



FIX-SKKU

성균관대학교 시설물 고장 관리 서비스

소프트웨어공학개론 12팀

이황근, 함장건, 이정빈, 전용호, 곽태윤, 박소정





픽스꼬

"학교 시설물 신고를 편하게 할 수 없을까?"



오프라인 수업 시에 책상이 흔들리거나 의자가 고장난 경우 수업에 집중하는데 방해가 됩니다.

그런데 이를 신고하는 일은 매우 번거로운 일이라서, 신고 하기보다는 다음 수업부터는 그 자리를 피해서 앉게 됩니다.

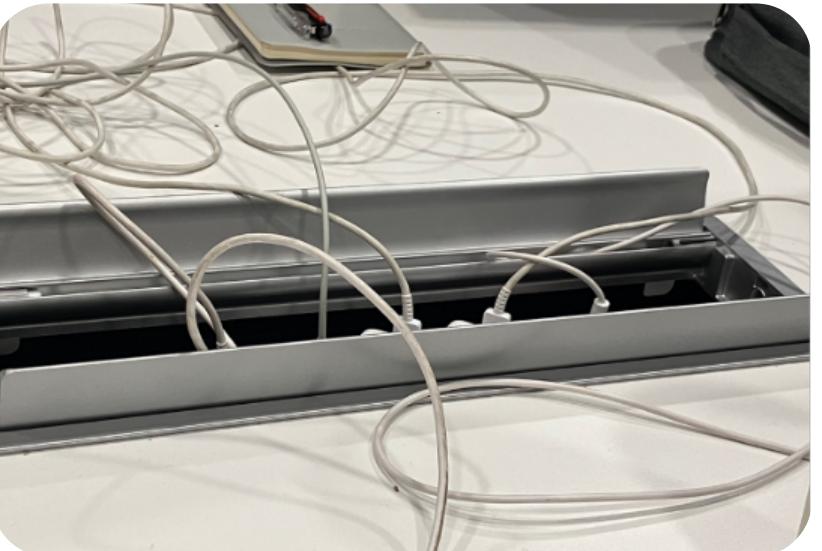
결국 누군가는 고장난 책상과 의자에 앉아서 수업을 들어야 하는 결과가 발생합니다.

"고장 처리를 더 효율적으로 할 방법은 없을까?"

관리자 입장에서 캠퍼스, 건물, 시설물 별 신고 분류가 미비해서 수동으로 배정을 해야 되는 어려움이 있습니다.

그러면 중간 관리자를 통해서 한 번 더 소통을 해야하고 그만큼의 시간이 걸리게 됩니다.

결국 고장 처리를 지연되게 되고 고장 처리에 대한 Follow가 어려워지게 됩니다.

1 공공시설 사용 문제: 책임감 저하

공공시설 사용 문제:
주인 의식 부족으로 인한 정리 및 신고 기피

2 현존 시스템의 접근성 문제

NEW 법학관 20203 강의실 콘센트 고장 문의

NEW ↳ 법학관 20203 강의실 콘센트 고장 문의

NEW ↳ 법학관 20203 강의실 콘센트 고장 문의

현존 신고 관리 게시판 접근성 문제
게시판에 접속하는 과정에서 너무 많은 프로세스

3 관리 분류 미비

학사처

'전체보기'에서는 담글을 쓸 수 없습니다. 해당 부서 페이지로 이동후 사용해주시기 바랍니다.

전체	기획조정처	교무처	학생처	산학협력단
입학처	총무처	국제처	정보통신처	학사처
학술정보관	성균여학원	관리팀/비상계획관실		

전체 고장

Total 16건, 1 / 2

No.	제목	작성자	등록일
16	NEW 법학관 20203 강의실 콘센트 고장 문의		2024-03-2

학사처 통합 관리로 인한 분류 미비
학사처측에서 불필요한 보고 과정 존재

Solution**캠퍼스, 건물, 강의실, 시설물 위치까지 손 쉽게 선택 가능한 시설물 고장 관리 어플리케이션**

성균관대학교 학생은 신속한 시설물 고장 신고를 가능하게 하고 여러 시설물의 현황 파악을 가능하게 한다.

관리자는 어플리케이션을 이용하여 간편하게 신고 확인이 가능하고 신고 처리 현황을 신고자에게 보여줄 수 있다.



Explanation

사용자는 상세한 신고를 제출하고, 신고 상태를 확인하며, 업데이트를 받을 수 있는 직관적인 기능을 이용할 수 있습니다. 기존에 있던 번거로운 신고 과정을 생략하여 언제 어디서든 쉽게 시설물 신고를 진행할 수 있습니다. QR코드를 이용하여 어플리케이션에 접속하지 않아도 그 자리에서 바로 신고가 가능합니다.

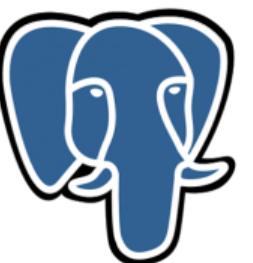
관리자는 현재 접수가 되어있는 신고를 리스트 UI로 쉽게 확인할 수 있으며
현재 신고 처리 과정을 쉽게 변경할 수 있어 사용자와의 인터랙션을 확보할 수 있습니다.
그리고 리스트 UI에서 분류를 쉽게 확인할 수 있어 시설물 관리자 개인에게 해당되는 신고를
쉽게 처리할 수 있습니다.



Figma

android studio The Android Studio logo, featuring a green Android robot icon inside a blue folder-like shape.

Front-End



PostgreSQL



Spring
Boot

Back-End



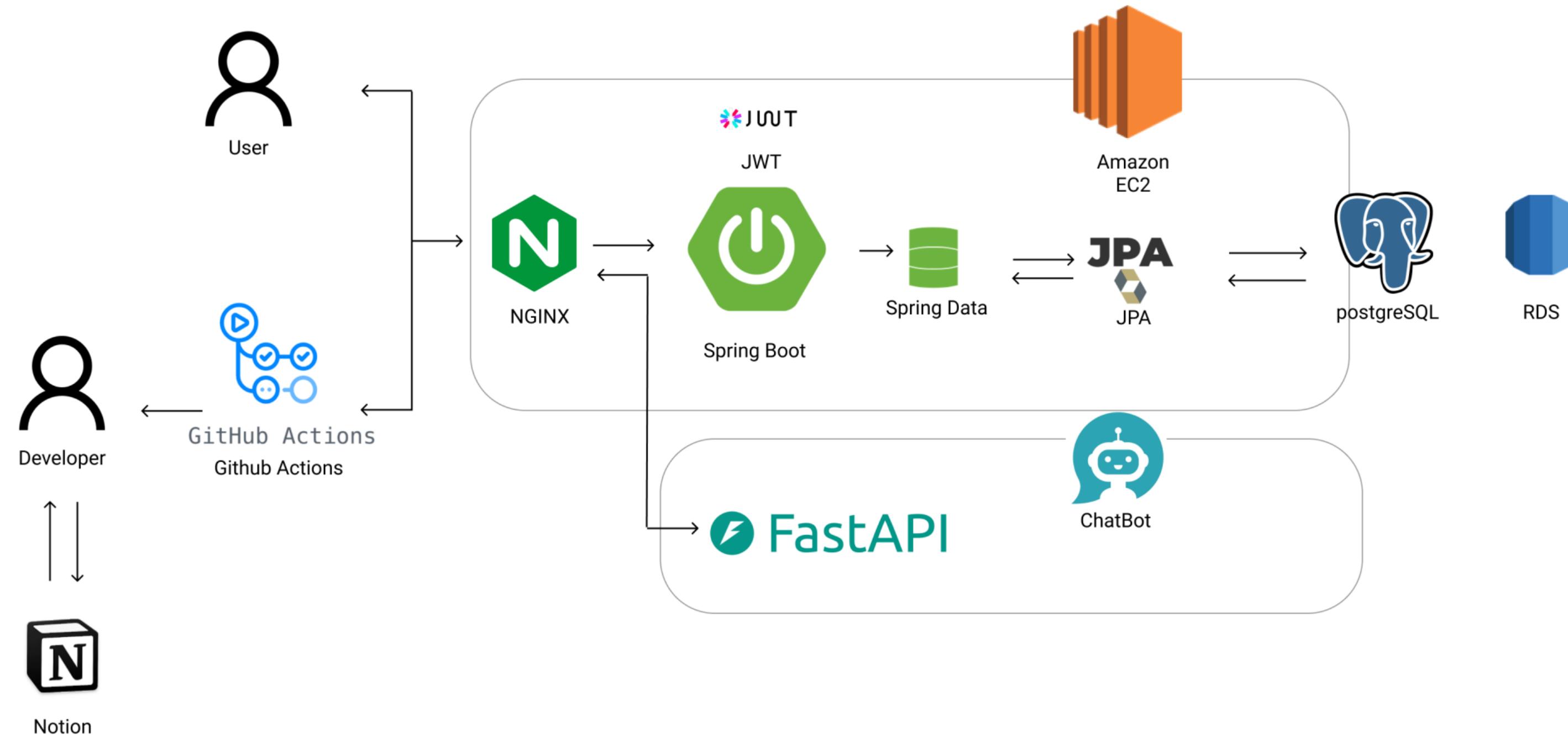
FastAPI

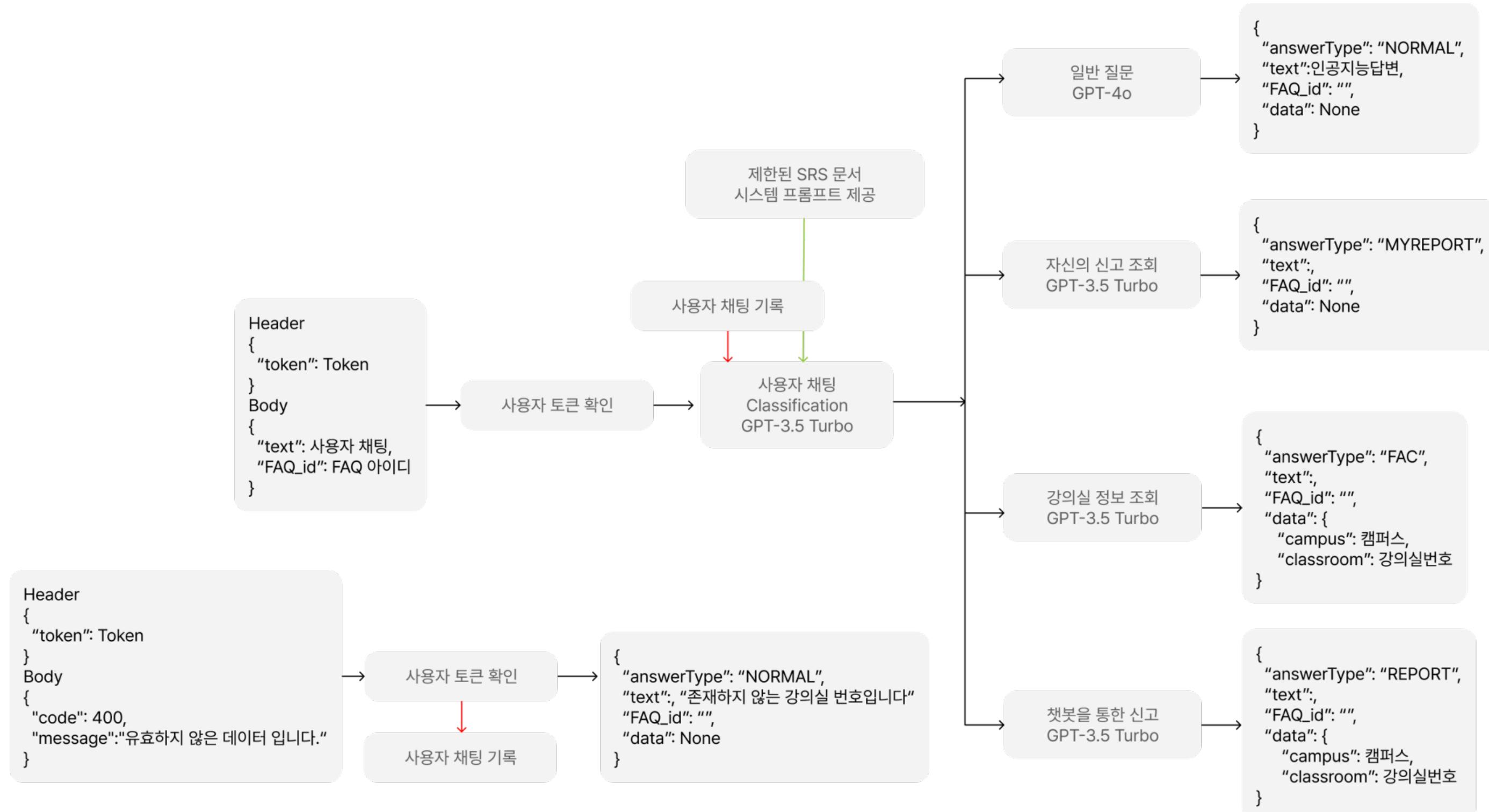


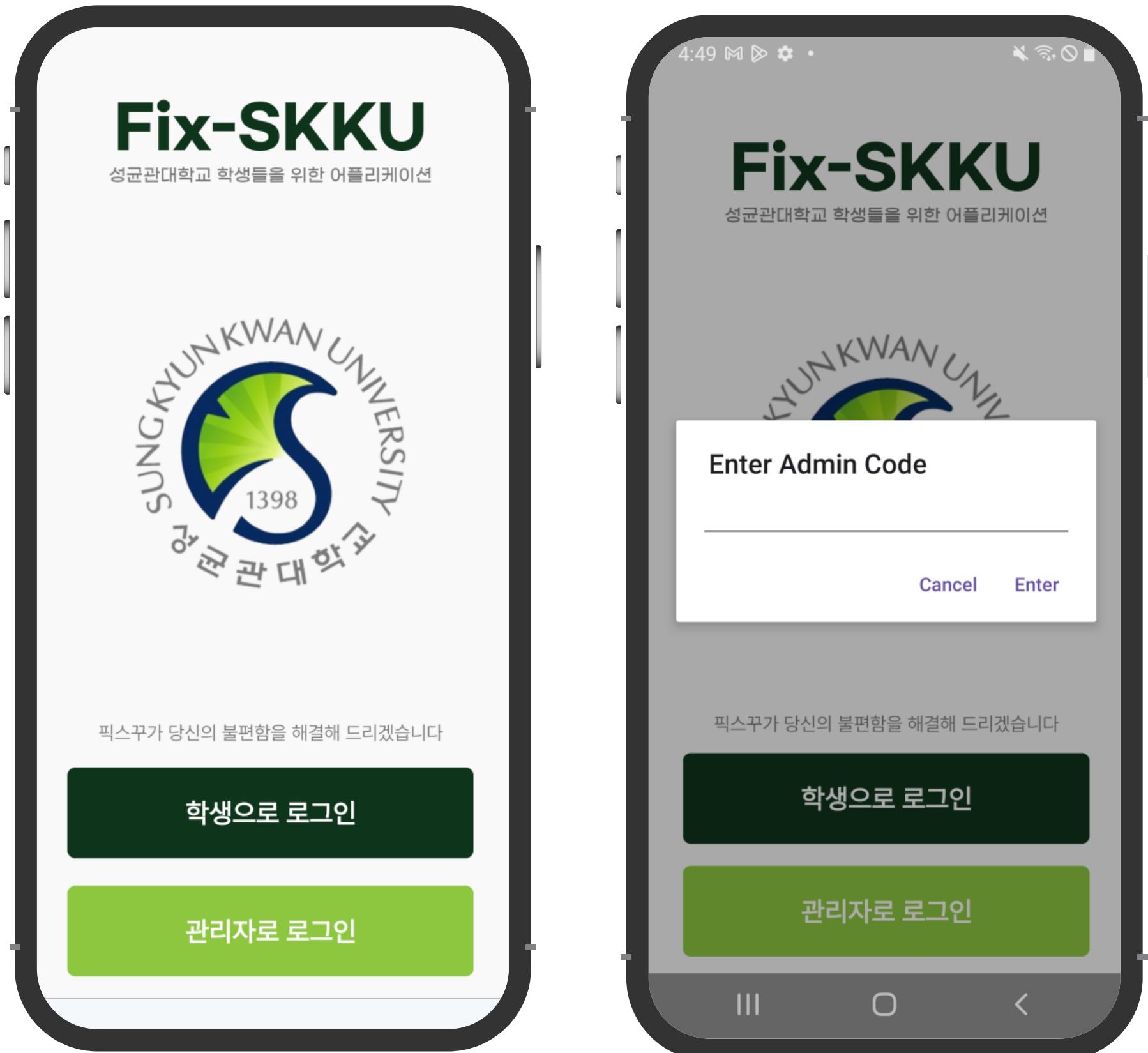
python™



AI







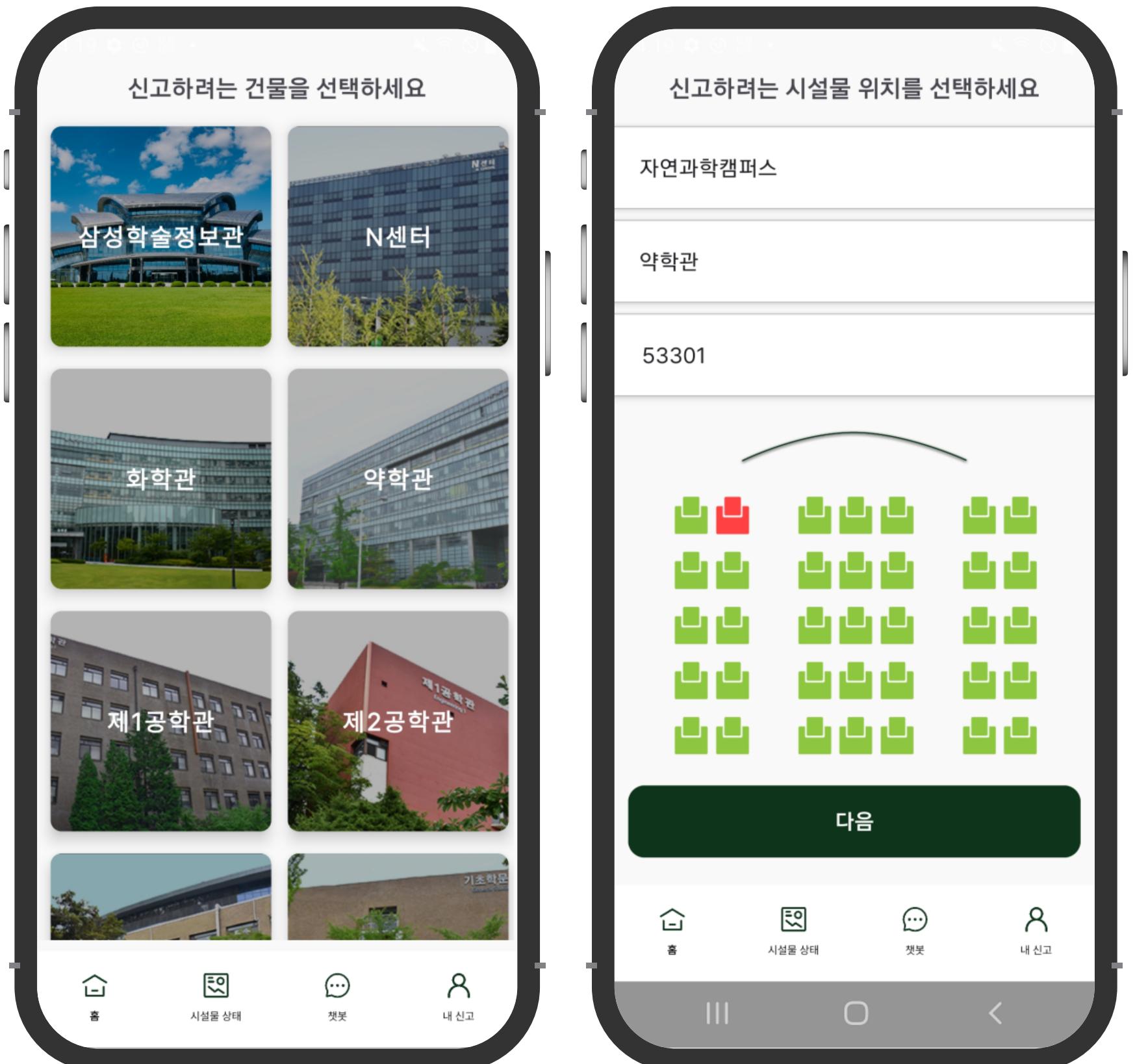
로그인 기능

학생으로 로그인

- 기기 ID를 이용하여 무기한 JWT 발급
- 모바일 앱 데이터 디렉토리에 저장
- API 요청시 헤더에 토큰값을 담아서 전달
- 학생 로그인 시 UserActivity로 이동

관리자로 로그인

- 관리자 코드 입력을 통한 관리자 로그인
- 별도의 어플리케이션 설치 X
- 관리자 로그인 시 AdminActivity로 이동



고장 신고 기능

고장 신고 기능

캠퍼스 선택

건물 선택

강의실 선택

좌석 선택

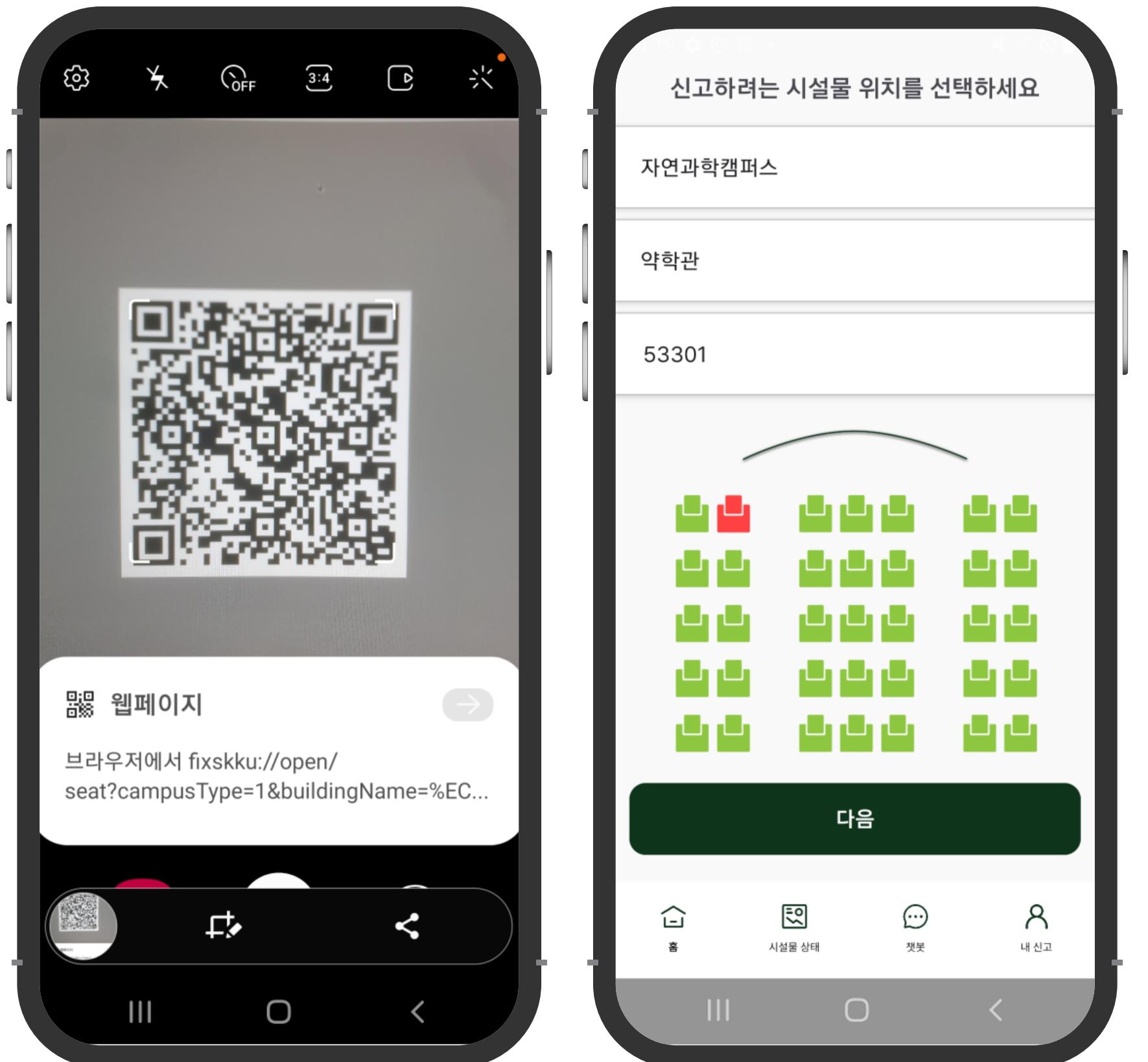
상세 정보 기입

기대 효과

연속적인 ImageView 선택으로 불편함 해소

자신의 위치를 SeatView에서 쉽게 선택 가능

각 자리에 시설물 ID를 부착하여 정보 습득 가능



QR코드 신고 가능

QR코드를 이용한 신고 기능

강의실 시설물에 QR코드를 부착

QR 코드에 캠퍼스명, 건물명, 강의실명, 시설물 ID 정보 포함

어플리케이션에 들어가지 않아도 QR코드로 이동

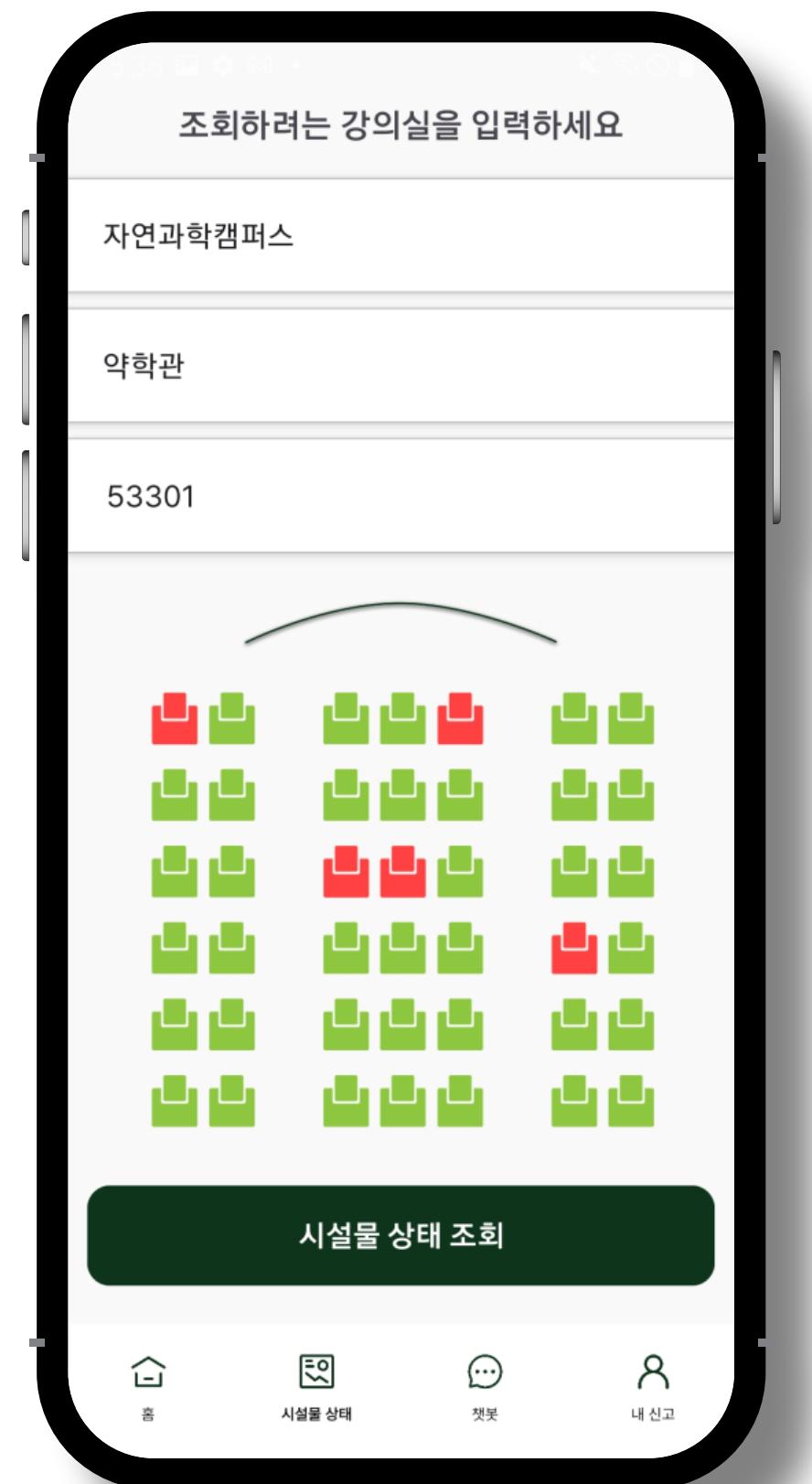
QR코드의 스키마를 인식해서 신고 Activity로 이동

기대 효과

각 시설물의 QR코드 부착을 통해 시설물 ID 부여

어플리케이션을 킬 필요 없이 바로 신고 가능

현재 위치, 시설물 ID에 대한 정보 부재 상황에도 효과적



시설물 조회 기능

강의실 별 시설물 현황 조회 기능

원하는 건물을 선택하여 진행

찾고자 하는 정확한 강의실 정보 입력

현재 강의실 시설물 현황 조회 가능

기대 효과

자신이 사용하려는 시설물을 사전에 확인 가능

시설물 확인으로 학습의 질 향상



내 신고 확인 가능

내 신고 목록

내가 제출한 Report를 한 눈에 확인

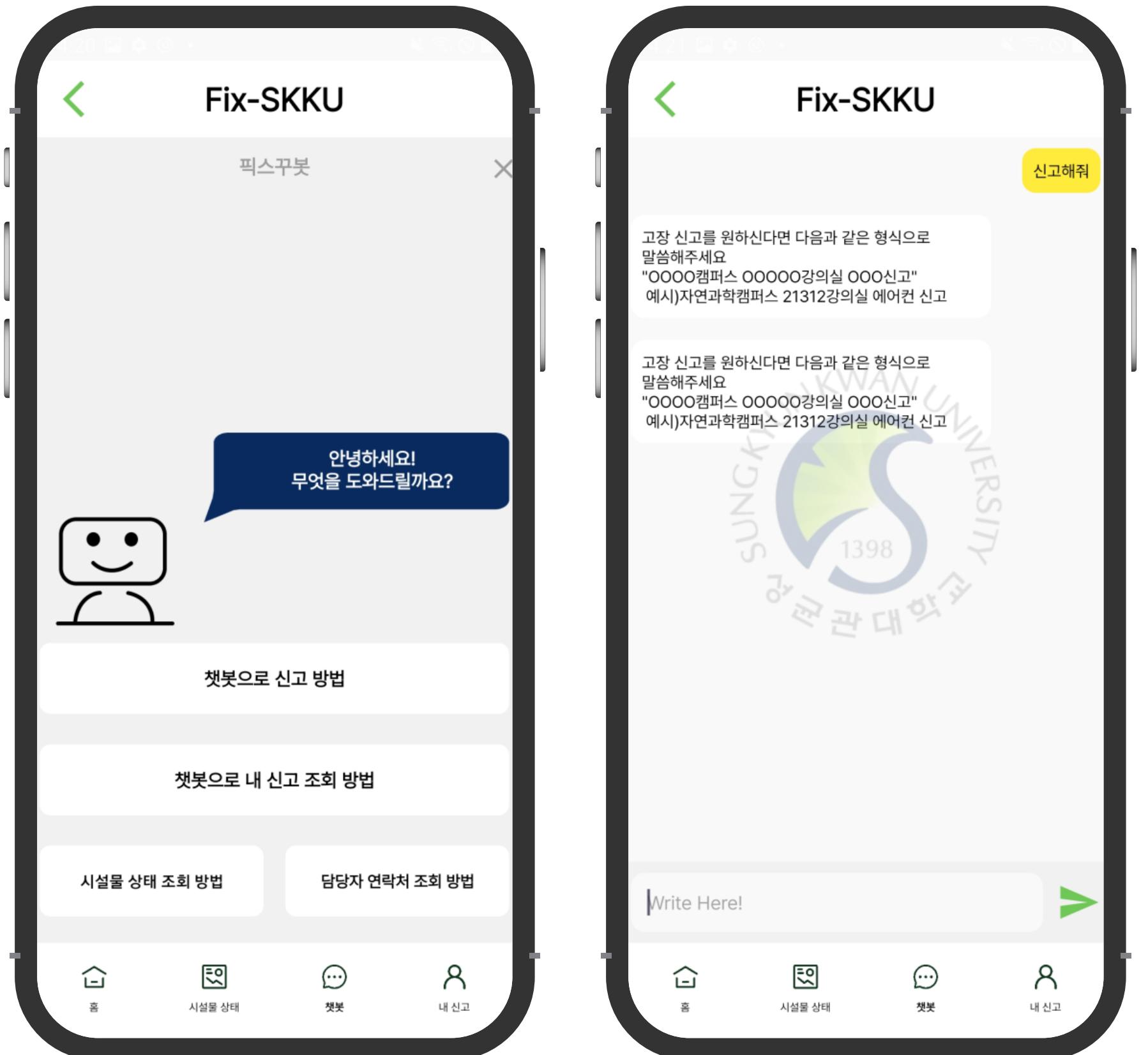
신고 Title, 위치, 시설물 ID 등 한 눈에 확인

현재 신고의 진행 상태 확인 가능

기대 효과

내가 제출한 Report의 진행 사항 Follow

간결한 UI로 신고 목록 확인 용이



챗봇 기능

FAQ 기능

FAQ 이용해서 자주 묻는 질문을 즉각 답변

시설물 신고, 조회, 상태 조회 등 필수적인 기능 제공

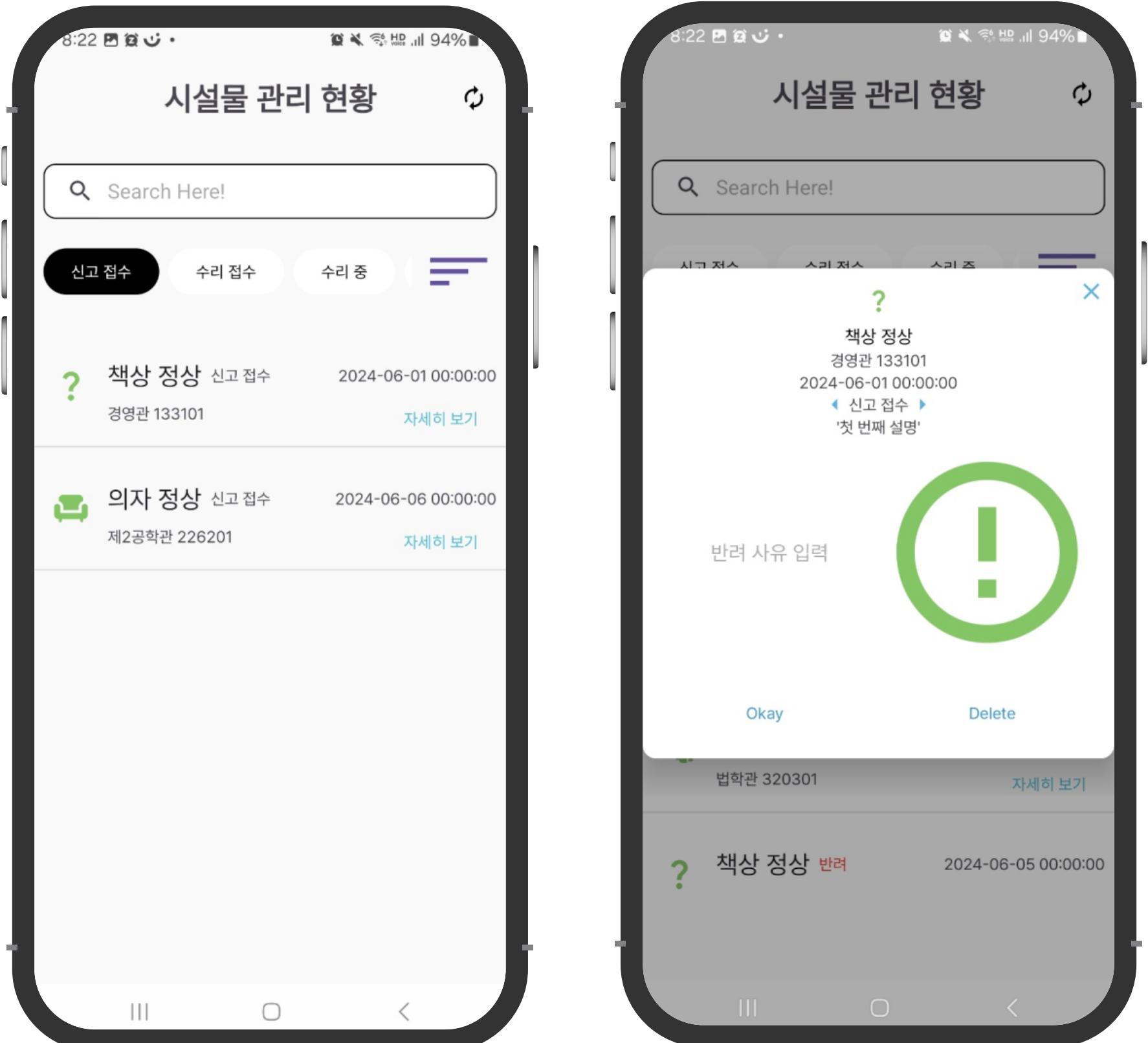
챗봇 AI 기능

사용법을 모르는 User도 쉽게 사용 할 수 있게 하는 기능

정확한 사용법을 몰라도 Chat Bot을 이용한 기능 사용

Chat Bot에 신고 요청

Chat Bot에 조회 요청



관리자 기능

신고 리스트 확인 기능

현재 제출된 Report를 한 눈에 확인

신고 접수, 수리 접수, 수리 중 태그를 활용하여 필터링

신고 접수된 시간순으로 필터링

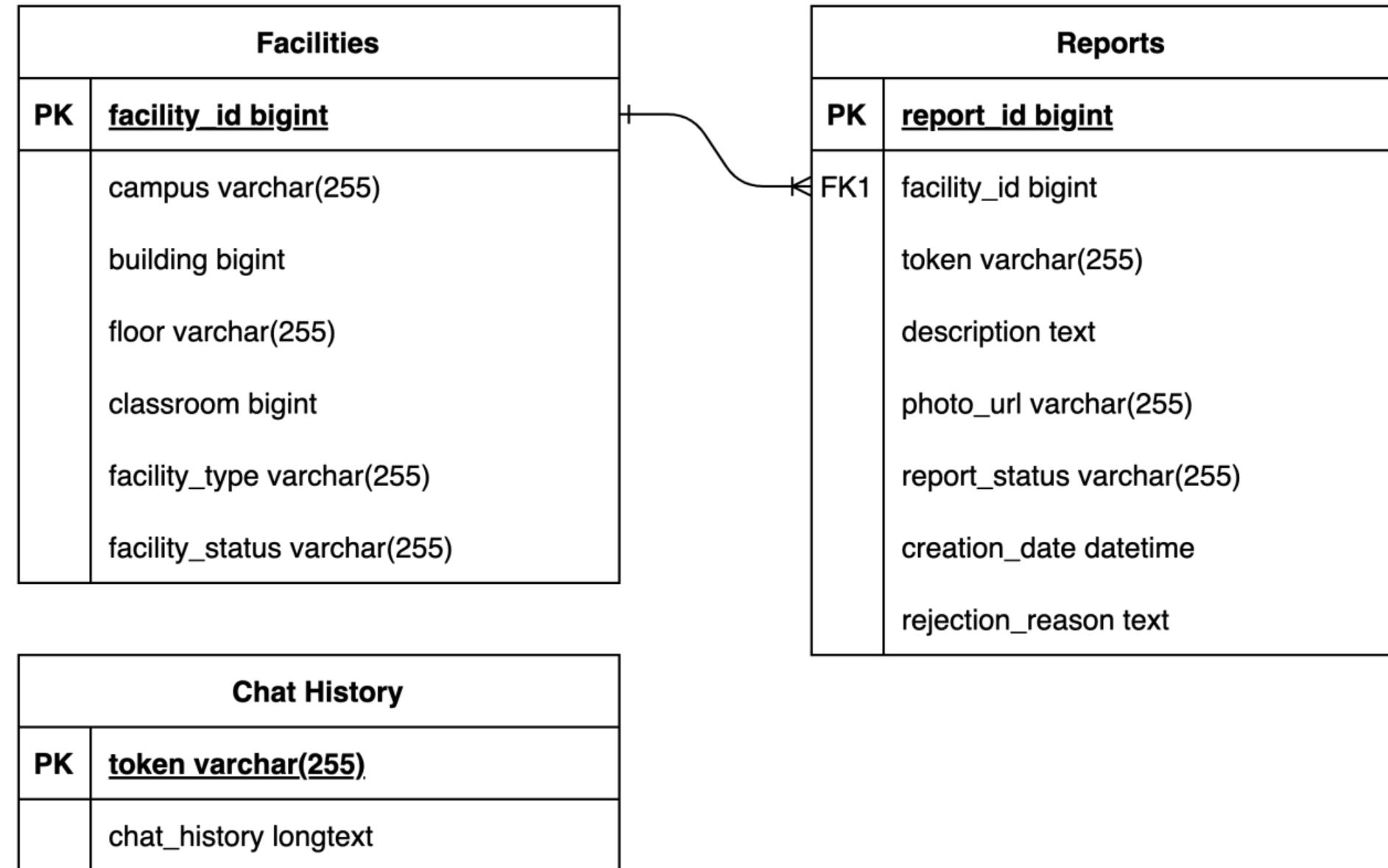
Report 검색 기능

수리 상태 변경 기능

Report 리스트에서 자세히 보기 클릭시 세부 사항 확인

세부 사항 내에서 현재 신고 상태 변경

신고 상태 변경으로 User가 자신의 신고 Follow



데이터베이스

토큰 발급

토큰 발급
토큰 확인

관리자

관리자 신고 리스트 조회
관리자 상세 조회
신고 관리 기능

신고

신고 기능
자신의 신고 조회
자신의 신고 상세조회

챗봇

챗봇 호출 및 결과 반환

시설물

강의실 시설물 현황 조회

상태	요구사항 분류	메서드	API	Aa 기능	설명	책임자
완료	Member	GET	/users/new-token	▣ 토큰 발급		전용호
완료	Member	GET	/users/token	▣ 토큰 확인	앱 실행 시 한번만 실행	전용호
완료	ADMIN	GET	/admin/list	▣ 관리자 신고 리스트 조회		전용호
완료	ADMIN	GET	/admin/{id}	▣ 관리자 신고 상세 조회		전용호
완료	ADMIN	PATCH	/admin/{id}	▣ 신고 관리 기능	신고 현황 수정/ 반려할 경우 반려 사유 입력	전용호
시작 전	REPORT	POST	/report	▣ 신고기능	앱을 통한 사진 업로드 DB 저장 어떻게 할지 생각	이정빈
시작 전	REPORT	GET	/report/list?token=	▣ 자신의 신고 조회 기능		이정빈
시작 전	REPORT	GET	/report/{id}?token=	▣ 자신의 신고 상세조회 기능		이정빈
완료	FAC	GET	/fac/{강의실}	▣ 강의실 시설물 현황 조회 기능	모바일에 URI가 있는지 모르겠지만 URI가 있으면 URI 반환 없으면 특정 식별자 반환(ex 신고 페이지면 1을 조회 페이지면 2를 반환)	전용호
시작 전	CHATBOT	POST	/chatbot	▣ 챗봇 호출(from FE) 및 결과 반환 기능		이정빈



이름 김소공 (22세)

직업 대학생

특징 자연과학캠퍼스 약학과에 재학중이며

수업 시간에 의자나 책상 상태를 신중하게 살펴 수업을 듣는다

평소에 고장 시설물을 신고 하고 싶어도 매번 미루게 된다.

"강의실에 고장난 책상 좀 신고하고 싶은데 너무 귀찮아"

"그리고 매번 고장난 자리에 앉으니까 수업 내내 힘들어"

Scenario

성균관대학생으로 수업을 들으면서 불편한 시설물을 신고하려고 한다.

평상시에 불편함을 많이 겪었지만 신고하는 것이 번거로워 매번 미루게 되었다.

이번에 FIX-SKKU 어플리케이션 런칭 소식을 들었고 어플리케이션을 사용해보려고 한다.

Fix-SKKU

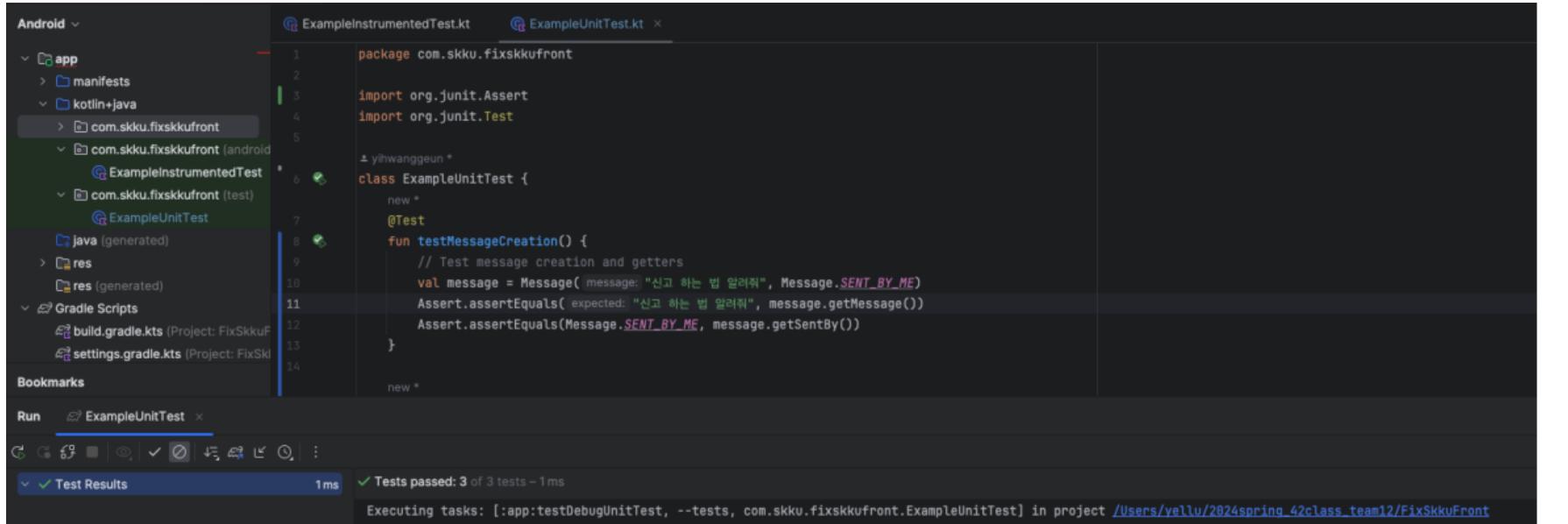
성균관대학교 학생들을 위한 어플리케이션



픽스꾸가 당신의 불편함을 해결해 드리겠습니다

학생으로 로그인

관리자로 로그인

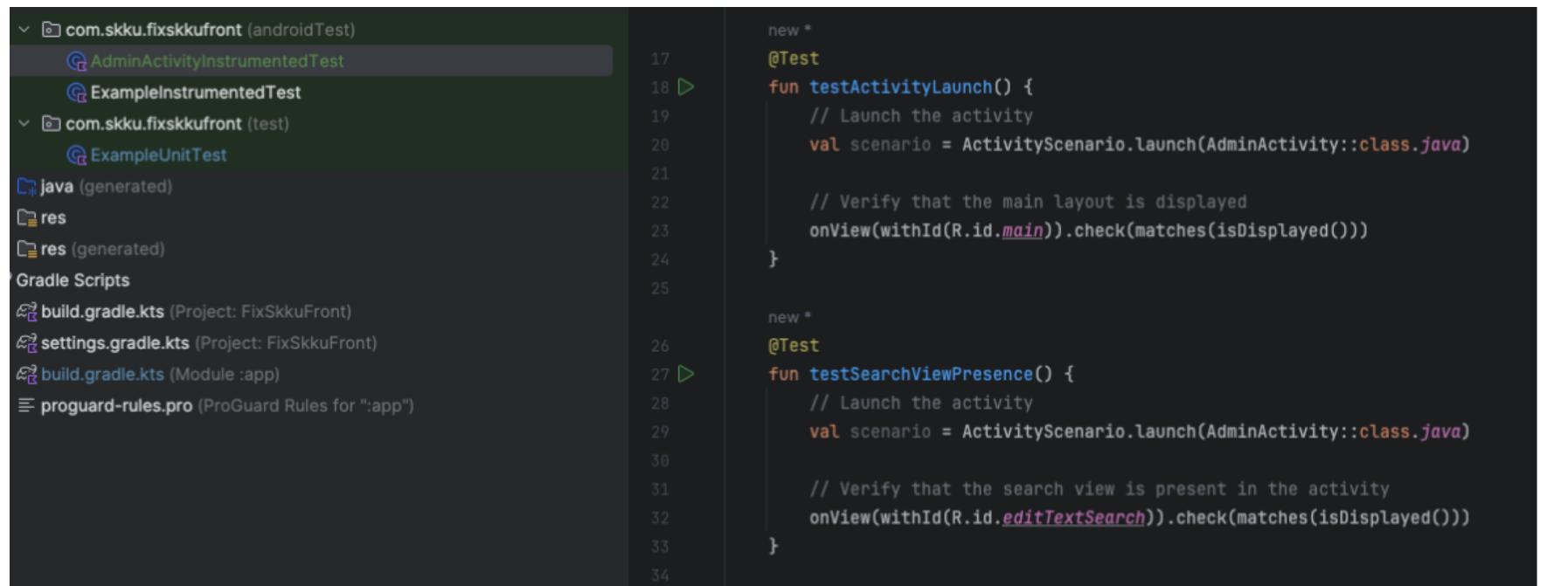


```

Android v
ExampleInstrumentedTest.kt ExampleUnitTest.kt
1 package com.skku.fixskkufront
2
3 import org.junit.Assert
4 import org.junit.Test
5
6 /**
7 * @author yihwanggeun
8 */
9
10 class ExampleUnitTest {
11     new *
12     @Test
13     fun testMessageCreation() {
14         // Test message creation and getters
15         val message = Message("신고 하는 법 알려줘", Message.SENT_BY_ME)
16         Assert.assertEquals(expected: "신고 하는 법 알려줘", message.getMessage())
17         Assert.assertEquals(Message.SENT_BY_ME, message.getSentBy())
18     }
19 }
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34

```

Run ExampleUnitTest
 Test Results 1ms Tests passed: 3 of 3 tests – 1ms
 Executing tasks: [:app:testDebugUnitTest, --tests, com.skku.fixskkufront.ExampleUnitTest] in project /Users/yellu/2024spring_42class_team12/FixSkkuFront



```

com.skku.fixskkufront (androidTest)
  AdminActivityInstrumentedTest
  ExampleInstrumentedTest
com.skku.fixskkufront (test)
  ExampleUnitTest
java (generated)
res
res (generated)
Gradle Scripts
  build.gradle.kts (Project: FixSkkuFront)
  settings.gradle.kts (Project: FixSkkuFront)
  build.gradle.kts (Module :app)
  proguard-rules.pro (ProGuard Rules for ":app")

```

```

17
18     new *
19     @Test
20     fun testActivityLaunch() {
21         // Launch the activity
22         val scenario = ActivityScenario.launch(AdminActivity::class.java)
23
24         // Verify that the main layout is displayed
25         onView(withId(R.id.main)).check(matches(isDisplayed()))
26     }
27
28     new *
29     @Test
30     fun testSearchViewPresence() {
31         // Launch the activity
32         val scenario = ActivityScenario.launch(AdminActivity::class.java)
33
34         // Verify that the search view is present in the activity
35         onView(withId(R.id.editTextSearch)).check(matches(isDisplayed()))
36     }

```

Unit Test (with TC)

Android Unit Test (JUnit)

- 앱의 개별 단위 테스트를 통한 해당 단위 유효성 판단
- 로컬 JVM에서 실행하여 Test
- 데이터 변환, 클래스 테스트

Android Instrumented Test

- 앱의 UI와 전체 기능 테스트
- 실제 안드로이드 기기 또는 에뮬레이터에서 실행
- UI Event에서 유기적 연결 상태 점검

케이스 설명	예상 출력
사용자가 모든 필터를 적용한 다음에 시간 순서로 정렬	필터가 적용된 뒤 정렬됨.
사용자가 반려 사유를 입력하지 않고 확인	반려 사유가 공백인 채로 업데이트됨.
사용자가 필터를 적용 후 신고 목록 업데이트	선택한 신고 목록이 업데이트됨.
사용자가 신고 목록을 업데이트한 뒤에 필터를 적용 및 정렬	신고 목록에 업데이트되고, 필터 및 정렬됨.
서버에서 신고 목록을 업데이트한 뒤 사용자가 새로고침 버튼 클릭	업데이트된 신고 목록이 조회됨.
사용자가 빠른 속도로 FAQ 버튼을 연타	FAQ가 한번만 출력됨
사용자가 메시지를 공백으로 입력	공백으로 입력됨

Unit Test (with TC)

Unit Test (with Test Case)

모든 기능이 정상적으로 작동하는지 확인
직접 Test Case를 만들어서 예상 출력 확인

Chat Bot Test

FAQ 생성
메시지 전송 & 응답
HTTP 통신

시설물 관리자

신고 목록 조회
신고 현황 수정
신고 목록 필터 (상태별)
신고 목록 정렬 (시간순)

```
18  @SpringBootTest
19  @Rollback
20  class ChatbotServiceImplTest {
21
22      @Autowired
23      ChatbotService chatbotService;
24
25      @Autowired
26      ReportRepository repository;
27
28      ResponseEntity<ResponseApi.Body<ToFrontResDto>> responseEntity;
29
30      @Test
31      public void 챗봇을_통한_FAQ_질문_경우() throws Exception{
32          //given
33          String token = "token1";
34          String text = "사용법을 알려줘";
35
36          //when
37          responseEntity = (ResponseEntity<ResponseApi.Body<ToFrontResDto>>) chatbotSe
38          int code = responseEntity.getBody().getCode();
39          String message = responseEntity.getBody().getMessage();
40          ToFrontResDto data = responseEntity.getBody().getData();
41          String response = data.getResponse();
42
43          //then
44          assertEquals(HttpStatus.OK.value(), code);
45          assertEquals("챗봇 일반 질문 응답 성공", message);
46          assertNotNull(response);
47      }
48
49      @Test
50      public void 챗봇을_통한_일반_질문_경우() throws Exception{
51          //given
52          String token = "token1";
53          String text = "사용법을 알려줘";
54
55          //when
56          responseEntity = (ResponseEntity<ResponseApi.Body<ToFrontResDto>>) chatbotSe
57          int code = responseEntity.getBody().getCode();
58          String message = responseEntity.getBody().getMessage();
59          ToFrontResDto data = responseEntity.getBody().getData();
60          String response = data.getResponse();
```

Junit5 사용

스프링부트의 Junit5를 사용

Controller와 Service 클래스를 중심으로
스프링 부트 테스트 작성

아래의 기능들을 바탕으로
통합 UseCase 스프링 부트 테스트 작성

токен 발급 및 확인 기능

관리자 신고 관련 기능

일반 사용자 신고 관련 기능

챗봇 관련 기능

Element	Missed Instructions	Cov.
com.skku.fixskku.report.service	█	100%
com.skku.fixskku.facility.service	█	100%
com.skku.fixskku.token.controller	█	100%
com.skku.fixskku.chatbot.service	██████	97%
com.skku.fixskku.chatbot.constants	█	91%
com.skku.fixskku.common.apipayload	█	90%
com.skku.fixskku.common.domain	█	88%
com.skku.fixskku.token.service	█	87%
com.skku.fixskku.facility.controller	█	77%
com.skku.fixskku.report.domain	█	69%
com.skku.fixskku.facility.domain	█	65%
com.skku.fixskku.admin.controller	█	64%
com.skku.fixskku.report.controller	█	63%
com.skku.fixskku.chatbot.controller	█	47%
com.skku.fixskku.admin.service	█	39%
com.skku.fixskku	█	37%
com.skku.fixskku.report.dto.res	███████████	17%
com.skku.fixskku.admin.dto.req	█	15%
com.skku.fixskku.chatbot.dto.res	███████████	13%
com.skku.fixskku.chatbot.dto.req	█	12%
com.skku.fixskku.report.dto.req	███████████	10%
com.skku.fixskku.facility.dto.res	█	9%
com.skku.fixskku.common.apipayload.exception	█	0%
Total	2,432 of 4,384	44%

Jacoco 사용

Jacoco를 활용한 코드 커버리지 측정

신속한 시설물 고장 신고



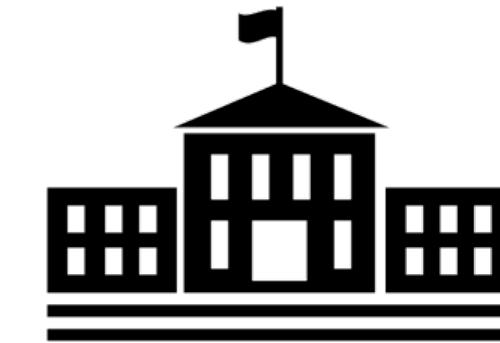
보다 간소화된 절차와 향상된 접근성으로 신속한 대처 가능

안전을 위한 사전 예방



조속한 시설물 고장 수리 및 대처로
학생들의 부상 예방

학업 집중도 향상



학습에 있어 물리적인 장애 환경을
개선함으로써 학업 집중도 향상



Thank you for Watching!

전에 없던 새로운 채팅의 세계, 미리또

