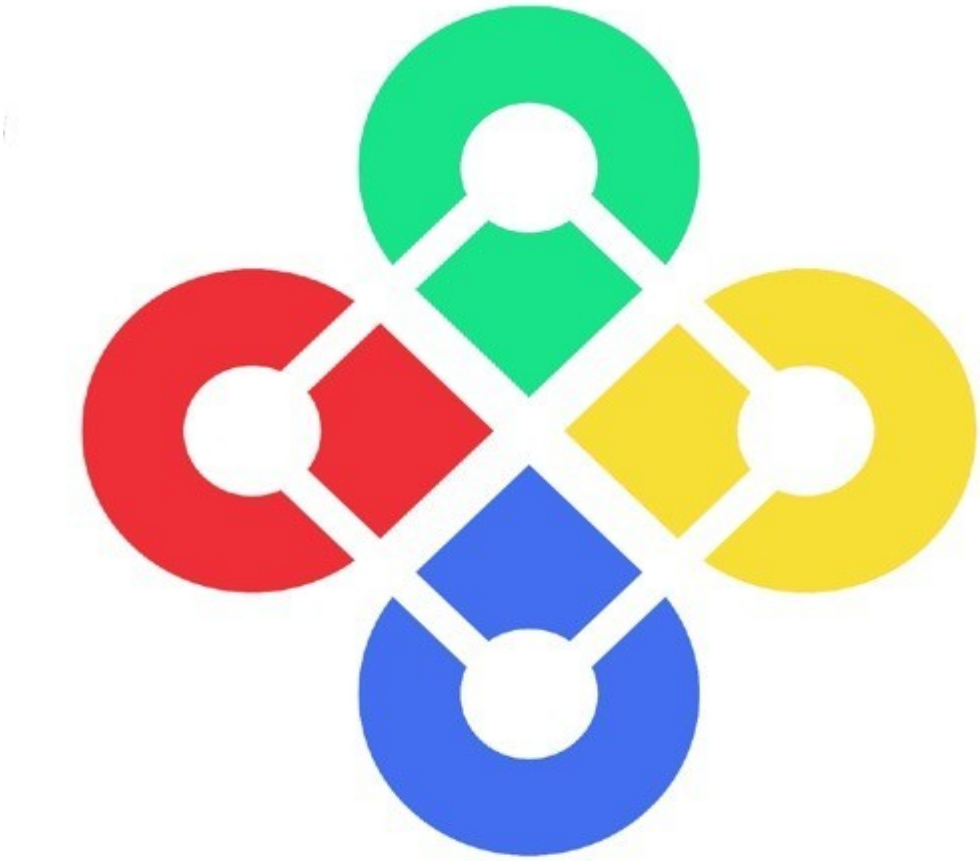
Yeungnam University

# 3. Design

# Billiard Reservation and Payment System



| Student NO | 22112042 |
|---|---|
| Name | 김동욱 |
| E-MAIL | dw_426@naver.com |

# [ Revision history ]

| Revision date | Version # | Description | Author |
|---|---|---|---|
| 2025/03/28 | 1.00 | 초기 버전 | 김동욱 |
| 2025/05/11 | 1.01 | 클래스 관계 설정 | 김동욱 |
| 2025/06/05 | 1.02 | 기능 수정 | 김동욱 |
| | | | |
| | | | |
| | | | |

= Contents =

1. Introduction .....1

2. Class diagram .....2

3. Sequence diagram .....6

4. State machine diagram .....12

5. Implementation requirements .....15

6. Glossary .....17

7. References .....18

## 1. Introduction

예로부터 사람들은 공동의 목적이나 취미를 공유하기 위해 모임을 형성하고, 이러한 만남을 통해 친목을 쌓고 협력하는 문화를 발전시켜왔다. 현대에 들어서는 이와 같은 모임의 형태가 더 다양해지고 전문화되며, 특히 여가와 취미 활동을 기반으로 한 동호회나 커뮤니티가 활발히 운영되고 있다.

그 중에서도 당구는 남녀노소 모두가 즐길 수 있는 대표적인 스포츠로 자리 잡고 있으며, 이에 따라 당구장 예약 수요도 꾸준히 증가하고 있다고 생각을 했다. 그러나 이러한 모임과 예약 활동이 증가하면서 여러 가지 불편한 상황도 함께 발생하고 있다. 예를 들어, 예약 중복 문제, 정보 공유의 비효율성, 관리자의 수기 관리 부담, 예약 내역 분실 등은 사용자와 관리자 모두에게 피로감을 주는 요소이다.

이러한 문제를 해결하고자 우리는 간편하고 직관적인 당구장 예약 시스템을 개발하게 되었다. 본 시스템은 단순한 예약 플랫폼을 넘어, 사용자 중심의 인터페이스, 투명한 예약 내역 관리, 관리자의 효율적인 전체 관리 기능 등을 통합적으로 제공한다.

첫 번째 목표는 누구나 쉽게 접근할 수 있는 예약 환경을 만드는 것이다. 사용자는 회원가입을 통해 이메일 기반으로 로그인하고, 원하는 날짜, 시간, 당구장 이름, 위치, 당구대 수 등을 입력하여 간편하게 예약을 진행할 수 있도록 서비스를 구축하도록 노력했다.

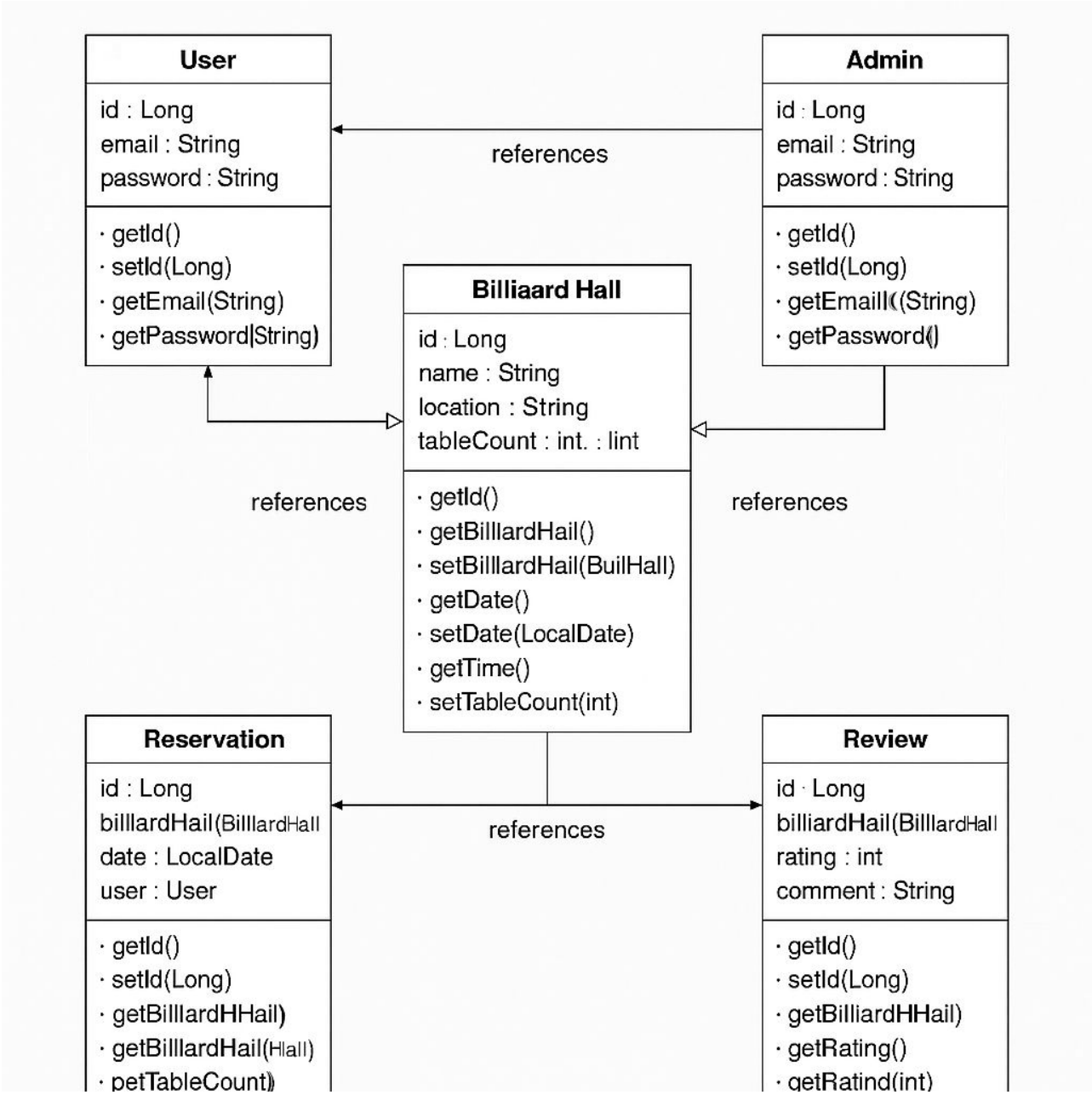
두 번째 목표는 예약 정보의 투명한 관리이다. 사용자는 자신이 예약한 내역을 확인하거나, 불필요한 예약을 직접 취소할 수 있으며, 모든 데이터는 서버에 안전하게 저장된다. 관리자는 전용 계정을 로그인하여 전체 예약 현황을 실시간으로 확인할 수 있고, 필요시 예약 내역을 삭제하거나 관리할 수 있는 기능을 추가하였다.

세 번째 목표는 관리자의 부담을 줄이고 사용성을 향상시키는 것이다. 일반 사용자는 예약과 확인, 취소 중심의 흐름을 따르고, 관리자는 대시보드 페이지를 통해 시스템을 한 눈에 파악할 수 있게 구성했다. 또한 로그인 상태에 따라 기능을 제한하고, 보안 및 사용자 경험을 고려한 세션 기반 인증 방식 또한 적용하도록 노력했다.

이 시스템은 단순 예약을 넘어, 모임 관리와 예약 편의성 향상, 운영자 관리 효율성 극대화, 사용자 경험 중심의 UI/UX 구현을 통해 중소 규모의 다양한 커뮤니티와 운영자들에게 실질적인 도움을 줄 수 있도록 설계되었다.

Design 문서에서는 위 시스템의 설계 원리와 구현 목표를 기반으로, 후속 단계인 구현을 위한 구체적인 설계 지침과 구조를 설명하고자 한다. 실제 소스코드와 매칭되는 내용을 기반으로 하여, 모두가 실무에 적용 가능한 시스템 구현을 할 수 있도록 돕는 것을 목표로 한다.

2. Class diagram




아래의 표를 바탕으로 Class Diagram에서 표현한 Class들의 대한 설명을 나타냈다.

| Class | Explanation |
|---|---|
| UserController | 사용자(일반회원)의 회원가입, 로그인, 로그아웃, 메인 페이지 이동을 담당한다.<br>- 로그인 시 이메일/비밀번호 확인<br>- 세션을 이용해 사용자 정보를 유지<br>- 로그인 성공 시 메인 페이지로 이동 |
| RegisterController | 별도의 회원가입 전용 페이지 (/register)에서 사용자를 등록할 수 있게 해주는 컨트롤러이다. |
| AdminController | 관리자 전용 로그인 기능과 대시보드 접근 권한을 관리한다.<br>- 관리자 ID : host, 비밀번호 : 1234 로 고정<br>- 로그인 성공 시 전체 예약 목록 확인이 가능 |
| AdminReservationController | 관리자가 모든 예약 내역을 확인할 수 있는 기능을 제공한다. |
| ReservationController | 사용자의 예약 등록, 예약 완료 처리, 예약 내역 보기, 예약 취소 등을 담당하는 핵심 컨트롤러이다.<br>- 예약 성공 시 success.html로 이동<br>- 예약 목록 조회 및 취소 기능 |
| UseService/ReservationService | 각각 사용자 및 예약에 대한 비즈니스 로직을 처리하는 서비스 계층이다 |
| User/Reservation | 데이터베이스에 저장되는 사용자 정보와 예약 정보를 정의한 클래스들이다.<br>- User는 이메일, 이름 비밀번호 등을 가짐<br>- Reservation은 당구장 이름, 위치, 시간 예약자 이메일 정보를 포함한다. |
| UserRepository/ReservationRepository | Spring Data JPA를 활용해서 DB와 연결해주는 역할을 한다.<br>- 사용자 정보나 예약 정보를 쉽게 조회/저장할 수 있게 해준다. |

- 로그인 기반 접근 제어 : 로그인한 사용자만 예약이 가능하다.
- 관리자 기능 포함 : 전체 예약 확인, 삭제가 가능하다.
- Thymeleaf 템플릿 엔진으로 UI 구성

- H2 DB 기본 사용, MySQL 확장 가능
- 세션 기반 사용자 정보 유지가 가능하다.

1. 사용자 (User) 기능
- 회원가입 및 로그인 기능
  - 사용자는 이메일, 이름, 비밀번호 등의 정보를 입력하여 회원가입을 진행한다.
  - 로그인 시 세션 기반 인증을 통해 사용자 정보를 유지하며, 로그인 된 사용자만 예약 기능을 사용할 수 있도록 제한되어 있다.
  - 아이디 중복 검사를 통해 데이터 무결성을 유지한다.
- 로그아웃 기능
  - 세션을 무효화하여 로그인 상태를 종료한다.
  - 로그아웃 이후에는 보호된 URL 접근이 불가능하며, 로그인 페이지로 자동 리다이렉션된다.
- 당구장 예약 기능
  - 사용자는 당구장 이름, 위치, 예약 날짜, 예약 시간, 사용하고자 하는 당구대 수를 입력하여 예약을 완료한다.
  - 예약은 ReservationController를 통해 처리되며, 완료 후 success.html 페이지에서 결과를 확인할 수 있다.
  - 예약 시 로그인 한 사용자 정보를 기반으로 자동으로 이메일이 저장된다.
- 예약 목록 확인 및 취소
  - 로그인한 사용자는 /my-reservations 페이지에서 자신의 예약 현황을 확인할 수 있다.
  - 각 예약 항목 옆에는 취소 버튼이 제공되며, 이를 통해 본인 예약을 삭제할 수 있다.
  - 예약 취소는 실제 DB에서 해당 예약 항목을 삭제하는 방식으로 구현된다.
- 리뷰 및 평가 기능
  - 사용자는 예약 후 당구장에 대한 후기를 남길 수 있다.
  - 후기에는 별점과 텍스트 리뷰가 포함되며, 관리자와 다른 사용자들에게 유용한 피드백을 제공한다.
  - 리뷰는 ReviewController, ReviewRepository, Review 엔티티 등을 통해 구현된다.

2. 관리자 기능
- 관리자 로그인
  - 관리자는 고정된 ID (host) 와 비밀번호 (1234)로 로그인한다.

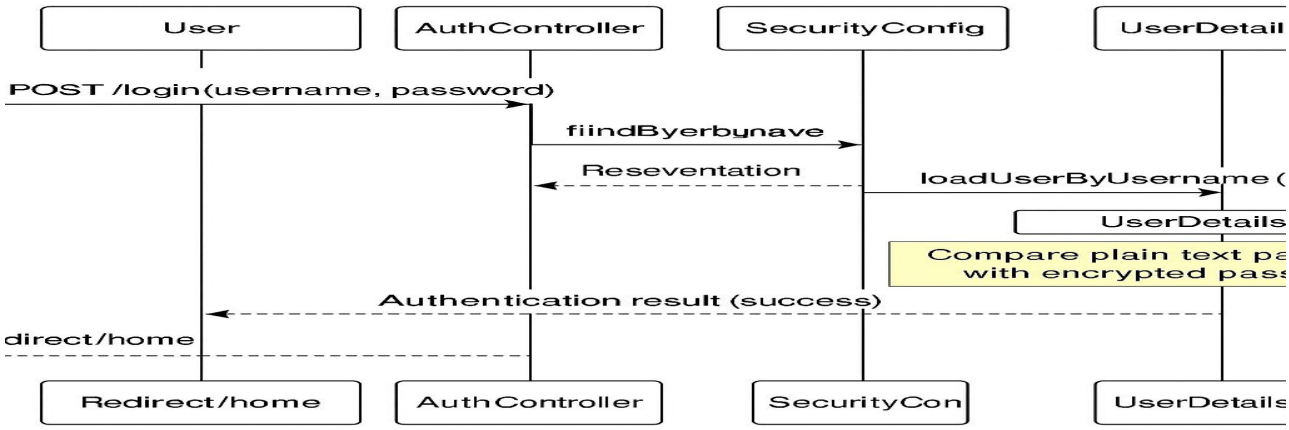
- 관리자 세션은 일반 사용자와 구분되어 isAdmin 속성을 통해 처리된다.

- 전체 예약 내역 확인
  - /admin/dashboard를 통해 현재까지 모든 예약 내역을 열람할 수 있다.
  - 예약자는 누구인지, 어느 당구장에서 언제 예약했는지 등의 정보를 모두 열람 가능하다.
- 관리자용 예약 취소 기능
  - 관리자 역시 모든 예약에 대해 삭제할 수 있으며, 예약자가 누구든 무관하게 처리 가능하다.
- 회원 리뷰 모니터링 기능
  - 작성된 리뷰를 열람하고 필요 시 삭제하거나 블라인드 처리할 수 있다.
  - 욕설, 부적절한 표현 등이 포함된 리뷰에 대한 관리 기능도 향후 추가 가능성 존재

3. 공통 기능 및 구조적 특징
- 세션 기반 인증 처리
  - 로그인 후 사용자 정보는 HttpSession에 저장되며, 컨트롤러에서는 세션에 등록된 사용자 정보를 통해 접근 제어를 수행한다.
- 템플릿 기반 화면 구성
  - 화면 구성은 모두 Thymleaf 리반으로 구현되어 있으며, 동적인 데이터 렌더링과 URL 경로 제어가 쉬운 구조이다.
  - 사용자 및 관리자 페이지는 역할에 따라 스타일을 분리하고, 공통적인 스타일은 style.css로 관리된다.
- 데이터베이스 연동
  - 현재는 H2 메모리 데이터베이스를 사용하고 있으며, 설정 파일을 통해 MySQL 등의 RDBMS로 쉽게 확장 가능하도록 설계되었다.
  - UserRepository, ReservationRepository는 Spring Data JPA를 활용하여 최소한의 코드로 데이터 접근을 수행한다.

4. 향후 확장 계획
- 비밀번호 암호화 및 보안 강화
  - 현재는 평문 비밀번호 저장이지만, 추후 BCrypt를 통해 해싱 저장으로 보안을 강화할 예정이다.
- 예약 제한 로직 도입
  - 동일 시간대의 중복 예약을 막기 위한 논리적 검사 추가 예정
- 관리자 기능 고도화
  - 통계, 예약 분석, 사용자 활동 분석 등의 대시보드 기능도 추가 가능성 있음

3. Sequence diagram



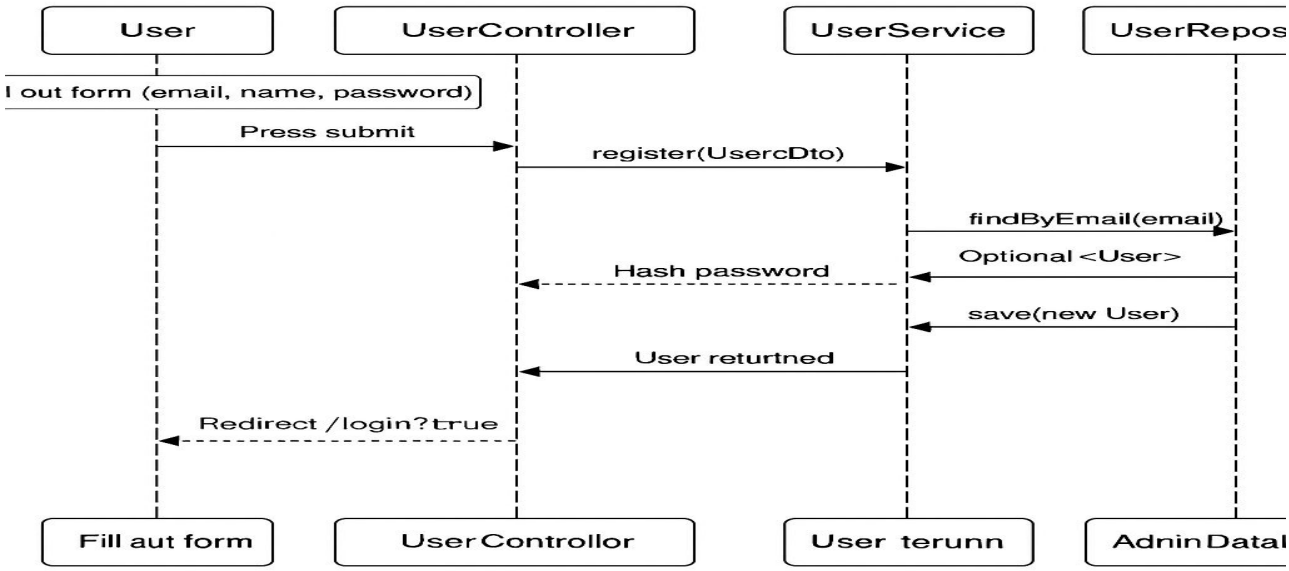

1. 로그인 기능

당구장 예약 시스템에서 사용자는 웹 브라우저를 통해 로그인 화면에 접속하여 자신의 이메일과 비밀번호를 입력한 뒤, 로그인 버튼을 클릭한다. 이때 클라이언트는 해당 정보를 POST /login 경로로 전송하며, 서버 측에서는 이 요청을 수신하여 로그인 절차를 진행한다.

로그인 처리는 Spring Framework의 UserController 클래스에서 담당하며, 내부적으로는 @PostMapping("/login") 메서드를 통해 구현되어 있다. 이 메서드는 먼저 사용자가 입력한 이메일을 기준으로 데이터베이스에서 사용자 정보를 조회한다. 이를 위해 UserRepository의 findByEmail(email) 메서드를 호출하며, 결과로 Optional<User> 객체를 반환받는다. 해당 이메일이 존재할 경우, 반환된 사용자 객체에서 저장된 비밀번호와 사용자가 입력한 비밀번호를 비교하여 일치 여부를 확인한다. 현재 시스템에서는 평문(Plain Text) 형태로 비교가 이루어지며, 향후에는 보안 강화를 위해 비밀번호를 암호화하여 비교할 계획이다.

비밀번호가 일치할 경우 서버는 해당 사용자 정보를 세션(HttpSession)에 저장하여 로그인 상태를 유지한다. 이후 사용자는 메인 페이지(/main)로 리다이렉트되며, 로그인된 사용자만 접근 가능한 기능(예: 예약, 내 예약 확인 등)을 사용할 수 있다. 반면, 비밀번호가 일치하지 않거나 해당 이메일이 존재하지 않는 경우에는 로그인에 실패하며, 오류 메시지를 모델에 담아 로그인 페이지로 다시 이동하게 된다.

추가적으로, 시스템에는 고정된 관리자 계정이 존재하며, 이메일에 "host", 비밀번호에 "1234"를 입력하면 일반 사용자 대신 관리자 권한으로 로그인된다. 이 경우 세션에는 isAdmin = true 속성이 저장되며, 로그인 후 /admin/dashboard 페이지로 이동하게 된다.



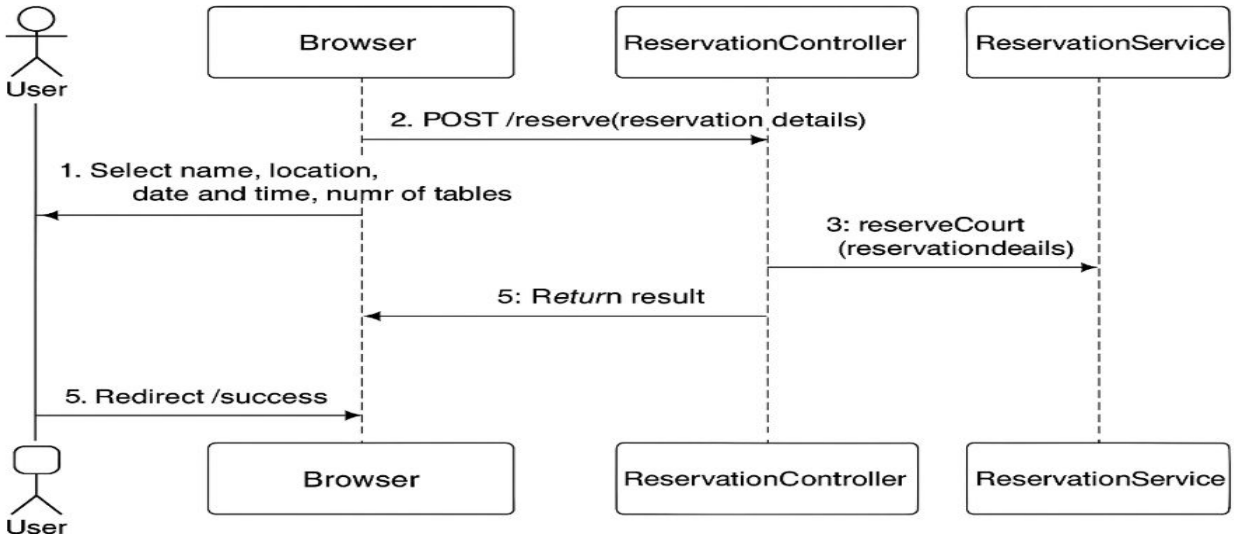

2. 회원가입 기능

회원가입 기능은 사용자가 당구장 예약 시스템에 최초로 접근하여 계정을 생성하는 과정을 포함한다. 사용자는 회원가입 페이지에서 이름, 이메일, 비밀번호와 같은 필수 정보를 입력한 후, "회원가입" 버튼을 눌러 서버로 POST 요청을 보낸다. 이 요청은 /signup 경로를 통해 처리되며, 이를 담당하는 컨트롤러는 UserController 클래스의 signup() 메서드이다.

컨트롤러는 가장 먼저 사용자가 입력한 이메일이 이미 등록된 이메일인지 확인하기 위해 UserRepository.findByEmail(email) 메서드를 호출한다. 이 메서드는 내부적으로 데이터베이스를 조회하며, 동일한 이메일이 존재할 경우 Optional 객체로 감싸서 반환된다. 만약 중복된 이메일이 존재한다면, 컨트롤러는 사용자에게 “이미 가입된 이메일입니다”라는 오류 메시지를 전달하고 다시 회원가입 화면으로 리턴시킨다.

반면, 중복된 이메일이 없는 경우에는 사용자가 입력한 정보를 바탕으로 새로운 User 객체를 생성하고, UserRepository.save(user)를 호출하여 데이터베이스에 저장한다. 비밀번호는 현재 평문으로 저장되지만, 향후 보안 강화를 위해 암호화 방식으로 전환될 예정이다.

회원가입이 완료되면 사용자는 로그인 페이지로 리다이렉트되며, 로그인 후 당구장 예약 시스템의 주요 기능을 사용할 수 있다. 이 모든 과정은 서버 측에서 세션 기반의 인증 처리와 함께 이루어지므로, 회원가입 이후 로그인하지 않은 상태로는 예약 등의 기능에 접근할 수 없다.



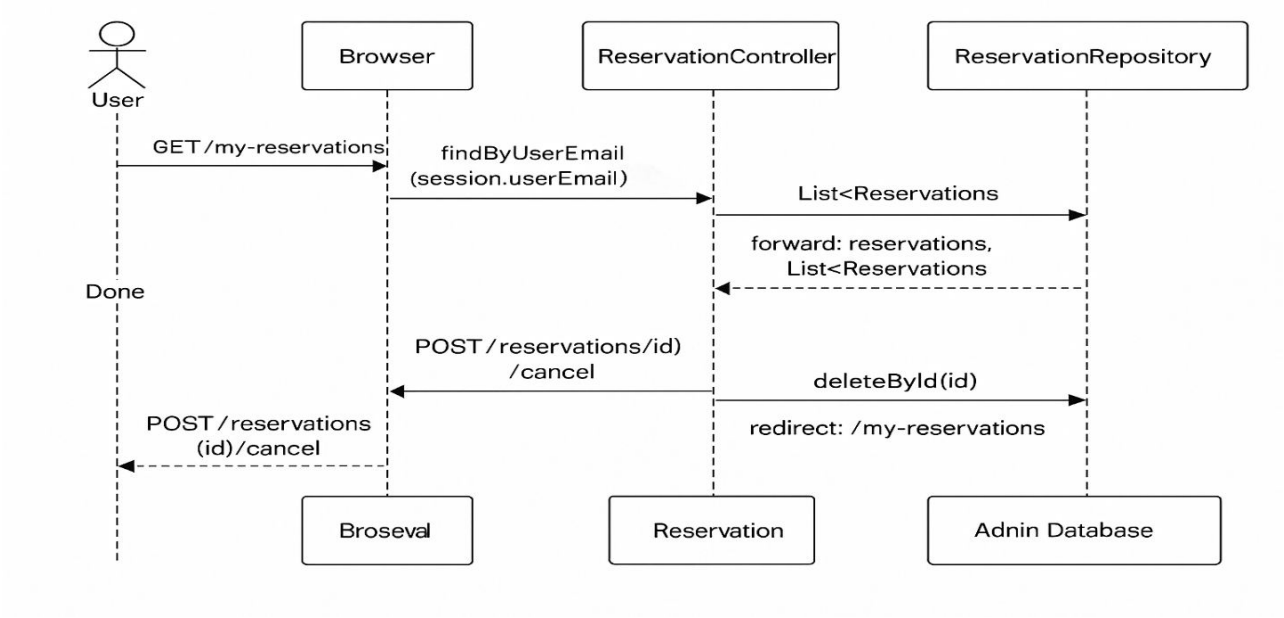

3. 예약 기능

당구장 예약 기능은 로그인된 일반 사용자가 당구장의 이름, 위치, 예약 날짜, 시간, 사용할 당구대 수를 입력하여 예약을 완료할 수 있도록 설계된 기능이다. 이 과정은 사용자 인터페이스를 통해 예약 정보를 입력한 후, /reservation 경로로 POST 요청을 전송하는 것으로 시작된다.

이 요청은 ReservationController에서 수신되며, 컨트롤러는 먼저 HttpSession을 통해 현재 로그인된 사용자 정보를 확인한다. 만약 로그인 상태가 아니라면 사용자는 로그인 페이지로 리다이렉트된다. 로그인된 사용자일 경우, 입력된 예약 정보는 Reservation 객체에 바인딩되고, 사용자 이메일이 userEmail 필드에 자동으로 할당된다. 이후 reservationRepository.save(reservation) 명령을 통해 예약 정보가 데이터베이스에 저장된다.

예약이 성공적으로 완료되면, 컨트롤러는 모델에 성공 메시지를 담아 success.html로 포워딩하며, 사용자에게 예약이 완료되었음을 시각적으로 안내한다. 만약 세션이 만료되거나 예외가 발생할 경우, 오류 처리를 통해 사용자를 다시 로그인 페이지로 안내하거나 예외 메시지를 출력할 수 있도록 구성되어 있다.

이 기능은 사용자 경험을 고려하여 예약 성공 이후에는 예약 확인 화면(/my-reservations)으로 이동할 수 있도록 유도하며, 사용자는 해당 화면에서 자신의 예약 상태를 확인하고 필요 시 취소할 수 있는 기능도 함께 제공받게 된다.




4. 예약 조회 및 취소 기능

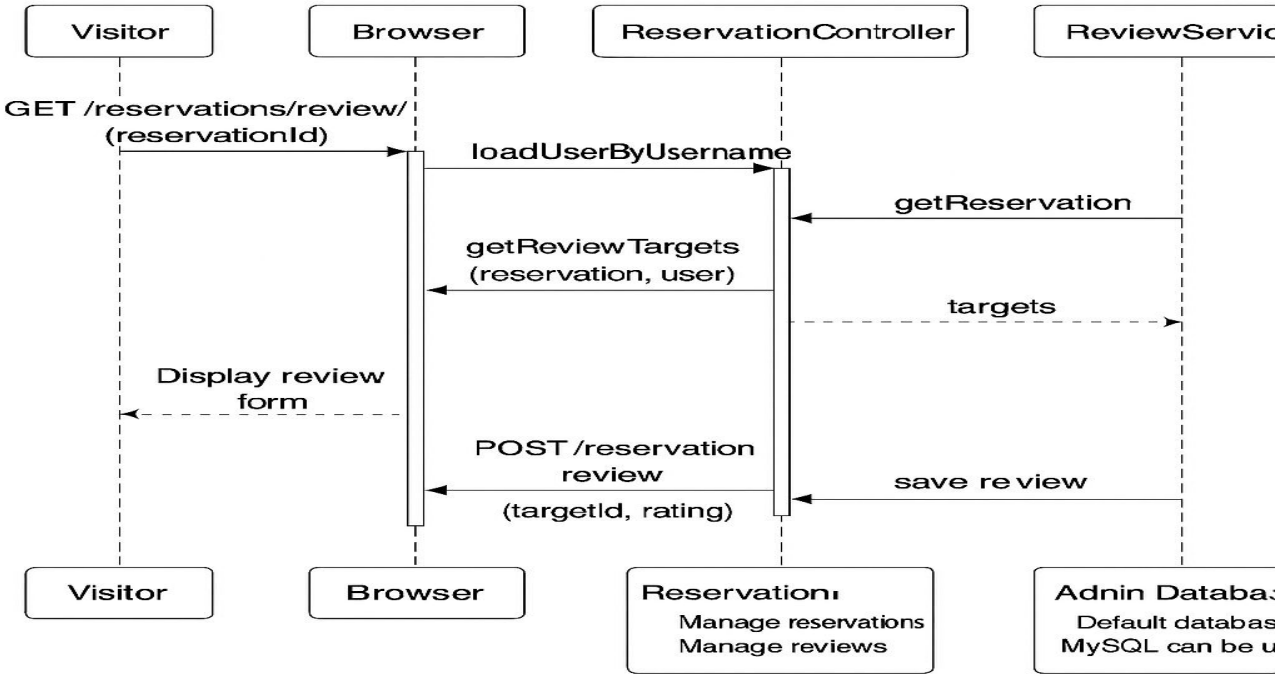
사용자는 당구장 예약 시스템에서 로그인한 이후 자신의 예약 현황을 확인할 수 있는 기능을 제공합니다. 이 기능은 /my-reservations 경로를 통해 접근 가능하며, 사용자가 해당 페이지를 요청하면 시스템은 현재 로그인된 사용자의 세션 정보를 기반으로 예약 데이터를 조회합니다.

먼저, 클라이언트(웹 브라우저)가 /my-reservations 경로로 GET 요청을 전송하면, 서버 측의 ReservationController는 현재 세션에 저장된 사용자 객체를 가져옵니다. 해당 사용자 객체에서 이메일 정보를 추출하고, 이 이메일을 기반으로 ReservationRepository.findByUserEmail(email) 메서드를 호출하여 사용자와 연결된 모든 예약 정보를 데이터베이스에서 조회합니다.

조회된 예약 목록은 리스트 형태로 Model 객체에 저장되며, 최종적으로 사용자에게 myReservations.html 템플릿이 렌더링되어 보여집니다. 이 화면에는 예약 날짜, 당구장 이름, 위치, 당구대 수, 예약 시간 등의 정보가 포함되어 있습니다.

사용자가 특정 예약에 대해 취소 버튼을 누르면, 해당 예약의 ID가 포함된 POST 요청이 /reservation/delete/{id} 경로로 전송됩니다.

ReservationController는 우선 세션 정보를 통해 사용자가 로그인되어 있는지를 확인합니다. 이후 전달받은 예약 ID에 해당하는 예약 정보를 데이터베이스에서 조회하고, 해당 예약의 이메일이 현재 로그인한 사용자와 일치하는 경우에만 해당 예약을 삭제합니다.



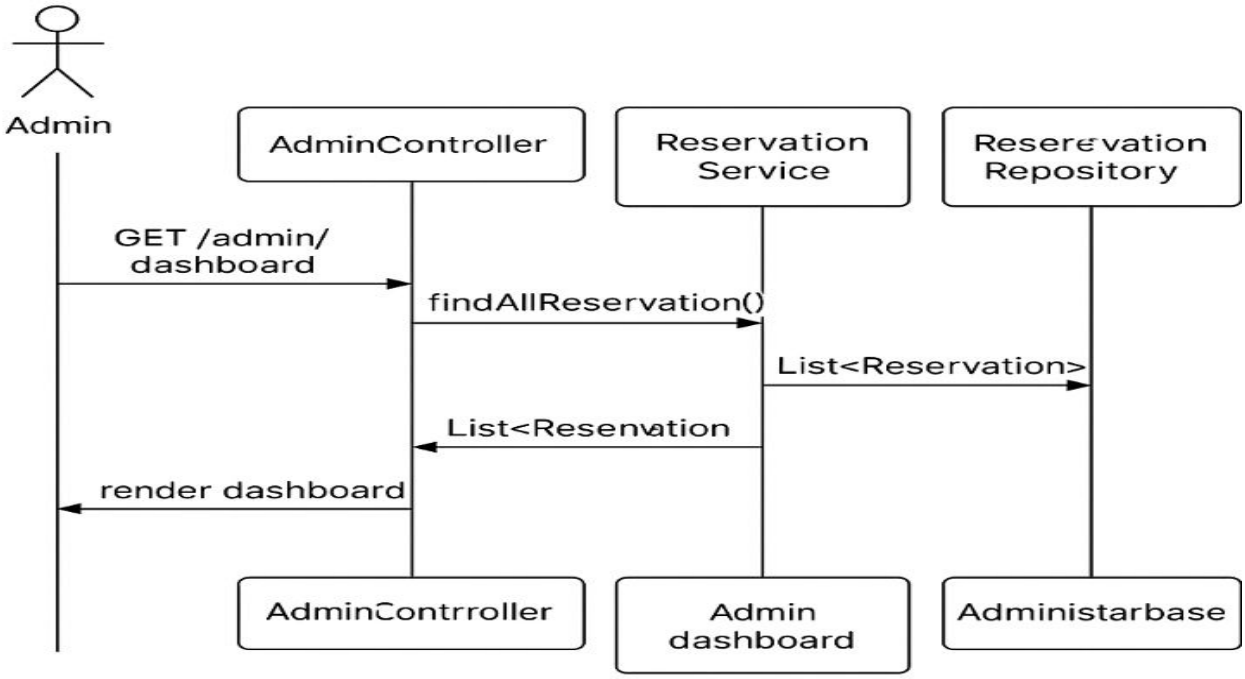

5. 리뷰 기능

리뷰 기능은 사용자가 당구장을 예약하고 실제로 이용한 후, 해당 경험에 대한 피드백을 남길 수 있도록 제공된다. 이는 사용자 개인의 후기 공유를 통해 타 사용자에게 참고 정보를 제공하고, 관리자에게 서비스 품질 향상을 위한 인사이트를 제공하기 위한 기능이다.

사용자는 로그인한 상태에서만 리뷰를 작성할 수 있으며, 리뷰 작성 버튼을 통해 리뷰 입력 화면으로 이동할 수 있다. 리뷰 입력 폼에는 별점(1~5점 사이의 정수형 평가)과 자유 텍스트 형식의 후기 내용이 포함되어 있다. 사용자는 자신의 예약 기록 중 특정 당구장 예약을 선택한 뒤, 리뷰를 등록할 수 있다.

리뷰 작성 요청은 POST 방식으로 서버에 전달되며, Spring Boot 기반의 ReviewController에서 해당 요청을 수신한다. 컨트롤러는 세션 정보를 기반으로 로그인한 사용자의 정보를 조회하고, 해당 사용자와 연결된 User 객체를 통해 사용자의 이메일 혹은 아이디를 식별한다. 이후 전달된 리뷰 데이터를 Review 엔티티 객체로 매핑하여 ReviewRepository를 통해 데이터베이스에 저장한다.

저장된 리뷰는 향후 관리자(Admin)가 열람 가능하도록 구현되어 있으며, 필요 시 부적절한 리뷰에 대해 삭제하거나 비공개 처리할 수 있는 기능도 제공된다. 또한, 일반 사용자도 당구장 상세 화면 혹은 리뷰 목록 페이지에서 타인의 리뷰를 확인할 수 있도록 설정되어 있다.



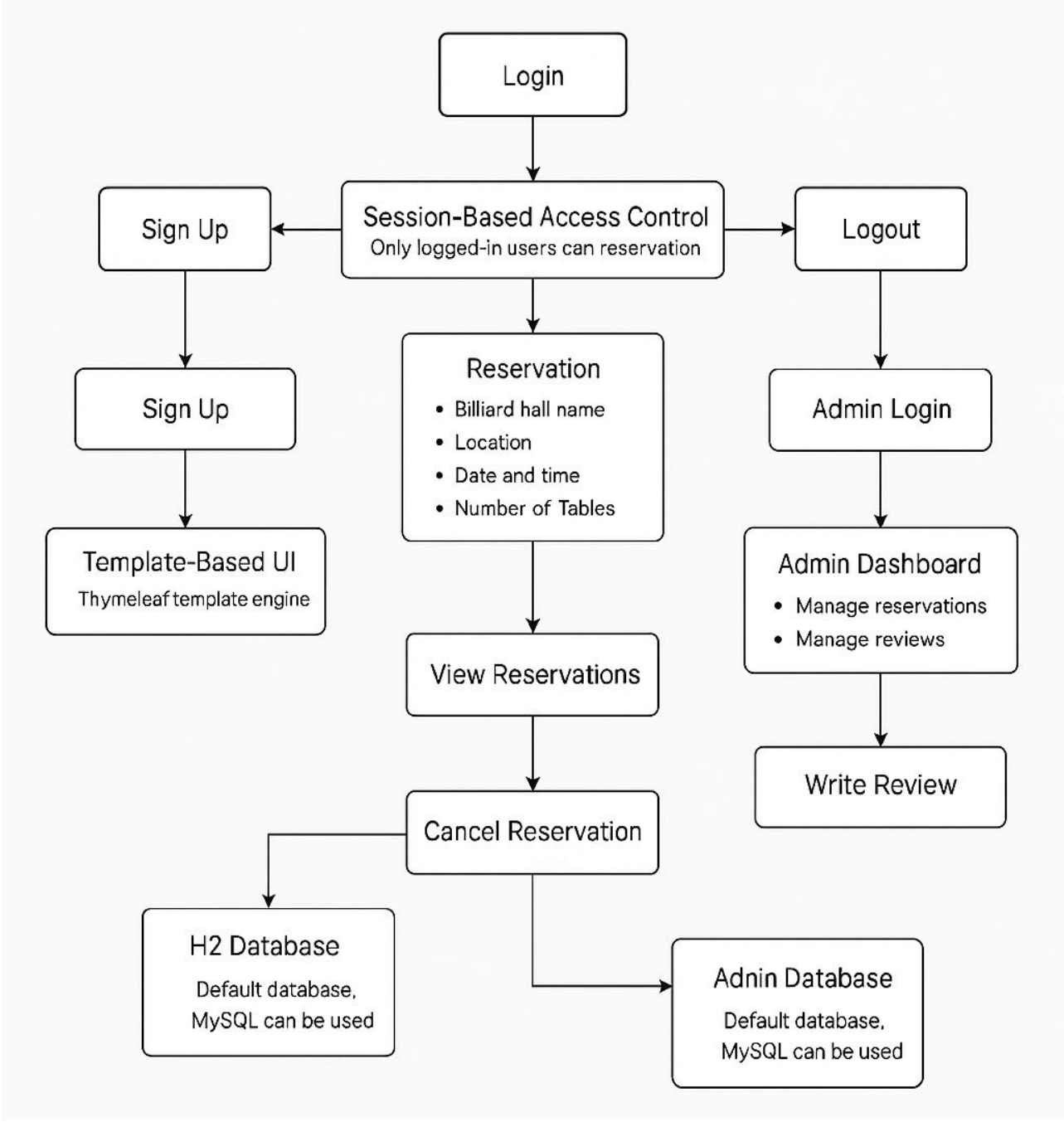

6. 관리자 예약 확인 기능

관리자의 예약 확인 기능은 당구장 예약 시스템의 핵심적인 관리 도구 중 하나로, 시스템의 전반적인 예약 현황을 파악하고 유지 관리하는 데 사용된다. 이 기능은 관리자가 로그인한 후 예약 데이터를 확인하고 필요한 경우 조치를 취할 수 있도록 설계되어 있다.

관리자는 먼저 관리자 로그인 페이지를 통해 고정된 관리자 ID와 비밀번호를 입력하여 로그인한다. 로그인 성공 시, 세션에 관리자 권한 정보가 저장되며, 이는 이후의 기능 접근 제어에 사용된다. 로그인 후 관리자는 /admin/dashboard와 같은 경로를 통해 관리자 전용 대시보드에 접근할 수 있다.

이 대시보드에서는 ReservationController 또는 AdminReservationController와 같은 컨트롤러가 실행되며, 내부적으로 ReservationRepository.findAll() 메서드를 호출하여 모든 예약 정보를 데이터베이스로부터 조회한다. 이렇게 조회된 예약 목록은 모델 객체에 추가되어, adminDashboard.html 템플릿에 전달된다. 이를 통해 관리자는 예약자의 이메일, 당구장 이름, 위치, 예약 날짜 및 시간, 당구대 수 등의 정보를 한눈에 확인할 수 있다.

4. State machine diagram



State Machine Diagram 설명 (당구장 예약 시스템)

1. 초기 상태 : Login
- 사용자는 시스템을 사용하기 위해 로그인해야 합니다.
- 로그인은 사용자와 관리자로 구분되며, 각각 다른 흐름을 갖는다.

2. 로그인 성공 후 : Session-Based Access Control
- 세션 기반 접근 제어가 적용된다.
- 로그인된 사용자만 예약, 예약 조회, 리뷰 작성 등의 기능을 사용할 수 있다.
- 비로그인 사용자는 /reservation, /my-reservations, /review 등에 접근 시 로그인 페이지로 리다이렉션된다.

(일반 사용자 흐름)

1. Sign up
- 사용자는 이메일, 이름, 비밀번호 등을 입력하여 회원가입을 진행한다.
- 아이디 중복 검사, 비밀번호 확인 검증이 포함되어 있다.

2. Reservation
- 로그인한 사용잔느 다음 정보를 입력해 예약을 진행한다.
- 당구장 이름, 위치, 날짜와 시간, 당구대 수
- ReservationController 가 입력을 처리하고 DB에 저장한다.

3. View Reservation
- 사용자는 /my-reservations 페이지에서 자신의 예약 내역을 확인할 수 있다.

4. Cancel Reservation
- 예약 목록에서 각 항목별로 삭제 버튼이 제공되며,
- 해당 버튼을 누르면 DB에서 예약이 삭제된다.
- 삭제는 사용자 본인의 예약에만 가능하다.

5. Write Review
- 사용자는 예약 후 당구장에 대한 리뷰를 남길 수 있다.
- 별점과 코멘트를 포함하며, 이는 관리자에게도 노출된다.

(관리자 흐름)

1. Admin Login
- 관리자 계정으로 로그인한다.
- 세션에 isAdmin=true가 저장되어 접근 권한을 부여한다.

2. Admin Dashboard
- 전체 예약 목록을 확인할 수 있으며,
- 예약 항목을 삭제할 수 있는 권한도 포함된다.
- 향후 리뷰 관리 기능 (삭제/블라인드)도 추가 예정한다.

3. Logout
- 로그아웃 시 세션이 종료되고, 보호된 모든 URL 접근이 차단된다.
- 이후에는 다시 로그인하지 않으면 기능 사용이 불가능한다.

(공통 시스템 요소)

1. Template-Based UI
- 모든 화면은 Thymeleaf 템플릿 엔진 기반으로 동작한다.
- 사용자와 관리자 페이지는 각각 다른 레이아웃으로 구성되어 있다.

2. H2 / MySQL Database
- 기본적으로 H2 DB를 사용하며, MySQL로 쉽게 확장 가능하도록 설계되어 있다.
- UserRepository, ReservationRepository 등은 Spring Data JPA를 기반으로 데이터 접근을 수행한다.

로그인 -> 세션 기반 권한 부여 -> 예약 및 관리 기능 실행 -> 로그아웃 또는 종료 까지, 모든 단계를 상태 기반으로 명확히 나눴다.

5. Implementation requirements

| 분류 | 항목 | 내용 |
|---|---|---|
| 개발 언어 | Java | Java 17 이상을 권장함 |
| 프레임워크 | Spring Boot | Spring Boot 3.x |
| 템플릿 엔진 | Thymeleaf | HTML 기반 서버 사이드 렌더링 템플릿 |
| 데이터 베이스 | H2 Database / MySQL | DB 환경 |
| JPA | Spring Data JPA | Hibernate 기반 ORM 사용 |
| 세션 관리 방식 | HttpSession | 로그인 상태 유지를 위한 기본 세션 저장 |
| IDE | Intellij IDEA | Java 및 Spring 개발에 최적화된 환경 |
| 프론트엔드 구성 | HTML + CSS 기반 | 반응형 UI를 위한 기본 스타일 적용 (style.css) |
| OS 환경 | Windows 10이상 | Java가 설치된 모든 OS에서 실행이 가능하다 |
| 브라우저 호환성 | Chrome | 기본 HTML 기반 페이지 |
| 인증 방식 | 세션 기반 인증 | Spring Security 사용 가능 |
| 관리자 인증 | 정적 ID/PW | host/1234 보안 기능 확장 가능 |
| 기타 확장성 | Docker/Docker compose | 향후 DB 배포 |

1. 개발 및 운영

시스템 개발은 Java 언어를 기반으로 한다. 프로젝트 전반에 걸쳐 Spring boot 프레임워크가 사용되며, 이는 MVC 구조 기반의 웹을 쉽고 일관성 있게 구성할 수 있도록 도와준다. Intellij 개발 환경을 통해 구성했고 Maven 빌드 툴을 통해 자동으로 처리된다.

2. 데이터베이스 구성

기본 데이터 저장소는 H2 Database로 설정되어 있으며, 이는 인메모리 방식으로 빠르고 간편한 테스트 및 개발이 가능합니다. application.properties에 JDBC URL을 jdbc:h2:mem:testdb 형식으로 지정하면 자동 연결됩니다.

운영 환경에서는 MySQL 또는 MariaDB로의 확장이 용이하도록 JPA 기반 구조를 사용하고 있으며, spring.datasource.url 설정만 변경하면 데이터 소스 전환이 가능합니다.

데이터베이스 연동은 Spring Data JPA를 기반으로 하며, UserRepository, ReservationRepository, ReviewRepository 등을 통해 CRUD 연산이 자동으로 매핑됩니

다.

3. 화면 구성 및 UI 설계

프론트엔드는 서버 사이드 렌더링 방식의 Thymeleaf 템플릿 엔진을 통해 구성되어 있으며, 모든 페이지는 templates/ 디렉터리에 .html 파일로 구성됩니다. 각 템플릿은 HTML5 문법과 함께 CSS 스타일링이 적용되어 있으며, 공통된 UI 요소(버튼, 입력창 등)는 static/style.css에서 일괄적으로 관리됩니다.

관리자 페이지(adminDashboard.html)와 사용자 메인 페이지(main.html)는 역할에 따라 명확히 분리되어 있고, UI/UX 측면에서도 각각 최적화된 화면 설계를 따릅니다.

4. 세션 기반 인증 및 접근 제어

본 시스템은 로그인한 사용자만 당구장 예약이 가능하도록 세션 기반 인증 방식(HttpSession)을 적용합니다. 사용자가 로그인에 성공하면 사용자 정보를 session.setAttribute("user", user) 형식으로 저장하고, 이후 접근하는 모든 보호 URL(/reservation, /my-reservations)에서 이 세션 정보를 기준으로 인증 및 접근 여부를 판단합니다.

관리자의 경우, isAdmin이라는 세션 키를 통해 사용자와 권한이 구분되며, 관리자 로그인은 고정된 ID/PW (예: host / 1234) 방식으로 구성되어 있습니다. 추후 Spring Security 또는 OAuth2를 적용하면 보다 강력한 인증 및 권한 관리가 가능합니다.

5. 지원 브라우저 및 반응형 UI

HTML과 CSS를 활용한 정적 페이지 기반으로 구성되어 있어, Chrome, Edge, Firefox 등 대부분의 최신 브라우저에서 완전한 호환성을 제공합니다. 모든 입력창, 버튼, 테이블은 기본적으로 반응형으로 구성되어 있어 데스크탑 및 노트북 화면에서도 문제 없이 사용할 수 있습니다. 향후 모바일 환경에 대응하기 위해 Bootstrap 또는 Tailwind와 같은 CSS 프레임워크 도입도 고려할 수 있습니다.

6. 기타 구성 및 확장성

테스트 프레임워크: Spring Boot와 연동되는 JUnit5 기반의 단위 테스트 구조가 적용되어 있으며, 컨트롤러 및 서비스 레이어의 핵심 로직에 대해 테스트 케이스 추가가 가능합니다.

도커(Docker): 추후 운영 배포 환경에서는 Dockerfile과 docker-compose.yml 파일을 통해 MySQL DB와 함께 컨테이너 기반 실행도 가능합니다.

보안 강화: 현재 비밀번호는 평문으로 저장되나, BCryptPasswordEncoder를 통해 암호화된 방식으로 개선할 예정이며, 이를 위해 Spring Security 도입도 함께 계획하고 있습니다.

6. Glossary

| 용어 | 정의 |
|---|---|
| 로그인 | 사용자가 이메일과 비밀번호를 입력하여 시스템에 인증을 요청하는 과정입니다. 인증이 성공하면 세션을 통해 사용자 정보를 유지한다. |
| 회원가입 | 사용자가 이름, 이메일, 비밀번호 등의 정보를 입력하고 시스템에 사용자로 등록하는 과정이다. |
| 세션 | 사용자의 로그인 상태를 유지하기 위한 서버 내 메모리 공간이다. Spring의 HttpSession 객체를 통해 구현된다. |
| 당구장 예약 | 사용자가 당구장 이름, 위치, 날짜, 시간, 당구대 수를 입력하여 예약 정보를 DB에 저장하는 기능이다. |
| 예약 확인 | 사용자가 자신의 예약 내역을 조회할 수 있는 기능이다. /my-reservation URL을 통해 접근한다. |
| 예약 취소 | 사용자가 예약한 내역을 삭제하는 기능이다. 실제 DB에서 해당 예약 정보가 삭제된다. |
| 관리자 | 일반 사용자와 구분되는 권한을 가진 계정으로, 전체 예약 내역을 확인하거나 삭제할 수 있다. 아이디와 비밀번호는 고정값이다. |
| 관리자 대시보드 | /admin/dashboard 경로로 접근할 수 있는 관리자 전용 화면으로, 모든 예약 정보를 열람하거나 조작할 수 있다. |
| Thymeleaf | HTML 템플릿 엔진으로, Java 객체를 HTML에 렌더링할 수 있게 도와주는 서버 사이드 뷰 기술이다. |
| Spring Boot | Java 기반의 웹 프레임워크로, 설정을 최소화하면서 웹 서버를 빠르게 구축할 수 있게 해준다. |
| ReservationController | 사용자 예약 요청을 처리하는 컨트롤러 클래스이다. 예약 생성, 확인, 삭제 등을 담당한다. |

| | |
|---|---|
| Review | 사용자가 당구장에 대해 남기는 후기이다. 별점 및 코멘트를 포함하며, 관리자 페이지에서 관리 가능한다. |
| 세션 기반 접근 제어 | 로그인된 사용자만 특정 기능을 사용할 수 있도록 제한하는 방식이다. |
| AdminSession | isAdmin 값을 세션에 저장하여 일반 사용자와 구분하는 논리적 인증 장치이다. |
| ReviewController | 사용자의 후기 등록과 관리 기능을 제공하는 컨트롤러이다. |
| Static Resource | 정적 리소스를 의미하며, CSS, JS 이미지 파일 등을 /static/ 디렉토리에 위치시켜 제공한다. |
| JPA Entity | Java클래스와 DB 테이블을 매핑시키는 객체이다. |

## 7. References

1. Spring Security
https://docs.spring.io/spring-security
2. Spring Boot Reference Documentation
https://docs.spring.io/spring-boot
3. Spring Data JPA
https://docs.spring.io/spring-data/jpa
4. Thymeleaf Documentation
https://www.thymeleaf.org/documentation.html