# Software Requirements

Version 51, last updated by [rylandm](https://uocis.assembla.com/profile/rylandm" \t "_blank) at 2018-02-02

# Software Requirements Specification (SRS)

Revision History:

|  |  |  |
| --- | --- | --- |
| Date | Author | Description |
| 3-20-19 | Zheng Chen | Introduction/Concept of Operation |
| 3-21-19 | Zheng Chen | Quality Requirements/Expected subsets |
| 3-22-19 | Zheng Chen | Behavioral Requirements |
| 3-23-19 | Zheng Chen | Use Cases/Behavioral Requirements |
| 3-23-19 | Zheng Chen | Fundamental Assumption/Appendices |
| 3-26-19 | Zheng Chen | Revise Use Cases and System Inputs and Outputs. |
| 3-26-19 | Qingzhong Chen | Revise Use Cases |
| 3-27-19 | Zheng Chen | Revise Use Cases and Fundamental Assumption |
| 4-3-19 | Zheng Chen, Pedro | Revise Use Cases |
| 2019.4.8 | Zheng Chen | Revise System Context, Use Case for Customers and hardware and Revise System Input for Web App. |
| 2019.4.10 | Zheng Chen, Rui Raposo | Revise Use Case 2.3.1 according to Rui’s advice. |
| 2019.5.9 | Zheng Chen | Add Use Case 2.3.5-2.3.15 According To  New Requirements From Users And Revise Behavioral Requirements part. |
| 2019.5.25 | Zheng Chen | 1.Add The Use Case:2.3.2 User checks building list or room list.  2.Revise The Use Case: Administrator changes priorities between roles.  3.Add A Use Case: Administrator wants to check the alarm logs |

## 1.  Introduction

### 1.1    Intended Audience and Purpose

This document is intended to provided information guiding development process, ensuring that all system requirements are met. The following entities may find the document useful:  
Customer - This page will detail all of the web app requirements as understood by the production team. The customer should be able to determine that their requirements will be correctly reflected in the final product through the information found on this page.  
Development Team - Details of specific requirements that the final software build must include will be located here. Developers can use this document to ensure the software addresses each of these requirements.  
QA Team - By developing testing procedures founded in the system requirements, the QA Team can create a comprehensive testing regimen that will guarantee requirements are met.

### 1.2    How to use the document

Table of Contents:  
1. Introduction  
2. Concept of Operations - broad description of the purpose of the application  
  2.1 System Context - details any specific system requirements the application will require to run  
  2.2 System Capabilities - description in prose of all capabilities available to the user in the address book  
  2.3 Use cases - A detailed look at each functional requirement, describing the application context both before and after an action is taken  
3. Behavioral Requirements - How the application will interact with a user  
  3.1 Input and output requirments - A description of allowed inputs and generated outputs  
    3.1.1 Input - Describes any restrictions that will be placed on allowed input  
    3.1.2 Output - Describes the range of outputs that can be generated  
  3.2 Detailed Output Behavior - Output descriptions in prose  
4. Quality Requirements - Requirements not pertaining to the function of the application will be listed here  
5. Expected Subsets - Expected levels of functionality at checkpoints during development  
6. Fundamental Assumptions - Some specifics about input, output, or behavior upon which other requirements are founded will be listed here  
7. Expected Changes - Future features and directions the project is expected to take  
8. Appendices - Details aiding the understanding of this document  
  8.1 Definitions and acronyms - Any technical terms or abbreviations will be spelled out here for ease of use of the document  
    8.1 Definitions - Definitions of technical or unusual terminology  
    8.1.2 Acronyms and Abbreviations - Any abbreviated terms will be expanded here  
  8.2 References - any external references necessary or helpful to understanding this document will be listed here

## 2.  Concept of Operations

The goal is to create an intelligent lighting control system Web APP with good user experience. It will allow users to login, add rooms, delete rooms, add sensors, delete sensors, add drivers, delete drivers, turn on/off lights, check the status of lights, check the status of light sensors, check whether there are people in the room. Web applications interact with users using drop-down menus, buttons, text boxes and information prompt windows. For more details on the usage and functionality of applications, read the *System Functions* section.

### 2.1 System Context

**System Requirements:**  
Requires a system with a GUI display and browser because all of the operations are performed through a GUI and a browser.

Windows:

* Windows 10 (8u51 and above)
* Windows 8.x (Desktop)
* Windows 7 SP1
* Windows Vista SP2
* Windows Server 2008 R2 SP1 (64-bit)
* Windows Server 2012 and 2012 R2 (64-bit)

Mac OS X:

* Intel-based Mac running Mac OS X 10.8.3+, 10.9+

Linux:

* Red Hat Enterprise Linux 5.5+1, 6.x (32-bit), 6.x (64-bit)2
* Red Hat Enterprise Linux 7.x (64-bit)2 (8u20 and above)
* Ubuntu Linux 12.04 LTS, 13.x
* Ubuntu Linux 14.x (8u25 and above)
* Ubuntu Linux 15.04 (8u45 and above)
* Ubuntu Linux 15.10 (8u65 and above)

### 2.2 System capabilities

Intelligent light control system Web APP is a web program that supports user interaction. On the web page, the user logins the account according to his personal ID and password, and then carries on the concrete operation to the intelligent light control system. Different kinds of users have different rights to intelligent light control system. There are three different permissions: students, teachers and administrators. The system functions are as follows:

1.User login. Users must be students, teachers or administrators of some schools.

2.Check the state of the light. All users have this permission.

3.Check whether a room is occupied. All three users have this permission.

4.Check the state of the light sensor. In this function, users can see the situation of ambient light.

5.Turn on/off the lights. Student users can only turn on the light when it is off and the classroom is occupied, and turn off the light when it is on and the classroom is empty. When the relevant operation can not be carried out, a window will pop up to show the reasons: For example, *There are people in the classroom, so you can not turn off the lights*. Teachers and administrators directly force the lights to be on/off. Students, teachers and administrators can operate the switch of a light or the main switch of all lights.

6.Add/delete new rooms. Administrators have this permission.

7.Add/delete sensors. Administrators have this permission. There are three kinds of sensors: switch sensor, light sensor and Presence sensor.

8.Add/delete actuators (lights). Administrators have this permission.

### 2.3  Use Cases for Customer

#### **2.3.1 User login**

|  |  |  |  |
| --- | --- | --- | --- |
| Use Case | user login | | |
| Version | 1.0 | Created | 3-23-19 |
| Author | Zheng Chen | | |
| Source | User stories | | |
| Purpose | User login and go into the web application’s user interface. | | |
| Goals | User login and go into the web application’s user interface. | | |
| Summary | Login by inputting account number, password and press login button. | | |
| Actors | user | | |
| Trigger | Inputting account number, password and press login button. | | |
| Precondition | None | | |
| Basic Flow | Actor | | System |
| 1 | User(student, teacher and administrator)input account number and password. | |  |
| 2 | User press login button | |  |
| 3 |  | | system will process the answer from the server. If the login was successful the user will be sent to his homepage for web app or home screen for android app, otherwise the system will alert the user that his password or account is not correct. |
| 4 | User will get into the homepage or home screen, or will get the alert for wrong account information | |  |
| Frequency |  | | |
| Type | Primary | | |
| Postconditions | The web page is displayed. | | |
| Chart |  | | |
| Alternate Flow | *Actor* | | System |
|  |  | |  |

#### 2.3.2 User checks building list or room list.

|  |  |  |  |
| --- | --- | --- | --- |
| Use Case | User checks building list or room list | | |
| Version | 1.0 | Created | 3-23-19 |
| Author | Zheng Chen | | |
| Source | User stories | | |
| Purpose | check building list or room list. | | |
| Goals | check building list or room list. | | |
| Summary | Check building list or room list. | | |
| Actors | User | | |
| Trigger | User Choose a teaching building/ a room | | |
| Precondition | Login  sensors” | | |
| Basic Flow | Actor | | System |
| 1 | User pressed *see all buildings* button or choose a teaching building from a menu. | |  |
| 2 |  | | UI will return a list of teaching buildings or rooms. |
| Frequency |  | | |
| Type | Primary | | |
| Postconditions | The check result of light are displayed. | | |
| Chart |  | | |
| Alternate Flow | *Actor* | | System |
| 1 |  | |  |

#### 2.3.3 User checks the list of sensors and actuators(lights and sirens).

|  |  |  |  |
| --- | --- | --- | --- |
| Use Case | User checks building list or room list | | |
| Version | 1.0 | Created | 5-25-19 |
| Author | Zheng Chen | | |
| Source | User stories | | |
| Purpose | User checks the list of sensors and actuators(lights and sirens). | | |
| Goals | User checks the list of sensors and actuators(lights and sirens). | | |
| Summary | User checks the list of sensors and actuators(lights and sirens). | | |
| Actors | User | | |
| Trigger | User chooses a building from buildings’ drop-down menu and then chooses a room from rooms’ drop-down menu. | | |
| Precondition | Login  sensors” | | |
| Basic Flow | Actor | | System |
| 1 | User chooses a building from buildings’ drop-down menu and then chooses a room from rooms’ drop-down menu. | |  |
| 2 |  | | UI part will send building name, room number and command to server. |
| 3 | UI will get a list of sensors and actuators (lights and sirens). | |  |
| Frequency |  | | |
| Type | Primary | | |
| Postconditions | The check result of light are displayed. | | |
| Chart |  | | |
| Alternate Flow | *Actor* | | System |
| 1 |  | |  |

#### **2.3.4 User checks the state of lights or light sensors or checks whether someone is in room.**

|  |  |  |  |
| --- | --- | --- | --- |
| Use Case | User checks the state of lights or light sensors or checks whether someone is in room | | |
| Version | 1.0 | Created | 3-23-19 |
| Author | Zheng Chen | | |
| Source | User stories | | |
| Purpose | check the state of lights or light sensors or check whether someone is in room | | |
| Goals | check the state of lights or light sensors or check whether someone is in room | | |
| Summary | Check all states of lights and sensors and whether someone is in room by choosing room number and choosing teaching building. | | |
| Actors | user | | |
| Trigger | choosing room number and choosing teaching building | | |
| Precondition | Login and press “lights and  sensors” | | |
| Basic Flow | Actor | | System |
| 1 | User chooses teaching building name and room number from drop-down list and press enter button. | |  |
| 2 |  | | To server: UI part will send account number, token(for identity validation of connection with server), room number, teaching building and checking command. |
| 3 | The user checks result. | |  |
| 4 |  | | The server return lights' and light sensors' information, switch sensors and presence sensors’ information and whether someone is in room. |
| Frequency |  | | |
| Type | Primary | | |
| Postconditions | The check result of light are displayed. | | |
| Chart |  | | |
| Alternate Flow | *Actor* | | System |
| 1 |  | |  |

#### **2.3.5 User turns on/off the lights.**

|  |  |  |  |
| --- | --- | --- | --- |
| Use Case | User Turn on/off the lights | | |
| Version | 1.0 | Created | 3-23-19 |
| Author | Zheng Chen | | |
| Source | User stories | | |
| Purpose | User turns on/off the lights | | |
| Goals | User turns on/off the lights | | |
| Summary | User turns on/off the lights | | |
| Actors | user | | |
| Trigger | User press the turn on/off button. | | |
| Precondition | User logins and chooses room number and choose teaching building and choose lights. | | |
| Basic Flow | Actor | | System |
| 1 | User presses turn on/off button | |  |
| 2 |  | | UI part will send teaching building name, room number, light name and command to server. |
| 3 |  | | Server returns operation result |
| 4 | UI will display that the operation succeeded or failed . After that, UI will renew light state. | |  |
| Type | Primary | | |
| Postconditions | The result is displayed. | | |
| Chart |  | | |
| Alternate Flow | *Actor* | | System |
| 1 |  | |  |

#### **2.3.6 Administrator wants to see what alarms are on**

|  |  |  |  |
| --- | --- | --- | --- |
| Use Case | Administrator wants to see what alarms are on | | |
| Version | 1.0 | Created | 2019-5-8 |
| Author | Zheng Chen | | |
| Source | New requirements from users | | |
| Purpose | Administrator sees what alarms are on | | |
| Goals | Administrator sees what alarms are on | | |
| Summary | Administrator sees what alarms are on | | |
| Actors | Administrator | | |
| Trigger | Administrator press the *see what alarms are on* button. | | |
| Precondition | Administrator logins. | | |
| Basic Flow | Actor | | System |
| 1 | Administrator presses *see what alarms are on* button | |  |
| 2 |  | | Web application’s UI part will send the command of requiring the ringing alarms’ information to the server. |
| 3 |  | | Server returns a list of alarms which are on. |
| 4 | UI will display the list of alarms which are on. | |  |
| Type | Primary | | |
| Postconditions | None | | |
| Chart |  | | |
| Alternate Flow | Actor | | System |
| 1 |  | |  |

#### **2.3.7 Administrator wants to see in which room the panic button was pressed**

|  |  |  |  |
| --- | --- | --- | --- |
| Use Case | Administrator wants to see in which room the panic button was pressed | | |
| Version | 1.0 | Created | 2019-5-8 |
| Author | Zheng Chen | | |
| Source | New requirements from users | | |
| Purpose | Administrator wants to see in which room the panic button was pressed | | |
| Goals | Administrator wants to see in which room the panic button was pressed | | |
| Summary | Administrator wants to see in which room the panic button was pressed | | |
| Actors | Administrator | | |
| Trigger | Administrator press the *see in which room the panic button was pressed* button. | | |
| Precondition | Administrator logins | | |
| Basic Flow | Actor | | System |
| 1 | Administrator presses *see in which room the panic button was pressed* button | |  |
| 2 |  | | Web application’s UI part will send the command of requiring the related rooms’ information to the server. |
| 3 |  | | Server returns the room’s information in which the panic button was pressed. |
| 4 | UI will display the room’s information in which the panic button was pressed. | |  |
| Type | Primary | | |
| Postconditions | None | | |
| Chart |  | | |
| Alternate Flow | Actor | | System |
| 1 |  | |  |

#### **2.3.8 Administrator wants to add/remove buildings**

|  |  |  |  |
| --- | --- | --- | --- |
| Use Case | Administrator wants to add/remove buildings | | |
| Version | 1.0 | Created | 2019-5-8 |
| Author | Zheng Chen | | |
| Source | New requirements from users | | |
| Purpose | Administrator adds/removes buildings | | |
| Goals | Administrator adds/removes buildings | | |
| Summary | Administrator adds building by inputting buildings’ information. Administrator removes buildings by choosing buildings. | | |
| Actors | Administrator | | |
| Trigger | Administrator presses *add buildings* button or presses *remove buildings* button. | | |
| Precondition | Administrator logins | | |
| Basic Flow | Actor | | System |
| 1 | Administrator presses *add buildings*  buttonor presses *remove buildings* button. | |  |
| 2 |  | | If *remove buildings* button is pressed, UI will update the page with several buildings’ information and let the administrator choose which buildings he/she wants to remove.  If *add buildings* button was pressed, UI  will let you input the new buildings’ information. |
| 3 | If *remove buildings* button is pressed, the administrator chooses buildings and presses *confirm* button.  If *add buildings* button is pressed, the administrator inputs the new buildings’ information and presses *confirm* button. | |  |
| 4 |  | | UI will send building information and command to server |
| 5 | Administrator gets the adding/removing result, such as “remove successfully!”. | |  |
| Type | Primary | | |
| Postconditions | None | | |
| Chart |  | | |
| Alternate Flow | *Actor* | | System |
| 1 |  | |  |

#### **2.3.9 Administrator wants to add a room to a building or remove rooms from a building**

|  |  |  |  |
| --- | --- | --- | --- |
| Use Case | Administrator wants to add/remove rooms | | |
| Version | 1.0 | Created | 2019-5-8 |
| Author | Zheng Chen | | |
| Source | New requirements from users | | |
| Purpose | Administrator adds/removes rooms | | |
| Goals | Administrator adds/removes rooms | | |
| Summary | Administrator adds a room to a building by inputting rooms’ information and choosing a teaching building. Administrator removes rooms by choosing rooms and choosing a building. | | |
| Actors | Administrator | | |
| Trigger | Administrator presses *add a room to a building* buttonor presses *remove rooms from a building* button. | | |
| Precondition | Administrator logins | | |
| Basic Flow | Actor | | System |
| 1 | Administrator presses *add a room to a building* buttonor presses *remove rooms from a building* button. | |  |
| 2 |  | | UI will update the page with a drop-down menu  and let the administrator choose one building. |
| 3 | If *add a room* button is pressed, the administrator will input room number and press *confirm* button.  If *remove rooms* button is pressed, the administrator will choose rooms and press *confirm* button. | |  |
| 4 |  | | UI will send adding/removing command, room number(s) and building name to the server. |
| 5 | Administrator gets the adding/removing result, such as “remove successfully!”. | |  |
| Type | Primary | | |
| Postconditions | None | | |
| Chart |  | | |
| Alternate Flow | *Actor* | | System |
| 1 |  | |  |

#### **2.3.10 Administrator wants to give/revoke teacher permission to force light state in a room**

|  |  |  |  |
| --- | --- | --- | --- |
| Use Case | Administrator wants to give/revoke teacher permission to force light state in a room. | | |
| Version | 1.0 | Created | 2019-5-8 |
| Author | Zheng Chen | | |
| Source | New requirements from users | | |
| Purpose | Administrator gives/revokes teacher permission to force light state in a room. | | |
| Goals | Administrator gives/revokes teacher permission to force light state in a room. | | |
| Summary | Administrator gives teacher permission to force light state in a room by pressing *give teacher permission* button and inputting teacher’s information.  Administrator revokes teacher permission to force light state in a room by pressing *revoke teacher permission* button and inputting teacher’s information. | | |
| Actors | Administrator | | |
| Trigger | Administrator presses *give teacher permission* button or presses *revoke teacher permission* button. | | |
| Precondition | Administrator logins | | |
| Basic Flow | Actor | | System |
| 1 | Administrator presses *give teacher permission* buttonor presses *revoke teacher permission* button. | |  |
| 2 |  | | UI will let the administrator input teacher’s information. |
| 3 | Administrator inputs the teacher’s information and presses *confirm* button. | |  |
| 4 |  | | UI will send teacher’s information and command to server |
| 5 | Administrator gets this operation’s result, such as “operation successfully!”. | |  |
| Type | Primary | | |
| Postconditions | None | | |
| Chart |  | | |
| Alternate Flow | *Actor* | | System |
| 1 |  | |  |

#### **2.3.11 Administrator changes priorities between roles**

|  |  |  |  |
| --- | --- | --- | --- |
| Use Case | Administrator changes priorities between roles | | |
| Version | 1.0 | Created | 2019-5-8 |
| Author | Zheng Chen | | |
| Source | New requirements from users | | |
| Purpose | Administrator changes priorities between roles | | |
| Goals | Administrator changes priorities between roles | | |
| Summary | Administrator changes priorities between roles by pressing *change priorities between roles* button, pressing one role’s *modify* button, inputting a new priority value. | | |
| Actors | Administrator | | |
| Trigger | Administrator presses *change user’s authority* button. | | |
| Precondition | Administrator logins | | |
| Basic Flow | Actor | | System |
| 1 | Administrator presses *change priorities between roles* button | |  |
| 2 | Administrator presses one role’s *modify* button | |  |
| 3 |  | | UI will let you input a priority value between 0 and 99. |
| 4 | Administrator inputs a priority value to change the role’s priority . | |  |
| 5 | Administrator gets this operation’s result, such as “operation successfully!”. | |  |
| Type | Primary | | |
| Postconditions | None | | |
| Chart |  | | |
| Alternate Flow | *Actor* | | System |
| 1 |  | |  |

#### **2.3.12 Administrator wants to set up how long will the light stay on without anyone in the room**

|  |  |  |  |
| --- | --- | --- | --- |
| Use Case | Administrator wants to set up how long will the light stay on without anyone in the room. | | |
| Version | 1.0 | Created | 2019-5-8 |
| Author | Zheng Chen | | |
| Source | New requirements from users | | |
| Purpose | Administrator sets up how long will the light stay on without anyone in the room. | | |
| Goals | Administrator sets up how long will the light stay on without anyone in the room. | | |
| Summary | Administrator sets up how long will the light stay on without anyone in the room by pressing *set up how long will the light stay on without anyone in the room* button and inputting the time you want to set up. | | |
| Actors | Administrator | | |
| Trigger | Administrator presses *set up how long will the light stay on without anyone in the room* button | | |
| Precondition | Administrator logins | | |
| Basic Flow | Actor | | System |
| 1 | Administrator presses *set up how long will the light stay on without anyone in the room* button. | |  |
| 2 |  | | UI will let the administrator input the time. |
| 3 | Administrator inputs the time and presses *confirm* button. | |  |
| 4 |  | | UI will send the time and related command to server |
| 5 | Administrator gets this operation’s result, such as “operation successfully!”. | |  |
| Type | Primary | | |
| Postconditions | None | | |
| Chart |  | | |
| Alternate Flow | *Actor* | | System |
| 1 |  | |  |

#### **2.3.13 Administrator wants to define the default state of the lights when the system powers up**

|  |  |  |  |
| --- | --- | --- | --- |
| Use Case | Administrator wants to define the default state of the lights when the system powers up. | | |
| Version | 1.0 | Created | 2019-5-8 |
| Author | Zheng Chen | | |
| Source | New requirements from users | | |
| Purpose | Administrator defines the default state of the lights when the system powers up. | | |
| Goals | Administrator defines the default state of the lights when the system powers up. | | |
| Summary | Administrator defines the default state of the lights when the system powers up by pressing *define the default state of the lights* button and choose the state you want to define from the state drop-down menu. | | |
| Actors | Administrator | | |
| Trigger | Administrator presses *define the default state of the lights* button. | | |
| Precondition | Administrator logins | | |
| Basic Flow | Actor | | System |
| 1 | Administrator presses *define the default state of the lights* button. | |  |
| 2 |  | | UI will let the administrator choose the state. |
| 3 | Administrator chooses the state from state drop-down menu and presses *confirm* button. | |  |
| 4 |  | | UI will send the state and related command to server |
| 5 | Administrator gets this operation’s result, such as “operation successfully!”. | |  |
| Type | Primary | | |
| Postconditions | None | | |
| Chart |  | | |
| Alternate Flow | *Actor* | | System |
| 1 |  | |  |

#### **2.3.14 Administrator wants to add sensors to a room or remove sensors from a room.**

|  |  |  |  |
| --- | --- | --- | --- |
| Use Case | Administrator wants to add sensors to a room or remove sensors from a room | | |
| Version | 1.0 | Created | 2019-5-8 |
| Author | Zheng Chen | | |
| Source | New requirements from users | | |
| Purpose | Administrator adds/removes sensors | | |
| Goals | Administrator adds/removes sensors | | |
| Summary | Administrator adds sensors to a room by pressing *add sensors* button and inputting sensors’ information and choosing a building and a room. Administrator removes sensors by pressing *remove sensors* button and choosing a building a room and sensors. | | |
| Actors | Administrator | | |
| Trigger | Administrator presses *add sensors* buttonor presses *remove sensors* button. | | |
| Precondition | Administrator logins | | |
| Basic Flow | Actor | | System |
| 1 | Administrator presses *add sensors* buttonor presses *remove sensors* button. | |  |
| 2 |  | | UI will update the page with two drop-down menus and let the administrator choose one building and a room. |
| 3 | Administrator chooses a building and a room from two drop-down menus.  If *add sensors* button is pressed, the administrator inputs sensors’ information and presses *confirm* button.  If *remove sensors* button is pressed, administrator chooses sensors and finally presses *confirm* button. | |  |
| 4 |  | | UI will send adding/removing command, room number(s) and building name and sensor’s information to the server. |
| 5 | Administrator gets the adding/removing result, such as “remove successfully!”. | |  |
| Type | Primary | | |
| Postconditions | None | | |
| Chart |  | | |
| Alternate Flow | *Actor* | | System |
| 1 |  | |  |

#### **2.3.15 Administrator wants to add actuators(lights and siren) to a room or remove actuators(lights and siren) from a room**

|  |  |  |  |
| --- | --- | --- | --- |
| Use Case | Administrator wants to add actuators(lights and siren) to a room or remove  actuators(lights and siren)from a room. | | |
| Version | 1.0 | Created | 2019-5-8 |
| Author | Zheng Chen | | |
| Source | New requirements from users | | |
| Purpose | Administrator adds/removes actuators | | |
| Goals | Administrator adds/removes actuators | | |
| Summary | Administrator adds actuators(lights and siren) to a room by pressing *add actuators* button and inputting actuators’ information and choosing a building and a room. Administrator removes actuators by pressing *remove actuators* button and choosing a building a room and actuators. | | |
| Actors | Administrator | | |
| Trigger | Administrator presses *add actuators* buttonor presses *remove actuators* button. | | |
| Precondition | Administrator logins | | |
| Basic Flow | Actor | | System |
| 1 | Administrator presses *add actuators* buttonor presses *remove actuators* button. | |  |
| 2 |  | | UI will update the page with two drop-down menus and let the administrator choose one building and a room. |
| 3 | Administrator chooses a building and a room from two drop-down menus.  If *add actuators* button is pressed, the administrator inputs actuators’ information and presses *confirm* button.  If *remove actuators* button is pressed, administrator chooses actuators and finally presses *confirm* button. | |  |
| 4 |  | | UI will send adding/removing command, room number(s) and building name and actuators’ information to the server. |
| 5 | Administrator gets the adding/removing result, such as “remove successfully!”. | |  |
| Type | Primary | | |
| Postconditions | None | | |
| Chart |  | | |
| Alternate Flow | *Actor* | | System |
| 1 |  | |  |

#### **2.3.16 Administrator wants to turn off the alarms**

|  |  |  |  |
| --- | --- | --- | --- |
| Use Case | Administrator wants to turn off the alarms | | |
| Version | 1.0 | Created | 2019-5-8 |
| Author | Zheng Chen | | |
| Source | New requirements from users | | |
| Purpose | Administrator turns off the alarms. | | |
| Goals | Administrator turns off the alarms. | | |
| Summary | Administrator turns off alarms by pressing *turn off alarms* button and choosing a building. | | |
| Actors | Administrator | | |
| Trigger | Administrator presses *turn off alarms* button | | |
| Precondition | Administrator logins | | |
| Basic Flow | Actor | | System |
| 1 | Administrator presses *turn off alarms* button. | |  |
| 2 |  | | UI will update the page with a buildings drop-down menu. |
| 3 | Administrator chooses a building and presses *confirm* button. | |  |
| 4 |  | | UI will send building information and related command to server. |
| 5 | Administrator gets this operation result, such as “turn off alarms successfully!”. | |  |
| Type | Primary | | |
| Postconditions | None | | |
| Chart |  | | |
| Alternate Flow | *Actor* | | System |
| 1 |  | |  |

#### **2.3.17 Administrator wants to check the alarm logs**

|  |  |  |  |
| --- | --- | --- | --- |
| Use Case | Administrator wants to check the alarm logs | | |
| Version | 1.0 | Created | 2019-5-24 |
| Author | Zheng Chen | | |
| Source | New requirements from users | | |
| Purpose | Administrator wants to check the alarm logs. | | |
| Goals | Administrator wants to check the alarm logs. | | |
| Summary | Administrator checks the alarm logs by pressing *check alarm logs* button. | | |
| Actors | Administrator | | |
| Trigger | Administrator presses *check alarm logs* button | | |
| Precondition | Administrator logins | | |
| Basic Flow | Actor | | System |
| 1 | Administrator presses *check alarm logs* button | |  |
| 2 |  | | UI will send related command to server. |
| 3 | Administrator gets the alarm logs. | |  |
| Type | Primary | | |
| Postconditions | None | | |
| Chart |  | | |
| Alternate Flow | *Actor* | | System |
| 1 |  | |  |

### 2.4 Use Case for Web App Designers

#### **2.4.1 Server verifies login**

|  |  |  |  |
| --- | --- | --- | --- |
| Use Case | Server verifies login | | |
| Version | 1.0 | Created | 3-23-19 |
| Author | Zheng Chen | | |
| Source | User stories | | |
| Purpose | verify login | | |
| Goals | server get login information, verify it and then go into the light system | | |
| Summary | Server get information and verify it. | | |
| Actors | server | | |
| Trigger | user press login button. | | |
| Precondition | None | | |
| Basic Flow | Actor | | System |
| 1 | UI send command, account number and password to server | |  |
| 2 |  | | Server returns result of login. |
| 3 | UI displays the result of login. If login succeed, the homepage of user will be displayed. If login fails, a window will be poped out, “account or password is wrong" . | |  |
| Frequency |  | | |
| Type | Primary | | |
| Postconditions | The web page is displayed. | | |
| Chart |  | | |
| Alternate Flow | *Actor* | | System |
|  |  | |  |

#### **2.4.2 Server receives checking command from UI.**

|  |  |  |  |
| --- | --- | --- | --- |
| Use Case | Server checking command from UI | | |
| Version | 1.0 | Created | 3-23-19 |
| Author | Zheng Chen | | |
| Source | User stories | | |
| Purpose | Server checks. | | |
| Goals | Server checks the state of lights or light sensors or check whether someone is in room | | |
| Summary | Server checks all states of lights and sensors and whether someone is in room | | |
| Actors | server | | |
| Trigger | UI sends check command to server | | |
| Precondition | Login and press “lights and sensors” | | |
| Basic Flow | Actor | | System |
| 1 | From UI : server gets account number, room number, teaching building and user’s current right. | |  |
| 2 |  | | Server return information for checking |
| 3 | If the user is an ordinary user(student or teacher), the server will return lights' and light sensors' information and whether someone is in room. If the user is an administrator, the server return lights' and light sensors' information, other sensors’ information and whether someone is in room | |  |
| Frequency |  | | |
| Type | Primary | | |
| Postconditions | The checking result are displayed. | | |
| Chart |  | | |
| Alternate Flow | *Actor* | | System |
| 1 |  | |  |

#### **2.4.3 Server turns on/off the lights**

|  |  |  |  |
| --- | --- | --- | --- |
| Use Case | Server turn on/off | | |
| Version | 1.0 | Created | 3-23-19 |
| Author | Zheng Chen | | |
| Source | User stories | | |
| Purpose | Server turns on/off the lights | | |
| Goals | Server turns on/off the lights | | |
| Summary | Server turns on/off the lights | | |
| Actors | user | | |
| Trigger | User presses the turn on/off button. | | |
| Precondition | Login and check | | |
| Basic Flow | Actor | | System |
| 1 | server gets teaching building name, room number, light name and command to server. | |  |
| 2 | Server return operation result | |  |
| 3 |  | | UI will display that the operation succeeded or failed . After that, UI will renew light state. |
| Frequency |  | | |
| Type | Primary | | |
| Postconditions | The result is displayed. | | |
| Chart |  | | |
| Alternate Flow | *Actor* | | System |
| 1 |  | |  |

### 2.5 Use Case for Communication Module

#### **2.5.1 hardware sends signals and gets command**

|  |  |  |  |
| --- | --- | --- | --- |
| Use Case | hardware sends signals and gets command | | |
| Version | 1.0 | Created | 3-23-19 |
| Author | Zheng Chen | | |
| Source | User stories | | |
| Purpose | hardware sends signals and gets command | | |
| Goals | hardware sends signals and gets command | | |
| Summary | hardware sends signals and gets command | | |
| Actors | user | | |
| Trigger | Sensors send their data to communication module. | | |
| Precondition |  | | |
| Basic Flow | Actor | | System |
| 1 | Communication module verify connection to the server | |  |
| 2 |  | | Server will accept the connection and  tell communication module. |
| 3 | Switch sensor tells communication module whether light was operated or not.  Presence sensor sends a picture to raspberry pi to communication module.  Light sensor sends its state to communication module.  Panic button sends its state to communication module.  Siren sends its state to  communication module.  Light sends its state to communication module. | |  |
| 4 |  | | Communication module sends the switch sensor’s information and 0(not operated)/1(operated)signals to server.  Communication module uses image recognition algorithm to judge whether someone is in room. And then it sends 0(nobody) or 1(someone) signal and presence sensor's information to server.  Communication module send 0(dark) or 1(bright) signal and light sensor's information to server.  Communication module send 0(dark) or 1(bright) signal and light 's information to server.  Communication module send 0(dark) or 1(bright) signal and siren's information to server.  Communication module sends the panic button’s information and 0(not operated)/1(operated)signals to server. |
| Frequency |  | | |
| Type | Primary | | |
| Postconditions |  | | |
| Chart |  | | |
| Alternate Flow | *Actor* | | System |
| 1 |  | |  |

#### **2.5.2 Server gets signals from communication module**

|  |  |  |  |
| --- | --- | --- | --- |
| Use Case | Server gets signals from communication module | | |
| Version | 1.0 | Created | 3-23-19 |
| Author | Zheng Chen | | |
| Source | User stories | | |
| Purpose | Server gets signals from communication module | | |
| Goals | Server gets signals from communication module | | |
| Summary | Server gets signals from communication module | | |
| Actors | user | | |
| Trigger | Sensors send their data to communication module. | | |
| Precondition |  | | |
| Basic Flow | Actor | | System |
| 1 | server verifies connection from hardware. | |  |
| 2 | Server gets the switch sensor’s information and 0(not operated)/1(operated)signals.  Server gets send 0(nobody) or 1(someone) signal and presence sensor's information.  Server gets 0(dark) or 1(bright) signal and light sensor's information.  Server gets 0(on) or 1(off) signal and light’s information.  Server gets 0(on) or 1(off) signal and siren’s information.  Server gets 0(on) or 1(off) signal  And panic button’s information. | |  |
| 3 | The Server decides whether the light should be on or not. | |  |
| 4 |  | | Communication module sends command to lights. |
| Frequency |  | | |
| Type | Primary | | |
| Postconditions |  | | |
| Chart |  | | |
| Alternate Flow | *Actor* | | System |
| 1 |  | |  |

## 3.    Behavioral Requirements

### 3.1 System Inputs and Outputs

#### 3.1.1 Inputs

The input of the application comes from the user.

Login interface comes at the beginning. There are two text boxes to be entered, account number and password.

In the navigation bar, there are "home page", "lights", "Sensors", "rooms", "current user identity" and "user personal information". Click on "lights" and there will be two drop-down menus of "building name" and "room number", "enter" and "return to the previous page" buttons on the left side of the interface. After clicking "Enter", there are all the lights in the room on the right side of the interface, as well as the switch of the lights, the check of the lights (full selection, reverse selection), the status of the light sensor and the prompt information box of the room.

For administrators, there is also a *managements* button in the navigation bar. If this button is pressed, there will be several buttons appearing, such as *see* *what alarms are on* button, *see in which room the panic button was pressed* button*, add buildings* button*, remove buildings* button*, add a room to a building* button*, remove rooms from a building* button*, give teacher permission* button*, revoke teacher permission* button*, change user’ s authority* button, *set up how long will the light stay on without anyone in the room* button, *define the default state of the lights* button, *add sensors* button*, remove sensors* button, *add actuators* button*, remove actuators* button and *turn off alarms* button*.*

Input at login interface:

\* Account: must be made up of numbers. It can only be one of the teaching number, teacher's work number and administrator's ID number.

\* Password: 6-20 characters.

\* Login: Click on this button to enter the next interface with the correct account number and password.

Under "sensors", click on the *Add* button and enter the following:

\* Sensor types: Only one of three types can be selected from the drop-down menu.

Under "rooms", click the Add button and enter:

\* Room number: Input cannot conflict with an existing room number. And it is less than 5 legal numbers or letters.

Input basic information:

\* Nickname: less than 20 characters

\* ID number: less than 10 digits

\* School: less than 300 characters

\* Professional: less than 20 characters

\* Class: less than 20 characters

"Modify password" input:

\* Old passwords: 6-20 characters

\* "New password": 6-20 characters.

#### 3.1.2 Outputs

Display graphical user interface. Each current interface contains all text boxes or interactive buttons created for users to enter.

Output to the user:

Login interface:

\* If the password or account is incorrect, a pop-up window will prompt "incorrect password or account".

Turn on the lights:

\* If the user is a student and the room is occupied, when the "turn on" button is pressed, a pop-up window will prompt "the room is occupied, the students can not turn off the lights at will". If the room is unoccupied, when the "turn off" button is pressed, a window will pop up to indicate that "the room is unoccupied", and students can not turn on the light at will. If the switch is checked, similar.

### 3.2 Detailed Output Behavior

Login interface comes at the beginning. There are two text boxes to be entered, account number and password.

In the navigation bar, there are "home page", "lights", "Sensors", "rooms", "current user identity" and "user personal information". Click on "lights" and there will be two drop-down menus of "building name" and "room number", "enter" and "return to the previous page" buttons on the left side of the interface. After clicking "Enter", there are all the lights in the room on the right side of the interface, as well as the switch of the lights, the check of the lights (full selection, reverse selection), the status of the light sensor and whether someone is in the room.

Click on "sensors" and there will be two drop-down menus of "building name" and "room number", "enter" and "return to the previous page" buttons on the left side of the interface. Click "Confirm" and all the sensors and their status will appear on the right side of the interface.

Click on "rooms" and there will be a drop-down menu of "teaching building name", "confirmation" and "return to the previous page" buttons on the left side of the interface. Click on the "Confirm" button and all the room numbers in this building will appear on the right side of the interface.

Click on "User Personal Information" and the buttons "Basic Information" and "Modify Password" appear on the left side of the interface. After clicking on the "basic information", there will be "nickname", "ID number", "school", "major" and "class" on the right side of the interface, as well as a "confirm modification" button. Click "Modify Password" and the text box of "New Password" and "Old Password" will appear on the right side of the interface, and the button "Confirm Modification" will appear.

For administrators, there is also a *managements* button in the navigation bar. If this button is pressed, there will be several buttons appearing, such as *see* *what alarms are on* button, *see in which room the panic button was pressed* button*, add buildings* button*, remove buildings* button*, add a room to a building* button*, remove rooms from a building* button*, give teacher permission* button*, revoke teacher permission* button*, change user’ s authority* button, *set up how long will the light stay on without anyone in the room* button, *define the default state of the lights* button, *add sensors* button*, remove sensors* button, *add actuators* button*, remove actuators* button and *turn off alarms* button*.*

Click on *see* *what alarms are on* button, and UI will display the list of alarms which are on in the main page. Click on *see in which room the panic* button, and UI will display the room’s information in which the panic button was pressed. Click on *add buildings* button *o*r *remove buildings* button*.* If *remove buildings* button is pressed, UI will show up a building drop-down menu and a *confirm* button. If *add buildings* button is pressed, UI will show up a *confirm* button and a text box which lets administrator input the new buildings’ information. Click on *add a room to a building* button or *remove rooms from a building* button. UI will update the page with a building drop-down menu and let the administrator choose one building. If *add a room* button is pressed, UI will show up a *confirm* button and a text box which lets administrator input room number. If *remove rooms* button is pressed, the UI will show up a *confirm* button and a list of check boxes which lets administrator choose rooms to remove. Click on *give teacher permission* button or *revoke teacher permission* button. UI will show up a *confirm* button and a text box which lets administrator input the teacher’s information. Click on *change user’s authority* button. UI will show up a text box which let administrator input user’s information, a drop-down menu which lets the administrator choose new authority and finally a confirm button. Click on *set up how long will the light stay on without anyone in the room* button. UI will show up a *confirm* button and a text box which lets the administrator input the time. Click on *define the default state of the lights* button. UI will show up a *confirm* button and a state drop-down menu which lets administrator choose the state. Click on *add sensors* button or  *remove* sensors button. Administrator chooses a building and a room from two drop-down menus. If *add sensors* button is pressed, UI will show up a *confirm* button and a text box which lets administrator input sensors’ information. If *remove sensors* button is pressed, UI will show up a *confirm* button and a list of check boxes which lets the administrator choose sensors.

Click on *add actuators* button or *remove actuators* button. Administrator chooses a building and a room from two drop-down menus. If *add actuators* button is pressed, UI will show up a *confirm* button and a text box which lets the administrator input actuators’ information. If *remove actuators* button is pressed, UI will show up a *confirm* button and a list of check boxes which lets administrators choose actuators.

Click on *turn off alarms* button. UI will show up a *confirm* button and a drop-down menu which lets administrator choose a building.

## 4. Quality Requirements

Interface aesthetics:

\* Simple, comfortable and elegant.

Performance: Response to user input

\* Operation execution time does not exceed 2 seconds (considering network delay and other factors).

\* The speed of opening web pages is fast. The access to login interface is not more than 0.5s, and the server configuration is better.

Reliability:

\* Ensure that the actions taken do not lead to errors and that the changes made to the light control system are durable.

Consistency: Information about the number, name, number of lights and rooms is persistent

\* Content should be modified after adding, so that the target field can be changed without affecting other field data.

Extensibility: Easy to extend application functionality

\* Applications should be modular so that when adding/extending functions, functions requires only changes to individual components and their interfaces, if applicable.

Safety:

\* Maintenance personnel regularly enhance and maintain the defensive capabilities of the website.

Stability:

\* In the case of high network traffic, we should increase the number of servers and other means to ensure the user's experience as good as possible.

## 5.    Expected Subsets

L0:

- Basic GUI.

- Users can log in. Ability to send data to back-end storage and call data from back-end storage.

L1:

- Complete GUI for Intelligent Lighting Control

- Ability to see the status of the light. All three users have this permission.

- Check if a room is occupied. All three users have this permission.

- Ability to check the status of the light sensor. All three users have this permission.

- Ability to turn on/off the light. All three users have this right.

## 6.    Fundamental Assumptions

The Web APP can run on any browsers which are chrome or firefox.

## 7.    Expected Changes

   Features to Add:

1. Adjust the brightness of the light

2. Personal Web Pages for Skin Change

3. Provide personalized web customization

4. Provide hotline for maintenance personnel.

5. Provide multilingual support.

6. Retrievable password and change password at any time

7. Support binding mobile phone number and login by phone number.

## 8.    Appendices

### 8.1    Definitions and acronyms

#### 8.1.1    Definitions

|  |  |
| --- | --- |
| **Keyword** | **Definitions** |
|  |  |
|  |  |
|  |  |

#### 8.1.2    Acronyms and abbreviations

|  |  |
| --- | --- |
| **Acronym or**  **Abbreviation** | **Definitions** |
| GUI | Graphical User Interface |
| W3C | World Wide Web Consortium |
|  |  |

### 8.2    References

The design of web page follows W3C standards.

Comments are disabled for this space. In order to enable comments, Messages tool must be added to project.

You can add Messages tool from Tools section on the Admin tab.