**강아지 프로젝트 제안서**

**대상**

본 시스템은 반려견을 키우고 있는 모든 사람들을 대상으로 한다.

**사업전략**

사용자들의 공간대여나 예약에는 제한이 없으며, 단순 조회 및 예약뿐만 아니라

특정 공간에 대한 후기열람 및 작성이 가능하여 공간에 대한 풍부한 정보를 제공한다.

또한 공간을 대여하는 회원에 한하여 결제금액의 10%를 마일리지로 적립하여 고객 유치가 용이하다.

**제안의 특징 및 장점**

화면 구성을 간편하게 구성하여 전 연령층 누구나 쉽게 이용 할 수 있다.

조건에 따른 공간들의 조회뿐 아니랴 예약 가능한 공간들만 보여주어 편리성을 제공한다.

또한 공간을 이용한 사용자들에 한해서 후기를 작성하기 때문에 공정성 있는 후기를 보장한다.

**기대효과**

단순 공간 조회뿐만 아니라 공간에 대한 후기열람, 작성이 가능하여

공간에 대한 정확한 정보를 제공 할 수 있다.

**향후 시스템발전 방향**

본 공간대여 시스템을 확장하여 공간뿐만 아니라 특정시설(실버타운, 어린이집, 문화센터)등에 대한 조회도 가능하게 하여 사용자층을 보다 더 넓게 확장 할 수 있다.

**1. 시스템 개요**

**1.1 목적**

본 서비스의 목적은 강아지가 동반 가능한 모든 장소들의 정보를 제공한다.

**1.2 개요**

- 본 서비스는 회원 전용 서비스이다.

- 전국 지도는 모든 사용자가 등록 한 장소를 조회 할 수 있다.

- 전국 지도에서 특정 장소를 선택하면, 해당 장소에 대한 모든 게시 글을 볼 수 있다.

- 전국 지도는 공공데이터를 기반으로 하여 애견동반장소를 제공한다. 또한 사용자들은 누락된 장소를 추가적으로 등록 할 수 있다.

- 사용자는 특정장소에 대하여 게시 글을 작성 할 수 있다.

- 사용자는 특정장소에 대하여 스크랩 할 수 있다.

- 자신이 등록한 모든 장소들을 조회 할 수 있다.

- 작성자를 조회하면 해당 작성자의 모든 게시 글을 조회 할 수 있다.

- 사용자는 부적절한 게시 글을 신고 할 수 있다.

- 사용자는 등록된 장소에 대하여 수정 및 삭제를 요청 할 수 있다.

- 공급자는 제공한 공간에 한해 예약내역을 조회할 수 있다.

- 수요자가 공간 조회 시, 서비스는 현재시간과 날짜를 기준으로 자정까지의

빈 공간들을 보여준다. 이 때 본 서비스의 회원일 경우 가입 정보를

추가 참조 하여 선호 지역의 특정 유형의 공간 정보를 제공한다.

- 검색 조건에는 지역과 시간, 공간유형 조건이 있다..

- 사용자는 수요자 또는 공급자로 구분하여 회원가입을 할 수 있다.

회원가입을 한 사용자는 개인정보를 수정하거나 탈퇴할 수 있다.

- 본 서비스는 사용자가 회원가입 시 공간을 대여하고자 하는 수요자와

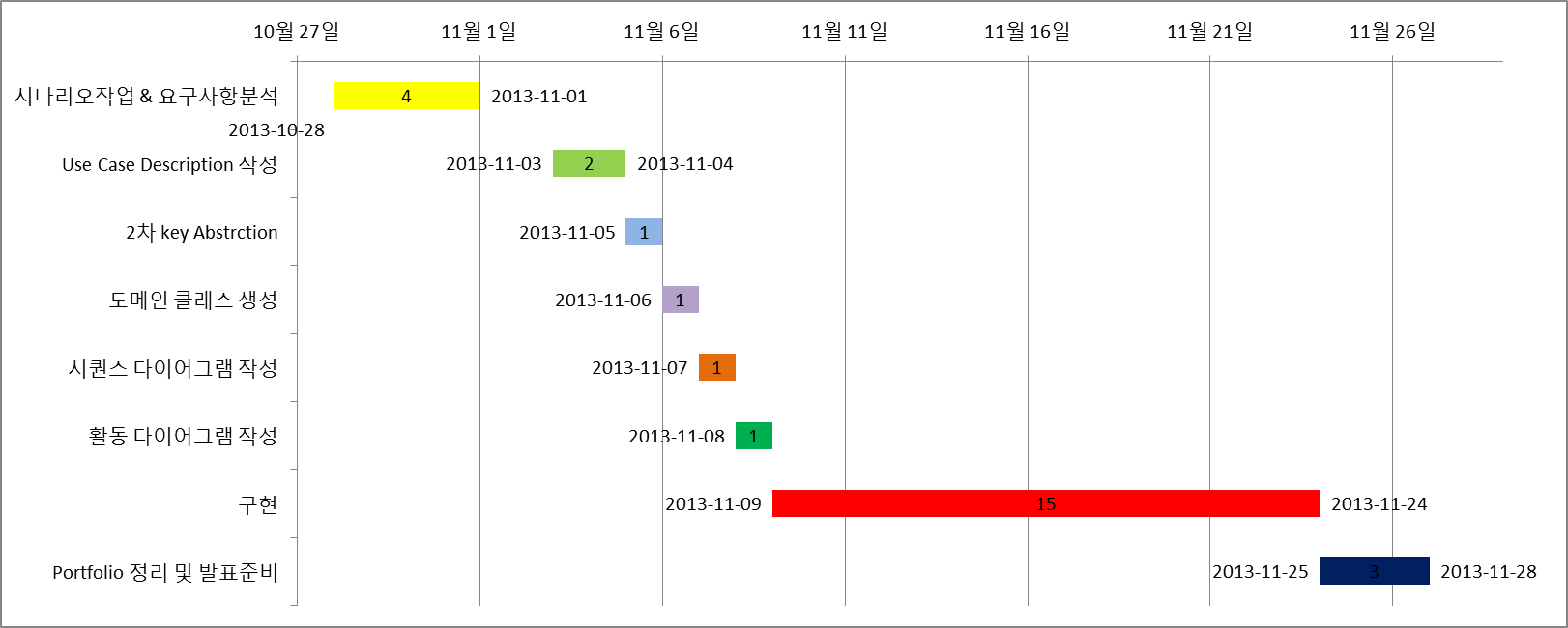
공간을 제공하는 공급자로 구분한다.

- 마일리지를 예약 결제 시 사용한다면, 적립이 되지 않는다.

**Team Info**

|  |  |  |
| --- | --- | --- |
| 팀원소개및역할 | | |
| 팀장 | **김성미** | **NoBackSpace Project 총괄, 협력업체관리 Part** |
| 팀원 | **강서윤** | **매칭 Part, 웹 디자인 Part** |
| **김재국** | **상담 Part, 회원관리 Part** |
| **서지민** | **개인정보관리 Part** |
| **이 훈** | **공지게시판, 자유게시판, 추천여행지 Part** |

**Project Plan**



**개발환경**

-OS : Windows XP 32bit, Windows 7 32bit, Window8 64bit

-Oracle DB 11g Express, JDK 1.7

**개발기술**

언어 : JAVA, Oracle SQL, HTML5, CSS3 , Java Script

-Framework Servlet(2.1)/JSP(2.5)

-JQuery-1.10.2, Ajax, Tiles2.2.2

-Slf4j1.6.6, SpringFramework3.1.1

-Apache Maven 3.1.1, JSon/Xml, Mybatis3.2.3

-Mybatis-spring 1.1.1, Ojdbc14 10.2.0.4

**개발툴**

-Eclipse IDE for Java Developers 1.4.2

-Oracle SQL Developer 3.2.20.09

-Enterprise Architect 9.0

-ER Date Modeler 7.3

-Subclipse 1.10.3

**분석설계**

-UML 2.0

**1차 Key Abstraction**

**명사** – 수요자, 공급자, 회원, 사용자, 서비스, 관리자, 공간, 시간, 후기, 공간유형, 대여일정, 지역, 이용인원, 규제조건, 환불규정, 선호지역, 특정유형, 이용대금

|  |  |  |
| --- | --- | --- |
| **명사(Actor)** | **동사(UseCase)** | |
| **수요자** | **공간을** | **조회하다** |
| **예약하다** |
| **스크랩 하다** |
| **신고하다** |
| **회원** | **가입하다** |
| **수정하다** |
| **탈퇴하다** |
| **후기** | **등록하다** |
| **수정하다** |
| **삭제하다** |
| **조회하다** |
| **신고하다** |
| **공간을** | **조회하다** |
| **취소하다** |
| **공급자** | **공간을** | **등록하다** |
| **수정하다** |
| **삭제하다** |
| **조회하다** |
| **회원을** | **가입하다** |
| **수정하다** |
| **탈퇴하다** |
| **후기를** | **조회하다** |
| **예약을** | **조회하다** |
| **관리자** | **신고 된 후기를** | **조회하다**  **삭제하다** |
| **신고 된 공간을** |
| **신고 된 회원을** | **조회한다.** |
| **탈퇴하다** |
| **예약을** | **조회하다** |

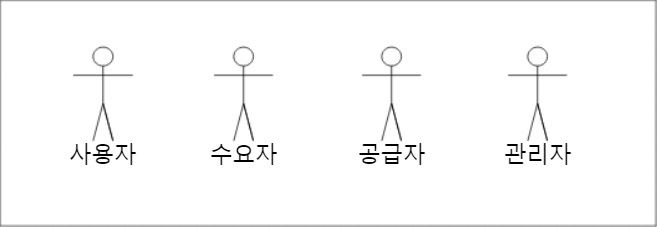
**동사** – 가입하다, 조회하다, 등록하다, 예약하다, 취소하다, 삭제하다, 스크랩하다, 신고하다,

**2. 기능적 요구사항(Functional Requirement)**

**2.1 Actor 목록**

|  |  |
| --- | --- |
| 액터 | 설명 |
| 사용자 | 사용자는 등록된 공간을 조회 할 수 있으며 다른 기능들을 이용하려면 수요자 또는 공급자로 회원 가입하여야 한다. |
| 수요자 | 수요자는 로그인을 하여 등록된 공간을 조회하고 원하는 공간을 예약, 스크랩, 신고 할 수 있으며 예약목록을 조회, 취소 할 수 있다.  또한 후기를 등록, 삭제, 수정, 조회, 신고 할 수 있다. |
| 공급자 | 공급자는 로그인을 하여 공간을 등록, 수정 및 삭제 할 수 있으며 공간에 예약된 목록을 조회할 수 있다.  또한 후기를 조회할 수 있다. |
| 관리자 | 관리자는 로그인을 하여 부적절한 후기와 허위 공간 정보에 대한 삭제, 조회 및 회원에 대한 조회, 탈퇴권한을 가진다. |

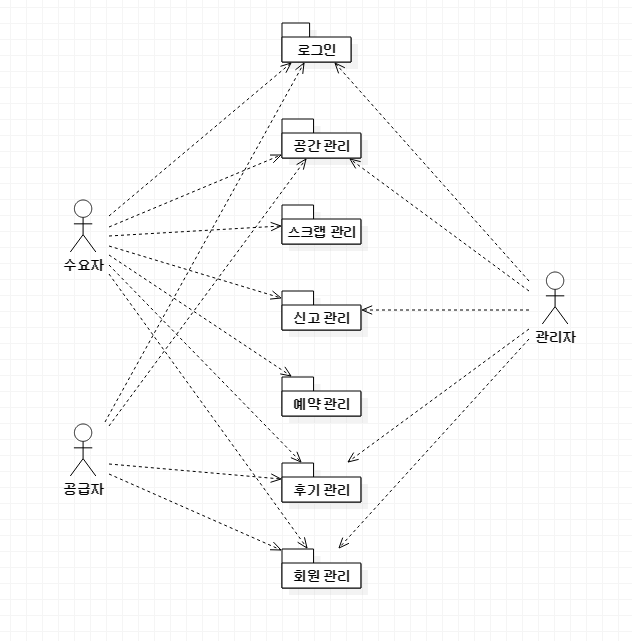
**2.2 Actor Diagram**



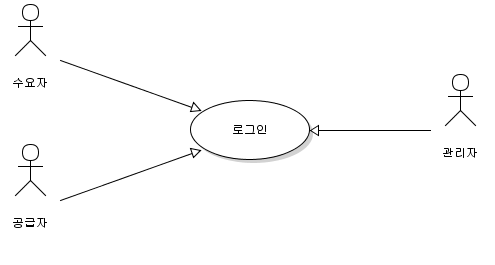
**2.3 UseCase 목록**

|  |  |  |  |
| --- | --- | --- | --- |
| ID | 이름 | 설명 | 우선순위 |
| U001 | 공간 등록 | 공급자는 공간 정보를 입력하고 공간을 등록 할 수 있다. | **중** |
| U002 | 공간 수정 | 공급자는 등록된 공간을 수정 할 수 있다. | **하** |
| U003 | 공간 삭제 | 공급자와 관리자는 공간을 삭제할 수 있다. | **하** |
| U004 | 공간 조회 | 수요자, 관리자, 공급자는 검색 조건(지역, 공간유형, 예약 일자, 예약 인원)을 입력하여 공간을 조회 할 수 있다. | **상** |
| U005 | 공간 신고 | 수요자는 공간 신고를 할 수 있다. | **중** |
| U006 | 공간 스크랩 | 수요자는 공간을 스크랩 할 수 있다. | **하** |
| U007 | 회원 가입 | 수요자, 공급자는 회원 정보를 입력하고 회원 가입을 할 수 있다. | **중** |
| U008 | 회원 수정 | 수요자, 공급자는 회원 정보를 수정 할 수 있다. | **하** |
| U009 | 회원 탈퇴 | 수요자, 공급자는 회원 탈퇴를 할 수 있다. | **하** |
| U010 | 회원 조회 | 관리자는 조건(전체조회, 메일주소 별 조회)에 따라 회원을 조회 할 수 있다. | **하** |
| U011 | 로그인 | 수요자, 공급자, 관리자는 메일과 비밀번호를 입력하여 로그인 할 수 있다. | **중** |
| U012 | 후기 등록 | 수요자는 공간에 대해 후기 등록을 할 수 있다. | **중** |
| U013 | 후기 수정 | 수요자는 등록 된 후기를 수정 할 수 있다. | **하** |
| U014 | 후기 삭제 | 수요자, 관리자는 등록 된 후기를 삭제 할 수 있다. | **하** |
| U015 | 후기 조회 | 수요자, 공급자, 관리자는 후기 조회를 할 수 있다. | **중** |
| U016 | 후기 신고 | 수요자는 후기 신고를 할 수 있다. | **하** |
| U017 | 공간 예약 | 수요자는 공간 예약을 할 수 있다. | **상** |
| U018 | 예약 취소 | 수요자는 예약된 공간을 취소 할 수 있다. | **중** |
| U019 | 예약 조회 | 수요자, 공급자, 관리자는 조건(전체조회, 메일주소 별 조회, 대여 장소 별 조회)에 따라 예약 목록을 조회 할 수 있다. | **하** |

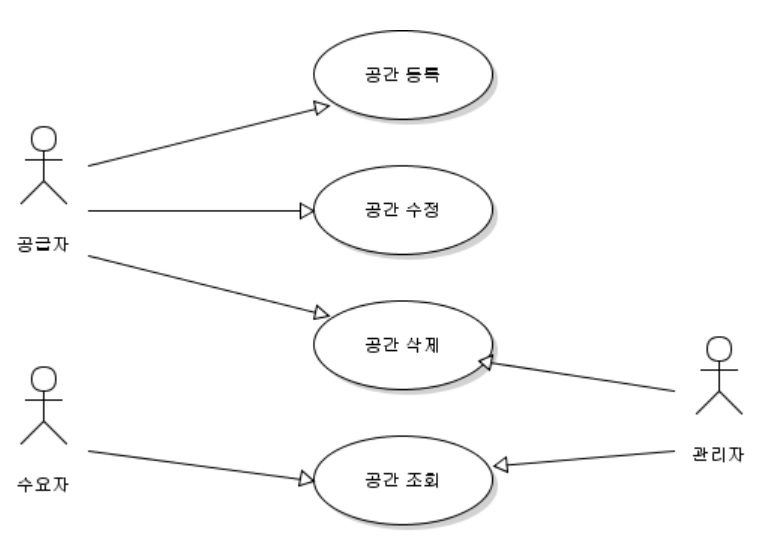
**2.4 UseCase Diagram**



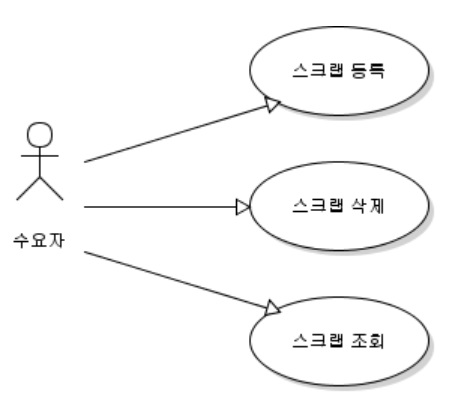
**2.4.1 로그인**



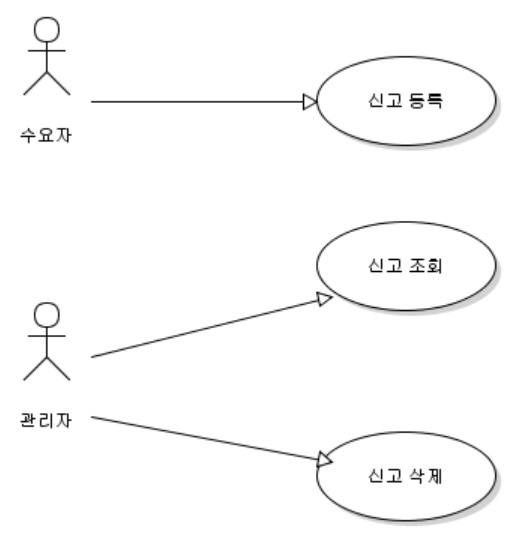
**2.4.2 공간관리**



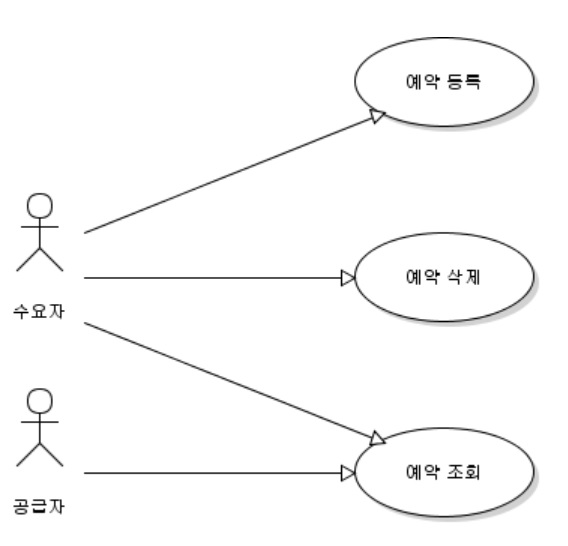
**2.4.3 스크랩 관리**

****

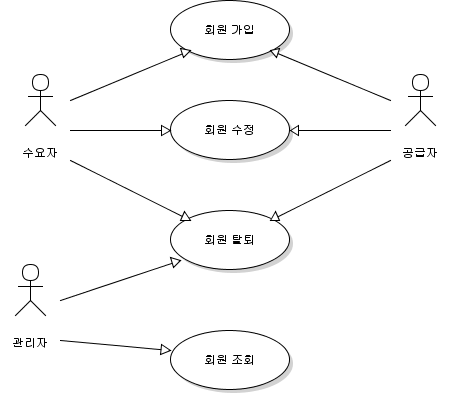
**2.4.4 신고 관리**



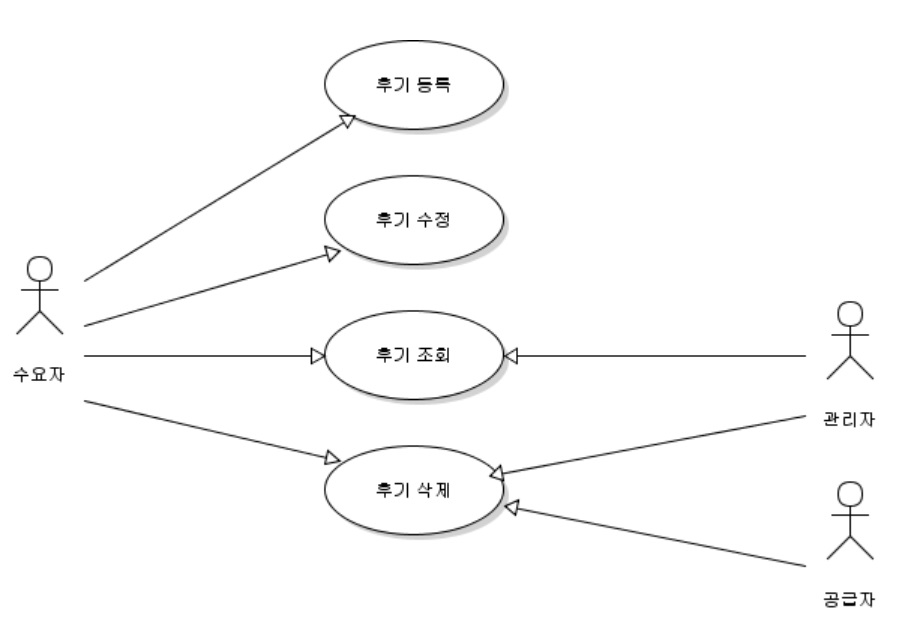
**2.4.5 예약 관리**

****

**2.4.6 회원 관리**

****

**2.4.7 후기 관리**

****

**3. 비기능적 요구사항(Non-functional Requirement)**

|  |  |  |
| --- | --- | --- |
| 품질 속성 | 비기능적 요구사항 | 우선 순위 |
| 가용성  (Availability) | 웹 페이지에서 24시간 멈추지 않고 서비스가 가동 되어야 하고 문제가 발생하는 경우 사용자에게 통지하고 적절하게 대처해야 한다. | **상** |
| 성능  (Performance) | 사용자 요청을 즉각적으로 처리하여 정보를 제공해야 한다. 또 사용자 정보는 오류가 발생 하지 않아야 한다. | **중** |
| 보안  (Security) | 사용자의 개인 정보가 외부에 유출 되지 않도록 보호되어야 한다. | **중** |
| 회원 간의 정보 공개 동의가 없을 경우 회원 정보가 공개 되어서는 안 된다. | **중** |
| 변경 가능성  (Modifiability) | 사용자 등급에 따라 권한이 달라진다. | **하** |
| 협력 업체 계약 함의 따라 즉각적으로 변경이 이루어져야 한다. | **상** |
| 사용성  (Usability) | 웹 페이지에 접근 하는 모든 사용자가 사용 할 수 있어야 하며, 회원가입 한 사용자는 공간을 예약하거나 제공할 수 있어야 한다. | **상** |
| 간결성  (sinblicity) | 공간을 대여하는 사용자가 예약 가능한 공간을 쉽게 조회할 수 있어야 한다. | **하** |
| 상호 운용성  (Interoperability) | 사용자가 개개인이 정보를 저장 하지 않고 서버에 Data Base System에서 일괄 처리 되어야 한다. | **상** |
| 타 사이트에서 운용하는 시스템기능(지도, 길찾기)을 사용하는데 있어서 오류 없이 처리되어야 한다. |  |

**4. UseCase Description**

**4.1 공간관리**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ID | U001 | Use-Case 명 | | 공간 등록 |
| 작성자 | 공동작성 | 수정 | |  |
| 최초 작성일 | 2017.09.11 | 최종 수정일 | | 2017.09.14 |
| Actor | 공급자 | | | |
| 사전 조건 | 로그인 | | | |
| 사후 조건 | 없음 | | | |
| 기본 흐름 | 액 터 행위 | | 시스템 행위 | |
|  | 1.공간등록을 선택한다.  3.공간 정보를 입력한다.  4.공간 등록을 요청한다. | | 2.공간등록 화면을 표시한다.  [필수입력사항을 미 입력 했을 경우]  5. 필수입력 항목에 미 입력 메시지를 표시한다.  [필수입력사항을 입력 했을 경우]  5.입력 조건의 유효성을 검사한다.  {유효성 검사에 통과하지 못한 경우}  6.통과하지 못한 입력 항목에 유효성 불합격 메시지를 표시한다.  {유효성 검사에 통과한 경우}  (등록에 성공한 경우)  6.등록 시 기입한 정보를 저장한다.  7.입력한 정보를 화면에 표시한다.  (등록에 실패한 경우)  6.등록 실패 오류 메시지를 표시한다. | |
| 비고 | 공간 정보 : 상호 명, 주소, 연락처, 대표 이미지, 편의시설, 여는 시간,  닫는 시간, 세부 공간, 소개 글, 공간유형  세부 공간 :　방 이름, 수용인원, 이미지, 가격(시간 당 이용 요금)  공간 유형 : 파티 룸, 회의실, 스터디 룸, 다목적 홀, 연습실  편의시설 : 주차, 와이파이, 프로젝터, 마이크, PC, 음식물반입, 화이트보드, 주류반입, 반려동물가능  필수입력 항목 : 상호 명, 주소, 연락처, 대표 이미지, 여는 시간, 닫는 시간, 공간 유형, 세부 공간(수용인원, 가격), 개수 값  유효성 검사 항목  1. 가격 형식 확인(숫자 형식 확인)  2. 연락처 형식 (000-0000-0000 숫자 형식 확인)  3. 대표 이미지 형식(확장자가 jpg, png, gif 형식 확인)  4. 여는 시간과 닫는 시간은 일치 하지 않음  5. 여는 시간은 닫는 시간보다 우선됨  필수입력 항목 미 입력 메시지 : “필수 입력 항목을 입력해주세요.”  유효성 불합격 메시지 : “올바른 입력 형식이 아닙니다.”  등록 실패 오류 메시지 : “등록에 실패 하였습니다.” | | | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ID | U002 | Use-Case 명 | | 공간 수정 |
| 작성자 | 이 훈 | 수정 | |  |
| 최초 작성일 | 2017.09.11 | 최종 수정일 | | 2017.09.14 |
| Actor | 공급자 | | | |
| 사전 조건 | 로그인 | | | |
| 사후 조건 | 없음 | | | |
| 기본 흐름 | 액 터 행위 | | 시스템 행위 | |
|  | 1.공간 수정을 선택한다.  4.수정 정보를 입력한다.  5.공간 수정을 요청한다. | | 2. 수정을 선택한 공간을 조회한다.  3. 수정할 공간 정보를 표시한다.  [유효성 검사에 통과하지 못한 경우]  6.통과하지 못한 입력 항목에 유효성 불합격 메시지를 표시한다.  [유효성 검사에 통과한 경우]  (수정에 성공한 경우)  6.수정 시 기입한 정보를 저장한다.  7.입력한 정보를 화면에 표시한다.  (수정에 실패한 경우)  6.수정 실패 오류 메시지를 표시한다. | |
| 비고 | 수정 정보 : 주소  세부 공간(방 이름, 수용인원, 이미지, 가격)  유효성 검사 항목  1. 금액 형식 확인(숫자 형식 확인)  2. 이미지 형식(확장자가 jpg, png, gif 형식 확인)  3. 수용인원 형식 확인(숫자 형식 확인)  수정 실패 오류 메시지 : “수정에 실패하였습니다.” | | | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ID | U003 | Use-Case 명 | | 공간 삭제 |
| 작성자 | 이 훈 | 수정 | |  |
| 최초 작성일 | 2017.09.11 | 최종 수정일 | | 2017.09.14 |
| Actor | 공급자, 관리자 | | | |
| 사전 조건 | 로그인 | | | |
| 사후 조건 | 없음 | | | |
| 기본 흐름 | 액 터 행위 | | 시스템 행위 | |
|  | 1.공간 삭제를 선택한다. | | (공간 삭제에 성공한 경우)  2.공간을 삭제한다.  2.1.세부공간을 삭제한다.  2.2.후기목록을 삭제한다.  2.3.공간 정보 삭제 성공 메시지를 표시한다.  (공간 삭제에 실패한 경우)  2.공간 정보 삭제 실패 메시지를 표시한다. | |
| 비고 | 삭제 확인 메시지: “공간이 삭제되었습니다.”  공간 정보 삭제 실패 메시지: “삭제에 실패하였습니다.” | | | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ID | U004 | Use-Case | | 공간 조회 |
| 작성자 | 이 훈 | 수정 | |  |
| 최초 작성일 | 2017.09.11 | 최종 수정일 | | 2017.09.14 |
| Actor | 수요자, 관리자, 공급자 | | | |
| 사전 조건 | 없음 | | | |
| 사후 조건 | 없음 | | | |
| 기본 흐름 | 액 터 행위 | | 시스템 행위 | |
|  | 1.검색 조건을 입력 후 조회를 요청한다.  4. 공간을 선택한다. | | [회원 일 경우]  2.회원 정보를 조회한다.  2-1.가입조건에 따라 공간목록을 조회한다.  3. 조회결과를 표시한다.  5. 선택 된 공간을 조회한다.  5.1. 해당 공간의 세부공간을 조회한다.  5.2. 해당 공간의 후기를 조회한다.  6. 조회결과를 표시한다. | |
| 비고 | 선택 검색 조건 : 지역, 공간유형, 예약 일자, 예약 인원  공간 유형 : 파티 룸, 회의실, 스터디 룸, 다목적 홀, 연습실  가입 조건: 선호지역을 선택한 경우, 선호하는 공간유형을 선택한 경우,  둘 다 선택한 경우, 둘 다 선택하지 않은 경우 | | | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ID | U005 | Use-Case 명 | | 공간 신고 |
| 작성자 | 김 성미 | 수정 | |  |
| 최초 작성일 | 2017.09.12 | 최종 수정일 | | 2017.09.14 |
| Actor | 수요자 | | | |
| 사전 조건 | 로그인 | | | |
| 사후 조건 | 없음 | | | |
| 기본 흐름 | 액 터 행위 | | 시스템 행위 | |
|  | 1.신고 할 공간을 선택한다.  [신고한 공간에 대한 예약 내역이 있는 경우]  4.신고 사유를 입력한다.  5.신고를 요청한다. | | 2. 신고한 수요자의 예약 내역을 조회한다.  [신고한 공간을 이용했을 경우]  3. 신고사유 입력화면을 표시한다.  [신고한 공간을 이용하지 않았을 경우]  3. 신고 불가 메시지를 표시한다.  6. 신고 된 공간의 상호 명과  공급자의 이름과 신고사유를 공간  신고 목록에 저장한다. | |
| 비고 | 신고 불가 메시지: “공간 미사용자는 신고가 불가 합니다.” | | | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ID | U006 | Use-Case 명 | | 공간 스크랩 |
| 작성자 | 이 훈 | 수정 | |  |
| 최초 작성일 | 2017.09.12 | 최종 수정일 | | 2017.09.14 |
| Actor | 수요자 | | | |
| 사전 조건 | 로그인 | | | |
| 사후 조건 | 없음 | | | |
| 기본 흐름 | 액 터 행위 | | 시스템 행위 | |
|  | 1.스크랩을 원하는 공간을 선택한다. 2.스크랩을 요청한다. | | 3. 해당 회원의 스크랩목록을 조회한다.  [선택 공간이 스크랩 되어 있지 않은 경우]  4.선택된 정보를 스크랩 목록에 저장한다.  5.스크랩 성공 메시지를 표시한다.  [선택 공간이 스크랩 되어 있는 경우]  4.스크랩 목록에서 선택된 항목을 삭제한다.  5.스크랩 해제 메시지를 표시한다. | |
| 비고 | 스크랩 목록 : 상호 명, 주소, 대표 이미지  스크랩 성공 메시지: “스크랩이 완료되었습니다.”  스크랩 해제 메시지: “스크랩이 해제되었습니다.” | | | |

**4.2 회원관리**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ID | U007 | Use-Case 명 | | 회원 가입 |
| 작성자 | 강 서윤 | 수정 | |  |
| 최초 작성일 | 2017.09.11 | 최종 수정일 | | 2017.09.14 |
| Actor | 수요자, 공급자 | | | |
| 사전 조건 | 없음 | | | |
| 사후 조건 | 없음 | | | |
| 기본 흐름 | 액 터 행위 | | 시스템 행위 | |
|  | 1.회원가입을 선택 한다.  3.수요자, 공급자를 선택한다.  5.메일주소를 입력한다.  8.비밀번호를 입력한다.  9.비밀번호 확인을 입력한다.  [수요자를 선택한 경우]  11.이름, 연락처, 선호지역,  선호하는 공간유형을 입력한다.  [공급자 선택 시]  11.이름, 연락처를 입력한다.  12.회원가입을 요청한다. | | 2.수요자, 공급자 선택화면을 표시한다.  4.회원가입 화면을 표시한다.  6.메일주소 중복 여부를 검사한다.  [중복일 경우]  7.중복메시지를 표시한다.  [중복이 아닐 경우]  7.메일주소 사용가능 메시지를 표시한다.  [비밀번호가 일치 할 경우]  10. ‘일치’ 메시지를 표시한다.  [비밀번호가 일치 하지 않을 경우]  10. ‘불일치’ 메시지를 표시한다.  [필수입력사항을 미 입력 했을 경우]  13. 필수입력 항목에 미 입력 메시지를 표시한다.  [필수입력사항을 입력 했을 경우]  13.입력 조건의 유효성을 검사한다.  [유효성 검사에 통과하지 못한 경우]  14.통과하지 못한 입력 항목에 유효성 불합격 메시지를 표시한다.  [유효성 검사에 통과한 경우]  (가입에 성공한 경우)  14.가입 시 기입한 정보를 저장한다.  15.입력한 정보를 화면에 표시한다.  (가입에 실패한 경우)  14.가입 실패 오류 메시지를 표시한다. | |
| 비고 | 회원 가입 필수 입력 조건 : 메일주소, 이름, 비밀번호, 연락처  유효성 검사 항목  1. 메일주소 중복 검사 통과  2. 메일주소 형식 확인([xxx@xxx.xxx](mailto:xxx@xxx.xxx) 형식 확인)  3. 비밀 번호와 비밀 번호 확인의 일치  4. 비밀 번호 형식 확인 ( 8자리 이상, 영문 대소문자 조합 )  5. 연락처 형식 (000-0000-0000 숫자 형식 확인)  6. 회원 구분(수요자, 공급자)  중복메시지: “중복된 메일입니다.”  메일주소 사용가능 메시지: “사용가능한 메일입니다.”  가입 실패 오류 메시지: “가입에 실패 하였습니다.” | | | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ID | U008 | Use-Case 명 | | 회원 수정 |
| 작성자 | 강 서윤 | 수정 | |  |
| 최초 작성일 | 2017.09.12 | 최종 수정일 | | 2017.09.14 |
| Actor | 수요자, 공급자 | | | |
| 사전 조건 | 로그인 | | | |
| 사후 조건 | 없음 | | | |
| 기본 흐름 | 액 터 행위 | | 시스템 행위 | |
|  | 1.회원 수정을 선택한다.  [수요자일 경우]  4.연락처 또는 비밀번호 또는 선호지역 또는 선호하는 공간 유형을 입력한다.  [공급자일 경우]  4.연락처 또는 비밀번호를 입력한다.  5.회원 수정을 요청한다. | | 2.회원을 조회한다.  3.회원정보를 표시한다.  [필수입력사항을 미 입력 했을 경우]  6. 필수입력 항목에 미 입력 메시지를 표시한다.  [필수입력사항을 입력 했을 경우]  6.입력 조건의 유효성을 검사한다.  [유효성 검사에 통과하지 못한 경우]  7.통과하지 못한 입력 항목에 유효성 불합격 메시지를 표시한다.  [유효성 검사에 통과한 경우]  (수정에 성공한 경우)  7.수정 시 기입한 정보를 저장한다.  8.입력한 정보를 화면에 표시한다.  (수정에 실패한 경우)  7.수정 실패 오류 메시지를 표시한다. | |
| 비고 | 유효성 검사 항목  1. 비밀번호와 비밀번호 확인의 일치  회원 수정 필수 입력 조건 : 비밀번호, 연락처 | | | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ID | U009 | Use-Case 명 | | 회원 탈퇴 |
| 작성자 | 강 서윤 | 수정 | |  |
| 최초 작성일 | 2017.09.12 | 최종 수정일 | | 2017.09.14 |
| Actor | 수요자, 공급자, 관리자 | | | |
| 사전 조건 | 로그인 | | | |
| 사후 조건 | 없음 | | | |
| 기본 흐름 | 액 터 행위 | | 시스템 행위 | |
|  | [수요자, 공급자 일 경우]  1.회원 탈퇴를 선택한다.  3.비밀번호를 입력한다.  4.탈퇴를 요청한다.  [관리자일 경우]  1.회원 탈퇴를 선택한다.  3.탈퇴 시킬 회원을 선택한다. | | 2.비밀번호 입력화면을 표시한다.  [비밀번호가 일치 한 경우]  5.회원정보 삭제 후 탈퇴 확인 메시지를 표시한다.  [비밀번호가 불일치한 경우]  5.회원 정보 삭제 실패 메시지를 표시한다.  2.신고 목록을 조회한다.  [회원정보가 삭제된 경우]  4.회원정보 삭제 후 탈퇴 확인 메시지를 표시한다.  [회원정보가 삭제되지 않은 경우]  4.회원 정보 삭제 실패 메시지를 표시한다. | |
| 비고 | 회원정보: 메일주소, 이름  탈퇴 확인 메시지: “탈퇴 되었습니다.”  회원 정보 삭제 실패 메시지: “비밀번호가 일치하지 않습니다.” | | | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ID | U010 | Use-Case 명 | | 회원 조회 |
| 작성자 | 강 서윤 | 수정 | |  |
| 최초 작성일 | 2017.09.11 | 최종 수정일 | | 2017.09.14 |
| Actor | 관리자 | | | |
| 사전 조건 | 로그인 | | | |
| 사후 조건 | 없음 | | | |
| 기본 흐름 | 액 터 행위 | | 시스템 행위 | |
|  | 1.회원 조회를 선택한다. | | 2.조건에 따라 회원을 조회한다.  3.조회 결과를 표시한다. | |
| 비고 | 조건 : 전체조회, 메일주소별 조회  전체 조회 결과 : 메일 주소, 이름, 연락처  메일주소별 조회 결과 : 메일 주소, 이름, 연락처, 선호 지역, 선호하는 공간 유형 | | | |

**4.3 로그인**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ID | U011 | Use-Case 명 | | 로그인 |
| 작성자 | 강 서윤 | 수정 | |  |
| 최초 작성일 | 2017.09.12 | 최종 수정일 | | 2017.09.14 |
| Actor | 수요자, 공급자, 관리자 | | | |
| 사전 조건 | 없음 | | | |
| 사후 조건 | 없음 | | | |
| 기본 흐름 | 액 터 행위 | | 시스템 행위 | |
|  | 1.로그인을 선택한다.  3.메일주소와 비밀번호를 입력한다.  4.로그인을 요청한다. | | 2.로그인 화면을 표시한다.  5.회원 정보를 조회한다.  [메일주소와 비밀번호가 일치 할 경우]  6.로그인 후 메인화면을 표시한다.  [메일주소와 비밀번호가 일치하지 않을 경우]  6.로그인 화면을 표시한다. | |
| 비고 | 로그인 필수 입력 항목 : 메일주소, 비밀번호 | | | |

**4.4 후기관리**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ID | U012 | Use-Case 명 | | 후기 등록 |
| 작성자 | 김 재국 | 수정 | |  |
| 최초 작성일 | 2017.09.12 | 최종 수정일 | | 2017.09.14 |
| Actor | 수요자 | | | |
| 사전 조건 | 로그인 | | | |
| 사후 조건 | 없음 | | | |
| 기본 흐름 | 액 터 행위 | | 시스템 행위 | |
|  | 1. 후기를 입력한다.  2. 후기 등록을 요청한다. | | 3.수요자의 예약 내역을 조회한다.  [해당 공간을 이용했을 경우]  4.후기 등록 화면을 표시한다.  [해당 공간을 이용하지 않았을 경우]  4.후기 등록 불가 메시지를 표시한다.  [등록에 성공한 경우]  5.등록 시 기입한 정보를 저장한다.  6.입력한 정보를 화면에 표시한다.  [등록에 실패한 경우]  5.등록 실패 오류 메시지를 표시한다. | |
| 비고 | 후기 등록 불가 메시지 : “공간 이용 후 후기를 등록해주세요.” | | | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ID | U013 | Use-Case 명 | | 후기 수정 |
| 작성자 | 김 재국 | 수정 | |  |
| 최초 작성일 | 2017.09.12 | 최종 수정일 | | 2017.09.14 |
| Actor | 수요자 | | | |
| 사전 조건 | 로그인 | | | |
| 사후 조건 | 없음 | | | |
| 기본 흐름 | 액 터 행위 | | 시스템 행위 | |
|  | 1. 후기 수정을 요청한다.  4. 수정할 후기내용을 입력한다.  5. 후기 수정 등록을 요청한다. | | 2. 후기 목록을 조회한다.  3. 수정할 후기내용을 표시한다.  [수정에 성공한 경우]  6.수정 시 기입한 정보를 저장한다.  7.입력한 정보를 화면에 표시한다.  [수정에 실패한 경우]  6.수정 실패 오류 메시지를 표시한다.  . | |
| 비고 |  | | | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ID | U014 | Use-Case 명 | | 후기 삭제 |
| 작성자 | 김 재국 | 수정 | |  |
| 최초 작성일 | 2017.09.12 | 최종 수정일 | | 2017.09.14 |
| Actor | 수요자, 관리자 | | | |
| 사전 조건 | 로그인 | | | |
| 사후 조건 | 없음 | | | |
| 기본 흐름 | 액 터 행위 | | 시스템 행위 | |
|  | 1. 삭제할 후기를 선택한다.  2. 후기 삭제를 요청한다. | | [삭제에 성공한 경우]  3.후기를 삭제하고 삭제 성공 메시지를 표시한다.  [삭제에 실패한 경우]  3.삭제 실패 오류 메시지를 표시한다. | |
| 비고 | 삭제 성공 메시지: “후기가 삭제되었습니다.”  삭제 실패 오류 메시지: “삭제에 실패하였습니다.” | | | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ID | U015 | Use-Case 명 | | 후기 조회 |
| 작성자 | 김 재국 | 수정 | |  |
| 최초 작성일 | 2017.09.12 | 최종 수정일 | | 2017.09.14 |
| Actor | 수요자, 공급자, 관리자 | | | |
| 사전 조건 | 없음 | | | |
| 사후 조건 | 없음 | | | |
| 기본 흐름 | 액 터 행위 | | 시스템 행위 | |
|  | 1.후기 조회를 요청한다. | | 2.조건에 따라 후기 목록을 조회한다.  3. 조회결과 목록을 표시한다. | |
| 비고 | 후기 목록 : 작성자명, 작성시간, 후기내용  조건 : 전체조회, 회원별 조회, 공간별 조회 | | | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ID | U016 | Use-Case 명 | | 후기 신고 |
| 작성자 | 김 재국 | 수정 | |  |
| 최초 작성일 | 2017.09.12 | 최종 수정일 | | 2017.09.14 |
| Actor | 수요자 | | | |
| 사전 조건 | 로그인 | | | |
| 사후 조건 | 없음 | | | |
| 기본 흐름 | 액 터 행위 | | 시스템 행위 | |
|  | 1. 후기신고를 선택한다.  3. 후기신고 사유를 입력한다.  4. 후기신고를 요청한다. | | 2. 후기신고 사유 입력 화면을 표시한다.  [신고에 성공한 경우]  5. 신고 사유, 신고자의 메일 주소, 작성자의 메일주소, 후기를 후기 신고 목록에 저장한다.  6. 후기 신고 성공 메시지를 표시한다.  [신고에 실패한 경우]  5.후기 신고 실패 메시지를 표시한다. | |
| 비고 | 후기 신고 성공 메시지: “후기신고가 접수 되었습니다.”  후기 신고 실패 메시지: “후기신고에 실패하였습니다.” | | | |

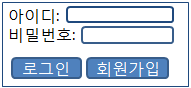
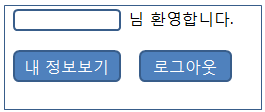
**4.5 예약관리**

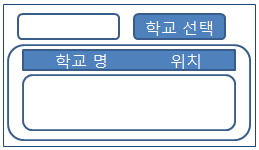
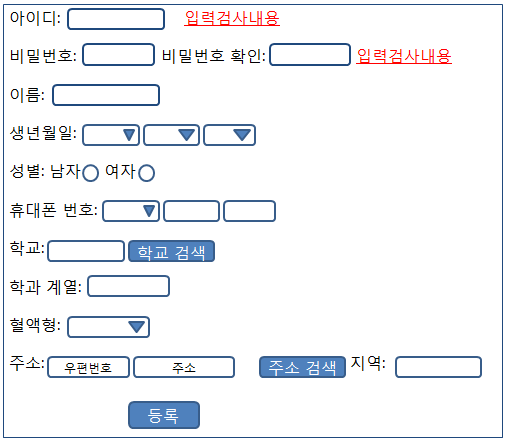
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ID | U017 | Use-Case 명 | | 공간 예약 |
| 작성자 | 김 성미 | 수정 | |  |
| 최초 작성일 | 2017.09.12 | 최종 수정일 | | 2017.09.14 |
| Actor | 수요자 | | | |
| 사전 조건 | 없음 | | | |
| 사후 조건 | 없음 | | | |
| 기본 흐름 | 액 터 행위 | | 시스템 행위 | |
|  | 1.세부 공간을 선택 후 예약하기를 선택한다.  [회원일 경우]  6.마일리지 유/무를 선택한다.  [비회원일 경우]  6.성명과 연락처를 입력한다.  [마일리지를 사용할 경우]  8.사용 할 마일리지를 입력하고 결제수단을 선택한다.  [마일리지를 사용하지 않을 경우]  8.결제수단을 선택한다.  [신용 카드 일 경우]  10. 카드정보를 입력한다.  11. 결제를 요청한다. | | 2.검색 시 입력한 조건 값을  표시한다.  3.선택한 공간을 조회한다.  4.상호명과 가격을 표시한다.  [회원일 경우]  5.회원을 조회한다.  5-1.이름과 연락처를 표시한다.  [비회원일 경우]  5.예약자 성명과 연락처 입력 화면을 표시한다.  [마일리지를 사용할 경우]  7.마일리지를 조회 후 표시한다.  [마일리지를 사용하지 않을 경우]  7.카드결제, 계좌이체 선택화면을  표시한다.  [마일리지를 입력한 경우]  8-1.유효성 검사를 한다.  (마일리지 유효성 검사 통과하지 못한 경우)  8-2.마일리지 사용 불가 메시지를 표시한다.  (마일리지 유효성 검사 통과한 경우)  8-2.결제 금액에서 마일리지만큼 차감한 후 표시한다.  [신용 카드 일 경우]  9. 카드 정보 입력 화면을 표시한다.  [계좌 이체 일 경우]  9.계좌번호, 은행명, 입금 주를 표시한다.  [카드 정보 사항을 미 입력 했을 경우]  12. 카드 정보입력 항목에 미 입력 메시지를 표시한다.  [카드 정보 사항을 입력 했을 경우]  (결제에 성공한 경우)  12.예약 시 기입한 정보를 저장한다.  (마일리지를 사용한 경우)  13.입력한 정보를 화면에 표시한다.  (마일리지를 사용하지 않은 경우)  13.결제 금액의 10%를 예약일 경과 1일 후 마일리지로 지급한다.  14.입력한 정보를 화면에 표시한다.  (결제에 실패한 경우)  12.결제 실패 오류 메시지를 표시한다. | |
| 비고 | 검색 시 입력한 조건 값 : 예약 일자, 입실 시간, 퇴실 시간, 사용 인원,  예약정보 : 예약 일자, 입실 시간, 퇴실 시간, 사용 인원, 대여 장소 명, 가격, 예약자 이름  카드 정보 : 카드구분, 카드 종류, 할부 기간, 카드 번호, 소유주 명, cvc  마일리지 유효성 검사 항목 :  보유 마일리지 금액 > 사용 요청한 마일리지 금액  마일리지 사용 불가 메시지: “마일리지가 부족합니다.”  결제 실패 오류 메시지: “결제에 실패하였습니다.” | | | |

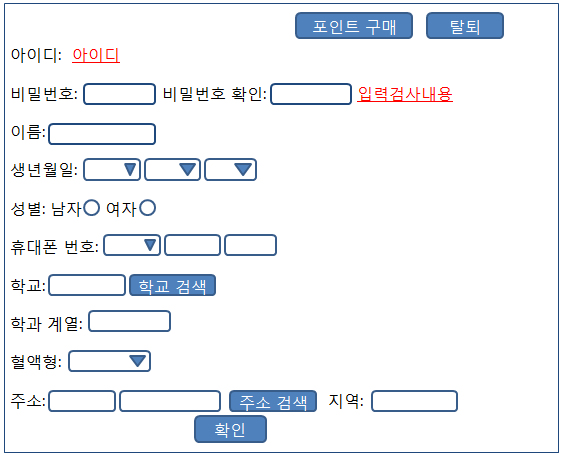
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ID | U018 | Use-Case 명 | | 예약 취소 |
| 작성자 | 김 성미 | 수정 | |  |
| 최초 작성일 | 2017.09.12 | 최종 수정일 | | 2017.09.14 |
| Actor | 수요자 | | | |
| 사전 조건 | 로그인 | | | |
| 사후 조건 | 없음 | | | |
| 기본 흐름 | 액 터 행위 | | 시스템 행위 | |
|  | 1.예약 취소를 선택한다.  3.취소하고자 하는 예약 항목을 선택한다. | | 2.예약 취소를 요청하는  수요자의 예약 목록을 표시한다.  4.선택된 예약항목을 조회한다.  5.환불 규정에 따라 환불 처리 후 예약을 취소한다. | |
| 비고 | 예약 항목 : 예약 일자, 대여 장소 명, 가격 | | | |

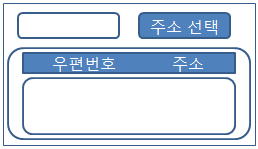
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ID | U019 | Use-Case 명 | | 예약 조회 |
| 작성자 | 김 성미 | 수정 | |  |
| 최초 작성일 | 2017/09/12 | 최종 수정일 | | 2017.09.14 |
| Actor | 수요자, 공급자 | | | |
| 사전 조건 | 없음 | | | |
| 사후 조건 | 없음 | | | |
| 기본 흐름 | 액 터 행위 | | 시스템 행위 | |
|  | 1. 예약 조회를 요청한다. | | 2.조건에 따라 예약 목록을 조회한다.  3. 조회 결과를 표시한다. | |
| 비고 | 조회 결과 :　예약 일자, 입실 시간, 퇴실 시간, 사용 인원, 대여 장소 명,  가격, 예약자 메일주소, 예약자 이름  조건 : 전체조회, 메일주소별 조회, 대여 장소별 조회 | | | |

**5. Layout Description**

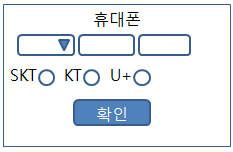
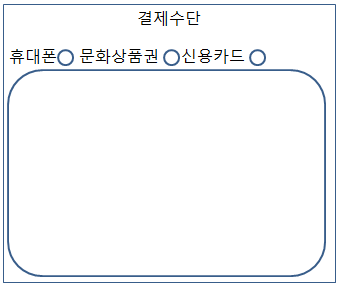
**5.1 로그인**

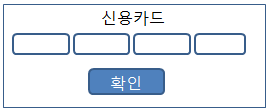
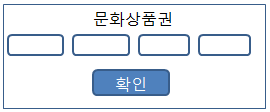
**5.2 개인정보관리**



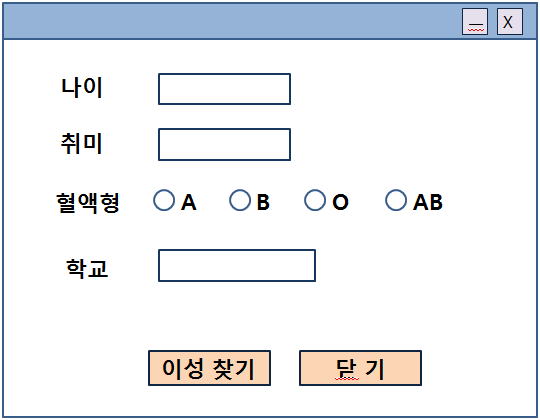


**(결제)**



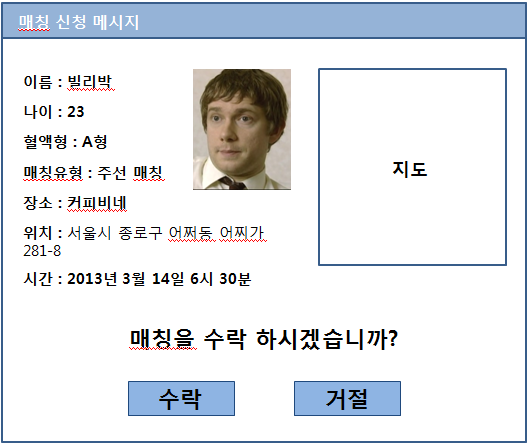


**5.3 매칭**

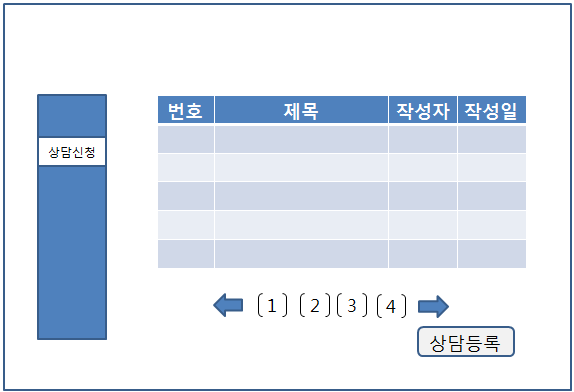
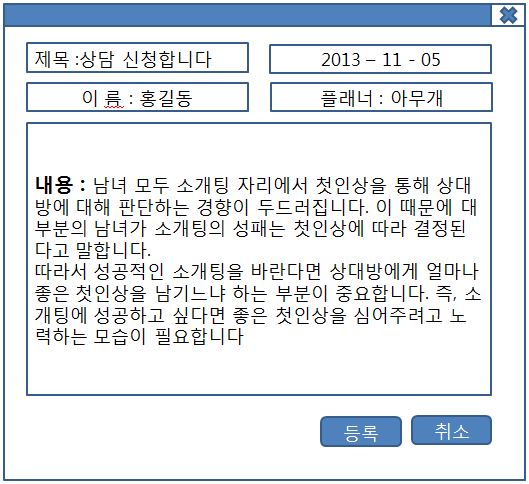




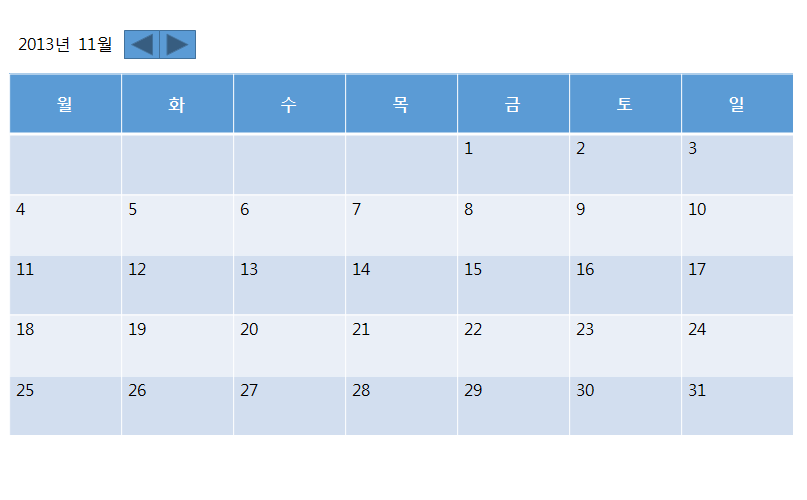


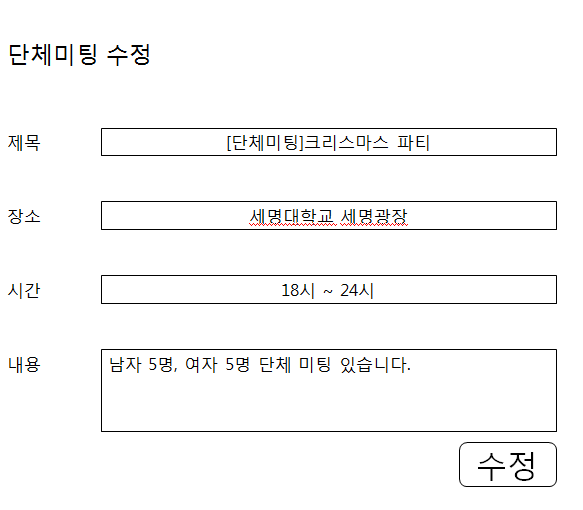


**5.4 상담**

****

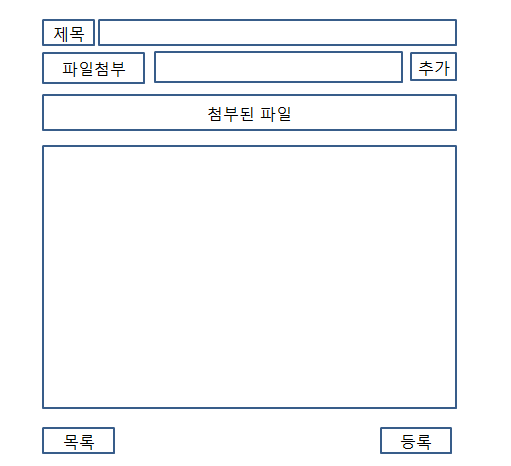
**5.5 단체미팅**

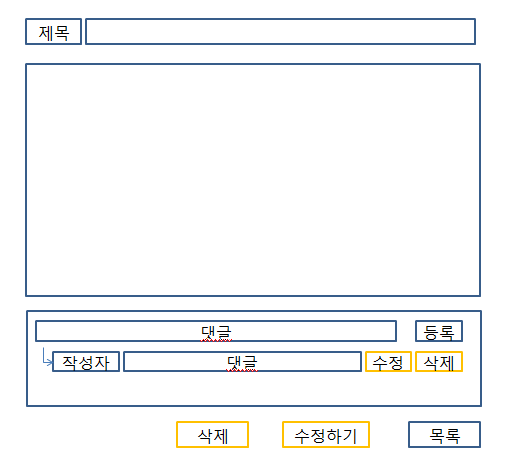
****

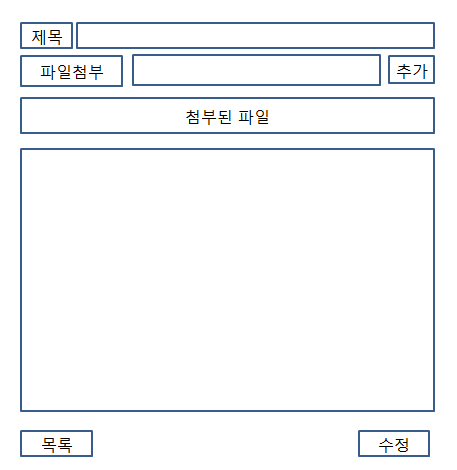




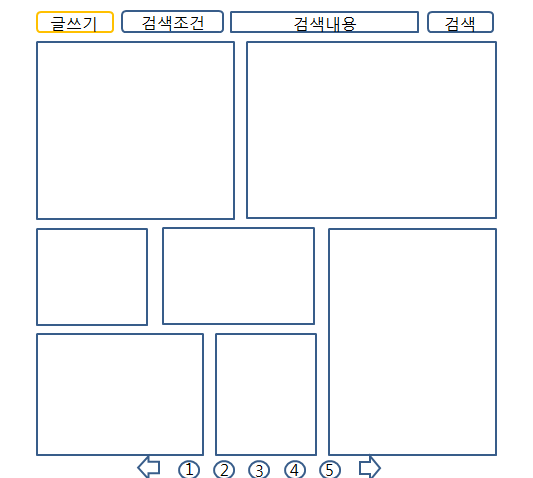
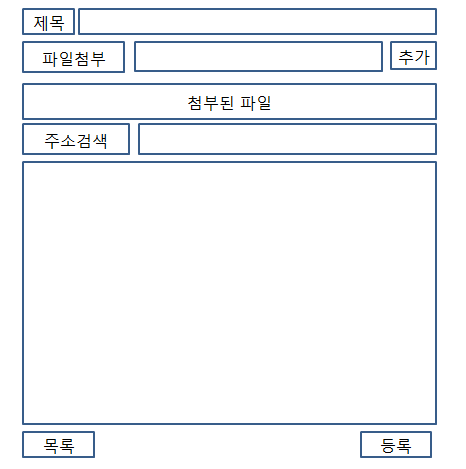
**5.6 자유 게시판, 공지 게시판**

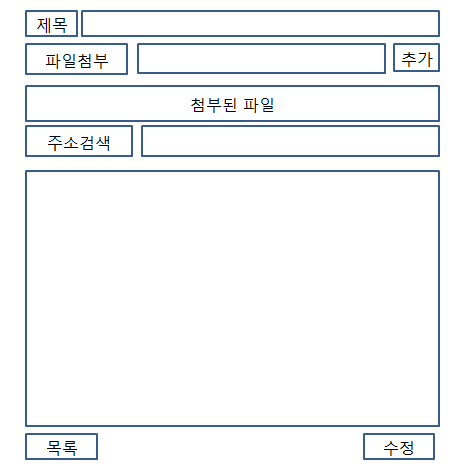
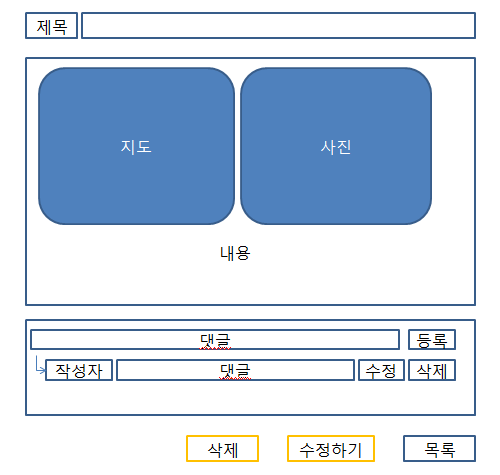




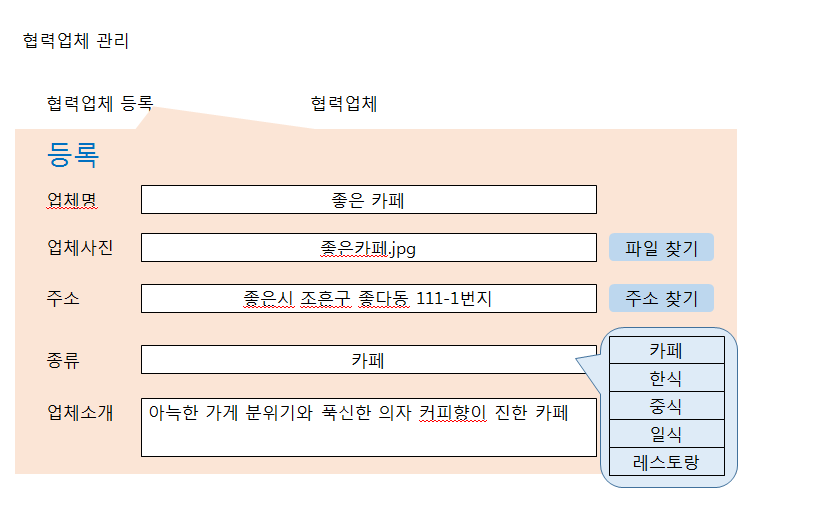
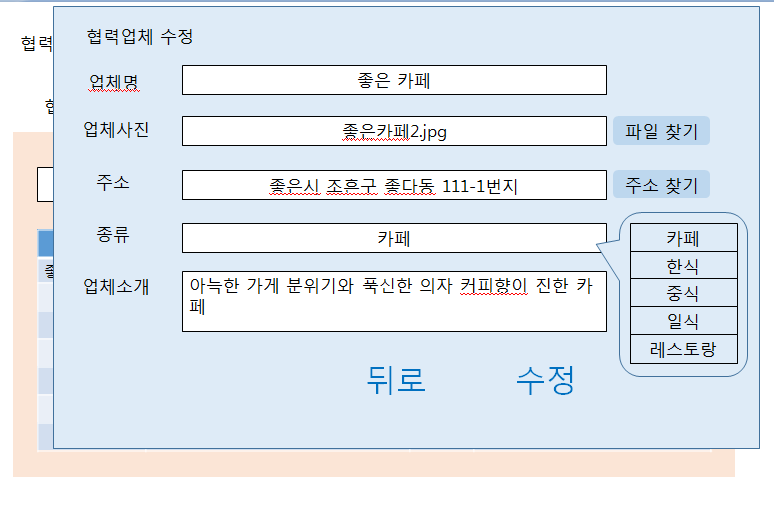


**5.7 추천 여행지 게시판**

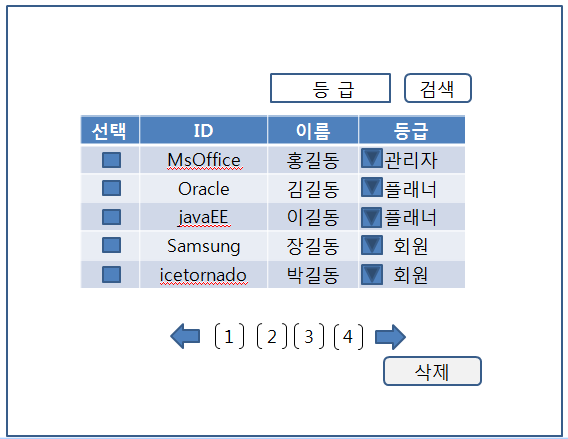
****



**5.8 협력업체관리**



**5.9 회원관리**



**6. 2차 Key Abstraction**

**6.1 로그인**

|  |  |  |
| --- | --- | --- |
| 동사 | | |
| 로그인 | 메일주소와 비밀번호를 | 조회한다. |

**6.2 회원**

|  |
| --- |
| 명사(회원) |
| 메일주소  회원구분  비밀번호  이름  연락처  선호지역  선호하는 공간유형  마일리지 |

|  |  |  |
| --- | --- | --- |
| 동사 | | |
| 로그인 | 메일주소 중복 여부를 | 검사하다. |
| 회원을 | 등록하다. |
| 회원 조회 | 메일주소로 회원을 | 조회하다. |
| 전체 회원을 |
| 회원 수정 | 회원을 | 조회하다. |
| 회원을 | 수정하다. |
| 회원 탈퇴 | 회원을 | 삭제하다. |

**6.3 공간**

|  |  |  |
| --- | --- | --- |
| 동사 | | |
| 공간 등록 | 공간을 | 저장하다 |
| 공간 수정 | 공간을 | 조회하다 |
| 공간을 | 수정하다 |
| 공간 삭제 | 공간을 | 삭제하다 |
| 공간 조회 | 회원정보(선호지역)로 공간을 | 조회하다 |
| 회원정보(선호하는 공간 유형)로 공간을 |
| 회원정보(선호지역, 선호하는 공간 유형)로 공간을 |
| 전체 공간을 |

|  |
| --- |
| 명사(공간) |
| 상호 명  주소  연락처  대표이미지  편의시설  여는 시간  닫는 시간  세부 공간  소개 글  공간 유형 |

|  |  |  |
| --- | --- | --- |
| 동사 | | |
| 공간 등록 | 공간을 | 저장하다 |
| 공간 수정 | 공간을 | 조회하다 |
| 공간을 | 수정하다 |
| 공간 삭제 | 공간을 | 삭제하다 |
| 공간 조회 | 회원정보(선호지역)로 공간을 | 조회하다 |
| 회원정보(선호하는 공간 유형)로 공간을 |
| 회원정보(선호지역, 선호하는 공간 유형)로 공간을 |
| 전체 공간을 |

|  |  |  |
| --- | --- | --- |
| 동사 | | |
| 공간 등록 | 공간을 | 저장하다 |
| 공간 수정 | 공간을 | 조회하다 |
| 공간을 | 수정하다 |
| 공간 삭제 | 공간을 | 삭제하다 |
| 공간 조회 | 회원정보(선호지역)로 공간을 | 조회하다 |
| 회원정보(선호하는 공간 유형)로 공간을 |
| 회원정보(선호지역, 선호하는 공간 유형)로 공간을 |
| 전체 공간을 |

|  |
| --- |
| 명사(세부공간) |
| 방 이름  수용 인원  이미지  가격 |

**6.4 세부공간**

**6.5 후기**

|  |  |  |
| --- | --- | --- |
| 동사 | | |
| 후기 등록 | 후기내용을 | 등록하다 |
| 후기 수정 | 후기내용을 | 수정하다 |
| 후기 삭제 | 후기를 | 삭제하다 |
| 후기 조회 | 회원의 메일주소로 후기를 | 조회하다 |
| 공간의 아이디로 후기를 |

|  |
| --- |
| 명사(후기) |
| 후기내용  작성자명  작성시간 |

**6.6 신고**

|  |
| --- |
| 명사(신고) |
| 날짜  제목  장소  시간  내용  일정보기  요일 |

|  |  |  |
| --- | --- | --- |
| 동사 | | |
| 신고 | 신고자의 메일주소를 | 저장하다 |
| 신고당한자의 메일주소를 |
| 신고 사유를 |
| 신고 항목을 |
| 신고 조회 | 전체 신고목록을 | 조회하다 |
| 신고당한자의 메일주소로 |
| 신고 삭제 | 신고를 | 저장하다 |

**6.7 스크랩**

|  |  |  |
| --- | --- | --- |
| 동사 | | |
| 스크랩 | 스크랩 목록을 | 조회하다. |
| 공간을 스크랩 목록에 | 저장하다. |
| 스크랩 삭제 | 스크랩 목록을 | 조회하다. |
| 공간을 스크랩 목록에서 | 삭제하다. |
| 스크랩 조회 | 회원의 메일주소로 | 조회하다. |

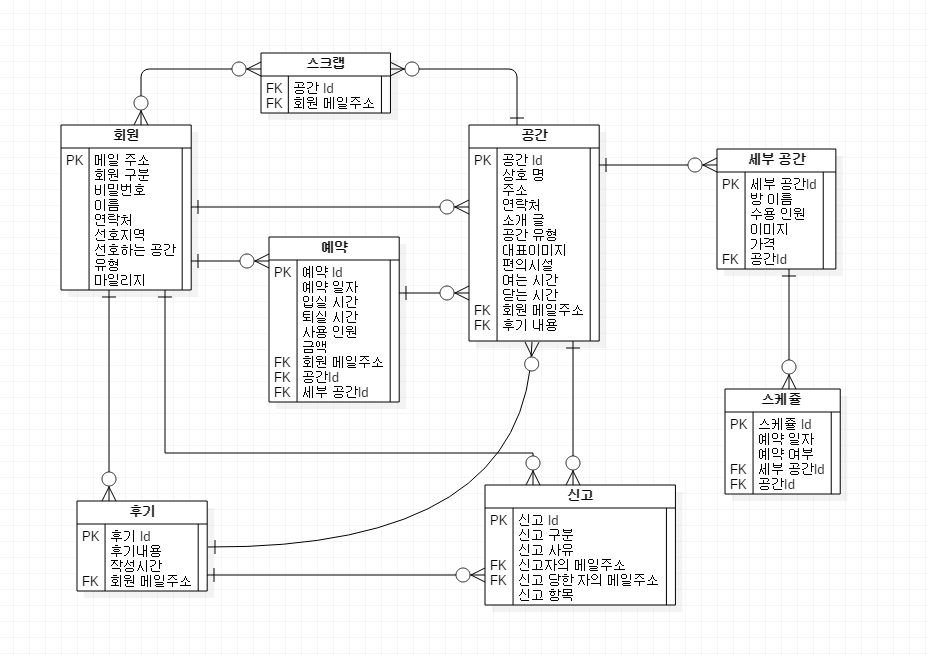
|  |
| --- |
| 스크랩 |
| 공간  회원 |

**6.8 예약**

|  |  |  |
| --- | --- | --- |
| 동사 | | |
| 공간 예약 | 예약정보(예약일자, 입실시간, 퇴실시간, 사용인원, 대여 장소 명, 가격, 예약자 이름)를 | 저장하다. |
| 예약 조회 | 전체 예약 목록을 | 조회하다. |
| 회원의 메일주소로 |
| 예약 취소 | 예약 항목을 | 저장하다. |

|  |
| --- |
| 명사 |
| 예약 일자  입실 시간  퇴실 시간  사용 인원  대여 장소 명  금액  예약자 이름 |

**7. ER Diagram**



**8. Class Description(영문)**

**8.1 Domain Class**

|  |
| --- |
| Member |
| -memberMail:String  -memberType:String  -password:String  -memberName:String  -contact:String  -favoriteLocation:String  -favoritePlaceType:String  -mileage:int |

|  |
| --- |
| Comment |
| -commentId:int  -contents:String  -memberMail:String  -commentDate:Date |

|  |
| --- |
| Report |
| -reportId:int  -reporterMail:String  -suspectMail:String  -reportType:String  -reportContents:String  -reportTargetId:String |

|  |
| --- |
| Place |
| -placeId:int  -placeName:String  -address:String  -placeContact:String  -thumbnail:String  -facility:String  -openTime:int  -closeTime:int  -detailPlaces:List<DetailPlace>  -comments:List<Comment>  -introduction:String  -placeType:String |

|  |
| --- |
| DetailPlace |
| -detailPlaceId:int  -detailPlaceName:String  -acceptableCount:int  -image:String  -price:int  -placeId:int |

|  |
| --- |
| Reservation |
| -reservationId:int  -reservationDate:Date  -checkIn:int  -checkOut:int  -userCount:int  -reservationPrice:int  -memberMail:String  -placeId:int  -detailPlaceId:int |

|  |
| --- |
| Scrap |
| -places:List<Place>  -memberMail:String |

|  |
| --- |
| Schedule |
| -detailPlaceId:int  -scheduleDate:Date  -reservationFlag:String  -reservationFlagList:List<String>  -placeId:int  -detailPlaceId:int |

**8.2 DAO Class**

|  |
| --- |
| MemberStore |
| +createMember(member : Member) : boolean  +retrieveAllMembers() : List<Member>  +retrieveMemberByMail(memberMail : String) : Member  +updateMember(member : Member) : boolean  +deleteMember(memberMail : String) : boolean  +updateMilege(String memberMail, int mileage):boolean |

|  |
| --- |
| ReportStore |
| +createReport(report : Report) : boolean  +retrieveAllReports() : List<Report>  +retrieveReportsBySuspectMail(suspectMail : String) : List<Report>  +retrieveReportsByReportType(reportType : String) : List<Integer>  +deleteReport(reportId : int) : boolean |

|  |
| --- |
| ReservationStore |
| +createReservation(reservation : Reservation) : boolean  +createSchedule(schedule : Schedule) : boolean  +updateSchedule( schdule : Schedule ) : Boolean  +retrieveScheduleByReservationId( reservationId : int) : Reservation  +retrieveScheduleByDate(scheduleDate : Date, detailPlaceId : int): Schedule  +retrieveReservationsByPlaceId(placeId : int) : List<Reservation>  +retrieveReservationsByMemberMail(memberMail : String) : List<Reservation>  +deleteReservation(reservationId : int ) : boolean |

|  |
| --- |
| ScrapStore |
| +createScrap(placeId : int , memberMail : String) : boolean  +retrieveScrapPlacesByMemberMail(memberMail : String) : List<Place>  +deleteScrap(placeId : int , memberMail : String) : boolean  +existMemberPlace(placeId : int , memberMail : String) : boolean |

|  |
| --- |
| DetailPlaceStore |
| +createDetailPlace(detailPlace : DetailPlace) : boolean  +retrieveDetailPlacesByPlaceId(placeId : int) : List<DetailPlace>  +updateDetailPlace(detailPlace : DetailPlace) : boolean  +deleteDetailPlace(detailPlaceId : int) : boolean  +deleteSchedule(detailPlaceId : int) : boolean |

|  |
| --- |
| CommentStore |
| +createComment(comment : Comment) : boolean  +retrieveCommentByCommentId(CommentId : int) : Comment  +retrieveReportedComments(CommentIds : List<Integer>) : List<Comment>  +retrieveCommentsByPlaceId(placeId : int) : List<Comment>  +updateComment(comment : Comment) : boolean  +deleteComment(commentId : int ) : boolean |

|  |
| --- |
| PlaceStore |
| +createPlace(place : Place, memberMail : String) : int  +retrieveEmptyPlacesByTime(date : Date , time : String): List<Place>  +retrievePlacesByLocation(date : Date, time : String ,location : String): List<Place>  +retrievePlacesByPlaceType(date : Date , time : String ,placeType : String) : List<Place>  +retrievePlacesByLoationPlaceType  (date : Date , time :String ,placeType : String,location : String) : List<Place>  +retrievePlaceByPlaceId(placeId : int) : Place  +retrievePlacesByHostMail(String memberMail) : List<Place>  +updatePlace(place : Place) : boolean  +deletePlace(placeId : int) : boolean |

**8.3 Service**

|  |
| --- |
| MemberServiceLogic |
| -MemberStore  -ScrapStore  -PlaceStore |
| +registMember(member : Member) : Boolean  +findMemberByMail(memberMail : String) : Member  +modifyMember(member : Member) : boolean  +removeMember(member : Member) : boolean |

|  |
| --- |
| ReservationServiceLogic |
| -ReservationStore |
| +registReservation(reservation : Reservation ) : boolean  +findScheduleByDate(scheduleDate : Date, detailPlaceId : int): Schedule  +findReservationByMemberMail(memberMail : String) : List<Reservation>  +findReservationByPlaceId(placeId : int) : List<Reservation>  +removeReservation(reservationId : int) : boolean |

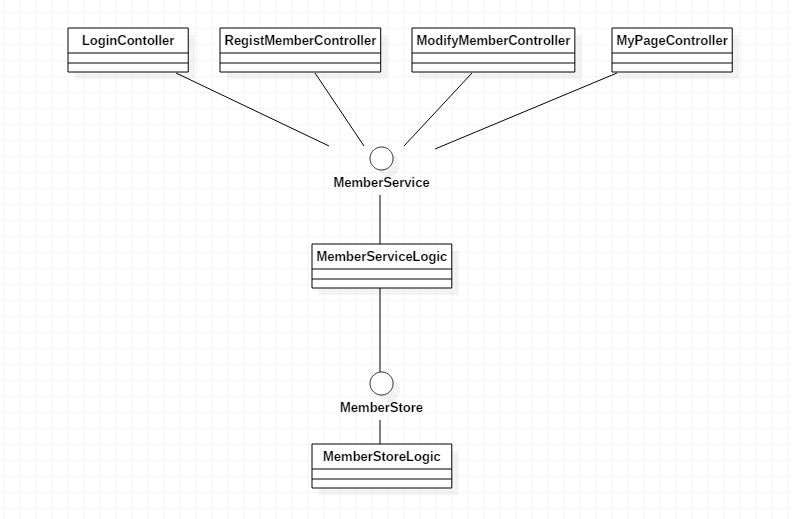
|  |
| --- |
| ReportServiceLogic |
| -ReportStore  -CommentStore |
| +registReport(report : Report) : boolean  +findAllReport() : List<Report>  +findReportsBySuspectMail(suspectMail : String) : List<Report>  +findReportedComment(reportType : String) : List<Comment>  +findReportedPlaceList(reportType : String) : List<Integer>  +removeReport(reportId : int) : boolean |

|  |
| --- |
| PlaceServiceLogic |
| -DetailPlaceStore  -CommentStore  -PlaceStore  -ScheduleStore |
| +registPlace(place : Place, memberMail : String) : int  +registDetailPlace(detailPlace : DetailPlace) : boolean  +registComment(comment : Comment) : boolean  +multiplyDetailPlace(detailPlace : DetailPlace , count : Int) : boolean  +modifyPlace(place : Place) : boolean  +modifyDetailPlace(detailPlace : DetailPlace) : boolean  +modifyComment(comment : Comment) :　boolean  +removePlace(placeId : String) : boolean  +removeDetailPlace(detailId : Int) : boolean  +removeComment(commentId : Int) ：boolean  +findCommentByCommentId(CommentId : int) : Comment  +findEmptyPlacesByLocation  (checkIn : int , checkOut : int , date : Date , location : String) :Map<String,List<Place>>  +findEmptyPlacesByPlaceType  (checkIn : int , checkOut : int , date : Date , placeType : String) : Map<String,List<Place>>  +findEmptyPlacesByLocationPlaceType  (checkIn : int , checkOut : int , date : Date , placeType : String, location : String) :  Map<String,List<Place>>  +findPlaceByPlaceId(PlaceId : String) : Place  +findAllEmptyPlaces  (checkIn : int , checkOut : int , date : Date) : Map<String,List<Place>> |

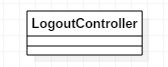
|  |
| --- |
| ScrapServiceLogic |
| -ScrapStore |
| +registScrap(placeId : int , memberMail : String) : boolean  +findScrapPlacesByMemberMail(memberMail : String) : List<Place>  +removeScrap(placeId : int , memberMail : String) : boolean |

**9. Class Diagram(영문)**

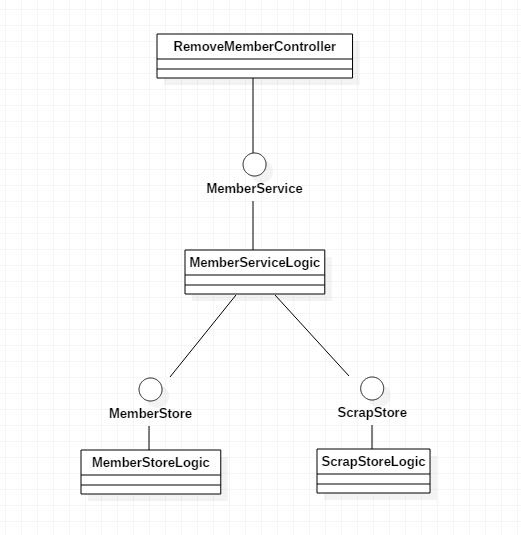
**9.1 회원관리**

****

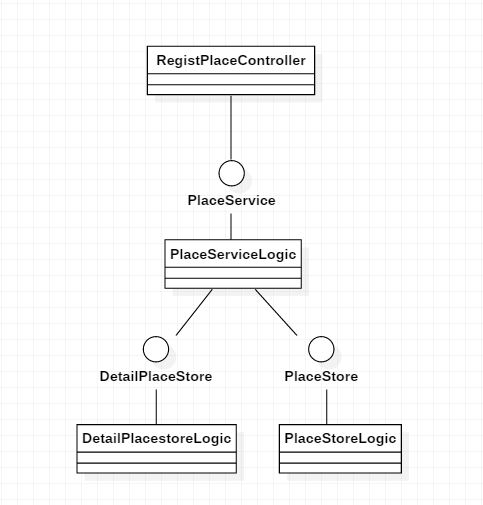
**9.1 회원관리(로그아웃)**

****

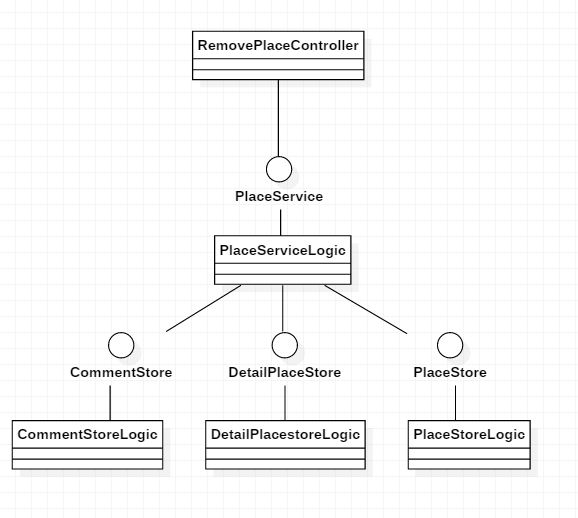
**9.1 회원관리(탈퇴)**

****

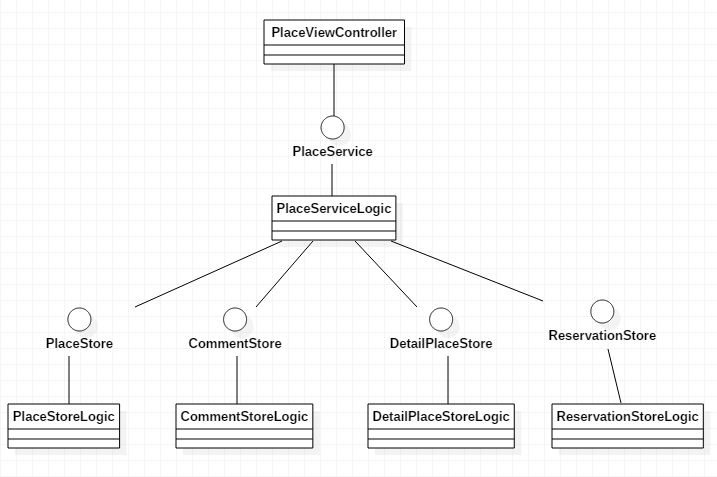
**9.2 공간관리(공간등록)**

****

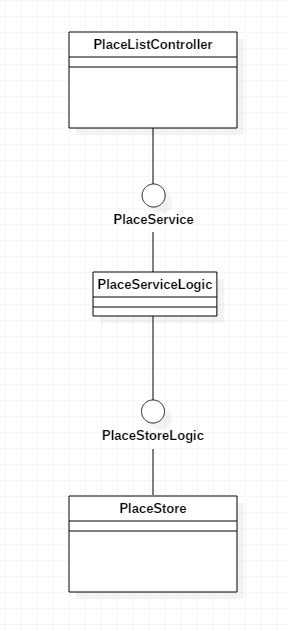
**9.2 공간관리(공간삭제)**

****

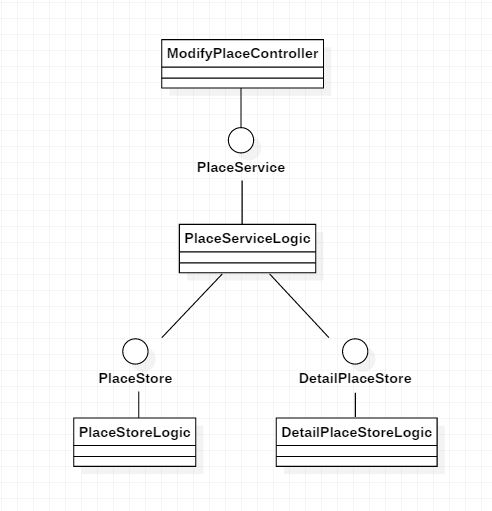
**9.2 공간관리(공간 조회)**

****

**9.2 공간관리(전체 공간 조회)**

****

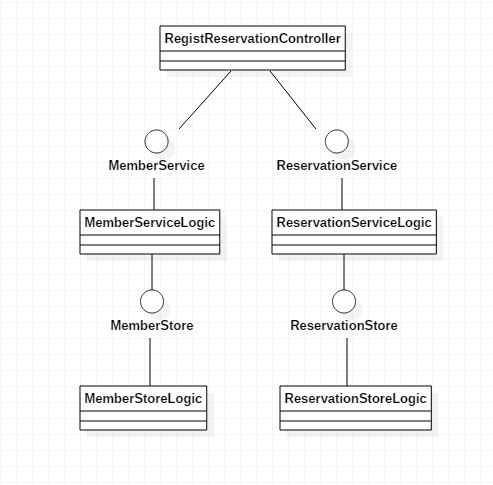
**9.2 공간관리(공간 수정)**

****

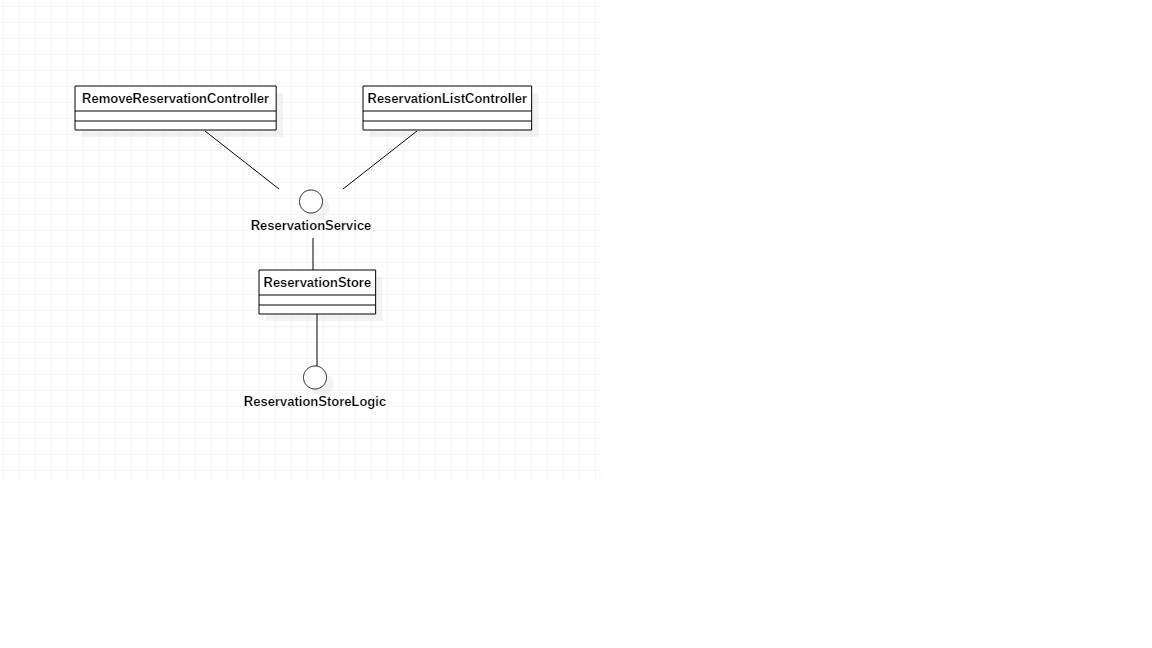
**9.2 공간관리(세부공간 관리)**

****

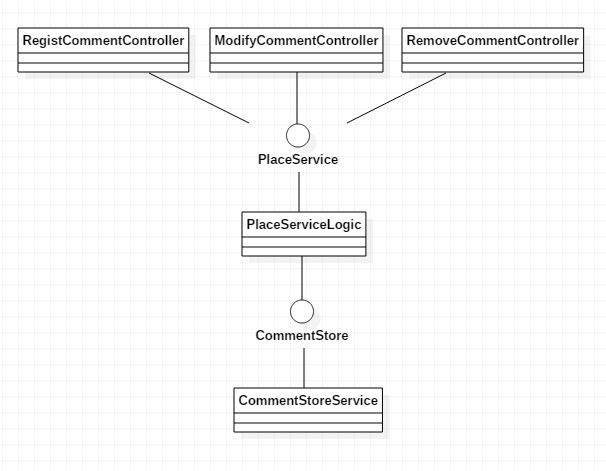
**9.3 예약 관리(예약하기)**

****

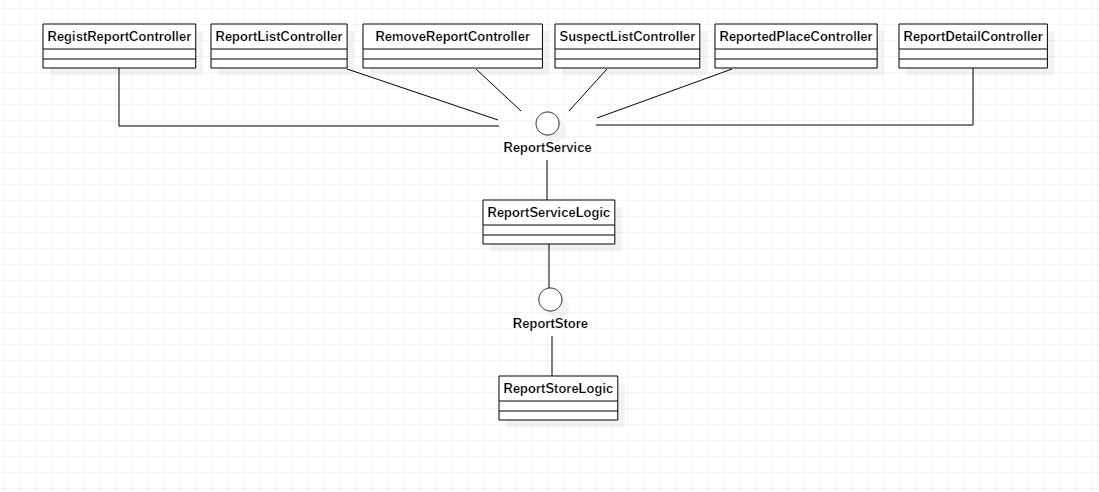
**9.3 예약관리(예약 취소/ 예약 조회)**

****

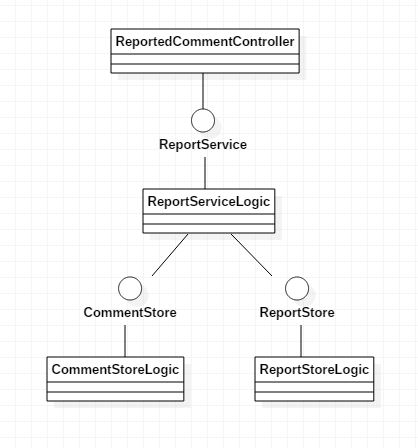
**9.4 후기관리**

****

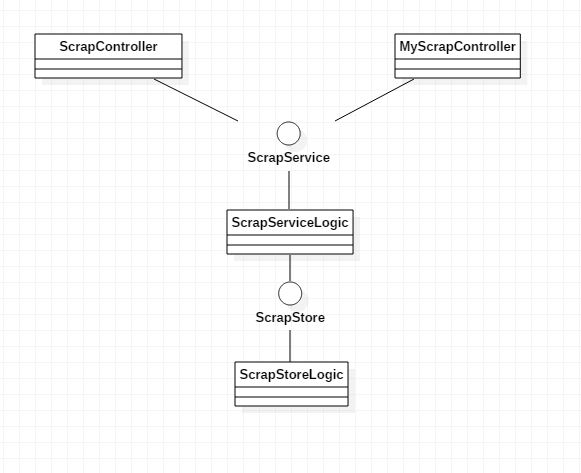
**9.4 신고관리**

****

**9.4 신고관리(신고된 후기 조회)**

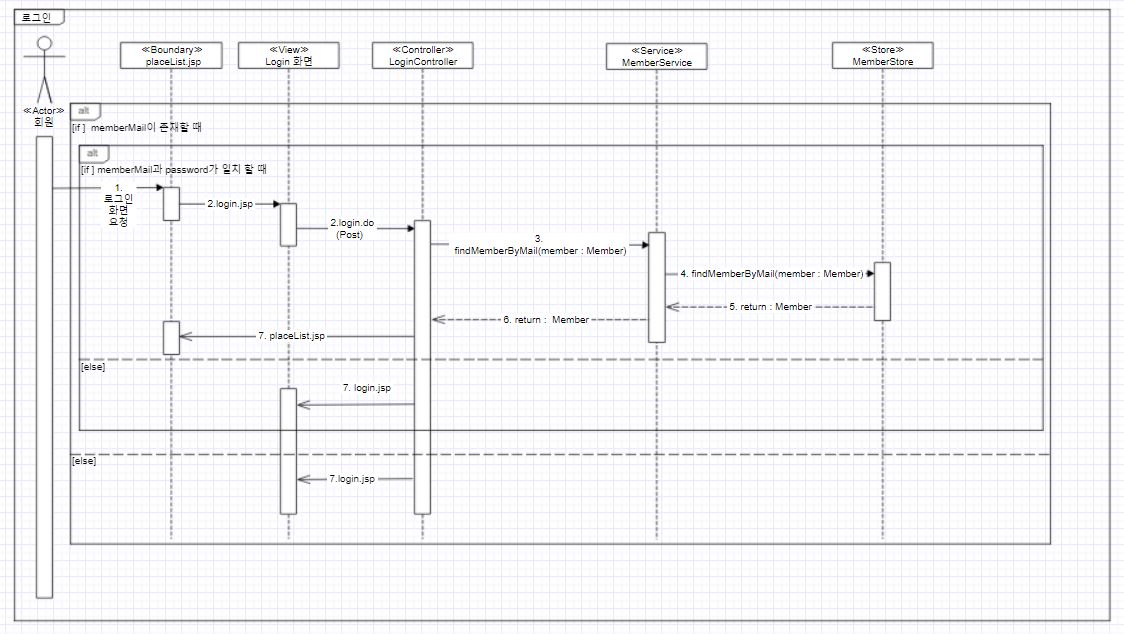
****

**9.5 스크랩 관리**

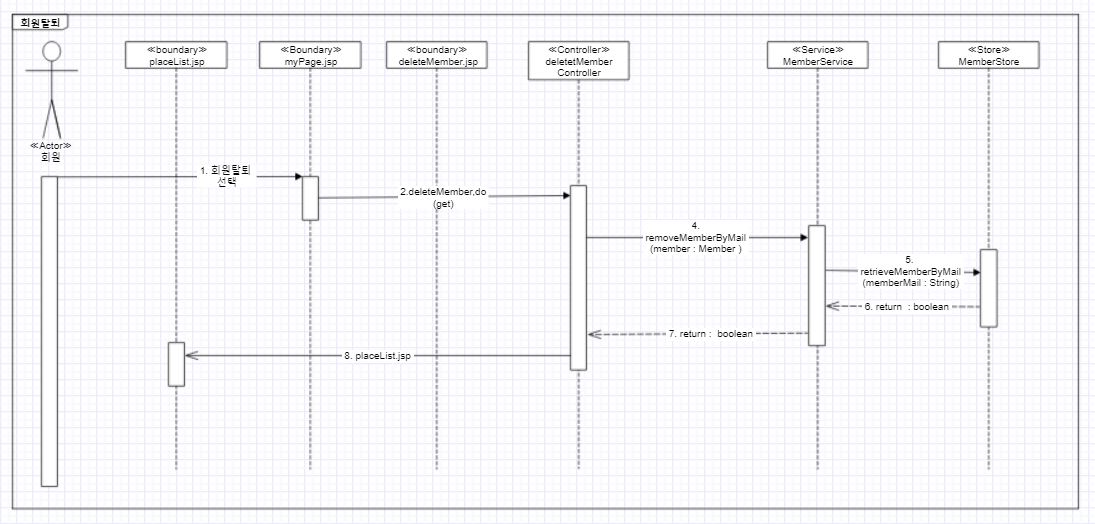
****

**10. Sequence Diagram**

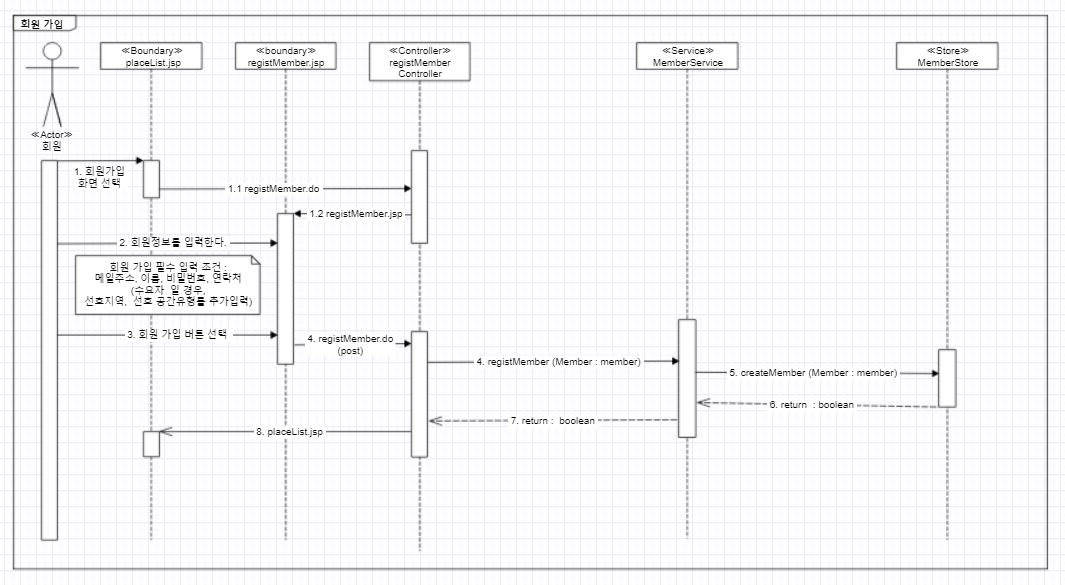
**10.1 로그인**

****

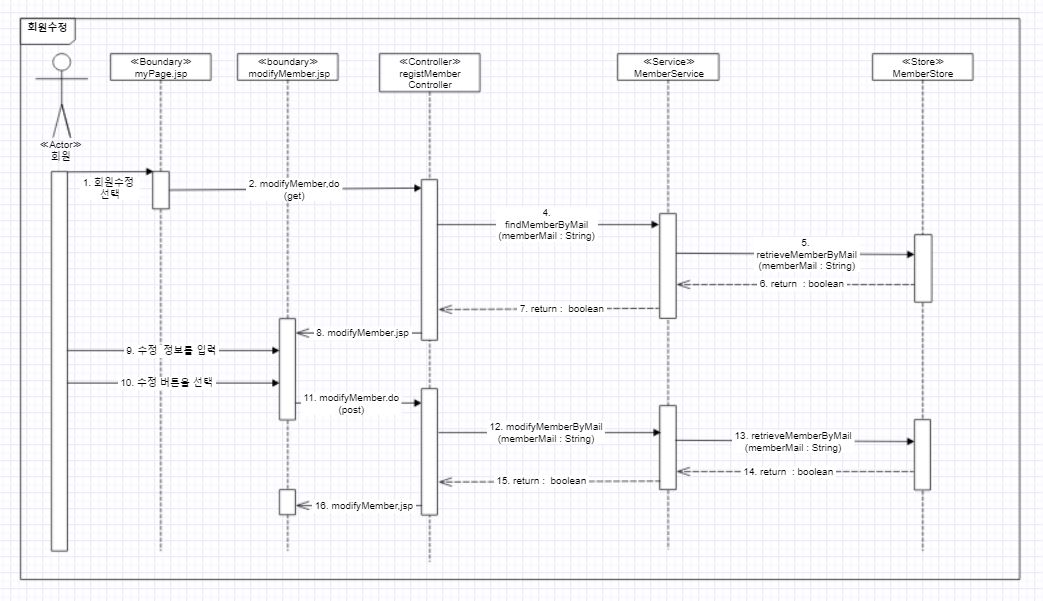
**10.2 개인정보관리(회원탈퇴)**



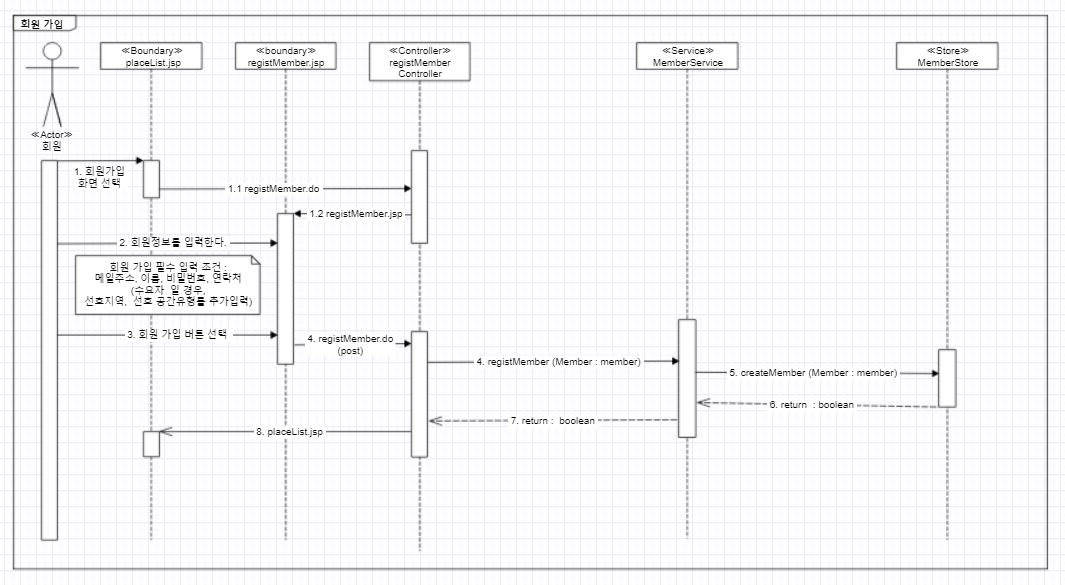
**10.2 개인정보관리(회원가입)**

****

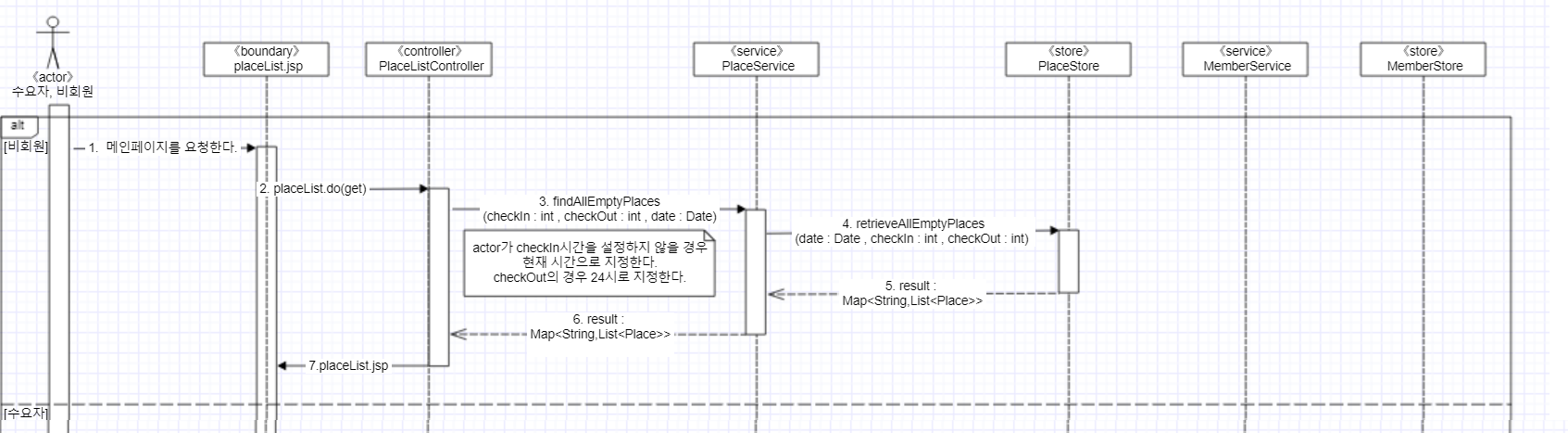
**10.2 개인정보관리(회원정보수정)**

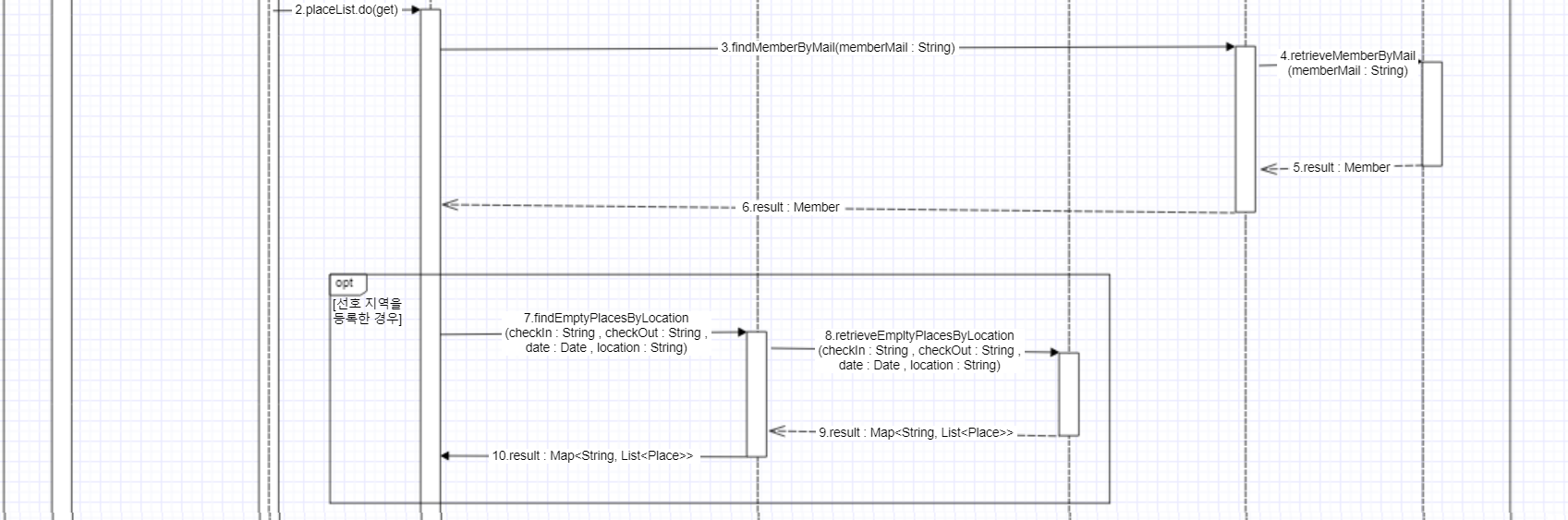


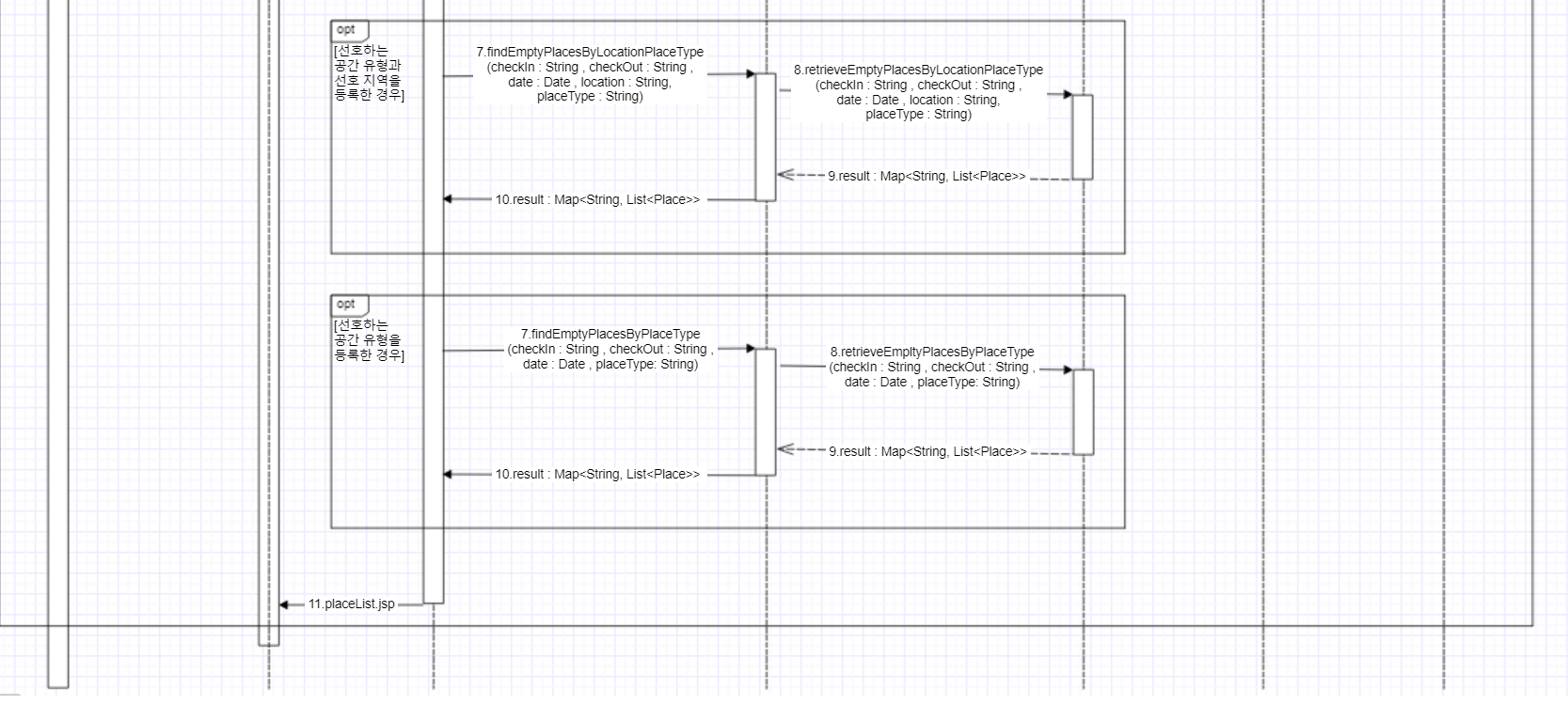
**10.2 회원관리 (회원조회)**

****

**10.3 공간관리(선호하는 공간유형,선호위치)조건 별 전체 공간 조회**

****

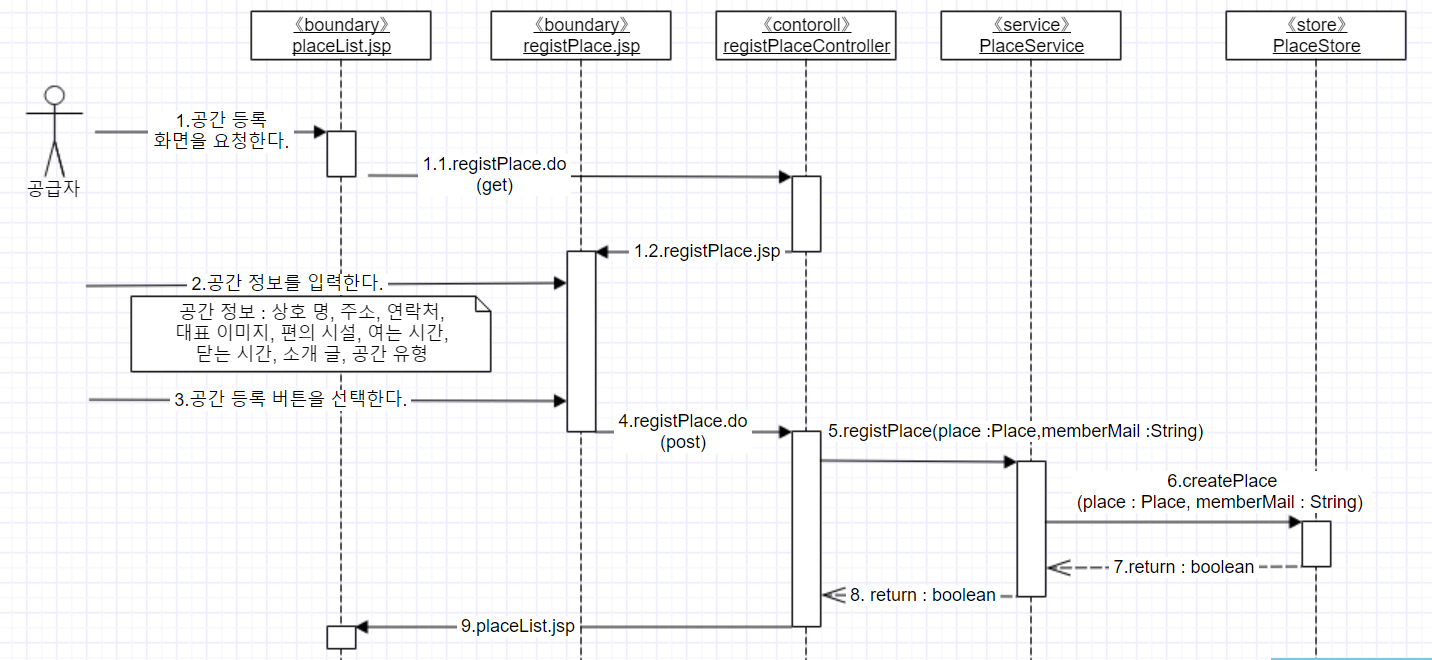
****

****

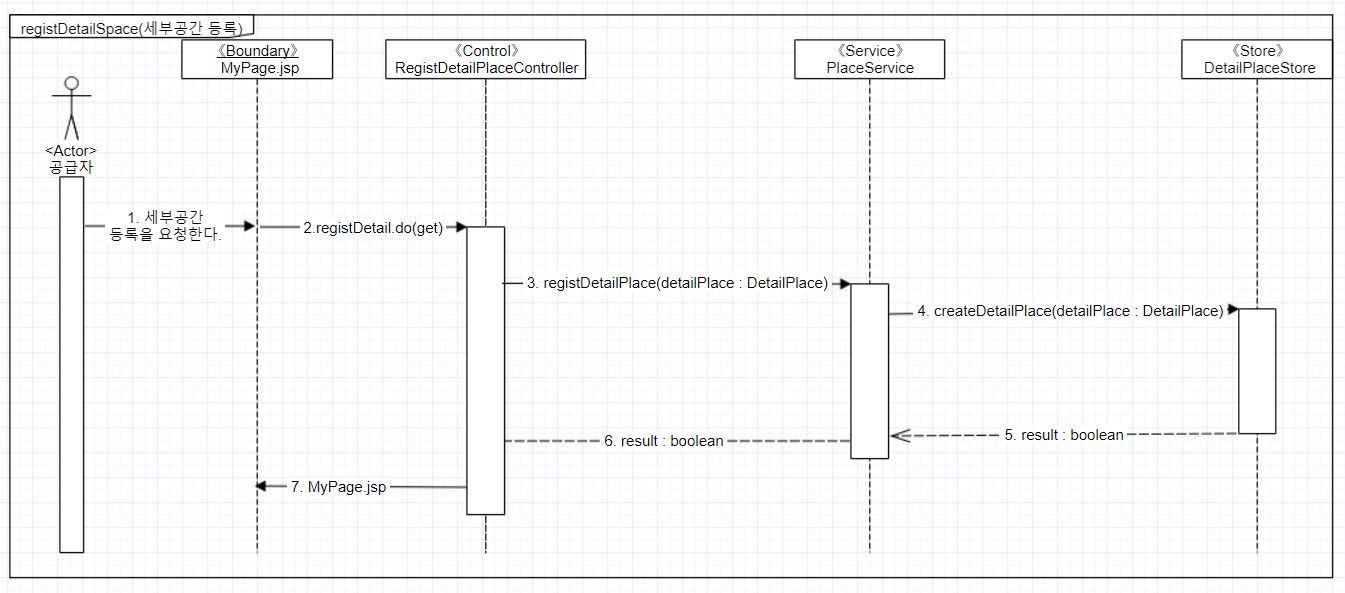
**10.3 공간관리 (특정 공간 조회)**



**10.3 공간관리(공간 등록)**



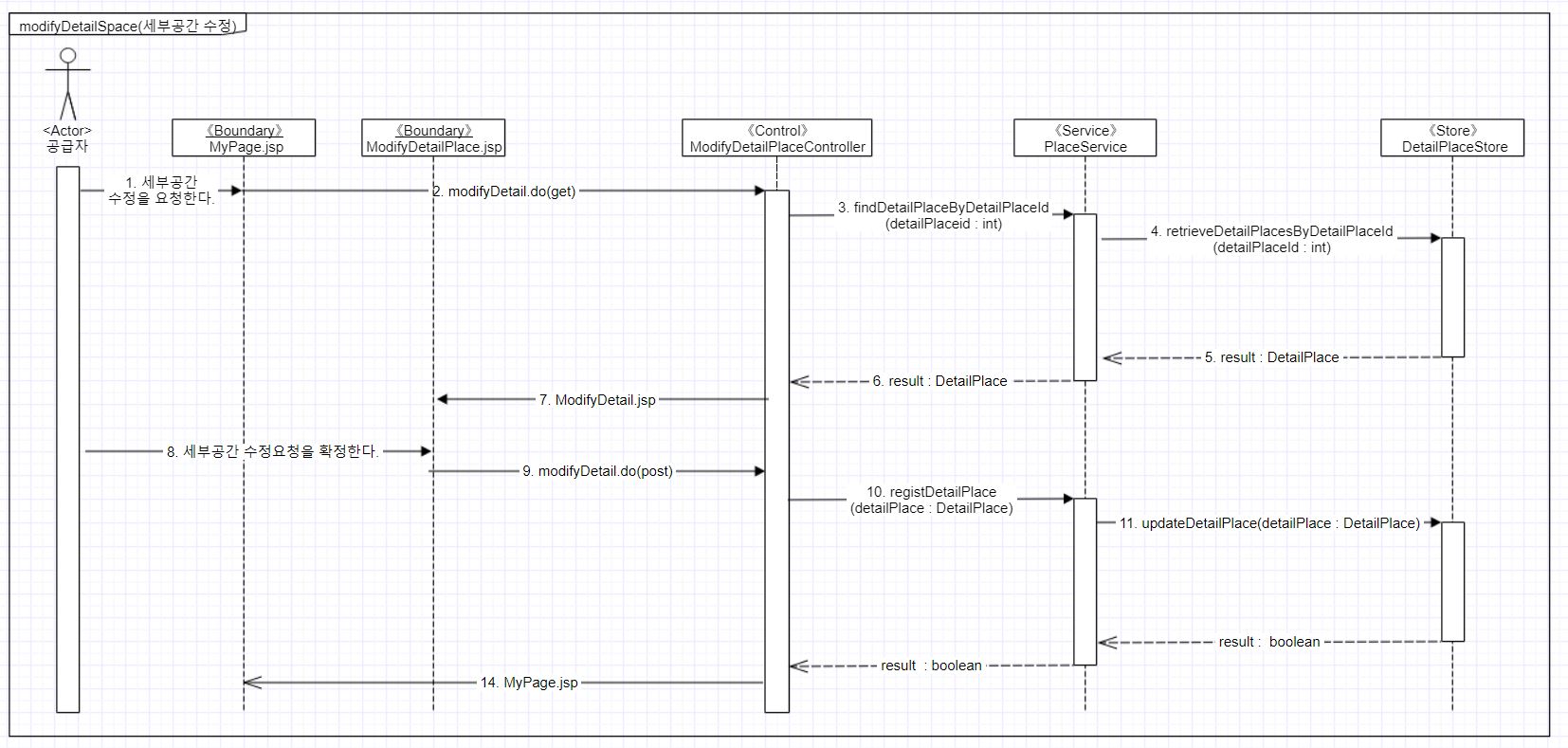
**10.3 공간관리(세부공간 등록)**



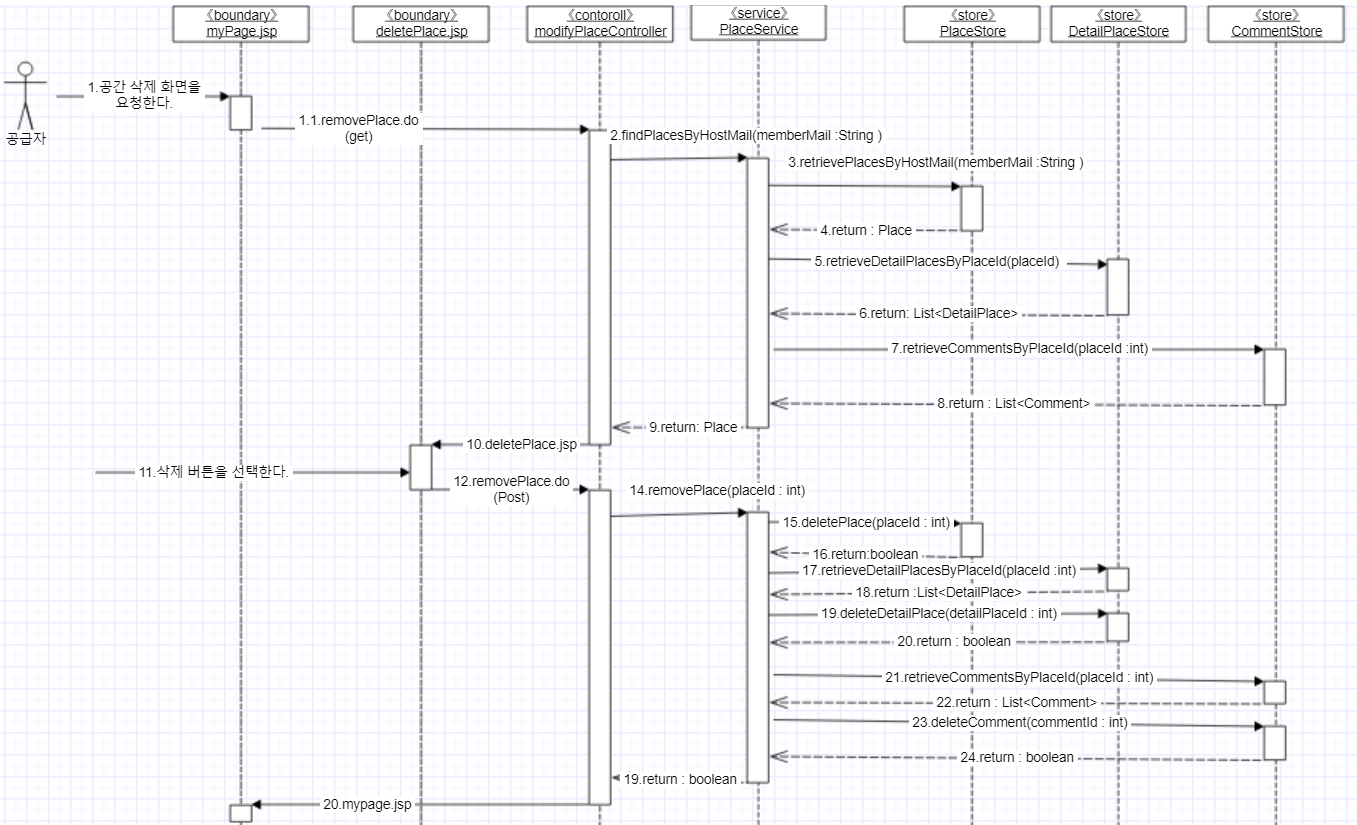
**10.3 공간관리(공간 수정)**



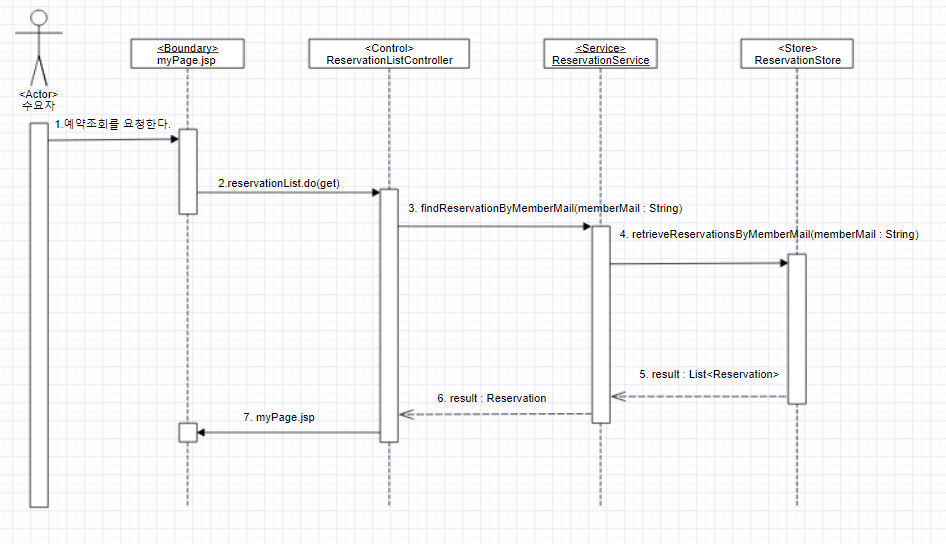
**10.3 공간관리 (세부공간 수정)**



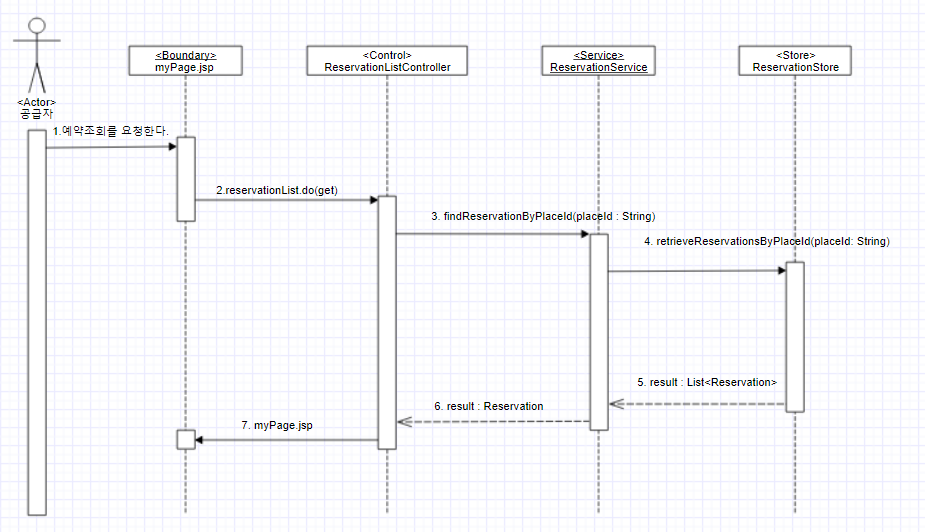
**10.3 공간관리(공간 삭제)**



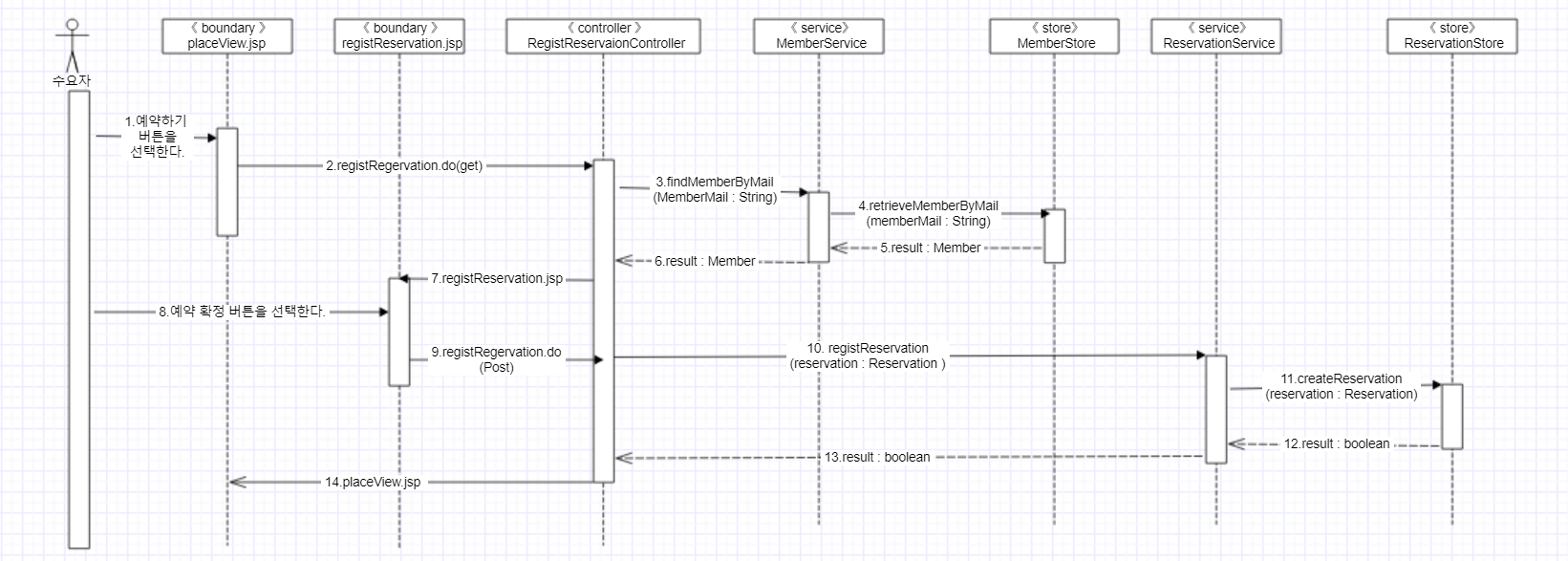
**10.4 예약관리 (수요자일 경우 예약조회)**



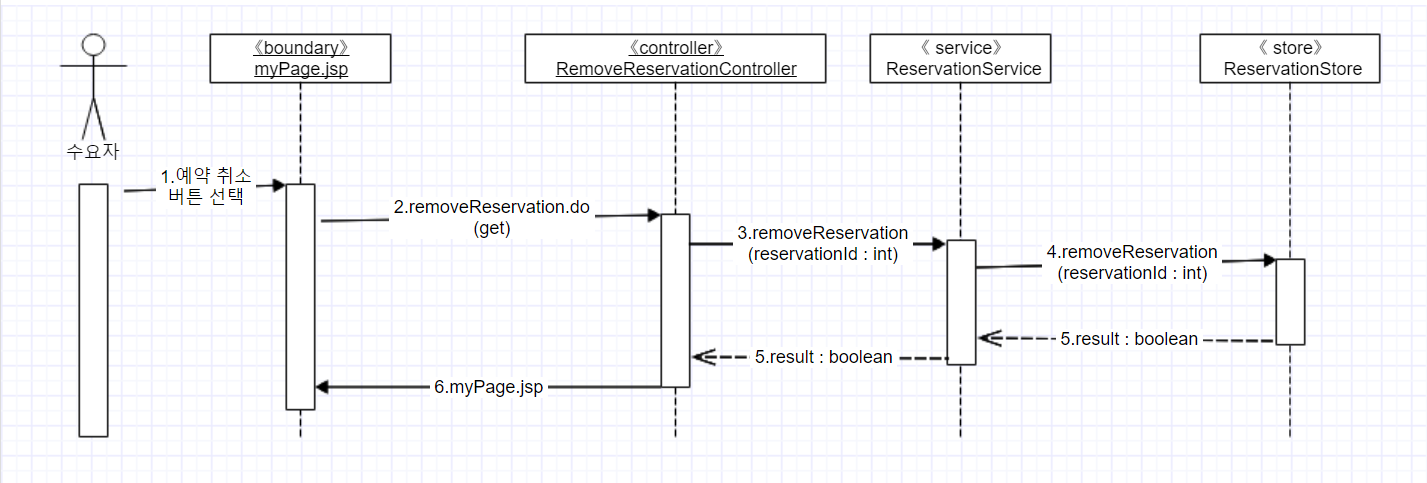
**10.4 예약관리 (공급자일 경우 예약조회)**



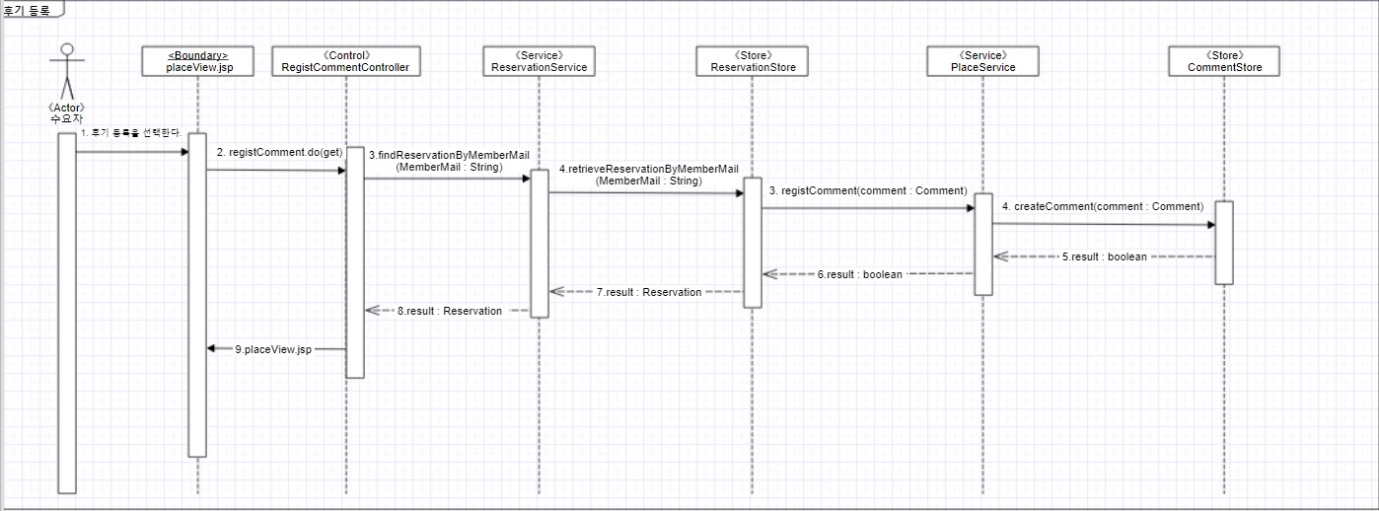
**10.4 예약관리(예약하기)**



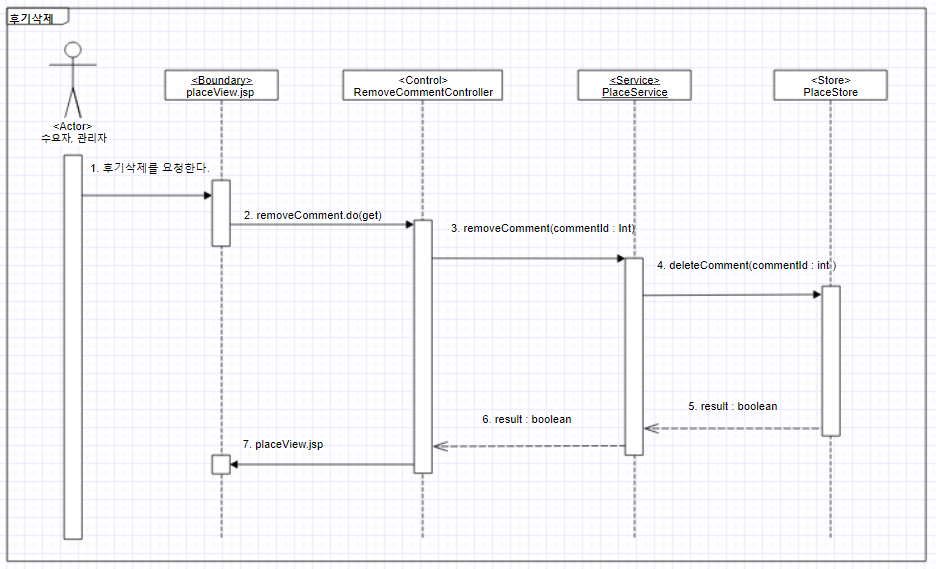
**10.4 예약관리(예약 취소)**



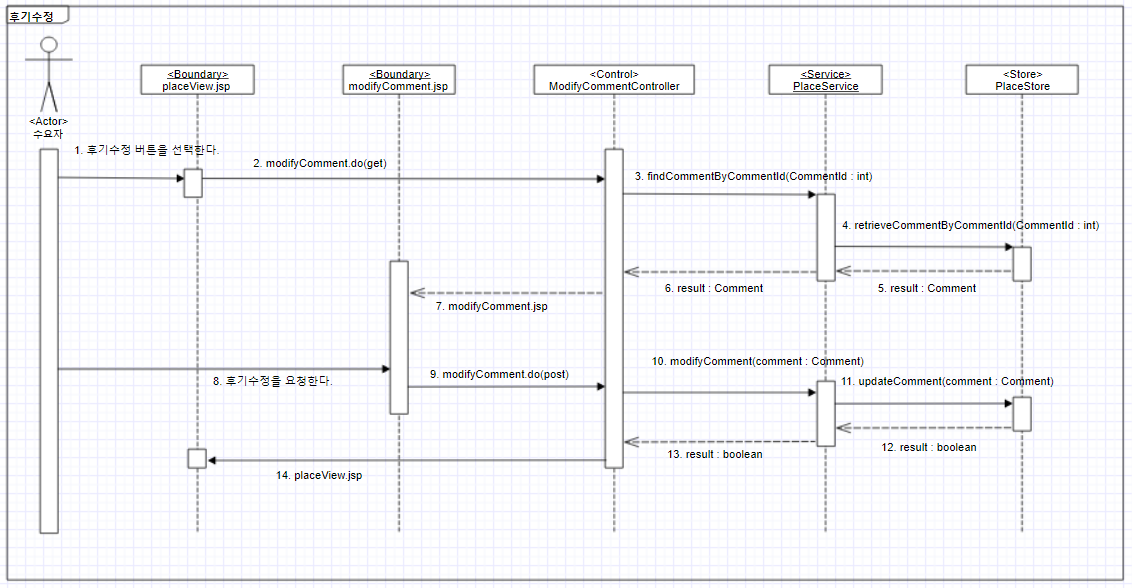
**10.5 후기관리(후기 등록)**

****

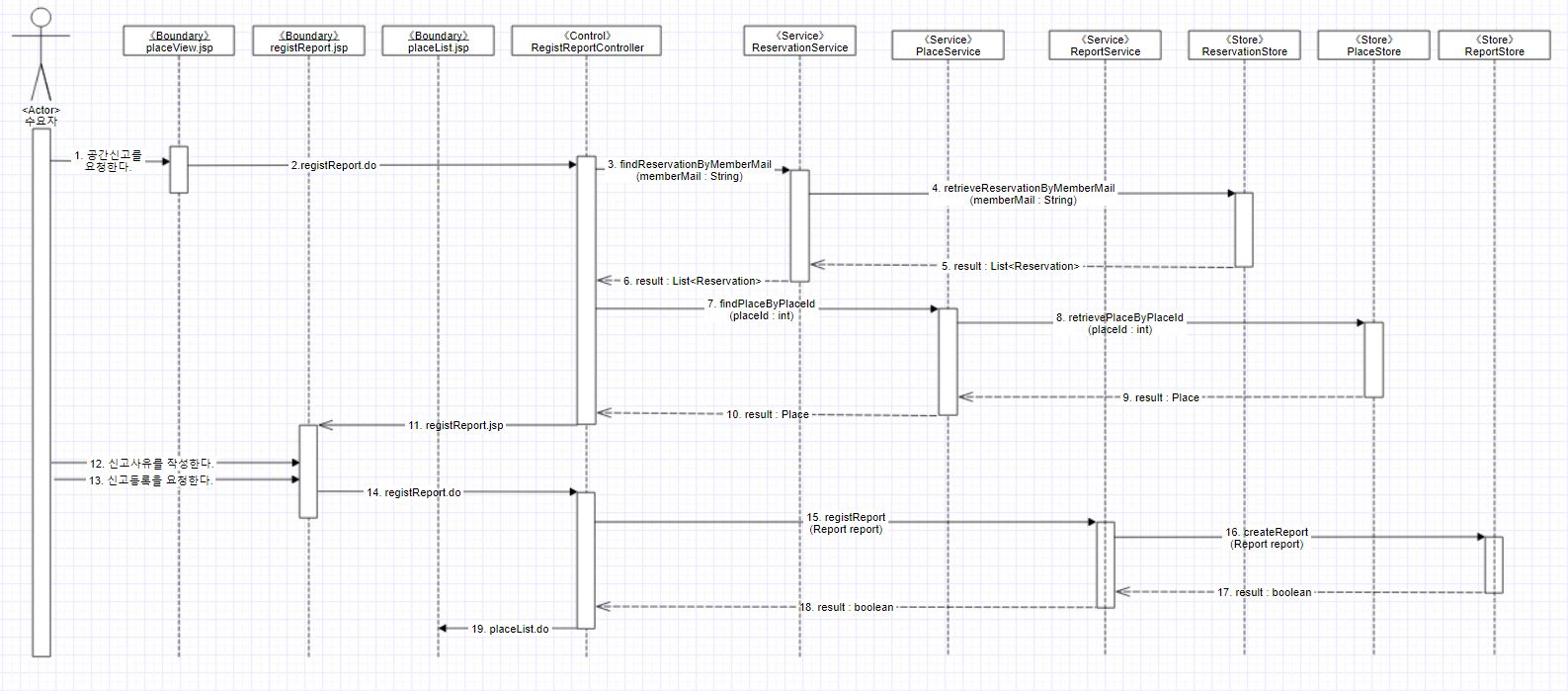
**10.5 후기관리(후기 삭제)**



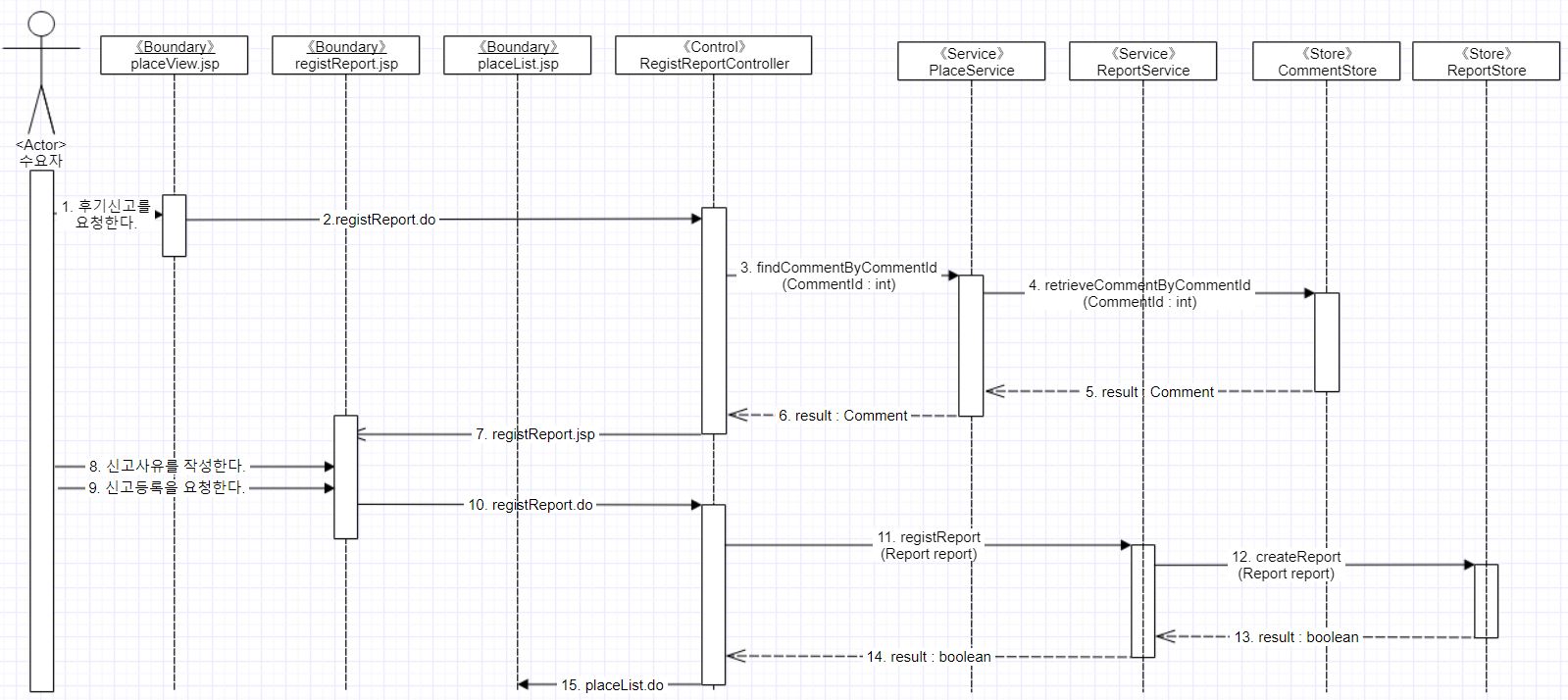
* 1. **후기관리(후기 수정)**



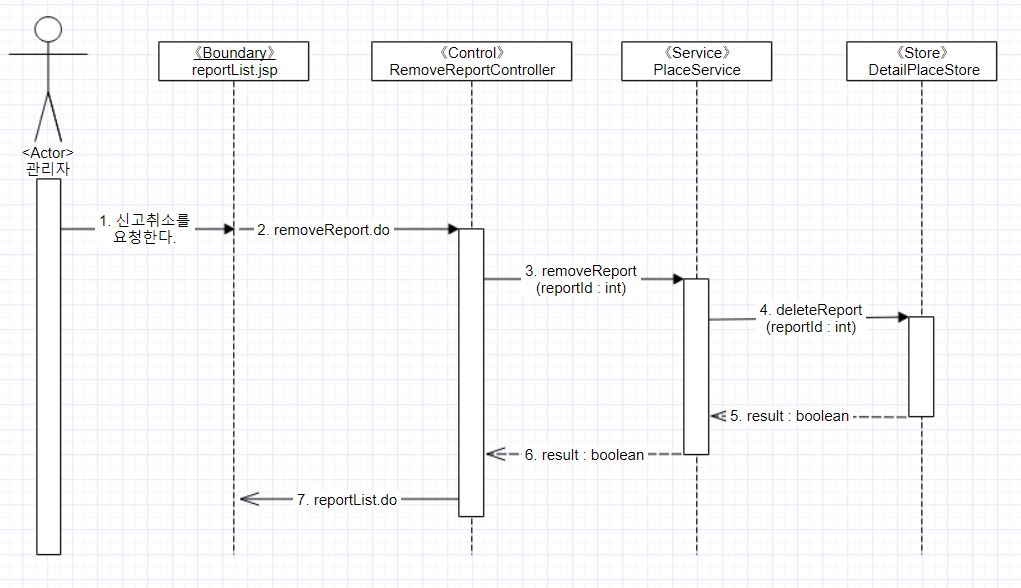
**10.6 신고관리(공간신고)**



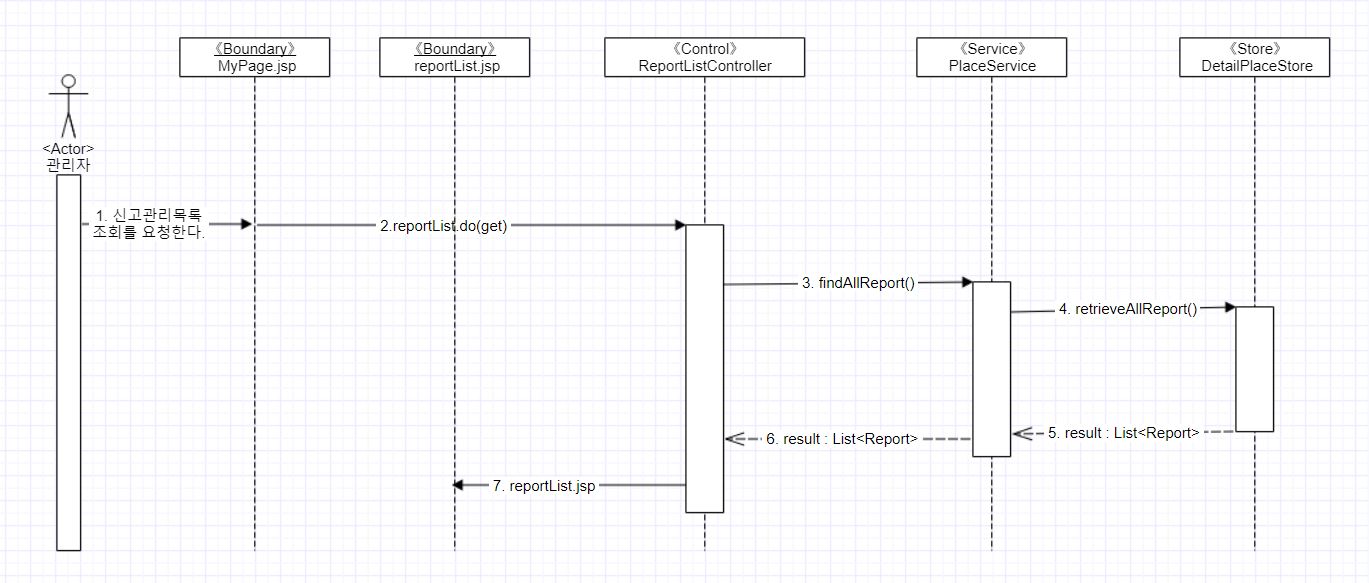
* 1. **신고관리(후기신고)**



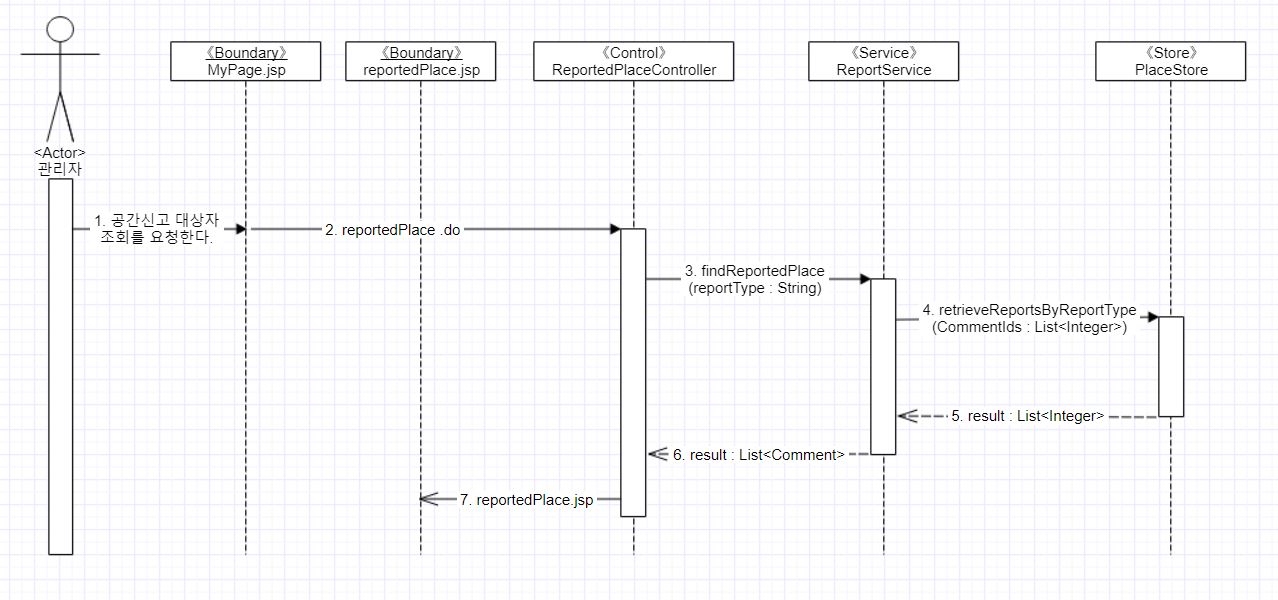
**10.6 신고 관리 (신고 취소)**



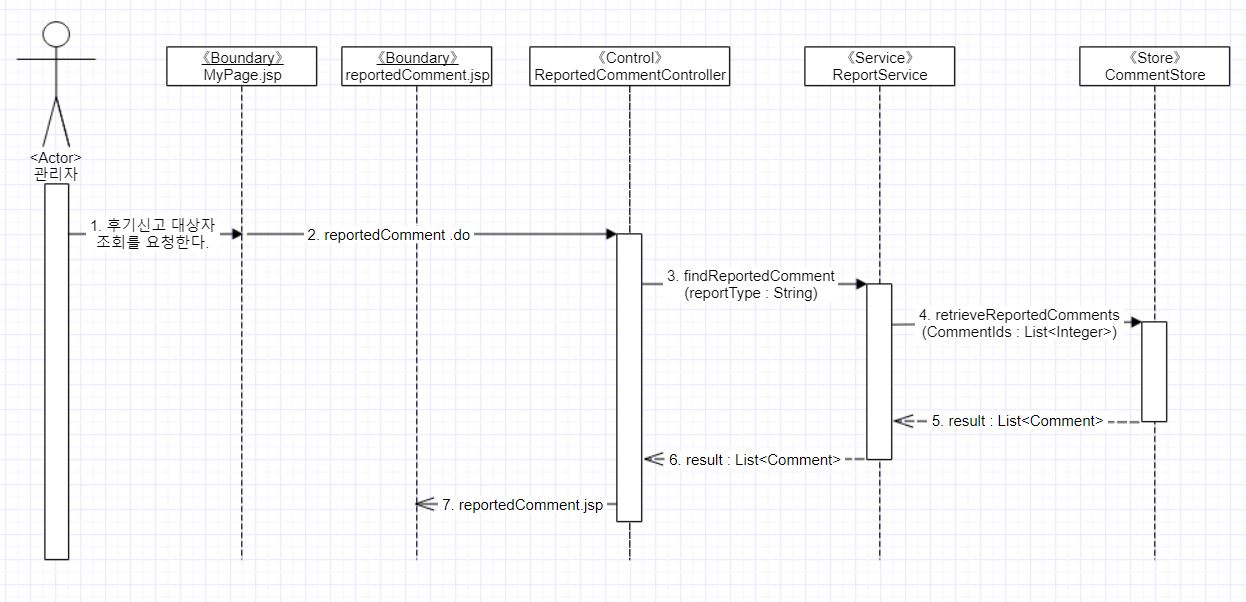
**10.6 신고 관리 (전체 신고 목록 조회)**



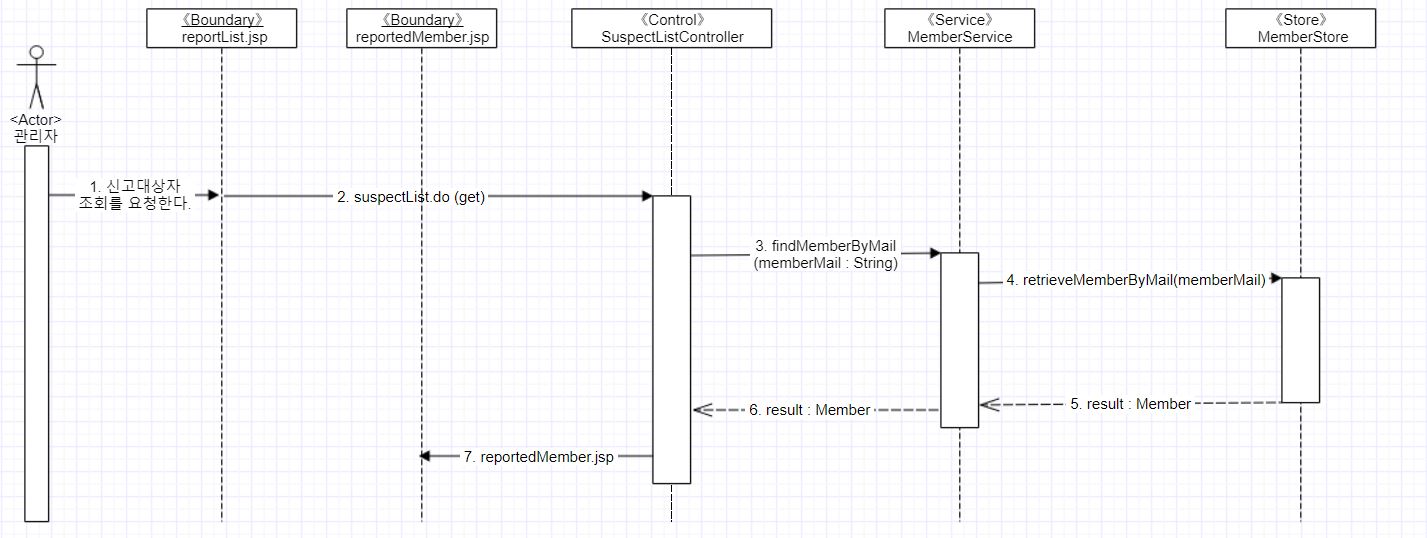
**10.6 신고 관리 ( 공간 신고 목록 조회)**



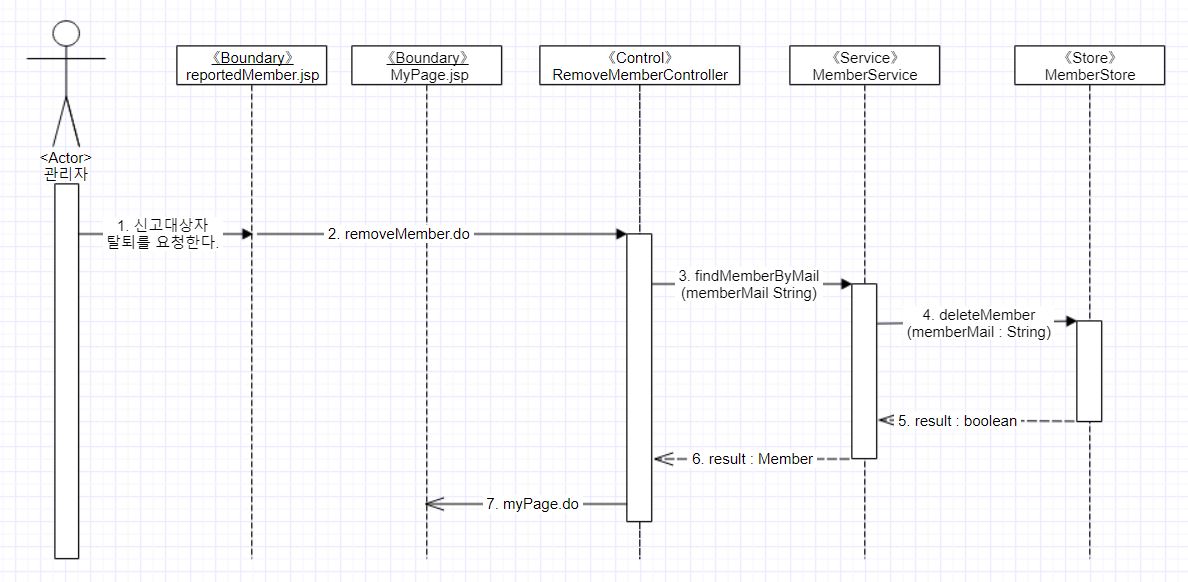
**10.6 신고 관리 (후기 신고 목록 조회)**



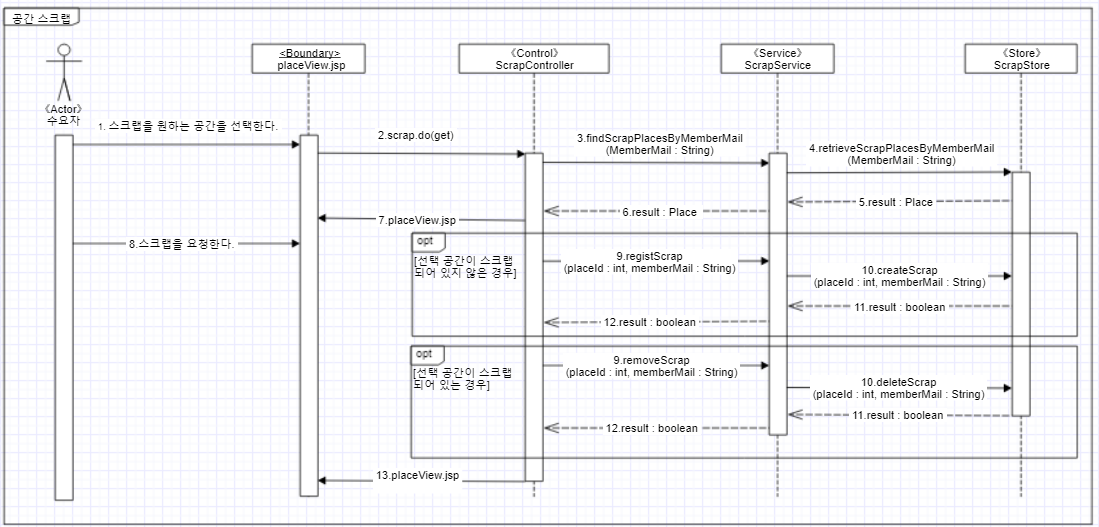
* 1. **신고 관리(신고 대상자 조회)**



* 1. **신고관리 (신고 대상자 탈퇴)**



**10.7스크랩관리**



-시나리오에서 공급자, 수요자, 비회원을 정의 하는 과정에서 의미가 명확하지 않아 수정

-후기신고, 공간신고 usecase가 필요하여 시나리오 수정 후, usecase추가

-각각의 description에 명사를 사용 할 때, 같은 역할 다른 이름으로 작성된 부분 수정

-시스템에서 정보를 표현할 때 조회하는 절차를 누락한 것에 대하여 추가 작성

-시스템에서 정보를 조회하는 절차가 필요하지 않을 경우에 대하여 삭제

-공간조회 usecase 중 검색 조건에서 시간, 예약인원이 필수 조건이었는데 선택조건으로 변경

-공간등록 usecase 중 유효성검사 누락사항 추가

(3. 대표 이미지 형식(확장자가 jpg, png, gif 형식 확인)

(4. 여는 시간과 닫는 시간은 일치 하지 않음

(5. 여는 시간은 닫는 시간보다 우선됨)

-신고조회usecase가 필요하여 검토하던 중, 신고 관리 usecase의 필요성을 느낌

>> 정의했다는 가정 하에 진행

-신고 도메인이 필요하여 usecase description에 명사를 추가 정의 한 후

2차 도메인에 신고 클래스 추가.

-신고 도메인의 필드 명 중 의미가 명확하지 않은 부분 재정의 후 usecase description 수정

(신고 대상에서 신고 항목으로 수정)

-1차,2차 Domain에 스크랩 예약목록 Domain추가

-모든 Table은 Primary Key를 보유해야 하는데

예약, 스크랩, 신고, 세부공간 Table에 Primary Key로 id부여

-세부공간 Table이 공간 id를 primary key 이자 foreign key로 사용하고 있었는데

세부공간id를 primary key로 지정하고 공간 id를 foreign key로 사용

-스크랩 Table에 id 제거

-Member 도메인과 table에 수요자와 공급자를 구분하는 회원구분 (필드/칼럼) 추가

-1차 도메인 작성 중에 후기와 후기들의 차이를 구분

-Place 도메인에 openTime과 closeTime의 타입을 int에서 Date로 변경

-MemberService와 MemberStore에 전체 회원 조회하는 method추가

-ReservationService와 ReservationStore에 공간이름으로 예약목록을 조회하는 method를

공간 id로 조회하도록 변경

-CommentService와 CommentStore에 회원 메일로 후기를 조회하는 method를

후기 id로 조회하도록 변경

-Schdule Table에 scheduleDate칼럼과 00시부터 24시의 예약여부를 판단하는 25개의 칼럼추가

-ScheduleStore를 삭제 후 ReservationStore에 예약 일자로 스케줄을 조회하는method 추가

-MemberService와 MemberStore에 마일리지 변경하는 method추가

-PlaceDetailController에 CommentStore 추가

-공간List 조회 Method 반환Type - List<Place>에서 Map<String, List<Place>>로 변환

-공간검색 Method 중 시간으로, 공간유형으로 검색에서 시간&공간유형으로 검색 추가

-LogoutController 추가

-ReservationStatusController 제거

(ReservationStatusController에 있던 일정 조회 method를 PlaceViewController에 추가)

-DetailPlaceListController 제거

- DetailPlaceStore 추가

-RemovePlaceController에 CommentStore 추가

-RemoveMemberController에 ScrapStore 추가

-ReportDetailController 추가하고 신고정보 출력하는 method 정의

-ReportStore에 신고구분으로 조회 추가

-CommentStore에 신고된 후기 조회 추가

-ReportService에 신고된 공간 조회와 신고된 후기 조회 추가