



Fix-SKKU

고장 시설물 신고 관리 어플리케이션

소프트웨어공학개론 Team 12

이항근, 함장건, 이정빈, 전용호, 곽태운, 박소정

목차

| | |
|---|----------|
| 1. Introduction..... | 6 |
| 1.1. Purpose..... | 6 |
| 1.2. Scope..... | 6 |
| 1.3. Definition, Acronyms, and Abbreviation | 7 |
| 1.4. References..... | 8 |
| 1.4.1. IEEE 표준 830 에 대한 가이드 | 8 |
| 1.4.2. SRS 우수 사례..... | 8 |
| 1.4.3. Android 공식 문서..... | 8 |
| 1.5. Overview | 8 |
| 2. Overall Description..... | 9 |
| 2.1. Product Perspective | 9 |
| 2.1.1. Context and Origin..... | 9 |
| 2.1.2. Major Components | 9 |
| 2.1.3. System Interface..... | 10 |
| 2.1.4. User Interface..... | 11 |
| 2.1.5. Hardware Interface..... | 11 |
| 2.1.7. Communication Interface..... | 12 |
| 2.1.7. Memory Constraints | 12 |
| 2.2. Product Function | 12 |
| 2.2.1. 로그인 및 회원가입..... | 12 |
| 2.2.2. 신고 기능 | 13 |
| 2.2.5. 챗봇 기능 | 13 |
| 2.2.6. 강의실 시설물 현황 조회 | 14 |
| 2.2.8. 관리자 신고 현황 정렬 및 검색 기능..... | 14 |
| 2.2.9. 신고 현황 수정 기능 | 14 |
| 2.3. User Classes and Characteristics..... | 15 |
| 2.3.1. 학생..... | 15 |
| 2.3.2. 시설물 관리자 | 15 |
| 2.4. Operating Environment | 15 |
| 2.5. Constraints..... | 15 |
| 2.6. Assumptions and Dependencies..... | 16 |

| | |
|--|-----------|
| 3. Specific Requirement..... | 17 |
| 3.1. External Interface Requirements | 17 |
| 3.1.1 User Interfaces | 17 |
| 3.1.2. Hardware Interfaces..... | 30 |
| 3.1.3. Software Interfaces | 30 |
| 3.1.4 communication Interfaces | 31 |
| 3.2. Functional Requirements..... | 33 |
| 3.2.1. Use case..... | 33 |
| 3.2.2. Use Case Diagram | 42 |
| 3.2.3. Data Dictionary | 43 |
| 3.2.4. Data Flow Diagram..... | 45 |
| 3.3 Performance Requirements | 46 |
| 3.3.1 Static numerical requirements | 46 |
| 3.3.2 Dynamic numerical requirements | 46 |
| 3.4 Logical Database Requirements | 46 |
| 3.5 Design Constraints | 47 |
| 3.6 Standard Compliances..... | 47 |
| 3.7 Software System Characteristics | 47 |
| 3.7.1. Product Requirements | 47 |
| 3.7.2. Organizational Requirements | 48 |
| 3.7.3. External Requirements | 48 |
| 3.8 Organizing Specific Requirements | 49 |
| 3.8.1 Context Model..... | 49 |
| 3.8.2 Process Model..... | 49 |
| 3.8.3 Interaction Model | 50 |
| 3.8.4 Behavior Model | 50 |
| 3.9 System Architecture | 51 |
| 3.10 System Evolution..... | 52 |
| 3.10.1 Limitation and Assumption | 52 |
| 3.10.2 System Evolution | 53 |
| 4. 추가 정보..... | 54 |
| 4.1. 서식 | 54 |
| 4.2. 문서 시간표 | 54 |

표 목차

| | |
|------------------------------------|----|
| 표 1 DEFINITION | 7 |
| 표 2 ACRONYMS AND ABBREVIATION | 7 |
| 표 3 USER INTERFACE - 시작 화면 | 17 |
| 표 4 USER INTERFACE - 캠퍼스 선택(HOME) | 18 |
| 표 5 USER INTERFACE - 사용자 신고 확인 | 19 |
| 표 6 USER INTERFACE - QR 코드 | 20 |
| 표 7 USER INTERFACE - 챗봇 초기 화면 | 21 |
| 표 8 USER INTERFACE - 챗봇 메인 화면 | 22 |
| 표 9 USER INTERFACE - 이용 가능한 시설물 확인 | 23 |
| 표 10 USER INTERFACE - 건물 선택 | 24 |
| 표 11 USER INTERFACE - 강의실 선택 | 25 |
| 표 12 USER INTERFACE - 고장 시설물 선택 | 26 |
| 표 13 USER INTERFACE - 신고 제출 | 27 |
| 표 14 USER INTERFACE - 관리자 비밀번호 로그인 | 28 |
| 표 15 USER INTERFACE - 시설물 관리자 화면 | 29 |
| 표 16 HARDWARE INTERFACE | 30 |
| 표 17 SOFTWARE INTERFACE | 30 |
| 표 18 COMMUNICATION INTERFACE - 사용자 | 31 |
| 표 19 COMMUNICATION INTERFACE - 관리자 | 32 |
| 표 20 사용자의 토큰 확인 케이스 | 33 |
| 표 21 사용자의 토큰 발급 케이스 | 33 |
| 표 22 사용자의 시설물 고장 신고 케이스 | 34 |
| 표 23 QR 코드를 통한 사용자의 시설물 고장 신고 케이스 | 34 |
| 표 24 사용자의 자신의 신고 상세 조회 케이스 | 35 |
| 표 25 강의실 시설물 현황 조회 케이스 | 36 |
| 표 26 챗봇을 이용한 사용자의 시설물 고장 신고 케이스 | 36 |
| 표 27 봇을 통한 사용자의 자신의 신고 조회 케이스 | 37 |
| 표 28 챗봇을 통한 강의실 시설물 현황 조회 케이스 | 38 |
| 표 29 챗봇을 통한 FAQ 조회 케이스 | 38 |
| 표 30 챗봇을 통한 일반 질문 케이스 | 39 |
| 표 31 관리자의 신고 리스트 조회 케이스 | 39 |
| 표 32 표 관리자의 신고 상세 정보 조회 케이스 | 40 |

| | |
|------------------------------------|----|
| 표 33 관리자의 신고 관리 케이스 | 41 |
| 표 34 DATA DICTIONARY - REPORTS | 43 |
| 표 35 DATA DICTIONARY - FACILITIES | 44 |
| 표 36 DATA DICTIONARY - CHATHISTORY | 44 |
| 표 37 문서 시간표 | 54 |

그림 목차

| | |
|--------------------------|----|
| 그림 1 USE CASE 다이어그램 | 42 |
| 그림 2 E-R 다이어그램 | 45 |
| 그림 3 DATA FLOW 다이어그램 | 45 |
| 그림 4 CONTEXT MODEL | 49 |
| 그림 5 PROCESS MODEL | 49 |
| 그림 6 SEQUENCE DIAGRAM | 50 |
| 그림 7 SYSTEM ARCHITECTURE | 51 |

1. Introduction

1.1. Purpose

이 문서는 성균관대학교 시설물 고장 관리 어플리케이션 'FixSkku'에 대한 요구 명세서이다.

FixSkku 는 수업 시간이나 학교 생활에서 발생하는 시설물 고장을 손쉽게 신고하고 관리할 수 있도록 설계되었다. 학생들은 어플리케이션 그리고 QR 코드를 이용하여 손 쉽게 시설물을 신고할 수 있고, 시설물 관리자는 관리자 어플리케이션을 사용하여 신고 내용을 쉽게 조치할 수 있게 한다. 이 어플리케이션을 통해 학생들은 불편함을 해소하여 학업 집중도를 높일 수 있고, 관리자는 효율적으로 시설물을 관리할 수 있다.

이 SRS 문서는 대학 캠퍼스 내에서 학생이 시설 고장을 신고하고 시설 관리자가 관리할 수 있는

어플리케이션의 소프트웨어 요구사항을 설명하며 디자인 및 구현은 본 문서에 따라 진행될 예정이다.

FixSkku 는 성균관대학교 소프트웨어 공학 개론 수업의 12 조 학생이 디자인하고 설계 하였고 성균관대학교 학생과 시설물 관리자들을 주요 사용자로 상정하고 있다. 이 문서의 예상 독자는 소프트웨어 공학 개론 수강 학생과 조교 그리고 교수님이다.

1.2. Scope

학교라는 장소 특성상 공용 공간이라는 특성과 신고 플랫폼의 부재를 문제로 보고, 이를 제공할 수 있는 어플리케이션을 제공하려고 한다. 그리고 관리자 입장에서든 정확한 고장 시설물 위치 판단의 어려움과 정해지지 않은 보고 체계의 문제점을 해결하고자 고장 신고 내용을 쉽게 파악할 수 있는 어플리케이션을 제공하고자 한다. 따라서 FixSkku 를 통해 학생들은 간단한 UI 를 이용하여 고장 시설물을 신고하고 자신의 신고 내용을 팔로우 할 수 있으며 사용 예정인 시설물의 고장 현황을 파악할 수 있게 하고 관리자는 신고 내용을 쉽게 파악하여 손쉽게 수리를 진행을 하는 것을 목표로 한다.

먼저 사용자 기능 중 신고 기능은 캠퍼스, 건물, 강의실로 분류해둔 UI 를 사용하여 처음 보는 사용자도 손 쉽게 신고 페이지에 들어갈 수 있으며 신고 페이지에서는 시설물에 부착 되어있는 시설물 ID 를 사용하여 정확한 신고를 할 수 있게 한다. 그리고 QR 코드 기능을 추가하여 이 어플리케이션에 직접 들어오게 하지 않더라도 QR 코드에 캠퍼스, 건물, 강의실 그리고 시설물 ID 의 정보까지 담아 자동으로 신고 페이지로 이동이 가능하게 한다. 신고 기능을 통해 간소화된 절차와 향상된 접근성을 통한 신속한 대응을 목표로 한다.

두 번째 사용자 기능으로는 시설물 신고 현황 조회 기능을 이용하여 사용자는 서버에 등록되어 있는 신고를 파악하여 자신이 사용하려는 시설물이 문제가 있는지 확인할 수 있다. 이를 통해 시설물의 상태를 사전에 파악하고 필요한 조치를 미리 준비할 수 있게 된다.

마지막 사용자 기능으로는 챗봇 기능은 처음 사용하는 사용자도 쉽게 어플리케이션을 사용하는 것을 목표로 한다. 사용자는 신고 절차에 대한 안내를 받을 수 있으며, 궁금한 점이나 문제 해결 방법에 대해 실시간으로 지원을 받을 수 있다. 이 기능은 사용자 경험을 개선하고 어플리케이션의 접근성을 높이는 데 기여할 것이다.

관리자 기능 중 신고 확인 기능은 사용자가 신고한 내역을 리스트로 한 눈에 파악할 수 있게 하여 신속한 처리를 하는 것을 목표로 한다. 리스트로 구성된 신고 내역을 한 눈에 파악할 수 있고 클릭을 하여 상세 내역을 확인할 수 있으며 진행 사항을 표시하여 신고자 자신의 신고 내역을 팔로우 할 수 있도록 한다.

1.3. Definition, Acronyms, and Abbreviation

표 1 Definition

| 용어 | 설명 |
|---------|---|
| QR 코드 | 정보를 나타내는 매트릭스 형태의 데이터로 간단한 스캔을 통해 다양한 데이터를 얻을 수 있게 해주는 방식 |
| 사용자 | Fix-SKKU 어플리케이션을 사용하는 사람 |
| 시설물 관리자 | 성균관대학교의 시설물을 관리하고 수리해주는 자 |
| 챗봇 | 어플리케이션 사용 중 질문이나 문제 해결을 실시간으로 지원하는 대화형 도구 |

표 2 Acronyms and Abbreviation

| 약어 | 설명 |
|-----|-----------------------------------|
| UI | User Interface |
| DB | Database |
| FAQ | Frequently Asked Questions |
| API | Application Programming Interface |
| RAM | Random Access Memory |

1.4. References

1.4.1. IEEE 표준 830 에 대한 가이드

EEE Std 830-1998 IEEE Recommended Practice for Software Requirements Specifications, In IEEEExplore Digital Library <http://ieeexplore.ieee.org/Xplore/guesthome.jsp>

1.4.2. SRS 우수 사례

Team1. “Project Highlight”. SKKU, Last Modified: Apr 25 2021.
https://github.com/skkuse/2021spring_41class_team1/blob/main/doc/Project%20Highlight_Requirement%202.0.4.pdf

1.4.3. Android 공식 문서

<https://developer.android.com/reference>

1.5. Overview

첫 번째 챕터에서는 Fix-SKKU 애플리케이션의 기본 개요를 다룬다. 여기에는 애플리케이션의 목적, 범위, 문서 표기 규칙, 대상 독자와 읽기 제안, 그리고 참조 자료가 포함된다. 이 챕터는 독자가 명세서의 목적과 구조를 이해하고, 각 섹션의 내용을 효과적으로 파악할 수 있도록 돕는다.

두 번째 챕터는 Fix-SKKU 애플리케이션의 전반적인 개요를 제공한다. 이 챕터에서는 애플리케이션의 전체적인 구조와 동작 방식을 설명하고, 주요 기능들을 명확히 정의한다. 또한, 사용자 그룹과 그들의 특성을 설명하며, 운영 환경과 설계 및 구현 제약 조건을 다룬다. 이 섹션은 제품의 관점, 주요 기능, 사용자 클래스 및 특성, 운영 환경, 설계 및 구현 제약 사항, 사용자 문서, 가정 및 의존성 등을 포괄적으로 다룬다. 이를 통해 독자는 애플리케이션이 어떻게 구성되고, 어떤 방식으로 작동할지에 대한 전체적인 그림을 그릴 수 있다.

세 번째 챕터는 Fix-SKKU 애플리케이션의 구체적인 요구사항을 상세히 기술한다. 여기에는 사용자 인터페이스, 하드웨어 및 소프트웨어 인터페이스, 기능적 요구사항, 비기능적 요구사항, 데이터베이스 구조 및 시스템 아키텍처 등이 포함된다. 이 챕터는 Fix-SKKU 애플리케이션이 제공해야 할 모든 기능과 성능 기준을 명확히 규정하며, 시스템 구현을 위한 구체적인 지침을 제공한다.

이와 같이 구성된 SRS 문서는 Fix-SKKU 애플리케이션 개발 과정에서 발생할 수 있는 모든 요구사항을 명확히 정의하고, 이해관계자 간의 원활한 의사소통을 지원하며, 프로젝트 성공을 위한 기반을 제공한다.

2. Overall Description

2.1. Product Perspective

Fix-SKKU 애플리케이션은 성균관대학교 캠퍼스 내에서 시설 고장을 보고하고 관리하는 과정을 간소화하기 위해 설계되었다. 기존의 수작업 및 비효율적인 시스템을 대체하는 포괄적인 솔루션으로, 학생들과 시설 관리자 모두에게 강력한 플랫폼을 제공한다. 이 섹션에서는 Fix-SKKU 애플리케이션의 맥락, 전체 구조 및 주요 상호작용에 대해 설명한다.

2.1.1. Context and Origin

Fix-SKKU는 성균관대학교의 소프트웨어 공학 개론 수업 프로젝트의 일환으로 고안된 혁신적인 애플리케이션이다. 주요 목적은 학생들이 시설 문제를 보고하고 시설 관리 팀이 이러한 보고서를 효율적으로 관리하고 해결할 수 있도록 하는 것이다. 이 애플리케이션은 사용자 친화적인 인터페이스를 통해 문제를 보고하고, 강력한 백엔드를 통해 이러한 보고서를 추적 및 관리할 수 있도록 하여 학생들과 시설 관리자 간의 격차를 줄이는 것을 목표로 한다.

2.1.2. Major Components

FixSkku 애플리케이션은 기능을 제공하기 위해 상호작용하는 여러 주요 구성 요소로 구성된다

2.1.2.1. 사용자 애플리케이션

학생들이 시설 문제를 보고하는 데 사용하는 프론트엔드 애플리케이션이다. 사용자가 상세한 신고를 제출하고, 신고 상태를 확인하며, 업데이트를 받을 수 있는 직관적인 사용자 인터페이스를 갖추고 있다. 그리고 다양한 시설물에 대한 신고 현황을 조회할 수 있으며 챗봇 기능을 사용하여 사용자 인터랙션을 확보하고 있다.

2.1.2.2. 관리자 애플리케이션

시설 관리자가 학생들이 제출한 신고를 수신, 추적 및 관리하는 데 사용하는 프론트엔드 애플리케이션이다. 시설 관리자는 이 신고를 파악하여 빠른 대응이 가능하게 한다.

2.1.2.3. 백엔드 서버

신고를 처리하고 저장하는 데 필요한 로직 및 데이터베이스 상호작용을 처리한다. 사용자 및 관리자 애플리케이션 간의 원활한 통신을 보장하는 중심 허브 역할을 한다. 그리고 챗봇 AI 서버와 통신하는 과정에서 프론트엔드와 AI 서버의 직접적인 통신을 제거하고 백엔드 서버를 거쳐서 상호작용이 가능하게 한다.

2.1.2.4. 데이터베이스

사용자 정보, 신고, 상태 업데이트 및 모든 활동의 모든 관련 데이터를 저장하는 구조화된 데이터베이스이다.

2.1.2.5. 챗봇

챗봇 AI와 사용자의 대화를 통해 어플리케이션의 사용 방법이나 신고 절차에 대한 안내를 받을 수 있으며 자동 질의 응답을 통해 궁금한 점이나 문제를 실시간으로 도움 받을 수 있다.

2.1.2.6. QR 코드 시스템

캠퍼스 주변의 다양한 시설에 배치된 QR 코드를 스캔하여 학생들이 신고 기능에 빠르게 접근할 수 있도록 한다. 각 QR 코드는 특정 시설에 대한 정보를 포함하여 스캔시 그 정보에 해당하는 시설물 신고 페이지로 이동하여 정확하고 신속한 신고가 가능하도록 한다.

2.1.3. System Interface

User ID 는 간소화를 위해 Device ID 를 이용하여 서버에 전송하고 서버에서 Token 을 발급하여 사용한다. 사용자에게 의해 생성된 시설물 신고는 모두 백엔드 서버로 전송된다. 저장된 신고의 검색과 저장 그리고 관리는 PostgreSQL 을 통해 진행된다. 모든 데이터는 JSON 포맷을 따라 전송되고 각자의 URI 를 통해 쿼리를 생성한다. 그리고 챗봇 AI 서버와 유저 어플리케이션이랑 직접적인 통신을 진행하지 않고 백엔드 서버를 거쳐서 진행된다.

2.1.4. User Interface

2.1.4.1 학생

학생은 총 4 가지 기능을 이용할 수 있다.

하단바에 신고 탭을 누르게 되면 신고 페이지로 넘어가고 캠퍼스, 건물, 층수, 강의실까지 순차적인 버튼 클릭으로 넘어갈 수 있고 상세 신고 페이지에 접근할 수 있다.

하단바에 시설물 현황 탭을 누르게 되면 신고 탭과 유사하게 단계별로 강의실까지 접근할 수 있고 강의실에 들어가게 되면 각 좌석이 표시 되어있는 UI 를 통해 내가 사용하려는 시설물이 현재 사용 가능한지 수리중인지 파악할 수 있다.

하단바에 챗봇 탭을 누르게 되면 챗봇 AI 와 대화를 진행할 수 있다. 기본 대화창으로는 FAQ 내용을 확인할 수 있고 어플리케이션 사용 방법이나 문제 상황이 있을 때 자유롭게 대화를 통해 원하는 정보를 얻을 수 있다.

하단바에 내정보를 누르게 되면 현재 내가 작성한 신고를 확인할 수 있다. 나의 신고가 접수가 되었는지, 진행중인지, 완료가 되어있는지 시설물 관리자가 기입한 내용을 확인할 수 있다.

2.1.4.2 시설물 관리자

현재 접수가 되어있는 신고를 리스트 UI 를 통해 확인할 수 있다. 한 눈에 필수 정보를 먼저 확인하고 관련 리스트 엔트리를 클릭하여 상세 정보를 파악할 수 있다. 상세 정보에서 현재 수리 현황을 기입하여 신고를 한 학생과 소통을 할 수 있다.

2.1.5. Hardware Interface

Fix-SKKU 는 Android Native 플랫폼으로 개발 되어 안드로이드 기반 모바일 기기를 대상으로 개발 되었다. 기기는 최소 2GB 의 RAM 용량과 1.6GHz 프로세서를 내장하고 있어야 한다. 그리고 백엔드 서버와 통신을 위해 WiFi 또는 셀룰러 데이터 연결이 가능한 기기여야한다.

2.1.6. Software Interface

이 어플리케이션은 최소 Android 10.0(SDK 29) 이상을 지원하는 기기에서의 호환을 보장한다. 이 앱은 안드로이드 10.0 버전을 타겟팅하여 구현되었으며, 이를 통해 최신 안드로이드 기능과 보안 개선 사항을 최대한 활용할 수 있다. 현재 iOS 버전은 지원하지 않으며, 앞으로 업데이트에서 iOS 지원 추가를 고려할 수 있다.

2.1.7. Communication Interface

서버와 통신을 위해 WiFi 또는 셀룰러 데이터 연결을 지원해야하며 기기와 서버는 HTTP 프로토콜에 따라 JSON 포맷으로 통신한다.

2.1.7. Memory Constraints

이 어플리케이션을 사용하기 위해서는 최소 2GB 이상 RAM 용량을 요구하며, 어플리케이션 설치 및 실행을 위해 100MB 의 저장 용량을 필요로 한다.

2.2. Product Function

2.2.1. 로그인 및 회원가입

모든 유저는 이 어플리케이션은 별도의 회원가입과 로그인 과정 필요 없이 사용할 수 있다. UUID 를 이용하여 초기에 1 번 서버에 로그인 신청을 하게 되고 서버는 그 UUID 를 기반으로 JWT Token 을 발행해주게 된다. 이 Token 을 어플리케이션 글로벌 변수에 저장하여 한 번 로그인을 진행했으면 별 다른 절차 없이 로그인이 가능하게 해준다.

시설물 관리자는 별도의 관리자 코드를 사용하여 로그인한다. 관리자 코드는 특정 권한과 연결되어 있으며, 이 코드를 입력하는 순간 관리자 인증이 완료된다. 인증된 관리자는 애플리케이션에서 관리자 전용 기능에 접근할 수 있으며, 전체 신고 현황과 같은 관리자 전용 데이터를 조회할 수 있다. 관리자 로그인은 별도의 회원가입 과정 없이 진행되며, 관리자 코드만을 통해 간편하게 접속할 수 있다.

2.2.2. 신고 기능

사용자는 특정 강의실이나 시설물을 선택하여 신고할 수 있다. 사용자가 캠퍼스, 건물, 층수, 강의실을 선택한 후, 의자, 책상, 에어컨, 콘센트 등과 같은 특정 시설물을 선택할 수 있다. 선택된 시설물에 대한 배치도가 제공되며, 고장 유형에 맞는 데이터를 함께 제출할 수 있다. 사용자는 사진과 상세 설명을 추가하여 신고를 완료할 수 있으며, 신고를 완료하면 서버에 JSON 형태로 전달되어 데이터베이스에 저장된다.

2.2.3. QR 를 이용한 신고 유입 기능

QR 코드를 스캔하여 신고 절차를 시작할 수 있는 기능을 제공한다. 이 기능은 딥링크를 활용하여 구현되며, 시설물에 부착되어있는 QR 를 스캔하여 신속하게 신고 절차를 진행할 수 있다.

QR 코드에는 캠퍼스, 건물, 강의실 그리고 시설물 ID 까지 모든 정보를 포함하고 있으며 스캔시 신속하게 위의 정보가 입력되어 신고 페이지로 이동하게 된다.

2.2.4. 신고 조회 기능

사용자는 자신이 제출한 신고 내역을 확인할 수 있다. 이 기능은 사용자의 JWT 토큰을 확인하여 개인 신고 내역만 보여주며, 자동으로 최신순으로 정렬된다. 개인 신고 내역에는 자신이 제출한 신고서의 내용을 확인할 수 있으며 현재 그 신고가 어떻게 처리가 되고 있는지 파악할 수 있다. 이를 통해 사용자는 언제든지 자신의 신고 상태를 확인할 수 있다.

2.2.5. 챗봇 기능

챗봇 기능은 사용자가 어플리케이션 내에서 신속하게 정보를 얻고 상호작용할 수 있게 돕는 도구이다.

이 기능을 통해 사용자는 자주 묻는 질문(FAQ)에 대한 빠른 답변을 받을 수 있으며, 특정 질문에 대해 챗봇이 사전에 정의된 답변을 제공한다. 또한, 사용자가 시설물 고장을 신고하거나 신고한 내역을 조회하고 싶을 때, 챗봇은 이를 지원하는 기능을 수행한다. 사용자는 텍스트 입력을 통해 질문을 하며, 챗봇은 이에 적절한 응답을 구분하여 처리한다. 이 과정에서 챗봇은 백엔드 시스템과 통신하여 필요한 데이터를 처리하고 사용자에게 적합한 정보를 전달한다.

2.2.6. 강의실 시설물 현황 조회

강의실 시설물 현황 조회 기능은 사용자가 강의실의 현재 시설 상태를 실시간으로 확인할 수 있도록 한다. 사용자는 애플리케이션을 통해 강의실을 선택하고, 해당 강의실 내의 모든 시설물의 상태를 시각적으로 확인할 수 있다. 이 기능은 각 시설물의 상태를 색상 또는 아이콘으로 표시하여 사용자가 쉽게 이해할 수 있도록 한다. 예를 들어, 고장 난 의자나 책상은 특정 색으로 표시되어, 사용자가 강의실을 사용하기 전에 시설물의 상태를 파악하고 필요한 조치를 취할 수 있다.

2.2.7. 관리자 신고현황 조회

관리자는 로그인 후 전체 신고 현황을 조회할 수 있다. 이 기능은 관리자가 신고 현황의 전체 목록을 보거나 필터링하여 특정 시간, 수리 단계별(수리중, 수리완료, 반려)로 데이터를 필터링할 수 있다. 그리고 이 전체 목록에서 특정 수리 신고서를 클릭하면 그 수리에 대한 자세한 위치와 내용까지 확인할 수 있다.

2.2.8. 관리자 신고 현황 정렬 및 검색 기능

관리자는 신고 데이터를 시간 순으로 정렬하거나 키워드(건물, 층, 강의실 등)를 기반으로 검색할 수 있다.

2.2.9. 신고 현황 수정 기능

관리자는 수리 단계별로 신고 현황을 수정할 수 있다. 각 신고에 대해서 수리중, 수리완료, 반려 등의 상태를 변경할 수 있으며 상태 변경시 백엔드 서버에 전송되어 신고를 한 사용자에게 즉각적인 공유가 될 수 있게 한다.

2.3. User Classes and Characteristics

2.3.1. 학생

성균관대학교 학생은 이 어플리케이션의 주 사용자로, 학교를 이용하면서 고장난 시설물을 신고하는 데 주로 사용한다. 학생들은 일반적으로 기술에 익숙하고 사용하는 방법을 알지만 복잡한 과정을 거치는 것보다는 직관적인 사용자 인터페이스를 선호한다. 따라서 이 어플리케이션은 직관적인 UI 와 QR 코드를 이용하는 등 학생들이 빠르고 간편하게 문제를 신고할 수 있도록 설계되어야 한다.

2.3.2. 시설물 관리자

관리자는 신고된 내용을 검토하고 처리하는 데 이 어플리케이션을 사용한다. 기존에 존재하던 보고 체계의 어려움과 여러 보고 체계에 따른 수리 시간 지연 문제를 해결하여 효율적인 관리와 신속한 응답을 제공해야 한다. 그리고 별도의 어플리케이션 사용 교육 시간을 줄이기 위해 직관적인 UI 를 사용하여 신고 처리 상태를 업데이트하고 통계를 모니터링 할 수 있게 해야 한다.

2.4. Operating Environment

이 어플리케이션은 Android 기반의 모바일 디바이스에서 운영되며, 최소 요구 사항은 Android 10.0(SDK 29)이다. 이 시스템은 별도의 서버와 연결 되어야 하며, 데이터는 서버와의 안정적인 통신을 위해 HTTP 프로토콜을 사용하여 전송된다.

2.5. Constraints

FixSkku 어플리케이션의 성공적인 구현과 운영을 위해 다음과 같은 제약사항을 설정하였다. 이 제약사항들은 개발 과정 및 어플리케이션 운영에 필수적인 요소로, 반드시 준수되어야 한다.

- 어플리케이션은 사용자 기기의 특정 기능에 접근하기 위해 필요한 권한을 요청해야 하며, 사용자는 이러한 권한을 관리하고 변경할 수 있어야 한다.
- 어플리케이션의 최종 용량은 사용자 기기에 부담을 주지 않도록 합리적인 수준에서 유지되어야 하며 모바일 기기에 영향을 미칠 정도의 용량에 도달했으면 경고를 할 수 있어야 한다.
- 최소 Android Version 10.0(API level 29) 으로 개발한다.
- 코드는 Git 을 이용하여 체계적으로 관리되어야 하며, 명확한 주석과 문서화를 통해 다른 개발자가 쉽게 이해하고 수정할 수 있어야 한다.
- 시스템 아키텍처는 미래의 확장성을 고려하여 모듈식으로 설계되어야 한다.
- 사용자의 신고나 사용자의 정보가 유출되지 않고 보호 되어야 한다.

2.6. Assumptions and Dependencies

사용자를 외부 학생이 아닌 성균관대학교 학생, 그리고 성균관대학교 시설 관리자로 전제한다. 그리고 이 학생들은 모두 Android Version 10.0 이상을 사용하고 iOS 가 아닌 Android 를 사용하는 것을 전제한다. 따라서 iOS 를 사용하거나 낮은 Version 의 Android 를 사용하면 모든 기능을 활용하지 못 할 수도 있다. Wifi 또는 셀룰러 데이터를 원활하게 사용하지 못 할 경우 자신의 신고가 정상적으로 서버에 전송되지 않을 수 있고 누락이 될 수 있다. 또한 카메라를 이용하지 못 하는 상태이거나 카메라 권한을 부여하지 않으면 기획한 사용자 경험을 체험하지 못 할 수 있다.

3. Specific Requirement

3.1. External Interface Requirements

3.1.1 User Interfaces

표 3 User Interface - 시작 화면


| Name | 시작 화면 |
|---------------------------------------|--|
| Description of purpose | Fix-skku 에 접속하면 뜨는 메인 화면, 사용자에게 따라 유저로 입장하거나 관리자로 입장할 수 있다. |
| Input source /Output destination | 사용자 / Android OS 가 장착된 사용자 장치 *Android OS smart phone (Android 10.0 이상) |
| Valid range /Accuracy/Tolerance | N/A |
| Units of measure | Screen |
| Timing | 비동기 사용자 입력/사용자 명령의 즉각적인 실행 |
| Relationship with other input/outputs | 토큰을 전송해 서버에서 토큰이 없는 사용자라면 토큰을 전달하고, 토큰이 있는 사용자라면 서버에서 사용자 데이터를 클라이언트에 전송한다. |
| Format and configuration of screen | <div> <p>1. 사용자는 처음 앱을 실행하면서 입장하기 버튼을 클릭 시 지속 가능한 토큰을 발급받는다.</p> <p>2. 사용자가 관리자 버튼을 클릭 시 비밀번호를 입력해야 입장할 수 있다.</p> </div>  |
| Format and configuration of window | N/A |
| Data formats | Button, Image |
| Command formats | Instruction mapping according to the value of Token |
| End message | N/A |

표 4 User Interface - 캠퍼스 선택(Home)


| Name | 캠퍼스 선택 |
|---------------------------------------|--|
| Description of purpose | 사용자가 시설물 고장 신고를 위해 캠퍼스 위치를 선택하는 화면이다. 하단 바를 통해 홈, 신고 현황, 챗봇, 시설물 조회 화면으로 이동이 가능하다. |
| Input source /Output destination | 사용자 / Android OS 가 장착된 사용자 장치 *Android OS smart phone (Android 10.0 이상) |
| Valid range /Accuracy/Tolerance | N/A |
| Units of measure | Screen |
| Timing | 비동기 사용자 입력/사용자 명령의 즉각적인 실행 |
| Relationship with other input/outputs | N/A |
| Format and configuration of screen | <p>화면 하단의 4 개의 하단바, 화면 상단의 인사캠/자과캠 으로 나뉘어진 캠퍼스 이미지 버튼으로 구성된 화면이다. 하단바를 통해 각 기능(홈, 사용자 신고 목록 확인, 챗봇, 이용 가능한 시설물 확인)을 이용할 수 있는 화면으로 이동 가능하다.</p>  |
| Format and configuration of window | N/A |
| Data formats | Int(Button Code) |
| Command formats | Instruction mapping according to the value of button code |
| End message | N/A |

표 5 User Interface - 사용자 신고 확인


| Name | 사용자 신고 확인 |
|--|---|
| Description of purpose | 사용자는 본인이 신고한 시설물 신고 목록을 확인할 수 있다. |
| Input source /Output destination | 사용자 / Android OS 가 장착된 사용자 장치 *Android OS smart phone (Android 10.0 이상) |
| Valid range /Accuracy/Tolerance | N/A |
| Units of measure | Screen |
| Timing | 비동기 사용자 입력/사용자 명령의 즉각적인 실행 |
| Relationship with other input/outputs | 서버에서 토큰에 맞는 사용자가 신고한 시설물 목록을 반환받는다. |
| Format and configuration of screen | <p>화면 하단의 4 개의 하단바와 화면 상단의 사용자가 신고한 시설물 목록이 나타난다.</p> <p>시설물 목록은 일렬로 표시된다.</p>  |
| Format and configuration of window | N/A |
| Data formats | string(name), string(locate), string(state), time(date), int(Button Code) |
| Command formats | Query |
| End message | N/A |

표 6 User Interface - QR 코드

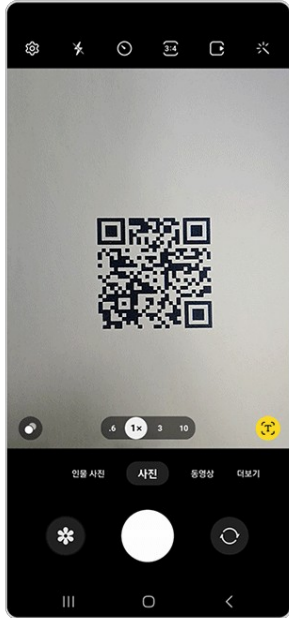
| Name | QR 코드 |
|---------------------------------------|--|
| Description of purpose | QR 코드를 카메라앱으로 촬영하면, 시설물의 위치 정보가 기입된 신고 제출 폼으로 이동. |
| Input source /Output destination | 사용자 / Android OS 가 장착된 사용자 장치 *Android OS smart phone (Android 10.0 이상) |
| Valid range /Accuracy/Tolerance | N/A |
| Units of measure | Screen |
| Timing | 비동기 사용자 입력/사용자 명령의 즉각적인 실행 |
| Relationship with other input/outputs | <p>사용자가 카메라로 QR 코드를 찍으면, 앱으로 시설물 위치 정보가 전송됨.</p>  |
| Format and configuration of screen | N/A |
| Format and configuration of window | N/A |
| Data formats | N/A |
| Command formats | N/A |
| End message | N/A |

표 7 User Interface - 챗봇 초기 화면


| Name | 챗봇 초기 화면 |
|--|---|
| Description of purpose | 사용자가 자주 묻는 질문(FAQ)을 봇에게 물어볼 수 있다. |
| Input source /Output destination | 사용자 / Android OS 가 장착된 사용자 장치 *Android OS smart phone (Android 10.0 이상) |
| Valid range /Accuracy/Tolerance | N/A |
| Units of measure | Screen |
| Timing | 비동기 사용자 입력/사용자 명령의 즉각적인 실행 |
| Relationship with other input/outputs | 자주 묻는 질문 버튼을 클릭하면, 앱에 저장된 답변을 챗봇이 생성한다. |
| Format and configuration of screen | <p>화면의 상단에는 챗봇 화면임을 알려주는 텍스트가 표시된다.</p> <p>화면 중앙에는 챗봇 이미지와 첫 대화 텍스트가 표시된다.</p> <p>그 아래에는 자주 묻는 질문 4 개가 일렬로 표시된다.</p>  |
| Format and configuration of window | 뒤로 이동(“<”) 버튼을 누르면 이전 화면으로 이동한다. |
| Data formats | Int(Button Code), Text |
| Command formats | Instruction mapping according to the value of button code |
| End message | N/A |

표 8 User Interface - 챗봇 메인 화면

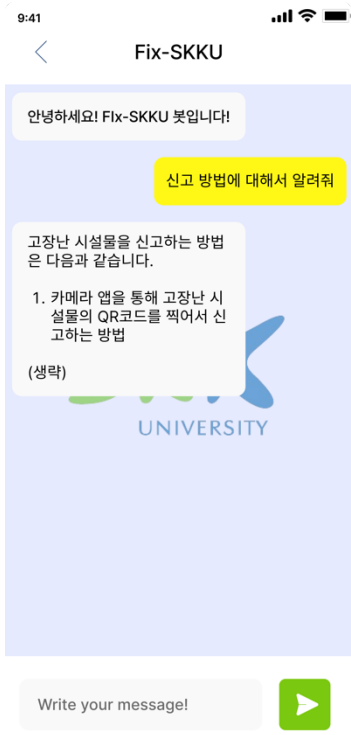
| Name | 챗봇 메인 화면 |
|--|---|
| Description of purpose | 챗봇과 대화를 나눌 수 있는 채팅 화면이다. |
| Input source /Output destination | 사용자 / Android OS 가 장착된 사용자 장치 *Android OS smart phone (Android 10.0 이상) |
| Valid range /Accuracy/Tolerance | N/A |
| Units of measure | Screen |
| Timing | 비동기 사용자 입력/사용자 명령의 즉각적인 실행 |
| Relationship with other input/outputs | 사용자가 입력한 채팅을 서버로 전송한다. 서버에서 전송한 챗봇 답변 메시지를 읽어온다. |
| Format and configuration of screen | <p>화면 맨 하단에 메시지를 입력할 수 있는 Text input 과 전송 버튼이 있다.</p> <p>대화 내용은 상단에서부터 시간 순서대로 나열되어 표시된다.</p> <p>사용자가 보낸 메시지는 우측에, 챗봇이 보낸 메시지는 좌측에 Align 되며 두 박스의 색 또한 다르게 표시된다.</p> <p>메시지 입력 후 전송버튼을 누르면 내용이 서버로 전송된다.</p>  |
| Format and configuration of window | 뒤로 이동("<") 버튼을 누르면 이전 화면으로 이동한다. |
| Data formats | Int(Button Code), Text |
| Command formats | Instruction mapping according to the value of button code |
| End message | N/A |

표 9 User Interface - 이용 가능한 시설물 확인

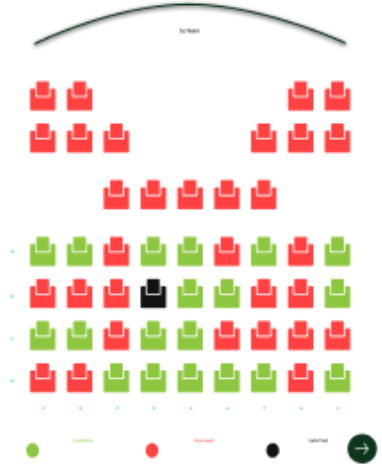
| Name | 이용 가능한 시설물 확인 |
|--|---|
| Description of purpose | 사용자가 강의실의 이용 가능한 시설물을 확인할 수 있다. |
| Input source /Output destination | 사용자 / Android OS 가 장착된 사용자 장치 *Android OS smart phone (Android 10.0 이상) |
| Valid range /Accuracy/Tolerance | N/A |
| Units of measure | Screen |
| Timing | 비동기 사용자 입력/사용자 명령의 즉각적인 실행 |
| Relationship with other input/outputs | 사용자가 강의실을 검색하면, 서버로부터 해당 강의실의 정보를 반환한다. |
| Format and configuration of screen | <p>화면 하단의 4 개의 버튼과 상단에 강의실 위치를 검색하는 서치뷰가 존재한다.</p> <p>사용자가 위치를 검색하면, 화면 중단에 해당하는 강의실의 이용 가능한 시설물이 표시된다.</p> <p>시설물은 Grid 형태로 표시된다.</p>  |
| Format and configuration of window | 뒤로 이동("<") 버튼을 누르면 이전 화면으로 이동한다. |
| Data formats | Int(Button Code), Text |
| Command formats | Instruction mapping according to the value of button code |
| End message | N/A |

표 10 User Interface - 건물 선택


| Name | 건물 선택 |
|--|--|
| Description of purpose | 사용자가 시설물 고장 신고를 위해 건물을 선택하는 화면이다. |
| Input source /Output destination | 사용자 / Android OS 가 장착된 사용자 장치 *Android OS smart phone (Android 10.0 이상) |
| Valid range /Accuracy/Tolerance | N/A |
| Units of measure | Screen |
| Timing | 비동기 사용자 입력/사용자 명령의 즉각적인 실행 |
| Relationship with other input/outputs | N/A |
| Format and configuration of screen | <p>화면 하단에 4 개의 버튼이 존재한다.</p> <p>건물을 선택할 수 있는 이미지 버튼이 Grid 형태로 표시된다.</p> <p>버튼을 클릭하면 해당 건물에 대한 정보를 로컬에 저장한다.</p>  |
| Format and configuration of window | N/A |
| Data formats | Int(Button Code), Text |
| Command formats | Instruction mapping according to the value of button code |
| End message | N/A |

표 11 User Interface - 강의실 선택

| Name | 강의실 선택 |
|---------------------------------------|---|
| Description of purpose | 사용자가 시설물 고장 신고를 위해 강의실을 선택하는 화면이다. |
| Input source / Output destination | 사용자 / Android OS 가 장착된 사용자 장치 *Android OS smart phone (Android 10.0 이상) |
| Valid range/ Accuracy/Tolerance | N/A |
| Units of measure | Screen |
| Timing | 비동기 사용자 입력/사용자 명령의 즉각적인 실행 |
| Relationship with other input/outputs | N/A |
| Format and configuration of screen | 화면 하단에 4 개의 버튼이 존재한다. 강의실을 선택할 수 있는 이미지 버튼이 일렬로 표시된다. 버튼을 클릭하면 해당 강의실에 대한 정보를 로컬에 저장한다. |
| Format and configuration of window | N/A |
| Data formats | Int(Button Code), Text |
| Command formats | Instruction mapping according to the value of button code |
| End message | N/A |

표 12 User Interface - 고장 시설물 선택

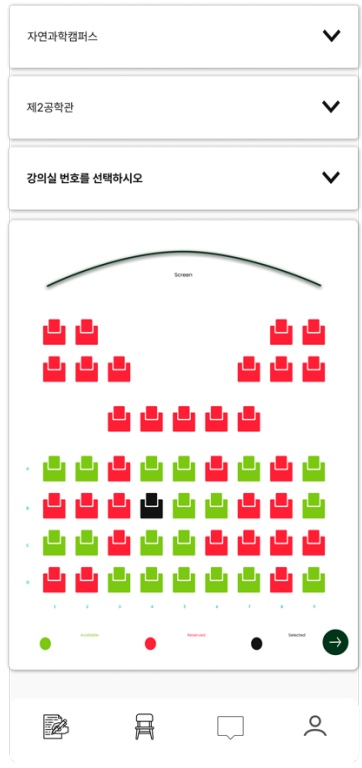
| Name | 고장 시설물 선택 |
|---------------------------------------|--|
| Description of purpose | 사용자가 시설물 고장 신고를 위해 시설물을 선택하는 화면이다. |
| Input source / Output destination | 사용자 / Android OS 가 장착된 사용자 장치 *Android OS smart phone (Android 10.0 이상) |
| Valid range/ Accuracy/Tolerance | N/A |
| Units of measure | Screen |
| Timing | 비동기 사용자 입력/사용자 명령의 즉각적인 실행 |
| Relationship with other input/outputs | N/A |
| Format and configuration of screen | <p>화면 하단의 4 개의 하단바와 화면 상단에 시설물 종류를 선택할 수 있는 Select Box 가 배치된다. 시설물 종류를 선택 시 해당 시설물이 Grid 형태로 표시된다.</p> <p>하단바 오른쪽 위에 우측 화살표 기호 버튼이 표시되어있으며, 버튼을 클릭하면 신고 제출 화면으로 이동한다.</p>  |
| Format and configuration of window | N/A |
| Data formats | Int(Button Code), Text |
| Command formats | Instruction mapping according to the value of button code |
| End message | N/A |

표 13 User Interface - 신고 제출

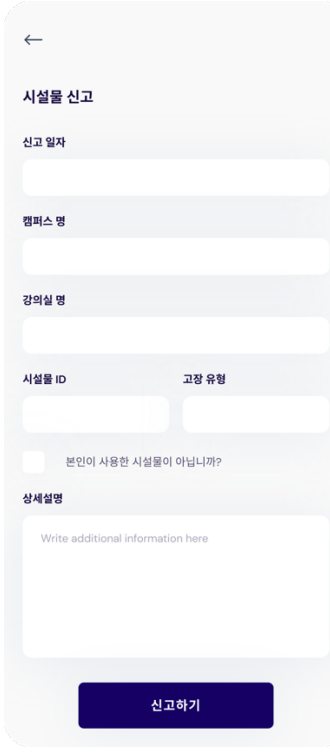
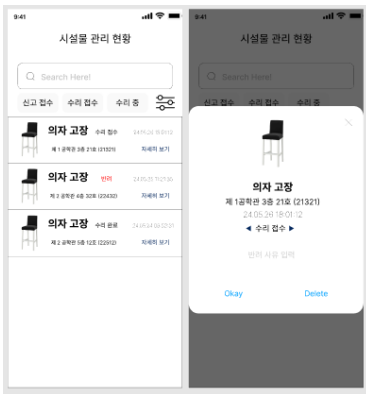
| Name | 신고 제출 |
|---------------------------------------|---|
| Description of purpose | 사용자가 시설물 고장 신고를 제출하는 화면이다. |
| Input source / Output destination | 사용자 / Android OS 가 장착된 사용자 장치 *Android OS smart phone (Android 10.0 이상) |
| Valid range/ Accuracy/Tolerance | N/A |
| Units of measure | Screen |
| Timing | 비동기 사용자 입력/사용자 명령의 즉각적인 실행 |
| Relationship with other input/outputs | 제출 버튼을 누르면 입력된 신고 정보를 서버에 전송한다. |
| Format and configuration of screen | <p>화면은 하단의 하단바와 디테일한 신고 정보 UI로 구성된다.</p> <p>상단에서부터 일자, 캠퍼스, 건물, 강의실, 시설물 ID에 대한 텍스트를 표시한다.</p> <p>또한, 사용자가 입력할 수 있는 상세설명 텍스트칸이 존재한다.</p> <p>그 밑에는 선택적으로 시설물 사진을 첨부할 수 있는 항목이 존재하고, 밑에는 제출 버튼이 존재한다.</p> <p>스크롤뷰로 화면이 구성된다.</p>  |
| Format and configuration of window | N/A |
| Data formats | time(date), string(text), image(option) |
| Command formats | Instruction mapping according to the value of button code |
| End message | N/A |

표 14 User Interface - 관리자 비밀번호 로그인

| Name | 관리자 비밀번호 로그인 |
|---------------------------------------|---|
| Description of purpose | 관리자로 사용자가 로그인하기 위한 화면이다. 사용자 구분없이 관리자는 공통 비밀번호를 공유한다. |
| Input source / Output destination | 사용자 / Android OS 가 장착된 사용자 장치 *Android OS smart phone (Android 10.0 이상) |
| Valid range/ Accuracy/Tolerance | N/A |
| Units of measure | 클릭 한 번 |
| Timing | 비동기 사용자 입력/사용자 명령의 즉각적인 실행 |
| Relationship with other input/outputs | 입력받은 비밀번호를 서버로 전송해 관리자 로그인 성공 여부를 반환 받는다. |
| Format and configuration of screen | 비밀번호를 입력할 수 있는 TextInput 과 입력 정보를 서버로 보낼 수 있는 로그인 버튼, 시작 화면으로 이동하기 위한 뒤로가기 버튼으로 구성된다. |
| Format and configuration of window | N/A |
| Data formats | Int(Button Code) |
| Command formats | Instruction mapping according to the value of a button code |
| End message | N/A |

표 15 User Interface - 시설물 관리자 화면

| Name | 시설물 관리자 화면 |
|---------------------------------------|--|
| Description of purpose | 시설물 신고를 조회할 수 있는 화면이다. |
| Input source / Output destination | 사용자 / Android OS 가 장착된 사용자 장치 *Android OS smart phone (Android 10.0 이상) |
| Valid range/ Accuracy/Tolerance | 시설물 신고 목록 수에 따른 범위 / 사용자의 터치 정확도 / 터치 감도 오차 한계 |
| Units of measure | Screen |
| Timing | 비동기 사용자 입력/사용자 명령의 즉각적인 실행 |
| Relationship with other input/outputs | 사용자가 시설물 신고를 수정하고 Okay 버튼/Delete 버튼을 누르면 수정사항이 서버로 전송되고, 신고 목록이 업데이트 된다. |
| Format and configuration of screen | <p>시설물을 검색할 수 있는 EditText, 시설물 신고 상태 별로 필터링할 수 있는 버튼, 신고 목록을 시간 순서대로 정렬할 수 있는 버튼, 각 신고를 자세히 볼 수 있는 버튼으로 구성된다.</p>  <p>자세히 보기 버튼을 클릭 시 신고 상태를 수정할 수 있는 버튼이 있고, 반려 사유를 적을 수 있는 EditText, 그리고 수정사항을 반영하는 Okay 버튼, 신고를 삭제하는 Delete 버튼이 있다.</p> |
| Format and configuration of window | N/A |
| Data formats | string(name), string(locate), string(state), time(date), int(Button Code) |
| Command formats | Instruction mapping according to the value of a button code |
| End message | N/A |

3.1.2. Hardware Interfaces

표 16 Hardware Interface

| Name | System 에 적합한 기종 |
|-----------------------|---|
| Purpose / Description | 해당 조건을 만족하는 hardware 을 이용하여 사용자들이 시스템의 서비스를 제공받을 수 있도록 한다. / Android OS smart phone (Android 10.0 이상) |

3.1.3. Software Interfaces

표 17 Software Interface

| Name | PostgreSQL Database |
|--|---|
| Purpose / Description | Multimedia / meta data 를 관리하기 위한 Query input/output |
| Input source / Output destination | Host server / User, User, Host Server |
| Valid range/ Accuracy/Tolerance | N/A |
| Units of measure | Query |
| Timing | Instant Reaction |
| Relationship with other input/outputs | Server 의 모든 input/output 과 연관된다. |
| Format and configuration of screen | N/A |
| Format and configuration of window | N/A |
| Data formats | Query |
| Command formats | Query statement |
| End message | N/A |

3.1.4 communication Interfaces

표 18 Communication Interface - 사용자

| Name | Client and host |
|---------------------------------------|---|
| Purpose / Description | 각 Client 는 Host 에게 토큰 발급, 사용자의 신고 내역, 챗봇 호출, 이용 가능한 시설물을 요청한다. Host 는 Client 에게 각 사용자의 신고 내역과 챗봇 결과, 이용 가능한 시설물을 반환한다. |
| Input source / Output destination | Client / Host Server |
| Units of measure | Packet |
| Timing | Instant Reaction |
| Relationship with other input/outputs | Client 와 Server 의 모든 input / output 과 연관된다. |
| Format and configuration of screen | N/A |
| Format and configuration of window | N/A |
| Data formats | Query, text |
| Command formats | Query statement |
| End message | N/A |

표 19 Communication Interface - 관리자

| Name | Client and host |
|---------------------------------------|--|
| Purpose / Description | Client 는 Host 에게 관리자 신고 리스트와 변경 사항을 요청한다. Host 는 Client 에게 관리자 신고 리스트와 변경 사항을 반환한다. |
| Input source / Output destination | Client / Host Server |
| Units of measure | Packet |
| Timing | Instant Reaction |
| Relationship with other input/outputs | Client 와 Server 의 모든 input / output 과 연관된다. |
| Format and configuration of screen | N/A |
| Format and configuration of window | N/A |
| Data formats | Query, text |
| Command formats | Query statement |
| End message | N/A |

3.2. Functional Requirements

3.2.1. Use case

표 20 사용자의 토큰 확인 케이스

| Use Case Name | 사용자의 토큰 확인 |
|----------------|--|
| Actor | 앱을 실행하는 사용자 |
| Description | 사용자가 토큰을 발급 받았는지 확인한다. |
| Normal Course | 1. 사용자가 앱을 실행한다. 2. 앱이 시작되었을 때 사용자의 토큰이 있는지 확인한다. |
| Pre-condition | 앱이 설치되어있어야 한다. |
| Post-condition | - 사용자가 토큰을 발급받지 않았으면 토큰을 발급 받아야한다. |
| Assumptions | 해당 사항 없음 |

표 21 사용자의 토큰 발급 케이스

| Use Case Name | 사용자의 토큰 발급 |
|----------------|--|
| Actor | 사용자 토큰을 발급 받지 않은 사용자 |
| Description | 사용자에게 토큰을 발급한다. |
| Normal Course | 1. 사용자에게 토큰을 발급한다. 2. 사용자의 토큰을 기기 내에 저장한다. |
| Pre-condition | - 앱이 설치되어있어야 한다. - 사용자 토큰이 발급되지 않은 상태여야 한다. |
| Post-condition | - 토큰을 발급 받은 사용자가 신고기능, 챗봇 기능, 자신의 신고 조회 기능을 사용할 수 있어야한다. |
| Assumptions | 해당 사항 없음 |

표 22 사용자의 시설물 고장 신고 케이스

| Use Case Name | 시설물 고장 신고 |
|----------------|---|
| Actor | 토큰을 발급받은 사용자 |
| Description | 사용자는 고장난 시설물을 신고한다. |
| Normal Course | <ol style="list-style-type: none"> 1. 홈 탭에서 캠퍼스를 선택한다. 2. 해당 캠퍼스의 건물을 선택한다. 3. 해당 건물의 강의실을 선택한다. 4. 강의실의 배치도에서 해당 시설물을 클릭한다. 5. 신고 페이지 폼에 맞게 고장 유형(필수), 상세설명 텍스트(선택), 사진(선택)을 입력한 뒤 제출버튼을 누른다. |
| Pre-condition | <ul style="list-style-type: none"> - 앱이 설치되어있어야 한다. - 사용자 토큰 이미 발급된 상태여야 한다. |
| Post-condition | <ul style="list-style-type: none"> - 제출 버튼을 누르면 신고 정보가 DB 에 저장되어야 한다. - 제출 버튼을 누르면 신고 탭으로 이동해야한다. - 뒤로가기 버튼을 누르면 뒤로 갈 수 있어야한다. |
| Assumptions | 해당 사항 없음 |

표 23 QR 코드를 통한 사용자의 시설물 고장 신고 케이스

| Use Case Name | QR 코드를 통한 시설물 고장 신고 |
|----------------|--|
| Actor | 토큰을 발급받은 사용자 |
| Description | 사용자는 고장난 시설물을 신고한다. |
| Normal Course | <ol style="list-style-type: none"> 1. 강의실에서 QR 코드를 카메라로 인식하여 신고 페이지 폼에 진입한다. 2. 신고 페이지 폼에 맞게 고장 유형(필수), 상세설명 텍스트(선택), 사진(선택)을 입력한 뒤 제출버튼을 누른다. |
| Pre-condition | <ul style="list-style-type: none"> - 앱이 설치되어있어야 한다. - 사용자 토큰 이미 발급된 상태여야 한다. |
| Post-condition | <ul style="list-style-type: none"> - 제출 버튼을 누르면 신고 정보가 DB 에 저장되어야 한다. - 제출 버튼을 누르면 신고 탭으로 이동해야한다. - 뒤로가기 버튼을 누르면 뒤로 갈 수 있어야한다. |
| Assumptions | 시설물에 QR 코드가 부착되어있어야함 |

표 24 사용자의 자신의 신고 상세 조회 케이스

| Use Case Name | 자신의 신고 상세 조회 |
|----------------|--|
| Actor | 토큰을 발급받은 사용자 |
| Description | 사용자는 자신이 신고한 시설물을 상세 조회한다. |
| Normal Course | <ol style="list-style-type: none"> 신고 탭을 클릭하여 신고 탭에 진입한다. 자신의 신고 리스트에서 조회를 원하는 신고 항목을 찾는다 <ol style="list-style-type: none"> 검색어를 입력하여 검색할 수 있다 필터를 통해 수리 현황에 따라 필터링하여 신고 항목 리스트를 볼 수 있다.(수리 현황 - 신고접수/수리접수/수리중/수리완료/반려) 자신의 신고 리스트에서 조회를 원하는 신고 항목을 클릭한다. 사용자는 신고 상세 조회 페이지에서 신고 정보를 확인할 수 있다. <ol style="list-style-type: none"> 신고 정보 <ol style="list-style-type: none"> 일자 캠퍼스 건물 강의실 고장 유형 시설물 ID 고장 상세 설명 텍스트(선택) 사진(선택) |
| Pre-condition | <ul style="list-style-type: none"> - 앱이 설치되어있어야 한다. - 사용자 토큰 이미 발급된 상태여야 한다. - 사용자가 신고를 한 상태여야한다. |
| Post-condition | 뒤로가기 버튼을 누르면 뒤로 갈 수 있어야한다. |
| Assumptions | 해당 사항 없음 |

표 25 강의실 시설물 현황 조회 케이스

| Use Case Name | 강의실 시설물 현황 조회 |
|----------------|---|
| Actor | 모든 사용자 |
| Description | 사용자는 강의실 시설물 현황을 조회한다. |
| Normal Course | <ol style="list-style-type: none"> 1. 강의실 시설물 현황 조회 탭을 클릭하여 진입한다. 2. 해당 캠퍼스의 건물을 선택한다. 3. 해당 건물의 강의실을 선택한다. 4. 강의실 시설물 현황 조회 페이지에서 강의실 배치도 안의 고장 신고된 시설물들은 X 자 표시가 되어 나타난다. |
| Pre-condition | 앱이 설치되어있어야 한다. |
| Post-condition | 뒤로가기 버튼을 누르면 뒤로 갈 수 있어야한다. |
| Assumptions | 해당 사항 없음 |

표 26 챗봇을 이용한 사용자의 시설물 고장 신고 케이스

| Use Case Name | 챗봇을 이용한 사용자의 시설물 고장 신고 |
|----------------|--|
| Actor | 토큰을 발급받은 사용자 |
| Description | 사용자는 챗봇을 이용하여 고장난 시설물을 신고한다. |
| Normal Course | <ol style="list-style-type: none"> 1. 챗봇 탭을 클릭하여 진입한다. 2. 챗봇에서 제시하는 예시 형식 대로 고장 신고 텍스트를 입력하고 전송 버튼을 클릭한다. 3. 입력된 정보를 바탕으로 신고 페이지로 이동한다. 4. 신고 페이지 폼에 맞게 고장 유형(필수), 상세설명 텍스트(선택) , 사진(선택)을 입력한 뒤 제출버튼을 누른다. |
| Pre-condition | <ul style="list-style-type: none"> - 앱이 설치되어있어야 한다. - 사용자 토큰 이미 발급된 상태여야 한다. |
| Post-condition | <ul style="list-style-type: none"> - 제출 버튼을 누르면 신고 정보가 DB 에 저장되어야 한다. - 제출 버튼을 누르면 신고 탭으로 이동해야한다. - 뒤로가기 버튼을 누르면 뒤로 갈 수 있어야한다. |
| Assumptions | 해당 사항 없음 |

표 27 봇을 통한 사용자의 자신의 신고 조회 케이스

| Use Case Name | 챗봇을 통한 사용자의 자신의 신고 상세 조회 |
|----------------|--|
| Actor | 토큰을 발급받은 사용자 |
| Description | 사용자는 챗봇을 이용하여 자신이 신고한 시설물을 상세 조회한다. |
| Normal Course | <ol style="list-style-type: none"> 1. 챗봇 탭을 클릭하여 진입한다. 2. 자신의 신고를 조회하고 싶다는 텍스트를 입력하고 전송 버튼을 클릭한다. 3. 자신의 신고 리스트에 진입한다. 4. 자신의 신고 리스트에서 조회를 원하는 신고 항목을 찾는다 <ol style="list-style-type: none"> a. 검색어를 입력하여 검색할 수 있다. b. 필터를 통해 수리 현황에 따라 필터링하여 신고 항목 리스트를 볼 수 있다.(수리 현황 - 신고접수/수리접수/수리중/수리완료/반려) 5. 자신의 신고 리스트에서 조회를 원하는 신고 항목을 클릭한다. 6. 사용자는 신고 상세 조회 페이지에서 신고 정보를 확인할 수 있다. <ol style="list-style-type: none"> a. 신고 정보 <ol style="list-style-type: none"> i. 일자 ii. 캠퍼스 iii. 건물 iv. 강의실 v. 고장 유형 vi. 시설물 ID vii. 고장 상세 설명 텍스트(선택) viii. 사진(선택) |
| Pre-condition | <ul style="list-style-type: none"> - 앱이 설치되어있어야 한다. - 사용자 토큰 이미 발급된 상태여야 한다. - 사용자가 신고를 한 상태여야한다. |
| Post-condition | 뒤로가기 버튼을 누르면 뒤로 갈 수 있어야한다. |
| Assumptions | 해당 사항 없음 |

표 28 챗봇을 통한 강의실 시설물 현황 조회 케이스

| Use Case Name | 챗봇을 통한 강의실 시설물 현황 조회 |
|----------------|---|
| Actor | 토큰을 발급받은 사용자 |
| Description | 사용자는 챗봇을 이용하여 강의실 시설물 현황을 조회한다. |
| Normal Course | <ol style="list-style-type: none"> 1. 챗봇 탭을 클릭하여 진입한다. 2. 시설물 현황을 조회하고 싶다는 텍스트를 강의실의 번호를 포함하여 입력하고 전송 버튼을 클릭한다. 3. 강의실 시설물 현황 조회 탭의 강의실 시설물 현황 조회 페이지에서 강의실 배치도 안의 고장 신고된 시설물들은 X 자 표시가 되어 나타난다. |
| Pre-condition | <ul style="list-style-type: none"> - 앱이 설치되어있어야 한다. - 사용자 토큰 이미 발급된 상태여야 한다. |
| Post-condition | 뒤로가기 버튼을 누르면 뒤로 갈 수 있어야한다. |
| Assumptions | 해당 사항 없음 |

표 29 챗봇을 통한 FAQ 조회 케이스

| Use Case Name | 챗봇을 통한 FAQ 조회 |
|----------------|---|
| Actor | 토큰을 발급받은 사용자 |
| Description | 사용자는 챗봇을 이용하여 FAQ 를 조회할 수 있다. |
| Normal Course | <ol style="list-style-type: none"> 1. 챗봇 탭을 클릭하여 진입한다. 2. 챗봇이 제시하는 FAQ 버튼 중 사용자가 궁금한 FAQ 버튼을 클릭한다. 3. 챗봇이 해당 FAQ 의 내용을 보여준다. |
| Pre-condition | <ul style="list-style-type: none"> - 앱이 설치되어있어야 한다. - 사용자 토큰 이미 발급된 상태여야 한다. |
| Post-condition | 뒤로가기 버튼을 누르면 뒤로 갈 수 있어야한다. |
| Assumptions | FAQ 가 미리 등록되어 있어야한다. |

표 30 챗봇을 통한 일반 질문 케이스

| Use Case Name | 챗봇을 통한 일반 질문 답변 |
|----------------|--|
| Actor | 토큰을 발급받은 사용자 |
| Description | 사용자는 챗봇을 이용하여 일반적인 질문에 답변을 받을 수 있다. |
| Normal Course | 1. 챗봇 탭을 클릭하여 진입한다. 2. 사용자가 원하는 질문 텍스트를 입력하고 전송 버튼을 클릭한다. 3. 챗봇이 이에 대한 답변을 보여준다. |
| Pre-condition | - 앱이 설치되어있어야 한다. - 사용자 토큰 이미 발급된 상태여야 한다. |
| Post-condition | 뒤로가기 버튼을 누르면 뒤로 갈 수 있어야한다. |
| Assumptions | 해당 사항 없음 |

표 31 관리자의 신고 리스트 조회 케이스

| Use Case Name | 관리자의 신고 리스트 조회 |
|----------------|--|
| Actor | 관리자 |
| Description | 관리자가 전체 신고 리스트를 조회한다 |
| Normal Course | 1. 관리자가 화면의 특정 부분을 클릭한 후 비밀번호를 입력하여 신고 리스트 화면에 진입한다. 2. 전체 신고 리스트에서 조회를 원하는 항목을 찾을 수 있다. <ol style="list-style-type: none"> 검색어를 입력하여 검색할 수 있다. 필터를 통해 수리 현황에 따라 필터링하여 신고 항목 리스트를 볼 수 있다.(수리 현황 - 신고접수/수리접수/수리중/수리완료/반려) 날짜를 통해 필터링하여 리스트를 볼 수 있다. |
| Pre-condition | - 앱이 설치되어있어야 한다. - 관리자가 비밀번호 인증 단계를 통과한 상태여야한다. |
| Post-condition | - 뒤로가기 버튼을 누르면 뒤로 갈 수 있어야한다. - 원하는 신고의 상세정보를 조회 할 수 있어야 한다. |
| Assumptions | 사용자들이 신고를 완료한 상태임을 가정한다. |

표 32 표 관리자의 신고 상세 정보 조회 케이스

| Use Case Name | 관리자의 특정 신고 상세 정보 조회 |
|----------------|---|
| Actor | 관리자 |
| Description | 관리자는 특정 신고에 대한 정보를 상세 조회한다. |
| Normal Course | <ol style="list-style-type: none"> 1. 관리자가 비밀번호를 입력하여 신고 리스트 조회 화면에 진입한다. 2. 전체 신고 리스트에서 조회를 원하는 신고 항목을 찾는다. <ol style="list-style-type: none"> a. 검색어를 입력하여 검색할 수 있다. b. 필터를 통해 수리 현황에 따라 필터링하여 신고 항목 리스트를 볼 수 있다.(수리 현황 - 신고접수/수리접수/수리중/수리완료/반려) c. 날짜를 통해 필터링하여 리스트를 볼 수 있다. 3. 신고 리스트에서 조회를 원하는 신고 항목을 클릭한다. 4. 사용자는 신고 상세 조회 페이지에서 신고 정보를 확인할 수 있다. <ol style="list-style-type: none"> a. 신고 정보 <ol style="list-style-type: none"> i. 일자 ii. 캠퍼스 iii. 건물 iv. 층수 v. 강의실 vi. 고장 유형 vii. 시설물 ID viii. 신고 진행 상태 ix. 고장 상세 설명 텍스트(선택) x. 사진(선택) xi. 거절 이유(선택) |
| Pre-condition | <ul style="list-style-type: none"> - 앱이 설치되어있어야 한다. - 관리자가 비밀번호 인증 단계를 통과한 상태여야한다. |
| Post-condition | <ul style="list-style-type: none"> - 뒤로가기 버튼을 누르면 뒤로 갈 수 있어야한다. - 신고 진행 상태를 변경할 수 있어야 한다. - 상세 설명 및 거절 이유를 작성할 수 있어야 한다. |
| Assumptions | 사용자들이 신고를 완료한 상태임을 가정한다. |

표 33 관리자의 신고 관리 케이스

| Use Case Name | 관리자의 특정 신고 관리 |
|----------------|--|
| Actor | 관리자 |
| Description | 관리자는 특정 신고에 대해 신고 진행 상태 및 반려 사유를 수정 할 수 있다. |
| Normal Course | <ol style="list-style-type: none"> 관리자는 특정 신고에 해당하는 시설물에 대한 수리 현황을 업데이트 할 수 있다 <ol style="list-style-type: none"> 수리 현황 - 신고접수/수리접수/수리중/수리완료/반려 관리자의 판단 하에 수리가 불필요하거나 불가 할때 반려 사유를 업데이트하여 사용자에게 전달 할 수 있다. |
| Pre-condition | <ul style="list-style-type: none"> - 앱이 설치되어있어야 한다. - 관리자가 비밀번호 인증 단계를 통과한 상태여야한다. |
| Post-condition | 뒤로가기 버튼을 누르면 뒤로 갈 수 있어야한다. |
| Assumptions | 사용자들이 신고를 완료한 상태임을 가정한다. |

3.2.2. Use Case Diagram

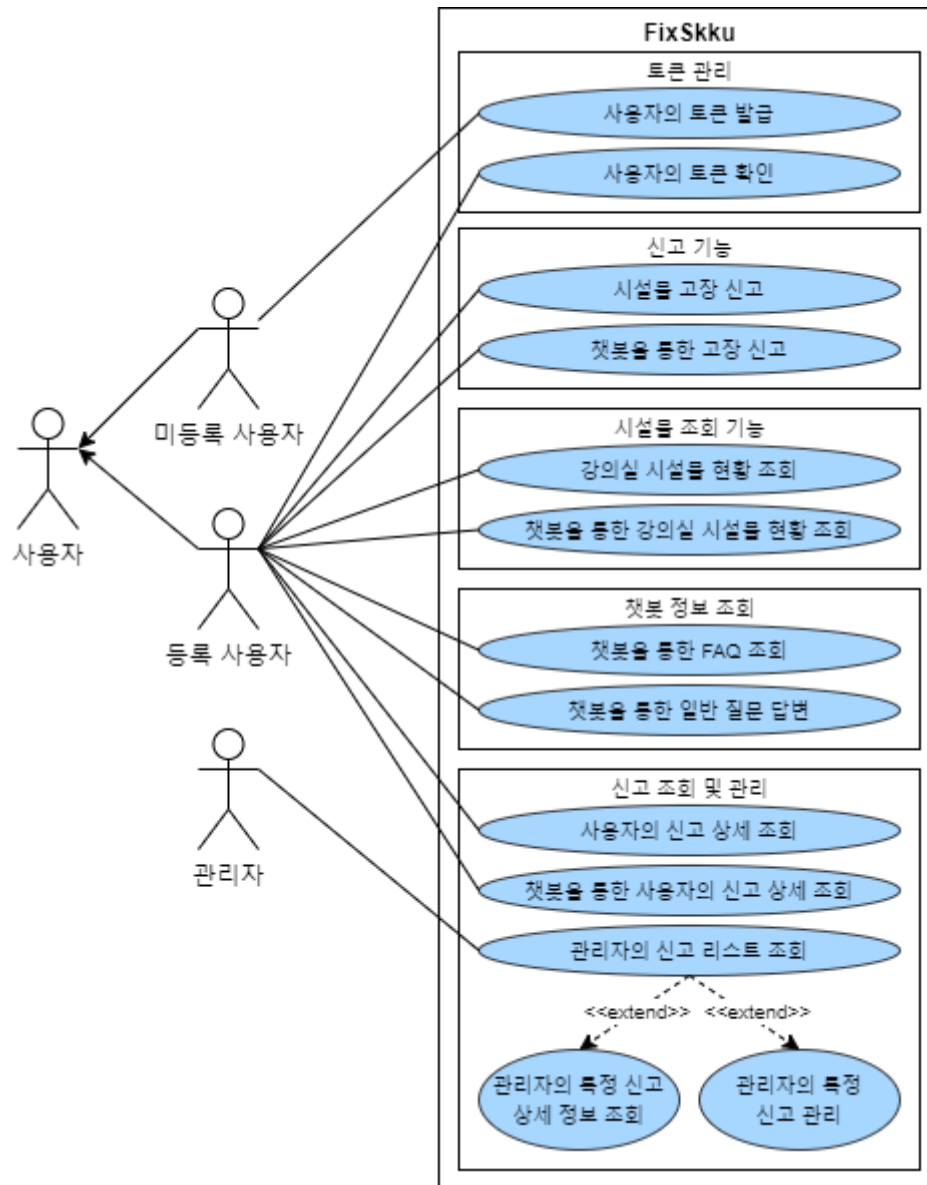


그림 1 Use Case 다이어그램

3.2.3. Data Dictionary

표 34 Data Dictionary - Reports

| Field | Primary | Foreign | Type | Description |
|------------------|---------|---------|--------------|-------------|
| report_id | O | | BIGINT | 신고 id |
| facility_id | | O | BIGINT | 시설물 id |
| token | | | VARCHAR(255) | 토큰 |
| description | | | TEXT | 설명 |
| photo_url | | | VARCHAR(255) | 사진 URL |
| report_status | | | VARCHAR(255) | 신고 상태 |
| creation_date | | | DATETIME | 생성 날짜 |
| rejection_reason | | | TEXT | 거절 이유 |

표 35 Data Dictionary - Facilities

| Field | Primary | Foreign | Type | Description |
|-----------------|---------|---------|--------------|-------------|
| facility_id | O | | BIGINT | 시설물 id |
| campus | | | VARCHAR(255) | 캠퍼스 이름 |
| building | | | BIGINT | 건물 번호 |
| floor | | | VARCHAR(255) | 층수 |
| classroom | | | BIGINT | 강의실 번호 |
| facility_type | | | VARCHAR(255) | 시설 유형 |
| facility_status | | | VARCHAR(255) | 시설 상태 |

표 36 Data Dictionary - ChatHistory

| Field | Primary | Foreign | Type | Description |
|--------------|---------|---------|--------------|-------------|
| token | O | | VARCHAR(255) | 토큰 |
| chat history | | | LONGTEXT | 채팅 기록 |

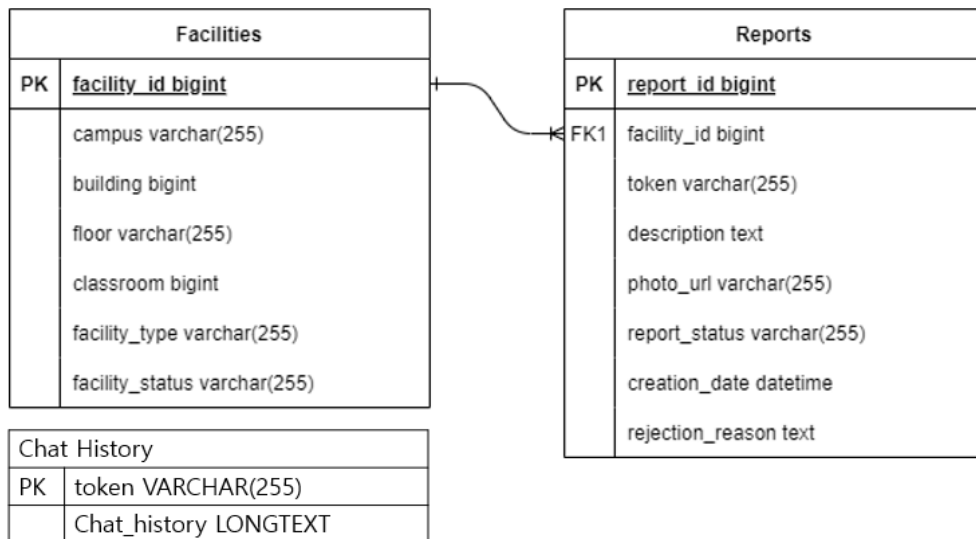


그림 2 E-R 다이어그램

3.2.4. Data Flow Diagram

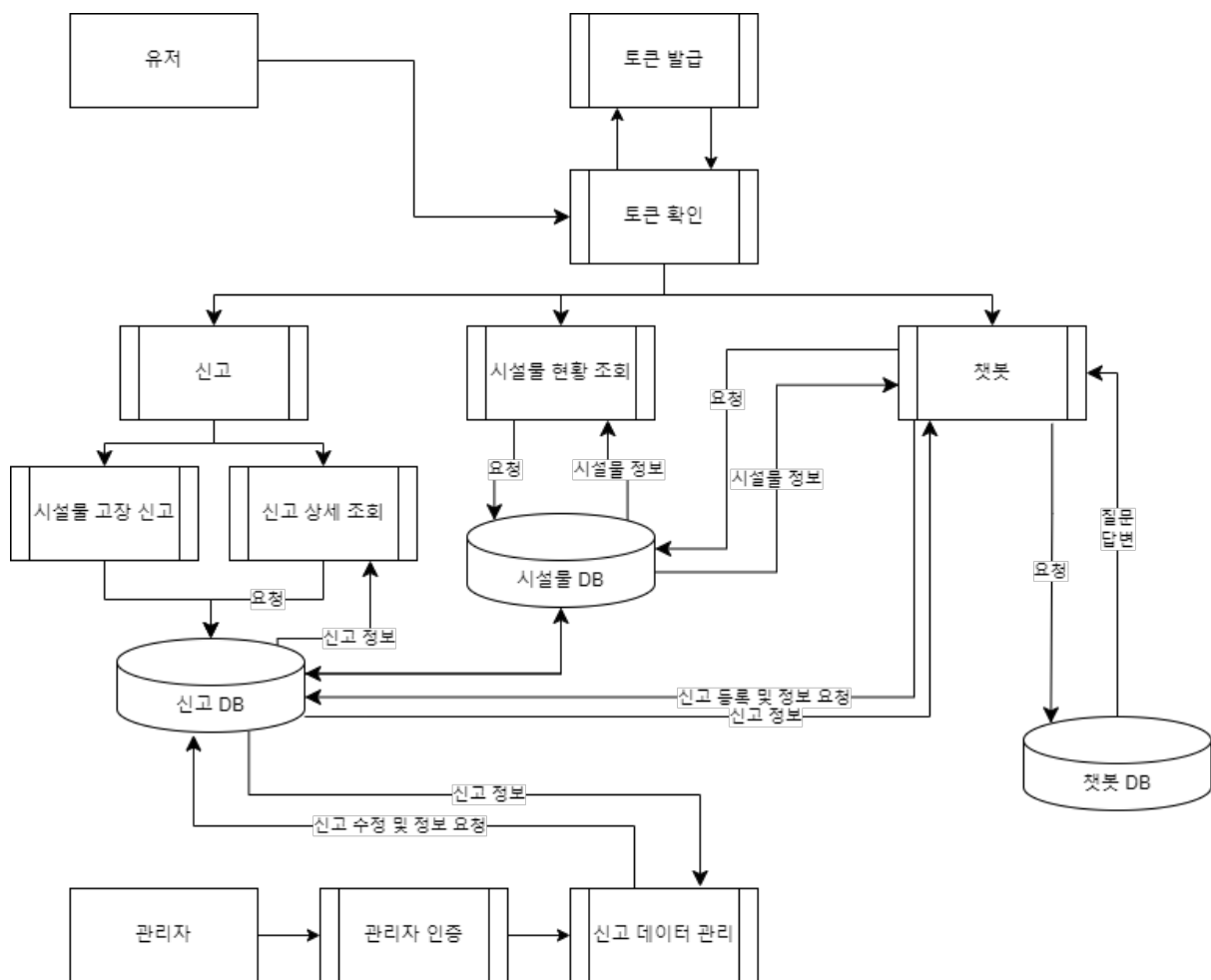


그림 3 Data Flow 다이어그램

3.3 Performance Requirements

Fix-SKKU 어플리케이션의 성능 요구사항은 사용자의 원활한 경험을 보장하기 위해 설정되었다. 이 요구사항들은 어플리케이션의 초기 설계 단계에서 추정된 것으로, 개발 과정에서 변경될 수 있다. 주요 성능 지표로는 응답 시간, 처리 속도, 시스템 가용성 등이 포함된다. 또한, 다양한 사용 시나리오와 예상되는 사용자 수를 고려하여 최적화가 필요하다. 최종 성능 요구사항은 테스트와 검증을 통해 확정될 예정이다.

3.3.1 Static numerical requirements

- 시스템은 최대 100 명의 동시 사용자를 지원하여 여러 요청을 효율적으로 처리할 수 있다.
- 시스템은 최소 2GB 의 RAM 과 100MB 이상의 여유 메모리를 가진 모바일 장치에서 최적화되어 실행된다.
- 최신 버전의 안드로이드 (10.0 이상)와 호환된다.
- 권장 장치에서 애플리케이션은 3 초 이내에 로드되어야 한다.
- 시스템은 99.5%의 가동 시간을 유지하여 높은 가용성을 보장해야 한다.

3.3.2 Dynamic numerical requirements

- 시스템은 최대 300 명의 동시 사용자를 최적으로 처리하고 최소 10,000 명의 회원을 관리할 수 있어야 한다.
- 애플리케이션 내의 모든 검색 기능은 2 초 이내에 결과를 제공해야 한다.
- 사용자 인증 과정은 1 초 이내에 완료되어야 한다.
- 실시간 알림은 이벤트 발생 후 1 초 이내에 사용자에게 전달되어야 한다.
- 시스템은 성능 저하 없이 피크 시간 동안 사용자 부하가 50% 증가하는 것을 처리할 수 있어야 한다.

3.4 Logical Database Requirements

시스템은 PostgreSQL 데이터베이스를 통해 사용자 정보와 시설물 신고 내역을 관리한다. 사용자 토큰과 신고 내역은 안전하게 저장되고 효율적으로 조회되어 원활한 애플리케이션 기능을 보장한다.

데이터베이스는 높은 성능을 위해 최적화되어 있으며, 정보의 빠른 접근과 처리를 가능하게 한다. 데이터의 무결성과 보안이 유지되어 사용자 정보를 보호하고 신뢰할 수 있는 운영을 보장한다. 또한, 시스템은 사용자 수와 시설물 신고가 증가함에 따라 늘어나는 기록을 처리할 수 있도록 설계되어야 한다.

3.5 Design Constraints

시스템은 사용된 모든 구성 요소와 라이브러리가 오픈 소스 라이선스에 준수하거나 적절한 라이선스 계약을 갖추고 있어야 한다. 애플리케이션은 Android 및 iOS 운영 체제를 사용하는 다양한 모바일 장치에서 접근할 수 있어야 한다. 시스템 관리자는 전용 관리자 애플리케이션을 통해 시스템에 접근하고 관리할 수 있어야 한다. 사용자 및 시설물 수리 정보를 저장하는 데이터베이스는 PostgreSQL 을 사용해야 하며, 호환성과 최적의 성능을 보장해야 한다. 또한, 시스템은 사용자 기반 및 데이터 양의 미래 성장을 수용할 수 있도록 확장성을 염두에 두고 설계되어야 한다.

3.6 Standard Compliances

시스템의 모든 프로그램은 Kotlin 표준에 따라 작성되었다. 애플리케이션은 Android Studio 를 사용하여 개발되어 Android 장치와의 호환성을 보장한다. 인공지능 서버는 Python 의 FastAPI 를 사용하여 구축되었으며, Python 코딩 표준을 준수한다. 모든 구성 요소는 성능과 보안을 위한 업계의 모범 사례 및 관련 표준을 준수하도록 설계되었다.

3.7 Software System Characteristics

시스템에서 충족해야 할 기준과 특성들 정의하며 제품이 어떻게 작동해야 하는지에 대한 요구사항들을 규정한다.

3.7.1. Product Requirements

다음 요구사항은 추정에 기반하고 있으며, 어플리케이션 개발이 완료됨에 따라 변경될 수 있다. "Fix-SKKU" 어플리케이션의 효과적인 기능과 사용자 만족을 위해 필수적인 제품 요구사항을 아래와 같이 정리하였다.

3.7.1.1. Usability Requirement

사용자가 직관적인 UI 와 아이콘으로 별도의 설명 없이 서비스를 사용할 수 있어야 한다.

즉, 쉽고 간단한 인터페이스로 사용성이 좋아야 하고 쓰기 편해야 한다. 시스템에서는 소비자가 클릭하기 편하도록 클릭할 수 있는 토큰 인터페이스를 크게 만들었다. 그리고 더 간편하고 빠른 이용을 위해 챗봇을 만들어 사용자들이 제품을 사용하는데 있어서 오류를 발생시키지 않을 수 있도록 사용성들에 대한 직관적인 이용사항을 다루고 있다.

3.7.1.2. Performance Requirement

제품의 작동속도가 빨라야 한다. 앱이 실행되면 3 초 안에 메인 화면에 진입해야 하며 로그인 프로세스는 2 초 내에 완료되어야 한다. 한 화면에서 다른 화면으로 진입할 때는 1 초 내에 완료되어야 한다.

모든 자료 업로드는 최소 10Mbps 최대 100Mbps 의 속도를 지원해야 하며 자료 다운로드는 최소 10Mbps 최대 200Mbps 의 속도를 지원해야 한다. 실시간 공유 문서는 최대 1 초의 딜레이가 허용되어야 한다.

시스템에서 우선순위 필터 시간순서, 수리 전/중/후/반려에서도 쓰이는 매핑 알고리즘은 사용자에게 결과를 5 초 이내에 제공하는 것을 목표로 한다. 매핑알고리즘은 시스템에서 가장 시간이 많이 소요되는 작업으로, 너무 많으면 사용성을 저하시킬 수 있다.

3.7.1.3. Security Requirement

보안상의 문제 해결을 위해 시스템에서는 시작 시 입장하기(사용자)/관리자로 나누어서 사용자는 MEMDI_LOGIN_USER 사용자 토큰 발급하고 관리자는 MEMDI_LOGIN_ADMIN 모달에 관리자 코드 입력 하여 구분 하도록 한다. 또한 신고한 당사자만 수정 가능할 수 있도록 하여 혼동을 막는다.

3.7.2. Organizational Requirements

이 섹션은 정책과 관련된 다양한 요구사항을 다룬다. "Fix-SKKU" 어플리케이션의 성공적인 구현과 운영을 위해 조직 내에서 준수해야 할 필수적인 요구사항들을 포함한다.

3.7.2.1. Environmental Requirement

시스템에서는 외부 API 인 GPT 와 LLAMAF 를 가져와 서비스 정보를 주고받을 수 있도록 해준다.

3.7.3. External Requirements

이 섹션은 외부 요인에 대한 요구사항을 다룬다. "Fix-SKKU" 어플리케이션의 효과적인 운영을 위해 외부 환경과의 상호 작용 및 관련 외부 기준을 충족시키는 것이 필수적이다.

3.7.3.1. Safety Requirement

시스템은 외부 시스템 사용자가 시스템에 접근할 수 없도록 설계되어야 한다. 또한, 외부 충격으로부터 데이터가 손상되지 않도록 안전이 제공되어야 한다.

3.8 Organizing Specific Requirements

3.8.1 Context Model

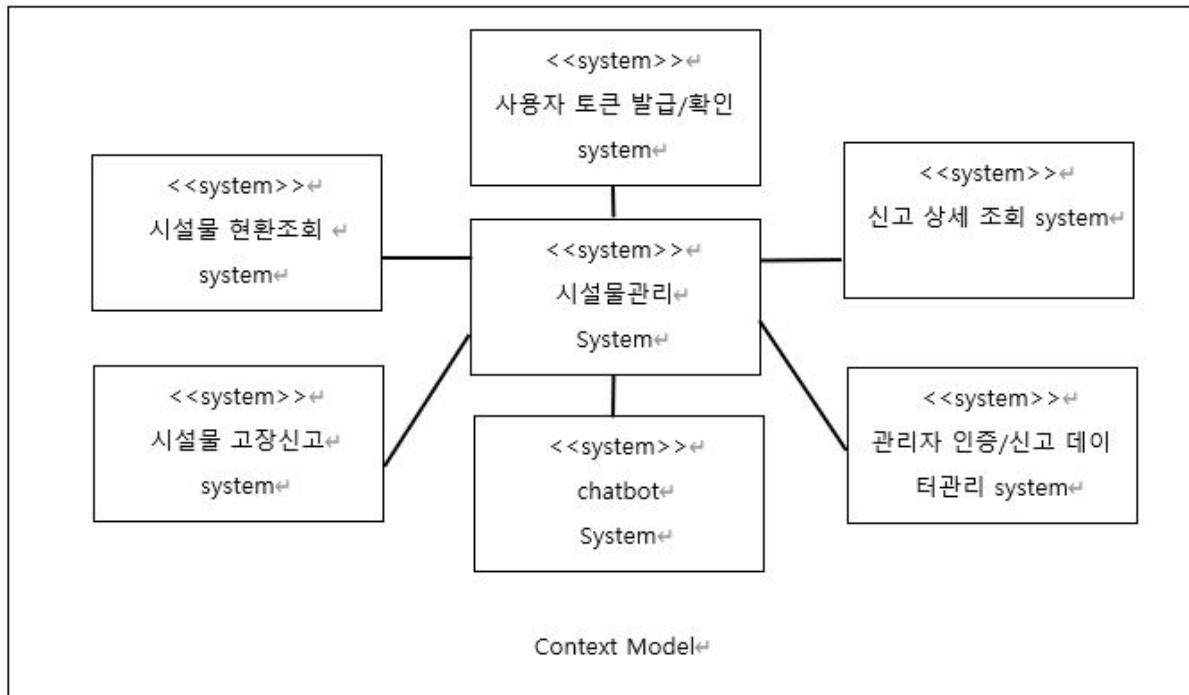


그림 4 Context Model

3.8.2 Process Model

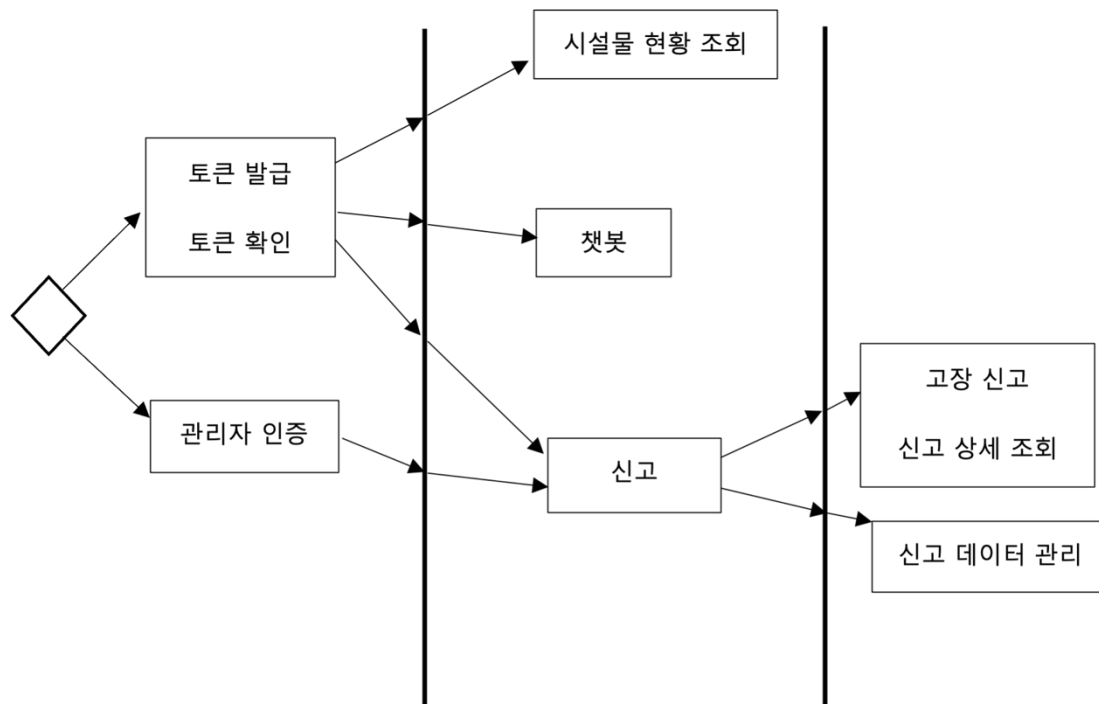


그림 5 Process Model

3.8.3 Interaction Model

다음을 참고하시오 3.2.2 Use Case Diagram

3.8.4 Behavior Model

행동 모델은 시스템을 통해 데이터가 어떻게 이동하고, 다양한 구성 요소가 시간에 따라 어떻게 상호작용하는지를 보여준다. 이 섹션에서는 데이터 흐름도(DFD)와 시퀀스 다이어그램을 사용하여 애플리케이션의 행동 모델을 설명한다. 데이터 흐름도는 여러 프로세스와 데이터 저장소 간의 정보 흐름을 나타낸다. 시퀀스 다이어그램은 사용자, 기본 서버, 인공지능 서버 간의 상호작용 순서를 보여준다. 이러한 다이어그램은 시스템의 동적 행동과 상호작용 패턴을 명확하게 이해하는 데 도움을 준다.

3.8.4.1 Sequence Diagram

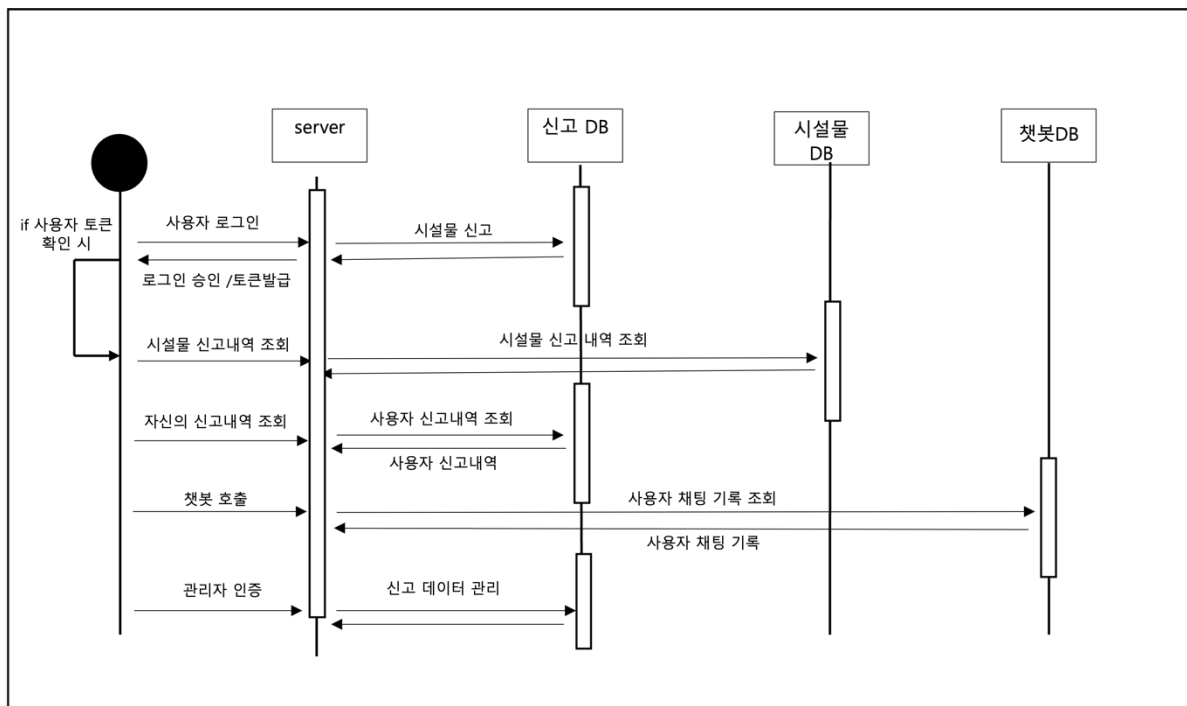


그림 6 Sequence Diagram

3.9 System Architecture

이 섹션은 Three-Tier Architecture 를 사용하여 시스템 모듈간의 관계와 기능을 설명한다.

애플리케이션은 프레젠테이션 계층, 애플리케이션 계층, 데이터 계층으로 구성된다.

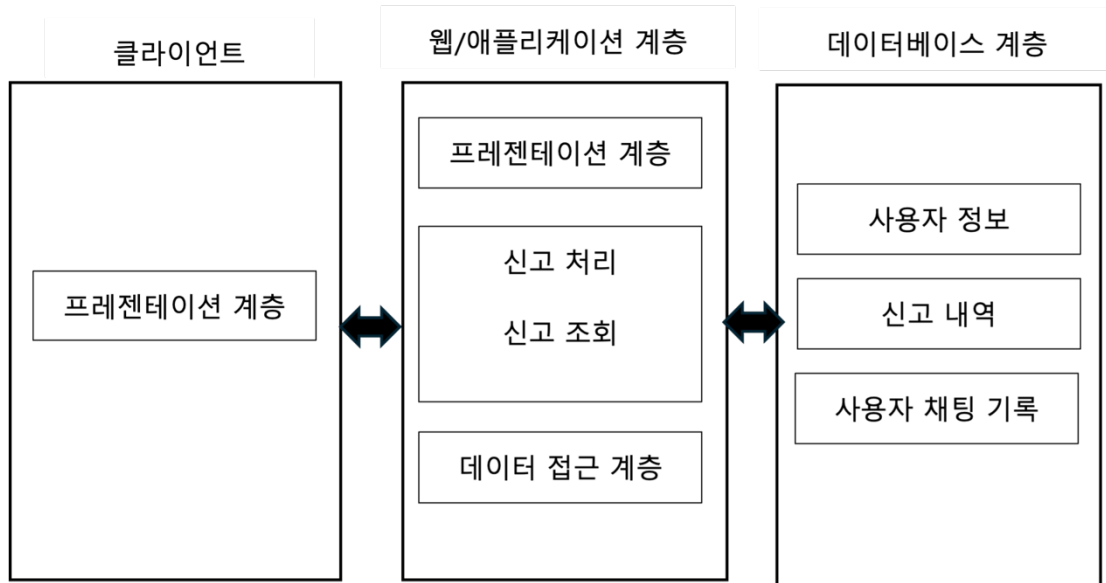


그림 7 System Architecture

1. 프레젠테이션 계층: 사용자 인터페이스를 담당하며, 사용자가 애플리케이션과 상호작용할 수 있도록 한다. 모바일 애플리케이션이 이 계층에 속한다.
2. 애플리케이션 계층: 비즈니스 로직을 처리하고, 프레젠테이션 계층과 데이터 계층 간의 중재 역할을 한다. 기본 서버가 이 계층에 속하며, 인공지능 서버도 이 계층에 포함되어 사용자 요청을 처리하고 필요한 연산을 수행한다.
3. 데이터 계층: 사용자 정보와 시설물 수리 내역을 저장하는 데이터베이스를 포함한다. PostgreSQL 데이터베이스가 이 계층에 속한다. 인공지능 서버는 기본 서버를 통해 이 데이터베이스에 접근한다.

이러한 아키텍처는 각 계층이 독립적으로 개발 및 유지 관리될 수 있도록 하여, 시스템의 모듈화와 확장성을 촉진한다.

3.10 System Evolution

이 섹션은 시스템 디자이너에게 유익한 시스템 변경 사항과 개선 사항을 다룬다. 또한, 하드웨어 기술의 발전과 요구 사항의 변화로 인해 시스템에 발생하는 변화를 설명한다. 향후 업데이트에는 성능 최적화, 보안 기능 강화, 사용자 피드백에 의해 주도되는 새로운 기능 추가가 포함될 수 있다. 또한, 시스템 아키텍처는 확장성과 신기술과의 통합을 수용할 수 있도록 설계되었다. 이를 통해 애플리케이션이 시간이 지나도 견고하고 효율적이며 관련성을 유지할 수 있다.

3.10.1 Limitation and Assumption

먼저, 본 애플리케이션의 목표는 성균관대학교 내의 시설물 문제를 신고하고 관리하는 것이다. 현재는 주로 성균관대학교 캠퍼스 내의 특정 시설물 정보와 관련하여 서비스하고 있다. 향후 기술 발전을 통해 더 많은 사용자가 애플리케이션을 사용하게 되거나 더 다양한 범위의 시설물 정보를 제공할 수 있게 된다면, 애플리케이션을 그에 맞추어 업데이트할 계획이다. 또한, 사용자의 필요에 따라 현재 서비스 중인 다양한 기능들을 포함하여 애플리케이션의 기능을 지속적으로 업데이트할 예정이다.

3.10.1.1 한계

1. 시설물의 한계: "Fix-SKKU"는 성균관대학교 캠퍼스 내에서 발생한 시설 문제를 보고하는 기능만을 지원한다. 또한 특정 시설물들의 신고기능만을 지원한다.
2. 기술적 제약: 앱은 최신 스마트폰 플랫폼에서 기능하도록 설계되었다. 구형 하드웨어나 운영 시스템을 사용하는 사용자는 기능이 제한되거나 호환성 문제를 겪을 수 있다.
3. 인터넷 의존성: "Fix-SKKU"는 문제를 보고하고 챗봇과 소통하기 위해 활성 인터넷 연결을 요구한다. 오프라인 기능은 지원되지 않아 연결성이 떨어지는 지역에서는 사용이 제한된다.

3.10.1.2 가정

1. 사용자 참여: 사용자들이 문제를 적극적으로 정확하게 보고할 동기가 있다고 가정한다. 시스템의 효과는 사용자 참여와 정확한 보고에 크게 의존한다.
2. 관리자 응답: 애플리케이션은 대학 관리자가 정기적으로 보고된 시설 문제를 검토하고 해결할 것으로 가정한다. 응답 시간과 해결 품질은 대학 관리 기준에 따라 사용자의 기대를 충족해야 한다.

3. 기술적 성장: 앱은 향후 기술 발전이 더 넓은 기능성을 허용할 수 있으며, 사용자 상호 작용과 문제 해결을 개선하기 위해 더 발전된 데이터 분석 및 AI 기능의 통합 가능성을 가정하고 설계되었다.
4. 안정적 운영 환경: "Fix-SKKU"의 지속적 운영 및 지원은 성균관대학교가 앱 유지 관리 및 업데이트에 필요한 인프라를 제공한다는 가정하에 이루어진다.

3.10.2 System Evolution

"Fix-SKKU" 어플리케이션은 성균관대학교 내의 시설물 관리 및 신고를 위해 개발되었다. 현재는 주로 교내 시설 문제를 신고하는 데 초점을 맞추고 있으나, 기술의 발전과 사용자 요구의 변화에 따라 애플리케이션의 기능과 범위는 지속적으로 진화할 필요가 있다.

1. 기능 향상: 초기 버전에서 제공하는 기본적인 신고 및 피드백 시스템 외에도, 사용자 경험을 향상시키기 위해 인터페이스 개선, 사용자 맞춤형 알림 시스템, AI 기반의 문제 진단 기능 등을 추가할 수 있다.
2. 데이터 분석 활용: 수집된 신고 데이터를 분석하여, 시설 관리에 있어서의 문제점을 파악하고 예방 조치를 취할 수 있는 인사이트를 제공한다. 이를 통해 더 효과적인 유지보수 계획을 수립할 수 있다.
3. 통합 관리 시스템: 다른 캠퍼스 관리 시스템과의 통합을 통해 보다 광범위한 관리가 가능하도록 확장할 수 있다. 예를 들어, 다른 대학과의 협력을 통해 비슷한 플랫폼을 공유하거나 통합하는 방안을 모색할 수 있다.
4. 사용자 피드백 및 개선: 사용자로부터 받은 피드백을 기반으로 지속적인 기능 개선 및 업데이트를 진행한다. 이는 사용자의 만족도를 높이고 신뢰성 있는 서비스를 제공하는 데 중요하다.
5. 스마트 캠퍼스 기술 도입: IoT 기술을 도입하여 자동화된 시설 관리 및 신고 시스템을 구축할 수 있다. 예를 들어, 센서 데이터를 활용하여 시설의 이상을 자동으로 감지하고 신고하는 시스템을 개발할 수 있다.

4. 추가 정보

4.1. 서식

이 요구사항 명세서는 IEEE Recommendation (IEEE Recommended Practice for Software Requirements Specifications, IEEE-Std-830). 서식을 따라 제작되었다.

4.2. 문서 시간표

표 37 문서 시간표

| 이름 | 날짜 | 변경 사항 | 버전 |
|-----------|------|-----------------------|-------|
| 전체 | 5/15 | 문서 작업 시작(Google Docs) | 0.0.0 |
| 이황근 | 5/17 | 1 | 0.1.1 |
| 함장건 | 5/17 | 3.1 | 0.2.1 |
| 이정빈 | 5/18 | 3.2.1 / 3.2.2 | 0.2.2 |
| 전용호 | 5/18 | 3.2.3 / 3.2.4 | 0.2.3 |
| 이황근 | 5/18 | 2 | 0.3.1 |
| 곽태윤 | 5/18 | 3.3 ~ 3.7 | 0.3.2 |
| 박소정 | 5/19 | 3.8 | 0.3.3 |
| 이황근 | 5/21 | 문서 통합 | 1.0.0 |
| 함장건 | 5/22 | 3.1 UI 변경 | 1.0.1 |
| 이정빈 | 5/23 | Use Case 추가 | 1.0.2 |
| 전용호 | 5/23 | E/R 수정 | 1.0.3 |
| 곽태윤 / 박소정 | 5/24 | 3.4~3.7 내용 추가 | 1.1.1 |
| 곽태윤 / 박소정 | 5/25 | Sequence Diagram 수정 | 1.1.2 |
| 이황근 | 5/26 | 문서 통합 | 2.0.0 |
| 이황근 | 5/26 | 표, 그림 캡션 추가 | 2.0.1 |
| 이황근 | 5/26 | 최종 서식 수정 | 2.0.2 |