

생활관 별 휴대폰 반납 시스템 맡기시오

ERICSSON
9630 B771781
VP27137A
R2A

2020 군장병 공개SW 온라인 해커톤 발표자료
팀 797은거꾸로해도797
공군 병장 김정현, 김상윤

목차

1. 팀소개	3
2. 개발 동기	4~7
3. 핵심 기능	8
4. 설계	9~18
5. 시연영상	19
6. 개발 중점	20
7. 차별성과 장점	21
8. 목표 시장과 기대 효과	22
9. 애로 사항과 개선할 점	23

팀 소개

797은거꾸로해도797



797 Is Palindrome

팀원의 입대 기수인 797을 유머러스하게 풀어낸 팀명입니다.

김상운

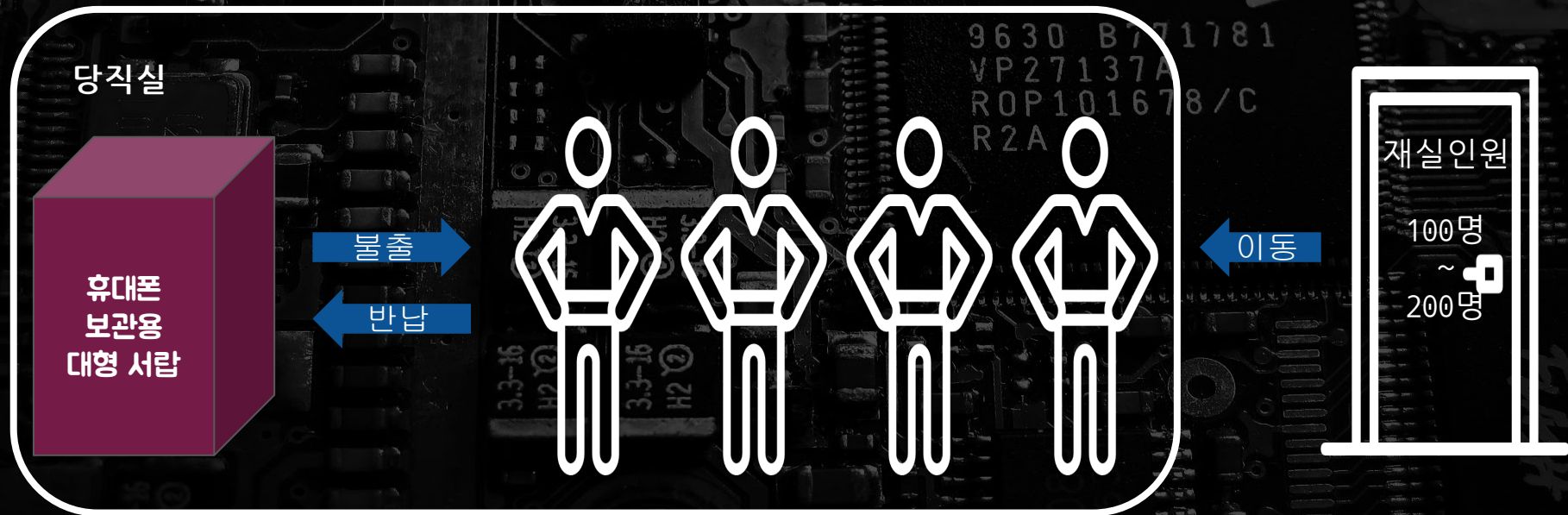
- 팀원
- IoT 분야 담당
- 하드웨어, IoT 통신
- Github @ndkim11

김정현

- 팀장
- Web 분야 담당
- 웹, API 설계
- Github @Dictor

개발 동기 : 현재 방식

현재 휴대폰 불출/반납 방식은 다음 그림과 같습니다:



개발 동기 : 현재 방식의 문제점

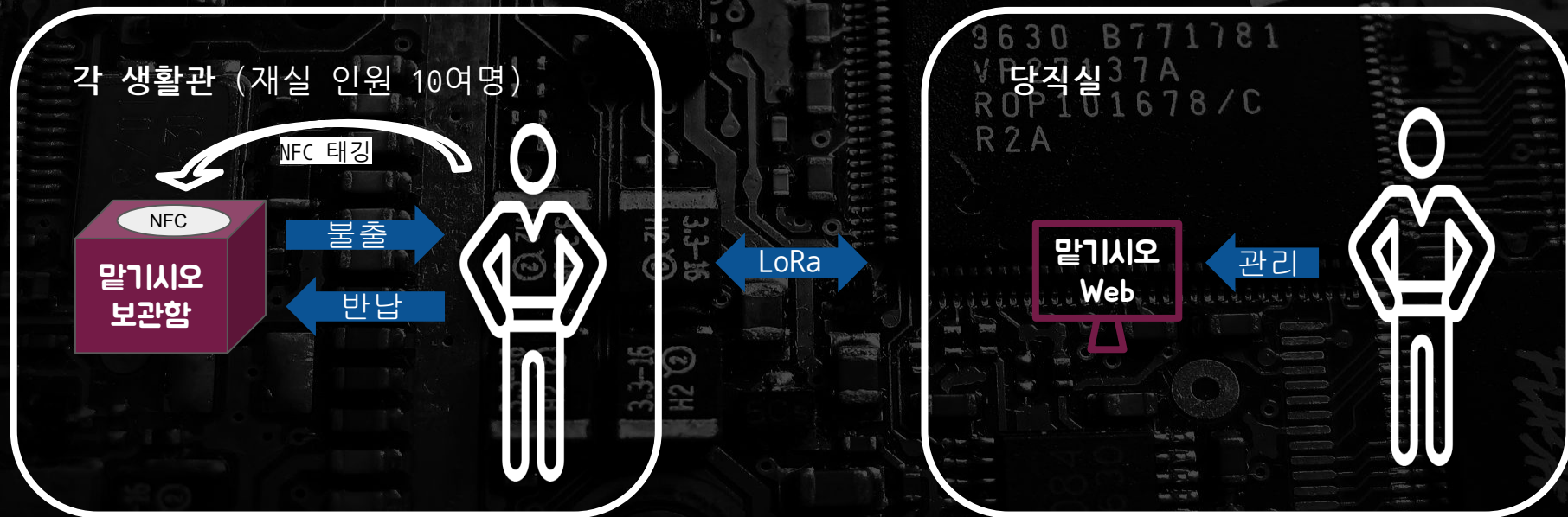
1. 하나의 보관함에 수백개의 휴대폰이 보관되면서 발생하는 파손, 분실의 위험 및 관리 주체의 모호함
2. 수백명의 인원이 불출/반납을 위해 한 장소에 밀집되면서 발생하는 전염의 위험성
3. 여러 근무 형태의 각기 다른 휴대폰 사용 시간에 따라 불출/반납을 통제하기 어려움 (감시/관리 인력 필요)

개발 동기 : 해결 방안

1. 파손, 분실의 위험 및 관리 주체의 모호 → 휴대폰 소유자들의 생활 공간에 보관함을 설치하고 관리 주체 설정
2. 전염의 위험성 → 반납 단위를 생활관 전체가 아닌, 최소 생활 단위(10여명)인 생활관으로 쪼개고 반납 장치를 배치
3. 불출/반납 통제의 어려움 → 휴대폰 이용 관리 자동화 및 위반 사항등 핵심 사항만 보고

개발 동기 : 해결 방안

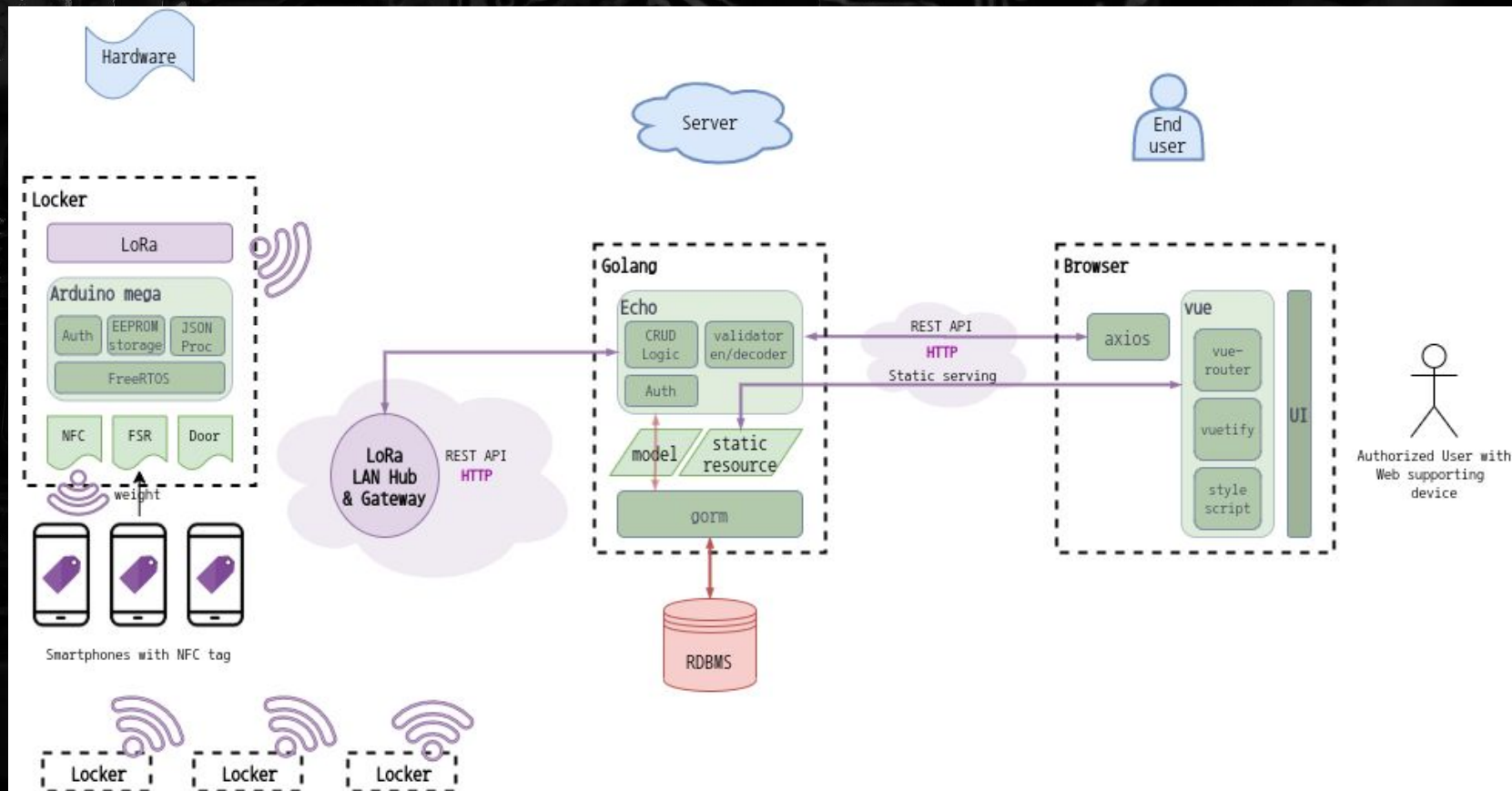
말기시오가 제안하는 불출/반납 방식은 다음과 같습니다:



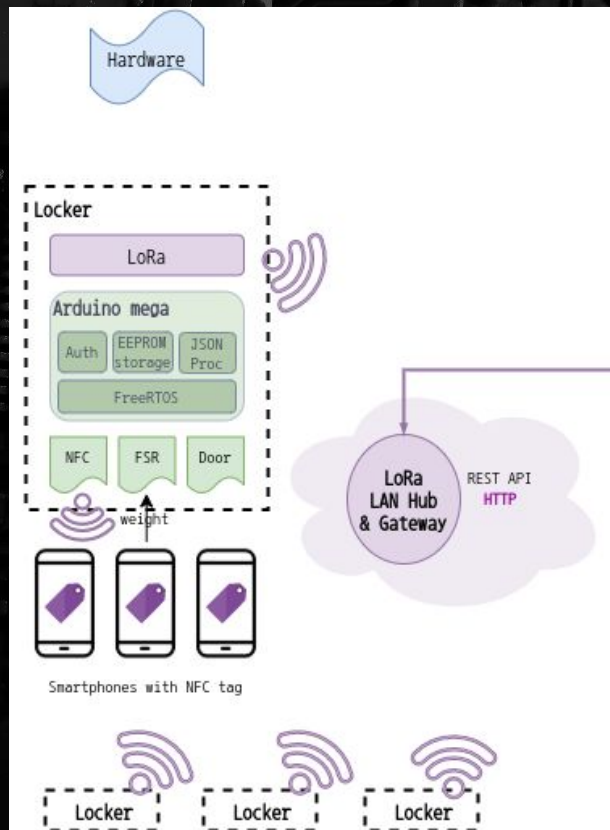
핵심 기능

1. 10여개의 휴대폰을 보관할 수 있고 생활관 단위로 설치할 수 있는 보관함 → 3D 프린팅, 오픈소스 H/W로 구현되어 간단하고 저렴
2. 휴대폰 정보와 이용시간 정보를 수신하고 중앙에 반납 상태를 주기적으로 발신하는 통신 기능 → IoT LoRa망 채택, 서버와 HTTP REST API로 통신
3. 악용을 방지하기 위해 반납된 휴대폰을 고유하게 구별하고 불출/반납 행위를 통제하는 기능 → NFC 태그, 무게 센서, 문개폐 센서 탑재
4. 중앙에서 반납 상태와 정보를 관리할 수 있는 체계 → Web 기반 관리 체계 제공

설계 : 전체 시스템

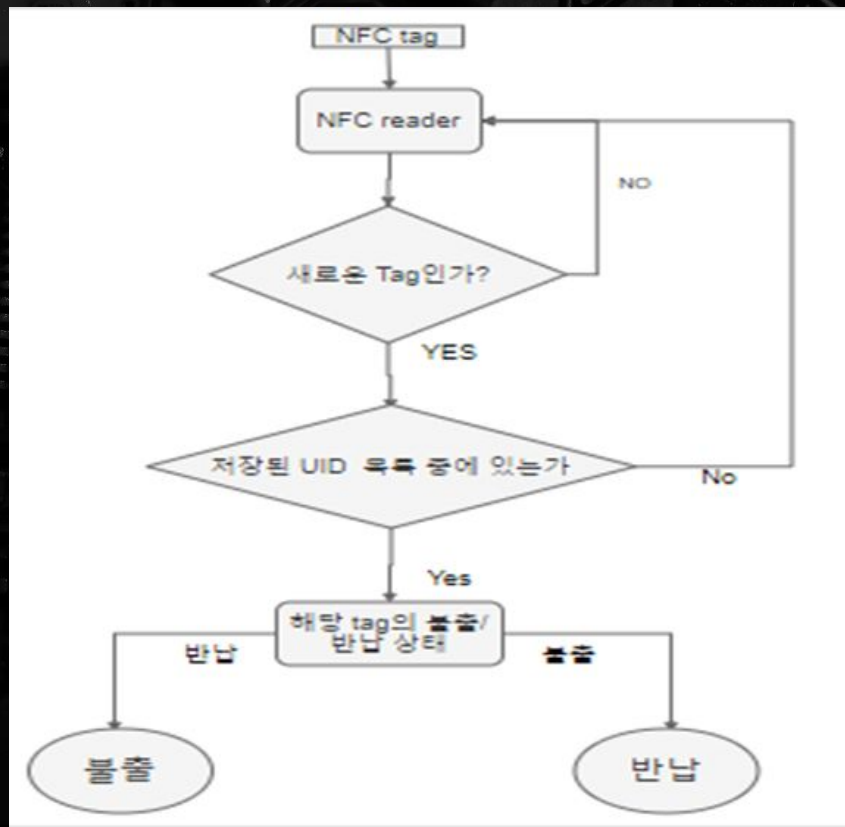


설계 : IoT



1. Locker(보관함)의 MCU는 Arduino Mega, Gateway의 MCU는 Arduino Uno로 구현
2. Locker와 Gateway간 통신은 915MHz LoRa 모듈로 UART + 자체 프로토콜을 통해, Gateway와 Server간에는 Ethernet로 HTTP + REST & JSON을 통해 통신
3. Locker는 하드웨어 수준에서 고유한 UID를 가지며 NFC 리더, FSR, 문개폐 센서의 입출력을 주기적으로 서버에 보고해 휴대폰 반납 관리
4. Locker는 의존성과 낮은 컴퓨팅 파워를 감안해 수집한 데이터 수집 및 보고만 하며 실질적 데이터 처리는 서버에서 수행

설계 : IoT 보관함 휴대폰 감지

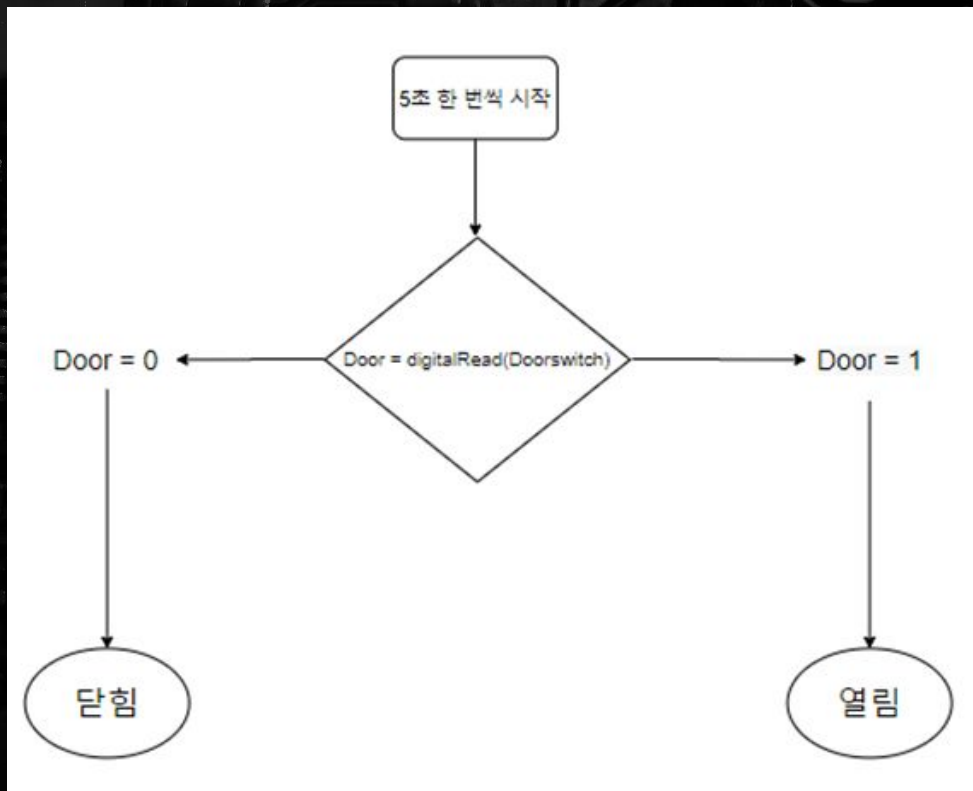


NFC 리더 RC-522

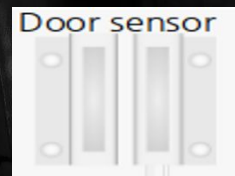


- Header : MFRC522.h
- 리더 아두이노와 SPI 통신을 통해 데이터 송수신
- NFC로 수신한 태그와 서버로부터 받은 태그를 비교하여 일치하면 자신이 유지하는 반납 상태 변경

설계 : IoT 보관함 문 개폐 감지

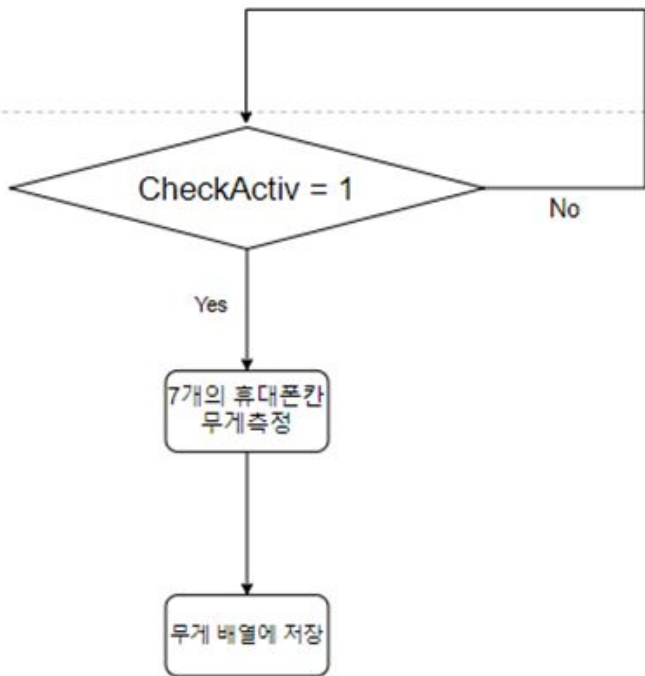


Magnetic Door Sensor



- 개폐센서는 스위치로 입력된 High/Low 전압을 디지털 I/O를 통해 읽는다.
- 5초에 한번씩 DoorSwitch에 해당하는 I/O의 전압을 읽고 상태를 갱신

설계 : IoT 보관함 무게 감지

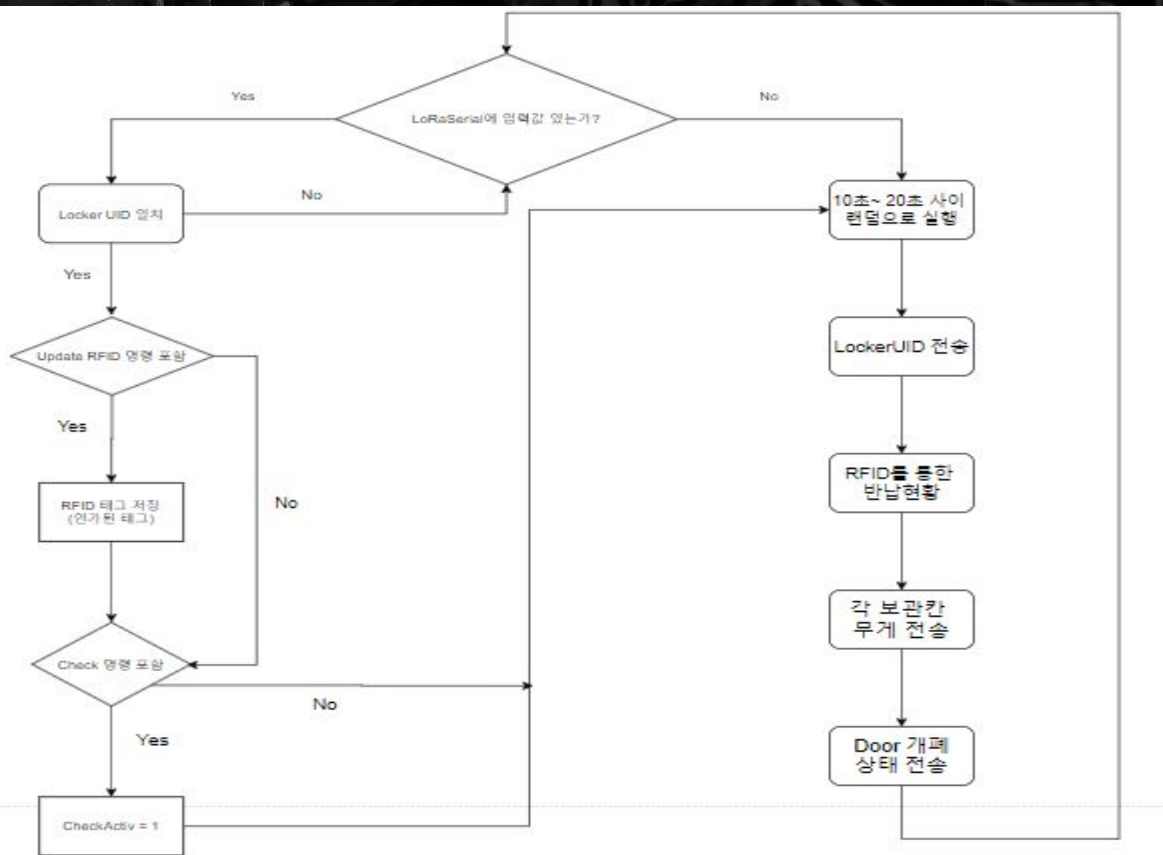


압력 감지 센서 FSR402

- Header : FSR.h
- 7개의 압력센서는 아날로그 핀을 통해 압력값을 읽는다.
- 압력 값을 배열에 저장되어 서버로 전송되어 적합/부적합을 판단한다.



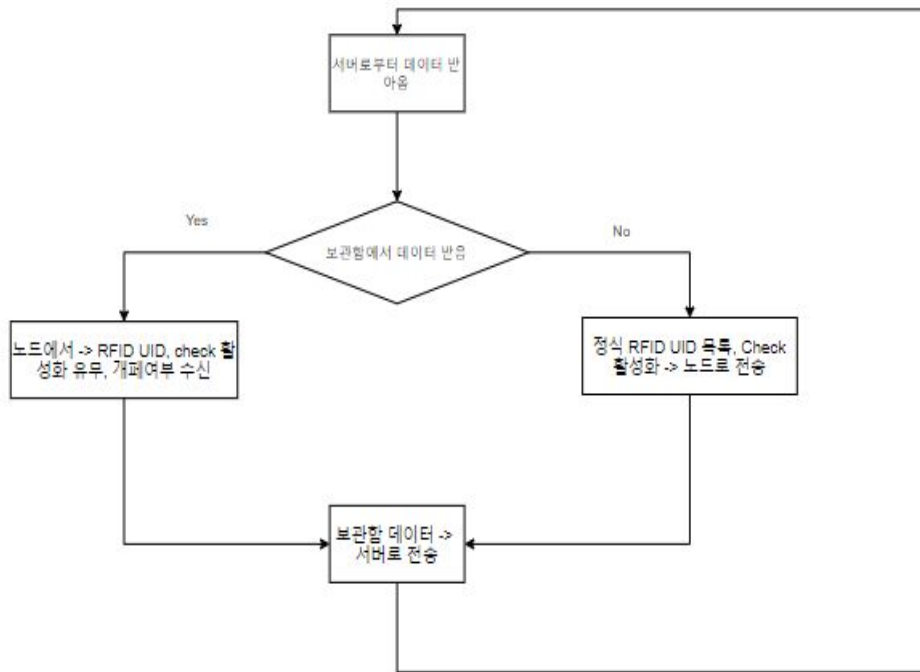
설계 : IoT 보관함 LoRa 모듈



RYLR896 (915MHz)

- LoRa 송수신기는 Serial(UART)을 통해 아두이노와 통신
- 서버에 UID를 통해 데이터를 요청하고 응답을 JSON 디코더로 parse

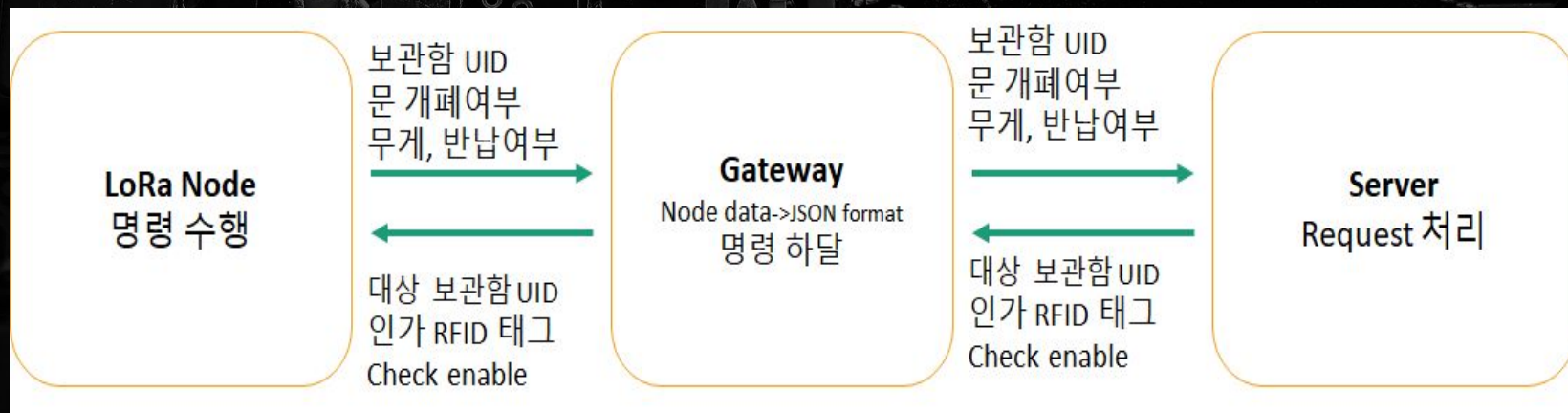
설계 : Gateway



- 구성 : UNO + LoRa antenna + Ethernet shield
- String data -> JSON format
- 노드들의 랜덤 수신으로 timesharing을 구현하여 싱글채널 문제 해결

설계 : LoRa망과 Gateway간 통신

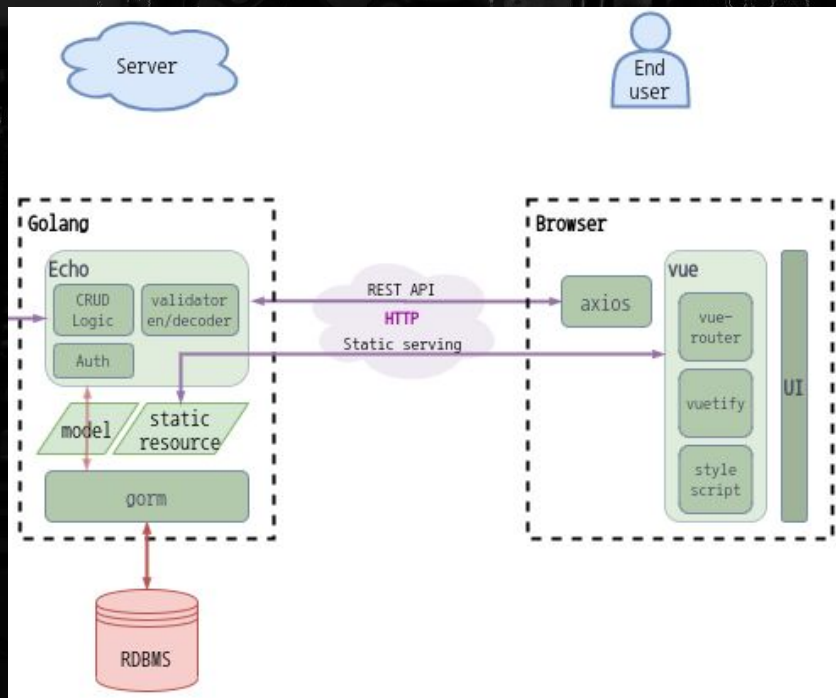
- LoRa Node(보관함)는 주기적으로 데이터 수집, 처리, 전송
- Gateway는 HTTP (REST) ↔ UART (LoRa) 메시지 변환 및 서버에 요청 중계



설계 : Gateway와 서버간 통신

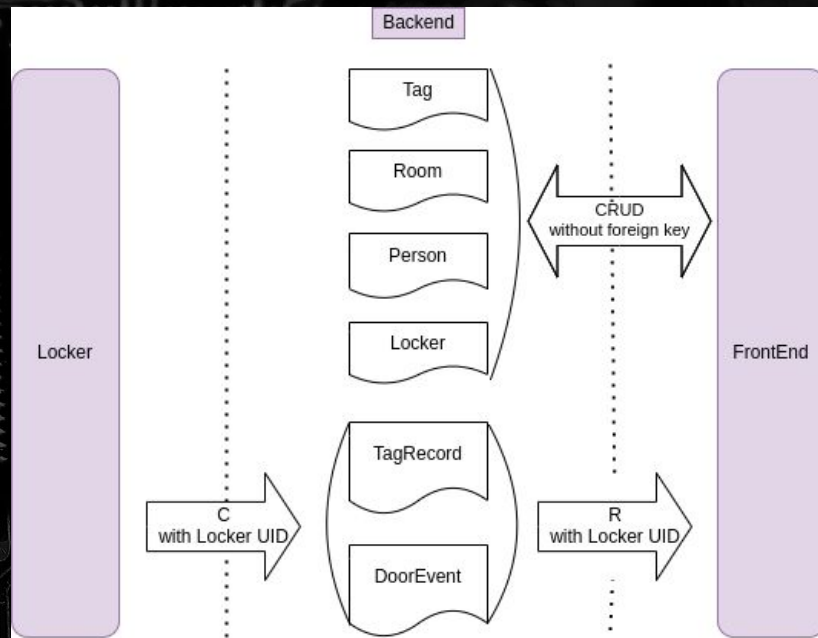
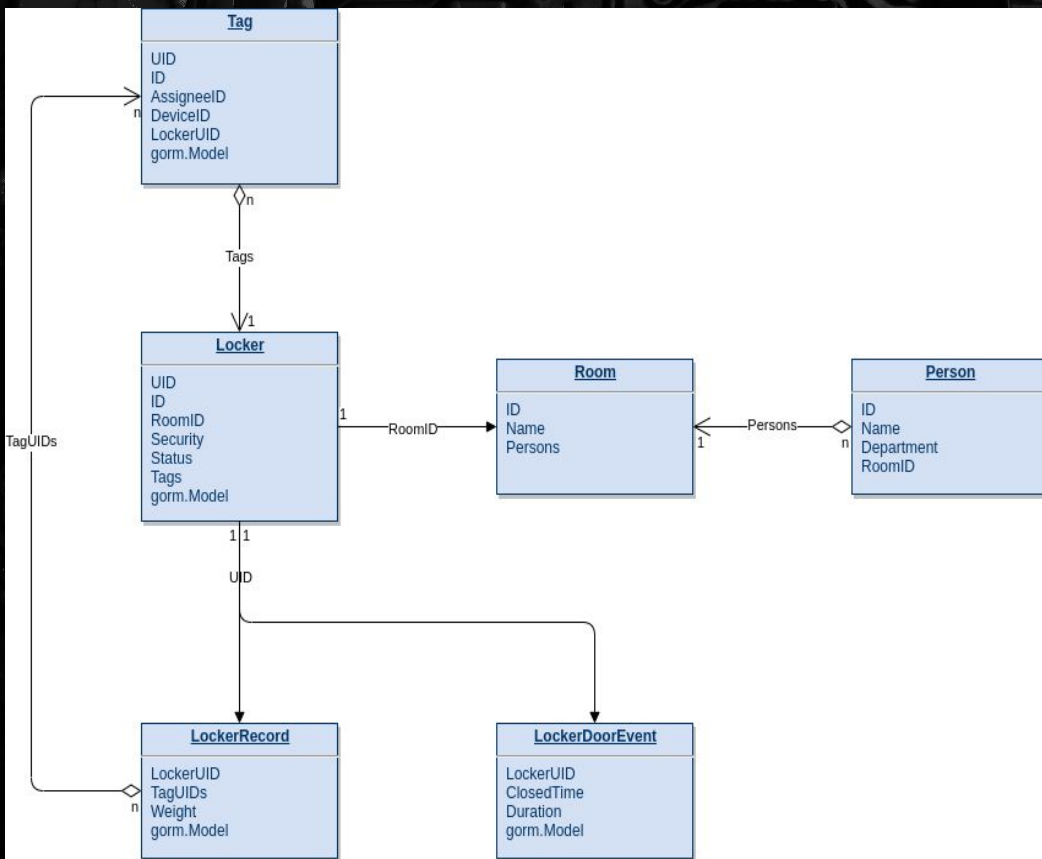
- BE는 프론트엔드를 위한 14개의 엔드포인트 외에, 라커와의 통신을 위한 현재 서버 시간 얻기, 반납된 휴대폰 현황 기록, 문 개폐 이벤트 기록, 보관함 자신에게 할당된 태그 목록 얻기 4개의 엔드포인트를 노출하고 있음 (route.go)
- 데이터를 정의된 모델의 형식에 맞게(model.go) JSON으로 인코딩해 HTTP 요청
- 모든 쓰기(POST) 요청은 보관함의 자격을 증명하기 위해 미리 설정된 비밀키를 이용한 HMAC 해시를 첨부하여 서명

설계 : Web



1. 프론트엔드(FE)는 Vue, 백엔드 (BE)는 golang으로 구현
2. FE와 BE가 REST API로만 상호작용하며 독립적인 CSR 방식
3. BE는 정의된 모델을 이용해 일관적인 CRUD API를 노출하며 ORM을 채택해 DB 종류에 의존없음
4. BE는 echo 프레임워크와 미들웨어를 통해 인증, 로깅, 입력 검증등을 수행
5. FE는 SPA 방식과 머티리얼 디자인 프레임워크 vuetify를 채택

설계 : Web 모델 설계



▲ 모델과 주체간 API 관계도

◀ 각 모델간 연관 관계

시연 영상

<https://www.youtube.com/watch?v=-X205mPk3jw>

개발 중점

다음은 설계와 구현 과정에서 중점적으로 고려한 사항입니다:

1. 대대급 생활관(100명 내외) 기준, 서버에 분당 수십개 이상의 요청이 예상되므로 서버의 충분한 수용능력을 확보 (테이블 전체 읽기 요청 35개/s, 현황 쓰기 요청 85개/s, 단순 읽기 요청 340개/s)
2. LoRa망에서 통신 보안을 확보하기 위해 게이트웨이를 두어 폐쇄망을 구성하고 별도 자격 증명 방식을 설정
3. 동시 통신이 불가능한 LoRa망 특성상 자체 시분할 메커니즘을 구현
4. 저대역폭인 군 회선을 고려하여 SPA, API 응답의 페이징 방식을 이용해 데이터 전송량을 줄이고 다양한 단말에 적용할 수 있는 UI 반응성, 이식성 확보

차별점과 강점

“맡기시오” 프로젝트가 현행 및 다른 방식에 대해 가지는 장점은:

1. 휴대폰에 물리적으로 부착된 등록증의 NFC 태그로만 구별 → 별도의 앱 설치 없음, 개인정보와 호환 문제에서 자유로움
2. 카메라를 사용하는 방식이 아닌, 더 간단한 방식의 센서 사용 → a)보안상 안전하고 b)유지보수가 더 용이 c) 설치비용 저렴
3. 휴대폰 반납/출입 기록을 전산화 할 수 있으며 기타 정보통신장비 인가 체계를 맡기시오에 통합가능 → 체계 유지보수 용이 및 행정력 절약

목표 시장과 기대 효과

- 목표시장은 병사 휴대폰 사용을 시행하는 전 군부대와 휴대폰 소유 통제가 필요한 기업체
- 주요 목표인 군 부대에 적용하여, 현재 발생하는 휴대폰 관련 파손/도난 분쟁과 관리 인력 소요 및 부담을 획기적으로 절감 가능
- 독립적으로 구성 가능한 API와 LoRa망을 채택하여 하드웨어와 웹 체계간 의존성이 적어 상용 체계 적용이 제한되는 군 부대 상황에서 유연하게 설치/관리 가능

애로 사항과 개선할 점

1. Web의 경우 해커톤 특성 상 최소기능제품(MVP) 수준으로 개발되어 웹 인증, HMAC 인증 미구현
2. 시간 부족으로 UI의 부가기능 (모델간 연관 관계를 통한 이동 기능 등등) 개발 제한
3. 보관함의 데이터를 송신주기가 일정하지 않아 노드마다 데이터가 총 업데이트되는 횟수가 달라질 수 있다 → Time sharing 방식 개선 필요
4. 보관함의 프로그램과 API 경량화로 문을 개방한 인원의 식별하는 기능까지는 구현이 제한

감사합니다!

https://github.com/osamhack2020/WEB_LeaveItUs_797IsPalindrome/
https://github.com/osamhack2020/IoT_LeaveItUs_797IsPalindrome/

아이콘 제작자 Smashicons from www.flaticon.com