**High-level Architecture Design**

Unicircle-서울시립대 동아리 플랫폼



Team Universe

24.12.13 (Version: 2.0)

2020920006 김동하

2020920008 김민회

2020920023 박세환

2020920029 배유찬

2020920061 주재원

2022920041 오승민

변 경 이 력

|  |  |  |  |
| --- | --- | --- | --- |
| **버전** | **일자** | **변경 내역** | **작성자** |
| 1.0 | 2024-11-17 22:00 | 최초 작성, High-level Architecture, Class Diagram 추가 | 백엔드 팀 |
| 2.0 | 2024-12-13 | 목차 수정 | 오승민 |
|  |  |  |  |
|  |  |  |  |

목 차

목차

[High-level Architecture Diagram 3](#_Toc184992441)

[3 Tier Architecture Diagram 3](#_Toc184992442)

[Class Diagram 5](#_Toc184992443)

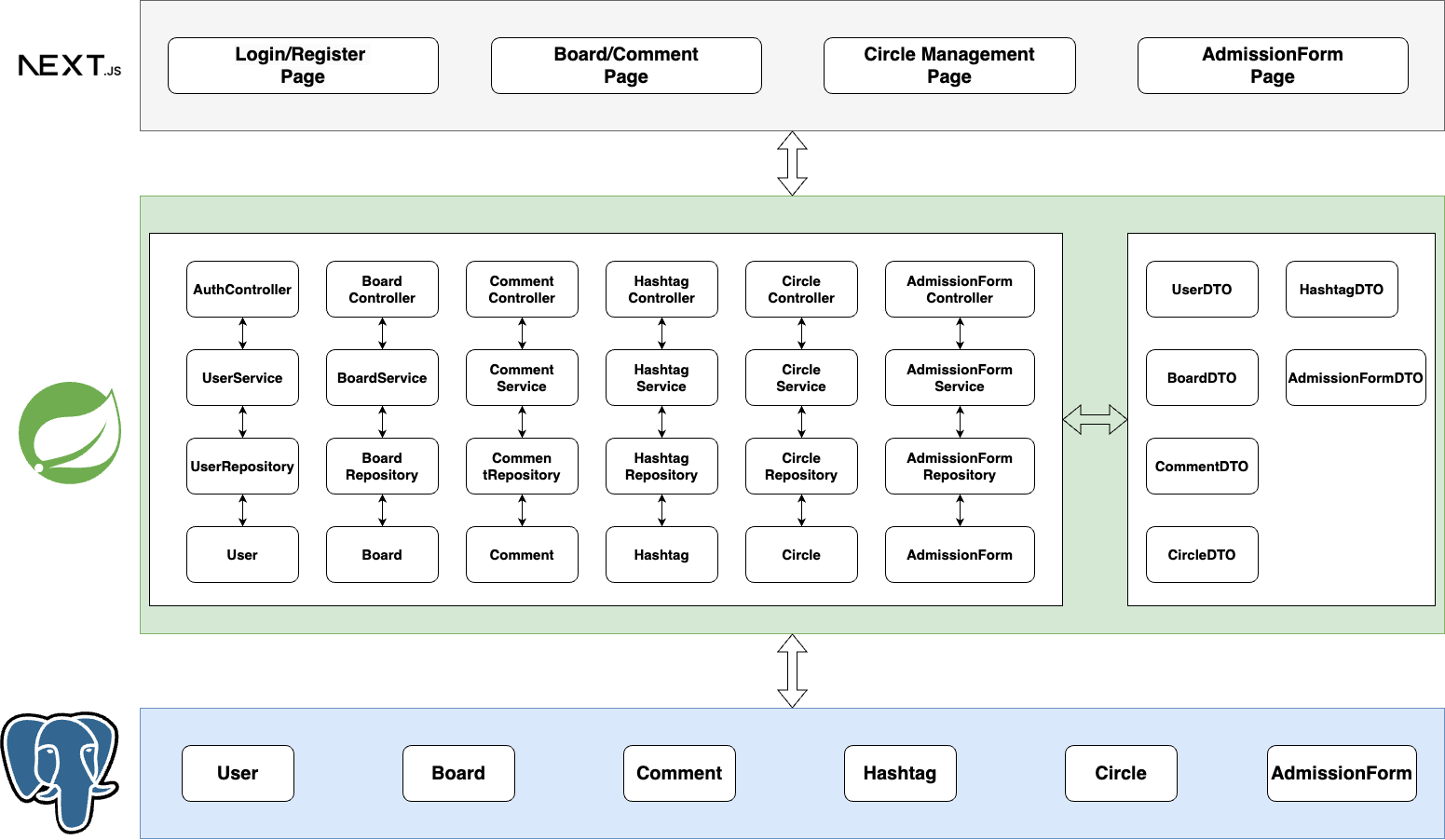
[Activity Diagram 6](#_Toc184992444)

[Description 10](#_Toc184992445)

[Sequence Diagram 11](#_Toc184992446)

# High-level Architecture Diagram

## 3 Tier Architecture Diagram



**Description**

Tier 1: Presentation Layer

- 기술: Next.js

- 역할:

- 사용자 인터페이스(UI) 제공.

- 사용자의 입력 데이터를 수집하고, 데이터를 시각적으로 표현.

- Spring 서버(Tier 2)와 REST API 또는 GraphQL을 통해 통신.

- 예: 동아리 목록을 조회하거나 특정 동아리 정보를 입력하는 화면 구성.

Tier 2: Application Logic Layer

- 기술: Spring Framework

- 역할:

- 비즈니스 로직 처리.

- 데이터 검증, 인증, 권한 관리 수행.

- 데이터 요청 및 응답 처리.

- PostgreSQL(Tier 3) 데이터베이스와 직접 통신하여 필요한 데이터를 조회하거나 저장.

- 예: 사용자가 제출한 동아리 질문 데이터를 검증하고, 데이터베이스에 저장.

Tier 3: Data Layer

- 기술: PostgreSQL

- 역할:

- 애플리케이션(Tier 2)의 요청에 따라 데이터를 저장, 수정, 삭제, 조회.

- 데이터 무결성과 일관성을 유지.

- 예: 동아리 정보, 유저 정보, 질문 목록 등을 저장하고 관리.

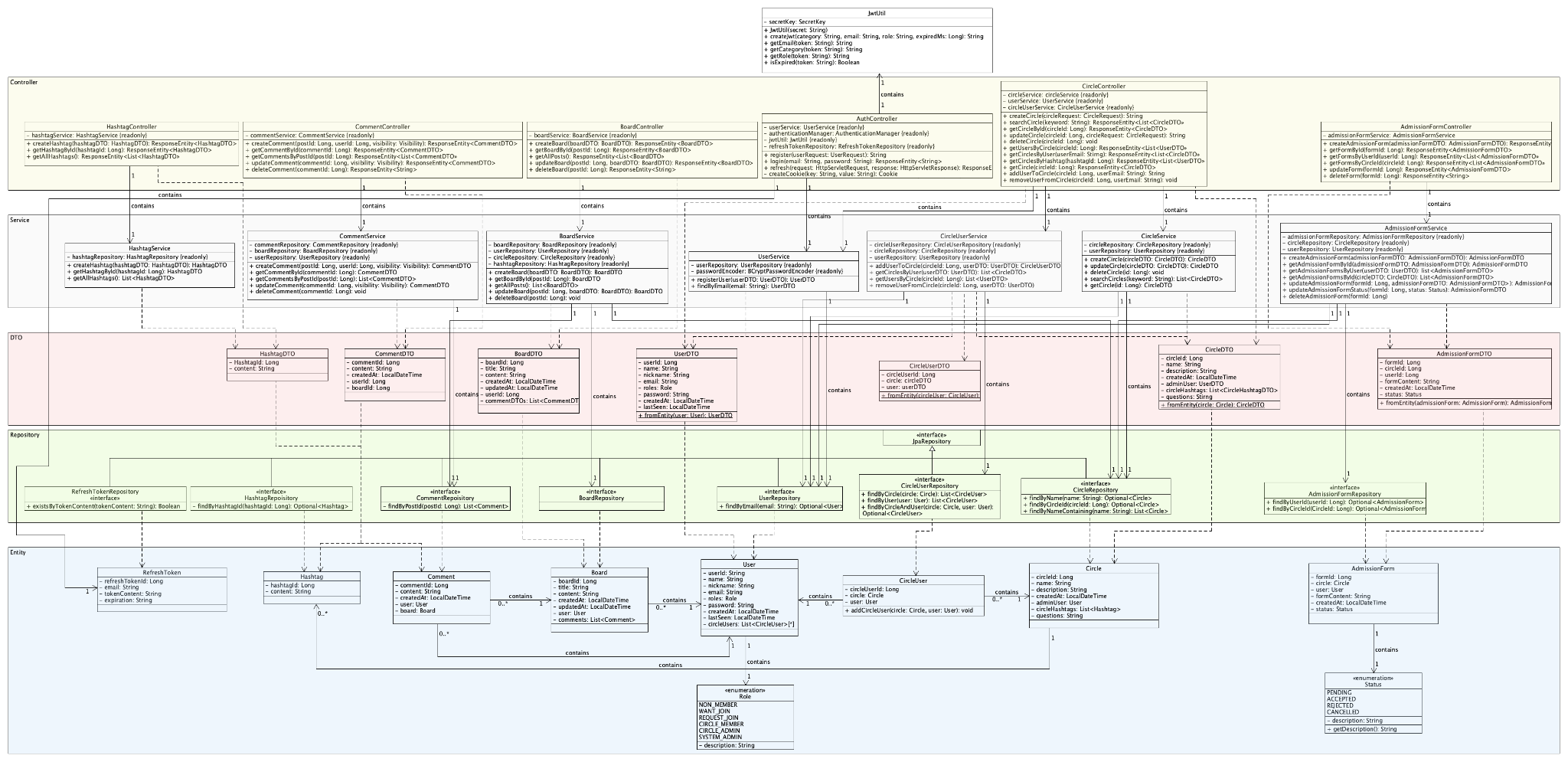
동작 흐름 예시

1. Tier 1: 사용자가 Next.js 웹 애플리케이션에서 동아리 질문을 입력하고 “저장” 버튼을 클릭
2. Tier 2: 입력된 데이터가 REST API를 통해 Spring 서버로 전달되며, 서버는 데이터 유효성을 검증하고 비즈니스 로직을 처리
3. Tier 3: Spring 서버가 PostgreSQL에 데이터를 저장하거나 기존 데이터를 수정

**Tier 3 → Tier 2 → Tier 1: PostgreSQL에서 응답된 데이터가 Spring 서버를 통해 프론트엔드로 전달되어 사용자에게 결과 표시**

# Class Diagram

* **Class Diagram**



자세히 보기 → Attached UML Low File

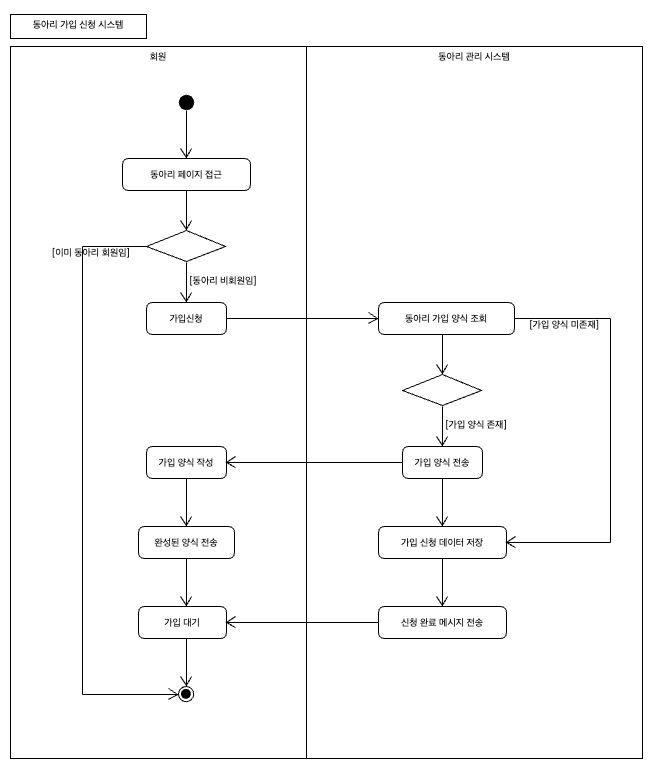
**Description**

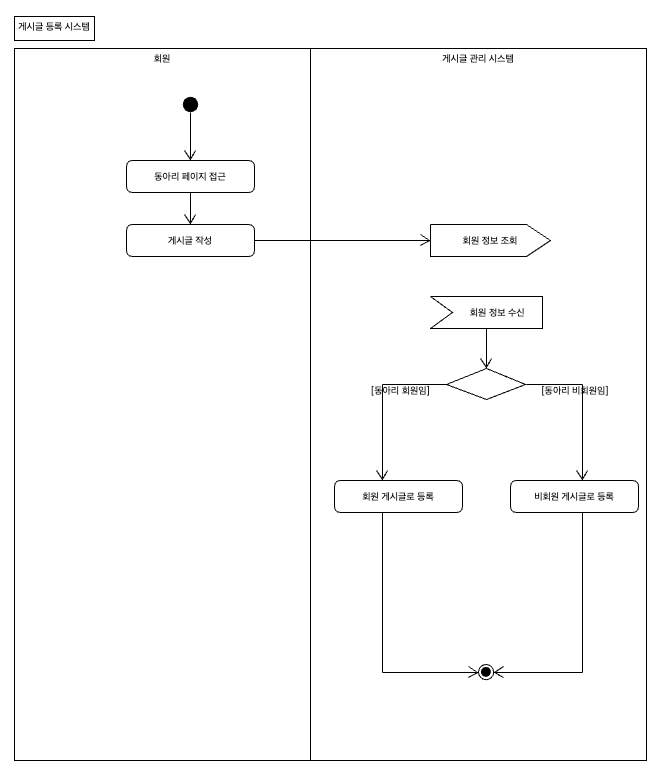
MVC 패턴을 기반으로 각 엔티티에 대해 Controller, Service, Repository, Entity, DTO 계층을 분리하여 설계한 구조.

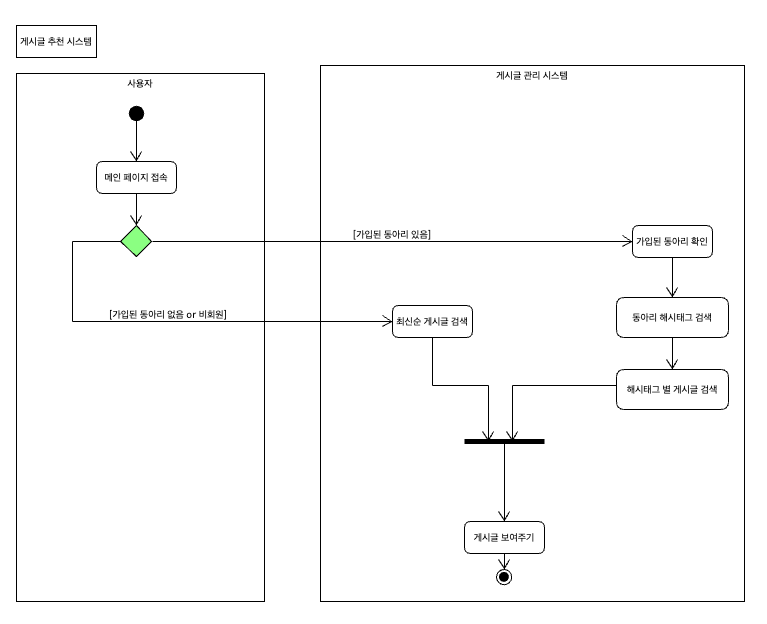
* Controller: 사용자 요청 처리.
* Service: 비즈니스 로직 담당.
* Repository: 데이터베이스 접근 관리.
* Entity: 데이터 모델 정의.
* DTO: 데이터 전달을 위한 객체로 사용.

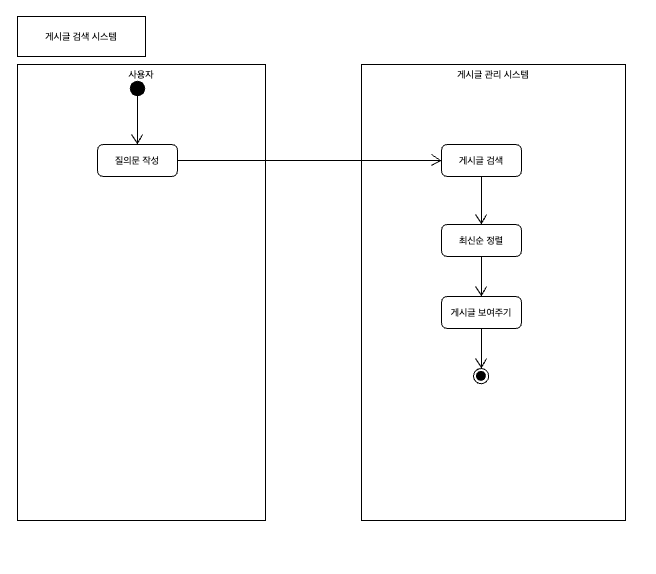
유지보수와 확장성을 염두하여 설계한 구조.

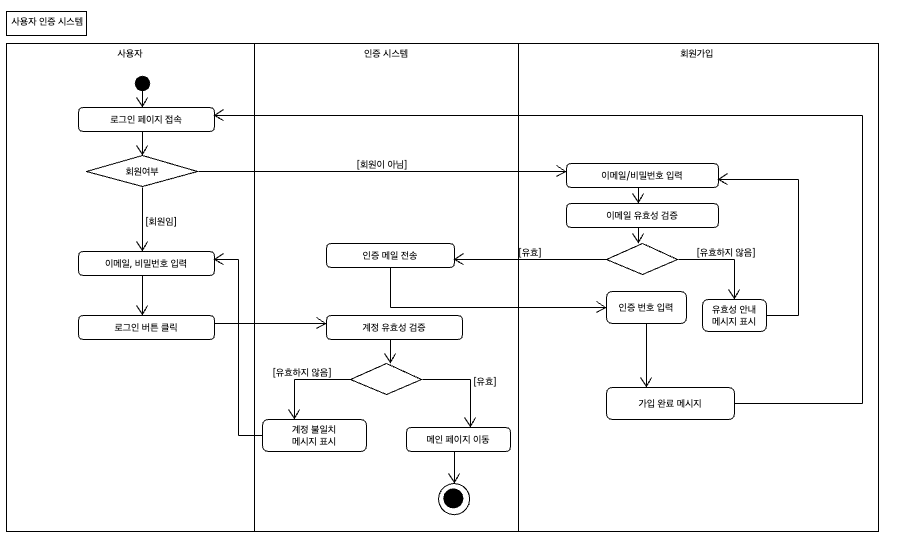
# Activity Diagram

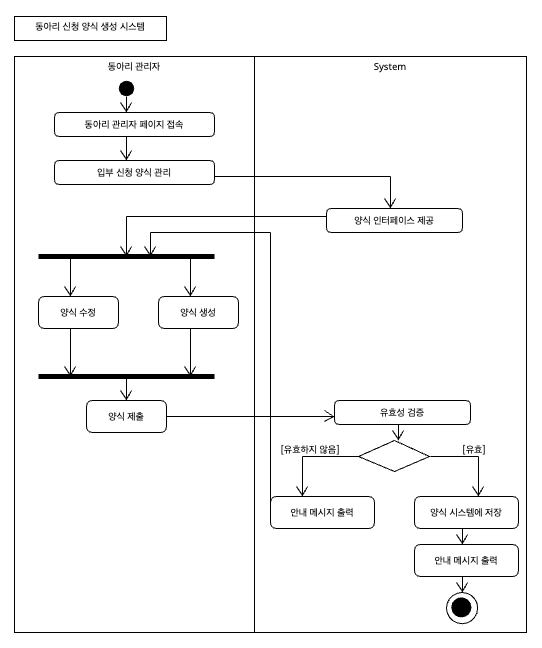










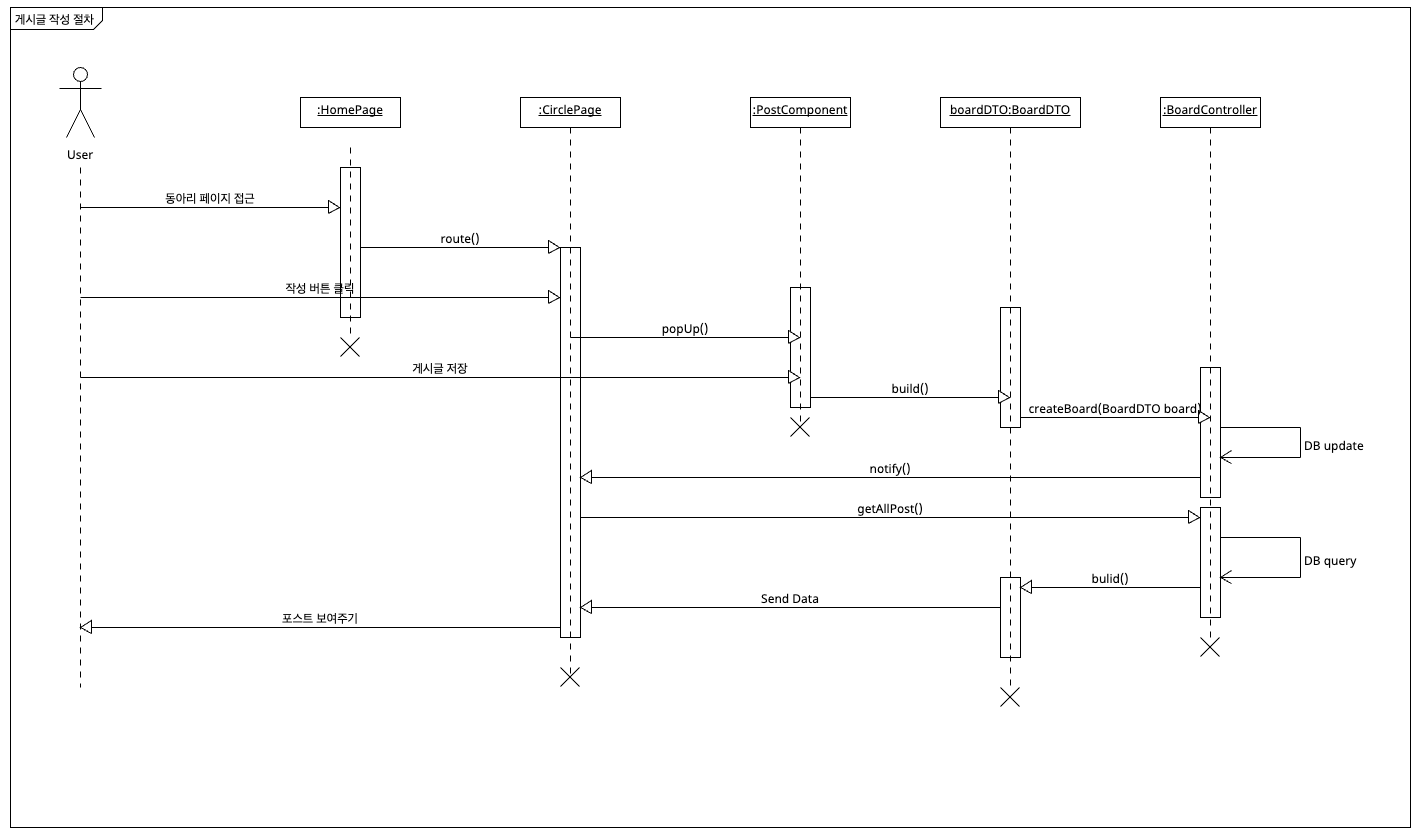


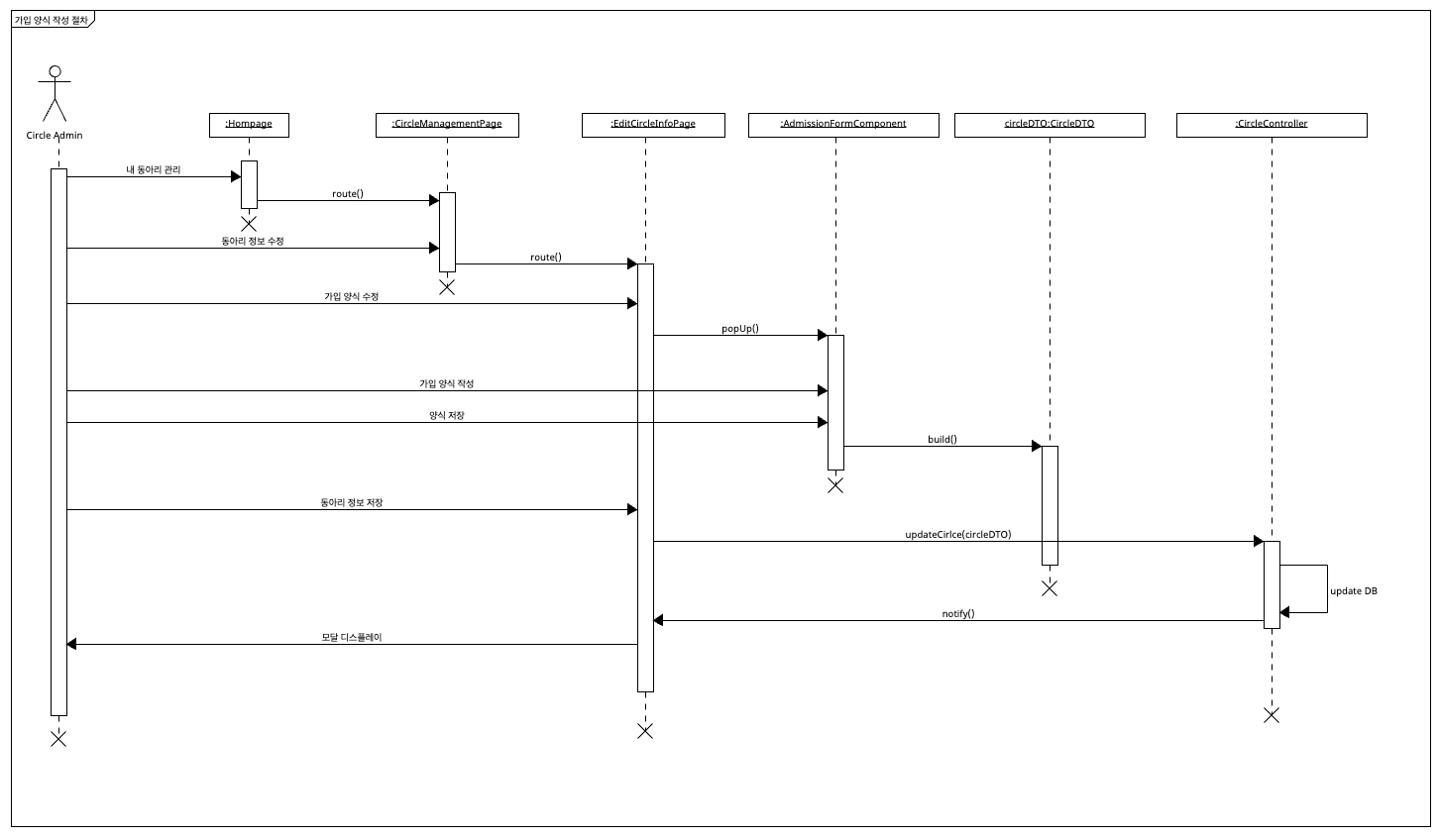
## Description

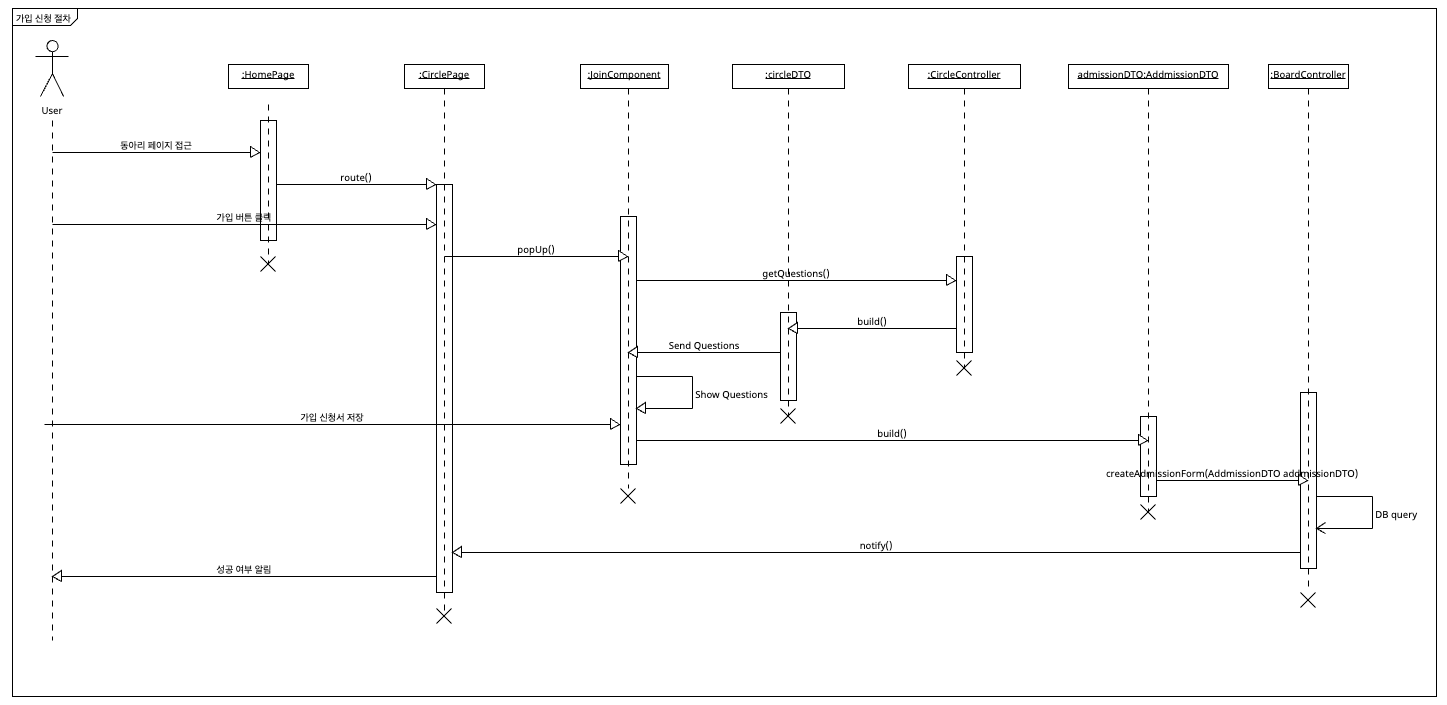
1. 동아리 신청 시스템
2. 게시글 작성 시스템
3. 게시글 추천 시스템
4. 게시글 검색 시스템
5. 사용자 인증 시스템
6. 동아리 신청 양식 생성 시스템

총 6 개의 주요 시스템을 ActivityDiagram을 작성

# Sequence Diagram



​​



**Description**

1. 게시글 작성 절차
2. 동아리 신청 절차
3. 동아리 가입 양식 제작 절차

총 3가지의 주요 절차를 SequenceDiagram으로 작성