



Smart Homepany: Customize your workplace

Software Requirement Specification

2022. 05. 01

Introduction to Software Engineering (SWE3002-42)

TEAM 1

| | |
|-------------|-----|
| Team Leader | 김종헌 |
| Team Member | 김영빈 |
| Team Member | 김민현 |
| Team Member | 오세훈 |
| Team Member | 이미소 |
| Team Member | 임세창 |
| Team Member | 홍권 |

목차

| | |
|--|-----------|
| 1. INTRODUCTION | 7 |
| 1.1 Purpose | 7 |
| 1.2 Scope | 7 |
| 1.3 Definitions, Acronyms and Abbreviation | 7 |
| 1.4 References | 8 |
| 1.5 Overview | 9 |
| 2. OVERALL DESCRIPTION..... | 10 |
| 2.1 Product Perspective..... | 10 |
| 2.1.1 System Interface..... | 10 |
| 2.1.2 User Interfaces | 10 |
| 2.1.3 Hardware Interfaces..... | 10 |
| 2.1.4 Software Interfaces | 11 |
| 2.1.5 Memory Constraints..... | 11 |
| 2.1.6 Operations..... | 11 |
| 2.1.6.1 IoT 기기 탐색 및 연동 | 11 |
| 2.1.6.2 프리셋 추가 및 수정..... | 11 |
| 2.1.6.3 프리셋 예약 및 타임라인 편집..... | 11 |
| 2.2 Product Functions..... | 11 |
| 2.2.1 지원하는 IoT 기기 탐색 및 연결 | 11 |
| 2.2.2 프리셋 관리 (생성, 수정, 삭제)..... | 12 |
| 2.2.3 프리셋 작동 타임라인 설정 | 12 |
| 2.2.4 세부 기능 (커스터마이징 대상) | 12 |
| 2.2.4.1 업무 알람 제어 (주거 환경 방해 금지 모드)..... | 12 |
| 2.2.4.2 주거 알람 제어 (업무 환경 방해 금지 모드)..... | 12 |
| 2.2.4.3 조명 조절 | 13 |
| 2.2.4.4 커튼 조절 | 13 |
| 2.2.4.5 온도 조절 | 13 |
| 2.3 User classes and Characteristics | 13 |
| 2.3.1 일반 사용자 | 13 |
| 2.4 Operating Environment | 14 |
| 2.4.1 Hardware..... | 14 |
| 2.4.1.1 IoT 기기..... | 14 |
| 2.4.1.2 모바일 디바이스 (소프트웨어 실행 환경)..... | 14 |
| 2.4.2 Software..... | 14 |
| 2.4.2.1 Android OS..... | 14 |
| 2.4.2.2 iOS..... | 15 |
| 2.5 Design and implementation Constraints | 15 |

| | | |
|------------|---|-----------|
| 2.6 | Assumptions and Dependencies | 16 |
| 3. | SPECIFIC REQUIREMENTS | 17 |
| 3.1 | External Interface Requirements | 17 |
| 3.1.1 | User Interfaces | 17 |
| 3.1.2 | Hardware Interfaces | 24 |
| 3.1.3 | Software Interfaces | 24 |
| 3.1.4 | Communication Interfaces | 25 |
| 3.2 | Functional Requirements | 25 |
| 3.2.1 | Use Case | 25 |
| 3.2.2 | Use Case Diagram | 35 |
| 3.2.3 | Data Dictionary | 36 |
| 3.2.4 | Data Flow Diagram | 38 |
| 3.3 | Nonfunctional Requirements | 39 |
| 3.3.1 | Product Requirements | 39 |
| 3.3.1.1 | Performance Requirements | 39 |
| 3.3.1.2 | Space Requirements | 39 |
| 3.3.1.3 | Security Requirements | 40 |
| 3.3.1.4 | Usability Requirements | 40 |
| 3.3.1.5 | Dependability Requirements | 40 |
| 3.3.2 | Organizational Requirements | 41 |
| 3.3.2.1 | Environmental Requirements | 41 |
| 3.3.2.2 | Operational Requirements | 41 |
| 3.3.2.3 | Development Requirements | 41 |
| 3.3.3 | External Requirements | 42 |
| 3.3.3.1 | Regulatory Requirements | 42 |
| 3.3.3.2 | Ethical Requirements | 42 |
| 3.3.3.3 | Accounting Requirements | 42 |
| 3.3.3.4 | Safety/Security Requirements | 42 |
| 3.4 | Organizing the Specific Requirements | 43 |
| 3.4.1 | Context Model | 43 |
| 3.4.2 | Process Model | 44 |
| 3.4.3 | Interaction Model | 47 |
| 3.4.3.1 | Use Case Diagram | 47 |
| 3.4.4 | Behavior Model | 48 |
| 3.4.4.1 | Data Flow Diagram | 48 |
| 3.4.4.2 | Sequence Diagram | 49 |
| 3.5 | System Architecture | 50 |
| 3.6 | System Evolution | 51 |
| 3.6.1 | Assumptions | 51 |
| 3.6.2 | Change of Requirements | 51 |
| 4. | SUPPORTING INFORMATION | 52 |
| 4.1 | Software Requirement Information | 52 |
| 4.2 | Document History | 52 |

그림 목차

| | |
|---|----|
| 그림 1 USER INTERFACE – 등록된 IOT 기기 목록 | 27 |
| 그림 2 USER INTERFACE - 프리셋 목록 및 등록 | 30 |
| 그림 3 USER INTERFACE – 프리셋 편집 | 31 |
| 그림 4 USER INTERFACE – TIMELINE, 프리셋 예약 | 33 |
| 그림 5 USE CASE DIAGRAM | 35 |
| 그림 6 DATA DICTIONARY | 37 |
| 그림 7 DATA FLOW DIAGRAM | 38 |
| 그림 8 CONTEXT MODEL | 43 |
| 그림 9 PROCESS MODEL (MAIN) | 44 |
| 그림 10 PROCESS MODEL (프리셋 예약) | 44 |
| 그림 11 PROCESS MODEL (IOT 기기 관리) | 45 |
| 그림 12 PROCESS MODEL (프리셋 관리) | 46 |
| 그림 13 USE CASE DIAGRAM (그림 5 동일) | 47 |
| 그림 14 DATA FLOW DIAGRAM (그림 7 동일) | 48 |
| 그림 15 SEQUENCE DIAGRAM (OVERALL) | 49 |
| 그림 16 SYSTEM ARCHITECTURE (MVC PATTERN) | 50 |

표 목차

| | |
|---|----|
| 표 1 DEFINITION | 7 |
| 표 2 ABBREVIATION AND ACRONYMS | 8 |
| 표 3 USER INTERFACE - 터치 스크린을 통한 메인 인터페이스 상 기본 상호작용..... | 17 |
| 표 4 USER INTERFACE - 타임라인 목록 | 18 |
| 표 5 USER INTERFACE - 타임라인 정보 | 19 |
| 표 6 USER INTERFACE - 타임라인 추가 | 20 |
| 표 7 USER INTERFACE - 프리셋 목록 | 21 |
| 표 8 USER INTERFACE - 프리셋 정보 | 22 |
| 표 9 USER INTERFACE - 프리셋 추가 | 23 |
| 표 10 USER INTERFACE - IOT 기기 탐색 | 24 |
| 표 11 HARDWARE INTERFACE - SYSTEM 에 적합한 기종 | 24 |
| 표 12 SOFTWARE INTERFACE – AWS RDS | 25 |
| 표 13 COMMUNICATION INTERFACE - CLIENT AND HOST | 25 |
| 표 14 USE CASE - 등록된 IOT 기기의 최신 펌웨어 조회 및 업데이트 | 26 |
| 표 15 USE CASE - SMART HOMEPANY 소프트웨어의 최신 버전 조회 및 업데이트 | 26 |
| 표 16 USE CASE - IOT 기기 탐색 | 27 |
| 표 17 USE CASE - IOT 기기 등록 | 28 |
| 표 18 USE CASE - IOT 기기 삭제 | 28 |
| 표 19 USE CASE – 업무 알람 제어 어플리케이션 지정 | 29 |
| 표 20 USE CASE - 프리셋 등록 | 30 |
| 표 21 USE CASE - 프리셋 편집 | 32 |
| 표 22 USE CASE - 프리셋 삭제 | 32 |
| 표 23 USE CASE - 프리셋 예약 | 33 |
| 표 24 USE CASE - 프리셋 예약 편집 | 34 |
| 표 25 USE CASE - 프리셋 예약 삭제 | 34 |
| 표 26 USE CASE - 프리셋 예약 비활성화 | 35 |
| 표 27 DATA DICTIONARY - USER | 36 |
| 표 28 DATA DICTIONARY - IOT DEVICE | 36 |
| 표 29 DATA DICTIONARY - CONNECTED DEVICE | 36 |
| 표 30 DATA DICTIONARY - PRESET | 36 |
| 표 31 DATA DICTIONARY - PRESET DEVICE | 36 |
| 표 32 DATA DICTIONARY – RESERVATION | 37 |

| | |
|---|----|
| ⌘ 33 DATA DICTIONARY - RESERVATION_REPETITION | 37 |
| ⌘ 34 DOCUMENT HISTORY | 53 |

1. Introduction

1.1 Purpose

본 문서는 재택 근무자를 위한 스마트 홈 소프트웨어 Smart Homepany 의 요구사항 명세서이다. Smart Homepany 는 재택 근무자에게 최적화된 업무 환경 및 주거 환경을 제공하기 위해 성균관대학교 2022 년 1 학기 소프트웨어공학개론 팀 1 에서 구상한 소프트웨어이다. 최근 국내 재택 근무자가 급격하게 증가하고 있음에도 불구하고, 기존의 스마트 홈 소프트웨어는 주로 주거 환경 개선에 집중하였다. 이에 Smart Homepany 는 사용자가 집에서 업무 시간에 업무에 집중할 수 있는 환경을 제공하고자 한다. 이뿐만 아니라, 사용자가 업무 시간 외에는 업무에서 벗어나 휴식을 취할 수 있는 주거 환경을 제공하고자 한다.

본 문서에는 Smart Homepany 에 대한 요구사항이 기술되어 있다. Smart Homepany 는 기존 스마트 홈 소프트웨어와 완전히 차별화된 기능을 제시한다. 따라서 본 문서에는 소프트웨어 전체에 대한 요구사항이 기술되어 있다. 주 독자는 성균관대학교 학생과 교수이다. Smart Homepany 는 본 문서에 의거하여 디자인 및 개발이 진행될 예정이다.

1.2 Scope

Smart Homepany 는 사용자가 집에서 업무 또는 휴식에 집중할 수 있도록 여러 IoT 기기들을 연동하고 제어할 수 있는 기능들을 제공한다. Smart Homepany 는 이러한 기능들을 하나의 프리셋으로 묶어 통합적으로 집 전체에 반영될 수 있도록 한다. 사용자는 각 기능을 커스터마이징하여 자신에게 최적화된 업무 환경 또는 주거 환경을 구성할 수 있다. 나아가, Smart Homepany 는 사용자가 일정을 등록하면 그에 맞춰 자동으로 프리셋을 전환하는 기능을 제공한다.

1.3 Definitions, Acronyms and Abbreviation

| 용어 | 정의 |
|------|--|
| 프리셋 | 연동된 IoT 기기들의 제어 값들의 설정을 묶어 놓은 하나의 모드를 지칭하는 용어이다. 시스템에서 기본적으로 제공되거나 사용자가 직접 추가 또는 변경할 수 있다. |
| 타임라인 | 특정 프리셋이 실행될 사용자 일정을 지칭하는 용어이다. 사용자가 직접 추가 또는 변경할 수 있다. |

표 1 Definition

| 약어 | 해설 |
|-------|-----------------------------------|
| API | Application Programming Interface |
| DB | Database |
| IoT | Internet of Things |
| OS | Operating System |
| RAM | Random Access Memory |
| UI | User Interface |
| Wi-Fi | Wireless Fidelity |

표 2 Abbreviation and Acronyms

1.4 References

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

"Project Highlight_Requirements 2.0.4" SKKU 2021 Introduction to Software Engineering Team 1, Last Modified on April 25, 2021.

https://github.com/skkuse/2021spring_41class_team1/blob/main/doc/Project%20Highlight_Requirement%202.0.4.pdf

"Requirements Speicification_Team 12" SKKU 2021 Introduction to Software Engineering Team 12, Last Modified on April 24, 2021.

https://github.com/skkuse/2021spring_41class_team12/blob/main/Requirements%20Specification_Team%2012.pdf

"SE-2021-2-Team4-SRS" SKKU 2021 Introduction to Software Engineering Team 4, Last Modified on 21 Nov, 2021.

https://github.com/skkuse/2021fall_41class_team4/blob/main/docs/SE-2021-2-Team4-SRS.pdf

1.5 Overview

본 명세서는 Smart Homepany 의 요구 사항에 대해 총 3 장에 걸쳐 기술되어 있다.

두 번째 장에서는 Smart Homepany 의 전체적인 청사진을 제시한다. 첫째, Smart Homepany 의 거시적 구조를 여러 소프트웨어공학적 관점으로 분석한다. 둘째, Smart Homepany 가 제공할 기능들에 대해 명시한다. 셋째, 사용자에게 대해 분석한다. 넷째, 소프트웨어 실행 과정에서 있을 제약 사항에 대해 분석한다. 다섯 번째, Smart Homepany 에 대한 기타 가정이나 다른 시스템과의 의존성 관계에 대해 명시한다.

세 번째 장에서는 Smart Homepany 의 요구 사항에 대해 보다 상세히 기술한다. Smart Homepany 의 인터페이스, 기능, 데이터베이스 등에 대해 기술한다. 또한 외부 하드웨어 및 소프트웨어 인터페이스 관련 요구 사항, 기능적 요구 사항, 비기능적 요구 사항, 소프트웨어 모델링, 요구 사항 간의 구조, 시스템 아키텍처 등에 대해 기술한다.

2. Overall Description

2.1 Product Perspective

Smart Homepany 는 재택 근무자에게 일할 때에는 업무에 집중하고, 일을 하지 않을 때에는 온전히 휴식을 취할 수 있는 주거 환경을 제공하는 스마트 홈 소프트웨어이다. Smart Homepany 는 주변 IoT 기기들을 탐색하고 연결하고, 설정된 프리셋에 따라 연동된 기기 및 소프트웨어를 제어한다. 사용자는 커스터마이징을 통해 자신에게 최적화된 업무 환경 또는 주거 환경을 구성할 수 있다. 그리고 사용자가 타임라인을 추가하여 프리셋을 예약하면 예약 시간에 맞춰 자동으로 프리셋을 전환하여, 재택 근무자에게 최적화된, 효율적이면서도 건강한 주거 환경을 제공한다.

2.1.1 System Interface

사용자 정보, 연동된 IoT 기기 정보, 그리고 프리셋 및 타임라인에 대한 정보는 서버에 저장된다. 저장된 정보의 검색 및 관리는 아마존 관계형 데이터베이스 서비스(AWS RDS)를 통해 진행된다.

2.1.2 User Interfaces

사용자는 모바일 디바이스의 화면을 통해 어플리케이션 UI 를 확인할 수 있다. 본 소프트웨어의 UI 는 크게 세 가지로 나뉜다. 첫 번째는 주변 IoT 기기를 탐색하고 연결할 수 있는 페이지이다. 두 번째는 여러 가지 프리셋을 저장할 수 있는 Preset 이다. 사용자는 Preset 에서 새로운 프리셋을 추가하거나, 등록되어있는 프리셋의 기능 설정을 변경하고 저장할 수 있다. 마지막으로는 일정에 따라 프리셋을 배치할 수 있는 Timeline 이다. 사용자는 Timeline 에서 일정을 등록하고 해당 일정에 맞춰 실행할 특정 프리셋을 설정할 수 있다.

2.1.3 Hardware Interfaces

본 소프트웨어는 Android 및 iOS 모바일 기기 어플리케이션으로 구현될 예정이다. 본 소프트웨어를 실행하기 위해서는 최소 1GB 의 RAM 용량과 1.2GHz 이상의 CPU 클럭 속도가 요구된다. 원활한 실행을 위해서는 2GB 이상의 RAM 용량과 1.6 GHz 이상의 GPU 클럭 속도가 권장된다.

2.1.4 Software Interfaces

본 소프트웨어는 Android 10.0 (API 29) 또는 iOS 13.0 가 요구된다. 원활한 실행을 위해서는 Android 12.0 (API 31) 또는 iOS 15.0 이 권장된다.

2.1.5 Memory Constraints

본 소프트웨어를 설치하고 실행하기 위해 해당 모바일 기기에 최소 1GB 의 RAM 용량과 600MB 의 저장소 용량이 요구된다.

2.1.6 Operations

2.1.6.1 IoT 기기 탐색 및 연동

사용자는 Wi-Fi 및 Bluetooth 를 통해 모바일 디바이스와 연결할 수 있는 IoT 기기 목록을 확인하고, 목록을 선택하여 기기와 디바이스를 연동할 수 있다.

2.1.6.2 프리셋 추가 및 수정

사용자는 새로운 프리셋을 추가하여 연동된 IoT 기기 및 소프트웨어에 대해 제어 값을 설정할 수 있다. 또한 기존 프리셋의 설정을 변경하거나 프리셋을 삭제할 수 있다.

2.1.6.3 프리셋 예약 및 타임라인 편집

사용자는 지정된 시각에 특정 프리셋이 실행되도록 설정할 수 있다. 타임라인을 추가하여 실행하고자 하는 프리셋과 실행할 요일 및 시간 예약할 수 있다. 또한 타임라인의 설정을 변경, 삭제 또는 비활성화할 수 있다.

2.2 Product Functions

2.2.1 지원하는 IoT 기기 탐색 및 연결

Smart Homepany 소프트웨어는 사용자의 스마트폰과 같은 Wi-Fi 네트워크를 사용하는 IoT 기기 및 주변 Bluetooth 로 연결 주변 IoT 기기들을 자동으로 탐색 및 연결하는 기능을 지원한다. 사용자가 활용하는 IoT 기기는 Smart Homepany 에서 지원하는 규격 및 타입에 맞아야 한다. 연동이 완료되면 사용자는 소프트웨어 내에서 IoT 기기에 대한 설정 값을 지정 및 수정할 수 있다. 전파 간섭이 많은 곳 등 IoT 기기와 모바일 디바이스 간 통신이 원활하지 않을 수 있는 환경에서는 본 기능이 정상적으로 동작하지 않을 수 있다.

2.2.2 프리셋 관리 (생성, 수정, 삭제)

Smart Homepany 는 사용자가 소유하고 있는 IoT 기기들에 대하여 스스로 자유롭게 설정 값들을 조정하고 이를 저장할 수 있는 프리셋 기능을 지원한다. 사용자가 각 값들을 직접 설정하기에 앞서, Smart Homepany 는 대부분의 사용자가 가장 범용적으로 사용할 수 있는 두 기본 프리셋(주거 환경 모드 프리셋, 업무 환경 모드 프리셋)을 제공하며, 사용자는 두 기본 프리셋을 바탕으로 자신만의 커스텀 프리셋을 생성, 수정 및 삭제할 수 있다. 프리셋은 본 소프트웨어에서 지원하는 예약 모드 설정 시에 유용하게 사용될 수 있다.

2.2.3 프리셋 작동 타임라인 설정

Smart Homepany 는 사용자가 소유하고 있는 모든 프리셋에 대하여 각 프리셋이 언제 작동되고 언제 작동을 종료하는지 타임라인을 설정할 수 있다. 만약 사용자가 오전 9시부터 오후 18시까지 자택에서 근무를 진행 후 그 외의 시간에는 휴식을 취한다고 가정하면, 사용자는 오전 9시에 자동으로 주거 환경 모드 프리셋에서 업무 환경 모드 프리셋으로 전환, 오후 18시에는 자동으로 업무 환경 모드 프리셋에서 주거 환경 모드 프리셋으로 전환하는 타임라인을 설정할 수 있다. 근무 시간에 변동이 있는 등 모종의 이유로 특정 프리셋 시간의 연장 혹은 단축을 원한다면 사용자는 언제든지 타임라인을 수정할 수 있다.

2.2.4 세부 기능 (커스터마이징 대상)

여기서는 Smart Homepany 에서 기본적으로 지원하는 대표적인 5 가지 세부 기능에 대하여 설명한다. 본 항목에서 기술되는 내용은 추후 소프트웨어가 진화함에 따라 수정 및 추가될 수 있다. 현재 버전을 기준으로 Smart Homepany 는 총 5 가지의 세부 기능을 지원한다.

2.2.4.1 업무 알람 제어 (주거 환경 방해 금지 모드)

사용자가 업무 외 시간으로 설정한 시간대에 PC 및 모바일 디바이스(스마트폰, 태블릿 PC 등)에서 업무 관련 알람이 울리지 않도록 제어한다. 알람 제어 대상은 사내 메신저, 업무용 이메일 등 업무와 관련된 모든 어플리케이션을 포함한다. 본 기능의 설계 목적은 사용자가 업무 외 시간에 업무와 관련된 스트레스 및 부담을 느끼지 않도록 하는 것이므로, 본 기능은 사용자가 업무 관련 어플리케이션을 모두 제어 대상으로 설정할 것을 권장한다. 본 기능은 기본 프리셋 기준 주거 환경 모드에서 동작할 수 있도록 제공되지만, 사용자의 기호에 따라 여타 프리셋에서도 동작할 수 있도록 설정할 수 있다.

2.2.4.2 주거 알람 제어 (업무 환경 방해 금지 모드)

사용자가 업무 시간으로 설정한 시간대에는 자택 내 인터폰 알람이 울리지 않도록 제어하는 기능이다. 알람 제어 대상은 아래 공동 현관문과 집 앞 현관문, 아파트 내 안내 방송 등 모든 외부의

소음을 포함한다. 본 기능은 사용자가 업무 시간에 외부의 자극으로 인해 흐름이 끊기는 일이 없도록 하는 것을 목표로 한다. 본 기능은 기본 프리셋 기준 업무 환경 모드에서 동작할 수 있도록 제공되지만, 사용자의 기호에 따라 여타 프리셋에서도 동작할 수 있도록 설정할 수 있다.

2.2.4.3 조명 조절

사용자는 자택 내에 설치되어 있는 스마트 조명의 색상, 온도, 채도, 밝기 등을 조절할 수 있다. Smart Homepany 가 지원하는 규격의 조명이기만 하면, 여러 개의 조명이 있을 경우에도 각 조명에 대한 설정 값을 서로 다르게 지정할 수 있다. 사용자는 업무 및 휴식 환경에서 각자가 필요한 분위기에 따라 조명의 설정 값을 자유롭게 조절할 수 있다.

2.2.4.4 커튼 조절

사용자는 자택 내에 설치되어 있는 커튼이 열린 정도를 조절할 수 있다. Smart Homepany 가 지원하는 규격의 커튼이기만 하면, 여러 개의 커튼이 있을 경우에도 각 커튼에 대한 설정 값을 서로 다르게 지정할 수 있다. 사용자는 각 시간대마다 자신이 원하는 채광에 맞게 커튼이 열린 정도를 지정할 수 있다.

2.2.4.5 온도 조절

온도 센서가 자택 내의 온도를 파악하여 만약 자택의 온도가 사용자가 사전에 설정한 이상 범위 내에 위치하지 않으면 보일러, 에어컨 등 여러 온도 조절 시스템을 작동시켜 최적의 실내 온도를 유지한다. 사용자는 시간대별로 원하는 실내 온도를 사전에 지정할 수 있다.

2.3 User classes and Characteristics

Smart Homepany 는 사용자 개개인의 특성이 각자가 소유하고 있는 프리셋에 모두 반영될 수 있도록 돕기 위하여 사용자의 자유도를 최대화하고자 한다. 따라서 기본적으로 Smart Homepany 의 사용자 클래스는 일반 사용자 단 하나만 존재하며 일반 사용자에게 모든 권한 및 기능을 위임한다. 소프트웨어가 진화함에 따라 범용적으로 많은 사용자들이 선호할 수 있는 프리셋을 탐색할 것이고 이에 따라 기본 프리셋의 설정 값 및 종류가 지속적으로 변화할 수 있으나, 관리자 사용자 클래스를 별도로 두지 않고 소프트웨어 개발 단계에서 이를 제어하고자 한다.

2.3.1 일반 사용자

Smart Homepany 의 사용자는 기본적으로 모두 일반 사용자에게 해당한다. 본 소프트웨어는 기본 프리셋(주거 환경 모드 프리셋, 업무 환경 모드 프리셋)을 모든 일반 사용자에게 제공하며 이 기본 프리셋은 수정 및 삭제할 수 없다. 일반 사용자는 기본 프리셋에 대한 수정, 삭제 권한을 제외하면 Smart Homepany 내에 존재하는 모든 기능을 자유롭게 제어할 수 있는 권한이 있다.

사용자는 기본 프리셋을 제외한 모든 프리셋은 수정, 삭제할 수 있으며 새로운 프리셋을 추가할 수 있다. 사용자는 프리셋 작동 타임테이블을 임의로 수정하거나, 새롭게 생성하거나 수정한 프리셋을 원하는 시간대에 작동할 수 있도록 추가할 수 있다. 사용자는 각 프리셋을 생성 및 수정할 때 각 기능에 대하여 임의로 설정값을 조정할 수 있다.

사용자는 업무 알람 제어 기능을 활용하기 위해 업무와 관련된 모든 어플리케이션을 알람 제어 대상으로 설정하여야 하며, 주거 알람 제어 기능을 활용하기 위해 주거 관련 알람 중 어느 범위까지 알람을 제어할지 그 대상을 설정하여야 한다. 그 외 조명, 커튼, 온도에 대한 설정 값은 여타 IoT 기기들처럼 사용자가 자신에게 맞는 설정 값을 커스터마이징할 수 있다.

2.4 Operating Environment

2.4.1 Hardware

2.4.1.1 IoT 기기

- 모든 IoT 기기는 Smart Homepany 에서 지원하는 규격의 모델이어야 한다.
- 본 소프트웨어는 2.4GHz 대역의 Wi-Fi 네트워크에서만 동작한다. 2.4GHz 와 5GHz 를 모두 지원하는 듀얼 기능이 있는 네트워크를 사용하는 경우, 사용자는 2.4GHz 를 선택하여야 한다.
- 본 소프트웨어를 실행하기 위해 안정적인 Wi-Fi 네트워크가 요구된다. 불안정한 네트워크 환경에서는 소프트웨어가 의도한 대로 동작하지 않을 수 있다.

2.4.1.2 모바일 디바이스 (소프트웨어 실행 환경)

- 본 소프트웨어는 Android 및 iOS 모바일 디바이스를 타겟으로 하여 구현된다.
- 본 소프트웨어를 실행하기 위해 1GB 이상의 RAM 용량과 1.2GHz 이상의 CPU 클럭 속도가 최소 사양으로 요구된다.
- 본 소프트웨어를 실행하기 위해 2GB 이상의 RAM 용량과 1.6GHz 이상의 CPU 클럭 속도가 권장 사양으로 요구된다.

2.4.2 Software

2.4.2.1 Android OS

- 본 소프트웨어는 Android 10 Queen Cake (API 29) 이상의 운영체제가 최소 사양으로 요구된다.
- 본 소프트웨어는 Android 12 Snow Cone (API 31) 이상의 운영체제가 권장 사양으로 요구된다.

2.4.2.2 iOS

- 본 소프트웨어는 iOS 13.0 이상의 운영체제가 최소 사양으로 요구된다.
- 본 소프트웨어는 iOS 15.0 이상의 운영체제가 권장 사양으로 요구된다.

2.5 Design and implementation Constraints

Smart Homepany 소프트웨어는 하기 요구 사항을 충족할 수 있도록 구현될 예정이다. 본 내용은 추후 사용자의 세부 요구 사항 및 여러 개발 환경 제약에 따라 더 많은 내용을 포함하게 될 수 있으나 아래 요구사항은 반드시 준수하여야 한다.

- 본 소프트웨어는 실제 사용자의 편의성과 요구사항을 최우선으로 고려하여 개발되어야 한다.
- 본 소프트웨어는 개발 단계에서 일부 컴포넌트에 대하여 요구 기능을 충족하고 있는 오픈소스 소프트웨어가 이미 존재할 경우 해당 소프트웨어를 적극적으로 이용하여야 한다.
- 본 소프트웨어의 개발 단계에서 불필요한 메모리 및 시간 소모를 최소화하기 위해 지속적으로 소스 코드를 리팩토링 및 업데이트하여야 한다.
- 본 소프트웨어의 개발 단계에서는 추후 소프트웨어 진화 및 유지 보수에 대하여 항상 고려하여야 하며 이를 위해 각 기능에 대한 주석을 반드시 포함하여야 한다.
- 본 소프트웨어는 사용자에게 친숙한 유저 인터페이스(UI)를 제공함으로써 User document 를 자세히 읽지 않아도 쉽게 기능을 파악할 수 있도록 설계되어야 한다.
- 본 소프트웨어는 사용자에게 Bluetooth 및 위치 정보, 푸시 및 팝업 알람에 대한 권한을 요구할 수 있으며 사용자는 부여한 권한에 대한 변경이 언제든지 가능하여야 한다.
- 본 소프트웨어는 용량이 600MB 를 넘지 않도록 개발되어야 한다.
- 본 소프트웨어는 Android 10 Queen Cake (API 29) 및 iOS 13.0 이상의 운영체제에서 어떠한 오류도 없이 정상적으로 동작하여야 한다.
- IoT 기기는 1GB 의 RAM 용량과 1.2GHz 의 CPU 클럭 속도를 기준으로 IoT 기기는 3 초 이내에 사용자의 명령에 대응되는 액션을 수행하여야 한다.
- 본 소프트웨어는 2.4GHz 의 Wi-Fi 대역에서 IoT 기기는 3 초 이내에 사용자의 명령에 대응되는 액션을 수행하여야 한다.
- 본 소프트웨어에서 제공하는 기본 프리셋은 언제든지 쉽게 업데이트될 수 있도록 설계 및 개발되어야 한다.
- 본 소프트웨어는 사용자가 별도로 데이터 수집에 대하여 동의하지 않는 한, 사용자의 프리셋 및 모드 설정, 사용자의 타임테이블에 관한 데이터를 절대 수집하지 않는다.

2.6 Assumptions and Dependencies

본 소프트웨어는 1GB 의 RAM 용량 및 1.2GHz 의 CPU 클럭 속도를 지원하는 하드웨어와 Android 10 Queen Cake (API 29) 혹은 iOS 13.0 운영체제를 최소 사양으로 요구하며, 2GB 이상의 RAM 용량 및 1.6GHz 이상의 CPU 클럭 속도를 지원하는 하드웨어와 Android 12 Snow Cone (API 31) 이상 혹은 iOS 15.0 이상의 운영체제를 권장 사양으로 요구한다.

사용자는 업무 알람 제어 기능을 온전히 활용하기 위해 반드시 본인의 업무 관련 어플리케이션을 알람 제어 대상으로 지정하여야 하며, 주거 알람 제어 기능을 온전히 활용하기 위해 자택 내 인터폰 등을 알람 제어 대상으로 지정하여야 한다.

본 소프트웨어는 사용자의 모바일 디바이스에 Bluetooth 및 위치 정보, 푸시 및 팝업 알람에 대한 권한을 요구할 수 있으며 사용자는 이에 대해 선택적으로 권한을 허용할 수 있으나, 기본적으로 모든 요구 권한을 허용하는 것을 권장한다.

만약 상기 사항들이 지켜지지 않을 경우 Smart Homepany 소프트웨어는 개발자 및 사용자가 의도한 대로 동작하지 않을 수 있으며, 서비스 이용에 차질이 있을 수 있다.

3. Specific Requirements

3.1 External Interface Requirements

3.1.1 User Interfaces

| Name | 터치 스크린을 통한 메인 인터페이스 상 기본 상호작용 |
|--|--|
| Purpose/ Description | 사용자는 디바이스의 터치 스크린을 통해 명령을 전송한다. |
| Input source/ Output destination | User/Android OS 혹은 iOS 기반 모바일 디바이스 (Android 10.0 이상, iOS 13.0 이상) |
| Range/Accuracy/ Margin of error | 화면 상 UI 개수에 따른 범위/사용자의 터치 정확도에 따른 정확도/디바이스의 터치 감도 오차 한계 |
| Unit | 클릭 |
| Time/Velocity | 사용자의 비동기 입력/즉각적 실행 |
| Relationship with other input/outputs | 입력을 수신 시에, 디바이스 내부에서 자체적으로 입력 값을 처리하거나, 서버에 입력 데이터를 전송함을 통해 출력 데이터를 요청한다. |
| Format and configuration | <ol style="list-style-type: none"> 1. 화면은 Text, Image, Button, Dropdown, TimePicker 등의 위젯으로 구성되어 있다. 2. 화면의 위젯들은 정보를 표시하거나, 사용자로부터 입력 데이터를 받을 수 있다. 3. 사용자는 화면에서 원하는 버튼을 클릭함으로써 시스템과 상호작용 할 수 있다. |
| Data type | N/A |
| Instruction type | 버튼 코드 값, 디바이스에서 터치한 위치 값 |
| Exit message | N/A |

표 3 User Interface - 터치 스크린을 통한 메인 인터페이스 상 기본 상호작용

| Name | 타임라인 - 목록 |
|--|--|
| Purpose/ Description | 어플리케이션 실행 후 처음으로 나타나는 화면으로, 사용자가 서버에 저장한 프리셋 전환 예약 일정 목록을 띄운다. |
| Input source/ Output destination | Server/Client |
| Range/Accuracy/ Margin of error | N/A |
| Unit | 스크린 |
| Time/Velocity | N/A |
| Relationship with other input/outputs | N/A |
| Format and configuration | <ol style="list-style-type: none"> 레이아웃은 프리셋 전환이 예정된 타임라인들을 각각 전환 예정 시간과 함께 보여준다. 사용자는 일정을 클릭함으로써 타임라인의 상세 정보 페이지로 이동한다. 사용자가 추가 버튼을 클릭하면, 일정 타임라인 페이지로 이동한다. |
| Data type | 텍스트, 이미지, 위젯, 버튼 |
| Instruction type | N/A |
| Exit message | N/A |

표 4 User Interface - 타임라인 목록

| Name | 타임라인 정보 |
|-------------------------|--|
| Purpose/ Description | 타임라인 목록 페이지에서, 사용자가 특정 타임라인 위젯을 클릭하면, 해당 타임라인에 대한 상세 정보 페이지로 넘어간다. 상세 정보 페이지에서는 타임라인을 수정 및 삭제 가능하며, 전환하는 프리셋에 포함된 IoT 기기들의 기능 조절 정도를 볼 수 있다. |

| | |
|--|---|
| Input source/ Output destination | Client/Server |
| Range/Accuracy/ Margin of error | N/A |
| Unit | 페이지 |
| Time/Velocity | N/A |
| Relationship with other input/outputs | N/A |
| Format and configuration | <ol style="list-style-type: none"> 1. 사용자는 타임라인 상세 정보 페이지에서 해당 일정에 대한 예약 시간과 전환하는 프리셋을 변경 가능하다. 2. 사용자가 Edit 버튼을 클릭하면, 변경된 정보들이 반영되고, 시스템은 토스트 메시지를 띄운다. 3. 사용자가 좌측 화살표 버튼을 클릭했을 때 변경사항이 없다면, 타임라인 목록 페이지로 이동한다. 4. 사용자가 좌측 화살표 버튼을 클릭했을 때 변경사항이 있다면, 토스트 메시지를 띄운다. 5. 사용자는 Delete 버튼을 통해 해당 타임라인을 삭제할 수 있다. |
| Data type | Query |
| Instruction type | 버튼 코드 값에 따른 안내 문구 매핑 |
| Exit message | "변경 사항을 폐기 하시겠습니까?" / "변경 사항이 적용되었습니다" / "일정을 삭제 하시겠습니까?" |

표 5 User Interface - 타임라인 정보

| Name | 타임라인 추가 |
|-------------------------------------|--|
| Purpose/ Description | <p>타임라인 목록 페이지에서 사용자는 타임라인 추가 버튼을 통해 타임라인 추가 페이지로 넘어온다.</p> <p>타임라인 추가 페이지에서는 타임라인을 새로 추가할 수 있다.</p> |
| Input source/ Output destination | Client/Server |

| | |
|--|--|
| Range/Accuracy/ Margin of error | N/A |
| Unit | 페이지 |
| Time/Velocity | N/A |
| Relationship with other input/outputs | N/A |
| Format and configuration | <ol style="list-style-type: none"> 1. 사용자는 일정 추가 페이지에서 새로운 일정에 대한 예약 시간과 전환하는 프리셋을 설정 가능하다. 2. 사용자가 등록 버튼을 클릭하면, 설정한 정보들이 반영된 일정이 등록되고 시스템은 토스트 메시지를 띄운다. 3. 사용자가 좌측 화살표 버튼을 클릭하면, 설정한 정보가 없다면 일정 목록 페이지로 이동한다. 4. 시스템은 등록된 사항이 있다면, '등록' 버튼 대신 좌측 화살표 버튼을 클릭 시에 토스트 메시지를 띄운다. |
| Data type | Query |
| Instruction type | 버튼 코드 값에 따른 안내 문구 매핑 |
| Exit message | "등록 사항을 폐기 하시겠습니까?" / "새로운 일정이 추가되었습니다" |

표 6 User Interface - 타임라인 추가

| Name | 프리셋 목록 |
|-------------------------------------|---|
| Purpose/ Description | <p>메인 인터페이스에서 하단 프리셋 버튼을 통해 프리셋 목록 페이지로 넘어올 수 있다.</p> <p>사용자가 등록한 프리셋 및 기본적으로 세팅 되어 있는 프리셋의 목록을 보여준다.</p> |
| Input source/ Output destination | Server/Client |
| Range/Accuracy/ Margin of error | N/A |
| Unit | 스크린 |
| Time/Velocity | N/A |

| | |
|---------------------------------------|--|
| Relationship with other input/outputs | N/A |
| Format and configuration | <ol style="list-style-type: none"> 레이아웃은 등록된 프리셋 목록을 보여준다. 사용자는 프리셋을 클릭함으로써 프리셋의 상세 정보 페이지로 이동한다. 사용자가 추가 버튼을 클릭하면, 프리셋 추가 페이지로 이동한다. |
| Data type | 텍스트, 이미지, 위젯, 버튼 |
| Instruction type | N/A |
| Exit message | N/A |

표 7 User Interface - 프리셋 목록

| Name | 프리셋 정보 |
|---------------------------------------|--|
| Purpose/ Description | <p>일정 목록에서, 사용자가 특정 일정 위젯을 클릭하면, 해당 일정에 대한 상세 정보 페이지로 넘어간다.</p> <p>상세 정보 페이지에서는 일정을 수정 및 삭제 가능하며, 전환하는 프리셋에 포함된 IoT 기기들의 기능 조절 정도를 볼 수 있다.</p> |
| Input source/ Output destination | Client / Server |
| Range/Accuracy/ Margin of error | N/A |
| Unit | 페이지 |
| Time/Velocity | N/A |
| Relationship with other input/outputs | N/A |

| | |
|--------------------------|---|
| Format and configuration | <ol style="list-style-type: none"> 1. 사용자는 프리셋 상세 정보 페이지에서 해당 프리셋에 IoT 기기를 추가 혹은 삭제하거나, 등록된 IoT 의 기능의 정도를 조절할 수 있다. 2. 사용자가 추가 버튼을 누르면, 등록할 수 있는 IoT 기기 리스트에서 원하는 IoT 기기를 추가할 수 있는 페이지로 이동한다. 3. 사용자가 좌측 화살표 버튼을 클릭하면, 변경사항이 없다면 프리셋 목록 페이지로 이동한다. 4. 시스템은 변경사항이 있다면, Edit 버튼 대신 좌측 화살표 버튼을 클릭 시에 토스트 메시지를 띄운다. 5. 사용자는 삭제 버튼을 통해 해당 프리셋을 삭제할 수 있다. 6. 사용자는 존재하는 IoT 기기의 기능 정도를 위젯을 통해 변경 가능하다. |
| Data type | Query |
| Instruction type | 버튼 코드 값에 따른 안내 문구 매핑 |
| Exit message | "변경 사항을 폐기 하시겠습니까?" / "프리셋 변경 사항이 적용되었습니다" / "프리셋을 삭제 하시겠습니까?" |

표 8 User Interface - 프리셋 정보

| Name | 프리셋 추가 |
|--|---|
| Purpose/Description | <p>프리셋 목록 페이지에서 사용자는 프리셋 추가 버튼을 통해 프리셋 페이지 추가 페이지로 넘어온다.</p> <p>프리셋 추가 페이지에서는 프리셋을 새로 추가할 수 있다.</p> |
| Input source/ Output destination | Client / Server |
| Range/ Accuracy/ Margin of error | N/A |
| Unit | 페이지 |

| | |
|---------------------------------------|---|
| Time/ Velocity | N/A |
| Relationship with other input/outputs | N/A |
| Format and configuration | <ol style="list-style-type: none"> 1. 사용자는 프리셋 추가 페이지에서 새로운 프리셋을 구성하는 IoT 기기를 등록하여 프리셋을 등록할 수 있다.. 2. 사용자가 등록 버튼을 클릭하면, 설정한 IoT 와 조절 정도가 반영된 프리셋이 등록되고 시스템은 토스트 메시지를 띄운다. 3. 사용자가 좌측 화살표 버튼을 클릭하면, 설정한 정보가 없다면 프리셋 목록 페이지로 이동한다. 4. 시스템은 등록된 사항이 있다면, '등록' 버튼 대신 좌측 화살표 버튼을 클릭 시에 토스트 메시지를 띄운다. |
| Data type | Query |
| Instruction type | 버튼 코드 값에 따른 안내 문구 매핑 |
| Exit message | "등록 사항을 폐기 하시겠습니까?" / "새로운 프리셋이 추가되었습니다" |

표 9 User Interface - 프리셋 추가

| Name | IoT 기기 탐색 |
|---------------------------------------|---|
| Purpose/ Description | 사용자는 동일 네트워크에 존재하는 IoT 기기를 탐색할 수 있으며, 등록되지 않은 기기를 추가할 수 있다. |
| Input source/ Output destination | User / IoT device, IoT device / User, Client / Server |
| Range/ Accuracy/ Margin of error | IoT 기기 및 서버에 의존 |
| Unit | 스크린 |
| Time/ Velocity | N/A |
| Relationship with other input/outputs | 어느 IoT 기기를 등록할 것인지 사용자의 선택에 따라 서버에 등록할 기기가 결정됨. |

| | |
|--------------------------|---|
| Format and configuration | <ol style="list-style-type: none"> 1. 동일 네트워크에 등록 가능한 활성화된 IoT 기기 리스트가 나타난다. 2. 사용자가 리스트에서 등록하고자 하는 IoT 기기를 클릭한다. 3. 응답이 오고, 서버에 등록될 때까지 로딩 표시가 나타난다. 4. 완료 혹은 취소 시에 IoT 기기 목록 화면으로 돌아간다. |
| Data type | 버튼, 텍스트, 이미지 |
| Instruction type | 버튼 코드 값에 따른 안내 문구 매핑 |
| Exit message | "기기 탐색을 종료 하시겠습니까?", "예/아니오" |

표 10 User Interface - IoT 기기 탐색

3.1.2 Hardware Interfaces

| Name | System 에 적합한 기종 |
|-------------------------|---|
| Purpose/ Description | <p>해당 조건을 만족하는 Hardware 를 이용하여, 사용자들이 시스템의 서비스를 제공받을 수 있도록 한다.</p> <p>CPU: 코어 클럭 1.2GHz 이상</p> <p>Memory: 1GB RAM 이상</p> <p>OS: Android 10.0 이상, iOS 13.0 이상</p> <p>Wi-Fi: 2.4GHz</p> |

표 11 Hardware Interface - System 에 적합한 기종

3.1.3 Software Interfaces

| Name | AWS RDS |
|--|--|
| Purpose/Description | Preset 및 IoT 기기 정보를 관리하기 위한 Query input/output |
| Input source/ Output destination | Host server/ User, User/User, User/Host server |
| Unit | Query |
| Time/Velocity | 즉시 반응 |
| Relationship with other input/outputs | RDS Host Server 의 모든 input/ouput 과 연관되어 있다. |

| | |
|-----------|-----------------|
| Data type | Query statement |
|-----------|-----------------|

표 12 Software Interface – AWS RDS

3.1.4 Communication Interfaces

| Name | Client and Host |
|-------------------------------------|---|
| Purpose/Description | Client 는 Host 에게 서버 연결을 요청하고, 그 결과로 사용자에게 등록되어 있는 IoT 기기 리스트와 프리셋 리스트를 Host 로부터 제공받는다. Client 가 프리셋을 등록/변경/삭제하거나 IoT 기기를 등록/삭제 시에, Host 에게 서버 연결 및 변경 사항을 요청하며, Host 는 해당 요청에 대한 결과를 반환한다. |
| Input source/ Output destination | User/Host server, Host server/User |
| Unit | packet |
| Time/ Velocity | 서버의 성능에 의존한다. |

표 13 Communication Interface - Client and Host

3.2 Functional Requirements

3.2.1 Use Case

| Use case name | 등록된 IoT 기기의 최신 펌웨어 조회 및 업데이트 |
|----------------|---|
| Actor | 사용자, 모바일 디바이스, IoT 기기, 서버 |
| Description | 사용자는 등록된 IoT 기기의 모델명을 통해 서버에서 최신 펌웨어 버전을 조회할 수 있고 해당 IoT 기기의 현재 펌웨어 버전이 더 낮을 경우 최신 펌웨어를 서버로부터 다운로드 받아 IoT 기기에 적용할 수 있도록 한다. |
| Pre-Condition | 선택한 IoT 기기가 동일한 Wi-Fi 네트워크 상에 연결된 상태여야 한다. 모바일 디바이스가 서버와 통신 가능한 상태여야 한다. |
| Post-Condition | 모바일 디바이스가 서버와 통신 가능한 상태이며 선택한 IoT 기기가 동일 Wi-Fi 네트워크 상에 연결된 상태라면, IoT 기기는 자신의 펌웨어 버전을 사용자의 모바일 디바이스에 전달해야만 한다. |

| | |
|---------------|--|
| Normal Course | <ol style="list-style-type: none"> 1. IoT 기기에 현재 설치되어 있는 펌웨어의 버전을 조회한다. 2. 1 에서 조회한 펌웨어 버전과 서버 상의 최신 버전을 비교하여 IoT 기기의 업데이트 필요 여부를 확인한다. 3. 업데이트가 필요한 경우 사용자에게 이를 알리고 필요에 따라 변화된 약관 등을 고지하며 해당 업데이트를 적용할지 의사를 묻는다. 4. 업데이트에 동의할 경우 서버에서 최신 펌웨어를 다운로드 받는다. 5. 다운로드가 완료되면 해당 펌웨어를 IoT 기기에 적용 및 업데이트한다. |
|---------------|--|

표 14 Use Case - 등록된 IoT 기기의 최신 펌웨어 조회 및 업데이트

| Use case name | Smart Homepany 소프트웨어의 최신 버전 조회 및 업데이트 |
|----------------|--|
| Actor | 사용자, 모바일 디바이스, 서버 |
| Description | 사용자는 서버에서 Smart Homepany 소프트웨어의 최신 버전을 조회하고 현재 버전이 최신 버전이 아닐 경우 최신 버전을 다운로드 받아 업데이트 할 수 있도록 한다. |
| Pre-Condition | 모바일 디바이스가 서버와 통신 가능한 상태여야 한다. |
| Post-Condition | 모바일 디바이스가 서버와 통신 가능한 상태면 서버로부터 소프트웨어의 최신 버전을 조회해야만 한다. |
| Normal Course | <ol style="list-style-type: none"> 1. 사용자가 소프트웨어의 최신 버전을 서버로부터 확인한다. 2. 현재 소프트웨어의 버전과 서버 상의 최신 버전을 비교하여 소프트웨어의 업데이트 필요 여부를 확인한다. 3. 업데이트가 필요한 경우 사용자에게 이를 알리고 필요에 따라 변화된 약관 등을 고지하며 해당 업데이트를 적용할지 의사를 묻는다. 4. 업데이트에 동의할 경우 서버에서 최신 소프트웨어를 다운로드 받는다. 5. 다운로드가 완료되면 업데이트를 적용한다. |

표 15 Use Case - Smart Homepany 소프트웨어의 최신 버전 조회 및 업데이트

| Use case name | IoT 기기 탐색 |
|---------------|------------------------------------|
| Actor | 사용자, 동일 네트워크에 연결되어 있는 IoT 기기 |
| Description | 동일한 네트워크에 연결되어 있는 모든 IoT 기기를 탐색한다. |

| | |
|----------------|---|
| Precondition | 사용자와 IoT 기기가 동일한 네트워크에 연결되어 있어야 한다. IoT 기기가 본 소프트웨어에서 지원하는 규격에 맞아야 한다. 사용자는 기기 탐색 기능을 실행한다. |
| Post condition | 재탐색 버튼을 활성화한다. |
| Normal Course | <ol style="list-style-type: none"> 1. 사용자는 시스템에 IoT 기기를 등록하기 위해서 탐색을 진행한다. 2. 탐색 탭을 실행하면 시스템은 사용자와 동일한 네트워크에 연결되어 있는 기기를 모두 탐색한다. 3. 탐색이 완료되면 모든 기기의 목록을 보여준다. <div data-bbox="1081 420 1385 1136" data-label="Image"> </div> <p>그림 1 User Interface – 등록된 IoT 기기 목록</p> |
| Exceptions | 동일한 네트워크에 연결되어 있는 기기가 하나도 없는 경우 - 네트워크에 연결되어 있는 기기가 없다는 메시지를 보여주고, 재탐색 버튼을 활성화한다 |

표 16 Use Case - IoT 기기 탐색

| Use case name | IoT 기기 등록 |
|----------------|--|
| Actor | 사용자, 동일 네트워크에 연결되어 있는 IoT 기기, DB |
| Description | IoT 기기를 어플리케이션에서 조작하기 위해서 등록을 진행하고 DB에 저장한다. |
| Precondition | 기기 탐색이 완료되어 기기의 목록이 나타나 있어야 한다. |
| Post condition | 등록 완료 메시지가 나타나고 DB에 저장된다.. |
| Normal Course | <ol style="list-style-type: none"> 1. 사용자는 탐색이 완료된 기기 목록에서 등록하고자 하는 기기를 선택한다. 2. DB의 등록된 IoT Device entity table에 기기 정보를 추가한다. |

| | |
|------------|--|
| | 3. IoT 기기는 사용자의 연결된 디바이스 정보를 저장한다. |
| Exceptions | 연결이 원활하지 않은 경우 - 기기 등록 실패 메시지를 띄운다. |

표 17 Use Case - IoT 기기 등록

| Use case name | IoT 기기 삭제 |
|----------------|--|
| Actor | 사용자, 연결되어 있는 IoT 기기, DB |
| Description | 등록되어 있는 IoT 기기를 DB 에서 삭제한다. |
| Precondition | 사용자가 삭제하고자 하는 기기가 이미 시스템에 등록되어 있어야 한다. 사용자는 기기 관리 탭을 실행한다. |
| Post condition | 등록된 기기 리스트를 출력한다. |
| Normal Course | <ol style="list-style-type: none"> 1. 등록된 기기 목록을 출력한다. 2. 삭제하고자 하는 기기를 선택한다. 3. 삭제하고자 하는 기기 정보의 정보를 출력한다. 4. 사용자는 삭제 버튼을 클릭한다. 5. 시스템은 DB의 IoT Device entity table에서 기기 정보를 제거한다. 6. IoT 기기는 사용자의 디바이스 정보를 제거한다. 7. 시스템은 등록된 기기 목록에서 삭제된 기기를 제거한다. 8. 시스템은 기기가 삭제되었다는 메시지를 출력한다. |

표 18 Use Case - IoT 기기 삭제

| Use case name | 업무 알람 제어 어플리케이션 지정 |
|----------------|---|
| Actor | 사용자, DB |
| Description | '업무 알람 제어' 기능에서 제어할 앱을 지정한다. |
| Precondition | 사용자가 제어하고자 하는 앱이 설치되어 있어야 한다. 사용자는 프리셋 관리 탭을 실행한다. |
| Post condition | DB 에 '업무 알람 제어' 기능으로 제어할 앱이 저장된다. |
| Normal Course | <ol style="list-style-type: none"> 1. 현재 '업무 알람 제어' 기능으로 제어되는 어플리케이션들의 목록을 보여준다. 2. 사용자는 기존의 목록에서 제외할 앱을 클릭하여 제외하거나 추가 버튼을 클릭하여 설치된 어플리케이션들 중 목록에 없는 앱들 중 |

| | |
|--|---|
| | <p>에 추가할 앱을 선택하여 추가한다.</p> <p>3. 사용자는 적용 버튼을 눌러 변경 사항을 저장하거나 취소 버튼을 눌러 변경 사항을 취소한다.</p> <p>4. 사용자가 적용 버튼을 누른 경우 시스템은 Local DB에 삭제 내역을 전달한다.</p> |
|--|---|

표 19 Use Case – 업무 알람 제어 어플리케이션 지정

| Use case name | 프리셋 등록 |
|----------------|---|
| Actor | 사용자, 연결되어 있는 IoT 기기, DB |
| Description | 사용자는 새로운 프리셋을 등록한다. |
| Precondition | 프리셋 설정 시 원하는 IoT 기기가 동일한 네트워크에 연결되어 있어야 하고, 시스템에 등록되어 있어야 한다. 사용자는 프리셋 관리 탭을 실행한다. |
| Post condition | 시스템의 프리셋 목록이 업데이트된다. DB 의 Preset entity Table 에 새로 등록한 프리셋이 추가된다. |
| Normal Course | <ol style="list-style-type: none"> 1. 시스템은 저장되어 있는 프리셋 목록을 출력한다. 2. 사용자는 Edit 버튼을 클릭하여 프리셋 편집 모드를 실행한다. 3. 사용자는 추가 버튼을 클릭하여 새로운 프리셋 등록 모드를 실행한다. 4. 시스템은 현재 연결되어 있는 기기 목록을 출력한다. 5. 사용자는 프리셋에 포함하고자 하는 기기를 선택한다. 6. 시스템은 사용자가 선택한 기기들의 상태 값을 설정할 수 있는 창을 실행하고, 그 값은 현재 값으로 설정한다. 7. 사용자는 밝기, 온도 등 원하는 기기들의 상태 값을 설정한다. 8. 원하는 값을 모두 설정했다면 Save 버튼을 클릭하여 프리셋을 저장한다. 9. 시스템은 DB에 프리셋 정보를 전달한다. 10. DB는 Preset entity table에 새로운 행을 만들어 전달받은 정보를 저장한다. |

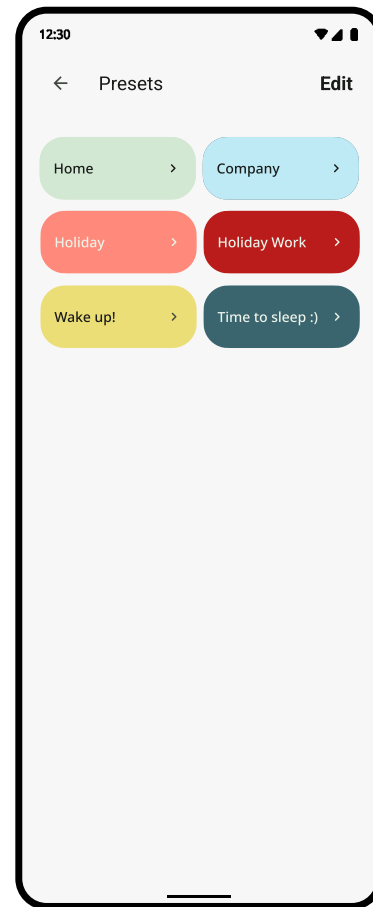
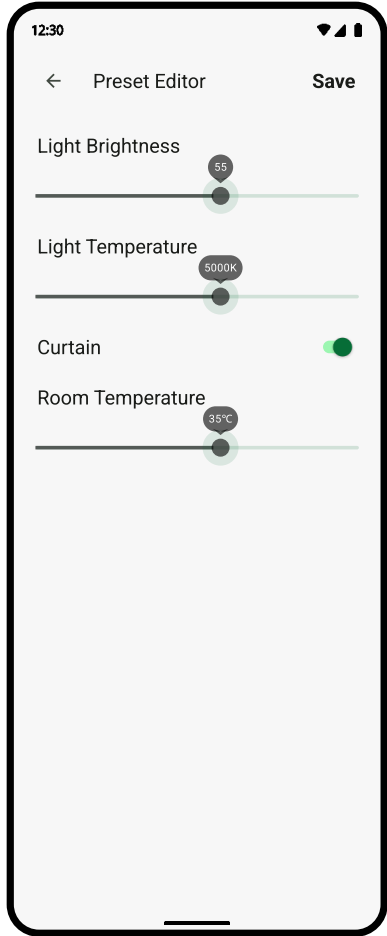


그림 2 User Interface - 프리셋 목록 및 등록

표 20 Use Case - 프리셋 등록

| Use case name | 프리셋 편집 |
|--------------------|---|
| Actor | 사용자, 연결되어 있는 IoT 기기, DB |
| Description | 사용자는 기존에 등록되어 있던 프리셋을 편집한다. |
| Precondition | 사용자가 편집하고자 하는 프리셋이 등록되어 있어야 한다. IoT 기기가 동일한 네트워크에 연결되어 있어야 하고, 시스템에 등록되어 있어야 한다. 사용자가 프리셋 관리 탭을 실행한다. |
| Post condition | 프리셋에서 변경을 원하는 값이 정상적으로 변경되어 DB 에 저장되어야 한다. |
| Normal Course | <ol style="list-style-type: none"> 1. 시스템은 저장되어 있는 프리셋 목록을 출력한다. 2. 사용자는 Edit 버튼을 클릭하여 프리셋 편집 모드를 실행한다. 3. 사용자는 편집을 원하는 프리셋을 선택한다. 4. 시스템은 사용자가 선택한 프리셋의 세부 정보를 출력한다. 5. 사용자는 변경하고자 기기의 상태 값을 변경한다. 6. 원하는 값을 모두 변경하였다면 Save 버튼을 클릭하여 변경된 프리셋을 저장한다. 7. 시스템은 DB에 변경된 프리셋 정보를 전달한다. 8. DB는 Preset entity table에 저장되어 있는 해당 행을 변경하여 저장한다.  <p style="text-align: right;">그림 3 User Interface – 프리셋 편집</p> |
| Alternative Course | <p>프리셋에 새로운 기기를 등록하고자 하는 경우 아래 순서를 따른다.</p> <ol style="list-style-type: none"> 1. 사용자는 새로운 기기 추가 버튼을 클릭한다. 2. 시스템은 현재 연결되어 있는 기기 목록을 출력한다. 3. 사용자는 등록을 원하는 기기를 선택한다. 4. 시스템은 사용자가 선택한 기기를 프리셋 편집 페이지에 추가한다. |

| | |
|--|------------------------------|
| | 5. Normal Course의 5번부터 수행한다. |
|--|------------------------------|

표 21 Use Case - 프리셋 편집

| Use case name | 프리셋 삭제 |
|----------------|--|
| Actor | 사용자, DB |
| Description | 사용자는 기존에 등록되어 있던 프리셋을 삭제한다. |
| Precondition | 사용자가 삭제하고자 하는 프리셋이 등록되어 있어야 한다. 사용자가 프리셋 관리 탭을 실행한다. |
| Post condition | 삭제하고자 하는 프리셋이 시스템의 프리셋 목록과 DB 에서 삭제되어야 한다. |
| Normal Course | <ol style="list-style-type: none"> 1. 시스템은 저장되어 있는 프리셋 목록을 출력한다. 2. 사용자는 Edit 버튼을 클릭하여 프리셋 편집 모드를 실행하고, 삭제하고자 하는 프리셋의 삭제 버튼을 선택한다. 3. 시스템은 사용자에게 삭제 전 경고 알림 메시지를 출력한다. 4. 사용자는 삭제 버튼을 클릭한다. 5. 시스템은 프리셋을 삭제하고 DB에 전달한다. 6. DB의 Preset entity table에서 해당 프리셋 정보를 제거한다. 7. 시스템은 삭제 완료 알림 메시지를 출력한다. |

표 22 Use Case - 프리셋 삭제

| Use case name | 프리셋 예약 |
|----------------|---|
| Actor | 사용자, DB |
| Description | 사용자는 등록되어 있는 프리셋을 원하는 시간대에 예약하고 활성화한다. 활성화된 프리셋은 사용자가 지정한 시간대에 실행되고, 프리셋에 설정된 값에 따라 IoT 기기들의 상태가 조작된다. |
| Precondition | 사용자가 예약하고자 하는 프리셋이 등록되어 있어야 한다. 사용자가 예약 관리 탭을 실행한다. |
| Post condition | DB 에 예약한 프리셋 정보가 저장된다. 설정한 요일의 해당 시간이 되면 프리셋에 설정되어 있는 상태 값에 맞춰 IoT 기기의 상태가 변화해야 한다. 설정한 요일이 되면 시스템의 타임라인 탭에 예약된 프리셋 목록이 출력되어야 한다. |

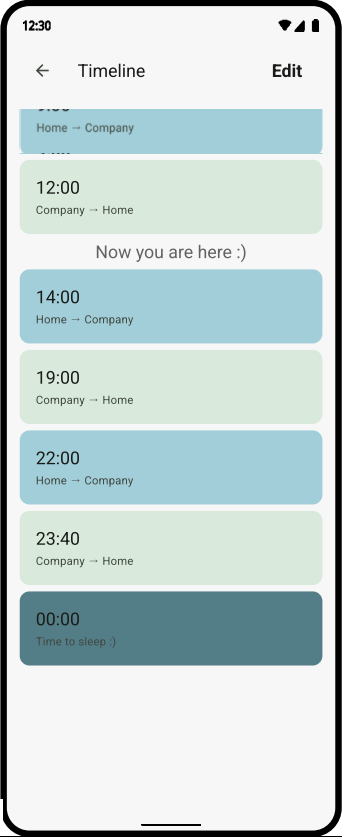
| | | |
|--------------------|--|--|
| Normal Course | <ol style="list-style-type: none"> 1. 현재 기기에 등록되어 있는 프리셋 예약 목록이 사용자에게 출력된다. 2. 사용자는 예약 추가 버튼을 클릭한다. 3. 예약하고자 하는 프리셋을 선택하고, 예약하고자 하는 요일, 시간을 설정한다. 4. 시스템은 프리셋 예약 정보를 DB에 전달한다. |  |
| Alternative Course | <p>프리셋 예약을 특정 날짜에만 반복하고자 하는 경우</p> <ol style="list-style-type: none"> 1. 예약하고자 하는 프리셋을 선택했을 때 고급 설정 메뉴에서 날짜를 선택한다. 2. Normal Course 의 3 으로 돌아간다. | |
| Exceptions | <p>등록하고자 하는 시간에 다른 프리셋이 이미 예약되었고, 활성화되어 있는 경우</p> <ol style="list-style-type: none"> 1. 해당 요일의 해당 시간에 다른 프리셋이 예약되어 있다는 메시지를 보여준다. 2. 요일을 수정할 수 있도록 커서를 이동시킨다. | |

표 23 Use Case - 프리셋 예약

| Use case name | 프리셋 예약 편집 |
|----------------|---|
| Actor | 사용자, DB |
| Description | 사용자는 프리셋 예약 정보를 편집하거나 비활성화할 수 있다. |
| Precondition | 사용자가 편집하고자 하는 예약정보가 등록되어 있어야 한다. 사용자가 예약 관리 탭을 실행한다. |
| Post condition | DB 에 저장된 프리셋 정보가 업데이트된다. |

| | |
|---------------|---|
| | 변경된 정보에 맞춰 타임라인 탭의 정보가 업데이트된다. |
| Normal Course | <ol style="list-style-type: none"> 1. 현재 기기에 등록되어 있는 프리셋 예약 목록이 사용자에게 출력된다. 2. 사용자는 편집하고자 하는 예약 내역을 선택하고 편집 버튼을 클릭한다. 3. 변경하고자 하는 요일, 시간 정보 등을 변경한다. 4. 시스템은 변경된 프리셋 예약 정보를 DB에 전달한다. |

표 24 Use Case - 프리셋 예약 편집

| Use case name | 프리셋 예약 삭제 |
|----------------|--|
| Actor | 사용자, DB |
| Description | 사용자는 등록되어 있는 프리셋을 예약 내역을 삭제한다. |
| Precondition | <p>사용자가 삭제하고자 하는 프리셋 예약 정보가 등록되어 있어야 한다.</p> <p>사용자는 예약 관리 탭을 실행한다.</p> |
| Post condition | DB 에서 프리셋 예약 정보가 삭제된다. |
| Normal Course | <ol style="list-style-type: none"> 1. 현재 기기에 등록되어 있는 프리셋 예약 목록이 사용자에게 출력된다. 2. 사용자는 삭제하고자 하는 예약 내역을 선택하고 편집 버튼을 클릭한 후 삭제 버튼을 클릭한다. 3. 시스템은 사용자에게 삭제 확인 메시지를 보여준다. 4. 사용자는 메시지를 확인하고 확인 버튼을 선택한다. 5. 시스템은 DB에 삭제 내역을 전달한다. |

표 25 Use Case - 프리셋 예약 삭제

| Use case name | 프리셋 예약 비활성화 |
|---------------|---|
| Actor | 사용자, 연결된 IoT 기기 |
| Description | 사용자는 타임라인 탭에서 오늘 예약되어 있는 프리셋 실행을 일시적으로 비활성화한다. |
| Precondition | <p>사용자가 비활성화 하고자 하는 프리셋 예약 정보가 등록되어 있어야 한다.</p> <p>사용자는 타임라인 탭을 실행한다.</p> |

| | |
|----------------|--|
| Post condition | 프리셋 예약 내역이 비활성화되어, 예약했던 시간이 되어도 실행되지 않는다. |
| Normal Course | <ol style="list-style-type: none"> 1. 오늘 실행되어야 하는 프리셋 예약 내역이 시간 순으로 사용자에게 출력된다. 2. 사용자는 비활성화하고자 하는 예약 내역을 선택한다. 3. 시스템은 해당 예약 내역이 비활성화되었음을 표시하고, 예약 시간이 되어도 IoT 기기에 실행 명령을 내리지 않는다. |

표 26 Use Case - 프리셋 예약 비활성화

3.2.2 Use Case Diagram

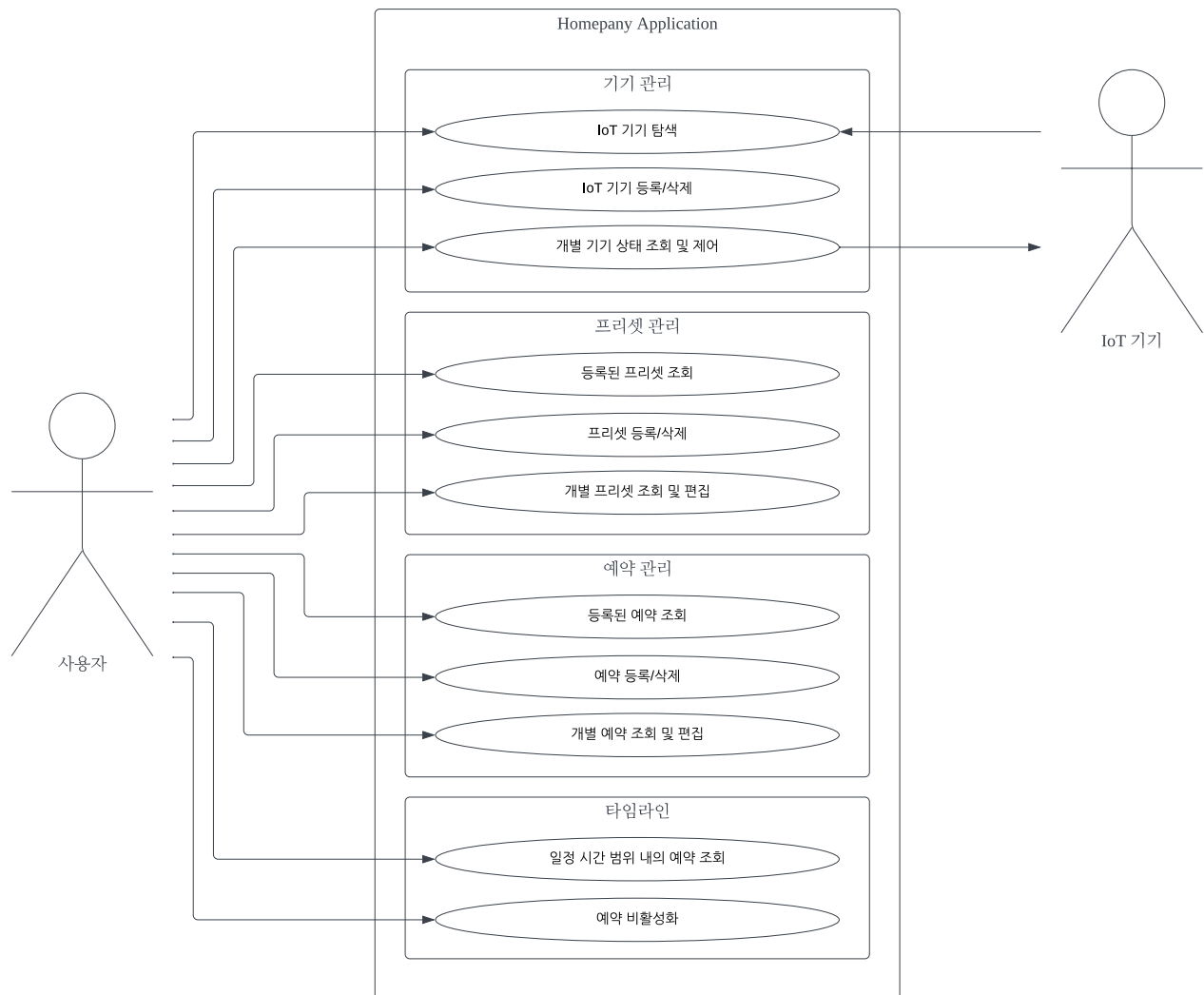


그림 5 Use Case Diagram

3.2.3 Data Dictionary

| Field | Primary | Foreign | Constraint | Type | Description |
|----------|---------|---------|------------|----------|--------------|
| user_id | ○ | | NOT NULL | Char(20) | 사용자 기기 식별 id |
| nickname | | | NOT NULL | Char(10) | 사용자 기기 닉네임 |

표 27 Data Dictionary - User

| Field | Primary | Foreign | Constraint | Type | Description |
|-----------|---------|---------|------------|----------|----------------------|
| device_id | ○ | | NOT NULL | Char(20) | IoT 기기 식별 id |
| firmware | | | NOT NULL | Char(30) | 기기의 firmware version |

표 28 Data Dictionary - IoT Device

| Field | Primary | Foreign | Constraint | Type | Description |
|-------------|---------|---------|------------|----------|-----------------|
| device_id | ○ | ○ | NOT NULL | Char(20) | 등록된 기기의 식별 id |
| con_user_id | | ○ | | Char(20) | 기기와 연결된 user id |
| nickname | | | | Char(10) | 기기의 nickname |
| cur_status | | | | int(3) | 기기의 현재 상태 |

표 29 Data Dictionary - Connected Device

| Field | Primary | Foreign | Constraint | Type | Description |
|-------------|---------|---------|----------------------|----------|---------------|
| preset_id | ○ | | NOT NULL | Char(10) | 등록된 preset id |
| preset_name | | | DEFAULT '이름없음' | Char(10) | 등록된 프리셋의 이름 |
| color | | | DEFAULT '#000000' | Char(7) | 등록된 프리셋의 색상값 |
| owner | | ○ | NOT NULL | Char(20) | 생성한 사용자 |

표 30 Data Dictionary - Preset

| Field | Primary | Foreign | Constraint | Type | Description |
|-----------|---------|---------|------------|----------|--------------------|
| device_id | | ○ | NOT NULL | Char(20) | preset 에 등록된 기기 id |
| preset_id | ○ | ○ | NOT NULL | Char(10) | 기기가 속해있는 preset id |

표 31 Data Dictionary - Preset Device

| Field | Primary | Foreign | Constraint | Type | Description |
|----------------|---------|---------|------------|----------|-------------|
| reservation_id | O | | NOT NULL | Char(20) | 예약 id |
| nickname | | | | Char(20) | 예약 nickname |
| preset_id | | O | | Char(10) | 예약된 preset |

표 32 Data Dictionary – Reservation

| Field | Primary | Foreign | Constraint | Type | Description |
|------------------|---------|---------|------------|----------|-------------|
| reservation_date | | | NOT NULL | datetime | 예약된 날짜 |
| reservation_id | O | O | NOT NULL | int(3) | 예약 id |

표 33 Data Dictionary - Reservation_repetition

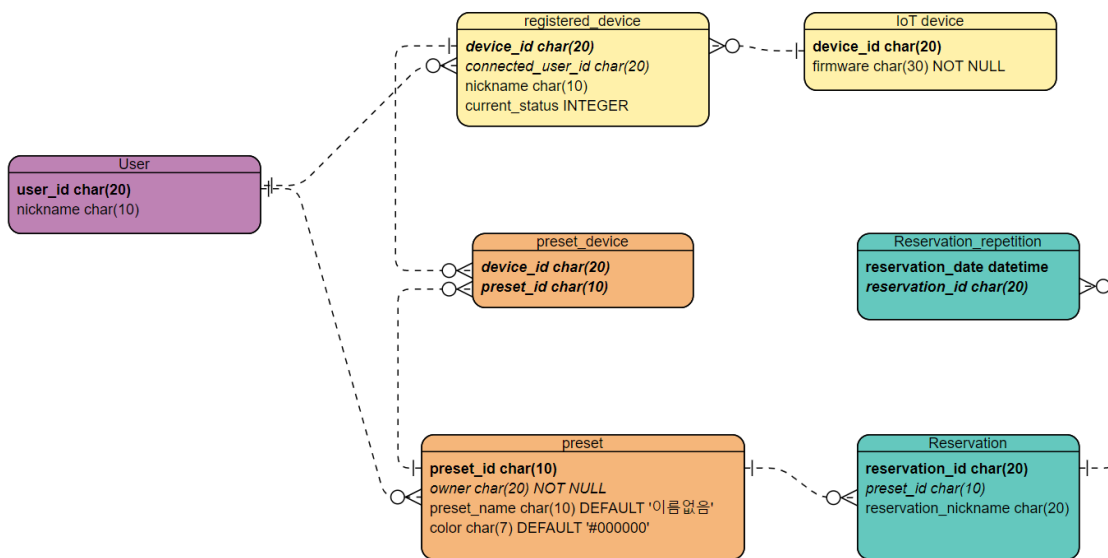


그림 6 Data Dictionary

3.2.4 Data Flow Diagram

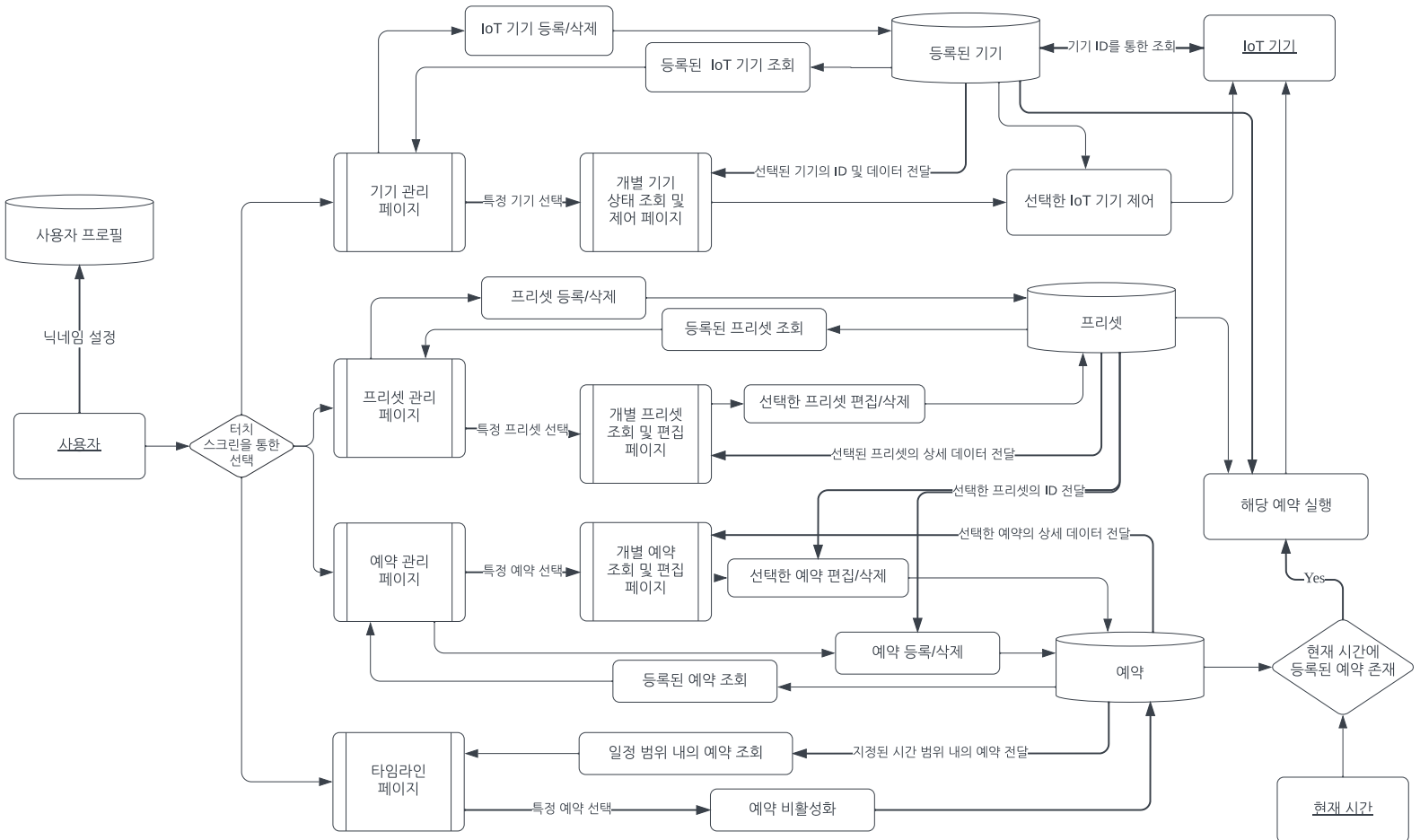


그림 7 Data Flow Diagram

3.3 Nonfunctional Requirements

다음은 비기능적 요구사항들에 대하여 다룬다.

3.3.1 Product Requirements

하기 사항들은 본 요구사항 명세서에서 다루고 있는 Smart Homepany 소프트웨어가 어떠한 방식으로 작동해야 하는지를 규정한다. 내용은 성능 요구사항, 공간 요구사항, 보안 요구사항, 사용성 요구사항, 신뢰성 요구사항으로 구성된다.

3.3.1.1 Performance Requirements

다음은 제품의 성능과 관련된 요구사항들을 다룬다. 다음의 요구사항들은 반드시 준수하여야 하며, 만약 하기 사항들이 지켜지지 않을 경우 Smart Homepany 소프트웨어는 개발자 및 사용자가 의도한 대로 동작하지 않을 수 있으며, 서비스 이용에 차질이 있을 수 있다.

- 본 소프트웨어의 최소 요구 사양은 Android 10 Queen Cake (API 29) 혹은 iOS 13.0 운영체제와 1GB의 RAM 용량과 1.2GHz의 CPU 클럭 속도이며, 권장 요구 사양은 Android 12 Snow Cone (API 31) 이상 혹은 iOS 15.0 이상의 운영체제와 2GB 이상의 RAM 용량과 1.6GHz 이상의 CPU 클럭 속도이다.
- 본 소프트웨어는 최소 사양을 기준으로 실행 직후 최대 2초 이내에 기기 고유 번호를 인식하고 메인 화면으로 진입하여야 한다.
- 본 소프트웨어는 최소 사양을 기준으로 IoT 기기는 최대 3초 이내에 사용자의 명령에 대응되는 액션을 수행하여야 한다.
- 본 소프트웨어는 최소 사양을 기준으로 프리셋을 추가 및 수정할 시 최대 0.5초 이내에 서버 간 데이터 트랜잭션 및 업데이트가 완료되어야 한다.
- 본 소프트웨어는 최소 사양을 기준으로 예약된 프리셋을 실행할 시 최대 1초 이내에 각 IoT 기기에 실행 요청 신호를 전달하여야 한다.
- 본 소프트웨어와 IoT 기기 간에는 최소 100Mbps의 신호 전달 속도를 지원해야 한다.

3.3.1.2 Space Requirements

다음은 제품의 공간 요구사항을 다룬다.

- 본 소프트웨어는 모바일 디바이스에서 최대 600MB의 저장 공간을 차지할 수 있다.
- 본 소프트웨어는 모든 기능을 수행함에 있어서 200MB 이상의 메모리를 점유하지 않아야 한다.
- 본 소프트웨어에서 사용자가 만든 프리셋은 모바일 디바이스의 고유 번호를 기준으로 원격 저장소(서버)에 저장되어야 한다.

3.3.1.3 Security Requirements

다음은 제품의 보안을 유지하기 위한 요구사항을 다룬다.

- 한 IoT 기기는 반드시 하나의 모바일 디바이스에만 연결될 수 있다.
- 한 모바일 디바이스는 여러 개의 IoT 기기와 연동될 수 있으나, 다른 기기와 이미 연동되어 있는 IoT 기기에는 접근할 수 없다.
- 악의적인 사용자가 소프트웨어와 IoT 기기 간 통신에서 발생하는 데이터를 인터셉트 할 수 없어야 한다.
- 소프트웨어와 IoT 기기 간 통신에서 데이터의 무결성이 보장되어야 한다.
- 본 소프트웨어를 실행하기 위해서는 사용자가 기존에 설정해 둔 비밀번호, 패턴 인식, 지문 인식 등의 보안 인증 과정을 거쳐야 한다.
- IoT 기기를 등록할 때 매번 접근 권한을 허용할 지 혹은 한 번만 등록한 이후에 앞으로도 계속 해당 권한을 유지할 지에 대하여 사용자에게 결정권을 부여한다.

3.3.1.4 Usability Requirements

다음은 제품의 사용성과 관련된 요구사항들을 다룬다. 아래 요구사항들은 사용자가 제품을 편리하고 올바르게 사용할 수 있게끔 한다.

- 사용자는 별도의 설명서 없이도 본 소프트웨어를 쉽게 사용할 수 있어야 한다.
- 사용자가 밝은 버전(라이트 모드)과 어두운 버전(다크 모드)의 화면을 자유롭게 선택할 수 있게 해야 하며, 별도의 선택이 없다면 모바일 디바이스의 기본 설정에 따르도록 한다.
- 사용자 인터페이스는 사용자 경험을 기반으로 편리한 사용감을 제공해야 한다.
- 모든 기능의 작동 방식이 쉽고 직관적이어야 한다.
- 각 IoT 기기에 대한 제어 설정 화면이 간결하고 가독성이 좋아야 한다.
- Timeline 화면이 시간대별로 실행되는 프리셋을 단번에 파악할 수 있을 만큼 직관적이어야 한다.

3.3.1.5 Dependability Requirements

다음은 제품의 신뢰성과 관련된 요구사항들을 다룬다. 제품이 제공하는 모든 기능들은 본래 의도한 바를 벗어나서 작동하면 안 된다.

- IoT 기기의 설정 값이 예기치 못한 원인으로 수정될 경우 실행되고 있는 프리셋은 종료되고 사용자에게 이를 알려야 한다.
- 소프트웨어에서 구동하고 있는 프리셋을 기준으로, 모든 IoT 기기는 사용자가 원하는 설정 값에 맞게 항상 무결한 동작을 수행해야한다.

3.3.2 Organizational Requirements

하기 사항들은 본 요구사항 명세서에서 다루고 있는 Smart Homepany 소프트웨어가 소프트웨어를 사용하는 고객, 고객이 속한 조직, 그리고 고객이 처한 환경적 상황을 고려하여 어떠한 방식으로 규제될 수 있는지 다루고 있다. 내용은 환경적 요구사항, 운영 요구사항, 개발 요구사항으로 구성된다.

3.3.2.1 Environmental Requirements

다음은 제품이 사용되는 환경과 관련된 요구사항들을 다룬다.

- 본 소프트웨어를 사용하기 위해 사용자는 소프트웨어에서 지원하는 IoT 기기를 최소 1개 이상 구비하여야 한다.
- 본 소프트웨어는 기본적으로 자택 내에서 동작하는 것을 전제로 한다.
- 가구 인원이 2인 이상이라도 실제 Smart Homepany의 IoT 기기 제어 권한을 가질 수 있는 인원은 1명으로 전제한다.

3.3.2.2 Operational Requirements

다음은 제품의 운영과 관련된 요구사항들을 다룬다. 사용자의 안전과 편안한 환경조성이 동시에 고려되어야 한다.

- 본 소프트웨어에서 기본으로 제공되는 주거 환경 모드 프리셋과 업무 환경 모드 프리셋은 사용자가 임의로 변경할 수 없다. 다만 경우에 따라 해당 프리셋들은 서버로부터 업데이트가 이루어질 수 있다.
- 사용자는 기본 프리셋 외에 스스로 커스터마이징한 모든 프리셋들을 수정 및 삭제할 수 있다.
- 각 사용자가 커스터마이징한 프리셋은 원격 저장소(서버)에 저장된다.
- 사용자 식별은 모바일 디바이스의 고유 번호를 통해 이루어지며, 해당 정보를 기반으로 서버에 프리셋 정보가 저장된다.

3.3.2.3 Development Requirements

다음은 제품의 개발 요구사항들을 다룬다. 제품 개발은 최소한 아래에서 제공하는 요구사항들을 만족하여야 한다.

- 본 소프트웨어 개발 과정에는 충분한 테스트 과정이 포함되어야 한다.
- 본 소프트웨어는 최소 Android 10 Queen Cake (API 29), 권장 Android 12 Snow Cone (API 31) 이상의 운영체제를 기반으로 개발되어야 한다.
- 본 소프트웨어는 최소 iOS 13.0, 권장 iOS 15.0 이상의 운영체제를 기반으로 개발되어야 한다.

다.

- Android 어플리케이션 개발은 Android Studio 2021.1.1 Patch 2 (Bumblebee) 환경에서 이루어진다.
- iOS 어플리케이션 개발은 Xcode 13.3.1 환경에서 이루어진다.

3.3.3 External Requirements

하기 사항들은 본 요구사항 명세서에서 다루고 있는 Smart Homepany 소프트웨어의 외부 시스템에 대한 규제 내용들을 다루고 있다. 내용은 규제 요구사항, 윤리적 요구사항, 회계 요구사항, 안전 및 보안 요구사항을 다룬다.

3.3.3.1 Regulatory Requirements

다음은 제품의 규제와 관련된 요구사항들을 다룬다.

- 본 제품에 사용되는 IoT 기기는 안전성 테스트를 거쳐서 신뢰성을 확보해야 한다.
- 본 소프트웨어는 사용자가 아닌 타인의 IoT 기기와 연결해서 악의적인 용도로 사용되어서는 안된다.

3.3.3.2 Ethical Requirements

다음은 제품의 개발과 사용에 있어서 비윤리적인 행위를 금하는 요구사항을 다룬다.

- 본 제품을 구매하지 않은 사람이 제품을 불법적으로 복제해서 사용해서는 안 된다.
- 사용자가 악의적으로 제품을 활용하여 동거인 혹은 주변인에게 범법 행위를 하거나 불쾌감을 조성하는 일이 없어야 한다.

3.3.3.3 Accounting Requirements

다음은 소프트웨어의 개발 및 사용 과정에서 발생할 수 있는 금전적인 문제에 대하여 다룬다.

- 본 시스템이 개발되고 운영되는 데 있어서 사용자에게 사전에 고지하지 않은 추가적인 금전적 요구는 없어야 한다.
- 시스템의 개발과 운영에 사용되는 비용은 명확하게 공개되어야 한다.

3.3.3.4 Safety/Security Requirements

다음은 안전/보안과 관련된 요구사항들을 다룬다.

- 본 제품에서 사용되는 모든 개인정보는 암호화되어야 한다.
- 본 제품에 사용자의 개인정보를 유출하는 백도어가 삽입되어서는 안된다.
- 본 시스템은 개인정보보호법을 준수하여 사용자의 개인정보는 사용자가 동의한 범위 내에서만 사용되어야 한다.

3.4 Organizing the Specific Requirements

3.4.1 Context Model

Context Model을 통해 Smart Homepany 시스템의 구성을 확인할 수 있다.

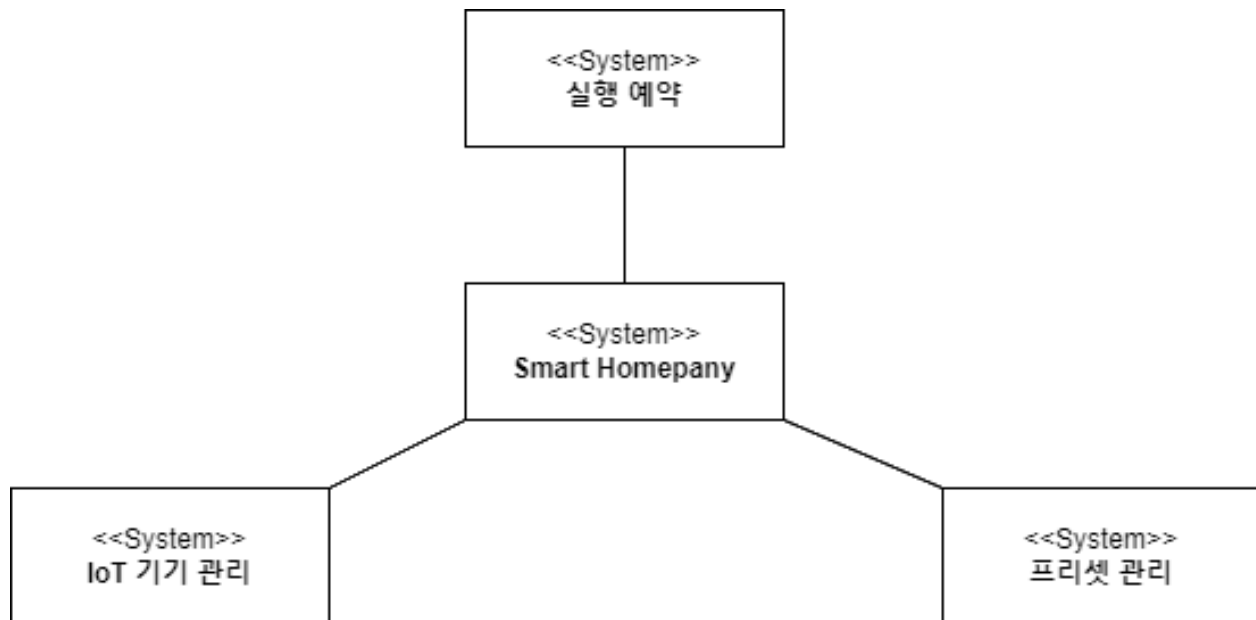


그림 8 Context Model

3.4.2 Process Model

Process Model 을 통해 각 프로세스가 수행하는 기능을 확인할 수 있다.

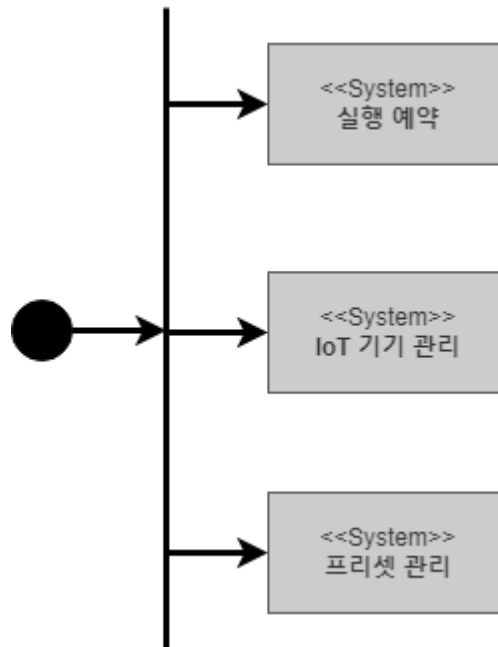


그림 9 Process Model (Main)

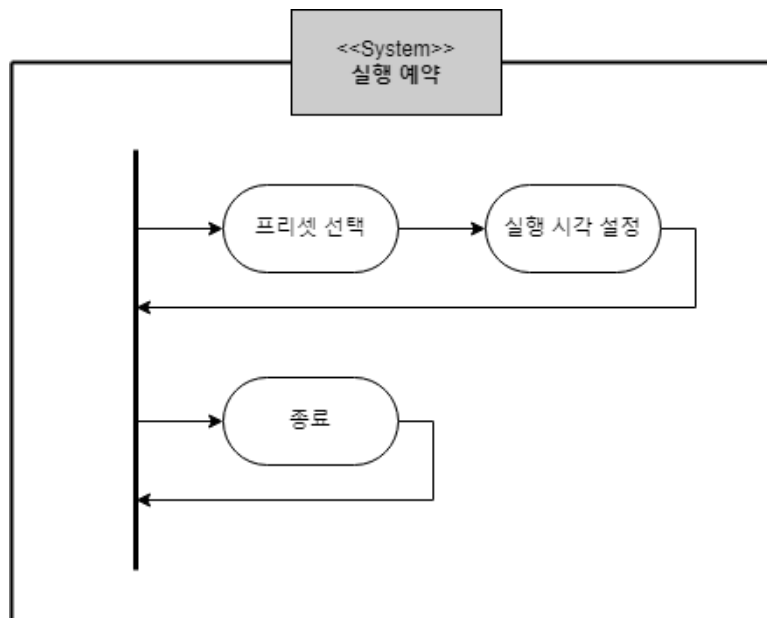


그림 10 Process Model (프리셋 예약)

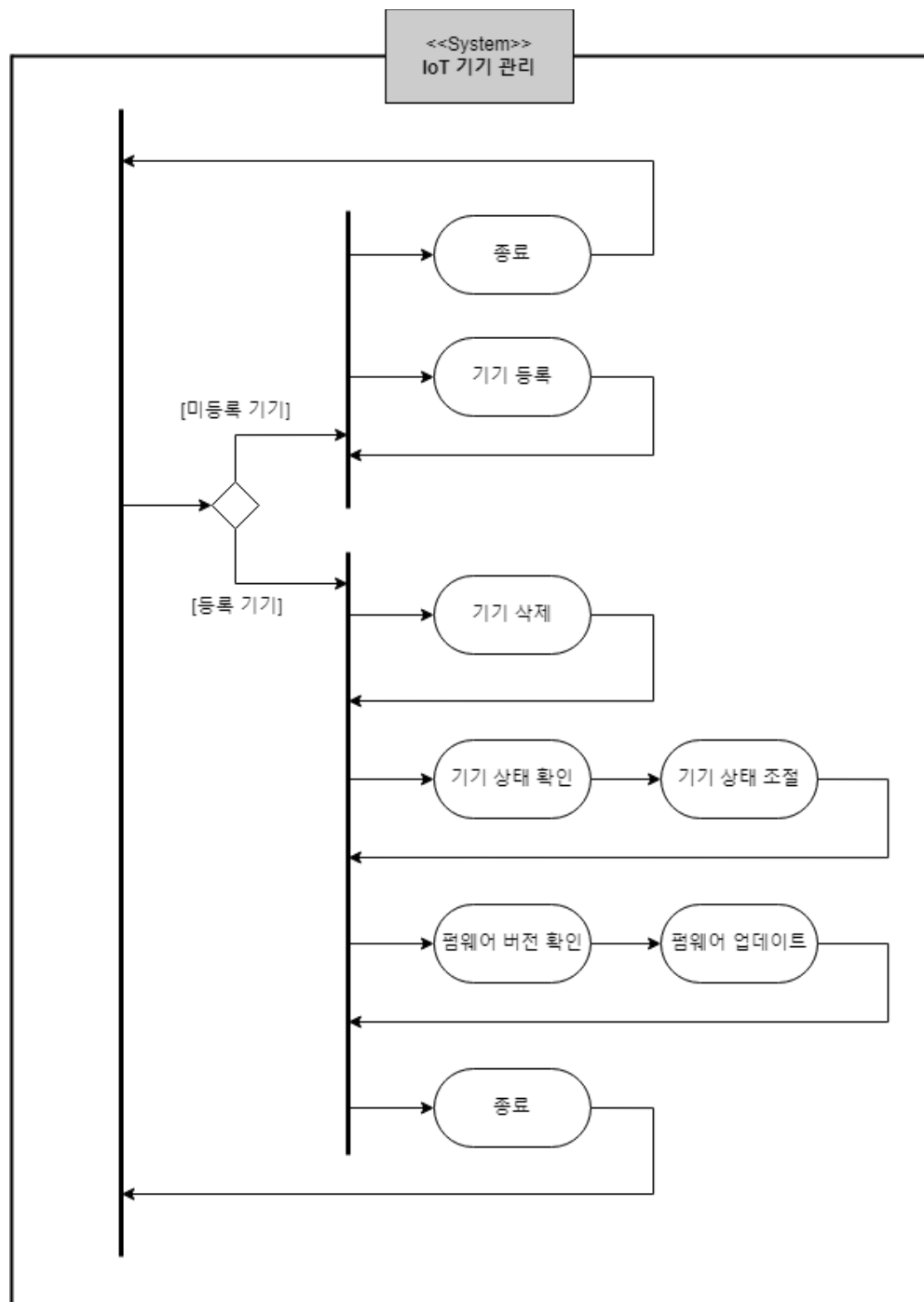


그림 11 Process Model (IoT 기기 관리)

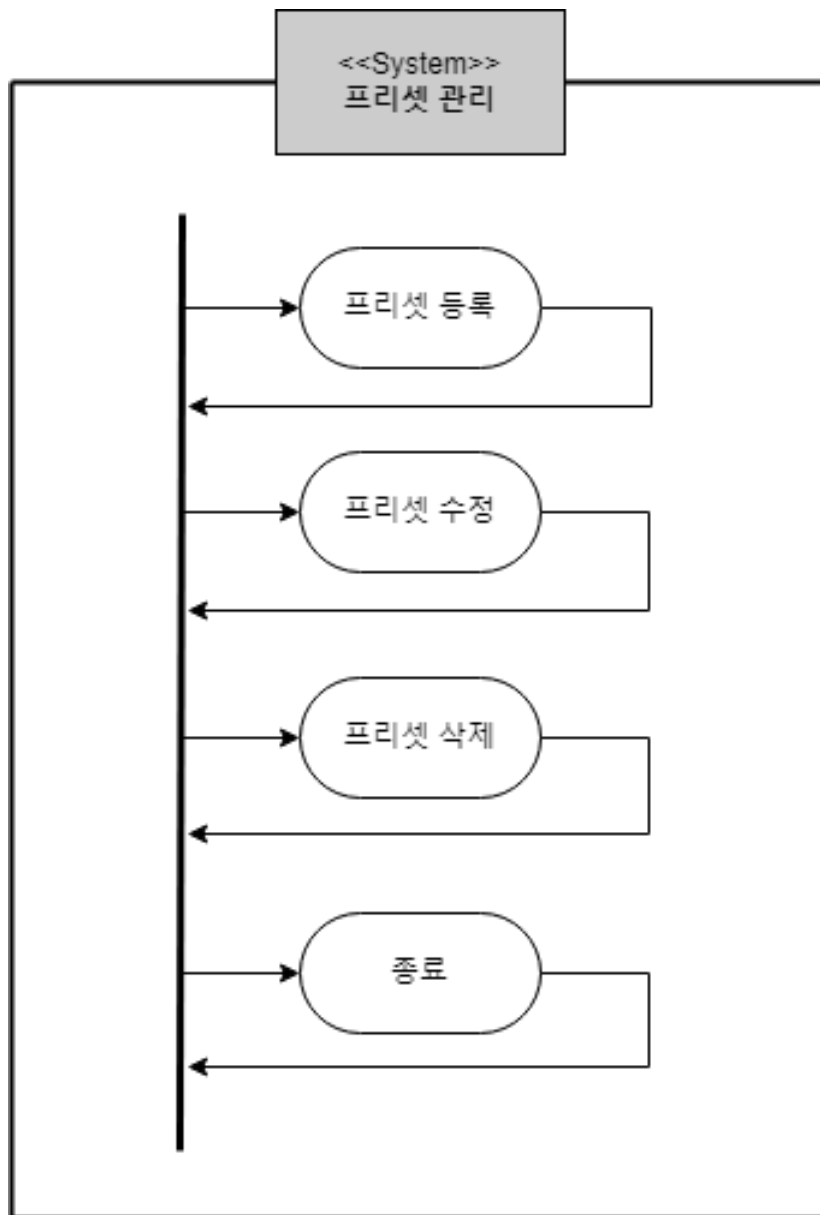


그림 12 Process Model (프리셋 관리)

3.4.3 Interaction Model

3.4.3.1 Use Case Diagram

Use Case Diagram 을 통해 Actor 와 시스템 간의 관계 및 상호작용을 확인할 수 있다.

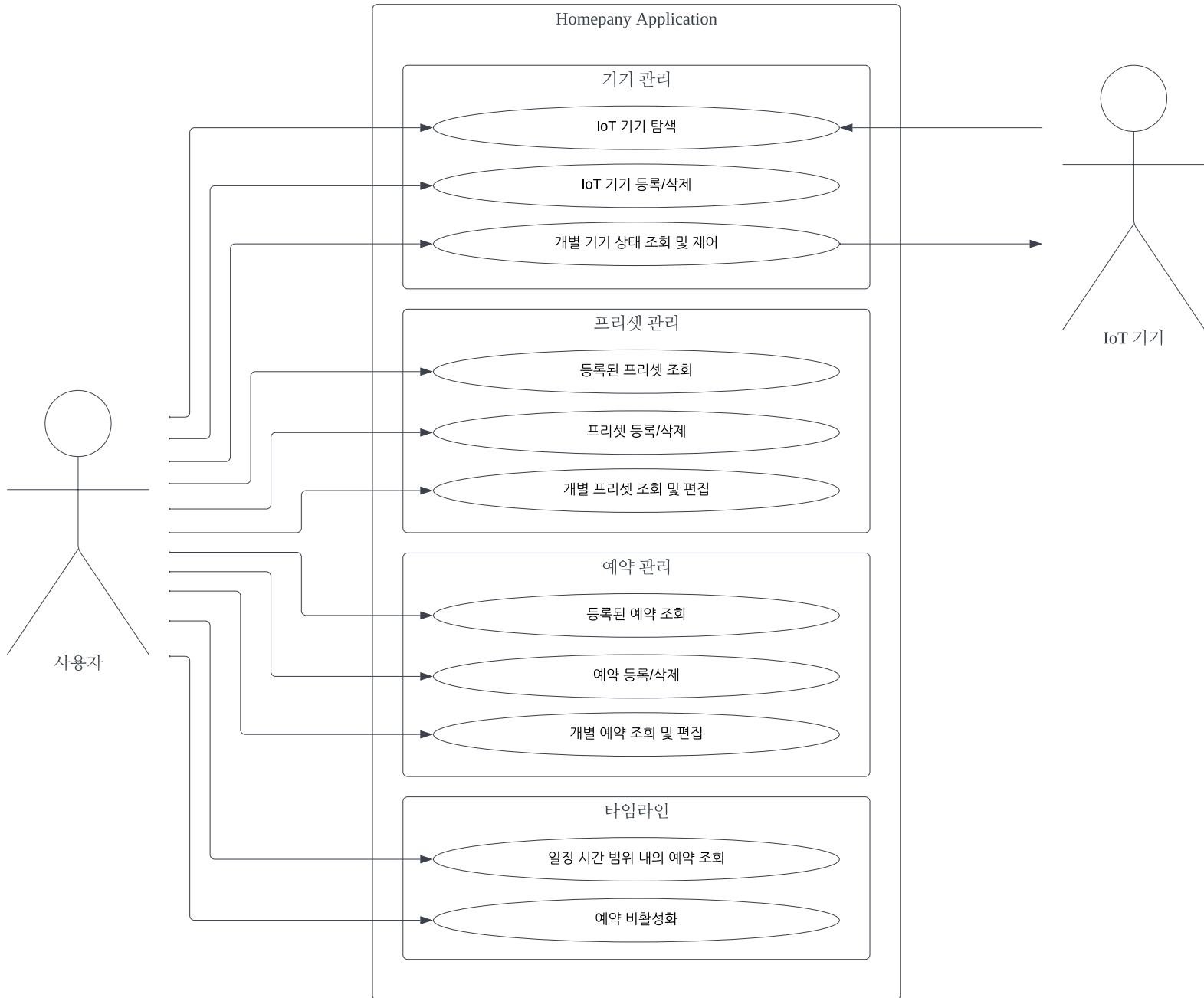


그림 13 Use Case Diagram (그림 5 동일)

3.4.4 Behavior Model

3.4.4.1 Data Flow Diagram

Data Flow Diagram 을 통해 시스템 내부의 데이터 흐름을 확인할 수 있다.

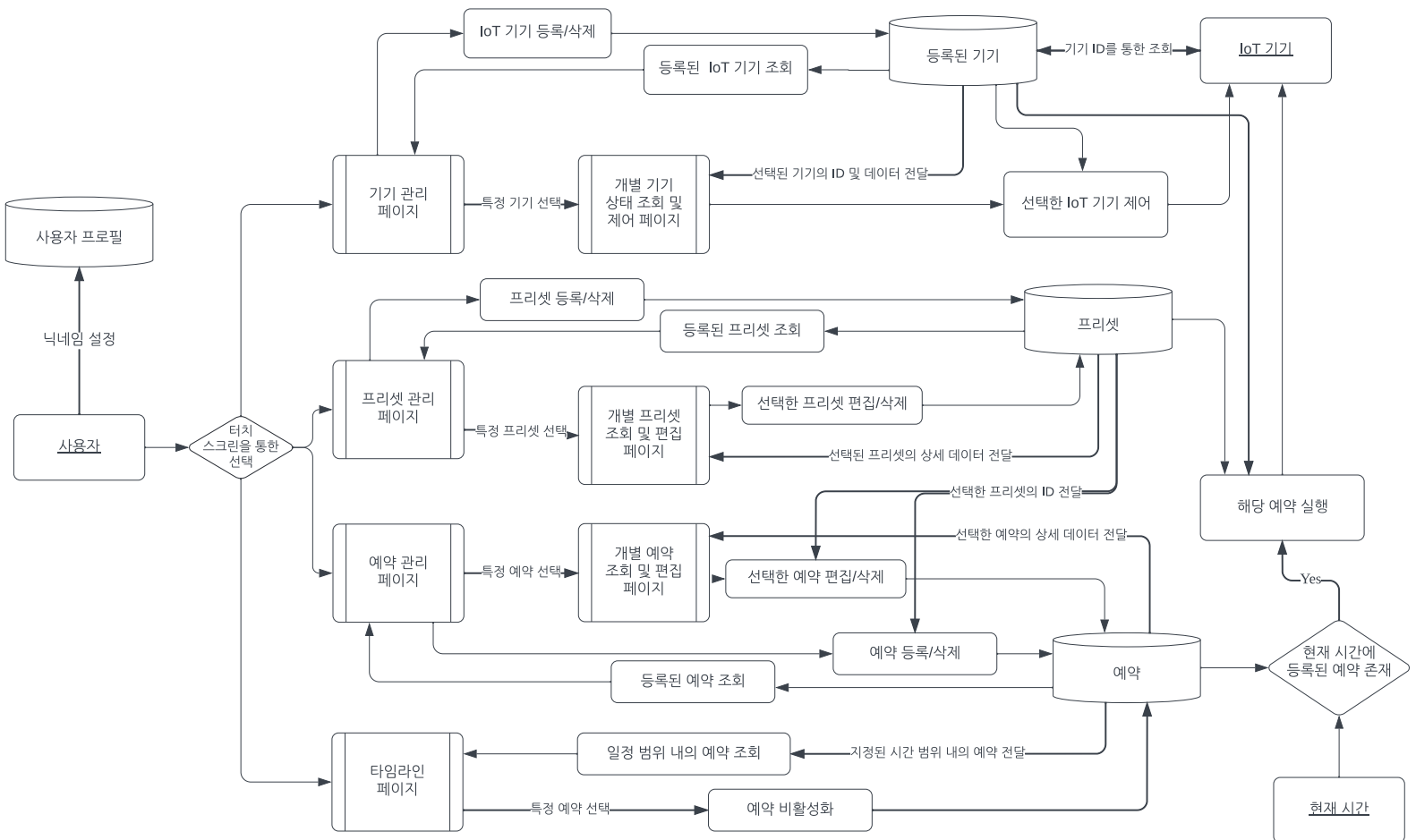


그림 14 Data Flow Diagram (그림 7 동일)

3.4.4.2 Sequence Diagram

Sequence Diagram 을 통해 시간의 흐름에 따른 기능의 수행을 확인할 수 있다.

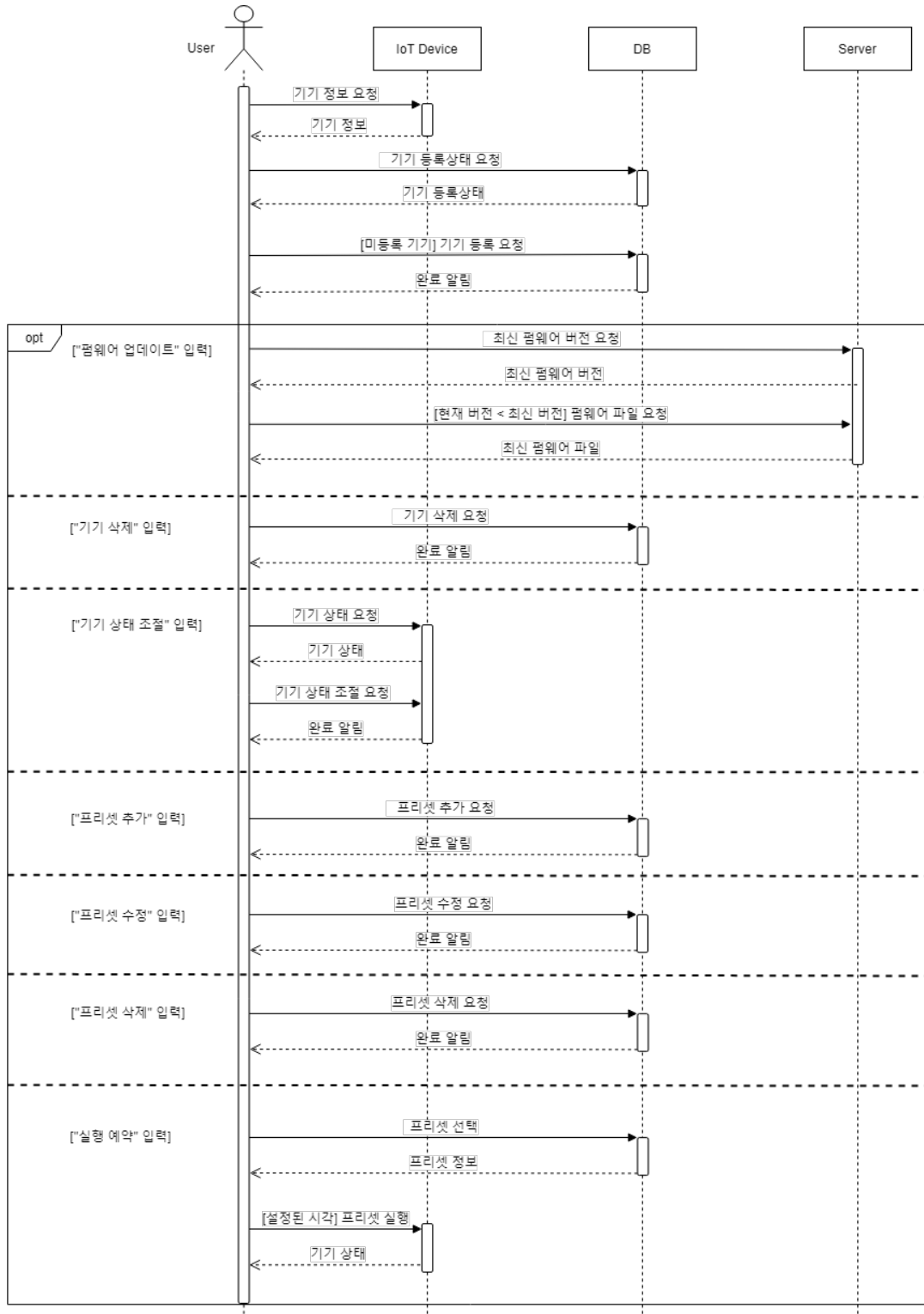


그림 15 Sequence Diagram (Overall)

3.5 System Architecture

아래에서는 MVC 패턴을 활용한 System Architecture 를 표현한다. Model 은 DB 역할을 하며 데이터 변경 시 View 에 변경 사항을 반영한다. Controller 는 이벤트를 처리하는 부분으로 처리 결과에 따른 변경 사항을 View 와 Model 에 반영한다. 마지막으로 View 는 사용자가 직접 볼 수 있는 부분으로 데이터 입력 등의 이벤트 발생 시 Controller 로 전달한다.

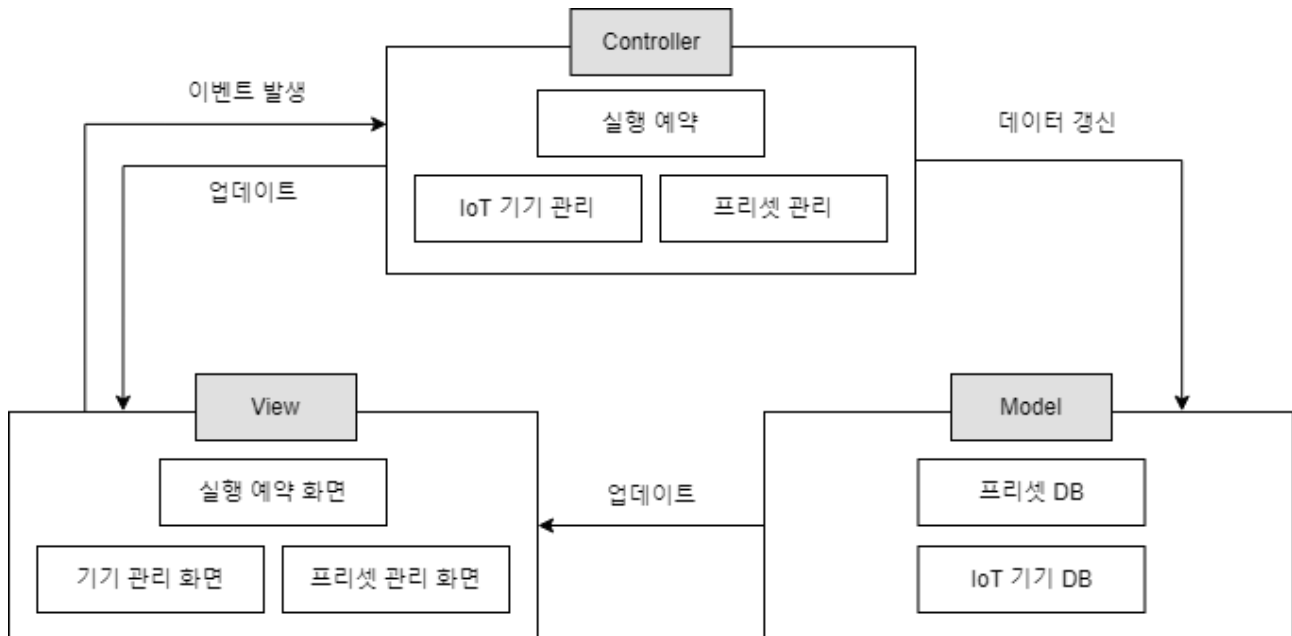


그림 16 System Architecture (MVC Pattern)

3.6 System Evolution

이 장에서는 시스템의 기본 가정과 예상되는 요구 변화 및 그 원인에 대해 서술함으로써, 추후 발생할 시스템 변경에 있어 설계가 용이하도록 한다.

3.6.1 Assumptions

본 명세서의 시스템은 재택근무자를 위한 스마트홈이며, 재택근무에 특화된 스마트홈이 시장에 존재하지 않는 상황과 이로 인한 문제 해결을 전제로 하여 설계되었다.

3.6.2 Change of Requirements

- IoT 기기, 모바일 디바이스 (스마트 기기)의 발전 및 내부 소프트웨어 변경으로 인한 요구 변화
- 새로운 IoT 기기의 도입으로 인한 요구 변화
- 새로운 통신 프로토콜로 인한 요구 변화
- 기능 추가 및 변경 필요에 따른 요구 변화

4. Supporting Information

4.1 Software Requirement Information

본 소프트웨어 요구사항 명세서 (Software Requirements Specification, SRS) 는 IEEE Recommendation (IEEE Recommended Practice for Software Requirements Specifications, IEEE Std 830-1993) 에 의거하여 작성되었다.

4.2 Document History

| Date | Version | Description | Writer |
|------------|---------|--------------|--------|
| 2022/04/02 | 0.0.0 | 문서 작업 시작 | 전원 |
| 2022/04/04 | 0.0.1 | 1, 2-1 | 이미소 |
| 2022/04/05 | 0.0.2 | 2.2-2.6 | 김종헌 |
| 2022/04/06 | 0.1.0 | 3.1 | 오세훈 |
| 2022/04/08 | 0.1.1 | 3.3 | 임세창 |
| 2022/04/09 | 0.2.0 | 3.4-3.6 | 김민현 |
| 2022/04/09 | 0.3.0 | 3.2 | 김영빈 |
| 2022/04/10 | 0.3.1 | 3.2 | 홍권 |
| 2022/04/11 | 0.4.0 | 수합 및 보완대상 검토 | 전원 |
| 2022/04/12 | 1.0.0 | 용어 통일 | 전원 |
| 2022/04/13 | 1.1.0 | 3.2 | 김영빈/홍권 |

| | | | |
|------------|-------|---------------|-------------|
| 2022/04/15 | 1.1.1 | 3.4, 3.6 | 김민현/오세훈/임세창 |
| 2022/04/16 | 1.1.2 | 1.1, 2.2, 4 | 김종헌/이미소 |
| 2022/04/18 | 1.2.0 | 문서 수합 및 서식 통일 | 김종헌/이미소 |
| 2022/04/30 | 2.0.0 | 오타자 점검, 어체 정리 | 전원 |
| 2022/05/01 | 2.0.1 | 최종 검토 및 수정 | 전원 |

표 34 Document History