를 반환하는 메서드이다. instance에는

		MenuDao 객체가 생성되어 있다.
	+insertMenu(dto : MenuDto) : int	매개변수로받은 MenuDto객체인 dto의 데이터를 데이터베이스에 insert하는 메서드이다. insert가 성공적으로 된다면 1을, 아니면 0을 반환한다.
	+lists() : ArrayList<MenuDto>	menu테이블의 모든 레코드를 데이터베이스에서 가져와서 MenuDto에 각각의 레코드를 set하고 레코드 데이터를 저장한 MenuDto객체의 리스트를 반환한다.
	+MenuCheck(businessNumber : String, menuNumber : String) : MenuDto	매개변수로 받은 businessNumber와 menuNumber 의 and 값에 해당하는 레코드를 가져와서 MenuDto 객체에 데이터를 저장하고 그 객체를 반환한다.
	+MenuCnt(businessNumber : String) : int	매개변수로 받은 businessNumber에 해당하는 레코드의 수를 반환한다.
	+deleteMenu(businessNumber : String, menuNumber : String) : int	매개변수로 받은 businessNumber와 menuNumber의 and 값에 해당하는 레코드를 삭제한다.
	-getConnection() : Connection	데이터베이스를 연결할 때 호출하는 메서드이다.
Others	java로 구현	

3.2.7) ReservationDao

ReservationDao
-instance : ReservationDao
-ReservationDao() : void +getInstance : ReservationDao +list(businessNumber : String) : ArrayList<ReservationDto> +lists() : ArrayList<ReservationDto> +deleteReservation(businessNumber : String, userId : String) : int -getConnection() : Connection

Class:ReservationDao		
Attributes	-instance : ReservationDao	MenuDao객체를 생성하여 instance 에 저장한다.
Methods	-ReservationDao() : void	ReservationDao의 생성자이다.
	+getInstance : ReservationDao	instance를 반환하는 메서드이다. instance에는 Reservation객체가 생성되어 있다.
	+list(businessNumber : String) : ArrayList<ReservationDto>	매개변수로 받은 businessNumber에 해당하는 레코드의 리스트를 가져와서 Reservation 객체의 리스트에 레코드의 데이터를 저장하여 반환하는 메서드이다.
	+lists() : ArrayList<ReservationDto>	reservation 테이블의 모든 레코드를 가져와서 Reservation 객체의 리스트에 레코드 각각의 데이터를 저장하여 반환하는 메서드이다.
	+deleteReservation(businessNumber : String, userId : String) : int	매개변수로 받은 businessNumber와 userId의 and 에 해당하는 레코드를 삭제시키는 메서드이다.
	-getConnection() : Connection	데이터베이스를 연결할 때 호출하는 메서드이다.
Others	java 로구현	

3.2.8) RestaurantDao

RestaurantDao
-dataSource : DataSource
+RestaurantDao() : void +insertRestaurant(restaurantName : String, category : String, businessNumber : String) +lists() : ArrayList<RestaurantDto> +getBuinessNumMember(businessNumber : String) : RestaurantDto

Class:RestaurantDao

Attributes	-dataSource : DataSource	데이터베이스 연결에 사용할 변수이다.
Methods	+RestaurantDao() : void	RestaurantDao객체의 생성자이다. 데이터베이스와 연결을 한다.
	+insertRestaurant(restaurantName : String, category : String, businessNumber : String)	매개변수로 받은 restaurantName 과 category와 businessNumber를 데이터베이스에 insert한다.
	+lists() : ArrayList<RestaurantDto>	restaurant 테이블에 있는 모든 레코드를 가져와서 RestaurantDto 리스트에 각각의 레코드 데이터를 넣고 리스트를 반환한다.
	+getBusinessNumMember(businessNumber : String) : RestaurantDto	매개변수로 받은 businessNumber에 해당하는 레코드를 가져온다. 가져온 레코드의 값을 RestaurantDto에 set하고 반환한다.
Others	java로 구현	

3.2.9) ReviewDao

ReviewDao
-instance : ReviewDao
-ReviewDao() : void +getInstance() : ReviewDao +getBusinessNumReview(businessNumber : String , userId : String) : ReviewDto +insertReview(dto : ReviewDto) : int +lists() : ArrayList<ReviewDto> +list(businessNumber : String) : ArrayList<ReviewDto> +getReviewCount(businessNumbeR : String) ; int +updateReview(dto : ReviewDto) : int

+deleteReview(businessNumber : String , userId : String) : int -getConnection() : Connection

Class:ReviewDao		
Attributes	-instance : CartDao	ReviewDao객체를 생성하여 instance 에 저장한다.
Methods	-ReviewDao() : void	ReviewDao의 생성자이다.
	+getInstance() : ReviewDao	instance를 반환하는 메서드이다.instance에는 ReviewDao 객체가 생성되어 있다.
	+getBusinessNumReview(businessNumber : String , userId : String) : ReviewDto	매개변수로 받은 businessNumber와 userId의 and 에 해당하는 레코드를 반환한다.
	+insertReview(ReviewDto dto) : int	매개변수로 받은 Dto객체인 dto의 값을 데이터베이스에 insert하는 메서드이다. 데이터베이스에 성공적으로 insert가 되면 1을, 그렇지않다면 0을 반환한다.
	+getReviewCount(String businessNumbeR) ; int	매개변수로 받은 businessNumber에 해당하는 레코드의 개수를 봔한하는 메서드 이다..
	+updateReview(ReviewDto dto) : int	매개변수로 받은 Dto객체인 dto의 값을 데이터베이스에 update하는 메서드이다.
	+deleteReview(businessNumber : String , userId : String) : int	매개변수로 받은 Dto객체인 dto의 값을 데이터베이스에 delete하는 메서드이다.
	+lists() : ArrayList<ReviewDto>	데이터베이스에 있는 레코드를 가져와서 레코드의 리스트를 만들어 반환하는 메서드이다.
	+list(String businessNumber) : ArrayList<CartDto>	매개변수로 받은 businessNumber에 해당하는 레코드의 리스트를 가져와서 ReviewDto 객체의 리스트에 레코드의 각각의 데이터를 저장하여 반환하는 메서드이다.
	-getConnection() :	데이터베이스를 연결할 때

	Connection	호출하는 메서드이다.
Others	java로 구현	

3.2.10) CartDto

CartDto
+restaurantName : String +businessNumber : String +menuName : String +menuText : String +menuPrice : String +menuNumber : String +userId : String
-CartDto(restaurantName : String ,businessNumber : String ,menuName : String ,menuText : String ,menuPrice : String ,menuNumber : String ,userId : String) : void +getRestaurantName() : String +setRestaurantName(restaurantName : String) : void +getBusinessNumber() : String +setBusinessNumber(businessNumber : String) :void +getMenuName() : String +setMenuName(menuName : String) : void +getMenuText() : String +setMenuText(menuText : String) : void +getMenuPrice() : String +setMenuPrice(menuPrice : String) : void +getMenuNumber() : String +setMenuNumber(menuNumber : String) : void +getUserId() : String +setUserId(userId : String) : void

Class:CartDto		
Attributes	+restaurantName : String	식당의 이름을 나타낸다.
	+businessNumber : String	식당의 번호를 나타낸다.
	+menuName : String	메뉴의 이름을 나타낸다.
	+menuText : String	메뉴의 가격을 나타낸다.
	+menuPrice : String	메뉴의 가격을 나타낸다.
	+menuNumber : String	메뉴의 번호를 나타낸다.
	+userId : String	사용자의 id 이름을 나타낸다.
Methods	-CartDto(restaurantName : String ,businessNumber : String ,menuName :	Cart의 정보를 받아와 각 변수에 저장한다.

	String ,menuText : String ,menuPrice : String ,menuNumber : String ,userId : String) : void	
	+getRestaurantName() : String	restaurantName를 반환하는 메서드이다.
	+setRestaurantName(restaurantName : String) : void	매개변수로 받은 restaurantName를 restaurantName에 저장하는 메서드이다.
	+getBusinessNumber() : String	businessNumber를 반환하는 메서드이다.
	+setBusinessNumber(businessNumber : String) :void	매개변수로 받은 businessNumber를 businessNumber 에 저장하는 메서드이다.
	+getMenuName() : String	menuName를 반환하는 메서드이다.
	+setMenuName(menuName : String) : void	매개변수로 받은 menuName를 menuName에 저장하는 메서드이다.
	+getMenuText() : String	menuTex를 반환하는 메서드이다.
	+setMenuText(menuText : String) : void	매개변수로 받은 menuTex를 menuTex에 저장하는 메서드이다.
	+getMenuPrice() : String	menuPrice를 반환하는 메서드이다.
	+setMenuPrice(menuPrice : String) : void	매개변수로 받은 menuPrice를 menuPrice에 저장하는 메서드이다.
	+getMenuNumber() : String	menuNumber를 반환하는 메서드이다.
	+setMenuNumber(menuNumber : String) : void	매개변수로 받은 menuNumber를 menuNumber에 저장하는 메서드이다.
	+getUserId() : String	userid을 반환하는 메서드이다.

	+setUserId(userId : String) : void	매개변수로 받은 userId를 userId에 저장하는 메서드이다.
Others	java로 구현	

※나머지 dto 들도 위와 동일한 개념, 예를들어, 데이터베이스에서 레코드를 받을 때, 그 레코드 각각의 속성의 데이터를 dto에 저장한다.

3.3) MVC 중 View

3.3.1) cart_view

cart_view
+cartDao.getInstance() : CartDao +session.getAttribute("문장“) : Object +request.getRealPath("문장“) : String +file.list() : String[] +cartDao.list(id : String) : ArrayList<CartDto> +cartDto.getMenuName() : String +cartDto.getMenuPrice() : String +cartDto.getMenuText() : String

Class:cart_view		
Attributes		
Methods	+cartDao.getInstance() : CartDao	cartDao객체를 가져온다.
	+session.getAttribute("문장“) : Object	문장에 해당하는 session의 값을 반환한다.
	+request.getRealPath("문장“) : String	문장에 해당하는 파일의 절대경로를 얻어온다.
	+file.list() : String[]	사진파일안에 있는 파일이름들을 리스트로 가져온다.
	+cartDao.list(id : String) : ArrayList<CartDto>	인수로 준 아이디와 같은 레코드들을 데이터베이스에서 가져온다.
	+cartDto.getMenuName() : String	cartDto객체에 저장된 메뉴 이름을 반환한다.
	+cartDto.getMenuPrice() : String	cartDto객체에 저장된 가격을 반환한다.

	+cartDto.getMenuText() : String	cartDto객체에 저장된 메뉴의 텍스트를 반환한다.
Others	jsp로 구현한다.	

3.3.2) detail_menu

detail_menu
+MenuDao.getInstance() : MenuDao +request.getRealPath("문장“) : String +request.getParameter("문장“) : String +MenuDao.lists(id : String) : ArrayList<MenuDao> +MenuDao.getMenuName() : String +MenuDao.getMenuPrice() : String +MenuDao.getMenuText() : String

Class:detail_menu		
Attributes		
Methods	+MenuDao.getInstance() : MenuDao	MenuDao객체를 가져온다.
	+request.getRealPath("문장“) : String	문장에 해당하는 파일의 절대경로를 얻어온다.
	+request.getParameter("문장“) : String	문장에 해당하는 request의 값을 반환한다.
	+MenuDao.lists(id : String) : ArrayList<MenuDao>	인수로 준 아이디와 같은 레코드들을 데이터베이스에서 가져온다.
	+MenuDao.getMenuName() : String	MenuDao객체에 저장된 메뉴이름을 반환한다.
	+MenuDao.getMenuPrice() : String	MenuDao객체에 저장된 가격을 반환한다.
	+MenuDao.getMenuText() : String	MenuDao객체에 저장된 메뉴의 텍스트를 반환한다.
Others	jsp로 구현한다.	

3.3.3) food_categories_view

food_categories_view

+file.list() : String[] +request.getRealPath("문장“) : String +request.getParameter("문장“) : String +RestaurantDao.lists() : ArrayList<RestaurantDto> +RestaurantDto.getRestaurantName() : String +RestaurantDto.getCategory() : String +RestaurantDto.getBusinessNumber() : String

Class:food_categories_view		
Attributes		
Methods	+file.list() : String[]	사진파일안에 있는 파일이름들을 리스트로 가져온다.
	+request.getRealPath("문장“) : String	문장에 해당하는 파일의 절대경로를 얻어온다.
	+request.getParameter("문장“) : String	문장에 해당하는 request의 값을 반환한다.
	+RestaurantDao.lists() : ArrayList<RestaurantDto>	restaurant테이블의 모든레코드를 가져온다.
	+RestaurantDto.getRestaurantName() : String	dto에 저장되어있는 식당의 이름을 반환한다.
	+RestaurantDto.getCategory() : String	dto에 저장되어 있는 카테고리를 반환한다.
	+RestaurantDto.getBusinessNumber() : String	dto에 저장되어 있는 사업자번호를 반환한다.
Others	jsp로 구현한다.	

3.3.4) index

index
+session.getAttribute("문장") : Object +MemberDao.getInstance() : MemberDao

Class:index		
Attributes		
Methods	+session.getAttribute("문장") : Object	문장에 해당하는 session의 값을 반환한다.
	+MemberDao.getInstance() :	MemberDao객체를 반환한

	MemberDao	다.
Others	jsp로 구현한다.	

3.3.5) menu_write

menu_write
+session.getAttribute("문장") : Object

Class:menu_write		
Attributes		
Methods	+session.getAttribute("문장") : Object	문장에 해당하는 session의 값을 반환한다.
Others	jsp로 구현한다.	

3.3.6) my_restaurant

my_restaurant
+session.getAttribute("문장") : Object +request.getParameter("문장") : String +RestaurantDao.getBusinessNumMember(businessNumber : String) : RestaurantDto +MemberDao.getInstance() : MemberDao +RestaurantDto.getRestaurantName() : String +request.getRealPath("문장") : String +file.list() : String[] +restaurantDao.lists() : ArrayList<RestaurantDto> +MenuDao.getInstance() : MenuDao +MenuDao.lists() : ArrayList<MenuDto> +MenuDto.getMenuNumber() : String +MenuDto.getMenuName() : String +MenuDto.getMenuPrice() : String +MenuDto.getMenuText() : String +GoogleMapDao.getInstance() : GoogleMapDao +GoogleMapDao.getBusinessNumGoogle(businessNumber : String) : GoogleMapDto +googleDto.getGoogledata() : String

+ReviewDao.getInstance() : ReviewDao +reviewDao.list(businessNumber : String) : ArrayList<ReviewDto> +reviewDao.getReviewCount(businessNumber : String) : int +reviewDto.getStarCount() : String +reviewDto.getUserName() : String +reviewDto.getReviewText() : String +reviewDto.getUserId : String

Class:my_restaurant		
Attributes		
Methods	+session.getAttribute("문장") : Object	문장에 해당하는 session의 값을 반환한다.
	+request.getParameter("문장") : String	문장에 해당하는 request의 값을 반환한다.
	+RestaurantDao.getBusinessNumMember(businessNumber : String) : RestaurantDto	인수로 준 businessNumber와 같은 레코드들을 데이터베이스에서 가져온다.
	+MemberDao.getInstance() : MemberDao	MemberDao의 객체를 반환한다.
	+RestaurantDto.getRestaurantName() : String	식당이름을 반환한다.
	+request.getRealPath("문장") : String	문장에 해당하는 파일의 절대경로를 얻어온다.
	+file.list() : String[]	사진파일안에 있는 파일이름들을 리스트로 가져온다.
	+restaurantDao.lists() : ArrayList<RestaurantDto>	restaurant테이블의 모든레코드를 가져온다.
	+MenuDao.getInstance() : MenuDao	MenuDao의 객체를 반환한다.
	+MenuDao.lists() : ArrayList<MenuDto>	menu테이블의 모든레코드를 가져온다.
	+MenuDto.getMenuNumber() : String	메뉴번호를 반환한다.
	+MenuDto.getMenuName() : String	메뉴이름을 반환한다.
	+MenuDto.getMenuPrice() : String	메뉴 가격을 반환한다.
	+MenuDto.getMenuText() : String	메뉴의 설명을 반환한다.
	+GoogleMapDao.getInstance() : GoogleMapDao	GoogleMapDao의 객체를 반환한다.
	+GoogleMapDao.getBusinessNumGoogle(businessNumber : String) :	인수로 준 businessNumber와 같은 레코드들을 데이터베이스에서 가져온다.

	GoogleMapDto	
	+googleDto.getGoogledata() : String	구글지도의 데이터를 가져온다.
	+ReviewDao.getInstance() : ReviewDao	ReviewDao의 객체를 반환한다.
	+reviewDao.list(businessNumber : String) : ArrayList<ReviewDto>	인수로 준 businessNumber와 같은 레코드들을 데이터베이스에서 가져온다.
	+reviewDao.getReviewCount(businessNumber : String) : int	인수로 준 businessNumber와 같은 레코드의 수를 반환한다.
	+reviewDto.getStarCount() : String	식당의 별점을 반환한다.
	+reviewDto.getUserName() : String	유저의 이름을 반환한다.
	+reviewDto.getReviewText() : String	유저가 등록한 리뷰의 텍스트를 반환한다.
	+reviewDto.getUserId : String	유저의 아이디를 반환한다.
Others	jsp로 구현한다.	

3.3.7) myMenu, restaurant-single

myMenu
+session.getAttribute("문장") : Object +MemberDao.getInstance() : MemberDao +request.getRealPath("문장") : String +file.list() : String[] +MenuDao.getInstance() : MenuDao +MenuDao.lists() : ArrayList<MenuDto> +MenuDto.getMenuName() : String +MenuDto.getMenuPrice() : String +MenuDto.getMenuText() : String +MenuDto.getBusinessNumber() : String +MenuDto.getMenuNumber() : String

Class:myMenu		
Attributes		
Methods	+session.getAttribute("문장") : Object	문장에 해당하는 session의 값을 반환한다.
	+MemberDao.getInstance() : MemberDao	MemberDao 객체를 반환한다.
	+request.getRealPath("문장") : String	문장에 해당하는 파일의 절대경로를 얻어온다.

	+file.list() : String[]	사진파일안에 있는 파일이름들을 리스트로 가져온다.
	+MenuDao.getInstance() : MenuDao	MenuDao의 객체를 반환한다.
	+MenuDao.lists() : ArrayList<MenuDto>	menu테이블의 모든 레코드를 가져온다.
	+MenuDto.getMenuName() : String	메뉴의 이름을 반환한다.
	+MenuDto.getMenuPrice() : String	메뉴의 가격을 반환한다.
	+MenuDto.getMenuText() : String	메뉴의 설명을 가져온다.
	+MenuDto.getBusinessNumber() : String	사업자번호를 가져온다.
	+MenuDto.getMenuNumber() : String	메뉴의 번호를 가져온다.
Others	jsp로 구현한다.	

3.3.8) reservation_view, reservation_user(점주, 고객)

reservation_view, reservation_user
+session.getAttribute("문장") : Object +MemberDao.getInstance() : MemberDao +ReservationDao.getInstance() : ReservationDao +ReservationDao.lists() : ArrayList<ReservationDto> +ReservationDto.getUserId() : String +ReservationDto.getUserMail() : String +ReservationDto.getUserName() : String +ReservationDto.getUserPhone() : String +ReservationDto.getUserTime() : String +ReservationDto.getUserSit() : String +ReservationDto.getBusinessNumber() : String +CartDao.getInstance() : CartDao +CartDao.lists() : ArrayList<CartDto> +CartDto.getMenuName() : String

Class:reservation_view, reservation_user(점주, 고객)		
Attributes		
Methods	+session.getAttribute("문장") : Object	문장에 해당하는 session의 값을 반환한다.
	+MemberDao.getInstance() : MemberDao	MemberDao 객체를 반환한다.
	+ReservationDao.getInstan	ReservationDao객체를 반환

	ce() : ReservationDao	한다.
	+ReservationDao.lists() : ArrayList<ReservationDto>	Reservation테이블의 모든 레코드를 가져온다.
	+ReservationDto.getUserId() : String	유저의 아이디를 가져온다.
	+ReservationDto.getUserMail() : String	유저의 메일주소를 반환한다.
	+ReservationDto.getUserName() : String	유저의 이름을 반환한다.
	+ReservationDto.getUserPhone() : String	유저의 전화번호를 반환한다.
	+ReservationDto.getUserTime() : String	유저가 식당을 예약한 시간을 반환한다.
	+ReservationDto.getUserSit() : String	유저가 예약한 식당의 자리수를 반환한다.
	+ReservationDto.getBusinessNumber() : String	사업자번호를 반환한다.
	+CartDao.getInstance() : CartDao	CartDao의 객체를 반환한다.
	+CartDao.lists() : ArrayList<CartDto>	cart테이블의 모든 레코드를 가져온다.
	+CartDto.getMenuName() : String	메뉴의 이름을 반환한다.
Others	jsp로 구현한다.	

3.3.9)restaurants, search_view_restaurant

restaurants
+file.list() : String[] +request.getRealPath("문장“) : String +request.getParameter("문장“) : String +RestaurantDao.lists() : ArrayList<RestaurantDto> +RestaurantDto.getRestaurantName() : String +RestaurantDto.getCategory() : String +RestaurantDto.getBusinessNumber() : String

Class:restaurants		
Attributes		
Methods	+file.list() : String[]	사진파일안에 있는 파일이름들을 리스트로 가져온다.
	+request.getRealPath("문장“) : String	문장에 해당하는 파일의 절대경로를 얻어온다.
	+request.getParameter("문	문장에 해당하는 request의

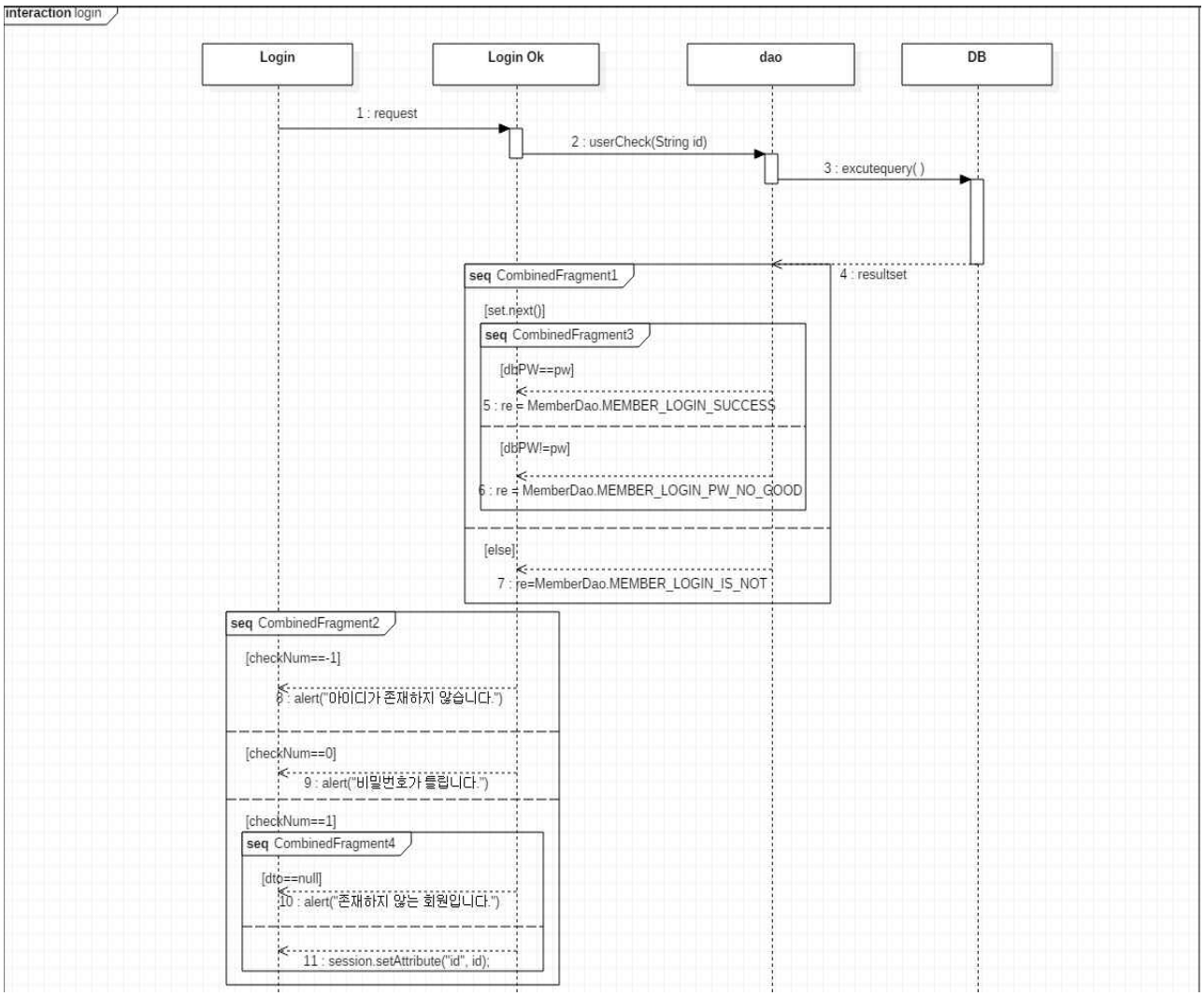
	장“) : String	값을 반환한다.
	+RestaurantDao.lists() : ArrayList<RestaurantDto>	restaurant테이블의 모든레코드를 가져온다.
	+RestaurantDto.getRestaurantName() : String	dto에 저장되어있는 식당의 이름을 반환한다.
	+RestaurantDto.getCategory() : String	dto에 저장되어 있는 카테고리를 반환한다.
	+RestaurantDto.getBusinessNumber() : String	dto에 저장되어 있는 사업자번호를 반환한다.
Others	jsp로 구현한다.	

4. Sequence diagram

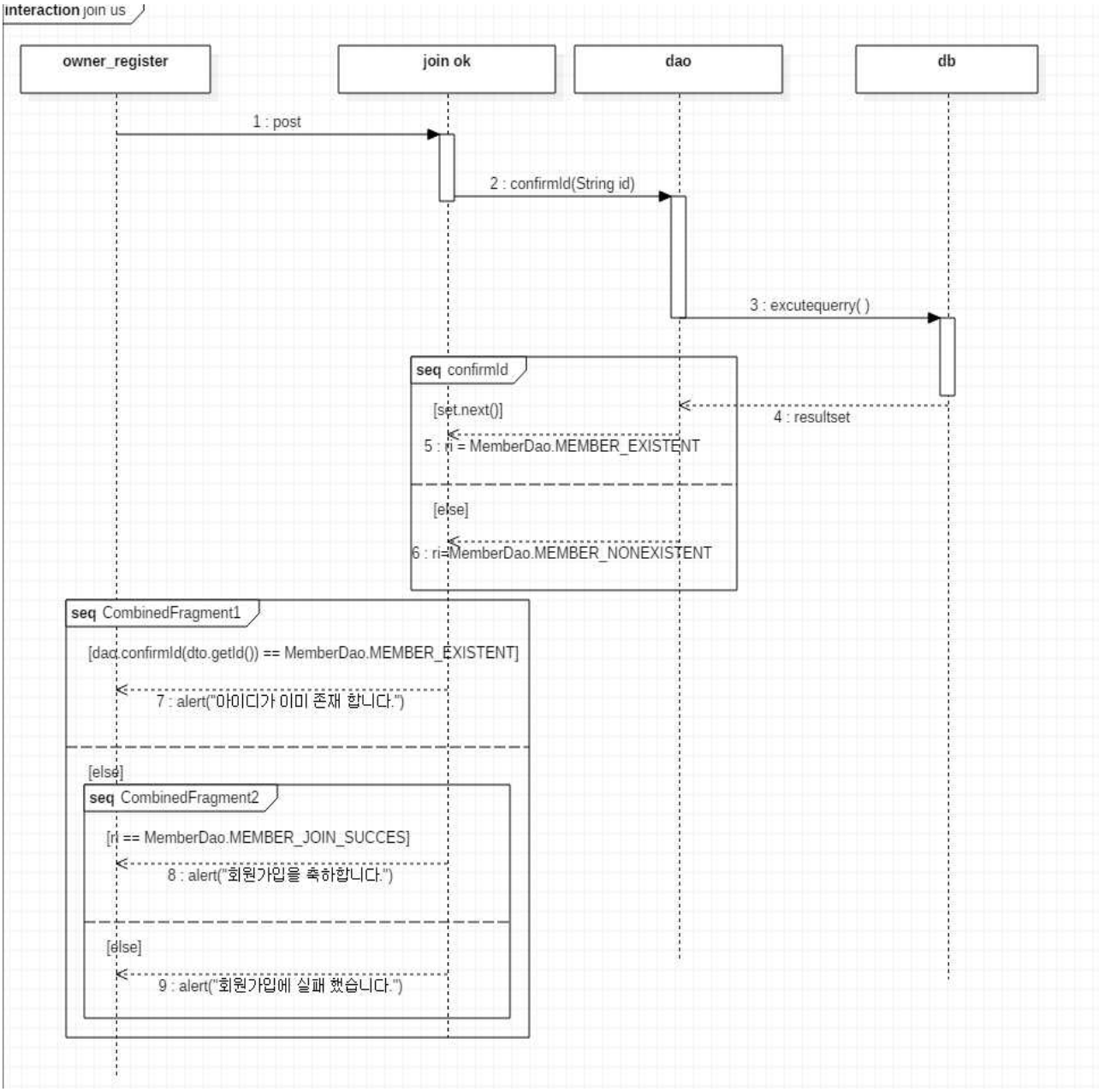
- Draw sequence diagrams for the whole functions of your system. - Explain each sequence diagram. - 12pt, 160%.

※ 앞선 다이어 그램과 유사한 로직이거나 프로그램 특성 상 UI적 요소가 많아 시퀀스 다이어그램의 의미가 적은 다이어그램은 생략하였습니다.

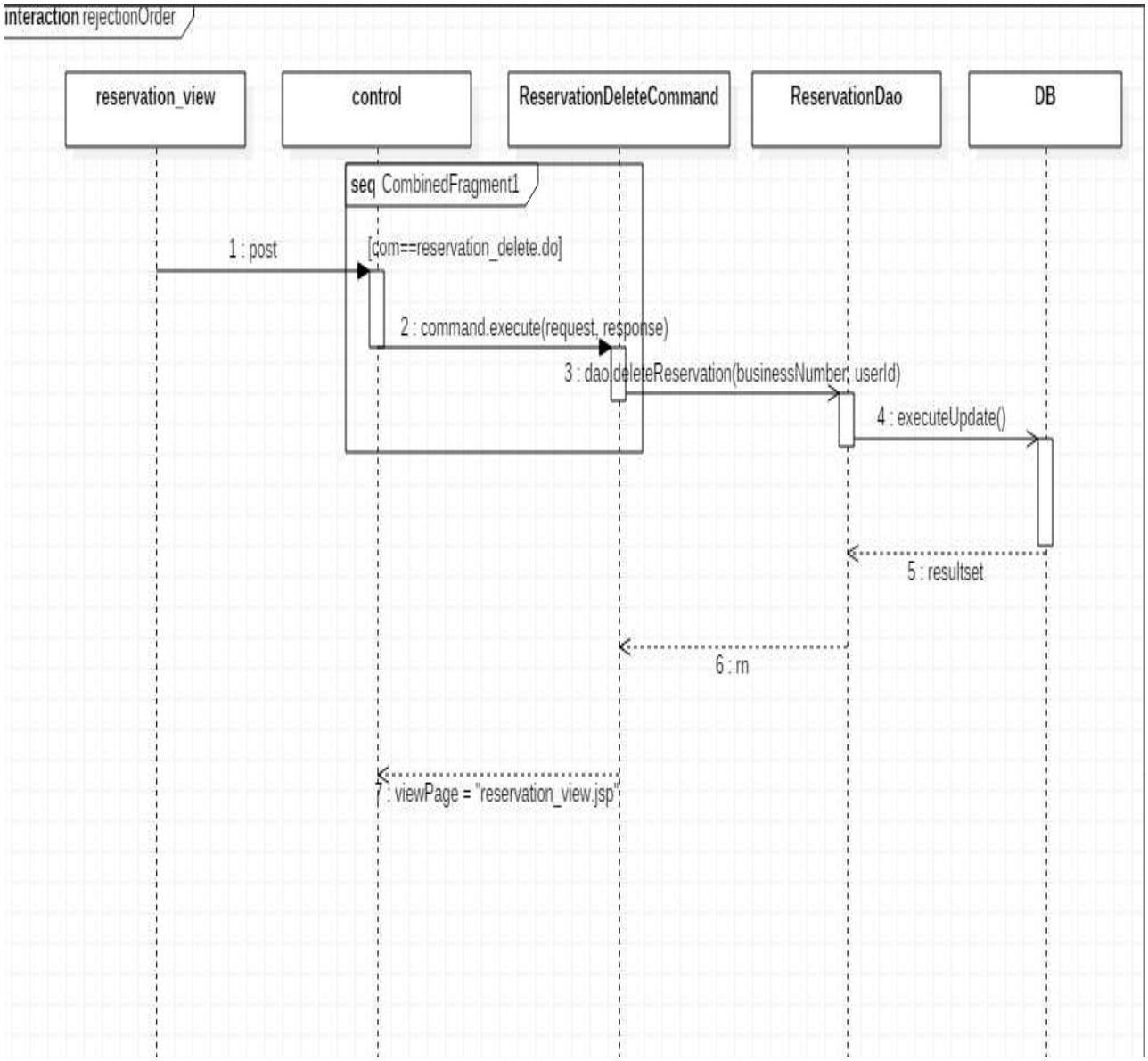
#1. Log-in



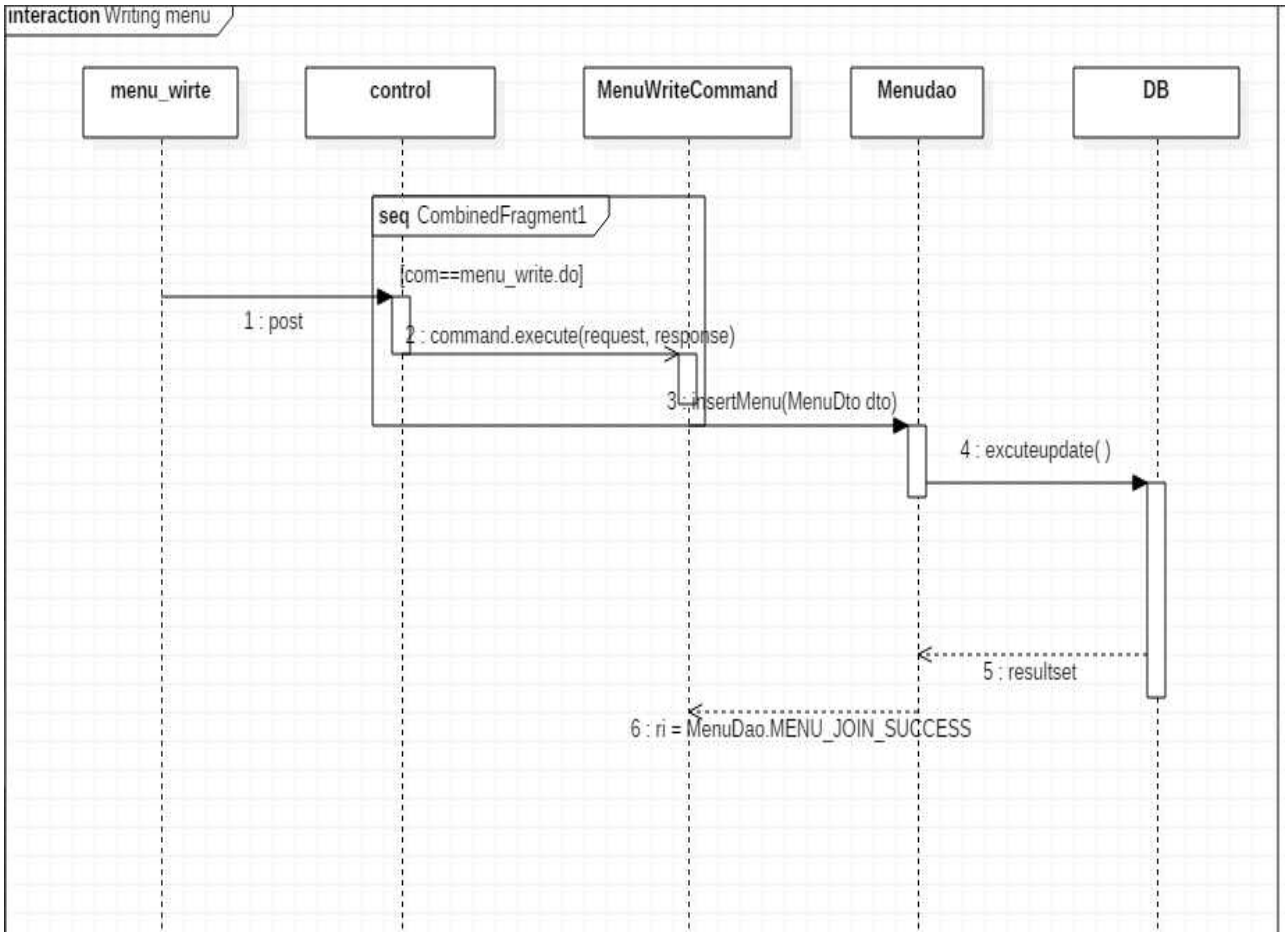
#2. Member Joining



#3. Rejection order

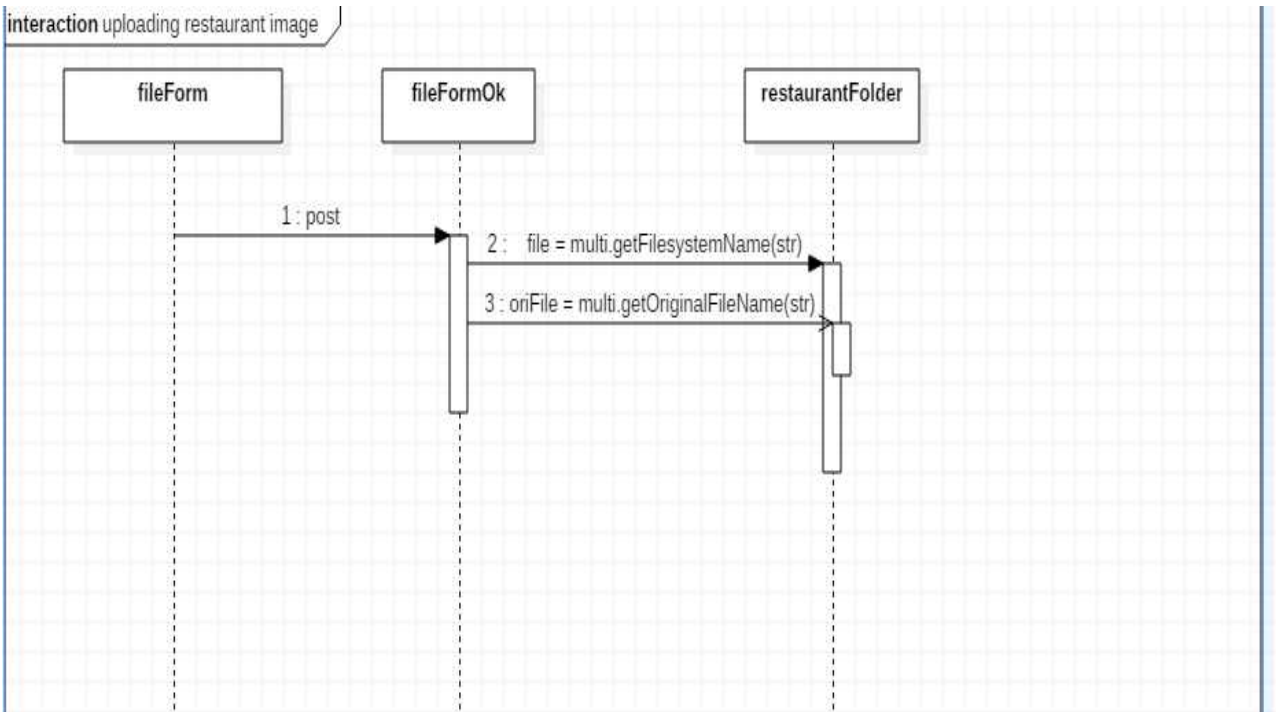


#4. Writing menu



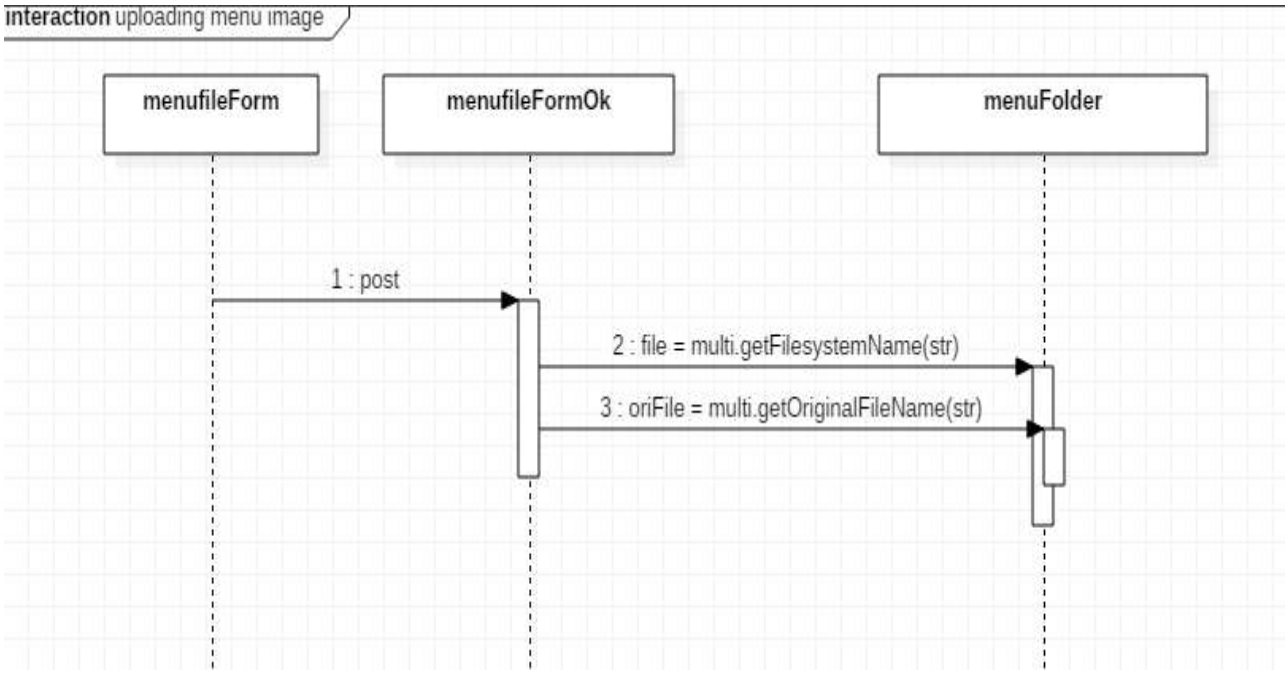
#5. Uploading restaurant image

```
MultipartRequest multi = new MultipartRequest(request, path, size, "utf-8", new DefaultFileRenamePolicy());

Enumeration files = multi.getFileNames();
String str = (String)files.nextElement();

file = multi.getFilesystemName(str);
oriFile = multi.getOriginalFileName(str);
```

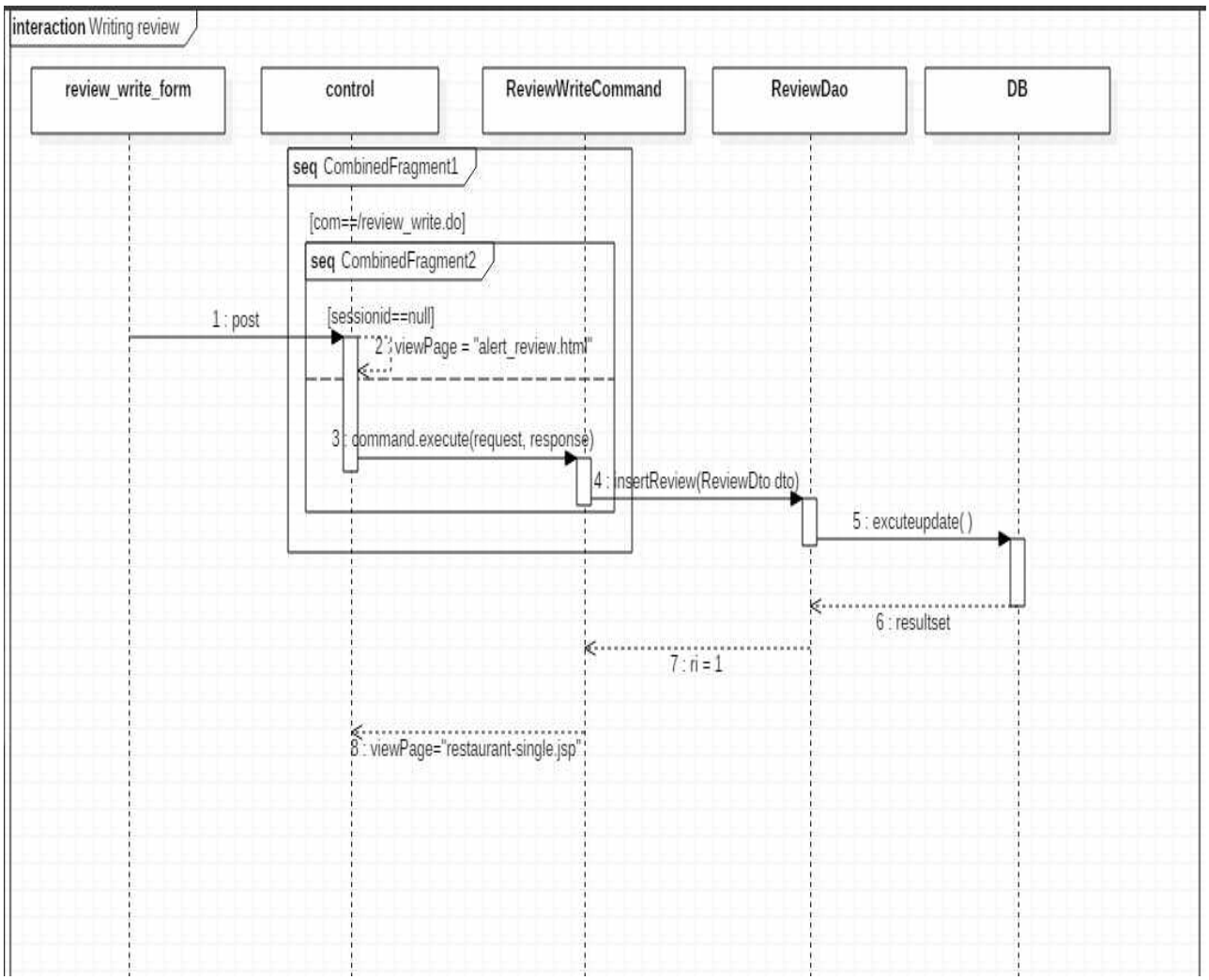


#6. Uploading menu image

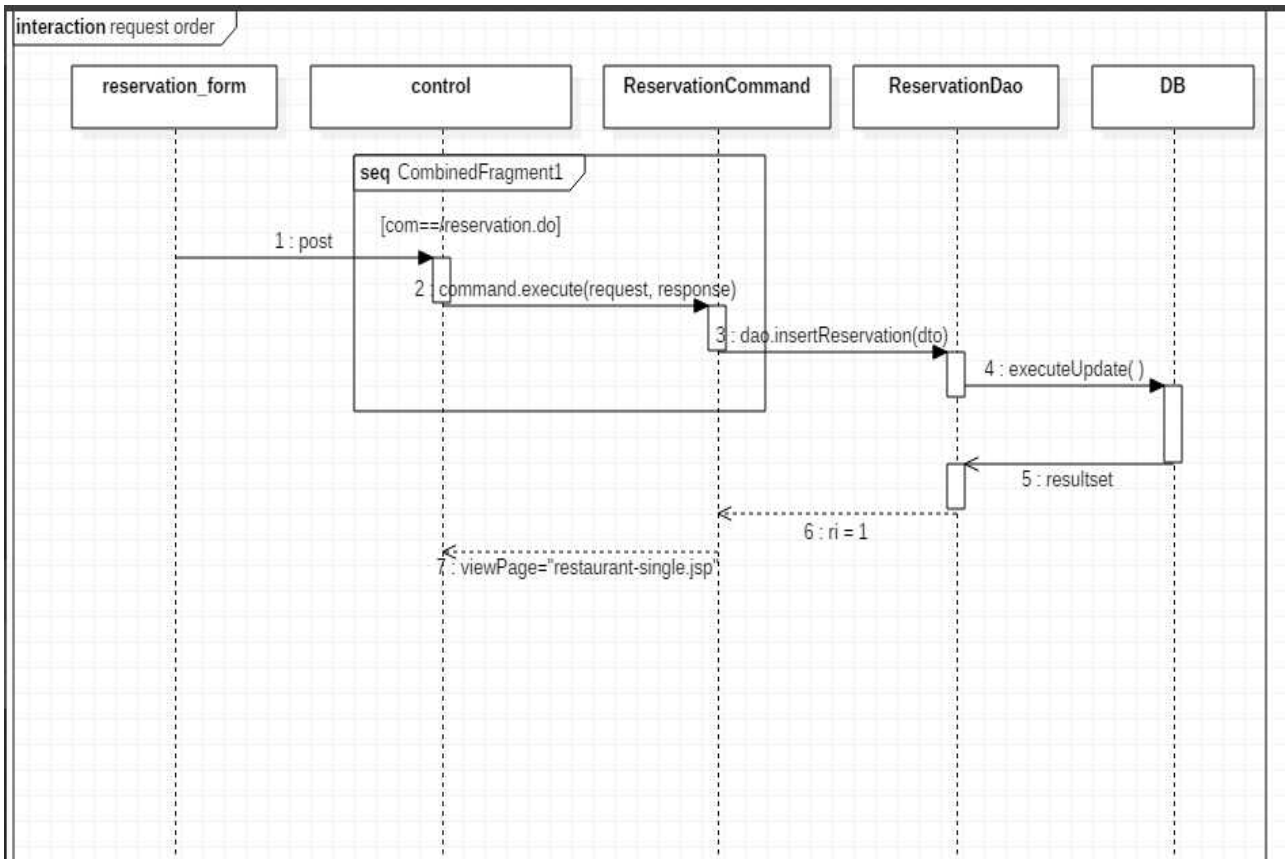
```
MultipartRequest multi = new MultipartRequest(request, path, size, "utf-8", new DefaultFileRenamePolicy());

Enumeration files = multi.getFileNames();
String str = (String)files.nextElement();

file = multi.getFilesystemName(str);
oriFile = multi.getOriginalFileName(str);
```



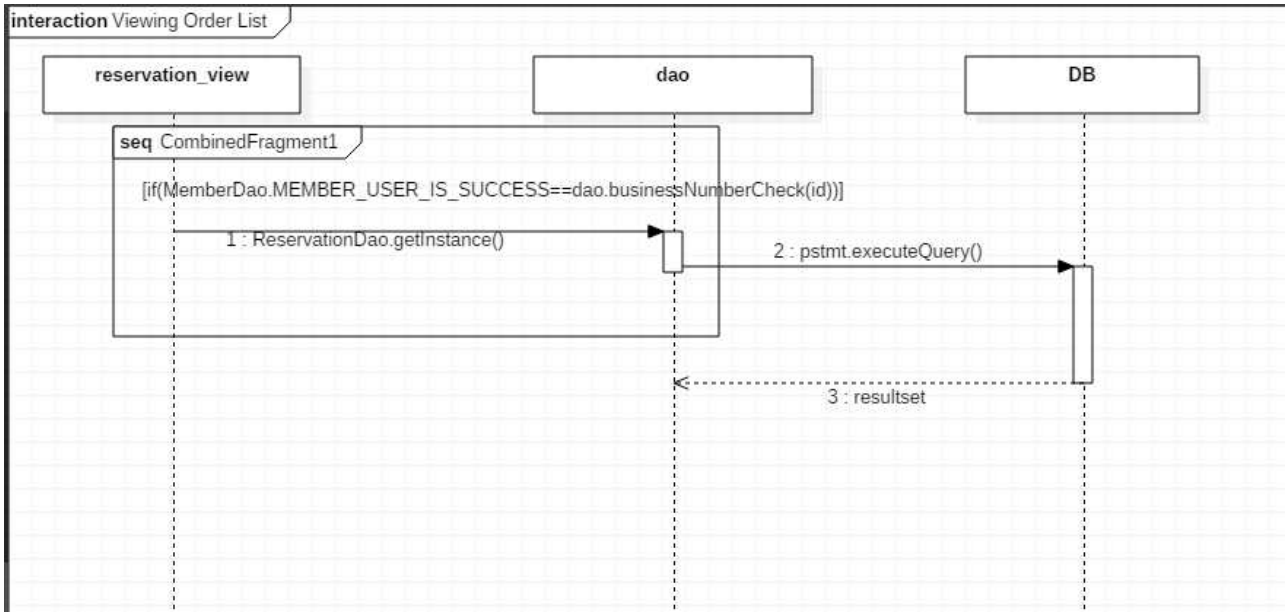
#8. Writing review



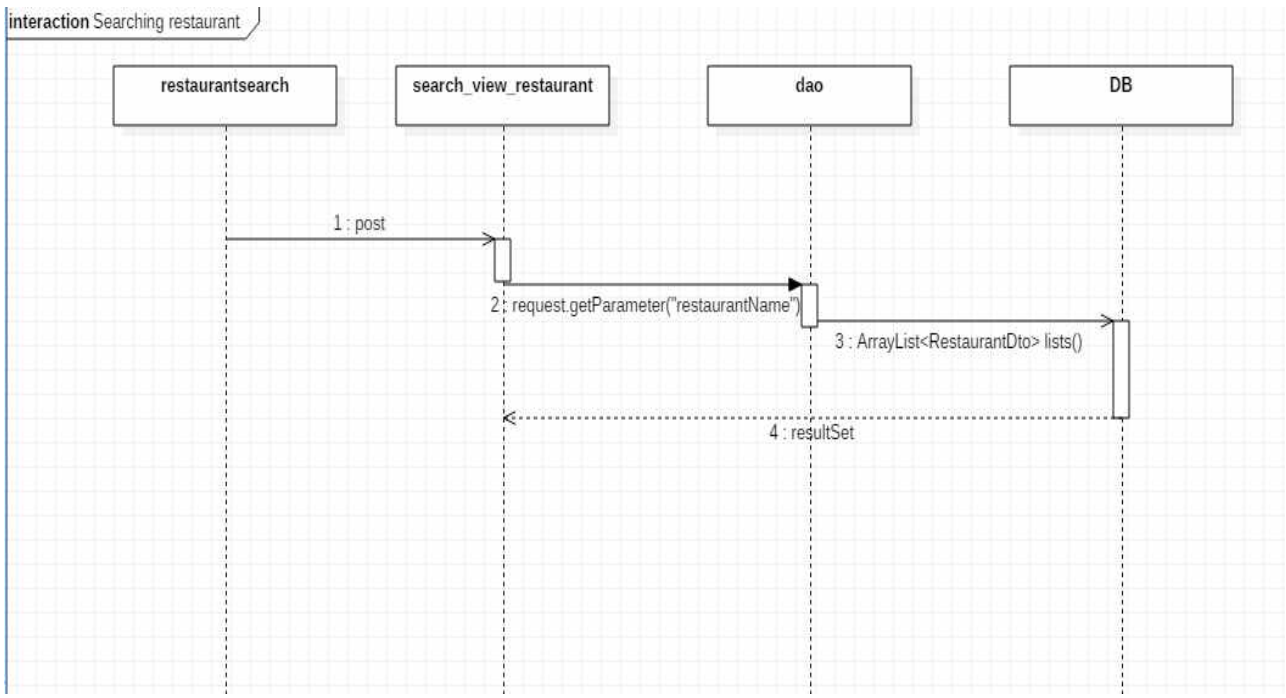
#9. Request order



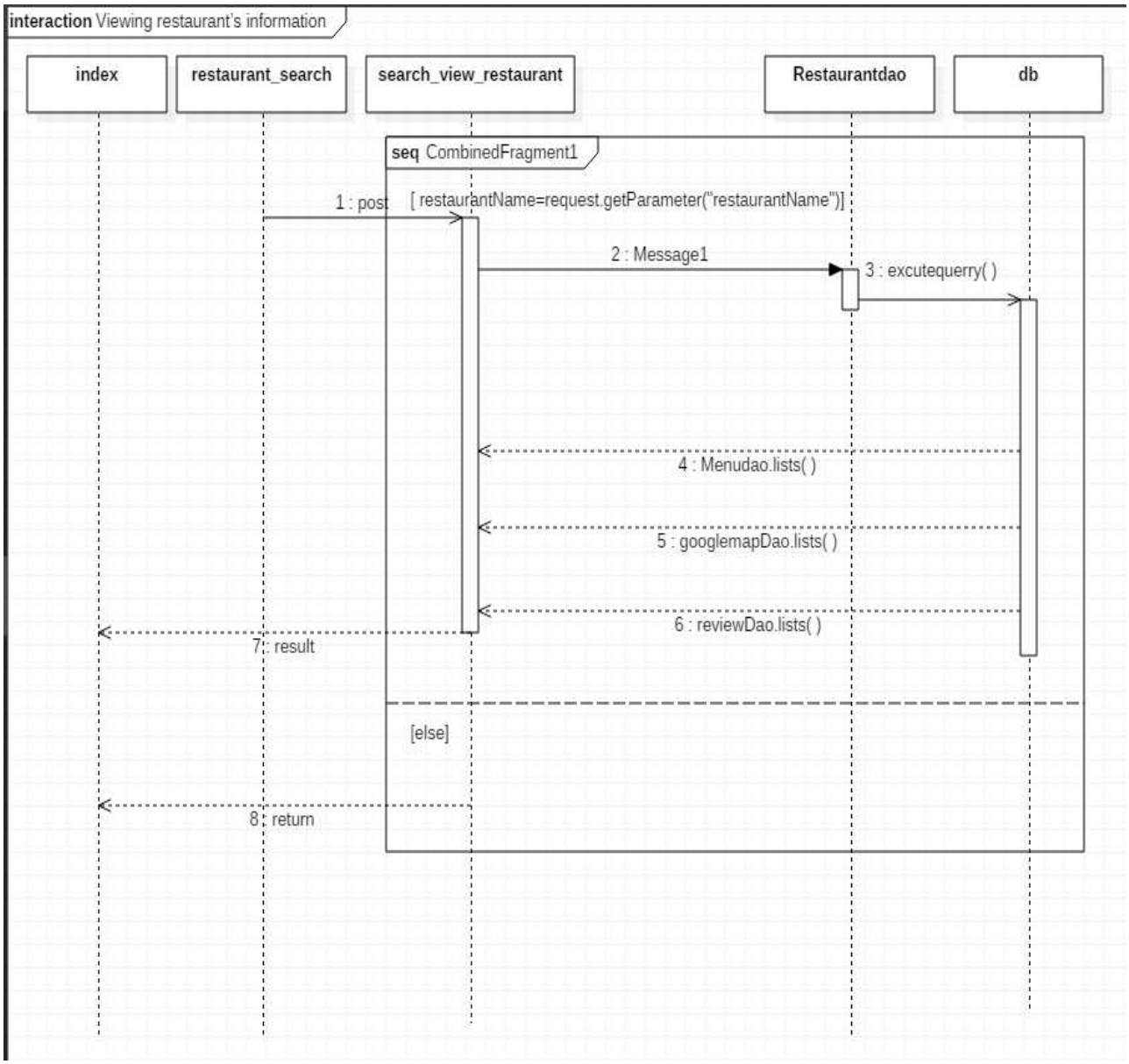
#10. Watching order list



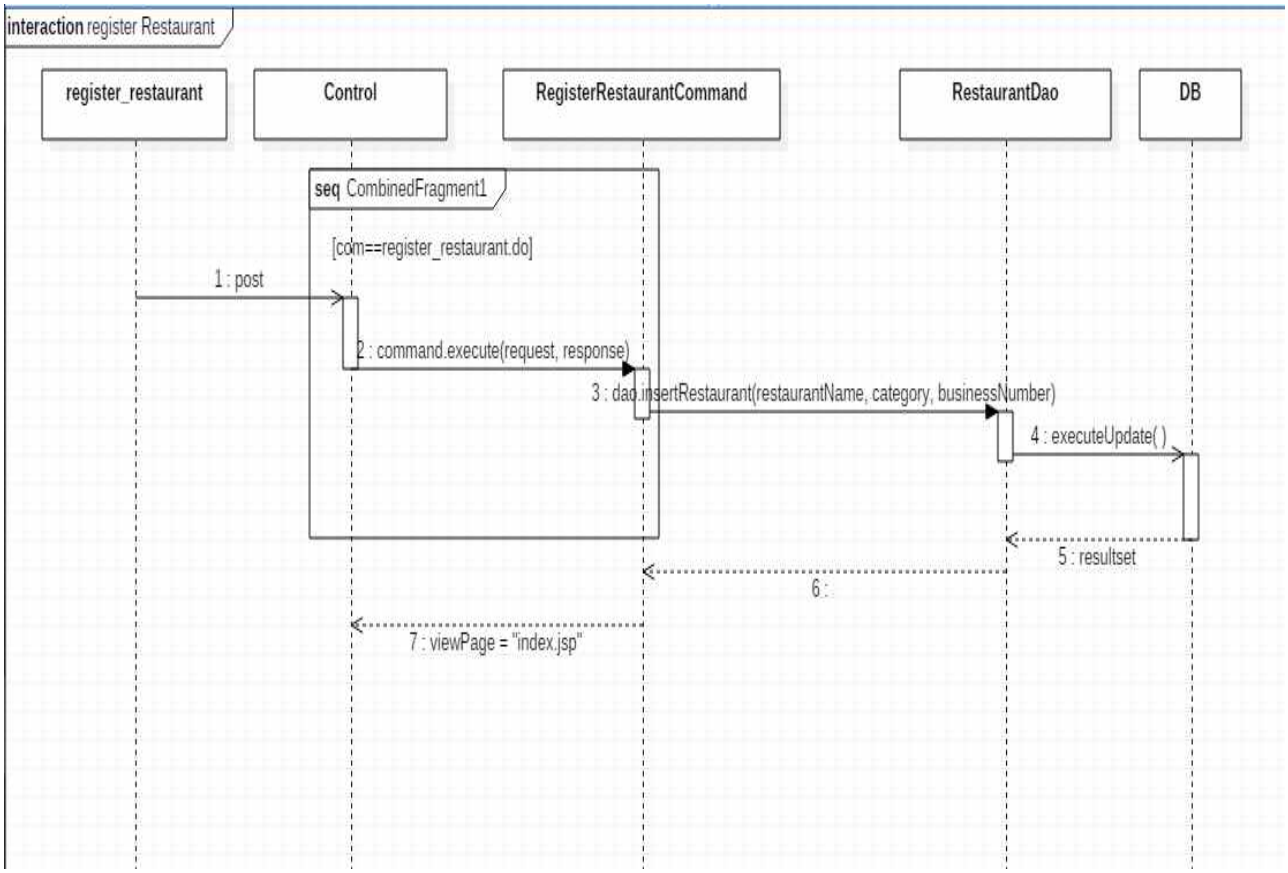
#11. Searching restaurant



#12. Viewing restaurant's Information

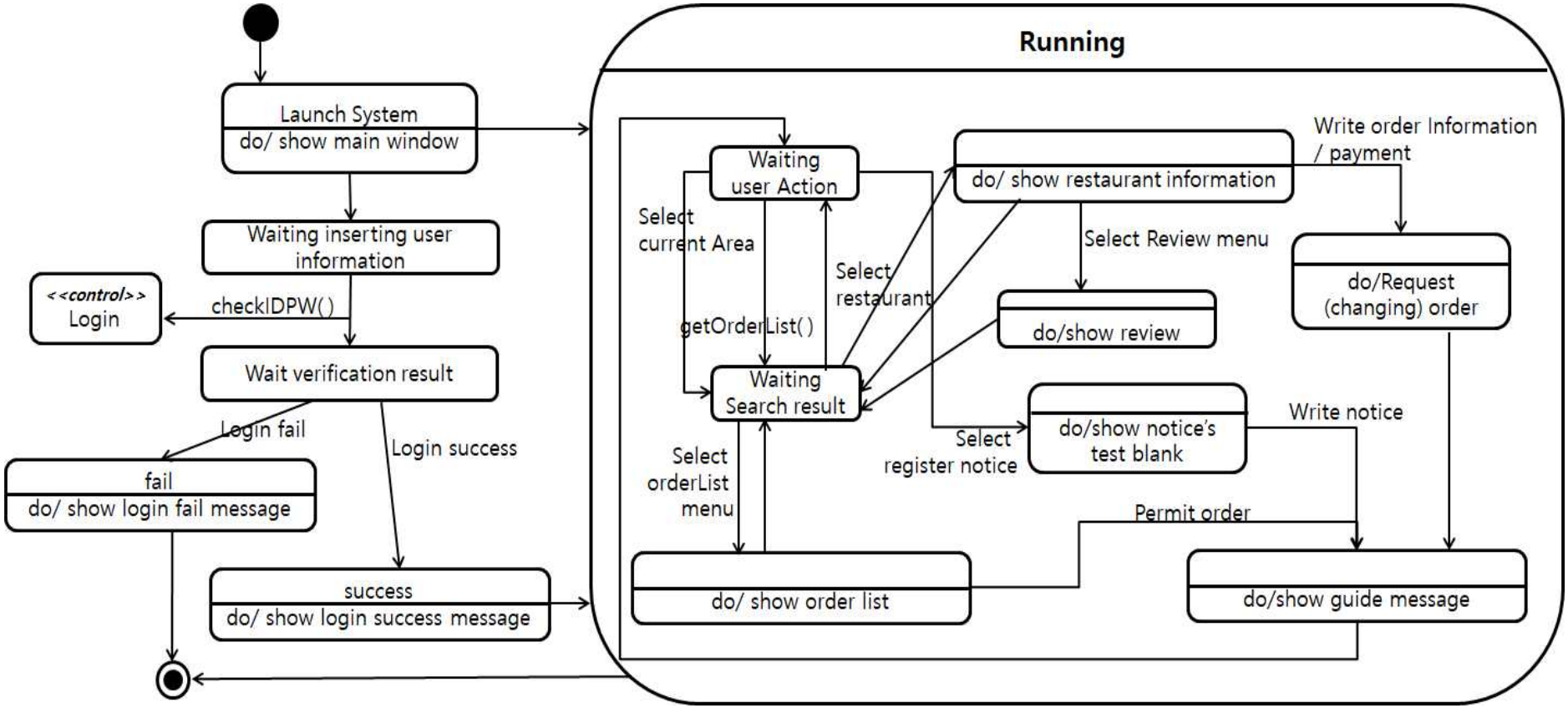


#13. register retaurant



5. State machine diagram

- Draw state machine diagrams for the client and the server system.
- Explain each state machine diagram.
- 12pt, 160%.



6. Implementation requirements

- Describe operating environments to implement your system.
- 12pt, 160%.

PC	
컴퓨터이름	DESKTOP-0036G3B
운영체제	Windows 10 HOME
프로세서	Intel(R) Core(TM) M-5Y10C CPU @ 0,80GHz (4 CPUs), ~ 1.0GHz
메모리	8192MB RAM
하드디스크	109GB

모바일	
제품명	GALAXY-A7 2017
운영체제	안드로이드 7.0
프로세서	1.9GHz 옥타코어
메모리	3GB RAM
사이즈	76.2 X 151 X 6.3mm
하드디스크	32GB

모바일	
제품명	GALAXY-S7
운영체제	안드로이드 7.0
프로세서	2.3Ghz + 1.6Ghz 옥타코어
메모리	4GB RAM
사이즈	142.4 X 69.6 X 7.9mm
하드디스크	32GB

7. Glossary

- Specifically describe all of the terms used in this documents.
- 12pt, 160%.

※ 일반적으로 사용하는 단어와 다른 의미를 가지거나 의미 파악이 어려운 단어만 위주로 작성하였습니다.

Term	Description
예약을 민족	식당 예약 및 식당에 대한 의견을 공유하는 시스템.
고객	식당을 예약하고자 하는 시스템 사용자.
식당	고객으로부터 예약을 받아 음식을 요리하는 업체.
관리자	Customer, Owner, 전체적인 시스템을 관리하는 관리자.
주문	고객이 선택한 음식점에 메뉴와 테이블을 선택하여 예약하는 행위.
Order	주문 참조.
Customer	고객 참조.
Owner	식당 참조.
menu	식당에서 제공하는 음식
table	식당에 고객이 앉을 수 있는 공간.
review	고객이 식당을 이용하고 식당에 대한 의견을 등록하는 공간.
application's notice	앱의 공지사항을 볼 수 있는 공간.
restaurant's notice	식당의 공지사항을 볼 수 있는 공간.
PermitOrder	주문을 승인하거나 주문을 취소하는 행위. 주문 참조.
orderlist	고객들이 식당에 요청한 예약 리스트.
restaurant's information	식당의 기본정보 메뉴종류와 가격, 현재 예약현황 등의 정보.
DB(Database)	정보를 저장하고 관리하는 저장소.
SERVERPORT	PC의 포트번호.
SERVERIP	PC의 아이피.
EnvironmentSetting	앱의 기본적인 환경, 정보.
MemberReport	주문자의 정보.

8. References

- Describe all of your references (book, paper, technical report etc). - 12pt, 160%.

-[SE] SDS_analysis example 2(FOOD PORTAL SYSTEM)

-[SE] SDS_design example 2(FOOD PORTAL SYSTEM)

-R.Elmasri and S.Navathe, Fundamentals of Database Systems (7th Edition)