



내가 기획하고 개발한 IT가
세상을 바꾼다면?

한이음 ICT멘토링 프로젝트 결과보고서

프로젝트명

안전성과 청결성을 높인 비 개방형 스마트 화장실

요약본

팀 정보									
팀 명	DP (Dynamic Programming), (Daejin Programers)								
팀구성	성명	소속	부서/학과	직위/학년	팀 사진				
멘 토	박미화	투이컨설팅	P I 팀	수석 컨설턴트					
지도교수									
팀 장	윤상현	대진대학교	컴퓨터공학과	4					
멘티 2	박현식	대진대학교	컴퓨터공학과	4					
멘티 3	엄선오	대진대학교	컴퓨터공학과	4					
멘티 4	최한규	대진대학교	컴퓨터공학과	4					
멘티 5	박현기	대진대학교	컴퓨터공학과	1					
프로젝트 정보									
주제영역	<input type="checkbox"/> 건강	<input type="checkbox"/> 생산성	<input type="checkbox"/> 생활	<input checked="" type="checkbox"/> 안전	<input type="checkbox"/> 엔터테인먼트				
기술분야	<input type="checkbox"/> 이동통신	<input type="checkbox"/> 방송스마트미디어	<input type="checkbox"/> 기반SW컴퓨팅	<input type="checkbox"/> 디지털콘텐츠	<input type="checkbox"/> 융합서비스 <input checked="" type="checkbox"/> 네트워크	<input type="checkbox"/> 정보보호	<input checked="" type="checkbox"/> ICT 디바이스	<input type="checkbox"/> 전파·위성	<input type="checkbox"/> SW
성과목표	<input checked="" type="checkbox"/> 논문게재 및 포스터발표	<input type="checkbox"/> 앱등록	<input type="checkbox"/> 프로그램등록	<input checked="" type="checkbox"/> 특허	<input type="checkbox"/> 기술이전	<input type="checkbox"/> 실용화	<input checked="" type="checkbox"/> 공모전(한이음 액스포)	<input type="checkbox"/> 기타()	
프로젝트명	안전성과 청결성을 높인 비 개방형 스마트 화장실								
프로젝트 소개	<ul style="list-style-type: none"> IoT 기술을 활용하여 안정성과 청결성, 편의성을 높인 공중 화장실 관리 시스템 ✓ 화장실 출입문에 설치한 비콘과 스마트폰 App을 통해 사용자를 인증하여 스마트 화장실의 문을 개방한다. ✓ 사용자는 스마트폰 App을 통해 화장실의 위치조회, 길안내, 사용 중인 인원, 빈칸 정보, 휴지의 양 등 여러 가지 정보를 조회할 수 있다. 								
개발배경 및 필요성	화장실에서 일어나는 범죄 중 강력 범죄 비율은 74%에 이르는데 범죄의 사후 방지책으로는 몇 가지 방안들이 있지만 사전 방지책은 존재하지 않아 안전성을 높이기 위함 익숙하지 않은 지역에서 화장실을 찾기가 불편하며, 찾아 가더라도 청결상태가 좋지 않거나 비품이 없는 문제들을 방지하고자 함								
프로젝트 주요기능	<ul style="list-style-type: none"> 비콘을 통해 사용자가 접근했음을 감지하고 올바른 성별의 화장실 출입문을 개방함 App에서 화장실의 각종 정보들을 확인하여 화장실 상태를 알 수 있음 App에서 화장실의 위치를 지도에서 확인 가능하며, 길 안내 서비스를 받을 수 있음 App을 통해 화장실의 불편 사항을 관리자에게 신고 할 수 있음 화장실의 자판기 물품을 App에서 구매 가능 								
작품의 기대효과 및 활용분야	<ul style="list-style-type: none"> 사용자는 안전하고 편리하며 청결한 화장실을 찾고, 이용 할 수 있음 공공화장실 뿐 아니라 카페, 건물, 회사 등의 화장실에 적용 가능하며, 화장실이 아닌 도서관, 사무실, 호텔 등 출입 관리가 필요한 곳에 적용 가능 								

(본문) 프로젝트 결과보고서

I. 프로젝트 개요

가. 프로젝트 소개

- IoT 기반 비 개방 스마트 화장실
 - IoT 기술을 활용하여 안정성과 청결성, 편의성을 높인 공중 화장실 관리 시스템
 - 비콘을 이용하여 인증된 사용자가 근접할 경우에만 성별에 맞게 출입문을 개방하는 비 개방 화장실
 - 센서를 활용하여 화장실 사용현황과 편의용품 현황을 파악하고 모바일 장치를 통해 화장실 위치와 이용현황을 확인할 수 있는 스마트 화장실
- 기획 의도
 - 개방 화장실의 경우 불특정 다수의 화장실 사용으로 인해 기물 파손, 소모품 부족, 청결상태 불량 등의 문제를 초래하여 사용자에게도 불편함을 안기기 때문에 이를 해결할 방법으로 구상함
 - 남녀화장실이 구분되어 있지만 이성의 화장실에 들어가는데 출입의 제한이 없어 몰래 들어가 옛보거나 찍는 범죄는 증가하는 추세임
 - 범죄의 사후 방지책으로 IoT를 활용하여 소리를 지르면 신고가 가거나 위급상황 버튼을 누르면 신고가 되는 등의 사례가 있지만, 사전 방지책으로는 별달리 제시 된 방안들이 없어 사전 방지책으로 구상하게 됨
- 작품 내용
 - 화장실 출입문에 설치한 비콘과 스마트폰 App을 통해 사용자를 인증하고, 허가된 사용자만 성별에 맞는 화장실의 출입을 허가함
 - 사용자는 스마트폰 App을 통해 화장실의 위치조회, 길안내, 사용 중인 인원, 빈칸 정보, 비치된 물품 정보 등 여러 가지 정보를 조회할 수 있음
 - 사용자는 스마트폰 App을 통해 사용하고 있는 화장실에 대한 불편사항을 신고할 수 있으며, 화장실 관리자는 센서를 통해 확보한 화장실 상황정보와 신고 된 불편사항 정보를 기반으로 화장실을 관리함

나. 개발배경 및 필요성

○ 제작 동기

(1) 공중 화장실 범죄 예방의 필요성

- 경찰청 자료에 따르면 2014년 공중 화장실에서 발생한 범죄는 모두 1,795건,

이 가운데 살인과 강도, 성범죄 같은 강력 범죄 비율은 74%에 이르고 있음

- 남녀가 분리된 화장실에도 성별이 다른 사람이 출입할 수 없도록 출입 제어를 할 수 있는 시스템이 필요함
- 현행법상 일정 규모 이상의 건물은 남녀 화장실을 반드시 구분해야 하지만, 2004년 법 시행 전에 지어졌거나 규모가 작은 건물은 남녀 공용화장실이 대부분임

(2) 청결하지 않고 불편한 화장실

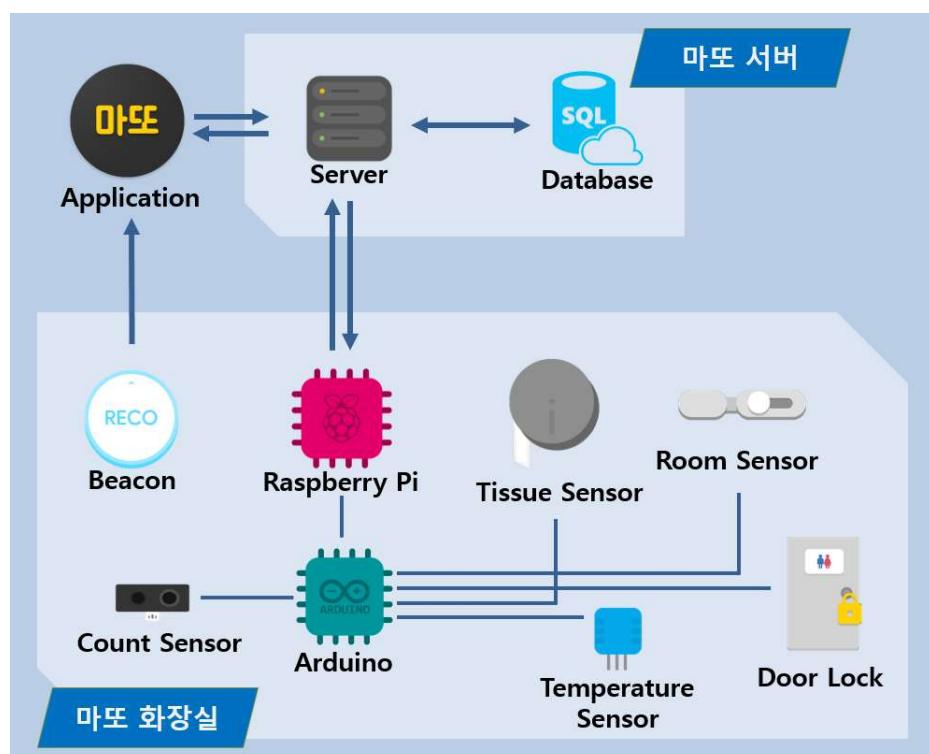
- 불특정 다수의 사용으로 화장실 사용량이 높을 경우 휴지나 용품이 없거나 청결하지 않은 경우가 종종 발생함
- 화장실이 급해서 방문했지만 사용 중인 경우 많이 기다려야 함

○ 목적

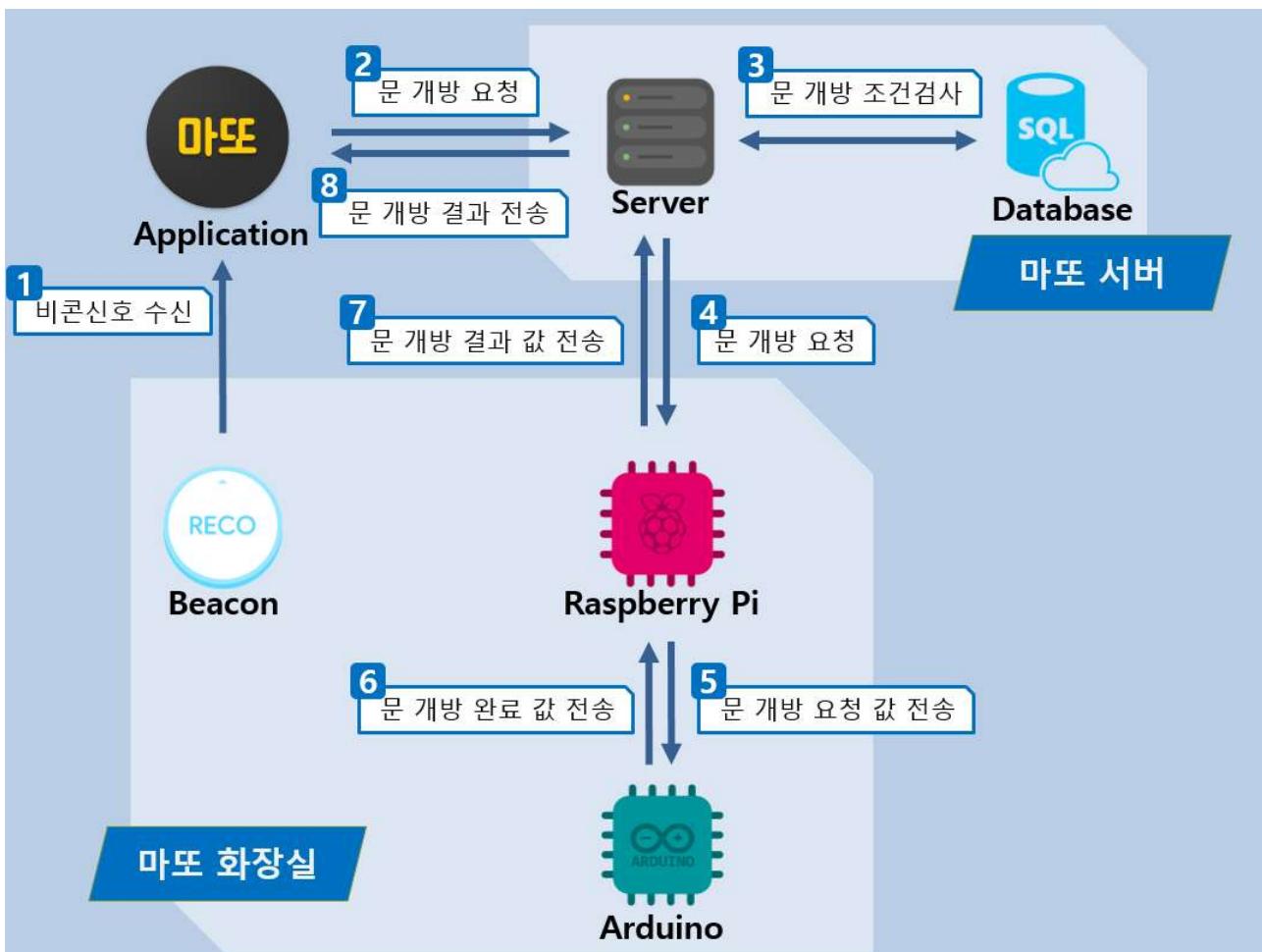
- 강남역 화장실 사건과 같은 화장실의 취약한 보안 문제를 해결 할 방법을 찾고자 함
- 스마트화로 App을 통해 화장실 사용자, 관리자 모두에게 화장실 정보를 제공하고 여러 편의사항들을 제공
- 비 개방 화장실의 구현으로 사용량 감소 효과와 관리의 용이성 확보

다. 작품 구성도

○ 시스템 구성도

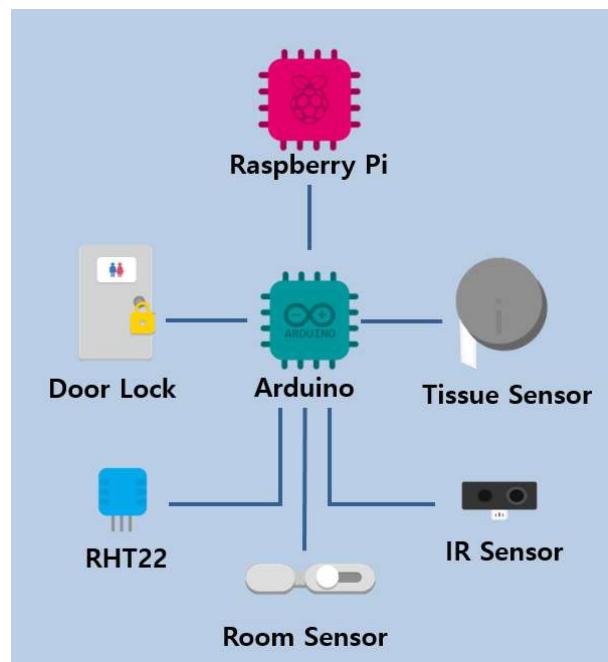


○ S/W구성도



- 1 Beacon신호 범위 내에 위치한 사용자의 Phone이 Beacon신호를 수신한다.
- 2 Server는 사용자정보와 Phone이 가지고있는 Beacon신호 값을 수신한다.
- 3 Server는 수신한 정보를 Database에 저장 된 정보와 대조하여 신원을 확인하고, 어떤 화장실에 대한 요청인지 확인한다.
- 4 Server는 확인 된 화장실로 문 개방을 요청한다.
- 5 문 개방 요청을 받은 화장실의 Raspberry Pi가 문을 개방한다.
- 6 Arduino는 문 개방 후 Raspberry Pi에게 그 결과를 전송한다.
- 7 Server는 Raspberry Pi로부터 문 개방 결과를 수신한다.
- 8 Server는 사용자의 Phone에 문 개방 결과를 전송한다.

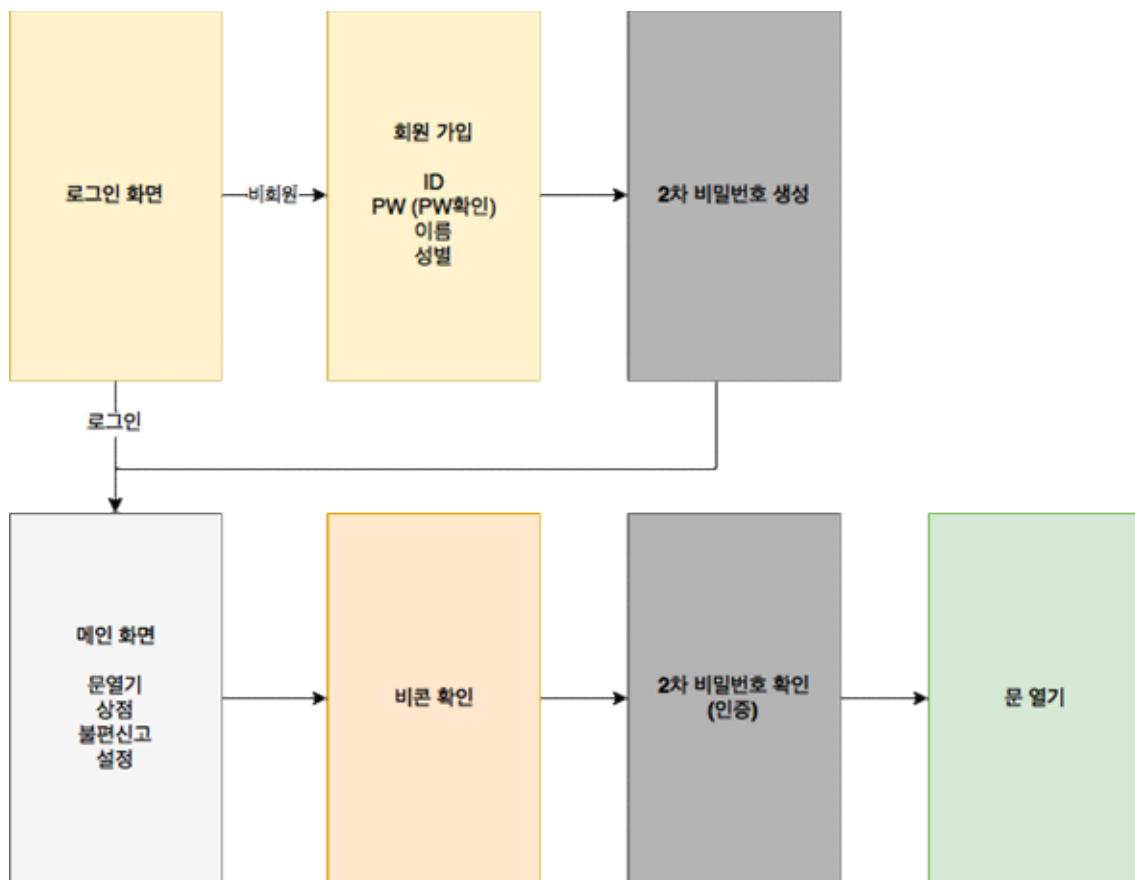
○ H/W구성도



○ 서비스 흐름도



○ 로그인 및 출입인증 흐름도



II. 프로젝트 수행결과

가. 주요기능

구분	기능	설명
S/W	회원 가입	App을 통한 회원 가입 기능
	로그인	App을 통한 로그인 기능
	로그아웃	App을 통한 로그아웃 기능
	회원 정보 수정	App을 통한 회원 정보 수정 기능
	회원 정보 조회	App을 통해 자신의 회원 정보 조회 기능
	비콘 인식 및 판별	비콘 신호를 인식한 출입 관리 기능
	화장실 찾기 기능	화장실 위치를 지도에 표시한다.
	화장실 길 안내 서비스	화장실까지의 길 안내 서비스를 제공한다.
	화장실 정보 조회	화장실의 인원 및 각종 정보들을 표시한다.
	화장실 내 물품 조회	App을 통해 화장실 내 물품을 조회할 수 있다.
H/W	화장실 내 물품 구입	App을 통해 화장실의 자판기 물품구매를 할 수 있다.
	불편 신고	불편 사항이 있을 경우 신고한다.
	화장실 출입문 잠금 제어	화장실 출입문의 잠금장치를 제어한다.
	출입 인원수 카운트	적외선 센서로 출입 인원을 카운팅한다.
	화장실의 빈칸 확인	화장실 내의 빈 칸을 센싱한다.
	온도 습도 측정	화장실의 온도와 습도를 측정한다.
	부족한 휴지 감지	휴지가 일정량 이하가 되는지 센싱한다.
	물품 구매 기능	App을 통해 자판기에서 물품을 구매 한다.

나. 프로젝트 개발환경

구분		상세내용
S/W 개발환경	OS	Windows 8.1 K 64bit, Linux Ubuntu 12.04.5 LTS, Mac OS X El Capitan 10.11.6
	개발환경	Android 4.3 (Jelly Bean), Express.js
	개발도구(IDE)	Android Studio 2.1.2, Eclipse Neon, Notepad++
	개발언어	JAVA, Node.js
	기타사항	XShell5(SSH), Filezilla(FTP), Postman
H/W 구성장비	디바이스	Arduino Leonardo, Raspberry Pi 3
	센서	적외선 센서, 온습도 센서
	통신	HTTP, Serial
	개발도구(IDE)	Arduino Sketches, Atom
	개발언어	C
	기타사항	Putty(SSH), WinSCP
프로젝트 관리환경	형상관리	Git, FlyWay
	이슈관리	Trello
	일정관리	Trello
	의사소통관리	Slack
	기타사항	Google Docs

다. 장비(기자재/재료) 활용

번호	품명	작품에서의 주요기능
1	스마트폰	- Application을 실행하고 비콘으로부터 신호를 인식하여 사용자에게 서비스를 제공한다.
2	클라우드 서버	- DB와 메인 서버를 구성하며, 각 화장실 서버인 라즈베리와 통신한다.
3	라즈베리파이	- 각 화장실에 설치되어 화장실 서버의 역할을 한다.
4	아두이노	- 라즈베리파이와 연결되어 화장실의 하드웨어를 제어한다.
5	비콘	- 화장실에 설치되어 지속적으로 신호를 발산하여 사용자가 화장실에 접근했음을 판단 할 수 있다.

라. 프로그램 작동 동영상

- <https://www.youtube.com/watch?v=twX8YfiCI48>

마. 상세 이미지



바. 도출성과

■ 논문게재 및 포스터발표	제자(발표)자명	논문(포스터)명	제자(발표)처	제자(발표)일자
	윤상현 외 5명	인전성과 청결성을 높인 IoT 기반 스마트 화장실 구축 방법	한국정보과학회	2016. 12. 21.
□ 앱(APP) 등록	등록자명	앱(APP)명	등록처	등록일자
□ 프로그램 등록	등록자명	프로그램명	등록처	등록일자
■ 특허/실용신안 출원	출원자명	특허/실용신안명	출원번호	출원일자
	윤상현 외 5명	IoT기반 비개방 스마트 화장실 관리 시스템	10-2016-0160841	2016. 11. 29.
□ 기술이전	기술이전기업명	기술명	금액	이전일자
■ 공모전	구분(교내/대외)	공모전명	수상여부(출품/수상)	상격
	대외	한이음 엑스포	수상	입선
□ 실용화				
□ 기타				

III. 프로젝트 수행방법

가. 업무분장

번호	성명	역할	담당업무
1	박미화	멘 토	프로젝트 지도
2		지도교수	
3	윤상현	팀 장	- HW 개발
4	박현식	팀 원2	- Server 개발
5	엄선오	팀 원3	- App 개발
6	최한규	팀 원4	- Server 개발
7	박현기	팀 원5	- Design

나. 프로젝트 수행일정

구분	추진내용	수행일정							
		5월	6월	7월	8월	9월	10월	11월	12월
계획	프로젝트 수행 계획								
분석	적용 기술 조사, 타당성 검토								
설계	서버 설계 (RESTful API 및 DB설계)								
	레이아웃 및 UI/UX 설계								
	H/W 설계								
	통신 프로토콜 설계								
개발	DB구축 및 웹서버 개발								
	App개발 및 비콘 통신 구현								
	H/W 및 라즈베리파이 환경 구축								
테스트	최종 테스트 및 유지보수								
종료	공모전 출전								
	프로젝트 종료								
오프라인 미팅	매월 최소 1회 오프라인 미팅								

다. 문제점 및 해결방안

- 프로젝트 관리 측면
 - 회의 장소와 일정
 - 한이음에서 지원하는 토즈를 이용하기 위해 거리가 먼 팀원의 장시간 이동에 따른 불편한 문제가 발생하여 팀원들 거주지의 중심을 분석하여 회의 장소를

결정함

- 팀원 모두와 멘토님이 같이 모이는 시간이 제한적인 경우가 많아, 팀 전체가 아닌 팀원 일부만으로도 회의를 진행

- 팀원의 교체

- 디자인을 담당하던 팀원의 개인사정으로 팀에서 나가게 됐고, 갑작스런 디자이너의 부재로 새롭게 디자이너를 그동안 진행되어왔던 디자인 업무를 새로 들어온 팀원이 재작업을 하게 됨

- 형상 관리

- 역할 분담에 따라 개발을 진행하고 합치는 방식으로 진행하였는데, 동시에 수정하기가 적합하지 않아 형상 관리가 필요하다 생각되어 Git을 통해 산출물을 공유함
- 개발 시 매번 모여 개발 할 수 없으므로 각자 개발을 한 후 협업 툴과 형상 관리 툴을 이용하여 진행사항을 공유하고 의견을 나눔
- 각종 문서들을 Google Docs를 통해 동시 편집으로 작업하고 형상관리 진행

- 작품 개발 측면

- 하드웨어

- 모든 하드웨어를 최소한의 재료로 직접 제작하였기 때문에 기술적인 측면 이외에도 작품의 모형을 만들기 위한 구조적 문제점들이 많이 발생함에 따라 구조적인 설계를 하여 모형을 제작함
- 각 부품들이 모듈별로는 정상적으로 동작하지만 통합 과정에서 코드적인 문제와, 전기적은 문제로 고전하였으나, 검색이나 지식을 가진 지인을 통해 해결책을 찾고 해결함
- 비콘 신호를 통해 측량된 정보를 이용하여 작동되는 기술의 오차가 발생했고, 이를 보정하기 위해 2개의 비콘을 사용한 알고리즘으로 위치 정보를 측정함
- 서버 통신을 JSON으로 사용함으로써, 호환성을 높이기 위해 아두이노에도 JSON을 사용하기 위해 직접 구현하려 했으나, 많은 예외상황들이 있고 추가 확장 가능성과 생산성을 위해 JSON라이브러리를 사용함

- 서버 개발

- 비동기 처리가 주 된 Node.js에서 로직 흐름 제어에 장애가 많아, 흐름 제어 모듈을 통해 해결함
- 사용자 인증 처리 방식을 세션에 유지하는 방법으로 적용했지만 서버

스트레스 테스트에서 메모리의 한계를 느껴 액세스 토큰 기반의 사용자 인증 방식으로 수정함

IV. 기대효과 및 활용분야

1. 작품의 기대효과

- 안정성 측면
 - 성별에 일치하는 화장실의 문만 열리기 때문에 이성의 화장실에 들어가는 경우가 없어져 성 범죄 예방에 효과적
- 청결성 측면
 - 관계없는 사용자들의 출입을 제한하여 화장실 사용량을 줄여 기존 화장실보다 청결성의 유지 시간을 연장함
- 편의성 측면
 - 사용자와 관리자에게 화장실에 대한 정보를 보다 편하게 제공하므로 사용자는 화장실 이용의 편리성을 더하고 관리자는 좀 더 신속, 정확하게 화장실의 관리가 가능함
 - 비콘을 사용하여 문을 개방하기 때문에 처음 가는 화장실이더라도 비밀번호가 필요하지 않으며 App만을 통해 개방하므로 편리성 강화
 - 화장실의 빈칸을 확인할 수 있어 화장실 앞에서 줄을 서서 기다릴 필요가 없어짐
- 시장성 측면
 - 2004년 법규 이전 건축물의 경우 이 시스템을 도입하여 기존의 남녀공용 화장실을 그대로 사용하지만, 출입관리가 가능한 화장실로 탈바꿈하여 새로 공사를 하지 않더라도 안정성을 확보할 수 있으므로 경제적, 시장성이 좋음
 - 빅 데이터를 활용한다면 관리자는 스마트 화장실의 비품들의 수요를 통계적으로 파악할 수 있을 것이고, 보다 효율적으로 물품관리를 하여, 경제적인 측면에서의 좋음
 - 스마트 화장실이 많아지면 App사용자들의 선택의 폭이 넓어지며, 사용자가 많아질수록 표본 데이터 축적이 가능하며 추출된 표본 데이터로 화장실 사용자들의 수요 패턴 등을 분석할 수 있고, 그로인해 경제적 이익을 얻어낼 수 있음

- 빅 데이터 분석을 통해 적절한 위치에 스마트 화장실 설치를 하여 특정 지역의 화장실 설치 부분에 대한 예산을 효율적으로 사용할 수 있을 것이므로 App의 시장성은 더욱더 커질 수 있음

○ 사회/문화적 효과

- 화장실을 청결하고 안전하게 함으로써 사용자들에게 공중화장실 이용에 대한 거부감을 없애 공중화장실 사용을 유도하고 내국인은 물론 외국인에게 화장실에 대한 이미지를 긍정적으로 유도
- 각 지역에 특색에 맞는 스마트 화장실을 구축하여, 특정 지역에만 존재하는 스마트 화장실 상품을 구매하거나, 스마트 화장실을 이용하기 위해 특정 장소에 방문욕구를 증대시킴
- 빅 데이터를 활용하여 화장실 사용량이 높은 지역의 경우 주변에 추가 화장실 설치 필요성 여부를 판단할 수 있으며 기물들의 점검 주기를 통계적으로 분석 가능
- 스마트 화장실의 이용 증대를 통하여, 외국인들에게 대한민국은 IT 강국이라는 인식 강화

○ 스마트 화장실의 IoT 기술 발전에 기여

- 현재 스마트 화장실에 존재하지 않는 비콘 기반으로 출입 통제 시스템을 구축하여 새로운 가능성을 제시

2. 작품의 활용분야

○ 스마트 화장실 적용 확대

- 공공화장실을 전제로 하였지만 유렵의 1인용 공용 화장실과 같은 곳에 적용 가능
- 공공개방을 해야만 하는 화장실이 아닌 상가, 건물, 회사 등 특정 사용자만을 위한 화장실의 출입 제어를 함으로써 화장실의 확실한 남/여 출입 구분을 실현

○ 비콘을 이용한 출입 관리 시스템의 활용분야

- 현재 프로젝트는 화장실을 주제로 하였지만 화장실이 아닌 기업 출입 시스템이나 개인 사유지 등 출입 관리가 필요한 도서관, 사무실, 호텔 등에 비콘을 통한 출입 관리 시스템 적용 가능
- 카페의 화장실은 도어락을 통해 비밀번호로 유지되는 경우가 많은데 NFC 태깅을 통해 사용자의 핸드폰을 인증장비로 사용하여, 일정 시간동안 접근

권한을 주는 방식으로 활용 가능

- 적외선 센서의 활용분야
 - 현재 프로젝트는 화장실 내 휴지의 크기를 이용한 휴지 부족 알림으로 사용하고 있지만, 화장실 휴지통이 가득 찼을 때 알릴 수 있는 용도로 활용 가능
 - 화장실 사용자를 감지하여 자동으로 변기의 물을 내려주고 자동으로 음악을 틀거나, 광고를 하는 등의 활용도 가능

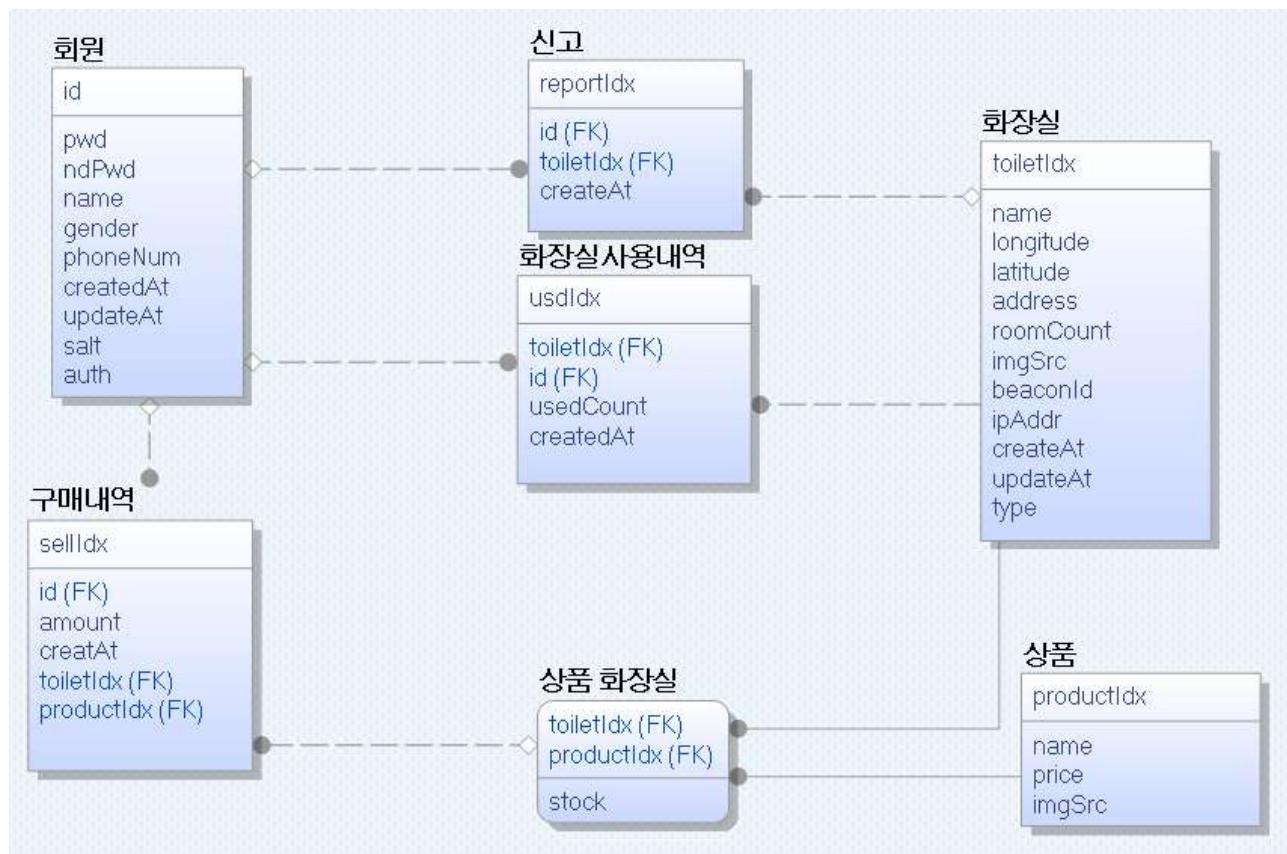
V. 참고자료

가. 참고 및 인용자료

- 변태우, 비콘 기반 실내측위 측정 오차 요인 심층 분석, 학위논문(석사), 2016년

[별첨] 한이음 ICT멘토링 프로젝트 산출물

o DataBase 관계도



○ App 화면별 기능 명세

App 화면	상세 설명	
	기능 명	로그인
	기능설명	<p>■ MATTO App 첫 화면</p> <p>기존 MATTO App의 사용자는 회원가입한 계정으로 로그인을 할 수 있다. 신규회원은 회원가입 하기(card view)를 통해 회원가입 후 로그인 할 수 있다.</p>
	처리내용	<p>■ MATTO App 기존회원의 로그인</p> <p>사용자가 회원가입 했던 계정으로 로그인을 할 수 있다. ID(Text hint : "이메일")와 PASSWORD(Text hint:"비밀번호")를 입력후 로그인(card view)를 누르면 로그인 할 수 있다.</p> <p>■ MATTO App 신규 회원의 로그인</p> <p>회원가입 하기(card view)를 누르고 id, pwd, pwdconfirm, gender, PhoneNum 정보를 입력하고, 2차 비밀번호 등록화면으로 넘어 간 뒤 ndPwd를 등록 후 회원가입 되며, 그 계정으로 로그인을 할 수 있다.</p>
	비고	■연결되는 기능 : 서버 연동(TOKEN)
	요구사항 명	서버 연동, 회원가입

App 화면	상세 설명	
	기능 명	회원가입
	기능설명	<p>■ MATTO App 회원가입</p> <p>기존 MATTO App의 사용자가 아닌 신규 사용자인 경우 회원가입을 통해 MATTO App를 사용할 수 있다.</p>
	처리내용	<p>■ MATTO App 회원 가입</p> <p>MATTO App 시작화면(로그인 화면)에서 회원가입 하기(card view)를 누르고 id,pwd,pwdconfirm,gender,phoneNum 정보를 입력 후, 2차 비밀번호 등록화면으로 넘어 간 뒤 ndPwd를 등록 후 회원가입 되며, 그 계정으로 로그인을 할 수 있다.</p>
	비고	■연결되는 기능 : 서버 연동(POST), 로그인
	요구사항 명	서버 연동

App 화면		상세 설명
SKT	71% 오후 11:43	기능 명 2차 비밀번호 등록
2차 비밀번호 2차 비밀번호를 입력해 주십시오	기능설명 ■ MATTO App 2차 비밀번호 등록 회원가입화면에서의 정보들을 모두 입력 후, 가입하기를 누르면 2차 비밀번호 등록을 할 수 있다. 2차 비밀번호 등록하면 회원가입을 성공적으로 마칠 수 있다.	
— — — — 1 2 3 4 5 6 7 8 9 ✓ 0 ✕	처리내용 ■ MATTO App 2차 비밀번호 등록 성공 회원가입 화면에서 가입하기를 누른 후 2차 비밀번호를 등록한다. 첫 2차 비밀번호 등록화면에서는 사용할 2차 비밀번호(4자리 숫자)를 입력후 체크모양을 누른 후에, 등록한 번호를 다시 한 번 입력하면 성공적으로 회원가입이 완료된다. ■ MATTO App 2차 비밀번호 등록 실패 회원가입 정보를 입력 후 2차 비밀번호를 등록하는데, 이 때 등록한 2차 비밀번호와 등록후 확인시에 들어가는 2차비밀번호가 다르면 2차 비밀번호 등록에 실패하는데 이 때, toast 메시지를 띄워 사용자에게 재등록을 요구한다.	
비고 : 서버 연동(POST), 회원 가입	■연결되는 기능 : 서버 연동(POST), 회원 가입	
요구사항 명	서버 연동, 회원가입	

App 화면		상세 설명
SKT ≡ MATTO 어서오세요. 안녕하세요 마려우면 도오리 마또(MATTO)입니다.	기능 명 메인 화면	
가장 가까이 있는 화장실 지도보기 50 M 10 간 중 1간 사용중...	기능설명 ■ MATTO App 메인 화면 MATTO App 사용자는 로그인 화면에서 로그인 후 메인 화면으로 올 수 있다. 주변에 비콘이 인식되지 않아 화장실이 없으면 문 열기 버튼과 물품 구매 버튼은 비활성화된다.	
문 열기 물품 구매 불편신고 설정	처리내용 ■ MATTO App 메인 화면 메인화면에는 회원정보 관리(원쪽 상단 버튼),지도보기, 문 열기, 물품 구매, 불편 신고, 설정 등을 할 수 있다.	
불편신고 설정	비고 : 서버 연동, 로그인, 문 열기, 물품 구매, 불편 신고, 설정, 지도	
요구사항 명	서버 연동, 로그인	

App 화면		상세 설명	
		기능 명	문 열기
		기능설명	■ MATTO App 문 열기 MATTO App 메인 화면에서 문 열기를 누르면 2차 비밀번호로 인증 후에 스마트 화장실 문을 열 수 있다.
		처리내용	■ MATTO App 문 열기 메인 화면에서 문 열기 버튼을 클릭 후 2차 비밀번호를 입력하여 인증을 마친다. 인증된 사용자에게는 "문이 열렸습니다."라는 문구의 toast message가 뜬다. 2차 비밀번호 인증에 실패한 사용자는 다시 메인 화면으로 돌아간다.
		비고	■ 연결되는 기능 : 서버 연동, 비콘 연동
		요구사항 명	서버 연동, 하드웨어 연동, 로그인

App 화면		상세 설명	
		기능 명	물품 구매
		기능설명	■ MATTO App 물품 구매 MATTO App을 통해 화장실 내 자판기의 물품 구매가 가능하다.
		처리내용	■ MATTO App 물품 구매 로그인한 사용자는 물품 구매를 누른 후 자판기의 물품들을 App을 통해 구매할 수 있다. 물품 구매를 누르면 구매할 수 있는 물품 목록이 나열되며 구매버튼을 누르면 물품 구매가 완료된다.
		비고	■ 연결되는 기능 : 서버 연동, 하드웨어 연동
		요구사항 명	서버 연동, 하드웨어 연동, 로그인

App 화면	상세 설명	
	기능 명	불편 사항 신고
	기능설명	<p>■ MATTO App 불편 사항 신고</p> <p>스마트 화장실 사용자는 메인 화면에서 불편 신고버튼을 누른 후 불편한 점이나, 요구 사항 등을 입력하여 관리자에게 문의할 수 있다.</p>
	처리내용	<p>■ MATTO App 불편 사항 신고</p> <p>로그인한 사용자는 메인 화면의 불편 신고를 누르면 text형식으로 문의사항을 입력하여 스마트 화장실 관리자에게 문의할 수 있다.</p>
	비고	<p>■연결되는 기능 : 서버 연동(POST), 로그인</p>
	요구사항 명	서버 연동, 로그인

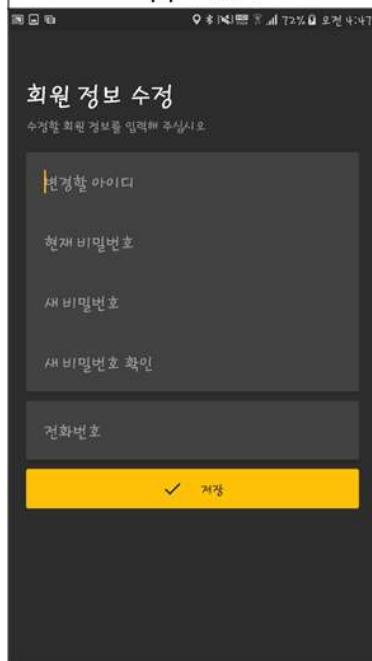
App 화면	상세 설명	
	기능 명	지도
	기능설명	<p>■ MATTO App 지도</p> <p>스마트 화장실로 등록된 화장실의 위치를 마커로 찍어 구글 지도상으로 보여준다.</p>
	처리내용	<p>■ MATTO App 지도</p> <p>메인 화면에서 지도보기를 누르면 스마트 화장실로 등록된 화장실의 위치를 볼 수 있다. Marker를 클릭하면 정확한 위도와 경도값을 알 수 있다.</p>
	비고	<p>■연결되는 기능 : 서버 연동(GET), 길찾기</p>
	요구사항 명	서버 연동, 로그인

App 화면		상세 설명	
	기능 명	길 찾기	
	기능설명	■ MATTO App 길 찾기 스마트 화장실로 등록된 화장실의 위치로 길 찾기가 가능하다..	
	처리내용	■ MATTO App 길 찾기 스마트 화장실로 등록된 화장실의 위치정보와 현재 나의 위치정보를 토대로 길 찾기 기능이 가능하다. 지도에 오른쪽 상단의 버튼을 눌러 내 위치를 확인하고 해당 화장실의 위치정보를 입력하거나 지도상에서 Marker를 클릭하여 편리한 길 찾기가 가능하다.	
	비고	■연결되는 기능 : 서버 연동, 지도	
	요구사항 명	서버 연동, 로그인	

App 화면		상세 설명	
	기능 명	설정	
	기능설명	■ MATTO App 설정 MATTO App을 보다 편리하게 사용하기 위하여 알림, 자동 로그인의 사용 여부를 선택 할 수 있다.	
	처리내용	■ MATTO App 설정 알림 기능의 On/Off와 자동 로그인 기능의 On/Off 기능을 지원한다.	
	비고	■연결되는 기능 : 서버 연동	
	요구사항 명	서버 연동, 로그인	

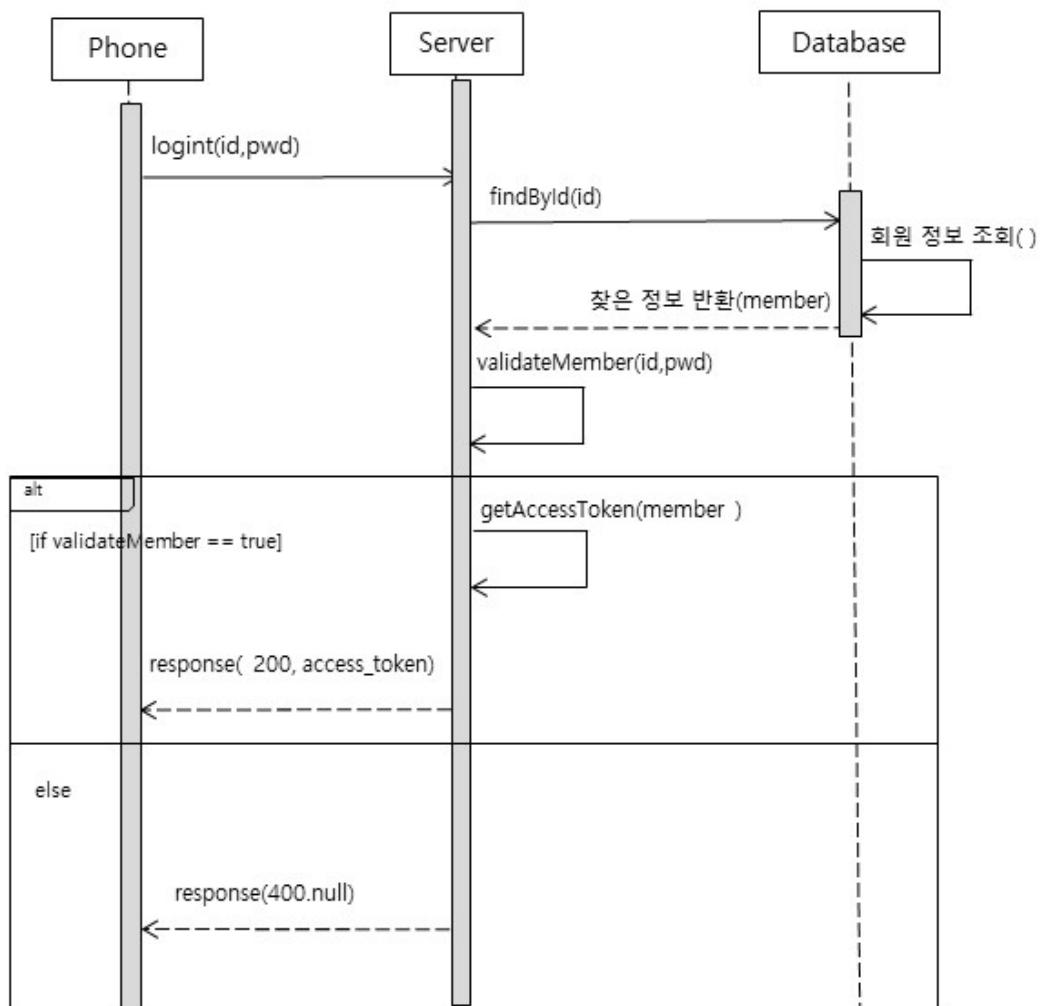
App 화면	상세 설명	
SKT 86% 오후 2:50 [공지] 공지 테스트 공지 테스트입니다. 16/10/30	기능 명	공지 사항
[공지] 공지 테스트 공지 테스트입니다. 16/10/30	기능설명	<p>■ MATTO App 공지사항</p> <p>MATTO App의 업데이트 소식이나 기타 공지 사항을 알려 주는 기능이다.</p>
[공지] 공지 테스트 공지 테스트입니다. 16/10/30	처리내용	<p>■ MATTO App 공지 사항</p> <p>메인화면에서 원쪽 상단의 버튼을 누른후 공지사항버튼을 누르면 업데이트 소식이나 기타 공지 사항을 확인할 수 있다.</p>
[공지] 공지 테스트 공지 테스트입니다. 16/10/30	비고	<p>■연결되는 기능</p> <p>: 서버 연동(GET)</p>
[공지] 공지 테스트 공지 테스트입니다. 16/10/30	요구사항 명	서버 연동, 로그인

App 화면	상세 설명	
SKT 72% 오후 5:46 MATTO 참가 멤버 박미화 멘토님 윤상현 최한규 박현식 엄선오 박현기 불편신고 설정	기능 명	팀 소개
	기능설명	<p>■ MATTO App 팀 소개</p> <p>MATTO App 개발팀에 대한 팀 소개 기능이다.</p>
	처리내용	<p>■ MATTO App 참가 멤버</p> <p>개발에 참여한 멤버를 소개하는 페이지이다.</p>
	비고	<p>■연결되는 기능</p> <p>: 서버 연동</p>
	요구사항 명	서버 연동, 로그인

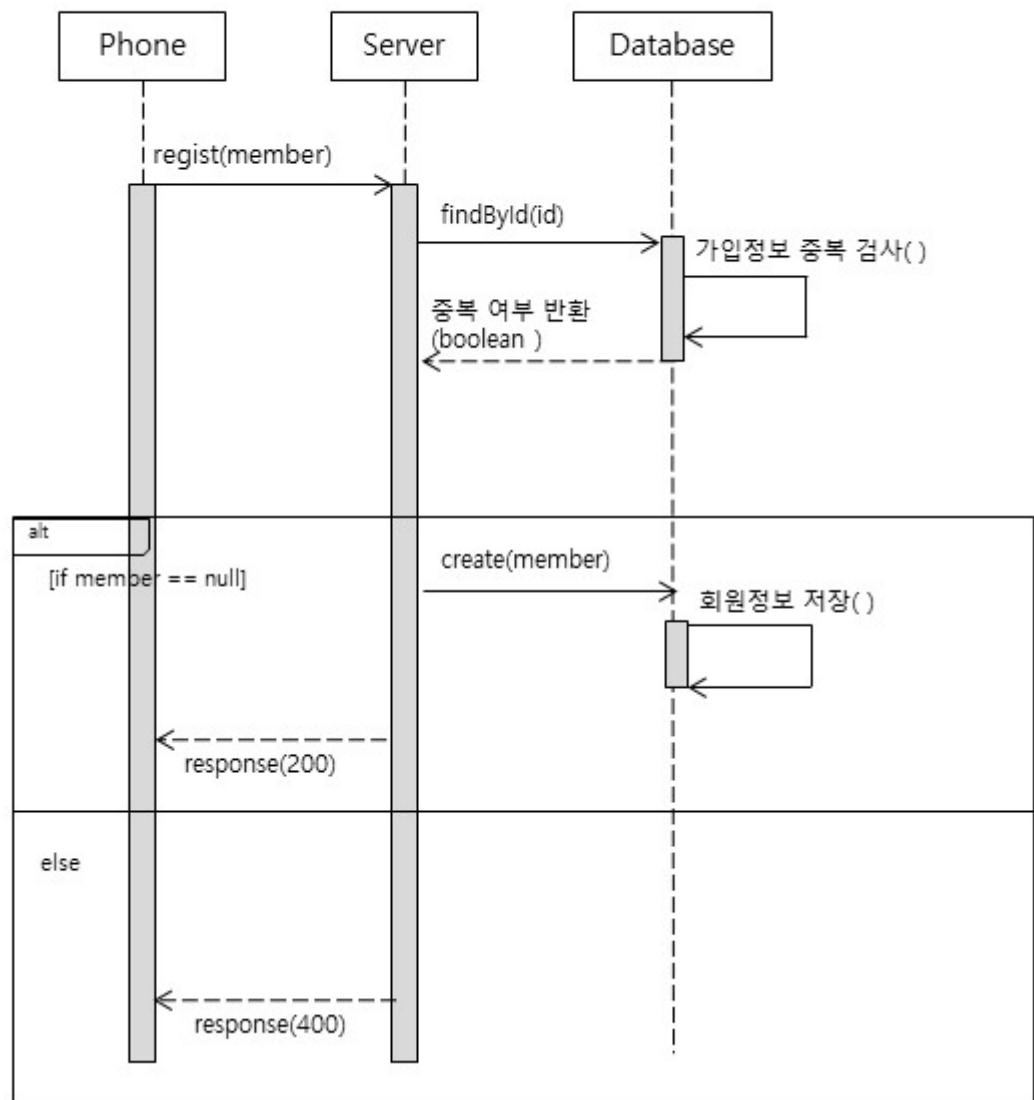
App 화면	상세 설명	
	기능 명	회원 정보 수정
	기능설명	<p>■ MATTO App 회원 정보 수정</p> <p>MATTO App의 회원은 회원 정보 수정 기능을 통해 회원 가입했을 때의 정보를 변경할 수 있다.</p>
	처리내용	<p>■ MATTO App 회원 정보 수정</p> <p>메인화면에서 왼쪽 상단의 버튼을 누른 후 회원 정보 수정을 누르면 로그인한 사용자의 정보를 수정할 수 있다.</p>
	비고	<p>■ 연결되는 기능 : 서버 연동(PUT)</p>
	요구사항 명	서버 연동, 로그인

○ 기능별 시퀀스 다이어그램

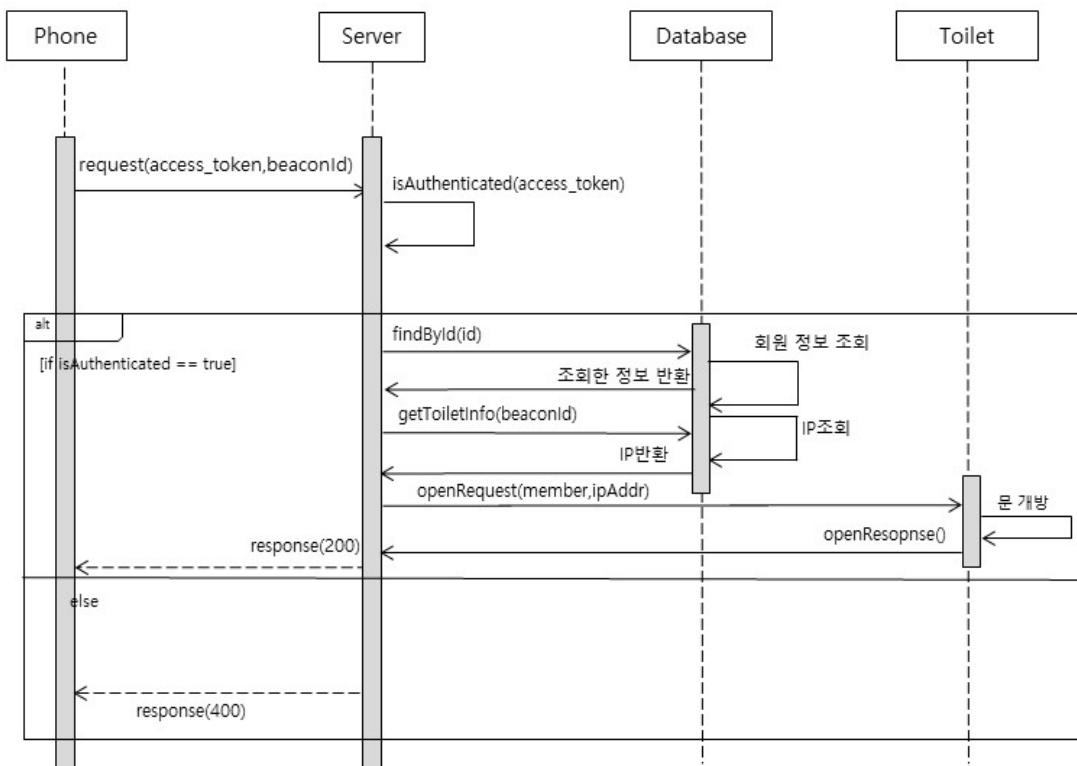
로그인 토큰 획득



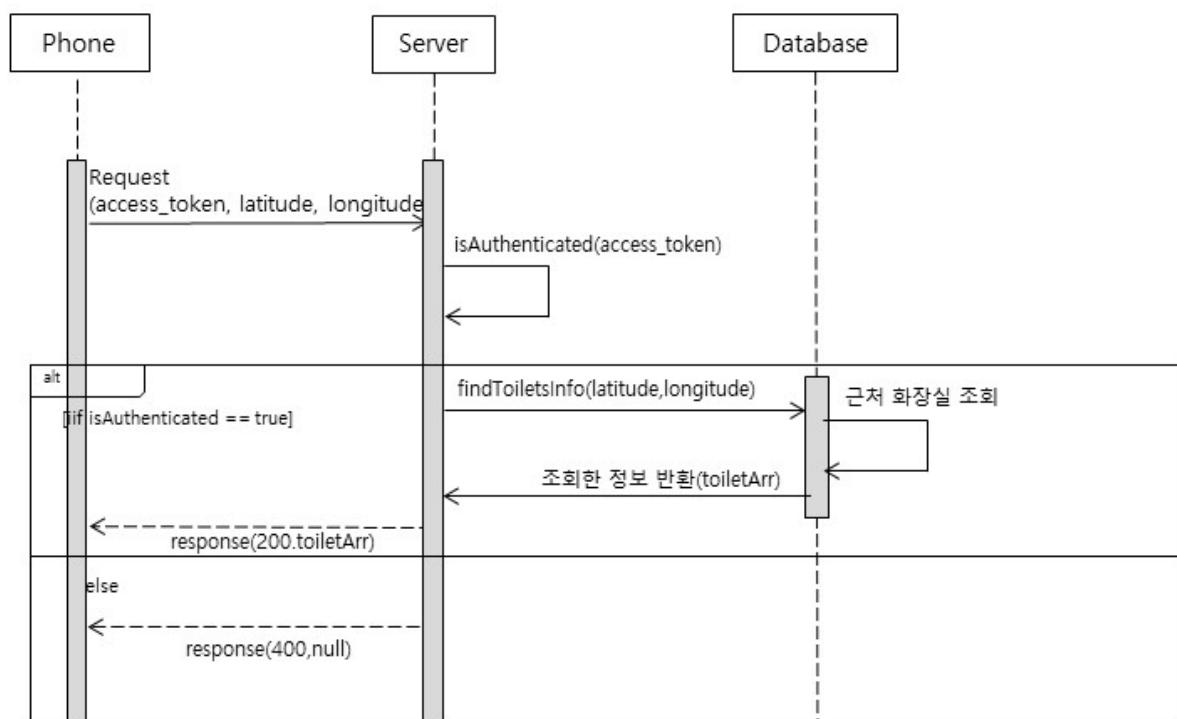
회원가입



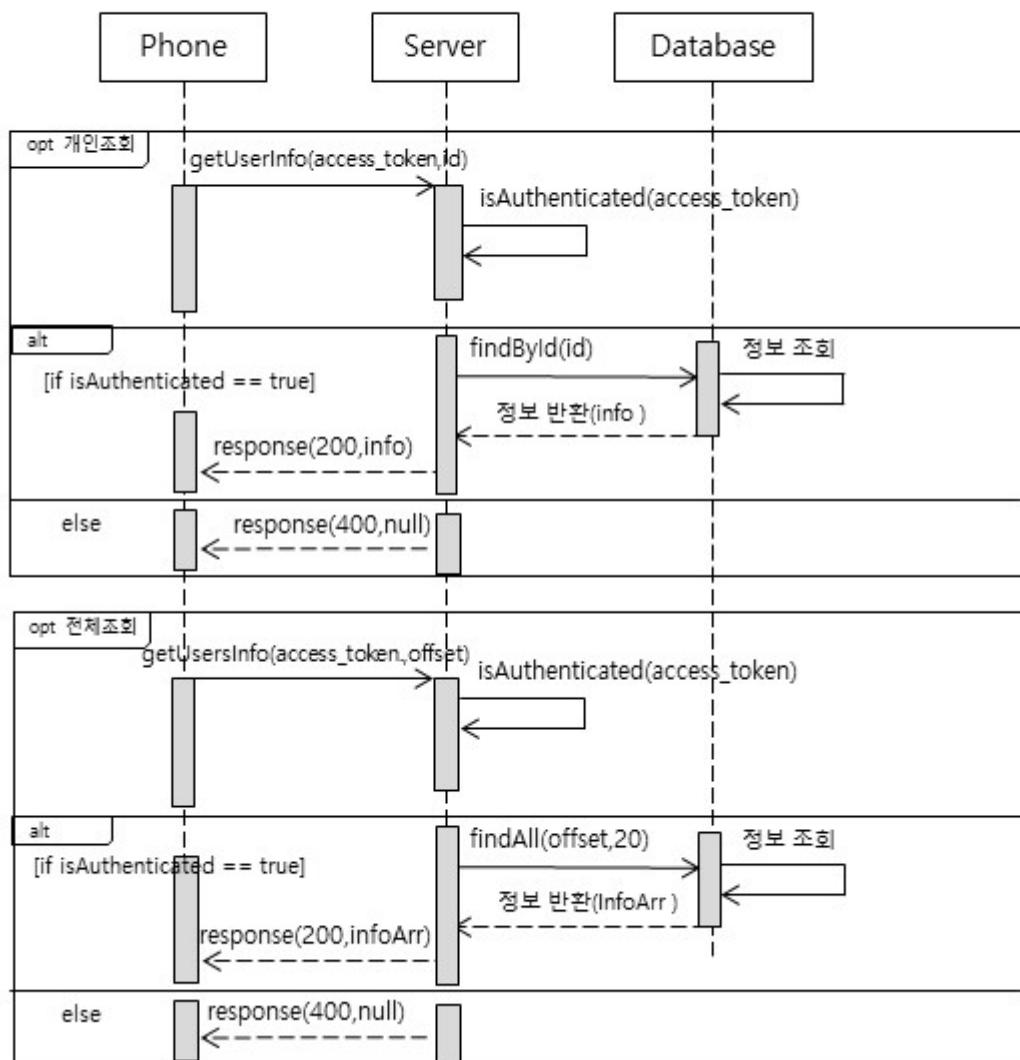
화장실 문 개방



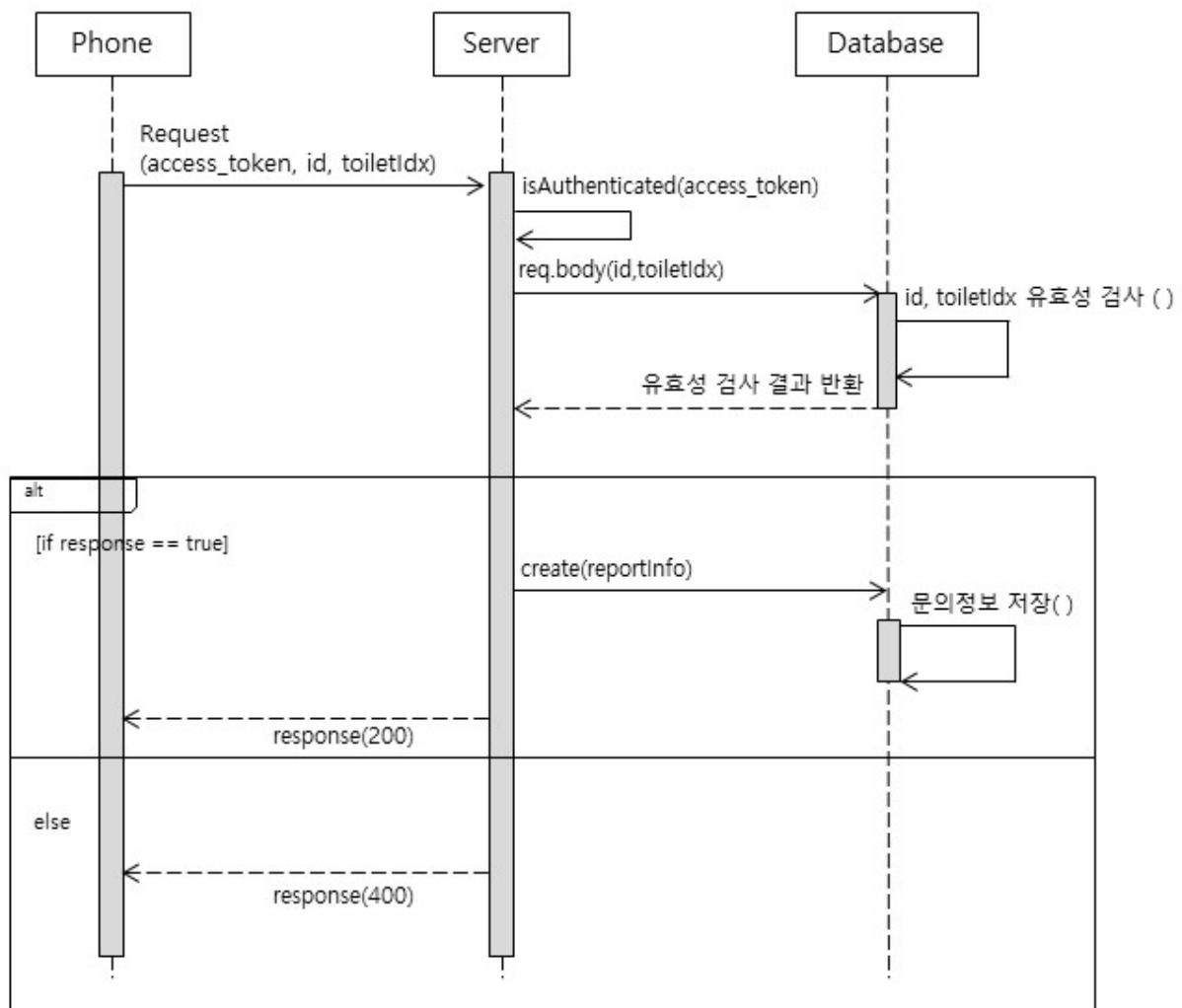
가까운 화장실 조회



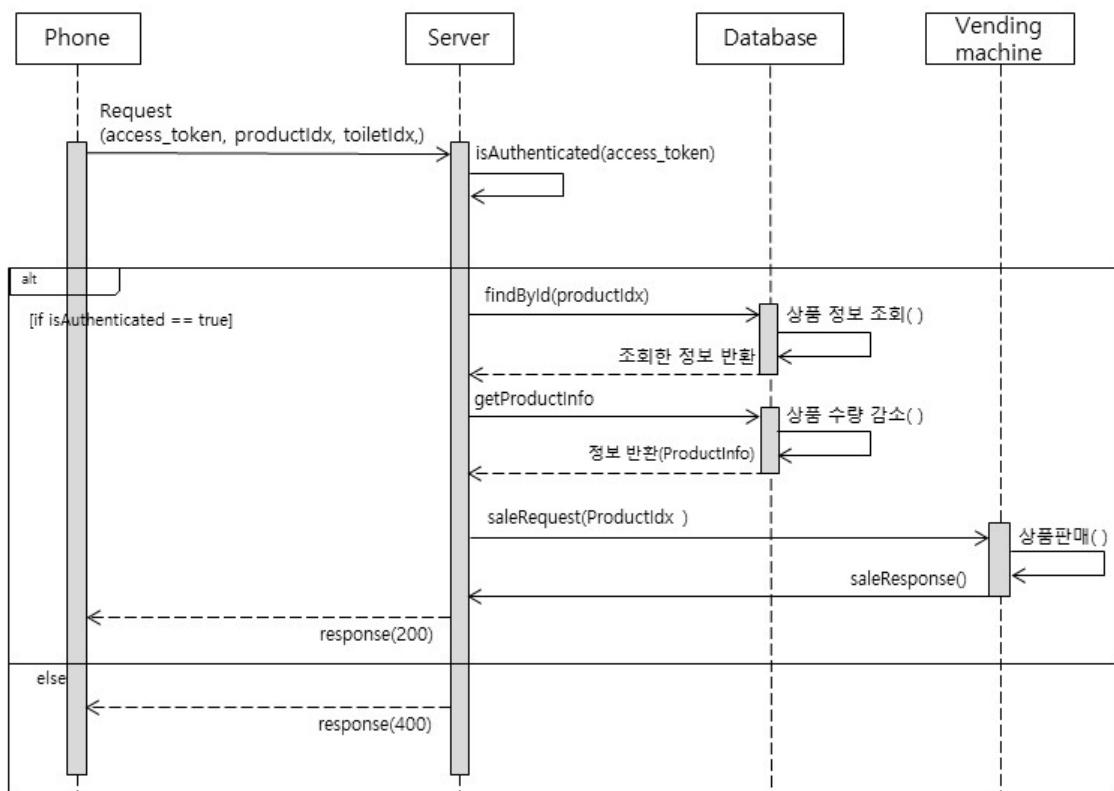
회원정보 조회



문의



상품구매



○ 소스 코드 (서버)

프로그램 ID	getAccessToken	프로그램 명	액세스 토큰 생성 - 1	작성일	2016.11.01	Page	1/2
개요	접속한 사용자의 회원 정보가 유효한지 확인한다.					작성자	박현식 최한규

```

var LocalStrategy = require('passport-local').Strategy;
module.exports = function() {
    console.log('passport-config');

    var strategy = new LocalStrategy({
        usernameField : 'id',
        passwordField : 'pwd',
    }, function(id, pwd, done) {
        models.Member.findById(id).then(function(member) {
            // not find member
            if (!member) {
                console.log('not find');
                return done(null, false);
            }
            // unvalid password
            if (!member.matchPassword(pwd)) {
                console.log('no match');
                return done(null, false);
            }

            return done(null, member);
        }, function(err) {
            if (err) {
                console.log('err');
                return done(err);
            }
        })
    });
}

```

프로그램 ID	getAccessToken	프로그램 명	액세스 토큰 생성 - 2	작성일	2016.11.01	Page	2/2
개요	접속한 사용자의 회원 정보를 기반으로 액세스 토큰을 생성한다.				작성자	박현식 최한규	

```

function getAccessToken(req,res,next){
    // custom callback
    passport.authenticate('local', function(err, member, info){
        var error = err||info
        if(err){
            return res.status(400).json(new ResultModel('F', error, null));
        }
        if(!member) {
            return res.status(400).json(new ResultModel('F', 'not exists', null));
        }
        var token = auth.signToken(member);
        res.json(new ResultModel('S', null, {access_token : token}));

    })(req,res,next);
}
function signToken(member) {
    var token = {};
    for(var key in member.dataValues){
        if(key !='pwd' && key !='ndPwd' && key !='salt'&&typeof(member[key])==='string'){
            token[key] = member[key];
        }
    }
    return jwt.sign({
        info : token
    }, SECRET_KEY, {
        expiresIn : EXPIRES
    });
}

```

프로그램 ID	isAuthenticated	프로그램 명	사용자 인증	작성일	2016.11.01	Page	1/1
개요	사용자에게 발급 된 토큰을 이용하여 인증한다.				작성자	박현식 최한규	

```

var validateJwt = require('express-jwt')({
    secret : SECRET_KEY
});

function isAuthenticated() {
    return compose().use(function(req, res, next) {
        if(req.query && req.query.hasOwnProperty('access_token')){
            req.headers.authorization = 'Bearer '+req.query.access_token;
        }
        validateJwt(req,res,next);
    }).use(function(req,res,next){
        next();
    });
}

```

프로그램 ID	openDoor	프로그램 명	도어락 개방 1	작성일	2016.11.01	Page	1/3
개요	도어락 개방을 위한 비동기 함수들을 순차적으로 호출한다.				작성자	박현식 최한규	

```

var commandRequest = function(info, cmd){
    this.info = info
    this.cmd = cmd
    commandRequest.prototype.setMember = function(member){
        this.member = member;
    }
    commandRequest.prototype.setToilet = function(toilet){
        this.toilet = toilet;
    }
    commandRequest.prototype.getMember = function(){
        return this.member;
    }
    commandRequest.prototype.getToilet = function(){
        return this.toilet;
    }
};

router.post('/:cmd/:beaconId', auth.isAuthenticated(), function(req, res) {
    var info = req.body;
    console.log(req.user);
    var member = req.user.info;
    var cmdRq = new commandRequest(info, req.params.cmd);
    console.log(cmdRq);
    async.waterfall([
        async.apply(matchPassword, member.id, cmdRq),
        getToiletInfo,
        validAccess,
        requestToilet
    ],
    function(err){
        if(err){
            res.status(400).json(new ResultModel('F', err, null));
        }else{
            res.status(200).json(new ResultModel('S', null, null));
        }
    });
});

```

프로그램 ID	openDoor	프로그램 명	도어락 개방 2	작성일	2016.11.01	Page	2/3
개요	비밀번호의 유효를 확인하고, 화장실 IP를 조회한다.				작성자	박현식 최한규	

```

function matchPassword(id, cmdReq, callback){
    models.Member.findById(id).then(function(member){
        var flag = member.matchNdPwd(cmdReq.info.ndPwd);
        if(flag){
            cmdReq.setMember(member);
            callback(null,cmdReq);
        }else {
            callback('not match password');
        }
    }, function(err){
        callback(err.message);
    });
}
function getToiletInfo(cmdReq, callback) {
    models.Toilet.findOne({
        where : {
            beaconId : cmdReq.info.beaconId
        }
    }).then(function(ret) {
        if (ret != null) {
            cmdReq.setToilet(ret);
            callback(null, cmdReq);
        }else{
            callback('not find toilet');
        }
    }, function(err) {
        callback(err.message);
    });
}

```

프로그램 ID	openDoor	프로그램 명	도어락 개방 3	작성일	2016.11.01	Page	3/3
개요	접근 권한을 확인하고 화장실 서버에 문 개방 요청을 보낸다.				작성자	박현식 최한규	

```

function validAccess(cmdReq, callback){
    var gender = cmdReq.getMember().gender;
    var type = cmdReq.getToilet().type;
    if(gender == type)
        callback(null, cmdReq);
    else
        callback('access denied');
}

function requestToilet(cmdReq, callback){
    var ipAddr = cmdReq.getToilet().ipAddr;
    var url = 'http://'+ipAddr+'/'+cmdReq.cmd;
    request.get(url, function (err,resp,body){
        console.log(body);
        if(err){
            callback(err.message);
        }else if(body.status == 'F'){
            callback(body.reason);
        }else{
            callback();
        }
    })
};

```

프로그램 ID	sequelizeConfig	프로그램 명	ORM 생성	작성일	2016.11.01	Page	1/1
개요	DB와 모델들을 매핑하고 관계를 설정한다.				작성자	박현식 최한규	

```

var sequelize = new Sequelize(DBInit.database, DBInit.username,
    DBInit.password, {
        dialect : DBInit.dialect,
        host : DBInit.host,
        port : DBInit.port,
        pool : DBInit.pool,
        define : {
            charset : 'utf8',
            collate : 'utf8_general_ci'
        }
    });
var db = {};

fs.readdirSync(__dirname)
.filter(function(file) {
    return (file.indexOf('.') !== 0) && (file != 'index.js' && file != 'association.js');
})
.forEach(function(file) {
    var model = sequelize.import(path.join(__dirname, file));
    db[model.name] = model;
});

config.initAssociations(db);

db.sequelize = sequelize;
db.Sequelize = Sequelize;

module.exports = db;

```

프로그램 ID	findToiletInfo	프로그램 명	사용자 주변 화장실 탐색	작성일	2016.11.01	Page	1/1
개요	지도 API에서 사용자의 위치를 기준으로 가까운 순으로 5개의 마포화장실을 탐색한다.			작성자	박현식 최한규		

```

function findToiletInfo(req, res){
    var qLatitude = parseFloat(req.query.latitude);
    var qLongitude = parseFloat(req.query.longitude);
    var result = {
        status : null,
        reason : null,
        toilets : null
    }
    models.Toilet.sequelize.query('SELECT toiletIdx, name, ( 6171 * acos( cos( radians(:qLatitude))*
cos( radians( latitude ) ) * cos( radians( longitude ) - radians(:qLongitude) ) + sin( radians(:qLatitude) ) *
sin( radians( latitude ) ) ) ) AS distance FROM toilets HAVING distance < 10 ORDER BY distance LIMIT 0, 5', {replacements :
{ qLatitude : qLatitude, qLongitude : qLongitude}})
    .then(function(ret){
        if(ret == null) {
            res.status(400);
            result.status = 'F';
            result.reason = 'not find toilet';
            res.json(result);
        } else {
            console.log(ret[0]);
            result.status = 'S';
            result.toilets = ret[0];
            res.json(result);
        }
    }, function(err) {
        console.log(err);
        res.status(400);
        result.status = 'F';
        result.reason = err.message;
        res.json(result);
    })
}

```

○ 소스 코드 (App)

프로그램 ID	NetworkManager	프로그램 명	서버통신(POST)	작성일	2016.11.02	Page	1/5
개요	Okhttp를 이용하여 NetworkManager를 통한 POST방식의 서버통신이 가능하다.			작성자	엄선오		

```
/*회원가입*/
private static final String SMART_TOILET_SIGNUP = SMART_TOILET_SEVER + "보안상 내용 대체";

public Request getUserSignUp(String id, String pwd, String ndPwd, String gender, String phoneNum, OnResultListener<SignUpResult> listener) {
    String url = SMART_TOILET_SIGNUP;

    RequestBody body = new FormBody.Builder()
        .add("id", id)
        .add("pwd", pwd)
        .add("ndPwd", ndPwd)
        .add("gender", gender)
        .add("phoneNum", phoneNum)
        .build();

    Request request = new Request.Builder()
        .head()
        .url(url)
        .post(body)
        .build();

    final NetworkResult<SignUpResult> result = new NetworkResult<>();
    result.request = request;
    result.listener = listener;
    mClient.newCall(request).enqueue(new Callback() {
        @Override
        public void onFailure(Call call, IOException e) {
            result.exception = e;
            mHandler.sendMessage(mHandler.obtainMessage(MESSAGE_FAIL, result)); //실패했는지
        }
    });
}
```

프로그램 ID	SignUp	프로그램 명	서버통신(POST)	작성일	2016.11.02	Page	2/5
개요	2자 비밀번호 등록 후 회원가입에 필요한 정보 id,pwd ,ndPwd,gender,phoneNum을 서버로 전송한다.					작성자	엄선오

```

if (passcode_check.equals(passcode)) {

    NetworkManager.getInstance().getUserSignUp(id, pwd, passcode_check, gender, phoneNum, new NetworkManager.OnResultListener<SignUpResult>() {
        @Override
        public void onSuccess(Request request, SignUpResult result) {

            SignUpResult data = result;
            Toast.makeText(SecurePswActivity.this, "전송 성공.", Toast.LENGTH_SHORT).show();

        }

        @Override
        public void onFailure(Request request, IOException exception) {
            Toast.makeText(SecurePswActivity.this, "전송 실패", Toast.LENGTH_LONG).show();
            Snackbar.make(view, "전송 실패", Snackbar.LENGTH_SHORT).show();
        }
    });
}

```

프로그램 ID	NetworkManager	프로그램 명	서버통신(GET)	작성일	2016.11.02	Page	3/5
개요	Okhttp를 이용하여 NetworkManager를 통한 GET방식의 MATTO App 전체 물품 조회 기능이다.			작성자	엄선오		

```

private static final String SMART_TOILET_AITEM = SMART_TOILET_SEVER + "/api/0.1v/products";
public Request getUserAitem(String aitemSn, OnResultListener<AitemResult> listener) {
    aitemSn = "0";

    String url = SMART_TOILET_AITEM+"?offset="+aitemSn;

    Log.d("url", "url 값 : " + url);
    Request request = new Request.Builder()
        .get()
        .head()
        .url(url)
        .post(body)
        .build();

    final NetworkResult<AitemResult> result = new NetworkResult<>();
    result.request = request;
    result.listener = listener;
    mClient.newCall(request).enqueue(new Callback() {
        @Override
        public void onFailure(Call call, IOException e) {
            result.exception = e;
            mHandler.sendMessage(mHandler.obtainMessage(MESSAGE_FAIL, result)); //실패했는지
        }

        @Override
        public void onResponse(Call call, Response response) throws IOException {
            Log.w("MATTO_Error", "에러부분 : " + response.body());
            List<Product> pro = new ArrayList<Product>();
            pro = response.body().charStream();

            String text = response.body().toString();
        }
    });
}

```

프로그램 ID	BeaconRangingListAdapter	프로그램 명	비콘 정보 수신 (1)	작성일	2016.11.02	Page	4/5
개요	블루투스로 연결된 특정 비콘의 Uuid,Major,Minor,TxPower,Proximity,Accuracy값을 수신한다. [] 값 안에는 원하는 거리값을 넣는다.(추가적으로 작업중)					작성자	엄선오

```

RECOBeacon recoBeacon = mRangedBeacons.get(position);

String proximityUuid = recoBeacon.getProximityUuid();

viewHolder.recoProximityUuid.setText(String.format("%s-%s-%s-%s-%s", proximityUuid.substring(0, 8), proximityUuid.substring(8, 12), proximityUuid.substring(12, 16));
viewHolder.recoMajor.setText(recoBeacon.getMajor() + "");
viewHolder.recoMinor.setText(recoBeacon.getMinor() + "");
viewHolder.recoTxPower.setText(recoBeacon.getTxPower() + "");
viewHolder.recoRssi.setText(recoBeacon.getRssi() + "");
viewHolder.recoProximity.setText(recoBeacon.getProximity() + "");

if(recoBeacon.getAccuracy() <= [ ]){
    viewHolder.recoAccuracy.setText(String.format("%.2f ", recoBeacon.getAccuracy()));
}
else {
    viewHolder.recoAccuracy.setText(String.format("사용할 수 없는 비콘 "));
}

}

if(recoBeacon.getAccuracy() <= 0.5){

    Log.w(SEONOH_TAG, '전송된 recoAccuracy값 : '+recoBeacon.getAccuracy());
}

return convertView;

```

프로그램 ID	RecoMonitoringListAdapter	프로그램 명	비콘 정보 수신 (2)	작성일	2016.11.02	Page	5/5
개요	블루투스를 연결하여 해당 위치에서 비콘 정보를 받아들일 수 있는지 없는지에 대한 식별을 해준다.				작성자	엄선오	

```

if(recoRegionState.equals(RECOBeaconRegionState.RECOBeaconRegionInside.toString()) && mMatchedBeaconCounts.get(recoRegion) == 0) {
    viewHolder.recoRegionBeaconCount.setText("You started monitoring inside of the region.");
    return convertView;
}

if(recoRegionState.equals(RECOBeaconRegionState.RECOBeaconRegionOutside.toString())) {
    viewHolder.recoRegionBeaconCount.setText("No beacons around.");
    return convertView;
}

viewHolder.recoRegionBeaconCount.setText("# of beacons in the region: " + recoBeaconCount);

return convertView;
}

```

○ 소스 코드 (라즈베리파이)

프로그램 ID	Server.js	프로그램 명	라즈베리파이 메인 코드	작성일	2016.10.28	Page	1/1
개요	시리얼 통신을 통해 아두이노에 JSON형식으로 명령어를 전달한다.				작성자	윤상현	

```

var express = require('express');
var app = express();
var SerialPort = require('serialport');
var bodyParser = require('body-parser');
var logger = require('morgan');app.listen(3000);
app.use(logger('dev'));
app.use(bodyParser.json());
app.use(bodyParser.urlencoded({ extended : false}));
app.get('/:cmd', function(req, res) {
    var cmd = '{"cmd" : ' + req.params.cmd + ', "data" : "' + req.query.data + '" }';
    console.log(cmd);
    var serial = new SerialPort('/dev/ttyACM0', {
        baudrate : 9600,
        lock : true,
        parser : SerialPort.parsers.readline('\n')
    }, function(err) {
        if (err) {
            res.end('err');
        } else {
            serial.write(cmd);
            serial.on('data', function(data) {
                console.log('Read Data : ' + data);
                res.end(data);
                serial.close();
            });
            serial.on('error', function(err) {
                console.log('Error: ', err.message);
                res.end(err);
            });
        }
    });
});

```

○ 소스 코드 (H/W)

프로그램 ID	Setup, loop	프로그램 명	아두이노 메인 함수	작성일	2016.10.28	Page	1/1
개요	아두이노 소스의 setup부와 loop부의 코드					작성자	윤상현

```

void setup(){
    pinMode(pin_tissueLed,OUTPUT);
    pinMode(pin_roomLed,OUTPUT);
    Serial.begin(9600);
    dht.begin();           //온습도센서 로딩
    //도어락 연결
    womDoorlock.attach(pin_doorlockWom);
    manDoorlock.attach(pin_doorlockMan);
    womDoorlock.write(lock);
    manDoorlock.write(lock);
    sensing_tissue();
    t.every(1000,sensing_tissue); //1분마다 휴지체크
}

void loop(){
    t.update();
    current_time = millis();
    sensing_door(pin_doorInWom,pin_doorOutWom,pin_doorInMan,pin_doorOutMan); //입
구센서 센싱
    doorCount();           //센싱값으로 인원처리
    if(Serial.available()){
        fromSerial();      //시리얼통신으로 받을게 있으면 커뮤니케이션 함수 실행
    }
}

```

프로그램 ID	fromSerial	프로그램 명	시리얼 수신 코드	작성일	2016.10.28	Page	1/1
개요	시리얼 통신으로 JSON형식으로 통신				작성자	윤상현	

```

void fromSerial(){      //수신 코드
    static int i=0;
    char data = Serial.read();          //일단 읽음
    if(needCommu){                    //읽는 중이었다면
        if(data == '}'){
            //마지막을 알리는 경우 설정값 초기화와 함께 파싱 진행
            in_data[i++]=data;
            in_data[i]='W0';
            i=0;
            needCommu = false;
            endCommu = true;
        }
        else{
            in_data[i++]=data;  //내용의 중간인 경우 in_data에 계속 추가
        }
    }
    else{
        if(data == '{'){
            needCommu = true; //내용의 시작을 알리면 읽기 모드 돌입
            in_data[i++]=data;
        }
    }
    if(endCommu){
        //시리얼 값 다 받으면 JSON파싱
        StaticJsonBuffer<200> jsonBufferIn;
        JsonObject& recv = jsonBufferIn.parseObject(in_data);
        //JSON분석을 통한 실행
        parseData(recv);
        endCommu=false;
    }
}

```

프로그램 ID	parseData	프로그램 명	데이터 파싱	작성일	2016.10.28	Page	1/1
개요	받은 JSON형식을 파싱하여 해당하는 명령어 실행				작성자	윤상현	

```

void parseData(JsonObject& parse){
    StaticJsonBuffer<200> jsonBufferOut;           //버퍼설정
    JsonObject& send = jsonBufferOut.createObject();
    char* cmd = parse["cmd"];                      //명령어 파싱
    //명령어가 dopen이면
    if(strcmp(cmd,"dopen")==0){
        char* data = parse["data"];
        if(strcmp(data,"F") || strcmp(data,"M")){
            if(doorOpen(data)==0){
                send["status"] = "S";
            } else {
                send["status"] = "F";
            }
        } else {
            send["status"] = "F";
        }
    }
    //명령어가 sensing이면
    else if(strcmp(cmd,"sensing")==0){
        //JSON오브젝트에 각 값을 추가
        send["status"] = "S";
        sensing_dht();
        if (isnan(humidity) || isnan(temperature)){    }
        else {
            send["tem"] = temperature;
            send["hum"] = humidity;
        }
        send["tissue"] = val_tissue;
        send["count"] = number_personWom;
        send["room"] = sensing_room();
    }
    else{
        send["status"] = "F";
    }
    //값 시리얼로 전송
    send.printTo(Serial); Serial.println("");
}

```

프로그램 ID	doorOpen doorClose	프로그램 명	문열림과 자동 닫힘	작성일	2016.10.28	Page	1/1
개요	문을 열고 내부의 카운트 센서를 통해 카운트가 증가하면 문을 닫음				작성자	윤상현	

```

int doorOpen(char* toilet){
    if (strcmp(toilet,"F")==0){
        womDoorlock.write(unlock);
        save_personWom = number_personWom;
        notPassWom = t.every(1000,doorClose);
    } else if(strcmp(toilet,"M")==0){
        manDoorlock.write(unlock);
        save_personMan = number_personMan;
        notPassMan = t.every(1000,doorClose);
    }
    else {
        return 1;
    }
    return 0;
}
void doorClose(){
    // Serial.println("Not Closed");
    if((notPassWom!=0) && (number_personWom>save_personWom)){
        womDoorlock.write(lock);
        t.stop(notPassWom);
        notPassWom=0;
        save_personWom = 0;
    } else if((notPassMan!=0) && (number_personMan>save_personMan)){
        manDoorlock.write(lock);
        t.stop(notPassMan);
        notPassMan=0;
        save_personMan = 0;
    }
}

```

프로그램 ID	Sensing	프로그램 명	센서의 센싱	작성일	2016.10.28	Page	1/2
개요	각종 센서들의 센싱 함수들				작성자	윤상현	

```

void sensing_tissue(){
    val_tissue = digitalRead(pin_tissueSens);
    //val_tissue를 반전시킨 값으로 LED를 온/오프 (휴지가 부족한 경우 val_tissue값이 0)
    digitalWrite(pin_tissueLed,val_tissue==1?0:1);
}

void sensing_dht(){
    humidity = dht.readHumidity(); //습도
    temperature = dht.readTemperature(); //온도 (C)
}

int sensing_room(){
    digitalWrite(pin_roomLed,1);
    delay(50);
    val_roomWom[0] = digitalRead(pin_room1);
    val_roomWom[1] = digitalRead(pin_room2);
    int sum=0;
    for(int cnt=0; cnt<2; cnt++){
        if(val_roomWom[cnt] == 1){
            sum++;
        }
    }
    digitalWrite(pin_roomLed,0);
    return sum;
}

```

프로그램 ID	Sensing	프로그램 명	센서의 센싱	작성일	2016.10.28	Page	2/2
개요	각종 센서들의 센싱 함수들				작성자	윤상현	

```

void sensing_door(char inPinWom, char outPinWom, char inPinMan, char outPinMan){
    int valInWom = analogRead(inPinWom);
    int valOutWom = analogRead(outPinWom);
    int valInMan = analogRead(inPinMan);
    int valOutMan = analogRead(outPinMan);
    float cmInWom = 2076/(valInWom-11); //그래프를 수식화한 식
    float cmOutWom = 2076/(valOutWom-11);
    float cmInMan = 10650.08 * pow(valInMan,-0.935) - 10;
    float cmOutMan = 10650.08 * pow(valOutMan,-0.935) - 10;
    // Serial.print(cmOutWom); Serial.print(" "); Serial.println(cmInWom);
    //여자 안
    if(cmInWom<cond_womEntDistance){
        val_doorInWom = 1;
    } else{
        val_doorInWom = 0;
    }
    //여자 밖
    if(cmOutWom<cond_womEntDistance){
        val_doorOutWom = 1;
    } else{
        val_doorOutWom = 0;
    }
    //남자 안
    if(cmInMan<cond_manEntDistance){
        val_doorInMan = 1;
    } else{
        val_doorInMan = 0;
    }
    //남자 밖
    if(cmOutMan<cond_manEntDistance){
        val_doorOutMan = 1;
    } else{
        val_doorOutMan = 0;
    }
}

```

프로그램 ID	doorCount	프로그램 명	센싱값에 따라 인원수 처리	작성일	2016.10.28	Page	1/1
개요						작성자	윤상현

```

void doorCount(){
    //안쪽센서 인식시
    if(val_doorInWom == 1 && val_doorOutWom == 0){
        if(out == false && !inMoment){
            // Serial.println("intrue");
            // delay(1000);
            in = true;
        } else if (out){
            number_personWom++;
            Serial.println(number_personWom);
            // delay(1000);
            inMoment = true;
            in=out=false;
            t.after(doorInterval,unMomentIn);
        }
    }
    //바깥쪽센서 인식시
    if(val_doorOutWom == 1 && val_doorInWom == 0){
        if(in == false && !outMoment){
            // Serial.println("outture");
            // delay(1000);
            out = true;
        } else if (in){
            if(number_personWom>0){
                number_personWom--;
                Serial.println(number_personWom);
                // delay(1000);
            }
            in=out=false;
            outMoment = true;
            t.after(doorInterval,unMomentOut);
        }
    }
}

```

- RESTful API 명세서

1. 회원 관리

1-1. 회원가입 API

URL : /members

method : POST

회원정보를 받아 DB에 기록한다.

Request

id	String	아이디
gender	String	성별
phoneNum	String	휴대폰 번호
pwd	String	1차 비밀번호
ndPwd	String	2차 비밀번호

Response(JSON으로 응답)

```
{
    status :String, //상태(성공(S),실패(F))
    reason : String //실패시 사유, 성공시 빈값(“”)
    resultData : null
}
```

1-2. 회원정보 수정 API

URL : /members/{:id}

method : PUT

회원정보를 받아 DB의 내용을 수정 한다.

Request

access_token	String	헤더 or 쿼리스트링에 토큰
phoneNum	String	휴대폰 번호
pwd	String	현재 비밀번호
chgPwd	String	변경할 비밀번호
ndPwd	String	2차 비밀번호

Response(JSON으로 응답)

```
{
    status :String, //상태(성공(S), 실패(F))
    reason : String //실패시 사유, 성공시 빈값(“”)
    resultData : null
}
```

1-3. 회원 단체 조회 API

URL : /members?offset={offset}

method : GET

Request

access_token	String	헤더 or 쿠리스트링에 토큰
offset	int	조회할 첫 위치

offset부터 20개까지 반환

Response(JSON으로 응답)

```
{
    status : String, // 상태(성공(S), 실패(F))
    reason : String, // 실패시 사유, 성공시 빈 값(“”)
    resultData : {
        lastOffset : String or Intger, // 마지막 오프셋
        members : [
            {
                id : String, // 조회한 아이디
                gender : String, // 성별
                phoneNum : String // 휴대폰 번호
            },
            {
                id : String, // 조회한 아이디
                gender : String, // 성별
                phoneNum : String // 휴대폰 번호
            } // lquantity 개수만큼 반복
        ]
    }
}
```

1-4. 회원 개인 조회 API

URL : /members/{:id}?type=

method : GET

Request

access_token	String	헤더 or 쿠리스트링에 토큰
type	String	조회타입

responses

type = id 일 경우

Response(JSON으로 응답)

```
{
    status : String, // 상태(성공(S),실패(F))
    reason : String // 실패시 사유, 성공시 빈 값(“”)
    resultData : null
}
```

responses

type = id 일 경우

Response(JSON으로 응답)

```
{
    status : String, // 상태(성공(S),실패(F))
    reason : String, // 실패시 사유, 성공시 빈 값(“”)
    resultData : {
        member : {
            id : String, // 조회한 아이디
            gender : String, //성별
            phoneNum : String //휴대폰 번호
        }
    }
}
```

1-5. 회원정보 삭제 API

URL : /members/{:id}

method : DELETE

Request

access_token	String	헤더 or 쿠리스트링에 토큰
id	String	아이디

Response(JSON으로 응답)

```
{
    status : String, // 상태(성공(S),실패(F))
    reason : String // 실패시 사유, 성공시 빈 값(“”)
    resultData : null
}
```

1-6. 관리자전용 회원정보 수정 API

URL : /members/authentic/{:id}

method : PUT

관리자 권한으로 회원정보를 받아 DB의 내용을 수정 한다.

Request

access_token	String	헤더 or 쿠리스트링에 토큰
pwd	String	현재 비밀번호
chgPwd	String	변경할 비밀번호
ndPwd	String	2차 비밀번호
name	String	사용자 이름
gender	String	성별
phoneNum	String	핸드폰 번호
auth	String	사용자 권한

Response(JSON으로 응답)

```
{
    status : String, // 상태(성공(S),실패(F))
    reason : String // 실패시 사유, 성공시 빈 값(“”)
    resultData : null
}
```

2. 화장실 관리

2-1. 화장실 등록 API

URL : /toilets

method : POST

화장실 정보를 등록한다.

Request

access_token	String	헤더 or 큐리스트링에 토큰
name	String	화장실 이름
longitude	String	위도
latitude	String	경도
address	String	주소
roomCount	int	화장실 칸 개수
imgSrc	String	화장실 이미지
beaconId	String	비콘 ID
ipAddr	String	ip주소(포트 포함)
type	String	남/여 화장실 구분

Response(JSON으로 응답)

```
{
    status :String, //상태(성공(S),실패(F))
    reason : String //실패시 사유, 성공시 빈값(“”)
    resultData : null
}
```

2-2. 화장실 수정 API

URL : /toilets/{:toiletIdx}

method : PUT

화장실 정보를 수정한다.

Request

access_token	String	헤더 or 쿼리스트링에 토큰
toiletIdx	int	화장실 idx
name	String	화장실 이름
longitude	String	위도
latitude	String	경도
address	String	주소
roomCount	int	화장실 칸 개수
imgSrc	String	화장실 이미지
beaconId	String	비콘 ID
ipAddr	String	ip주소(포트 포함)
type	String	남/여 화장실 구분

Response(JSON으로 응답)

```
{
    status :String, //상태(성공(S),실패(F))
    reason : String //실패시 사유, 성공시 빈값(“”)
    resultData : null
}
```

2-3. 전체 화장실 정보 조회 API

URL : /toilets?offset={integer}

method : GET

전체 화장실의 idx, name 조회한다.

Request

access_token	String	헤더 or 쿼리스트링에 토큰
offset	Integer	조회할 첫 위치

offset부터 20개까지 반환

Response(JSON으로 응답)

```
{
    status : String, //성공, 실패 여부
    reason : String, //실패 사유, 성공 시 빈 값(“”)
    resultData : {
        lastOffset : String or Intger, //마지막 오프셋
        toilets : [
            {
                toiletIdx : int, // 화장실 인덱스
                name : String //화장실 이름
            },
            {
                toiletIdx : int, // 화장실 인덱스
                name : String //화장실 이름
            } //length 만큼 반복
        ]
    }
}
```

2-4. 특정 화장실 조회 API

URL : /toilets/{:toiletIdx}

method : GET

특정 화장실의 정보를 조회한다.

Request

access_token	String	헤더 or 쿼리스트링에 토큰
toiletIdx	int	화장실 idx

Response(JSON으로 응답)

```
{
    status : String, //상태(성공(S),실패(F))
    reason : String, //실패 사유, 성공 시 빈 값(“”)
    resultData : {
        toiletInfo : {
            name : String, //화장실 이름
            longitude : Double, //위도
            latitude : Double, //경도
            address : String, //주소
            roomCount : String, //
            imgSrc : String, //화장실 이미지
            beaconId : String //비콘 ID
        }
    }
}
```

2-5. 근처 화장실 조회 API

URL : /toiletFind

method : GET

특정 화장실의 정보를 조회한다.

Request

access_token	String	헤더 or 쿠리스트링에 토큰
toiletIdx	int	화장실 idx

Response(JSON으로 응답)

```
{
    status : String, //성공/실패 여부
    reason : String //실패 사유, 성공 시 빈 값(“”)
    resultData : {
        toilets : [
            {
                toiletIdx : int,
                name : String,
                distance : float
            },
            {
                toiletIdx : int,
                name : String,
                distance : float
            }
        ]
    }
}
```

2-6. 화장실 정보 삭제 API

URL : /toilets/{:toiletIdx}

method : DELETE

특정 화장실의 정보를 삭제한다.

Request

access_token	String	헤더 or 쿼리스트링에 토큰
toiletIdx	int	화장실 idx

Response(JSON으로 응답)

```
{
    status : String, //성공/실패 여부
    reason : String //실패 사유, 성공 시 빈 값(“”)
    resultData : null
}
```

2-7. 화장실 인덱스 획득 API

URL : /toilets/getIdx/{:beaconId}

method : GET

비콘ID로 화장실 인덱스를 가져온다.

Request

access_token	String	헤더 or 쿼리스트링에 토큰
beaconId	int	비콘 ID

Response(JSON으로 응답)

```
{
    status : String, //성공/실패 여부
    reason : String //실패 사유, 성공 시 빈 값(“”)
    resultData : [
        {
            toiletIdx : Integer
        }
    ]
}
```

3. 상품 관리

3-1. 상품 등록 API

URL : /products

method : POST

상품 정보를 등록한다.

Request

name	String	상품 이름
price	int	상품 가격
imgSrc	String	상품 이미지
description	String	상품 설명

Response(JSON으로 응답)

```
{
    status :String, //상태(성공(S),실패(F))
    reason : String //실패시 사유, 성공시 빈값(“”)
    resultData : null
}
```

3-2. 상품 수정 API

URL : /products/{:productIdx}

method : PUT

상품 정보를 수정한다.

Request

productIdx	int	상품 idx
name	String	상품 이름
price	int	상품 가격
imgSrc	String	상품 이미지
description	String	상품 설명

Response(JSON으로 응답)

```
{
    status :String, //상태(성공(S),실패(F))
    reason : String //실패시 사유, 성공시 빈값(“”)
    resultData : null
}
```

3-3. 상품 삭제 API

URL : /products/{:productIdx}

method : DELETE

상품 정보를 삭제한다.

Request

productIdx	int	상품 idx
------------	-----	--------

Response(JSON으로 응답)

```
{  
    status :String, //상태(성공(S),실패(F))  
    reason : String //실패시 사유, 성공시 빈값(“”)  
    resultData : null  
}
```

3-4. 전체 상품 조회 API

URL : /products?offset={integer}

method : GET

전체 상품을 조회한다.

Request

access_token	String	헤더 or 쿠리스트링에 토큰
offset	int	조회할 첫 위치

//offset부터 20개까지 반환

Response(JSON으로 응답)

```
{
    status : String, //성공, 실패 여부
    reason : String, //실패 사유, 성공 시 빈 값(“”)
    resultData : {
        length : Integer, // 상품 갯수
        lastOffset : String or Integer, //마지막 오프셋
        products : [
            {
                productIdx : int, // 상품 인덱스
                name : String, //상품 이름
                price : int, //상품 가격
                imgSrc: String, //상품 이미지
                description : String // 상품 설명
            },
            {
                productIdx : int, // 상품 인덱스
                name : String, //상품 이름
                price : int, //상품 가격
                imgSrc: String, //상품 이미지
                description : String // 상품 설명
            } //length 만큼 반복
        ]
    }
}
```

3-5. 특정 상품 정보 조회 API

URL : /products/{:productIdx}

method : GET

특정 상품의 정보를 조회한다.

Request

access_token	String	헤더 or 쿼리스트링에 토큰
productIdx	int	상품 idx

Response(JSON으로 응답)

```
{
    status : String, //상태(성공(S),실패(F))
    reason : String, //실패 사유, 성공 시 빈 값(“”)
    resultData : {
        product_info : {
            name : String, //상품 이름
            price : int, //상품 가격
            imgSrc: String, //상품 이미지
            description : String // 상품 설명
        }
    }
}
```

3-6. 특정 회원의 상품 구매 내역 조회 API

URL : /product/sells/{:id}?type=user

URL : /product/sells/{:toiletIdx}?type=toilet

URL : /product/sells

method : GET

해당 아이디의 상품 구매 내역을 조회한다.

Request

access_token	String	헤더 or 쿼리스트링에 토큰
id	String	아이디
toiletIdx	int	화장실 idx

Response(JSON으로 응답)

```
{
    status : String, //성공, 실패 여부
    reason : String, //실패 사유, 성공 시 빈 값(“”)
    resultData : {
        length : Integer, // 구매 횟수
        sells : [
            {
                productIdx : int, // 상품 인덱스
                toiletIdx // 화장실 인덱스
                name : String, //상품 이름
                price : int, //상품 가격
                amount, //상품 개수
                t_price //상품가격 * 상품개수 (총 금액)
            },
            {
                productIdx : int, // 상품 인덱스
                toiletIdx // 화장실 인덱스
                name : String, //상품 이름
                price : int, //상품 가격
                amount, //상품 개수
                t_price //상품가격 * 상품개수 (총 금액)
            } //length 만큼 반복
            total_price : int // 상품 총 구매 가격
        ]
    }
}
```

4. 화장실 상품 관리

4-1. 화장실에 상품 등록 API

URL : /toilets/{:toiletIdx}/product

method : POST

{toiletIdx} 화장실에 상품을 등록한다.

Request

access_token	String	헤더 or 쿼리스트링에 토큰
productIdx	int	상품 idx
stock	String	상품 재고

Response(JSON으로 응답)

```
{
    status :String, //상태(성공(S),실패(F))
    reason : String //실패시 사유, 성공시 빈값(“”)
    resultData : null
}
```

4-2. 화장실에 상품 삭제 API

URL : /toilets/{:toiletIdx}/product/{:productIdx}

method : DELETE

Request

access_token	String	헤더 or 쿼리스트링에 토큰
--------------	--------	-----------------

Response(JSON으로 응답)

```
{
    status :String, //상태(성공(S),실패(F))
    reason : String //실패시 사유, 성공시 빈값(“”)
    resultData : null
}
```

4-3. 화장실 상품 구매 API

URL : /toilets/{:toiletIdx}/buy/{:productIdx}

method : GET

상품을 구매한다. 재고가 0개 이하면 F를 반환한다.

Request

access_token	String	헤더 or 쿼리스트링에 토큰
productIdx	int	상품 idx
toiletIdx	int	화장실 idx

Response(JSON으로 응답)

```
{
    status :String, //상태(성공(S),실패(F))
    reason : String //실패시 사유, 성공시 빈값(“”)
    resultData : null
}
```

4-4. 화장실에 등록된 상품 재고 수정 API

URL : /toilets/{:toiletIdx}/product/{:productIdx}

method : PUT

화장실에 등록된 상품의 재고 정보를 수정한다.

Request

access_token	String	헤더 or 쿼리스트링에 토큰
stock	String	상품 재고

Response(JSON으로 응답)

```
{
    status :String, //상태(성공(S),실패(F))
    reason : String //실패시 사유, 성공시 빈값(“”)
    resultData : null
}
```

4-5. 특정 화장실에 등록된 상품 목록 조회 API

URL : /toilets/{:toiletIdx}/product?offset={integer}

method : GET

특정 화장실에 등록된 상품 목록을 조회한다.

Request

access_token	String	헤더 or 쿼리스트링에 토큰
offset	int	조회할 첫 위치

//offset부터 20개까지 반환

Response(JSON으로 응답)

```
{
    status : String, //성공/실패 여부
    reason : String //실패 사유, 성공 시 빈 값(“”)
    resultData : {
        lastOffset : int // 마지막 참조 값
        products : [
            {
                toiletIdx : int // 화장실 인덱스
                productIdx : int // 상품 인덱스
                stock : int // 재고량
            },
            {
                toiletIdx : int // 화장실 인덱스
                productIdx : int // 상품 인덱스
                stock : int // 재고량
            } //offset만큼 반복
        ]
    }
}
```

5. 신고 관리

5-1. 신고 등록 API

URL : /toilets/{:toiletIdx}/product

method : POST

불편사항을 등록한다.

Request

access_token	String	헤더 or 큐리스트링에 토큰
toiletIdx	int	화장실 idx
contents	String	신고 내용

Response(JSON으로 응답)

```
{
    status :String, //상태(성공(S),실패(F))
    reason : String //실패시 사유, 성공시 빈값(“”)
    resultData : null
}
```

5-2. 관리자 전용 신고 전체 리스트 조회 API

URL : /reports

method : GET

관리자 권한으로 전체 신고 리스트를 조회한다.

Request

access_token	String	헤더 or 쿠리스트링에 토큰
offset	int	조회할 첫 위치

//offset부터 20개까지 반환

Response(JSON으로 응답)

```
{
    status : String, //성공, 실패 여부
    reason : String, //실패 사유, 성공 시 빈 값(“”)
    resultData : {
        length : Integer, // 신고 갯수
        lastIndex : String or Integer, //마지막 인덱스
        reports : [
            {
                report_idx : int, // 신고 인덱스
                id : String, //아이디
                toiletIdx : int, //화장실 인덱스
                report: String, //신고 내용
            },
            {
                report_idx : int, // 신고 인덱스
                id : String, //아이디
                toiletIdx : int, //화장실 인덱스
                report: String, //신고 내용
            } //length 만큼 반복
        ]
    }
}
```

5-3. 신고 특정 리스트 조회 API

URL : /reports/{:reportIdx}

method : GET

특정 신고 리스트를 조회한다.

Request

access_token	String	헤더 or 쿼리스트링에 토큰
reportIdx	int	신고 idx

Response(JSON으로 응답)

```
{
    status : String, //상태(성공(S),실패(F))
    reason : String, //실패 사유, 성공 시 빈 값(“”)
    resultData : {
        report_info : {
            id : String, //아이디
            toiletIdx : int, //화장실 인덱스
            report: String, //신고 내용
        }
    }
}
```

5-4. 특정 화장실 신고 리스트 조회 API

URL :/reports/{:toiletIdx}/reports?offset={integer}

method : GET

특정 화장실의 신고 리스트를 조회한다.

Request

access_token	String	헤더 or 쿠리스트링에 토큰
toiletIdx	int	화장실 idx

Response(JSON으로 응답)

```
{
    status : String, //성공, 실패 여부
    reason : String, //실패 사유, 성공 시 빈 값(“”)
    resultData : {
        length : Integer, // 신고 갯수
        lastIndex : String or Integer, //마지막 인덱스
        reports : [
            {
                toiletIdx : int // 화장실 인덱스
                report_idx : int, // 신고 인덱스
                id : String, //아이디
                toiletIdx : int, //화장실 인덱스
                report: String, //신고 내용
            },
            {
                toiletIdx : int // 화장실 인덱스
                report_idx : int, // 신고 인덱스
                id : String, //아이디
                toiletIdx : int, //화장실 인덱스
                report: String, //신고 내용
            } //length 만큼 반복
        ]
    }
}
```

6. 화장실 사용

6-1. 화장실 사용 내역 등록 API

URL : /used/{:toiletIdx}?id={:id}

method : POST

화장실 사용 내역을 추가한다.

(시간은 서버시간 기준으로 추가, 횟수는 직접 처리)

Request

access_token	String	헤더 or 쿼리스트링에 토큰
toiletIdx	int	화장실 idx
id	String	아이디

Response(JSON으로 응답)

```
{
    status :String, //상태(성공(S),실패(F))
    reason : String //실패시 사유, 성공시 빈값(“”)
    resultData : null
}
```

6-2. 화장실 사용내역 조회 API

URL : /used/

method : GET

화장실 사용 내역을 조회한다.

Request

access_token	String	헤더 or 쿼리스트링에 토큰
toiletIdx	int	화장실 idx
id	String	아이디

Response(JSON으로 응답)

```
{
    status :String, //상태(성공(S),실패(F))
    reason : String //실패시 사유, 성공시 빈값(“”)
    resultData : null
}
```

6-3. 출입 제어 API

URL : /conn/dopen

method : POST

특정 화장실의 문을 제어한다.

Request

access_token	String	헤더 or 쿠리스트링에 토큰
beaconId	String	비콘 아이디
ndPwd	int	2자 비밀번호

Response(JSON으로 응답)

```
{
    status :String, //상태(성공(S),실패(F))
    reason : String //실패시 사유, 성공시 빈값(“”)
    resultData : {
        status : 화장실 요청 성공 여부(성공(S), 실패(F))
    }
}
```

6-4. 센싱 풀링 API

URL : /conn/sensing

method : POST

특정 화장실의 센싱 값을 받아온다.

Request

access_token	String	헤더 or 쿠리스트링에 토큰
beaconId	String	비콘 아이디
ndPwd	int	2차 비밀번호

Response(JSON으로 응답)

```
{
    status :String, //상태(성공(S),실패(F))
    reason : String //실패시 사유, 성공시 빈값(“”)
    resultData : {
        status : 화장실 요청 성공 여부(성공 : “S”, 실패 : “F”)
        tem : double
        hum : double
        tissue : integer
        count : integer
        room : integer
    }
}
```

7. 화장실

7-1. 센싱 API

URL : /sensing?data={:gender}

method : POST

현재 화장실의 센싱 값을 반환한다.

Request

access_token	String	헤더 or 큐리스트링에 토큰
gender	String	성별

Response(JSON으로 응답)

```
{
    status : 성공 여부(성공 : “S” , 실패 : “F” )
    tem : double
    hum : double
    tissue : integer
    count : integer
    room : integer
}
```

7-2. 출입 API

URL : /dopen?data={:gender}

method : GET

현재 화장실의 문을 제어한다.

Request

access_token	String	헤더 or 쿼리스트링에 토큰
gender	String	성별

Response(JSON으로 응답)

{ status : 성공 여부(성공 : “S” , 실패 : “F”) }
--