

사용자 맞춤형 스마트 홈 서비스

캡스톤 디자인 II 중간보고서

Version 1.0

팀장 김원겸
팀원 김도연
손정선

대표 연락처 : 010-4616-0680
대표 e-mail : 20191768@edu.hanbat.ac.kr

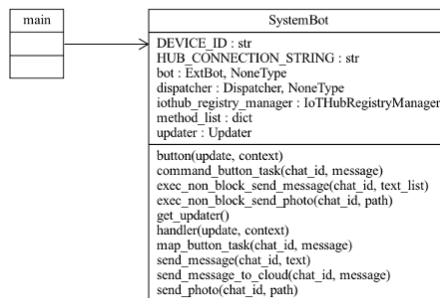
캡스톤 디자인 II 중간보고서

1. 요구사항 정의서에 명세된 기능에 대하여 현재까지 진척된 결과 및 그 내용을 기술하시오.

본 프로젝트의 스마트 홈서비스를 구현하기 위해, 제어 장치로 클라우드를 통해 사용자와 실시간 상호작용을 하는 부분을 우선적으로 연구했다.

미들웨어로 구현된 간단한 게임을 기반으로 게임 시스템 외부에 존재하는 ‘텔레그램’을 통해 사용자로부터 명령을 입력받는 구조로 이루어진다. 입력받은 명령으로 시스템 내부에서 동작된 결과를 클라우드를 통해 사용자에게 전달함과 동시에, 명령 및 결과를 데이터베이스에 저장하여 이를 통해 사용 패턴과 빈도 등을 분석하고자 한다.

텔레그램은 제어 장치 중 하나인 채팅 애플리케이션으로, 사용자로부터 입력받은 명령을 클라우드를 통해 미들웨어로 전달하는 기능을 수행한다.



<제어 장치의 클래스 다이어그램>

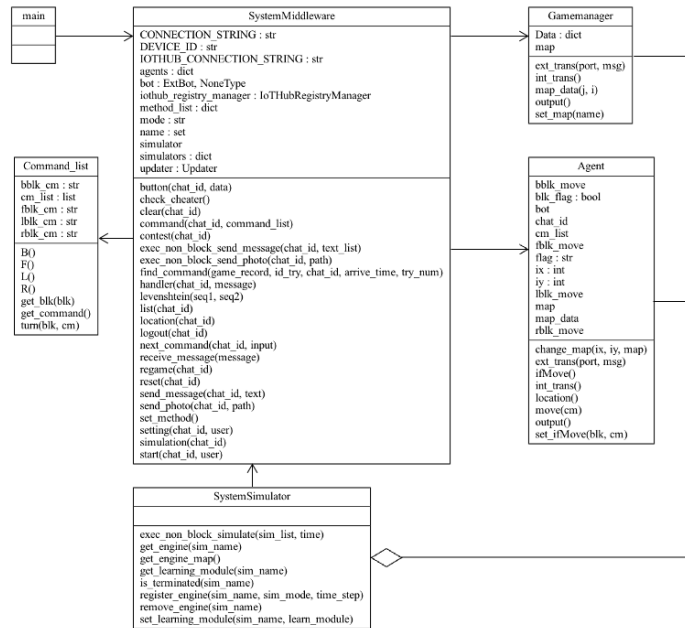
```
main.py
from system_bot import *

updater = SystemBot.get_updater()
updater.start_polling()
updater.idle()

system_bot.py
def send_message_to_cloud(chat_id, message):
    try:
        if message in SystemBot.method_list.keys():
            SystemBot.method_list[message](chat_id, message)
            registry_manager = IoTHubRegistryManager(SystemBot.HUB_CONNECTION_STRING)
            props={}
            props.update(messageId = chat_id)
            registry_manager.send_c2d_message(SystemBot.DEVICE_ID, message, properties=props)
            query_spec = QuerySpecification(query="SELECT tags FROM devices WHERE is_defined(tags.map)")
            query_result = SystemBot.iothub_registry_manager.query_iot_hub(query_spec, None, 100)
            for twin in query_result.items:
                data_dic = eval(str(twin))
                if message == 'location' or message == 'simulation':
                    SystemBot.exec_non_block_send_photo(chat_id, visualize_map(chat_id, data_dic['tags']['map']))
    except Exception as ex:
        print ( "Unexpected error {0}" % ex )
        return
    except KeyboardInterrupt:
        print ( "IoT Hub C2D Messaging service stopped" )
def handler(update, context):
    chat_id = str(update.effective_chat.id)
    message = update.message.text
```

```
SystemBot.send_message_to_cloud(chat_id, message)
```

미들웨어는 클라우드를 통해 입력된 명령에 따라 동작된 결과를 디바이스 트윈에 업데이트하여 사용자에게 전달하고, 동시에 명령 및 결과를 데이터베이스에 저장하는 기능을 수행한다.



<미들웨어의 클래스 다이어그램>

system_middleware.py

```

@staticmethod
def receive_message(message):
    command = vars(message)['data'].decode('utf-8').split()
    chat_id = vars(message)['message_id']

    SystemMiddleware.handler(chat_id = int(chat_id), message = command)

@staticmethod
def handler(chat_id=None, message=None):
    try :
        method = SystemMiddleware.method_list[message[0]]
        if message[0]=='start':
            text = method(chat_id, message[1])
        else :
            if chat_id in SystemMiddleware.agents.keys():
                text = method(chat_id)
    except :
        if chat_id in SystemMiddleware.agents.keys():
            method = SystemMiddleware.command
            text = method(chat_id, message)

```

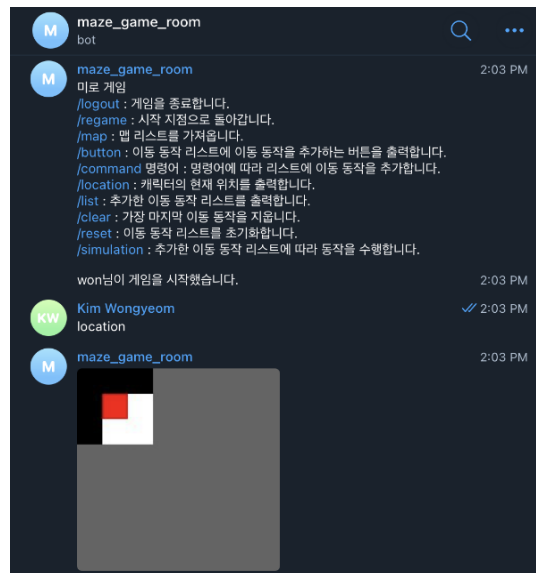
main.py

```

from system_middleware import *

CONNECTION_STRING = "{deviceConnectionString}"
client = IoTHubDeviceClient.create_from_connection_string(CONNECTION_STRING)
SystemMiddleware.set_method()
while True :
    try:
        client.on_message_received = SystemMiddleware.receive_message
        while True:
            time.sleep(1000)
    except :
        sys.exit(0)

```



이는 프로젝트의 전체 수행 과정 중 캡스톤 디자인I에 해당하여, 캡스톤 디자인II의 1, 2단계를 구현하기 위한 기반으로 사용될 것이다.

현재 캡스톤 디자인II의 범위에서는 가전기기 제어를 위한 하드웨어 기능 구현까지 완료된 상태이며, 그래픽 및 웹 구현을 위한 역량 강화 과정에 있다.

하드웨어 기능 구현 코드는 아래와 같다. 냉방기 제어를 위해 LG 에어컨을 사용했으며 추후 삼성 에어컨도 추가할 계획이다. 또, 가전기기 혹은 환경정보를 각각 개별적인 보드에서 수행 중이나 추후 하나의 브레드보드에 통합하는 과정을 거칠 예정이다.

dht11.ino (온습도 조회)	servo.ino (조명 스위치 제어)
<pre>#include <dht.h> dht DHT; #define DHT11_PIN 2 float hum; float temp; void setup(){ Serial.begin(9600); } void loop(){ DHT.read11(DHT11_PIN); hum = DHT.humidity; temp = DHT.temperature; Serial.print("Humidity: "); Serial.print(hum); Serial.print(" %,Temp: "); Serial.print(temp); Serial.println(" Celsius"); delay(2000); }</pre>	<pre>#include <Servo.h> Servo servo; void setup(){ Serial.begin(115200); servo.attach(2); servo.write(90); } void loop(){ long value = 0; if (Serial.available()) { value = Serial.parseInt(); if (value == 1) { Serial.println("Left"); servo.write(35);delay(300); servo.write(90);delay(300); } else if (value == 2){ Serial.println("Right"); servo.write(145);delay(300); servo.write(90);delay(300); } } }</pre>

mhz19b.ino (이산화탄소 농도 조회)	ir_transmitter_lg.ino (냉방기 제어)
<pre> #include <ESP8266WiFi.h> #include <ESP8266WebServer.h> #include <SoftwareSerial.h> #include <MHZ19.h> #define HOME_SSID "SSID 입력" #define HOME_PWD "PWD 입력" ESP8266WebServer server(80); SoftwareSerial ss(D3,D4); MHZ19 mhz(&ss); void setup() { Serial.begin(115200); Serial.println(F("Starting...")); ss.begin(9600); WiFi.begin(HOME_SSID, HOME_PWD); while (WiFi.status() != WL_CONNECTED) { delay(1000); Serial.print("."); } Serial.println(); Serial.print("IP address: "); Serial.println(WiFi.localIP()); server.on("/", check_sensor); server.begin(); Serial.println("Server listening"); } void loop() { server.handleClient(); } void check_sensor() { MHZ19_RESULT response = mhz.retrieveData(); if (response == MHZ19_RESULT_OK) { int sensor = mhz.getCO2(); String co2 = String(sensor); server.send(200, "text/html", "co2: " + co2); } else { Serial.print(F("Error, code: ")); Serial.println(response); } } </pre>	<pre> #include <Arduino.h> #include <IRremoteESP8266.h> #include <IRsend.h> const uint16_t kIrLed = 4; IRsend irsend(kIrLed); void setup() { irsend.begin(); Serial.begin(115200, SERIAL_8N1); } void loop() { long value = 0; if (Serial.available()) { value = Serial.parseInt(); } if (value == 1) { Serial.println("TURN ON"); irsend.sendLG2(0x8800B0B); //ON Serial.println("LG2 Turn On"); } if (value == 2) { Serial.println("TURN OFF"); irsend.sendLG2(0x88C0051); //OFF Serial.println("LG2 Turn Off"); } if (value == 3) { Serial.println("UP"); irsend.sendLG2(0x8808C04); //UP Serial.println("LG2 Up"); } if (value == 4) { Serial.println("DOWN"); irsend.sendLG2(0x8808B03); //DOWN Serial.println("LG2 Down"); } } </pre>

자세한 코드는 https://github.com/EoNjesajo/evsim_TGB에서 확인할 수 있다.

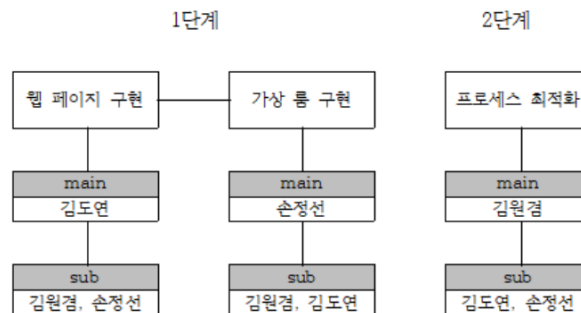
2. 프로젝트 수행을 위해 적용된 추진전략, 수행 방법의 결과를 작성하고, 만일 적용과정에서 문제점이 도출되었다면 그 문제를 분석하고 해결방안을 기술하시오.

- 프로젝트에 대한 이해와 역량 강화

- 해당 프로젝트에 사용될 3D Graphic API, Web Framework에 대한 이해도가 낮음을 파악하고 팀원 각자 맡은 부분에 대해 책, 혹은 교육용 영상을 한 가지 선택하여 공부하는 방식을 택함
- 학습 도중 어려운 부분은 팀원들과 공유하며 이해를 도움

- 프로젝트 관리 체계 수립

- 지난 학기에 수행한 캡스톤 디자인 I에서 예상한 II의 진행 방향과는 사뭇 달라졌으나 관리 체계는 유사한 방법을 택함. 전체적인 프로젝트 수행을 2단계로 나누고, 1단계에서는 웹 페이지 구현과 가상 룸 구현을 동시 진행하며, 2단계에서는 프로세스 최적화 및 수정, 보완을 진행함. 각 단계마다 팀원을 main과 sub로 나누어 효율적으로 역할을 수행.



- 단계별 일정

세부 내용	수행기간(월)				비고
	8	9	10	11	
하드웨어 기능 구현, 프로젝트 이해 및 역량 강화					
(1단계) 가상 룸 그래픽 및 웹 페이지 구현					
(2단계) 프로세스 최적화					
수정 및 보완, 시연 준비					
논문 작성					

[별첨1]

사용자 맞춤형 스마트 홈 서비스

소프트웨어 요구사항 정의서

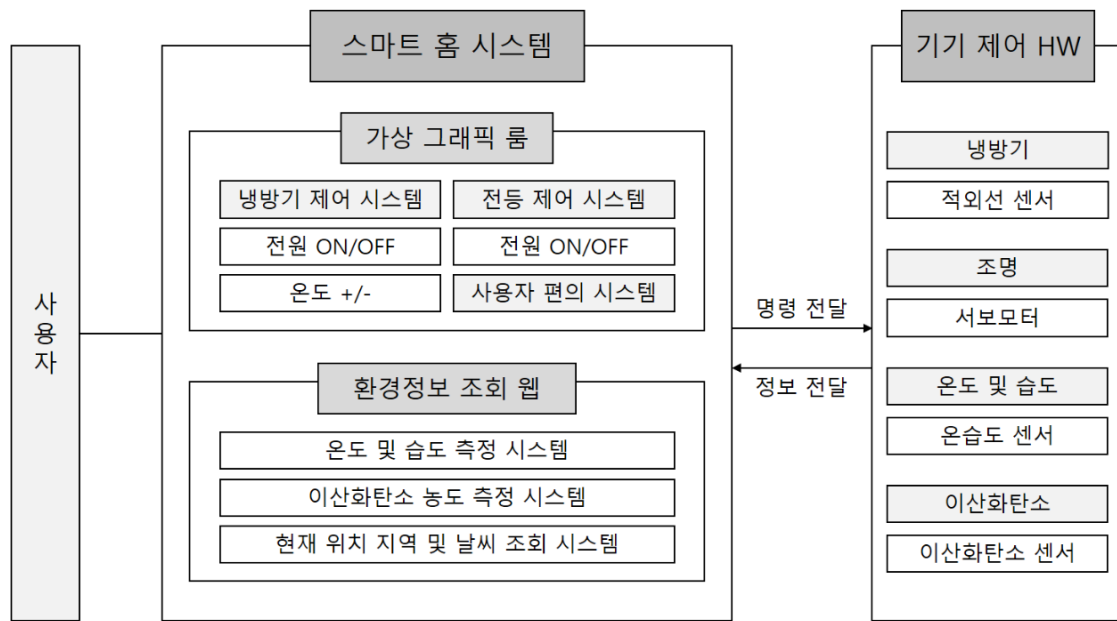
Version 1.0

팀장 김원겸
팀원 김도연
손정선

대표 연락처 : 010-4616-0680
대표 e-mail : 20191768@edu.hanbat.ac.kr

목차		작성자
1	개요	김도연
2	시스템 장비 구성 요구사항	손정선
3	기능 요구사항	김도연
4	성능 요구사항	김원겸
5	인터페이스 요구사항	손정선
6	데이터 요구사항	김원겸
7	테스트 요구사항	김도연
8	보안 요구사항	손정선
9	품질 요구사항	김원겸
10	제약 사항	김도연
11	프로젝트 관리 요구사항	김원겸

1. 시스템 개요



그래픽 API인 OpenGL을 활용하여 가상의 방과 방에 배치될 생활 가전을 구현한다. 그리고 가상의 생활 가전을 현실 세계의 생활 가전과 연결시켜 사용자가 기능을 제어하는 방식으로 스마트홈 서비스를 접목시키고자 한다.

또한 외부에 있는 사용자가 실시간으로 내부의 정보를 확인할 수 있도록 센서를 활용하여 방 안의 온도 및 습도, 이산화탄소 농도와 같은 환경 정보를 조회하는 웹 페이지를 제공한다. Node.JS로 서버를 관리하고 React를 활용하여 구성하고자 한다.

2. 시스템 장비 구성요구사항

요구사항 고유번호		ECR-001
요구사항 명칭		냉방기 제어 장비
요구사항 분류		시스템 장비구성
요구사항 상세 설명	정의	냉방기 제어 장비
	세부 내용	<ul style="list-style-type: none"> - 장비 품목: IR transmitter, IR receiver, NodeMCU - 장비 수량: 각 1개 - 장비 기능: 냉방기 제어를 위한 적외선 센서

요구사항 고유번호		ECR-002
요구사항 명칭		조명 제어 장비
요구사항 분류		시스템 장비구성
요구사항 상세 설명	정의	조명 제어 장비
	세부 내용	<ul style="list-style-type: none"> - 장비 품목: 서보모터, 우노보드 - 장비 수량: 각 1개 - 장비 기능: 조명 제어를 위한 스위치

요구사항 고유번호		ECR-003
요구사항 명칭		온·습도 센서 장비
요구사항 분류		시스템 장비구성
요구사항 상세 설명	정의	온·습도 센서 장비
	세부 내용	<ul style="list-style-type: none"> - 장비 품목: 우노보드, 온도 및 습도 센서 - 장비 수량: 각 1개 - 장비 기능: 온·습도 측정을 위한 센서

요구사항 고유번호		ECR-004
요구사항 명칭		이산화탄소 센서 장비
요구사항 분류		시스템 장비구성
요구사항 상세 설명	정의	이산화탄소 센서 장비
	세부 내용	<ul style="list-style-type: none"> - 장비 품목: 이산화탄소 센서, Wemos 보드 - 장비 수량: 각 1개 - 장비 기능: 이산화탄소 농도 측정을 위한 센서

3. 기능 요구사항

요구사항 고유번호		SFR-001
요구사항 명칭		가전기기 제어
요구사항 분류		기능
요구사항 상세 설명	정의	가전기기 제어 기능
	세부 내용	<p>제어 가능한 가전 기기는 냉방기와 조명으로 제한함</p> <p>① 냉방기 제어 기능</p> <ul style="list-style-type: none"> - 냉방기 전원 ON/OFF - 냉방기 설정 온도 +/- <p>② 조명 제어 기능</p> <ul style="list-style-type: none"> - 조명 ON/OFF
요구사항 고유번호		SFR-002
요구사항 명칭		환경정보 조회
요구사항 분류		기능
요구사항 상세 설명	정의	환경정보 조회 기능
	세부 내용	<p>조회 가능한 환경정보는 온·습도와 이산화탄소로 제한함</p> <p>① 온·습도 조회 기능</p> <ul style="list-style-type: none"> - 온·습도 조회 <p>② 이산화탄소 농도 조회 기능</p> <ul style="list-style-type: none"> - 이산화탄소 농도 조회 - 환기 권유 알림창
요구사항 고유번호		SFR-003
요구사항 명칭		사용자 편의
요구사항 분류		기능
요구사항 상세 설명	정의	이용 안내 및 도움말
	세부 내용	<p>① 시스템 이용 방법 안내</p> <p>② 튜토리얼 및 자주 발생하는 문제에 대한 답변 제공</p>

4. 성능 요구사항

요구사항 고유번호		PER-001
요구사항 명칭		평균 응답시간
요구사항 분류		성능
요구사항 상세 설명	정의	실행 응답시간
	세부 내용	- 프로그램 실행 기능은 사용자가 실행 명령을 입력한지 5초 이내에 실행 결과를 사용자에게 제시해야 함

5. 인터페이스 요구사항

요구사항 고유번호		SIR-001
요구사항 명칭		온라인 도움말
요구사항 분류		인터페이스
요구사항 상세 설명	정의	온라인 도움말 제공
	세부 내용	- 별도의 교육 없이도 사용자 스스로 프로그램을 사용할 수 있도록 하는 온라인 도움말을 제공

요구사항 고유번호		SIR-002
요구사항 명칭		버튼
요구사항 분류		인터페이스
요구사항 상세 설명	정의	버튼 기능
	세부 내용	- 버튼으로 동작을 실행시킬 수 있는 기능 제공 - 기능 요구사항에서 언급한 제어 기능을 버튼을 통해 조작할 수 있게 함

요구사항 고유번호		SIR-003
요구사항 명칭		웹 페이지
요구사항 분류		인터페이스
요구사항 상세 설명	정의	웹 페이지 조회 기능
	세부 내용	- 웹 페이지를 구현한 후 해당 웹 페이지에서 환경 정보를 조회할 수 있게 함

6. 데이터 요구사항

요구사항 고유번호	DAR-001
요구사항 명칭	초기자료 구축
요구사항 분류	데이터
요구사항 세부내용	<ul style="list-style-type: none"> - 사용 기록을 저장하기 위해 데이터베이스와 테이블이 생성되어 있어야 함 - 임계값(예, 냉방기 설정 온도)을 DB에 저장해놓아야 함

요구사항 고유번호	DAR-002
요구사항 명칭	데이터 보존 기간
요구사항 분류	데이터
요구사항 세부내용	<ul style="list-style-type: none"> - 저장된 데이터는 관리자의 제어 없이는 삭제되지 않고 무기한 저장되어야 함

7. 테스트 요구사항

요구사항 고유번호		TER-001
요구사항 명칭		테스트 방안
요구사항 분류		테스트
요구사항 상세 설명	정의	시스템 테스트와 점검
	세부 내용	<ul style="list-style-type: none"> - 시스템은 제공되기로 한 요구사항을 모두 제공하며, 초기 협의한 요구사항에서 변경관리 절차를 통해 승인을 획득한 요구사항을 최종 베이스라인으로 간주함 - 제공되기로 한 요구사항을 제공하는지 여부는 각 기능 요구사항의 검증(테스트) 활동을 통해 예상된 결과가 도출되었을 경우를 기준으로 평가함 - 통합 테스트, 시스템 테스트를 위한 구체적인 틀 및 적용 방안을 제안하여 기능의 오류를 줄일 수 있는 방안을 제시함 - 기능 구현 정확성은 사용자가 직접 테스트 수행 기간에 테스트를 수행함으로써 평가함

요구사항 고유번호		TER-002
요구사항 명칭		통합테스트
요구사항 분류		테스트
요구사항 상세 설명	정의	구체적 통합테스트 방안
	세부 내용	<ul style="list-style-type: none"> - 시스템 내 전체 모듈을 범위로 함 - 순차적 통합 방식을 택해 리스크를 줄임 - 결함 발견시 7일내 수정을 원칙으로 함

요구사항 고유번호		TER-003
요구사항 명칭		시스템테스트
요구사항 분류		테스트
요구사항 상세 설명	정의	구체적 시스템테스트 방안
	세부 내용	<ul style="list-style-type: none"> - 통합테스트 후 수행 - 모든 모듈/구성요소를 통합

8. 보안 요구사항

요구사항 고유번호		SER-001
요구사항 명칭		무선 통신 보안
요구사항 분류		보안
요구사항 상세 설명	정의	WIFI 보안 사항
	세부 내용	- WIFI의 ID와 password 는 기기 연동만을 위해 사용하고 악용되지 않아야 하며 이러한 형태가 발견 될 시 즉시 보고하고 개선하여야 함

9. 품질 요구사항

요구사항 고유번호		QUR-002
요구사항 명칭		어플리케이션 이해성 및 운용성
요구사항 분류		품질
요구사항 상세 설명	정의	품질관리(사용 관점)
	세부 내용	- 사용자는 가상 룸과 웹페이지에 대한 기본적인 사용 방법을 숙지하고 있어야 함 - 구체적인 사용 방법에 대해 가이드 제작하여 제공

10. 제약 사항

요구사항 고유번호	COR-001
요구사항 명칭	시스템 개발 버전
요구사항 분류	제약사항
요구사항 세부내용	<p>다음의 버전에 부합하는 경우에만 동작을 보장함</p> <ul style="list-style-type: none"> - Python v3.8.10, v3.9.9 - Node.JS v16.17.0 - npm v8.15.0 - React v18.2.0 - OpenGL GLFW v3.3.8, GLAD v3.3, GLM v0.9.9.8

요구사항 고유번호	COR-002
요구사항 명칭	필수 하드웨어
요구사항 분류	제약사항
요구사항 세부내용	<ul style="list-style-type: none"> - 가전기기 제어를 위해서는 온습도 센서, CO2 센서, 적외선 센서, 서보모터를 필수적으로 사용

11. 프로젝트 관리 요구사항

요구사항 고유번호		PMR-001
요구사항 명칭		품질 관리
요구사항 분류		프로젝트 관리
요구사항 상세 설명	정의	프로젝트 관리
	세부 내용	- 산출물 및 각종 안내서(시스템 운영자, 사용자 안내서 등)의 관리방안 제 시 - 개발 및 커스터마이징 보고서 제출 - 사업 수행 시 발주자와 상호 협의하여 조정할 수 있음

요구사항 고유번호		PMR-002
요구사항 명칭		프로젝트 수행 조직
요구사항 분류		프로젝트 관리
요구사항 상세 설명	정의	수행조직 구성 및 역할
	세부 내용	- 프로젝트 수행조직은 조장 김원겸, 조원 김도연, 손정선으로 구성 - 2단계로 이루어진 시스템 개발 및 운영 파트를 각각 main, sub로 나누어 관리

요구사항 고유번호		PMR-003
요구사항 명칭		프로젝트 일정
요구사항 분류		프로젝트 관리
요구사항 상세 설명	정의	프로젝트 일정 관리
	세부 내용	<ul style="list-style-type: none"> - 1단계 가상 룸 그래픽 및 웹 페이지 구현은 8월에 시작하여 10월에 완성하는 것을 원칙으로 함 - 2단계 프로세스 최적화는 1단계가 끝난 후 수행하여 10월에 완성하는 것을 원칙으로 함 - 1단계와 2단계가 끝난 후, 수정 및 보완을 하는 것을 원칙으로 함

요구사항 고유번호		PMR-004
요구사항 명칭		운영 지원 인력
요구사항 분류		프로젝트 관리
요구사항 상세 설명	정의	운영 지원 인력 개념 정의
	세부 내용	<ul style="list-style-type: none"> - 업무장애/개선사항이 발생하였을 경우를 대비하여 시스템 구축에 참여한 인력이 운영을 지원