# Smart Chandelier

Color every moment

- Software Requirements Specification (SRS) -


TEAM 1

| | |
|---|---|
| | Gyeongsu Kim |
| | Daeun Kim |
| | Hyuckjoong Yoon |
| | Sana Kang |

# Table Of Contents


**1. Introduction** **3**

**2. Overall Description** **3**

A. Product Perspective 3

B. Product Features 5

C. Operating Environment 5

D. Assumption and Dependencies 6

**3. System Features** **7**

A. Functional Requirements 7

B. Use case Diagram & Description 10

Account Management 10

Module Information Management 11

Event Management 12

Module Control 14

Manual Mode Management 15

Auto Mode Management 16

C. Sequence Diagrams 18

Login with existing account 18

Register new module 18

Add new event 19

Control power on/off 19

Set lighting pattern 20

Get lighting result 20

**4. Preliminary User Manual** **21**

**5. Non-functional Requirements** **23**

**6. Acknowledgements** **25**

# 1. Introduction

The “Smart Chandelier” is a smart IoT system that serves a special mood to the users through the light. The product consists of Smart Chandelier lighting modules, supplementary sensor modules, optional hardware, and a central hub.

The lighting module provides powerful lighting service that covers the entire indoor area. The Smart Chandelier replaces the main lighting in the living room, so it provides various functions including simple lighting, colorful effects and lighting supervision. The target users include general households, gym owners, cafe owners, and etc.

This product is expected to be used as a way to replace or upgrade the main lighting in customers' home automation as a high-quality light. It’s the reason that this product pursued a convenience while incorporating necessary functions for actual use as much as possible. It provides adequate lighting without the user’s care, and can also be customized if needed.

There are a few functional requirements introduced, that helps you to understand the main functions and the concept of this product.

**(i) The user is provided with lightning pattern ranking information in order of ratings in the Smart Chandelier user community.**

The product’s lighting module provides a complex lighting service to offer a special mood to the users, and the users will be able to travel the community looking for appropriate lighting patterns. So Smart Chandelier recommends popular lighting patterns from the community, so that users can utilize the lighting which they like.

**(ii) The user can set the “event”.**

One of the purposes of this product is to help users with home automation. According to the concept of smart lighting, our Smart Chandelier should offer a lighting pattern in a specific condition according to the user’s need. We call this function an event. The user could set the triggered situation, and the user could choose which lighting pattern will be used for that situation. For example, users can use the automated Smart Chandelier by turning on the lights brightly when they enter the house, or by turning on preset lights on special days.

**(iii) The user could use additional hardwares such as smart remote control since Smart Chandelier supports compatibility with SmartThings system.**

Smart Chandelier supports compatibility with SmartThings system, which is a powerful existing system, in order to provide scalability to those who have already used their own IoT ecosystem. It is one of the biggest advantages of being able to manage and interact with existing hubs or smart devices by connecting them.

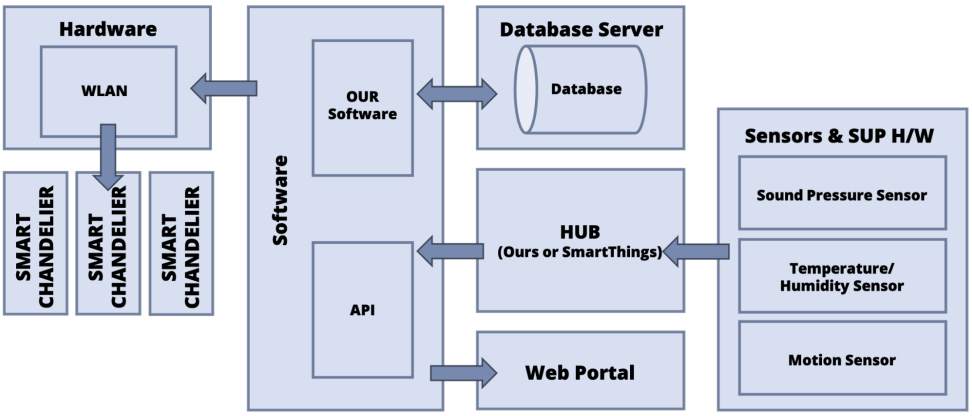
# 2. Overall Description

## A. Product Perspective

The "Smart Chandelier" is a follow-on member of the product family. The product line is as follows: Smart Chandeliers light modules, light covers of various designs, supplementary sensor modules, and optional hardware, and SmartThings hub to link them all. Likewise, the Smart Chandelier will be available in a variety of designs, including a cylindrical and a low rectangular form, and customers will be able to buy and install them as many as they need. Extra hardware, such as a remote control, and three types of supplementary sensors - a

temperature/humidity sensor, a motion sensor, and a sound pressure sensor - would make it easier for customers to operate the Smart Chandelier - or maybe create the "Smart Chandelier ecosystem."

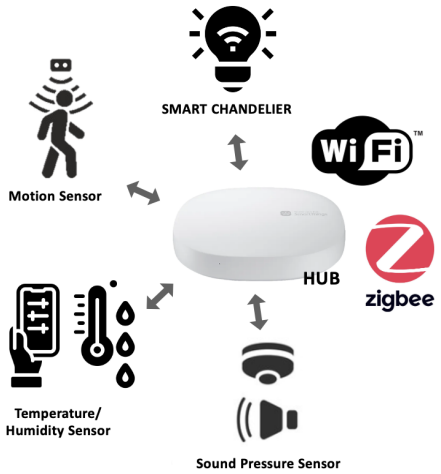
We will make use of three sensor modules as well as external APIs. Firstly, the system can use a motion sensor to gather quantitative numerical information on each user's specific location and movement. Secondly, it can also utilize the sound pressure sensor to measure the decibels of music or noises (e.g., the ambient environmental noise and the living noise). Finally, the temperature/ humidity sensor will gather accurate temperature and humidity information on the particular location of users, rather than collecting general weather information. If users do not utilize such sensor modules, they can alternatively use external APIs to obtain relevant information.




[Fig 1] A block diagram of the “Smart Chandelier”

As previously said, we would connect smart chandelier-related devices through the SmartThings hub platform. This allows users to build IoT by connecting other companies' AI speakers or household appliances. The hub connection needs the input of a sensor's unique serial number, and sensors connected to the hub can be linked to a variety of devices alternately. Furthermore, multiple users can register each "smart chandelier" by its unique serial number. These devices and sensors interact with each other through Zigbee, a high-level communication protocol based on IEEE 802.15.4-2003. We chose Zigbee communication because it is well-suited for large-scale building automation and home networking, thanks to its low power and low cost. Internet-related hardware will be the system's hardware interface, designed to function with 2.4 GHz Wi-Fi.

Moreover, we will select MySQL from the SQL DBs to build the database using the online backup approach. MySQL is an open-source RDBMS (relational database management system) that is relatively inadequate for managing large amounts of data but is affordable and performs well enough to provide essential services. Sensors, internet portals, and applications collect user data for storage in a database. Considering this project requires a few features for the web portal - community, ranking inquiry, and log in - the interface between mobile app and database will be essential rather than the interface between web portal and database.




[Fig 2] The interaction between the smart chandelier, sensors, and the hub.

## B. Product Features

Firstly, the "Smart Chandelier" provides users with life-enriching experiences based on numerous lighting patterns appropriate for their lives. The user can directly select the lighting pattern in manual mode or accept recommendations in auto mode following an automated evaluation of the surrounding environs. Users may improve their concentration on their jobs or make their experiences with friends and family more memorable. When users read books, for example, they can focus more on the content of the book by utilizing one of two modes: the "reading mode" (the pattern of subtle light with a high color temperature) or the "auto mode" (the pattern based on surroundings conditions)

Secondly, users may share their favorite customized or recommended lights with the community. Additionally, users can receive pattern ranking information based on quantitative evaluations and even download patterns posted by other users. This service will allow users to obtain a broader range of user experiences.

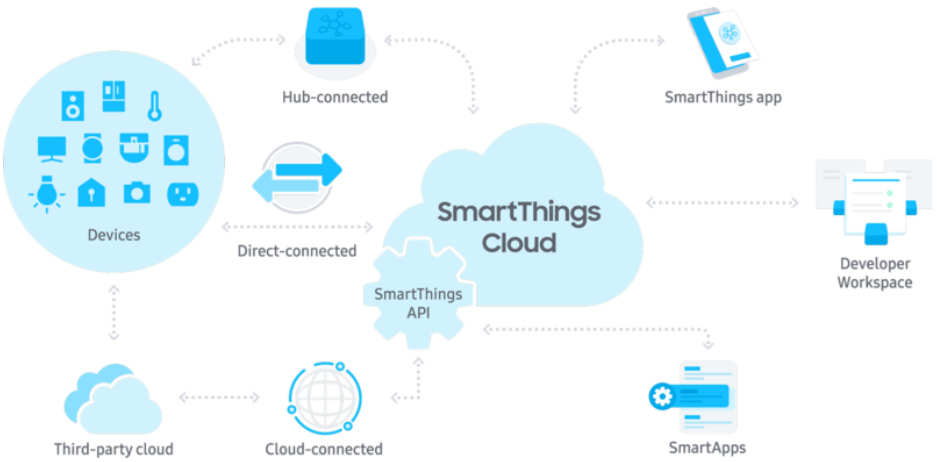
Thirdly, the "smart chandelier" service also provides an analysis of users' behavior. Health data on a user's sleep time, for example, can be calculated by measuring the activation time of "sleep mode" and "wake-up mode." It also keeps track of users' activities, such as an exercise or home party, and displays the statistics in a monthly calendar style.

Lastly, the user may utilize extra functionalities for convenience. For example, there is a "Warning" feature that activates the red light when it detects a baby screaming or falling out of bed.

## C. Operating Environment

One of the main modules, the smart chandelier lighting module is controlled through an embedded system. The lighting devices will be connected to one console, which this project will use Raspberry Pi 4.

Raspberry Pi 4 will be used as a main operating system of the smart chandelier lighting module, and only a single web program will be processed there. Since it provides recent linux kernels, the developers can easily adapt to the environment. It won’t require a lot of tasks for them to test the softwares to be developed.




[Fig 3] The SmartThings diagram.

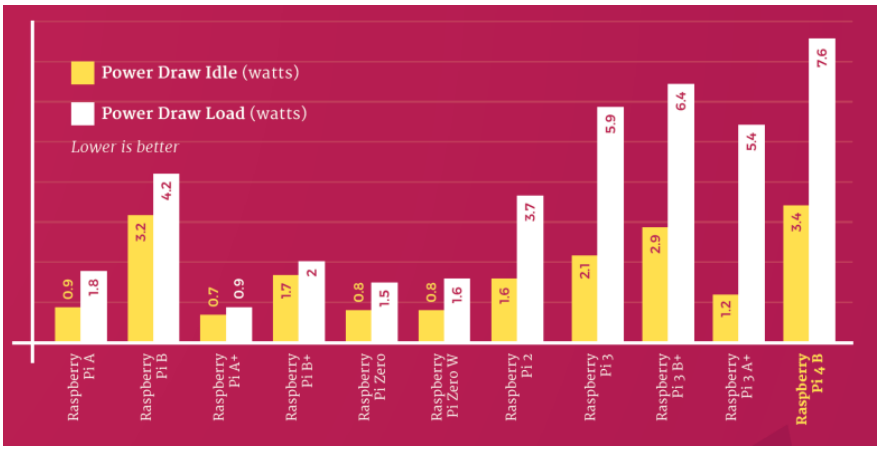
As the Smart Chandelier system uses a central hub to communicate with users, the lighting module's main job is also to communicate with the hub. It can be used with an existing hub platform, SmartThings, so the software of the module will follow the SmartThings API policy. And it also provides a strong protection using OAuth 2.0 Bearer Tokens, which is powerful enough to use for home appliances.

Because the SmartThings API is based on the RESTful API, actual main tasks of the lighting module and sensors will be HTTP requests. Raspberry Pi 4, as the OS of the lighting module, is expected to provide strong functions to do such jobs.

## D. Assumption and Dependencies

### (1) Raspberry Pi console's power amount.

Raspberry Pi 4 uses very little power. It results in using a 15W power adapter, even though this machine is frequently used to control electric hardware devices. However, the Smart Chandelier must provide powerful lights with many LEDs and the console's power is too low.




[Fig 4] A power draw benchmark of Raspberry Pi 4.

Developing a prototype, it's not a big problem. However for the actual build, it's inevitable to use another power source for the lighting module.

### (2) SmartThings policy

As the SmartThings documentation doesn't explicitly care about the version, we should assume that the policy would not change rapidly. Because it is reasonable to expect stable support for previous versions, it is supposed to be a small problem.

### (3) Smart sensor modules with SmartThings system

We expect to coexist with existing smart home appliances, especially SmartThings system, it's assumed that the user should use this lighting module with existing smart sensors which can be a part of SmartThings system.

However, as SmartThings produces selfmade smart sensors, hubs, and even home appliances, we don't expect a risky situation about this assumption.

### (4) Legal factors rapidly changing

Recently, along with the rapid development of IoT technology, many IoT products have appeared, attracting interest in related markets and laws. In the case of domestic, as related laws are being enacted very recently, unexpected new laws may be enacted.

This project assumes that the current situation, laws, and the general mood and trend of society will not change significantly. But clearly, this issue must be managed as a risk.

## 3. System Features

### A. Functional Requirements

| Hierarchy | Short Description | Requirement ID | Requirement Description | Actors | Priority |
|---|---|---|---|---|---|
| **Account Management** | User Account generation | A-1 | Users have to create a user account to use the Smart Chandelier service. Only one account can be created per user through phone/SNS authentication. The user account database system manages this data. | User, System | Critical |
| | User Registration to the hub | A-2 | The user must register a user account that has already been generated to the hub, existing in the living space. | User, System | Critical |
| | User login to the hub | A-3 | Once the user account has been registered to the hub, the user can log in with an easy way such as touchID/FaceId/PIN. After log-in, users could manipulate all modules connected to the hub, and could use the Smart Chandelier service offered by lighting modules in that hub. | User, System | Critical |
| | User account deletion from the hub | A-4 | After log-in to the hub, the user could delete the user account from the hub. | User, System | Critical |
| | User information change | A-5 | Users could change user information such as E-mail, nickname, phone number freely. | User, System | Minor |
| | User password change | A-6 | User could change his or her password to authenticate the user. | User, System | Minor |

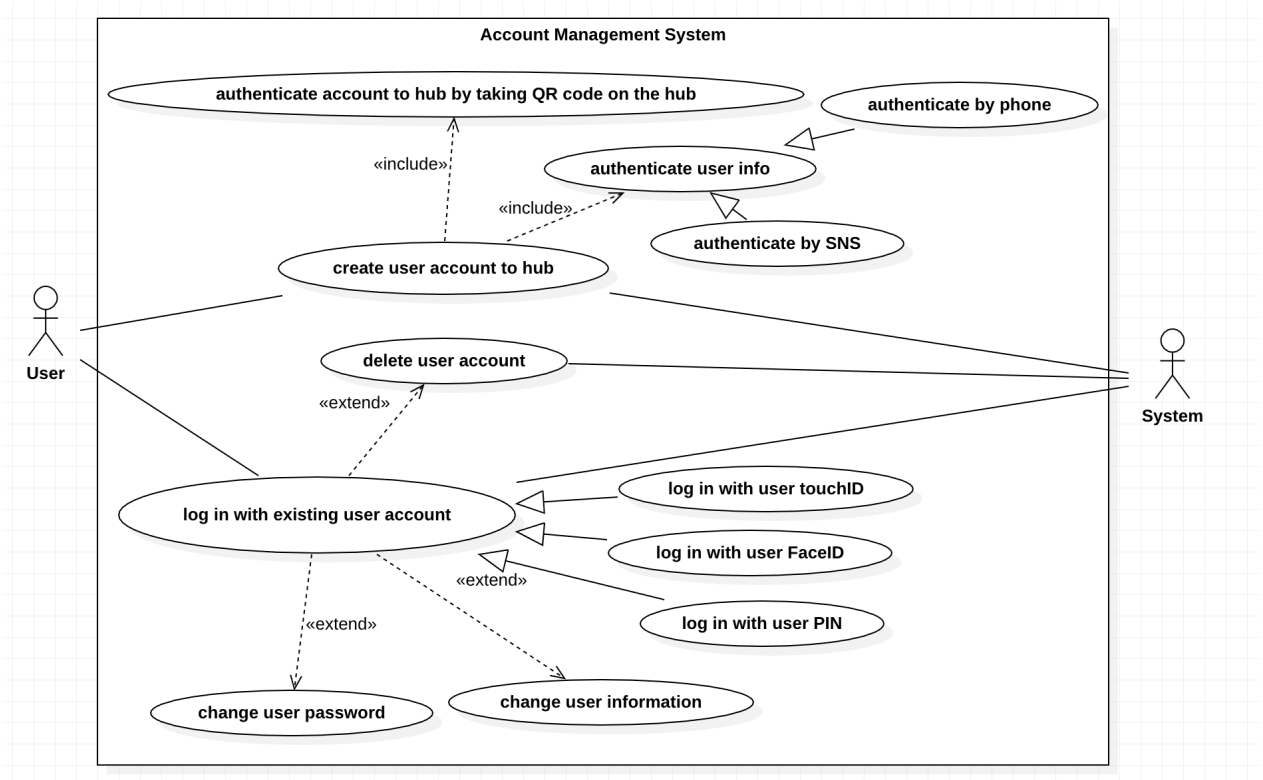
| | | | | | |
|---|---|---|---|---|---|
| **Module Information Management** | Module registration to the hub | B-1 | Users could register a new module to the hub, using the identifier of the module. There are 2 types of modules. One is the lighting module, which offers lighting service to the user. And the other one is a sensor module, which could collect the data to offer the lighting service. | User, System | Critical |
| | Module deletion from the hub | B-2 | Users could delete the registered module from the hub. | User, System | Critical |
| | Module information change | B-3 | Users could change the module information registered to the hub, such as its nickname. | User, System | Minor |
| | Module history reading | B-4 | Users could freely check the module history. For example, for the lighting module, the user could check the lighting data, and what "event"(It will be explained later.) is activated. And for the sensor module, users could get the collected data. Users freely refer to that data to set the current lighting pattern in the manual mode, or to set the event. To help read, the system offered a caliender that summarized those data by day or month. | User, System | Critical |
| | Module history deletion | B-5 | Users could freely delete the module history data. | User, System | Minor |
| **Event Management** | Adding event to the hub | C-1 | To help users with home automation, it would be useful to perform a specific lighting pattern in a specific condition according to the user's need. We call this function an event. To make an event, the user should set 1) event's name, 2) event's triggered situation, 3)event's lighting pattern. The user could set the triggered situation explicitly or implicitly, and the user chooses which lighting pattern will be used for that triggered situation. | User, System | Critical |
| | Editing(activate/ deactivate) existing event | C-2 | Users could freely change the components of the event, 1) event's name, 2) event's triggered situation, 3)event's lighting pattern. And there is another important function. Users could activate/deactivate this event for the selected lighting module. For one lighting module, there exists only one activated event. This activated event is used for "auto mode"(It will be explained later). | User, System | Critical |
| | Deleting event from the hub | C-3 | User could delete the registered event from the hub | User, System | Critical |
| **Module Control** | Module power on/off control | D-1 | Users could control the power of modules which are connected to the hub. Not only just controlling the power on/off status, there are additional functions. First, the user could set the timer for the | User, System | Critical |

| | | | | | |
|---|---|---|---|---|---|
| | | | duration of specified power status. Second, if the module is the sensor, the user could calibrate the sensor. To be more specific, to help the user automation, we could adjust the sensor module's current threshold. | | |
| | Selection between auto/manual mode | D-2 | If the module is a lighting module, the user could run that module via auto mode or manual mode. At the auto mode, the system automatically detects the change, and calculates the best light pattern and offers that to the user. In manual mode, the user could change the lighting pattern as he or she wishes. Details will be explained later. | User, System | Critical |
| | Checking the module state | D-3 | Users could check the module state. To be more specific, users could check who is using. And user could check the current collected sensor data. | User, System | Minor |
| **Manual mode management** | Setting the lighting pattern manually | E-1 | In manual mode, the user can change the lighting pattern as he or she wishes. System shows the various options to change the light pattern. 1) Set color, 2) Set timer(duration time), 3) Select pre-downloaded pattern from User community, and 4)Set advanced control such as gradation effect. | User, System | Critical |
| **Auto mode management** | Getting the lighting pattern automatically | F-1 | At the auto mode, the system automatically offers the lighting result to the lighting module. When the system detects the change, this action is triggered. There are two cases, 1) Activated events of this module are changed, and 2) If the amount of change in sensor data exceeds a certain threshold. When the change is detected, the system calculates the best light pattern by using the factors below. 1) Use sensor data, 2)Use activated events, 3)(optional) if there is insufficient data, it uses external data on the internet, such as weather information. | User, System | Critical |
| | Make feedback to the system | F-2 | When the auto mode is terminated, the user could give feedback to the system. Not only this situation, whenever the user wants feedback, the user could make detailed feedback to the system. | User, System | Minor |
| **Extra Function** | Additional HW registration to the hub | G-1 | Not only lighting/sensor modules but also additional HWs such as smart remote control since Smart Chandelier supports compatibility with SmartThings System. | User, System, | Critical |
| | Uploading lighting pattern to the community | G-2 | Users could upload its lighting pattern to the user community. | User, System, Smart Chandelier User | Minor |

| | | | | | |
|---|---|---|---|---|---|
| | | | | community | |
| | Downloading lighting pattern from the community | G-3 | Users could download its lighting pattern from the user community. To help users, the system offers the rating system(from0 to 5), so the user could refer to it. | User, System, Smart Chandelier User community | Minor |

## B. Use case Diagram & Description

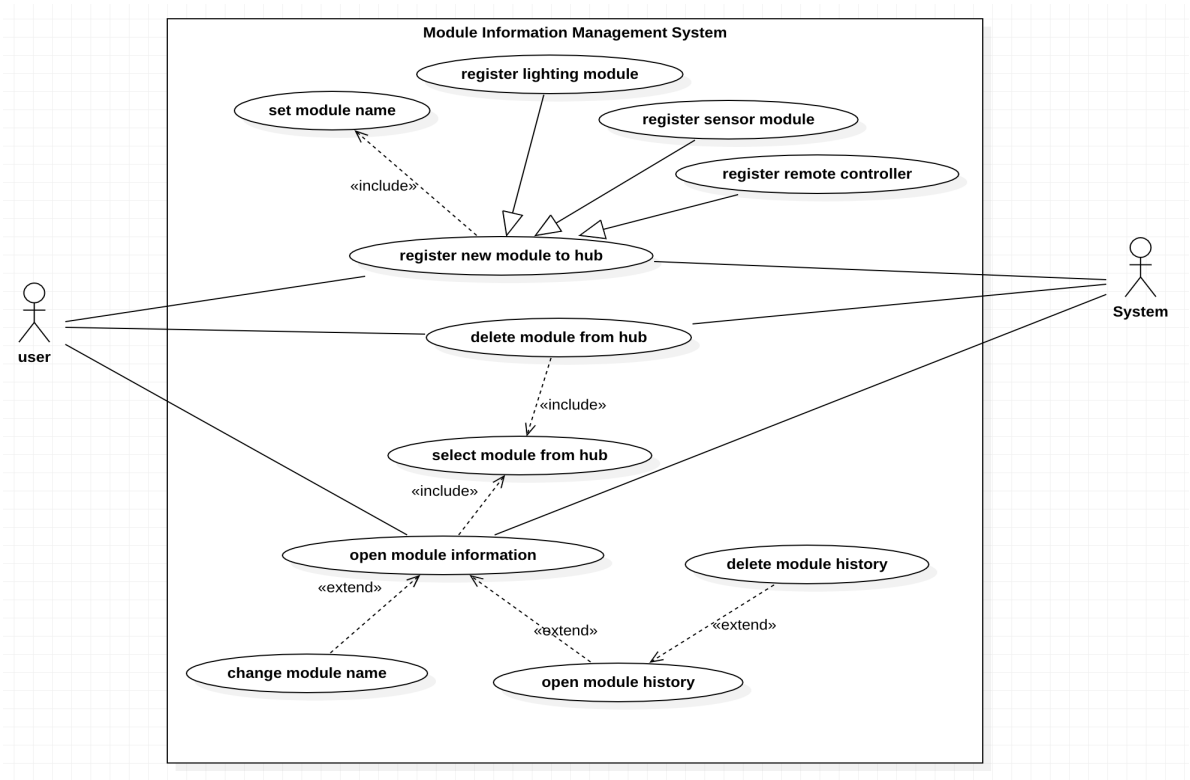
### (1) Account Management




| Use case name | Login with existing account | |
|---|---|---|
| Related Requirements | A-3 | |
| Goal in Context | User logs in with an existing user account to use the service of Smart Chandelier. | |
| Preconditions | The user account for the hub has already been created by the System. | |
| Successful End Condition | User successfully logged in to user account. | |
| Failed End Condition | User's attempt to log in is rejected. | |
| Primary Actors | User | |
| Secondary Actors | System | |
| Trigger | User asks the system to log in via Phone authentication API | |
| Main Flow | Step | Action |

| | | |
|---|---|---|
| | 1 | User asks the system to log in via Phone authentication API.<br>- Login with touch ID<br>- Or, login with FaceID<br>- Or, login with PIN |
| | 2 | User's login attempt is verified using the System's account database. |
| | 3 | User log-in with existing user account. |
| | 4 | Optionally, the user could modify or delete the user account.<br>- Change password<br>- Or, change user information<br>- Or, delete user account |
| Extensions | Step | Action |
| | 2.1 | System does not verify the user's login attempt. |
| | 2.2 | User's login attempt is rejected. |

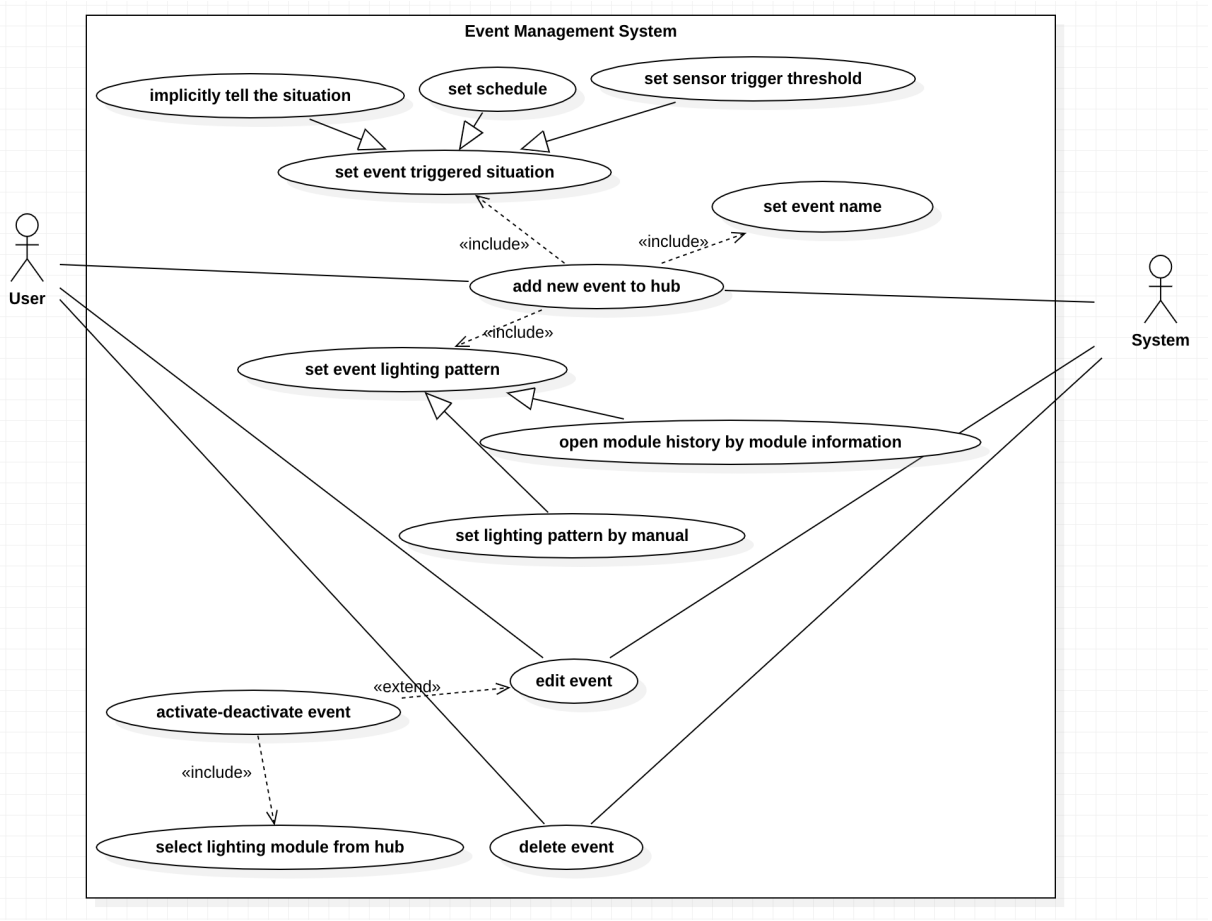
### (2) Module Information Management




| | |
|---|---|
| Use case name | Register new module |
| Related Requirements | B-1 |
| Goal in Context | The user registers a new module (lighting module and sensor modules to the hub. |
| Preconditions | User account for the hub has already been created in the System.<br>The user must be logged in with an existing user account.<br>The module must be able to connect to the hub by ZigBee. |

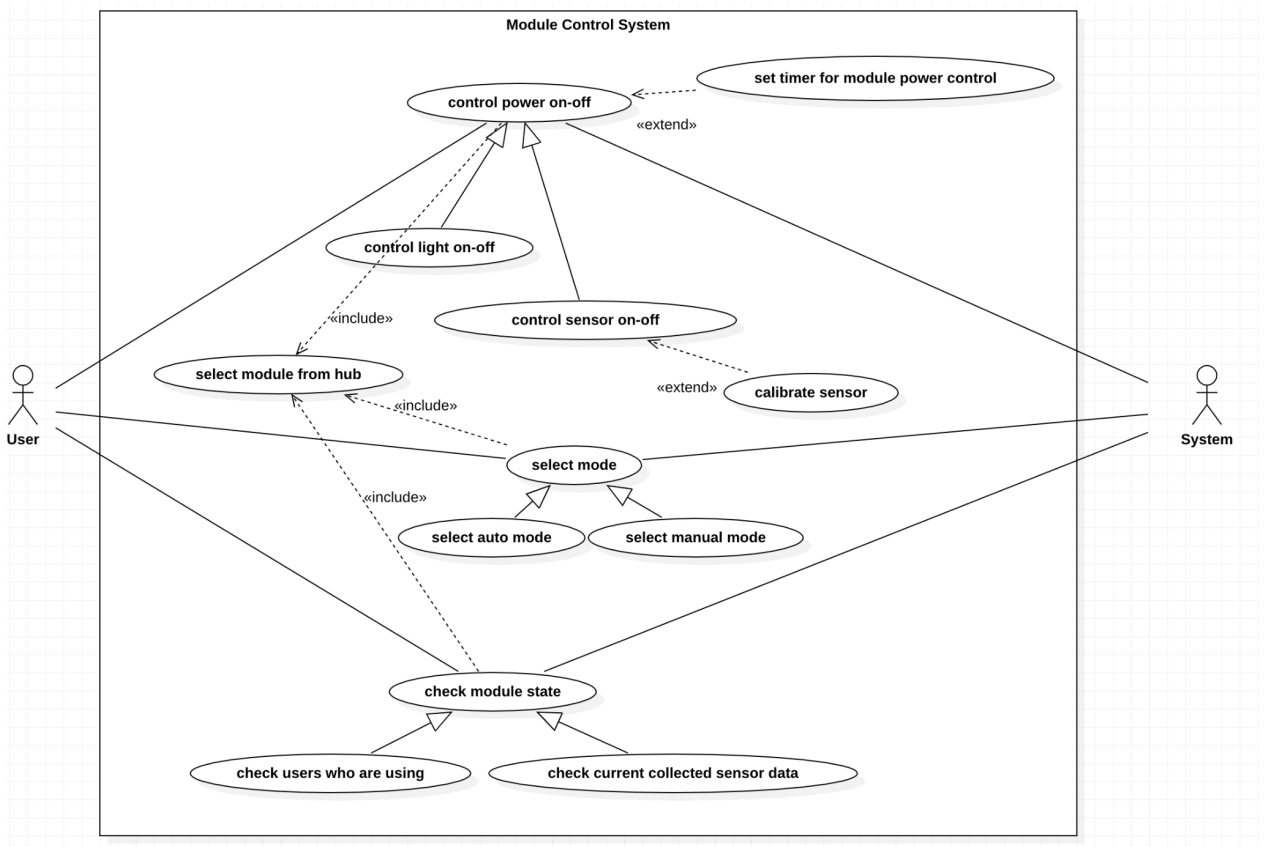
| | | |
|---|---|---|
| Successful End Condition | New module information is stored in the hub and it is connected to the hub. | |
| Failed End Condition | Module registration is rejected. | |
| Primary Actors | User | |
| Secondary Actors | System | |
| Trigger | The user asks the module information management system to register a new module. | |
| Main Flow | Step | Action |
| | 1 | The user asks the system to register a new module. |
| | 2 | The user selects the module type.(lighting, sensor) |
| | 3 | The user enters the module connection interface. |
| | 4 | The module is registered(The user sets the module name). |
| Extensions | Step | Action |
| | 3.1 | The module connection is lost. |
| | 3.2 | The module registration is rejected. |

### (3) Event Management



| | | |
|---|---|---|
| Use case name | Add new event | |
| Related Requirements | C-1 | |
| Goal in Context | The lighting event is added to the hub. | |
| Preconditions | User account for the hub has already been created in the System.<br>The user must be logged in with an existing user account.<br>The sensor modules and a lighting module used for the event which the user wants to set must be registered in the hub. | |
| Successful End Condition | Event is added to the hub. | |
| Failed End Condition | Adding events is rejected. | |
| Primary Actors | User | |
| Secondary Actors | System | |
| Trigger | The user asks the event management system to add a new event.<br>(Ex: The user invokes the event management system in auto mode by asking to add a new event.) | |
| Main Flow | Step | Action |
| | 1 | The user asks the system to add a new event. |
| | 2 | The user chooses which light pattern will be used for this event.<br>- Users could make a light pattern as manual mode(refer to *manual mode management system*).<br>- Or, checking module history, the user can refer to the previous light pattern(refer to *module information management system*). |
| | 3 | The user sets an event triggered situation.<br>- The user sets a sensor module trigger threshold.<br>- Or, the user sets the event schedule.<br>- Or, the user implicitly tells the situation. System automatically calculates the event triggered situation by using previous data). |
| | 4 | The user sets the event name. |
| | 5 | The event information is added to the event management system. |
| Extensions | Step | Action |
| | 2.1 | There are no lighting or sensor modules which can be used for the event. |
| | 2.2 | Adding an event is rejected. |

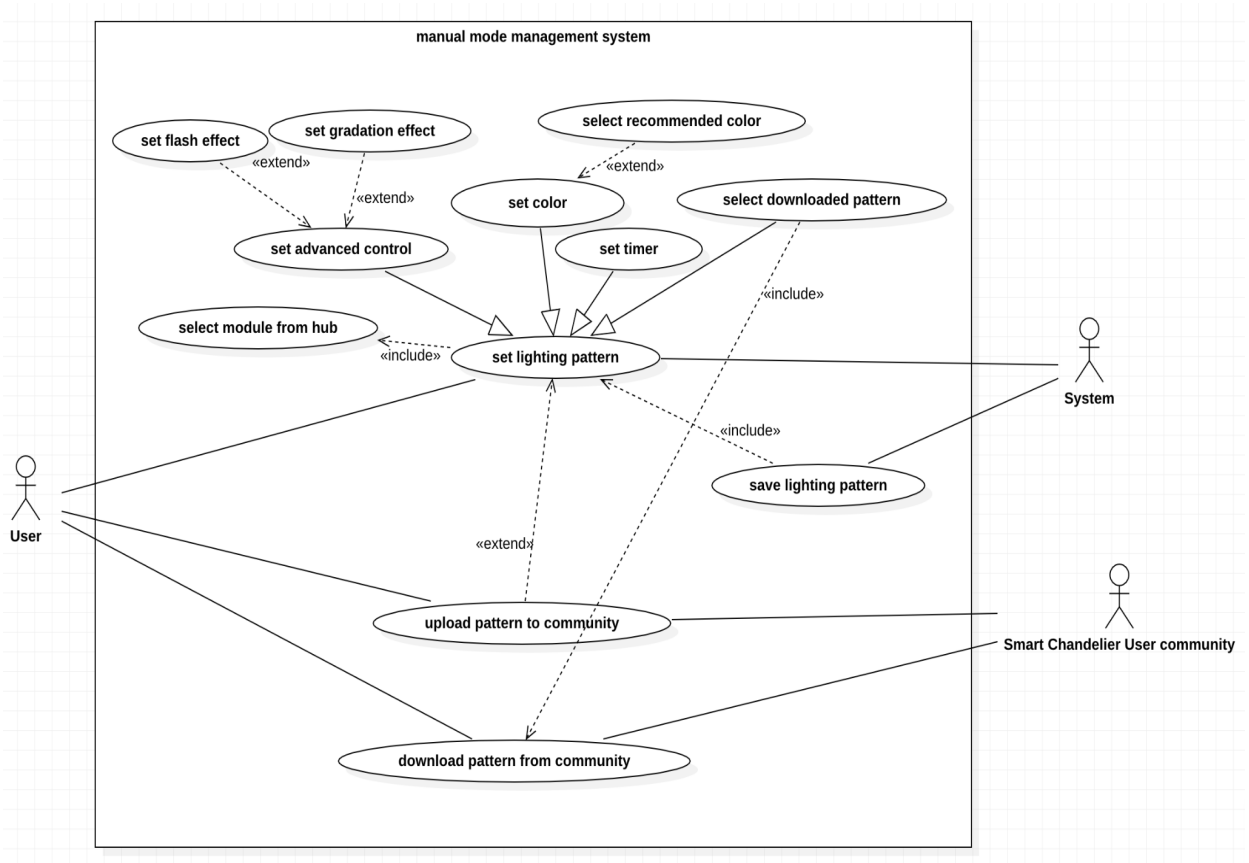
### (4) Module Control




| | | |
|---|---|---|
| Use case name | Control power on-off | |
| Related Requirements | D-1 | |
| Goal in Context | Control the power of modules which are connected to the hub. | |
| Preconditions | This system is limited to logged in users.<br>There must be at least one module(lighting or sensor) which is connected to the hub. | |
| Successful End Condition | The power of the module is successfully controlled by the user. | |
| Failed End Condition | Power control for the module is rejected. | |
| Primary Actors | User | |
| Secondary Actors | System | |
| Trigger | The user asks the module control system to control the module. | |
| Main Flow | Step | Action |
| | 1 | The user asks the system to control the module. |
| | 2 | The user selects one module. |
| | 3 | The user controls the power of the module..<br>- control lighting on-off (if the module is lighting)<br>- control sensor on-off (if the module is sensor)<br>- set timer to power |

| | | |
|---|---|---|
| | | - calibrate sensor(if the module is sensor) |
| Extensions | Step | Action |
| | 2.1 | There is no module connected to the hub. |
| | 2.2 | Power control is rejected. |

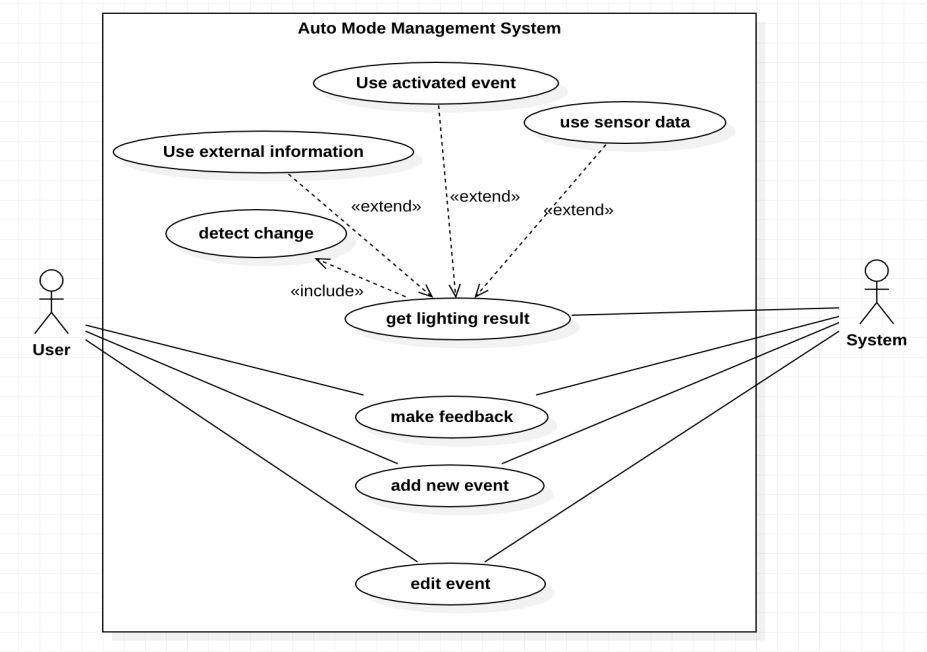
### (5) Manual Mode Management




| | |
|---|---|
| Use case name | Set lighting pattern |
| Related Requirements | E-1 |
| Goal in Context | User could change the lighting pattern as he or she wishes. |
| Preconditions | User account for the hub has already been created in the System.<br>The user must be logged in with an existing user account.<br>Users should select a lighting module to manipulate, and select manual mode. |
| Successful End Condition | Lighting pattern is verified by the System. |
| Failed End Condition | Lighting pattern is not verified by the System.<br>(Not sufficient information to make lighting pattern) |
| Primary Actors | User |
| Secondary Actors | System, Smart Chandelier User Community |
| Trigger | User asks the System to change the lighting pattern. |

| Main Flow | Step | Action |
|---|---|---|
| | 1 | User asks the system to change the lighting pattern from the current one. |
| | 2 | System shows the various options to change the light pattern.<br>- Set color<br>- Set timer(duration time)<br>- Select pre-downloaded pattern from Smart Chandelier User Community<br>- Set advanced control |
| | 3 | User selects a lighting pattern and sends it to the System. |
| | 4 | Lighting pattern is verified by the System and stored in the System. |
| | 5 | Current lighting pattern is changed. |
| | 6 | Optionally, users could upload the current pattern to the User Community. |
| Extensions | Step | Action |
| | 4.1 | Lighting pattern is not verified by the System.<br>(Not sufficient information to make light pattern) |
| | 4.2 | User's setting lighting pattern attempt is rejected. |

### (6) Auto Mode Management

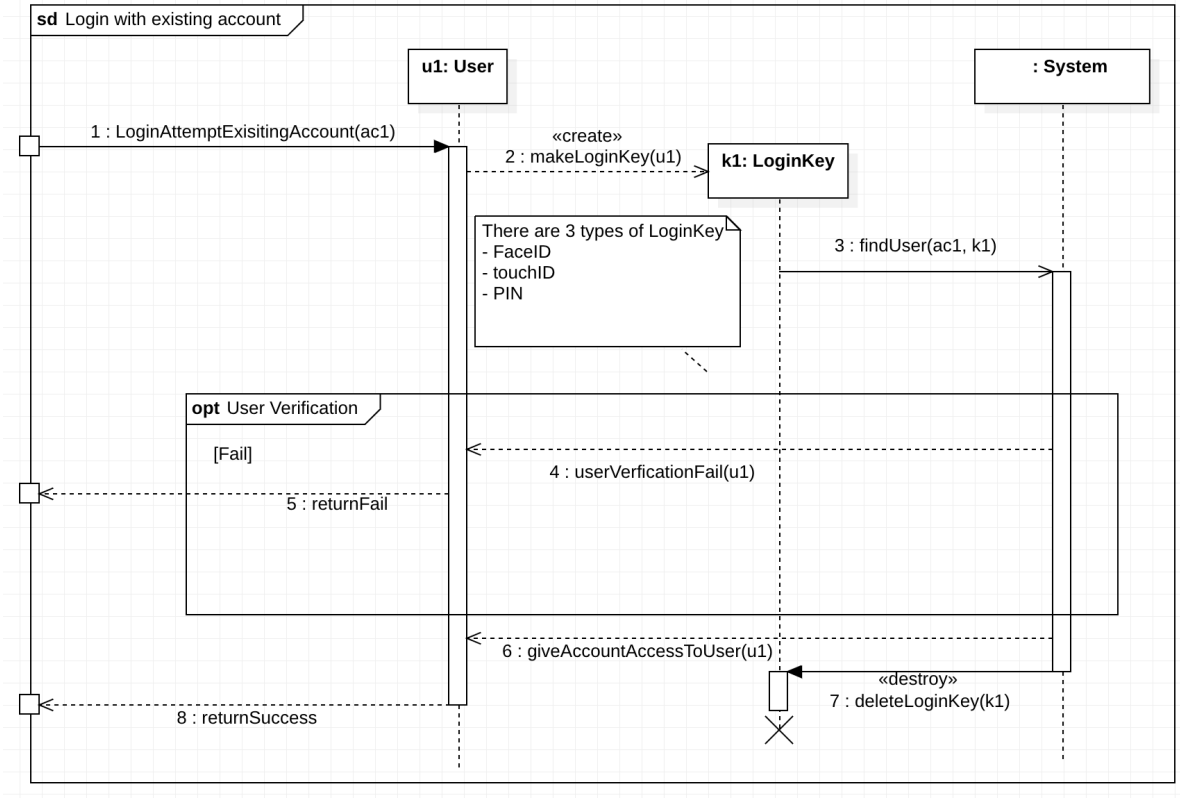



| Use case name | Get lighting result |
|---|---|
| Related Requirements | F-1 |
| Goal in Context | System automatically offers the lighting result to the lighting module. |
| Preconditions | User account for the hub has already been created by the System.<br>The user must be logged in with an existing user account.<br>User should select a lighting module to apply auto mode. |

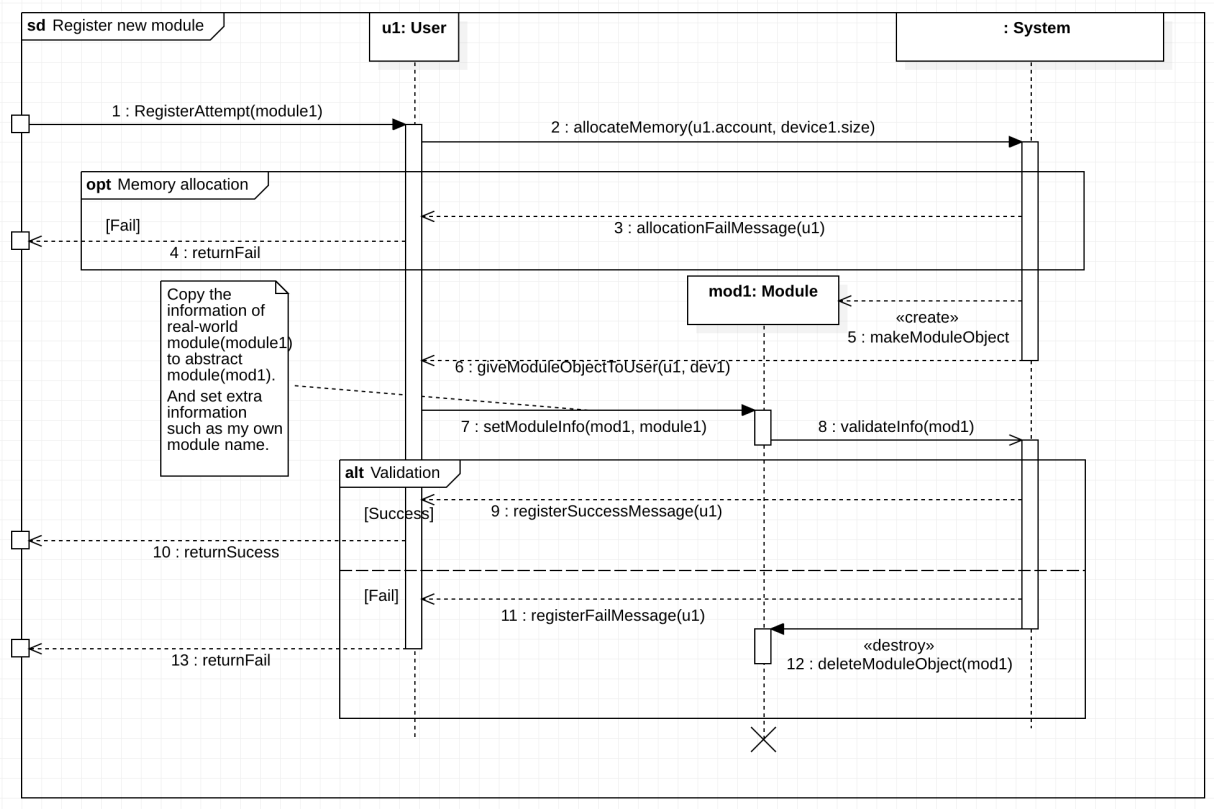
| | | |
|---|---|---|
| | User selects the auto mode. | |
| Successful End Condition | Lighting pattern is calculated automatically by the System. | |
| Failed End Condition | Lighting pattern is not calculated automatically by the System. | |
| Primary Actors | System | |
| Secondary Actors | User | |
| Trigger | System detects the change.<br>- Activated events are changed(refer to the event *management system*).<br>- The amount of change in sensor data exceeds a certain threshold. | |
| Main Flow | Step | Action |
| | 1 | System detects the change.<br>- An activated event is changed.<br>- The amount of change in sensor data exceeds a certain threshold. |
| | 2 | At the first time, System calculates the best light pattern.<br>- use sensor data<br>- use activated event<br>- (optional) if there is insufficient data, it uses external data on the internet, such as weather information. |
| | 3 | Current light pattern is changed. |
| | 4 | Optionally, user could do<br>- add a new event using the current light pattern.<br>- Or, edit an event using the current light pattern.<br>- Or, make feedback to the System. |
| Extensions | Step | Action |
| | 2.1 | Light pattern is not calculated automatically by the System. |
| | 2.2 | Auto mode is terminated. Tell the users by notification. |

## C. Sequence Diagrams

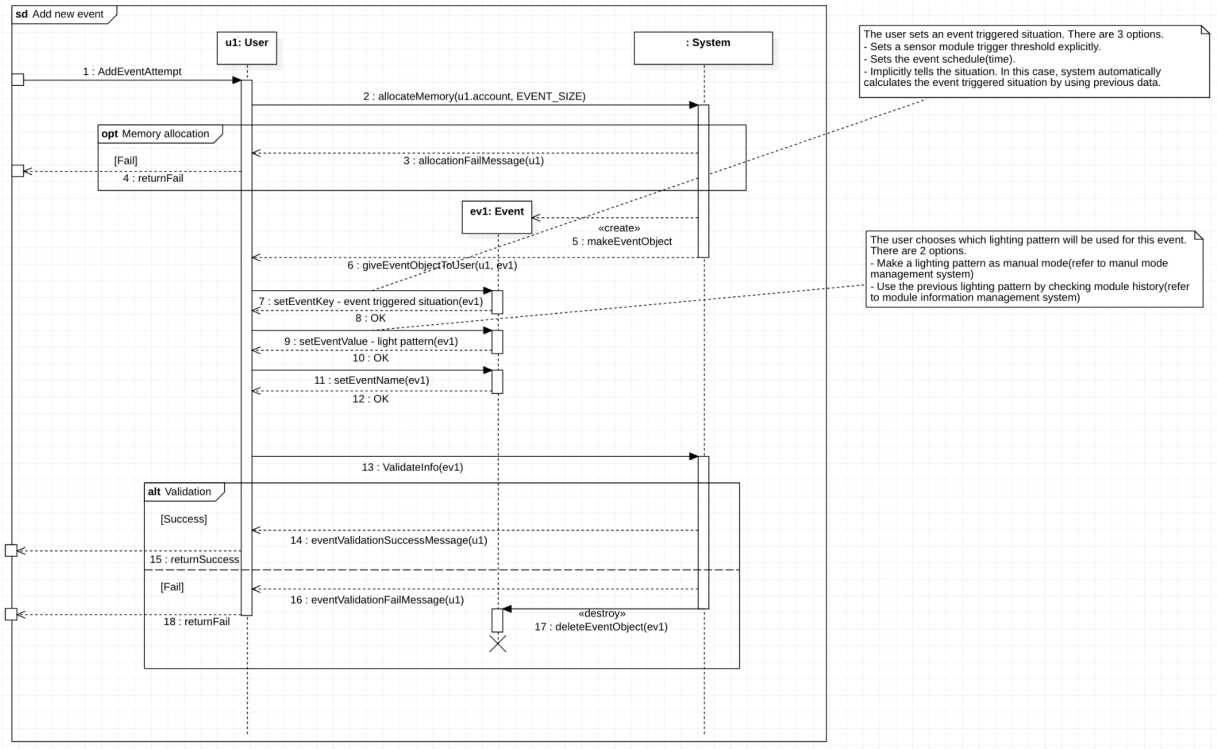
### (1) Login with existing account



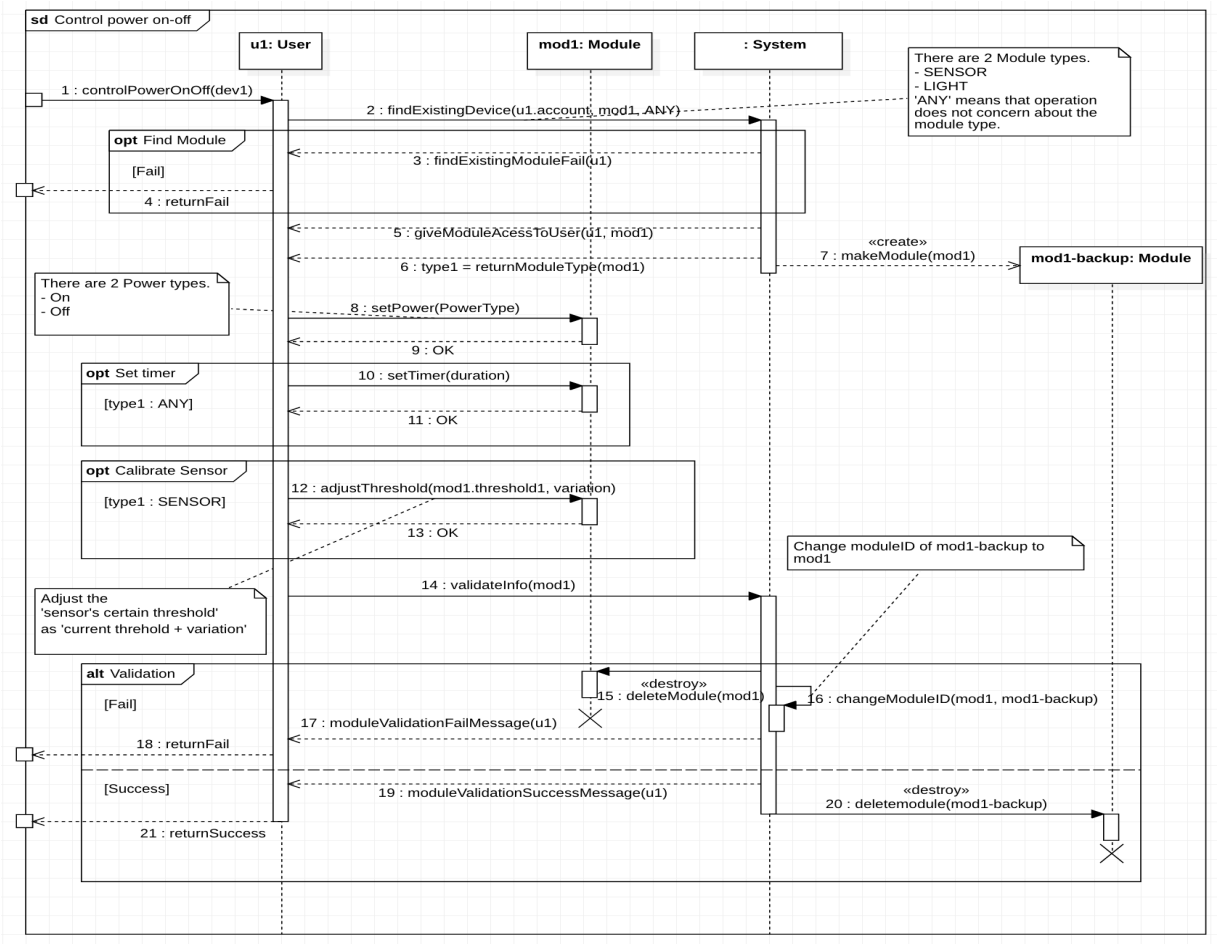

### (2) Register new module



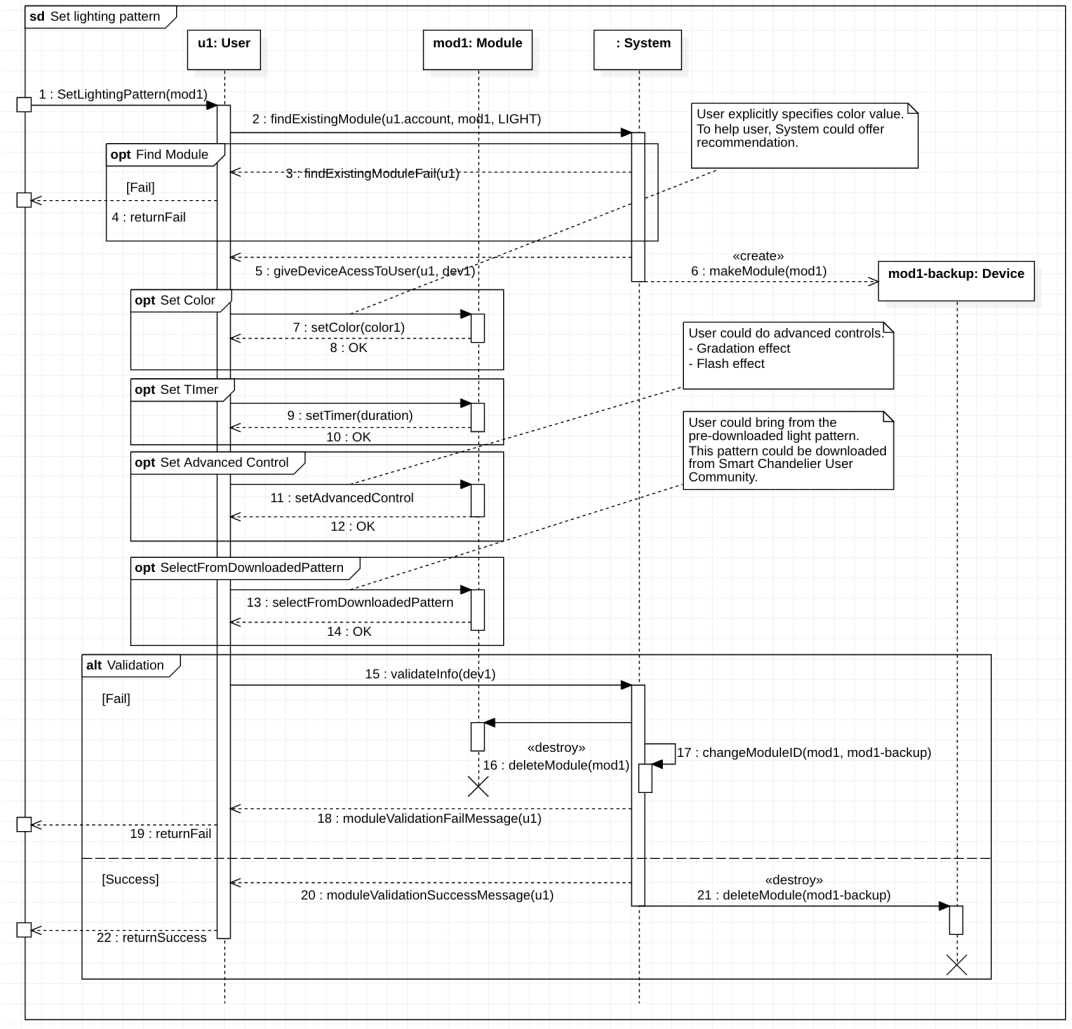
## (3) Add new event



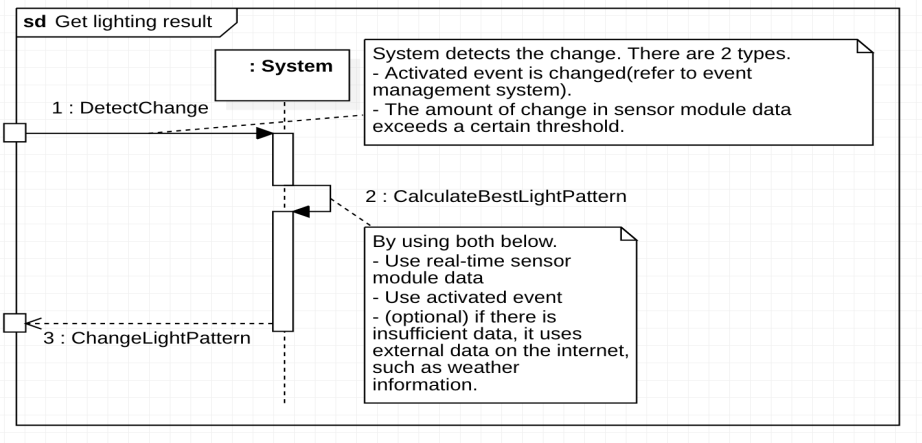

## (4) Control power on/off



### (5) Set lighting pattern




### (6) Get lighting result




## 4. Preliminary User Manual

### User of the manual

People who have installed the “Smart Chandelier” application in their smartphone.

People who have Smart Chandelier sensor modules, Smart Chandelier lighting modules, and SmartThings.

For referring to “A” GUI, please refer to this pdf Smart Chandelier GUI v2.1.pdf.
You can refer to a pdf page with “A” written on the top left of each page.

## Interface Manual

1. Account
   - 1.1. Create Account
     If you don’t have a Smart Chandelier user account, join in with your phone number, name, email address. This user information can be authenticated through SNS or mobile phone. When creating a user account, the hub can know the account information through the QR code on the hub device(The user should take a picture of QR code on the hub device while creating the account).
   - 1.2. Log in with account
     After you make an account, you can log in with that or existing account.
     If you forgot the password, you can find it.
   - 1.3. Add other user to hub account(refer to “Account_other users” GUI)
     You can log in to “Smart Chandelier” on many devices(phone, tablet...). The information of logged-in devices are shown in Account Information-Logged in Devices. You can change the name and profile photo of your device by the edit button.
   - 1.4. Manage Account Information (refer to “Account_info” GUI)
     You can change your hub account information. You can change phone numbers, passwords, etc…
     Account information shows various statistics using user data.
     Health (sleep) statistics are provided by analyzing sensor data records. A user behavior summary calendar is provided by analyzing the user's monthly movements and schedules.

2. Module Management
   In this section, the word “module” means lighting modules, sensor modules, and remote controllers.
   - 2.1. Register Module (refer to “register new module” GUI)
     In the Module Management Section, you can register your module. To register, press the “+” button. Select one of the connectable modules, and inform module type and module serial number. Also, you have to set the module name. The successfully registered module will be shown in the Module Management-Connected Modules section.
   - 2.2. See connected Modules (refer to “connected devices” GUI)
     You can see connected modules in the Module Management-Connected Modules section. You can enter a specific module control by pressing the edit(pen) button. You can delete the connected module by pressing the trash can button. You can control the on/off of each module.
   - 2.3. Control Module
     - 2.3.1. General Module Control
       In the Module Management-Connected Modules section, we can do general module control. This includes power on /off of each module.
     - 2.3.2. Specific Module Control (refer to “specific light control” GUI)
       You can change the name of the module. You can check the users who are currently using that module. You can add a timer to control power on/off of the module.
       - 2.3.2.1. If the module is lighting module (refer to “specific light control(2)” GUI)
         You can check the lighting history. Also, you can delete lighting history. You can select manual mode or auto mode. (You can enter ‘Auto Mode’ or ‘Manual Mode’ menu so that you can have a lot more control over it. )
       - 2.3.2.2. If the module is sensor module (refer to “specific sensor control” GUI)
         You can calibrate the sensor module.

You can check sensor-collected data.

3. Manual Mode

In manual mode, dedicated lighting module control is possible.

3.1. Manual Mode management (refer to “Manual mode” GUI)

To make manual control to a module, you have to choose one. You can see the devices list currently in Manual Mode. To enter manual mode, press the edit(pen) button of the device you want to control.

3.1.1. set color (refer to “Manual mode control” GUI)

You can set color by color picker. Any brightness, any color applies to the light. Smart chandelier's unique data analysis service suggests top 3 colors recommended in the current situation using sensor data.

3.1.2. set advanced Effect (refer to “Manual mode control(2)” GUI)

It supports not only color but also more advanced lighting effects. You can use gradients, flashing lights.

3.1.3. set light-on timer

You can set how long the lights set manually are turned on. This is done by a timer.

3.1.4. save the manual pattern and upload to community (refer to “Manual mode community” GUI)

If you like the pattern you created now, you can save it in your account. You can save it by setting a name when you save it. It is stored in Manual Mode Management-Manually saved patterns.

There is a smart chandelier user community. You can upload the lighting pattern created in manual mode to the community or download other people's patterns. Downloaded patterns are stored in Manual mode management-manually saved patterns.

You can check the lighting patterns in the community by sorting them in star order.

4. Event

“Event” refers to lighting patterns to be executed in specific situations. In other words, you can create a "situation" that is determined by a sensor or schedule, and set up your own lighting patterns to run in a "situation".

4.1. General event management (refer to “Event” GUI)

Events you create can be viewed in the Event Management-current events section. Events can be modified or deleted through edit button and delete button. You can add an event by pressing the + button. When the switch is turned off, the event is not referenced in auto mode.

4.2. Add new event (refer to “Add event” GUI)

To add an event, at least one of a sensor-based setup and a setting schedule is required. Setting a lighting pattern for each situation is essential.

4.2.1. Set name

You can set the event name.

4.2.2. Set sensor-based situation

The situation can be defined using sensors connected to the account. It can be based on when the sensor's data becomes a specific threshold. It can be based on the implicit sensor situation provided by a smart chandelier.

(Situation Example 1 : When the temperature sensor is 35 degrees or higher.

Situation Example 2 : When the temperature sensor is in a "hot" state.

The latter 'hot' situation is the implicit sensor situation provided by smart chandelier. )

4.2.3. Set schedule

You can use the schedule to define the situation.

For example, if you add a Christmas schedule on December 25, you can refer to the smart chandelier and provide red and blue lights suitable for Christmas.

5. Auto Mode

In Auto Mode, the Smart Chandelier recommendation system automatically analyzes your data and shows the result by lighting.

5.1. Auto mode management (refer to “Auto mode list” GUI)

If a lighting module is left as auto mode, auto mode is automatically applied to the module. Auto mode synthesizes sensor-collected data and events to produce recommended lighting patterns. If there is no sensor connected to the hub or no event, the lighting pattern is automatically set using external data(ex : weather) obtained through the API. If you want to give some control to auto mode(indicated below), you can select one module in “Modules currently in auto mode”.

5.1.1. give feedback (refer to “Auto mode control” GUI)

You can give feedback(5 star-rating) to the currently playing lighting pattern. Using these feedbacks, The Smart Chandelier recommendation system will be further customized.

5.1.2. add or edit new event

Since the auto mode recommends a lighting pattern based on events made by the user, you may want to modify or generate an event during execution of the auto mode. In this case, you can move to the event management system by pressing the “edit event” button.

5.1.3. Set lighting pattern

You defined the event situation above. It is time to set which lighting to execute in this situation. You have to choose which lighting pattern to set for which lighting. Setting the lighting pattern is the same as in manual mode.

Also, you can bring the lighting pattern from history.

## 5. Non-functional Requirements

| Type | Name | Description |
|---|---|---|
| Performance | Average TTFB speed on desktop | The average TTFB (Time to First Byte) speed is no more than 1.3 seconds on desktop |
| Performance | Average TTFB speed on mobile | The average TTFB (Time to First Byte) speed is no more than 2 seconds on mobile |
| Performance | Average page load time on desktop | The average Speed Index speed is no more than 3.5 seconds on desktop with a total 5000 simultaneous users. |
| Performance | Average page load time on mobile | The average Speed Index speed is no more than 4 seconds on mobile with a total 5000 simultaneous users. |
| Performance | Average size of a webpage on desktop | The average size of a website is no more than 0.8 Mb for desktop |
| Performance | Average size of a webpage on mobile | The average size of a website is no more than 0.6 Mb for Mobile |
| Performance | Average response time on an app | The average response time is no more than 1.5 seconds on the mobile app. |
| Performance | The uploadable file size | The playing time of each lighting pattern file should be less than 8 hours per unit. |
| Performance | Total number of files uploadable simultaneously | The user can upload three lighting pattern files at a time. |

| | | |
|---|---|---|
| Reliability | Total number of simultaneous log-ins | The system should be able to manage one million users without degradation in performance. |
| Reliability | Probability of complete access | Applicants have complete access to the community 99 percent of the time. |
| Availability | Total number of uploadable files throughout a week | Users can post 80 lightning patterns on the website throughout the week at any time during the day. |
| Availability | Total number of downloadable files in an hour | One user can download 30 lighting patterns in an hour. |
| Availability | Synchronizaton | When one user downloads a lighting pattern on a specific device, another user cannot download the pattern on the same device. |
| Availability | Favorite list | Users can add an uploader to their Favorites list. The maximum number of Favorites listed is 60. |
| Availability | Notification | The user gets notified when an uploader on their favorite list uploads a new pattern. |
| Available | Total number of simultaneous download accesses on one file | A download for the same item might be requested by up to 500 applicants. |
| Maintainability | Countermeasure | If the user community service becomes unavailable, it may be inaccessible for up to three hours while it is being maintained. |
| Maintainability | Backups | The system will retain mobile backups of all database updates for each record transaction to ensure data integrity. |
| Hardware Considerations | Sound pressure sensor | Noise levels between 35 dB pl and 90 dB pl are measured in 1 dB pl increments. |
| Hardware Considerations | Temperature sensor | Temperatures ranging from -40 to 80 degrees of celsius are measured in 1 degree increments, with an error range of 0.5 degrees. The frequency of data measurement is one hertz (Hz). |
| Hardware Considerations | Humidity sensor | Humidity ranging from 0 to 100 percent is measured, with an error range of 2~5 percent. The frequency of data measurement is one hertz (Hz). |
| Hardware Considerations | Motion sensor | The operating time is limited to less than 2 seconds. The range of detectable distance is from 0 to 12 meters. Additionally, it can detect between 0.2 m/s and 1.3 m/s of speed when the object moves more than 0.5m. Moreover, it should be installed at a height of 1.8 m to 3 m. |

| | | |
|---|---|---|
| User Interface | Countermeasure | If a temporary problem occurs, a notice instructing users to reload the site as well as an apology are displayed. |
| Documentation | Total pages of documentation | The user should read the documentation before beginning the service, which should be 5 pages for the mobile app and 3 pages for the web service. |
| Documentation | The content of documentation | The manual explains how to create, apply, and submit a pattern, as well as how to apply particular criteria to motions. |
| Documentation | Accessibility | The manual's font is at least 15pt, and numerous illustrations are included to make it simpler to read for people of all ages. |
| Environment | Device | The software system is accessible via a mobile app, a mobile webpage, and a PC webpage. |
| Environment | Physical Environment | Users may use as many Smart Chandeliers as they want. However, at least 8 square meters of space per Smart chandelier must be guaranteed. Users can purchase extra sensors separately and install them anywhere they want. |
| Security | Site admin | Only site administrators can view the applicant’s personal information. |
| Security | Password | Users may protect their history of patterns played and preferences with a password. |

## 6. Acknowledgements


| | |
|---|---|
| **Gyeongsu Kim** | 3. System Features<br>a. Functional Requirements<br>b. Use case Diagram & Description<br>c. Sequence Diagrams |
| **Daeun Kim** | 3. System Features<br>a. Functional Requirements<br>b. Use case Diagram & Description<br>4. Preliminary User Manual |
| **Hyuckjoong Yoon** | 1. Introduction<br>2. Overall Description<br>c. Operating Environment<br>d. Assumption and Dependencies<br>3. System Features<br>a. Functional Requirements |
| **Sana Kang** | 2. Overall Description<br>a. Product Perspective<br>b. Product Features<br>3. System Features<br>a. Functional Requirements<br>5. Non-functional Requirements |