

# Final Presentation

Smart Homepany : Customize your workplace

Team 1

김민현 | 김영빈 | 김종현  
오세훈 | 이미소 | 임세창 | 홍권

# INDEX

---

01

Introduction

02

Software Requirement Specification

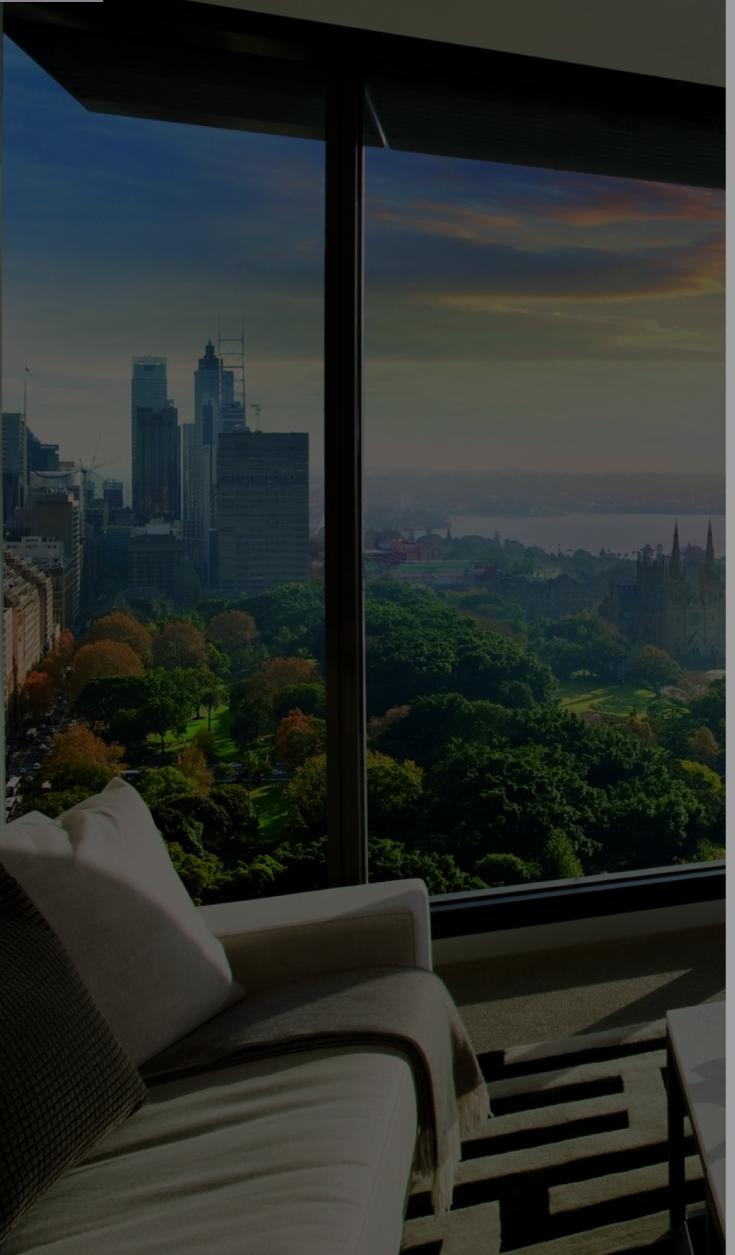
03

Software Design Specification

04

Effects

01



# Introduction

---

01

## Background



Telecommuters



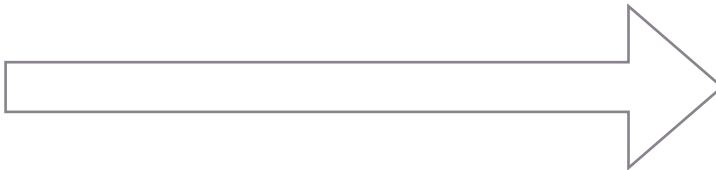
Creators



Freelancers

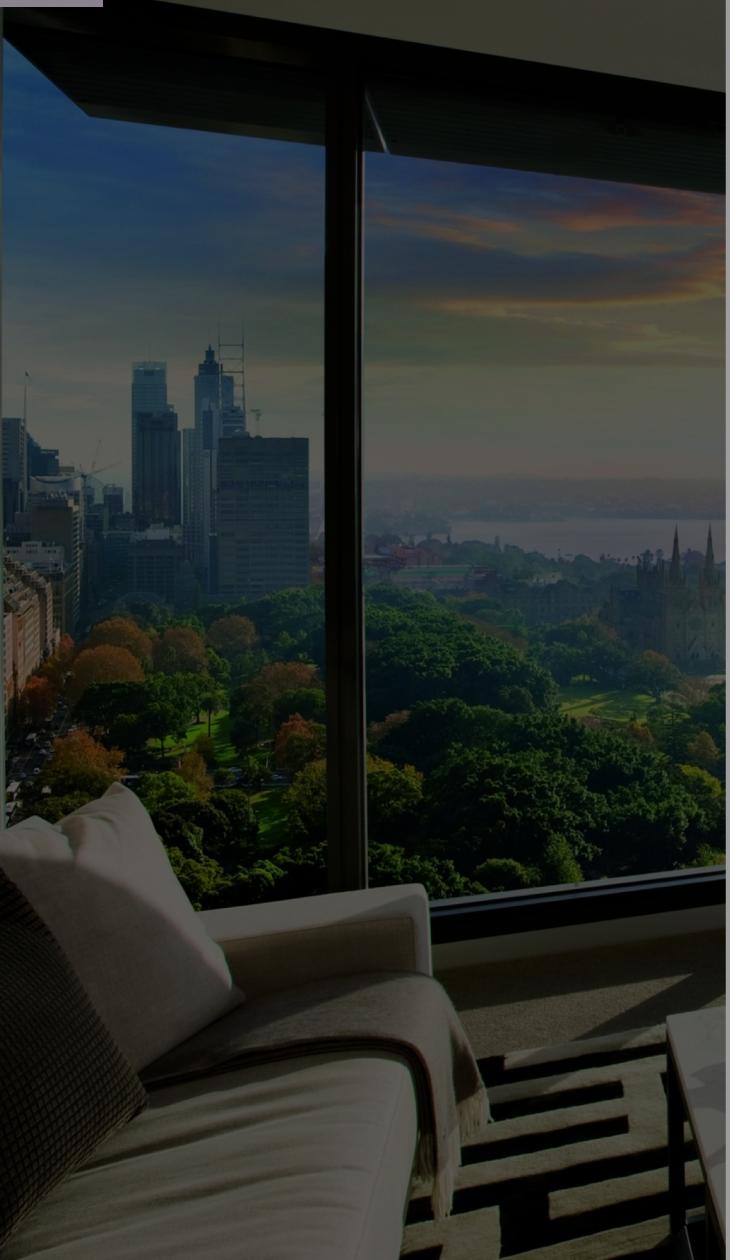


# Motivation



**NO Smart Home  
NO System**





# Software Requirement Specification

# Software Requirements Specification

Purpose & Contents

## Purpose

: Find out and document user requirements system requirements

## Contents

- 1 Introduction
- 2 Overall Description
- 3 Specific Requirements
- 4 Supporting Information

# Overall Description

Contents

- 1 Product Perspective
- 2 Product Functions
- 3 User Classes and Characteristics
- 4 Operating Environment
- 5 Design and Implementation Constraints
- 6 Assumption and Dependencies

# Overall Description

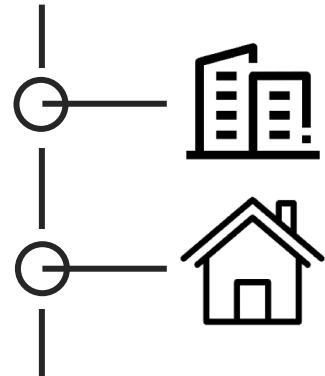
## Product Functions



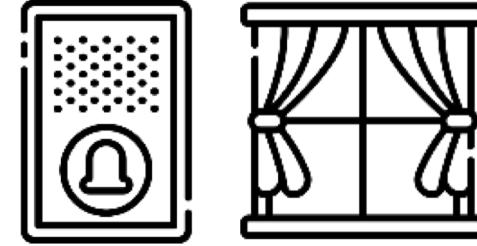
Search & connect to IoT device



Add, edit, remove presets



Reserve timelines



Customize settings of IoT device

# Specific Requirements

Contents

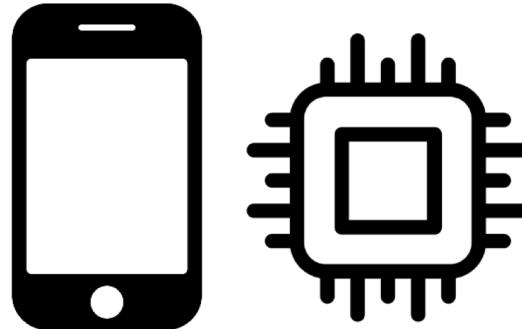
- 1 External Interface Requirements
- 2 Functional Requirements
- 3 Nonfunctional Requirements
- 4 Organizing the Specific Requirements
- 5 System Architecture
- 6 Assumption and Dependencies

# External Interface Requirements

Interfaces



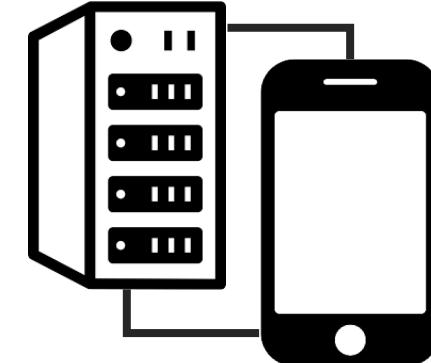
User Interfaces



Hardware Interfaces



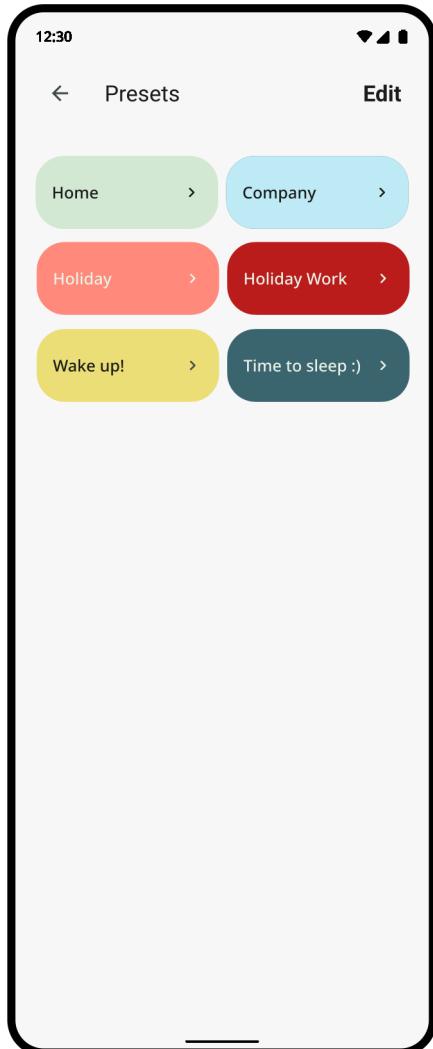
Software Interfaces



Communication Interfaces

# External Interface Requirements

User Interface Example

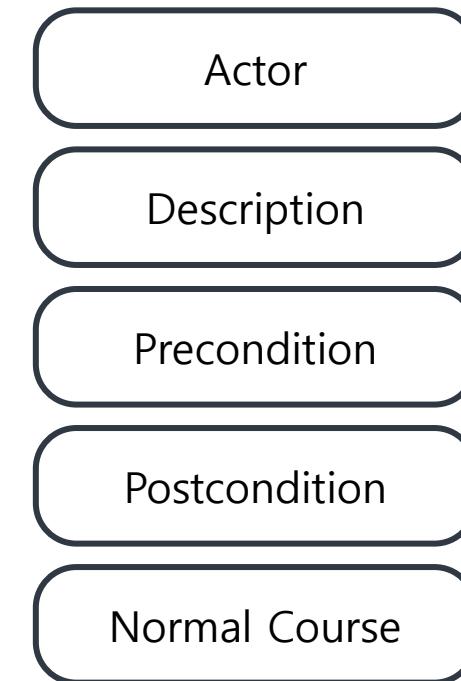


- ◆ Purpose/Description: It shows the list of basic presets and presets registered by the user.
- ◆ Input Source/Output Destination: Server/Client
- ◆ Range/Accuracy/Margin of Errors: N/A
- ◆ Unit: Screen
- ◆ Time/Velocity: N/A
- ◆ Relationship with other input/output: N/A
- ◆ Format/Configuration
  1. Layout shows the preset list.
  2. User can move detail page of each preset by clicking preset.
  3. User can move edit page by clicking edit button.
- ◆ Data Type: Text, Image, Widget, Button
- ◆ Instruction Type: N/A
- ◆ Exit Message: N/A

# Functional Requirements

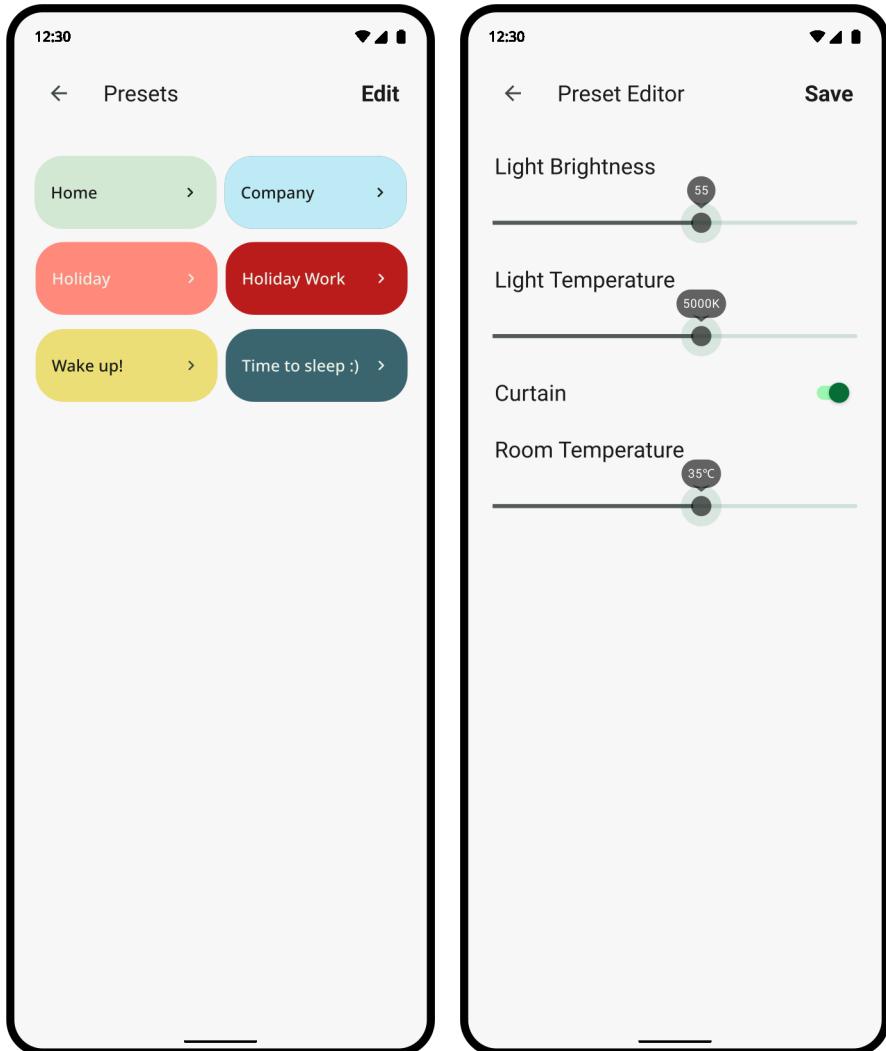
## Use Cases

- 01 Updating the latest firmwares of connected IoT devices
- 02 Updating the latest Smart Hompany software
- 03 Searching IoT devices
- ...
- 11 Editing preset reservation
- 12 Deleting preset reservation
- 13 Deactivating preset reservation



# Functional Requirements

Use Case Example: Adding Preset



- ◆ Actor: User, Connected IoT devices, DB
- ◆ Description: User adds new preset.
- ◆ Precondition
  1. Expected IoT devices should be connect to the same network as the mobile device.
  2. Expected IoT devices should be registered in the system.
  3. User clicks the presets button.
- ◆ Postcondition
  1. The list of presets is updated.
  2. New preset is added to Preset entity table in DB.
- ◆ Normal Course
  1. System shows the list of presets.
  2. User executes edit preset mode by clicking Edit button.
  3. User executes add preset mode by clicking Add button.
  - ...
  10. DB saves the information by adding new row to Preset entity table.

# Nonfunctional Requirements

Contents

## 1 Product Requirements

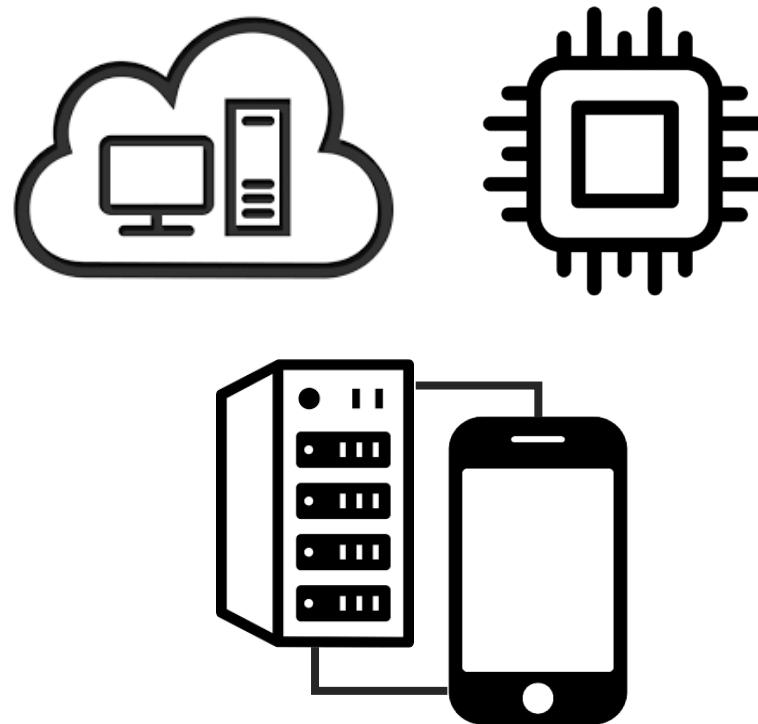
- Performance Requirements
- Space Requirements
- Security Requirements
- Usability Requirements
- Dependability Requirements

## 2 Organizational Requirements

- Environmental Requirements
- Operational Requirements
- Development Requirements

## 3 External Requirements

- Regulatory Requirements
- Ethical Requirements
- Accounting Requirements
- Safety/Security Requirements



# Nonfunctional Requirements

Examples

## EXAMPLE 1 Product Requirements

...  
*The software should transmit the signal from mobile device to each IoT devices under a second.*

...

## EXAMPLE 2 Organizational Requirements

...  
*To use the software, there should be at least one connectable IoT device in user's home.*

...

## EXAMPLE 3 External Requirements

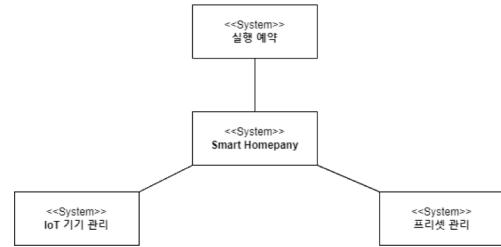
...  
*The software should not be used with malicious intention by connecting and controlling others' IoT devices.*

...

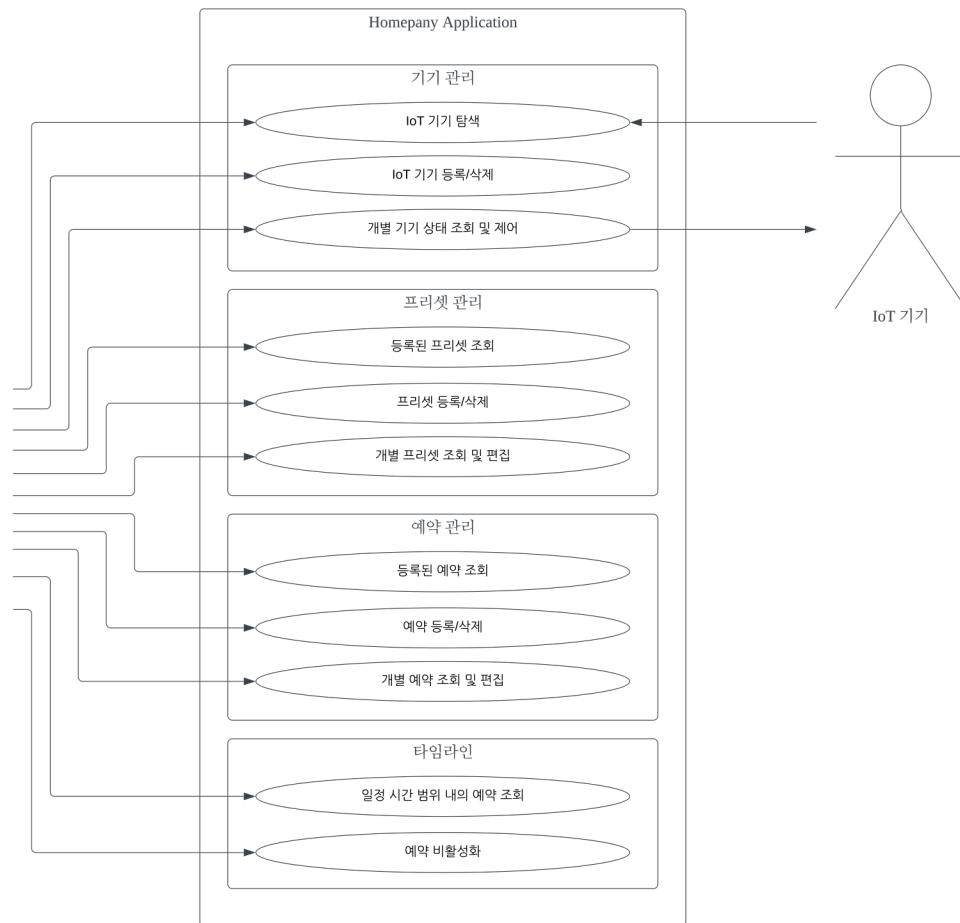
# Organizing the Specific Requirements

Models & Diagrams

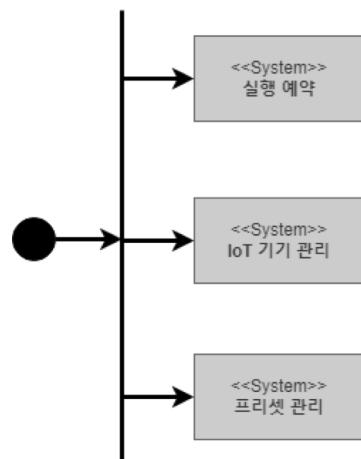
## 1 Context Model



## 3 Interaction Model



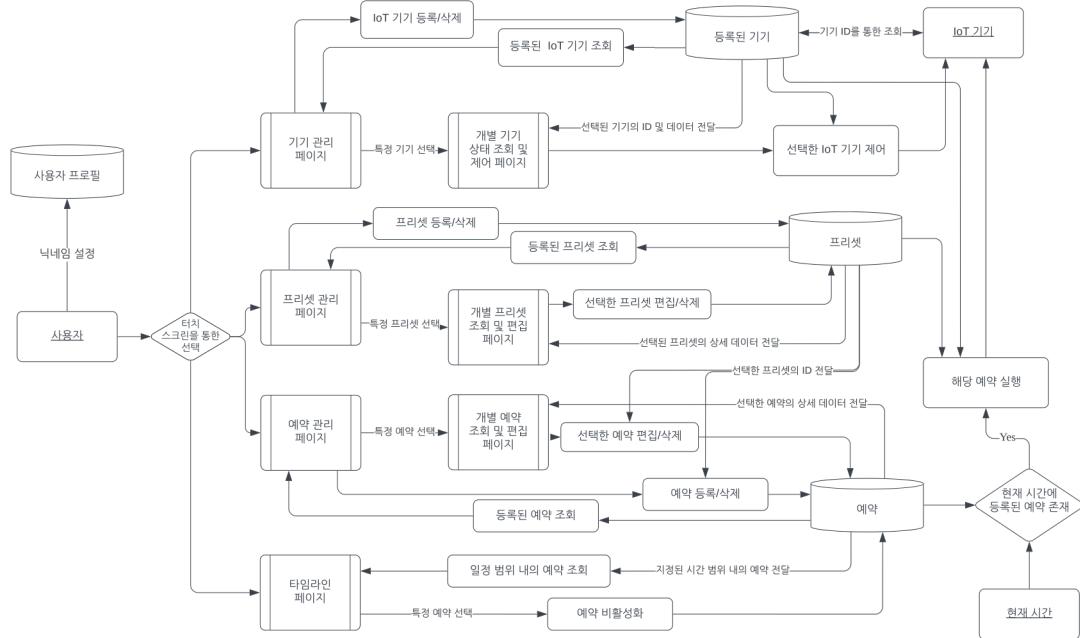
## 2 Process Model



# Organizing the Specific Requirements

## Models & Diagrams

## 4 Behavior Model

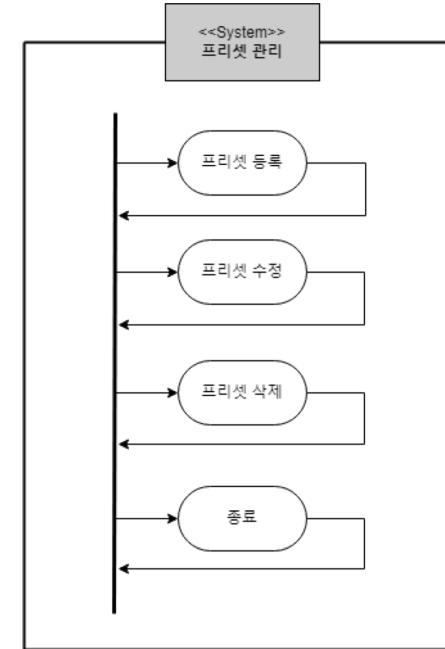
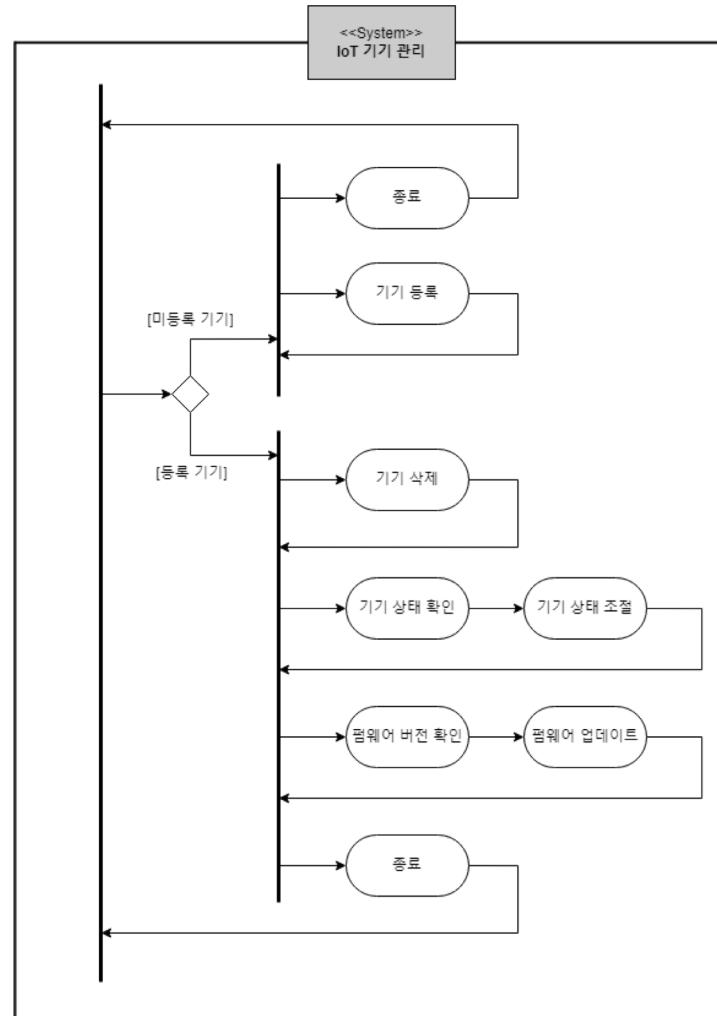
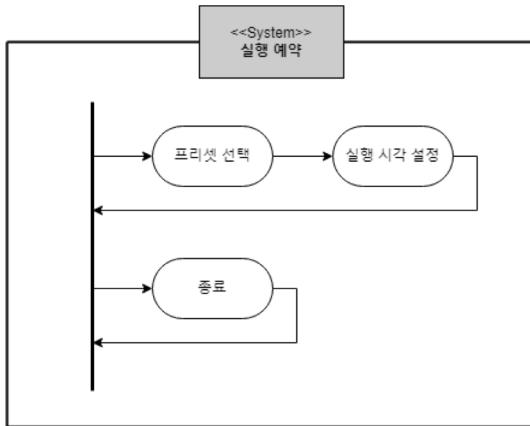


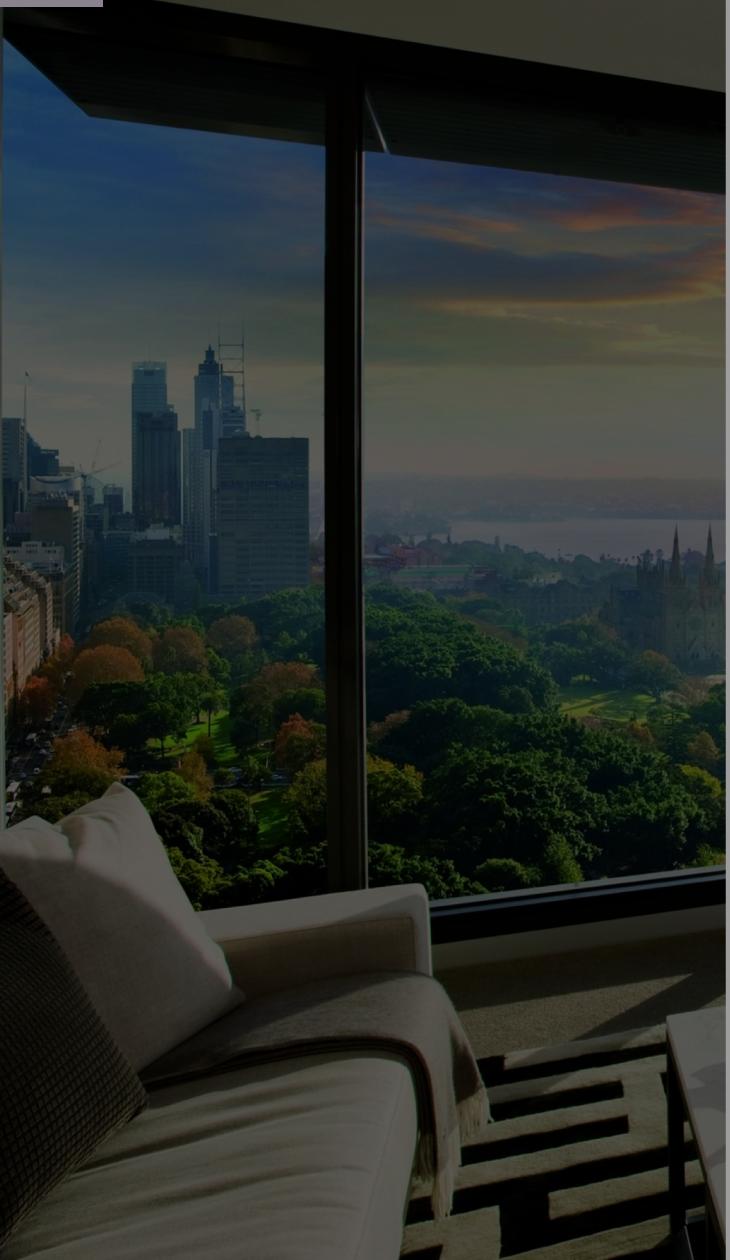
## 5 System Architecture

## 6 System Evolution

# Organizing the Specific Requirements

## Process Model





# Software Design Specification

# Software Design Specification

Purpose & Contents

## Purpose

: design a system organization which satisfy the functional and non-functional requirements of a system.

## Contents

01 Preface

02 Introduction

03 System Architecture - Overall

04 System Architecture – Frontend

05 System Architecture – Backend

06 Protocol Design

07 Database Design

08 Testing Plan

09 Development Plan

10 Supporting Information

# System Architecture: Frontend

Objective & Subcomponents

## Objectives

: explain components of the system architecture with class diagram and sequence diagram

## Subcomponents

Profile

MainPage

Manage  
Device

Device  
Detail

PresetPage

PresetDetail  
Page

NewPreset  
Page

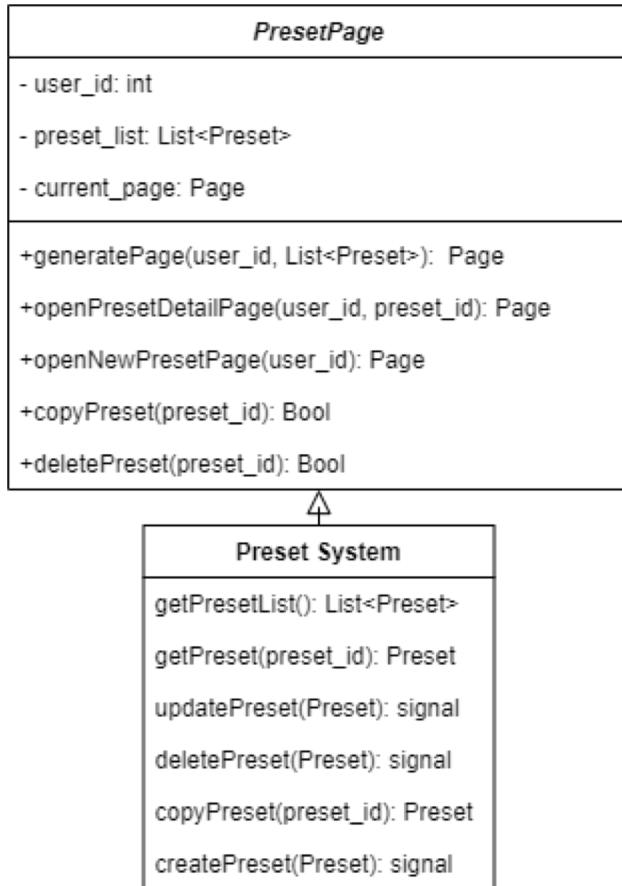
Timeline  
Page

Timeline  
DetailPage

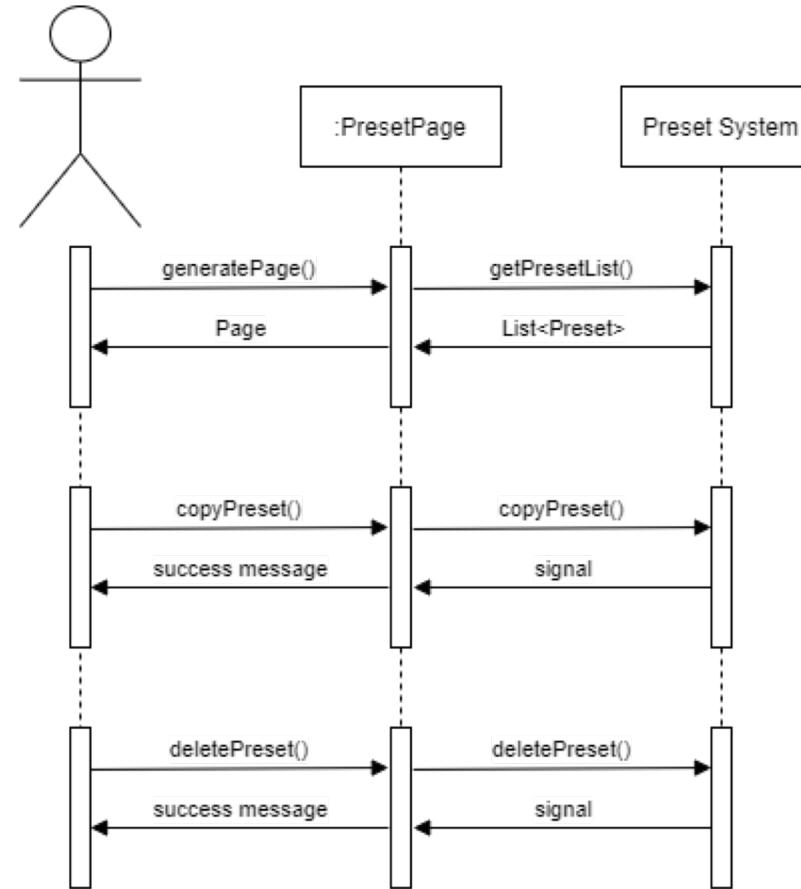
NewTimeline  
Page

# System Architecture: Frontend

Subcomponent Example: PresetPage



Class Diagram



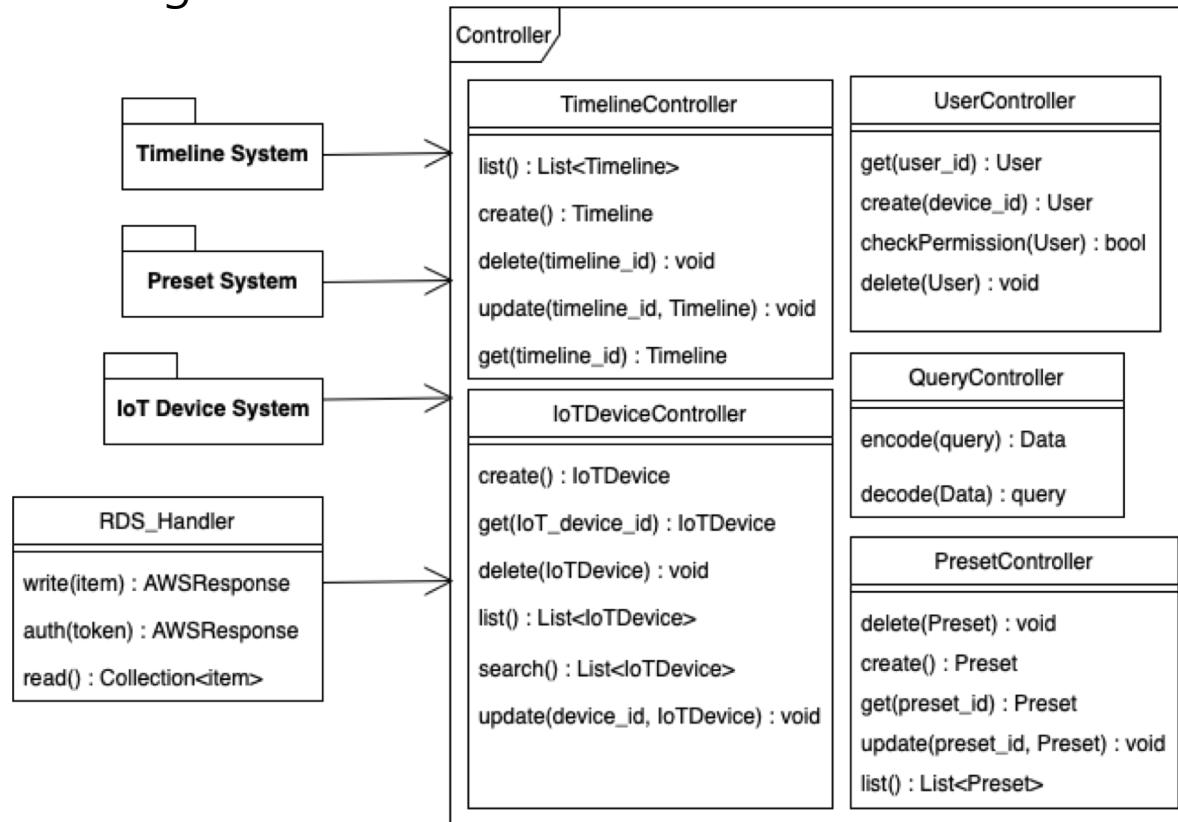
Sequence Diagram

# System Architecture: Backend

Objective & Overall Architecture

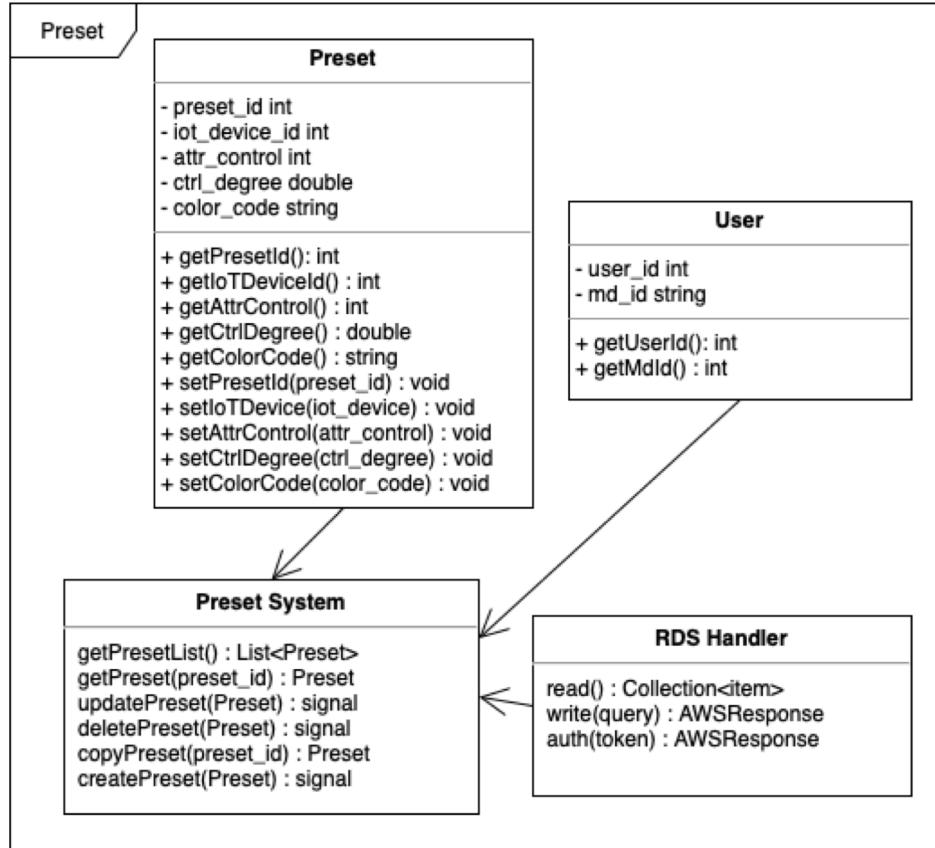
## Objectives

: explain the overall backend system and components of the system architecture with class diagram and sequence diagram

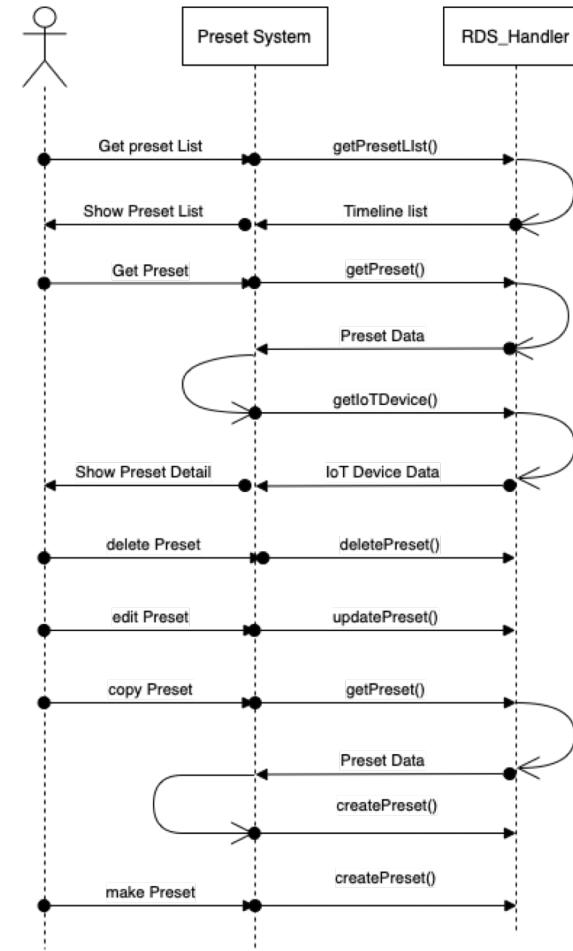


# System Architecture: Backend

Subcomponent Example: Preset System



Class Diagram



Sequence Diagram

# Protocol Design

Objectives & Contents

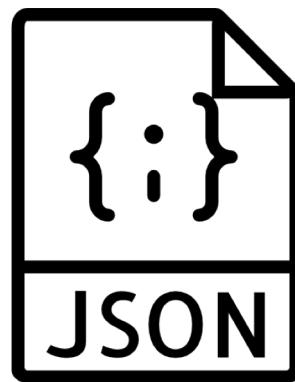
## Objectives

: explain how server, DB, and software communicate with each other

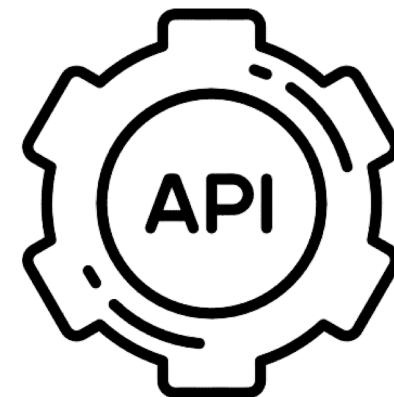
## Contents



Transfer Protocol



Data Type



REST

# Protocol Design

Example: Load Preset List

- ◆ Function: Load the preset list of user
- ◆ URL: {{url}}/preset/list

◆ METHOD: GET

◆ Parameter: user\_id(string)

◆ Success Return

    resultCode: 200(HTTP status code)

    PresetList: preset list of user(list)

◆ Failed Return

    resultCode: 400(HTTP status code)

◆ Type of Error

    Failed user authentication – resultCode: 400

```
{  
    "resultCode": 200,  
    "PresetList": [  
        {  
            "preset_id": 1,  
            "iot_id": "E06C3823FC827F63C3",  
            "attr_control": 2,  
            "ctrl_degree": 3.0,  
            "color_code": "0xfffffff"  
        },  
        {  
            "preset_id": 2,  
            "iot_id": "E06C3823FC827F63E3",  
            "attr_control": 33,  
            "ctrl_degree": 133.0,  
            "color_code": "0xfecd2f"  
        }  
    ]  
}
```

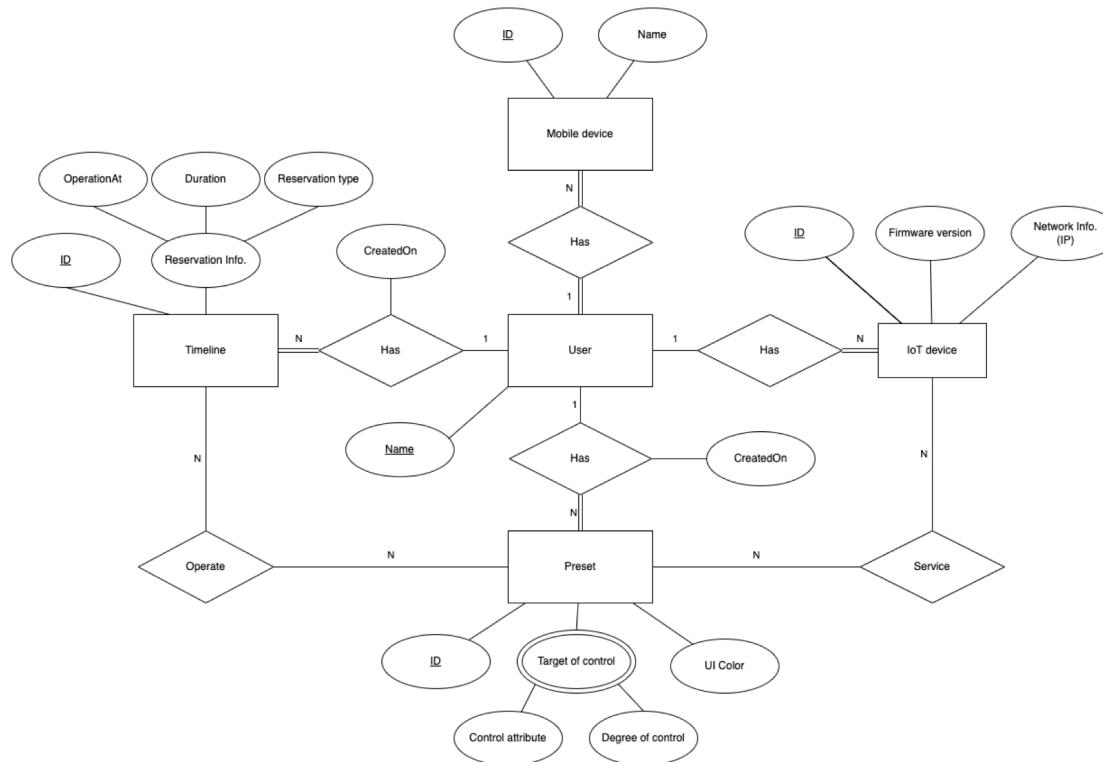
Example of JSON when return successfully

# Database Design

## Objectives & ER Diagram

## Objectives

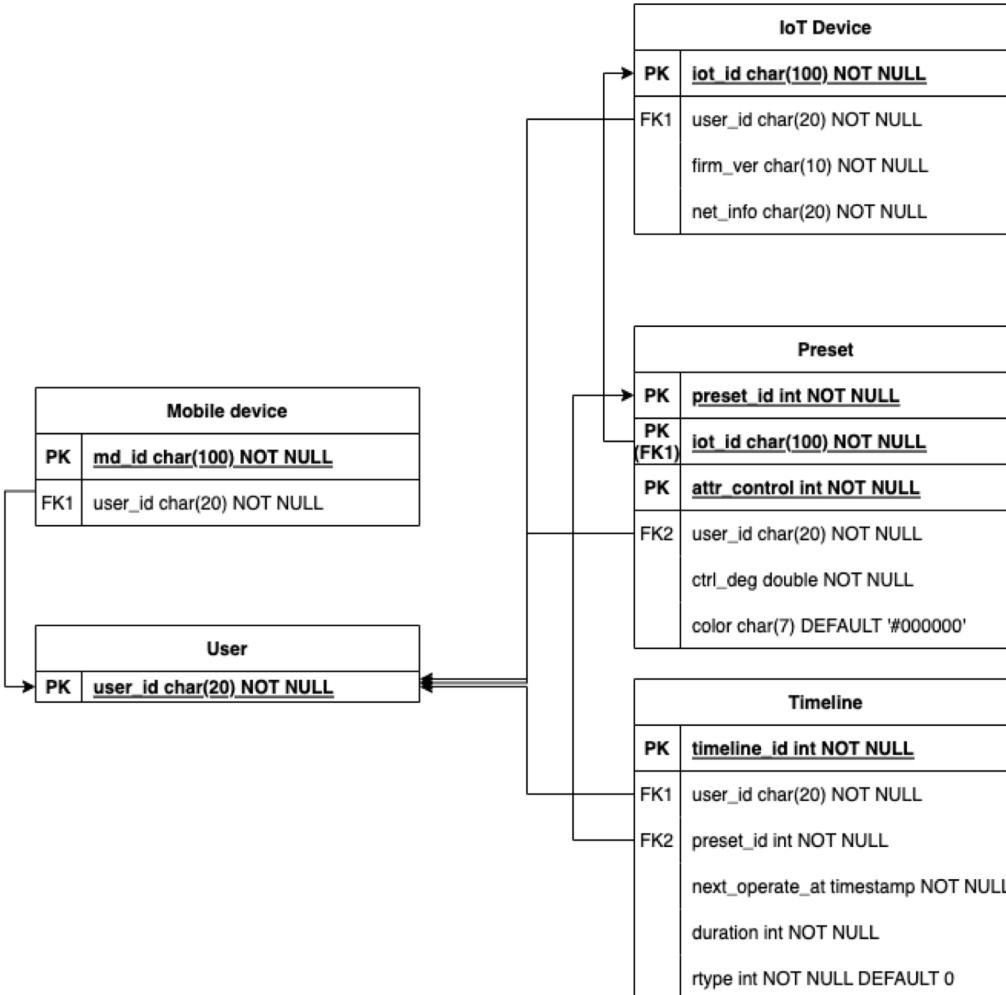
: explain components of the system architecture with class diagram and sequence diagram



ER Diagram

# Database Design

Relational Schema & SQL DDL Example



Relational Schema

```

CREATE TABLE `User` (
    `user_id` CHAR(20) NOT NULL,
    PRIMARY KEY (`user_id`)
);
    
```

```

CREATE TABLE `Preset` (
    `preset_id` INT NOT NULL AUTO_INCREMENT,
    `iot_id` CHAR(20) NOT NULL,
    `attr_control` INT NOT NULL,
    `user_id` CHAR(20) NOT NULL,
    `ctrl_deg` DOUBLE NOT NULL,
    `color` CHAR(7) DEFAULT '#000000',
    PRIMARY KEY (`preset_id`, `iot_id`,
    `attr_control`),
    FOREIGN KEY (`iot_id`) REFERENCES
    Iot_device(iot_id),
    FOREIGN KEY (`user_id`) REFERENCES
    User(user_id)
);
    
```

SQL DDL

### Objectives

: explain the plan of implementation, distribution and user test

### Contents

01 Development Test

02 Release Test

03 User Test

04 Test Case

# Development Plan

## Environments

Front-end



Back-end

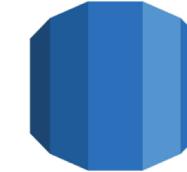
Server



Amazon EC2



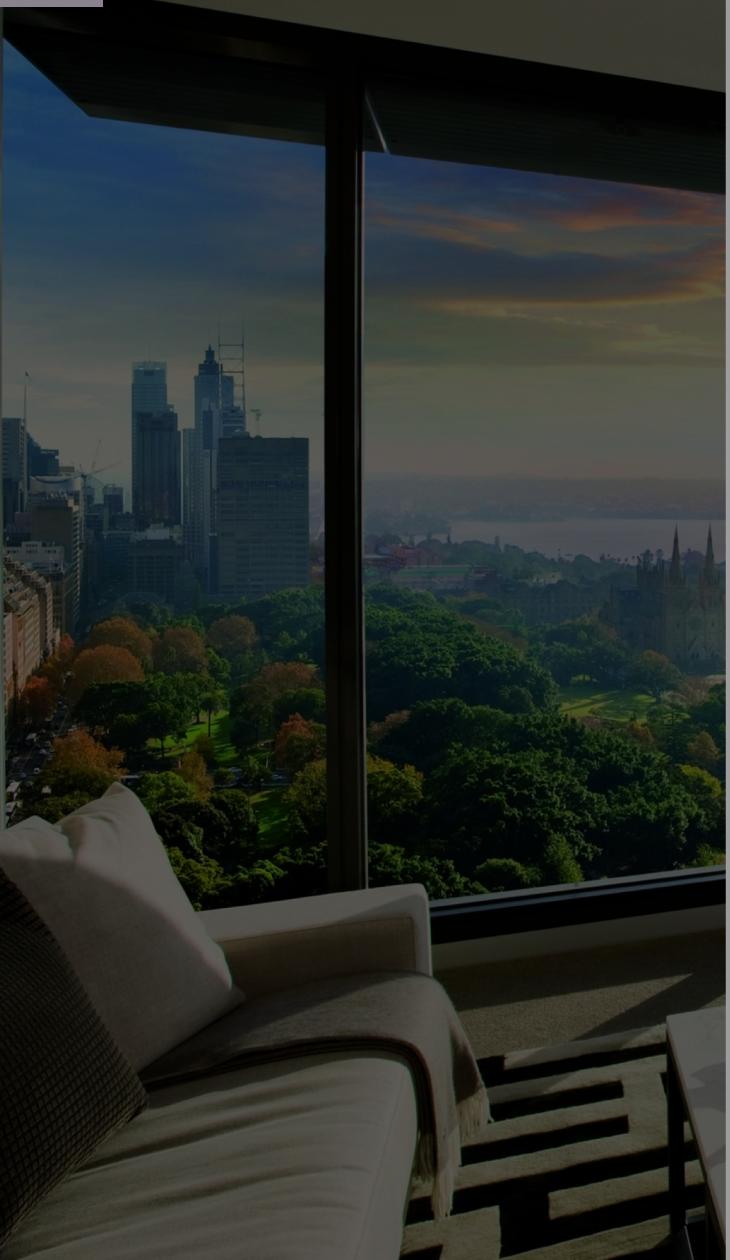
Database



Amazon RDS



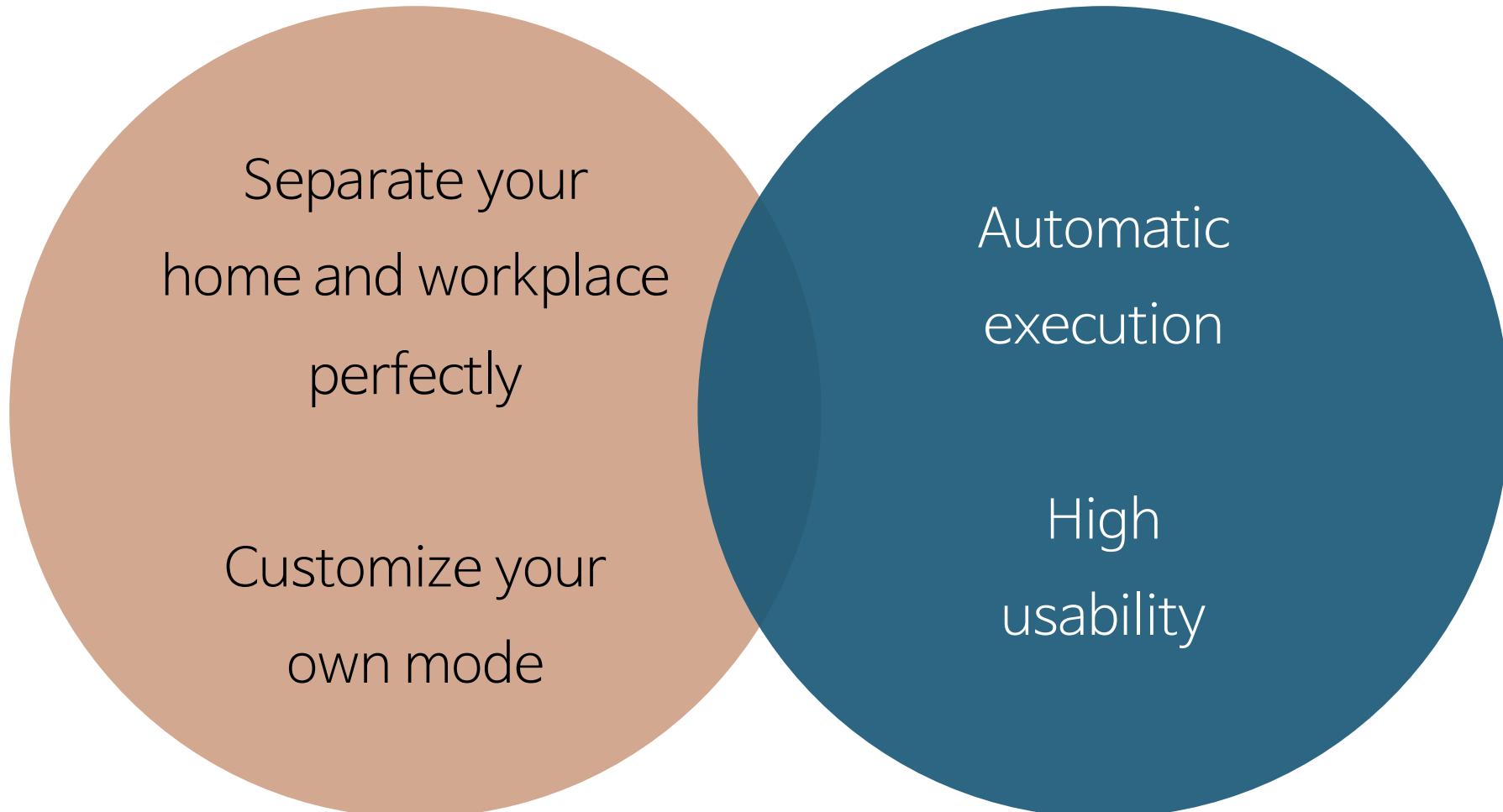
SQLite

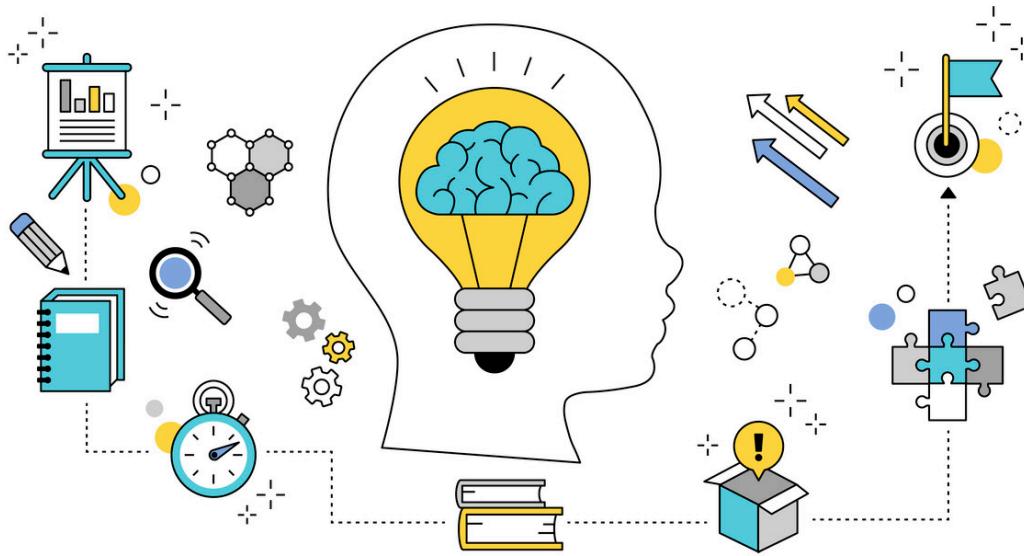


## EFFECTS

---

### Mode Change





Relieving compulsion to stay on-line all the time

Increasing work efficiency and concentration

Improving quality of life

---

THANK YOU