



(19) 대한민국특허청(KR)
(12) 공개특허공보(A)

(11) 공개번호 10-2018-0099149
(43) 공개일자 2018년09월05일

(51) 국제특허분류(Int. Cl.)

G07C 9/00 (2006.01)

(52) CPC특허분류

G07C 9/00071 (2013.01)

(21) 출원번호 10-2017-0026204

(22) 출원일자 2017년02월28일

심사청구일자 2017년02월28일

(71) 출원인

인천대학교 산학협력단

인천광역시 연수구 아카데미로 119 (송도동)

(72) 발명자

이은규

인천광역시 연수구 해돋이로120번길 16, 213동
1002호 (송도동, 송도풍림아이원2단지아파트)

황건주

인천광역시 계양구 계산새로 109, 324동 1901호
(용종동, 초정마을동아아파트)

(뒷면에 계속)

(74) 대리인

특허법인충정

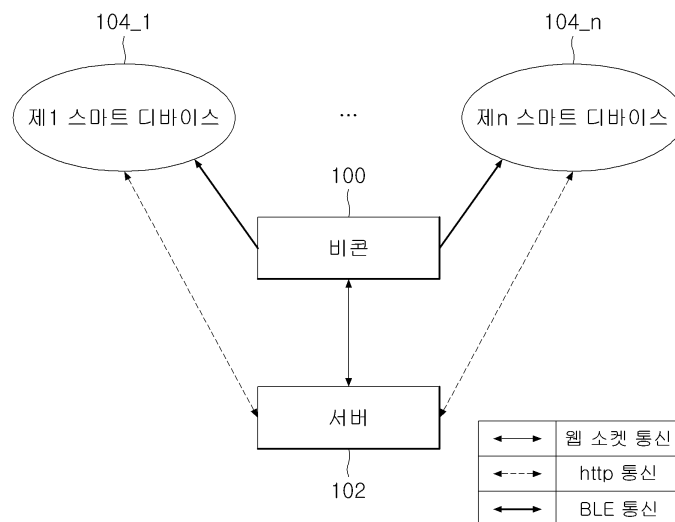
전체 청구항 수 : 총 12 항

(54) 발명의 명칭 블루투스과 스마트 디바이스 지문인식 센서를 이용한 출석 체크 시스템 및 방법

(57) 요약

본 발명의 일 실시예에 의한 블루투스과 스마트 디바이스 지문인식 센서를 이용한 출석 체크 시스템은, 서버에 강의실과 관련된 고유값을 전송하고 상기 서버로부터 상기 고유값에 해당하는 강의 UUID(universal unique identification)를 수신하며, 수신된 강의 UUID에 기반하여 애드버타이징 신호를 송출하는 비콘; 출석자가 소지하는 스마트 디바이스로서, 지문인식 센서를 포함하여 출석자의 지문인증을 수행하며, 상기 비콘으로부터 수신된 강의 UUID에 기반하여 상기 서버에 접속하는 스마트 디바이스; 및 상기 강의 UUID, 강의 정보, 출석 정보, 및 출석자 ID를 저장하는 데이터베이스를 포함하며, 상기 스마트 디바이스에서 출석자의 지문 인증이 성공하는 경우 상기 스마트 디바이스로부터 수신한 출석자 ID 및 강의 UUID에 기반하여 출석 체크를 수행하는 서버를 포함한다.

대표도 - 도1



(72) 발명자

오해남

인천광역시 남동구 문화서로3번길 28-9, 201호 (구 월동)

태호영

인천광역시 부평구 원적로 344, 208동 1901호 (산곡동, 금호 이수 마운트밸리)

이 발명을 지원한 국가연구개발사업

과제고유번호 201601750001

부처명 미래창조과학부

연구관리전문기관 한국연구재단

연구사업명 신진연구지원사업

연구과제명 차세대 IoT 서비스 실현을 위한 엣지 컴퓨팅과 데이터 네트워킹 기반의 엣지 네트워크 시스템 개발에 관한 연구

기 여 율 1/1

주관기관 인천대학교

연구기간 2016.06.01 ~ 2019.05.31

명세서

청구범위

청구항 1

서버에 강의실과 관련된 고유값을 전송하고 상기 서버로부터 상기 고유값에 해당하는 강의 UUID(universal unique identification)를 수신하며, 수신된 강의 UUID에 기반하여 애드버타이징 신호를 송출하는 비콘;

출석자가 소지하는 스마트 디바이스로서, 지문인식 센서를 포함하여 출석자의 지문인증을 수행하며, 상기 비콘 으로부터 수신된 강의 UUID에 기반하여 상기 서버에 접속하는 스마트 디바이스; 및

상기 강의 UUID, 강의 정보, 출석 정보, 및 출석자 ID를 저장하는 데이터베이스를 포함하며, 상기 스마트 디바이스에서 출석자의 지문 인증이 성공하는 경우 상기 스마트 디바이스로부터 수신한 출석자 ID 및 강의 UUID에 기반하여 출석 체크를 수행하는 서버를 포함하는, 블루투스와 스마트 디바이스 지문인식 센서를 이용한 출석 체크 시스템.

청구항 2

청구항 1에 있어서,

상기 스마트 디바이스는 상기 비콘으로부터 수신된 강의 UUID 및 상기 강의실과 관련된 고유값을 결합하여 상기 서버에 전송하고, 상기 서버는 인터넷주소를 상기 스마트 디바이스에 제공하여 상기 스마트 디바이스에서 웹브라우저에 기반한 출석 체크 어플리케이션이 실행되도록 하는, 블루투스와 스마트 디바이스 지문인식 센서를 이용한 출석 체크 시스템.

청구항 3

청구항 2에 있어서,

상기 어플리케이션이 처음 실행되는 경우, 상기 어플리케이션의 고유한 어플리케이션 UUID가 랜덤하게 생성되어 상기 스마트 디바이스에 저장되고, 상기 어플리케이션 UUID는 상기 서버로 전송되어 상기 데이터베이스에 저장되며,

상기 서버는, 상기 스마트 디바이스로부터 수신된 어플리케이션 UUID 및 출석자 ID가 상기 데이터베이스에 저장된 어플리케이션 UUID 및 출석자 ID와 일치하는 경우에만, 상기 스마트 디바이스에서 상기 출석 체크 어플리케이션이 실행되도록 하고, 상기 스마트 디바이스로부터 수신된 어플리케이션 UUID 및 출석자 ID가 상기 데이터베이스에 저장된 어플리케이션 UUID 및 출석자 ID와 일치하지 않는 경우에는 상기 출석자를 불량 사용자로 등록하고 상기 어플리케이션을 이용할 수 없게 하는, 블루투스와 스마트 디바이스 지문인식 센서를 이용한 출석 체크 시스템.

청구항 4

청구항 3에 있어서,

상기 어플리케이션의 삭제 및 변경 시 상기 어플리케이션 UUID가 삭제되는, 블루투스와 스마트 디바이스 지문인식 센서를 이용한 출석 체크 시스템.

청구항 5

청구항 1에 있어서,

상기 서버는 처음 가동 시 상기 강의 UUID 값을 랜덤하게 생성하여 상기 데이터베이스에 저장하는, 블루투스와 스마트 디바이스 지문인식 센서를 이용한 출석 체크 시스템.

청구항 6

청구항 5에 있어서,

상기 서버는 상기 강의 UUID를 소정 시간마다 랜덤하게 재생성하여 상기 강의 UUID를 초기화하는, 블루투스 및 스마트 디바이스 지문인식 센서를 이용한 출석 체크 시스템.

청구항 7

- (a) 강의실에 설치된 비콘이 서버에 상기 강의실과 관련된 고유값을 전송하는 단계;
- (b) 상기 서버가 상기 비콘으로 상기 고유값에 해당하는 강의 UUID를 전송하는 단계;
- (c) 상기 비콘이 상기 서버로부터 수신된 상기 고유값에 해당하는 강의 UUID에 기반하여 애드버타이징 신호를 송출하는 단계;
- (d) 출석자가 상기 강의실에 들어가는 경우, 상기 출석자가 소지하는 스마트 디바이스가 상기 비콘으로부터 상기 강의 UUID를 수신하는 단계;
- (e) 상기 스마트 디바이스에서 지문 인증을 실시하는 단계;
- (f) 상기 지문 인증이 성공하는 경우, 상기 스마트 디바이스가 출석자 ID 및 상기 강의 UUID를 상기 서버로 전송하여 출석 체크를 요청하는 단계; 및
- (g) 상기 서버가 상기 스마트 디바이스로부터 수신한 출석자 ID 및 강의 UUID에 기반하여 출석 체크를 수행하는 단계를 포함하는, 블루투스 및 스마트 디바이스 지문인식 센서를 이용한 출석 체크 방법.

청구항 8

청구항 7에 있어서,

상기 단계 (c) 이후에,

상기 스마트 디바이스가 상기 비콘으로부터 수신된 강의 UUID 및 상기 강의실과 관련된 고유값을 결합하여 상기 서버에 전송하는 단계; 및

상기 서버가 인터넷주소를 상기 스마트 디바이스에 제공하여 상기 스마트 디바이스에서 웹브라우저에 기반한 출석 체크 어플리케이션이 실행되도록 하는 단계를 더 포함하고,

상기 단계 (d)는 출석자가 상기 강의실에 들어가는 경우, 상기 출석 체크 어플리케이션을 통해 상기 스마트 디바이스가 상기 비콘으로부터 상기 강의 UUID를 수신하는 단계를 포함하며,

상기 단계 (e)는 상기 출석 체크 어플리케이션을 통해 상기 스마트 디바이스에서 지문 인증을 실시하는 단계를 포함하는, 블루투스 및 스마트 디바이스 지문인식 센서를 이용한 출석 체크 방법.

청구항 9

청구항 8에 있어서,

상기 어플리케이션이 처음 실행되는 경우, 상기 어플리케이션의 고유한 어플리케이션 UUID가 랜덤하게 생성되어 상기 스마트 디바이스에 저장되고, 상기 어플리케이션 UUID는 상기 서버로 전송되어 상기 데이터베이스에 저장되며,

상기 서버는, 상기 스마트 디바이스로부터 수신된 어플리케이션 UUID 및 출석자 ID가 상기 데이터베이스에 저장된 어플리케이션 UUID 및 출석자 ID와 일치하는 경우에만, 상기 스마트 디바이스에서 상기 출석 체크 어플리케이션이 실행되도록 하고, 상기 스마트 디바이스로부터 수신된 어플리케이션 UUID 및 출석자 ID가 상기 데이터베이스에 저장된 어플리케이션 UUID 및 출석자 ID와 일치하지 않는 경우에는 상기 출석자를 불량 사용자로 등록하고 상기 어플리케이션을 이용할 수 없게 하는, 블루투스 및 스마트 디바이스 지문인식 센서를 이용한 출석 체크 방법.

청구항 10

청구항 9에 있어서,

상기 어플리케이션의 삭제 및 변경 시 상기 어플리케이션 UUID가 삭제되는, 블루투스 및 스마트 디바이스 지문인식 센서를 이용한 출석 체크 방법.

청구항 11

청구항 7에 있어서,

상기 서버는 처음 가동 시 상기 강의 UUID 값을 랜덤하게 생성하여 상기 데이터베이스에 저장하는, 블루투스 및 스마트 디바이스 지문인식 센서를 이용한 출석 체크 방법.

청구항 12

청구항 11에 있어서,

상기 서버는 상기 강의 UUID를 소정 시간마다 랜덤하게 재생성하여 상기 강의 UUID를 초기화하는, 블루투스 및 스마트 디바이스 지문인식 센서를 이용한 출석 체크 방법.

발명의 설명

기술 분야

[0001] 본 발명은 블루투스 및 스마트 디바이스 지문인식 센서를 이용한 출석 체크 시스템 및 방법에 관한 것이다.

배경 기술

[0002] 현대 사회에서 기술이 발전함에 따라 삶의 편리성이 높아지고 있다. 그러나 아직도 많은 대학이나 회사, 학교 등은 마그넷 카드 같은 추가 물품을 사용하거나 아날로그식으로 출석 체크를 하고 있다.

[0003] 한편, 전자출결시스템(Electronic attendance-absence recording systems)이란 많은 수강인원이 참석하는 오프라인의 강의실 수업에서 기존의 호명식 출결처리 대신 지문 또는 RFID, 블루투스에 기반하여 신속하고 정확하게 출결처리를 하는 최적의 시스템이다.

[0004] 주요기능으로 강의시간 설정, 과목별 출석현황과 학생출석상태를 확인할 수 있는 전자출석부, 출석기록조회, 출석통계를 이용한 다양한 분석, 학사성적연계 등이 있다.

[0005] 기대효과로는 출결시스템과 학사시스템의 통합으로 대학교 학사시스템 통합관리, 정확한 출결관리를 통한 학생들의 수업참여도 향상, 대리출석 방지를 통한 학생들의 불만 최소화, 신속한 출결처리로 강의시간 확보, 대단위 강의실 출결 자동체크, 학생출결관리에 소모되는 인력감축, 학교 이미지 향상 등이 있다.

[0006] 시스템의 구성은 전자출결서버를 중심으로 출결단말기, 출석대상, 학사운영시스템을 동기화하여, 웹을 통해 실시간 출결상황을 조회할 수 있도록 되어있다. 최근에는 학생증 단말기 접촉 후 수업을 불참하거나, 한 학생이 여러 학생증을 이용하여 대리출석이 가능한 단점이 있어, RF 단말기에 카메라와 위조지문 판별센서를 장착하고, 학생이 강의실을 입퇴장할 시 RF단말기와 반응하도록 하는 등의 여러 기술이 접목되고 있다.

[0007] 최근에는 스마트폰을 이용한 온라인 출결 관리 시스템이 등장하여, 기존 출결 시스템의 높은 초기 투자 비용 등의 단점을 해결하였다.

[0008] 특허문헌1은 스마트폰을 이용하여 시간과 공간에 대한 검증을 함으로써, 수강자가 강의실 내부에 있음을 증명하여 대리출석을 방지하고, 병목현상 없이 빠른 시간 내에 출석 인증을 수행할 수 있는, 와이파이 비콘 프레임을 이용한 스마트 디바이스 기반 전자 출결 시스템의 인증 방법을 개시하고 있다.

선행기술문헌

특허문헌

[0009] (특허문헌 0001) KR 10-1425345 B1

발명의 내용

해결하려는 과제

[0010] 본 발명이 해결하고자 하는 과제는 개개인의 스마트 디바이스를 활용하므로 시스템의 도입 비용을 감소시킬 수

있고, 지문인식을 사용하므로 보안성을 강화하여 대리 출석 등 부정 출결을 방지할 수 있으며, 비콘을 활용함으로써 일정한 공간 내에서만 출석 체크가 가능하도록 할 수 있고, 일정시간마다 비콘의 UUID 값을 변경함으로써 타 장소에서의 부정출석을 방지할 수 있으며 서버의 보안성을 강화할 수 있는, 블루투스 및 스마트 디바이스 지문인식 센서를 이용한 출석 체크 시스템을 제공하는 것이다.

- [0011] 본 발명이 해결하고자 하는 다른 과제는 개개인의 스마트 디바이스를 활용하므로 시스템의 도입 비용을 감소시킬 수 있고, 지문인식을 사용하므로 보안성을 강화하여 대리 출석 등 부정 출결을 방지할 수 있으며, 비콘을 활용함으로써 일정한 공간 내에서만 출석 체크가 가능하도록 할 수 있고, 일정시간마다 비콘의 UUID 값을 변경함으로써 타 장소에서의 부정출석을 방지할 수 있으며 서버의 보안성을 강화할 수 있는, 블루투스 및 스마트 디바이스 지문인식 센서를 이용한 출석 체크 방법을 제공하는 것이다.

과제의 해결 수단

- [0012] 상기 과제를 해결하기 위한 본 발명의 일 실시예에 의한 블루투스 및 스마트 디바이스 지문인식 센서를 이용한 출석 체크 시스템은,
- [0013] 서버에 강의실과 관련된 고유값을 전송하고 상기 서버로부터 상기 고유값에 해당하는 강의 UUID(universal unique identification)를 수신하며, 수신된 강의 UUID에 기반하여 애드버타이징 신호를 송출하는 비콘;
- [0014] 출석자가 소지하는 스마트 디바이스로서, 지문인식 센서를 포함하여 출석자의 지문인증을 수행하며, 상기 비콘으로부터 수신된 강의 UUID에 기반하여 상기 서버에 접속하는 스마트 디바이스; 및
- [0015] 상기 강의 UUID, 강의 정보, 출석 정보, 및 출석자 ID를 저장하는 데이터베이스를 포함하며, 상기 스마트 디바이스에서 출석자의 지문 인증이 성공하는 경우 상기 스마트 디바이스로부터 수신한 출석자 ID 및 강의 UUID에 기반하여 출석 체크를 수행하는 서버를 포함한다.
- [0016] 본 발명의 일 실시예에 의한 블루투스 및 스마트 디바이스 지문인식 센서를 이용한 출석 체크 시스템에 있어서, 상기 스마트 디바이스는 상기 비콘으로부터 수신된 강의 UUID 및 상기 강의실과 관련된 고유값을 결합하여 상기 서버에 전송하고, 상기 서버는 인터넷주소를 상기 스마트 디바이스에 제공하여 상기 스마트 디바이스에서 웹브라우저에 기반한 출석 체크 어플리케이션이 실행되도록 할 수 있다.
- [0017] 또한, 본 발명의 일 실시예에 의한 블루투스 및 스마트 디바이스 지문인식 센서를 이용한 출석 체크 시스템에 있어서, 상기 어플리케이션이 처음 실행되는 경우, 상기 어플리케이션의 고유한 어플리케이션 UUID가 랜덤하게 생성되어 상기 스마트 디바이스에 저장되고, 상기 어플리케이션 UUID는 상기 서버로 전송되어 상기 데이터베이스에 저장되며,
- [0018] 상기 서버는, 상기 스마트 디바이스로부터 수신된 어플리케이션 UUID 및 출석자 ID가 상기 데이터베이스에 저장된 어플리케이션 UUID 및 출석자 ID와 일치하는 경우에만, 상기 스마트 디바이스에서 상기 출석 체크 어플리케이션이 실행되도록 하고, 상기 스마트 디바이스로부터 수신된 어플리케이션 UUID 및 출석자 ID가 상기 데이터베이스에 저장된 어플리케이션 UUID 및 출석자 ID와 일치하지 않는 경우에는 상기 출석자를 불량 사용자로 등록하고 상기 어플리케이션을 이용할 수 없게 할 수 있다.
- [0019] 또한, 본 발명의 일 실시예에 의한 블루투스 및 스마트 디바이스 지문인식 센서를 이용한 출석 체크 시스템에 있어서, 상기 어플리케이션의 삭제 및 변경 시 상기 어플리케이션 UUID가 삭제될 수 있다.
- [0020] 또한, 본 발명의 일 실시예에 의한 블루투스 및 스마트 디바이스 지문인식 센서를 이용한 출석 체크 시스템에 있어서, 상기 서버는 처음 가동 시 상기 강의 UUID 값을 랜덤하게 생성하여 상기 데이터베이스에 저장할 수 있다.
- [0021] 또한, 본 발명의 일 실시예에 의한 블루투스 및 스마트 디바이스 지문인식 센서를 이용한 출석 체크 시스템에 있어서, 상기 서버는 상기 강의 UUID를 소정 시간마다 랜덤하게 재생성하여 상기 강의 UUID를 초기화할 수 있다.
- [0022] 상기 다른 과제를 해결하기 위한 본 발명의 일 실시예에 의한 블루투스 및 스마트 디바이스 지문인식 센서를 이용한 출석 체크 방법은,
- [0023] (a) 강의실에 설치된 비콘이 서버에 상기 강의실과 관련된 고유값을 전송하는 단계;
- [0024] (b) 상기 서버가 상기 비콘으로 상기 고유값에 해당하는 강의 UUID를 전송하는 단계;
- [0025] (c) 상기 비콘이 상기 서버로부터 수신된 상기 고유값에 해당하는 강의 UUID에 기반하여 애드버타이징 신호를

송출하는 단계;

- [0026] (d) 출석자가 상기 강의실에 들어가는 경우, 상기 출석자가 소지하는 스마트 디바이스가 상기 비콘으로부터 상기 강의 UUID를 수신하는 단계;
- [0027] (e) 상기 스마트 디바이스에서 지문 인증을 실시하는 단계;
- [0028] (f) 상기 지문 인증이 성공하는 경우, 상기 스마트 디바이스가 출석자 ID 및 상기 강의 UUID를 상기 서버로 전송하여 출석 체크를 요청하는 단계; 및
- [0029] (g) 상기 서버가 상기 스마트 디바이스로부터 수신한 출석자 ID 및 강의 UUID에 기반하여 출석 체크를 수행하는 단계를 포함한다.
- [0030] 본 발명의 일 실시예에 의한 블루투스 및 스마트 디바이스 지문인식 센서를 이용한 출석 체크 방법은, 상기 단계 (c) 이후에,
- [0031] 상기 스마트 디바이스가 상기 비콘으로부터 수신된 강의 UUID 및 상기 강의실과 관련된 고유값을 결합하여 상기 서버에 전송하는 단계; 및
- [0032] 상기 서버가 인터넷주소를 상기 스마트 디바이스에 제공하여 상기 스마트 디바이스에서 웹브라우저에 기반한 출석 체크 어플리케이션이 실행되도록 하는 단계를 더 포함하고,
- [0033] 상기 단계 (d)는 출석자가 상기 강의실에 들어가는 경우, 상기 출석 체크 어플리케이션을 통해 상기 스마트 디바이스가 상기 비콘으로부터 상기 강의 UUID를 수신하는 단계를 포함하며,
- [0034] 상기 단계 (e)는 상기 출석 체크 어플리케이션을 통해 상기 스마트 디바이스에서 지문 인증을 실시하는 단계를 포함할 수 있다.
- [0035] 또한, 본 발명의 일 실시예에 의한 블루투스 및 스마트 디바이스 지문인식 센서를 이용한 출석 체크 방법에 있어서, 상기 어플리케이션이 처음 실행되는 경우, 상기 어플리케이션의 고유한 어플리케이션 UUID가 랜덤하게 생성되어 상기 스마트 디바이스에 저장되고, 상기 어플리케이션 UUID는 상기 서버로 전송되어 상기 데이터베이스에 저장되며,
- [0036] 상기 서버는, 상기 스마트 디바이스로부터 수신된 어플리케이션 UUID 및 출석자 ID가 상기 데이터베이스에 저장된 어플리케이션 UUID 및 출석자 ID와 일치하는 경우에만, 상기 스마트 디바이스에서 상기 출석 체크 어플리케이션이 실행되도록 하고, 상기 스마트 디바이스로부터 수신된 어플리케이션 UUID 및 출석자 ID가 상기 데이터베이스에 저장된 어플리케이션 UUID 및 출석자 ID와 일치하지 않는 경우에는 상기 출석자를 불량 사용자로 등록하고 상기 어플리케이션을 이용할 수 없게 할 수 있다.
- [0037] 또한, 본 발명의 일 실시예에 의한 블루투스 및 스마트 디바이스 지문인식 센서를 이용한 출석 체크 방법에 있어서, 상기 어플리케이션의 삭제 및 변경 시 상기 어플리케이션 UUID가 삭제될 수 있다.
- [0038] 또한, 본 발명의 일 실시예에 의한 블루투스 및 스마트 디바이스 지문인식 센서를 이용한 출석 체크 방법에 있어서, 상기 서버는 처음 가동 시 상기 강의 UUID 값을 랜덤하게 생성하여 상기 데이터베이스에 저장할 수 있다.
- [0039] 또한, 본 발명의 일 실시예에 의한 블루투스 및 스마트 디바이스 지문인식 센서를 이용한 출석 체크 방법에 있어서, 상기 서버는 상기 강의 UUID를 소정 시간마다 랜덤하게 재생성하여 상기 강의 UUID를 초기화할 수 있다.

발명의 효과

- [0040] 본 발명의 일 실시예에 의한 블루투스 및 스마트 디바이스 지문인식 센서를 이용한 출석 체크 시스템 및 방법에 의하면, 개개인의 스마트 디바이스를 활용하므로 시스템의 도입 비용을 감소시킬 수 있고, 지문인식을 사용하므로 보안성을 강화하여 부정 출석을 방지할 수 있으며, 비콘을 활용함으로써 일정한 공간 내에서만 출석 체크가 가능하도록 할 수 있고, 일정시간마다 비콘의 UUID 값을 변경함으로써 타 장소에서의 부정출석을 방지할 수 있으며 서버의 보안성을 강화할 수 있다.

도면의 간단한 설명

- [0041] 도 1은 본 발명의 일 실시예에 의한 블루투스 및 스마트 디바이스 지문인식 센서를 이용한 출석 체크 시스템의 블록도.

도 2는 본 발명의 일 실시예에 의한 블루투스 및 스마트 디바이스 지문인식 센서를 이용한 출석 체크 방법의 흐름도.

도 3은 비콘의 동작을 설명하기 위한 흐름도.

도 4는 스마트 디바이스와 서버의 동작을 설명하기 위한 흐름도.

도 5는 어플리케이션의 초기 화면을 도시한 도면.

도 6은 비콘 탐색을 완료한 화면을 도시한 도면.

도 7은 비콘 탐색 관련 코드를 도시한 도면.

도 8은 지문 인식 안내 다이얼로그를 도시한 도면.

도 9는 다른 지문 인식 안내 다이얼로그를 도시한 도면.

도 10은 지문 인식 관련 코드를 도시한 도면.

도 11은 서버 연결 관련 코드를 도시한 도면.

도 12는 http 통신 구현 코드를 도시한 도면.

도 13은 소켓(Socket) 통신 구현 코드를 도시한 도면.

도 14는 데이터베이스의 테이블 설계도를 도시한 도면.

도 15는 비콘의 동작 코드를 도시한 도면.

발명을 실시하기 위한 구체적인 내용

[0042] 본 발명의 목적, 특정한 장점들 및 신규한 특징들은 첨부된 도면들과 연관되어지는 이하의 상세한 설명과 바람직한 실시예들로부터 더욱 명백해질 것이다.

[0043] 이에 앞서 본 명세서 및 청구범위에 사용된 용어나 단어는 통상적이고 사전적인 의미로 해석되어서는 아니되며, 발명자가 그 자신의 발명을 가장 최선의 방법으로 설명하기 위해 용어의 개념을 적절하게 정의할 수 있는 원칙에 입각하여 본 발명의 기술적 사상에 부합되는 의미와 개념으로 해석되어야 한다.

[0044] 본 명세서에서 각 도면의 구성요소들에 참조번호를 부가함에 있어서, 동일한 구성 요소들에 한해서는 비록 다른 도면상에 표시되더라도 가능한 한 동일한 번호를 가지도록 하고 있음에 유의하여야 한다.

[0045] 또한, "제1", "제2", "일면", "타면" 등의 용어는, 하나의 구성요소를 다른 구성요소로부터 구별하기 위해 사용되는 것으로, 구성요소가 상기 용어들에 의해 제한되는 것은 아니다.

[0046] 이하, 본 발명을 설명함에 있어, 본 발명의 요지를 불필요하게 흐릴 수 있는 관련된 공지 기술에 대한 상세한 설명은 생략한다.

[0047] 이하, 첨부된 도면을 참조하여 본 발명의 바람직한 실시형태를 상세히 설명하기로 한다.

[0048] 본 발명의 일 실시예에 의한 블루투스 및 스마트 디바이스 지문인식 센서를 이용한 출석 체크 시스템 및 방법에서는, 현대인이라면 대부분이 지니고 있는 스마트폰과 같은 스마트 디바이스의 지문인식 시스템을 이용하여 출석 체크를 하고자 한다. 스마트폰은 현대인이 대부분 지니고 있다. 따라서 출석을 위해 부속물을 일일이 제작할 필요도 없으므로 시스템 도입시 드는 비용이 적으며 이름을 호명하는 등의 시간 소모를 줄이고 출석 체크 과정을 간편하게 할 수 있는 이점이 있다.

[0049] 또한 지문인식 기능과 같은 생체 인증을 활용하여 보안성을 높이고 대리 출석 등 부정행위를 방지할 수 있다. 또한 지금까지 지문인식을 활용하거나 스마트폰을 활용한 출결 시스템은 있었지만 이 둘을 모두 활용한 시스템은 없었다.

[0051] 사용장비 및 기술 소개

[0052] 라즈베리파이3(Raspberrypi3: iBeacon) : 라즈베리파이 최신모델로 내부에 블루투스 4.1을 내장하고 있어서 따로 동글을 구비하지 않아도 비콘으로 사용 가능하다. 출석을 위한 장소인 강의실 등에 설치하여 해당 비콘 범위

내의 스마트폰만 출석 가능하게 한다.

- [0053] 스마트 디바이스 : 스마트폰과 같은 개인용 통신 디바이스이다. 다양한 정보의 송수신이 가능하며 다양한 하드웨어(지문인식, 카메라 등)를 포함하고 있고 쉬운 어플리케이션 설치로 인한 다양한 기능의 소프트웨어를 손쉽게 사용할 수 있어 현재 대부분의 사람들이 사용하고 있다. 본 발명의 일 실시예에 의한 블루투스 스마트 디바이스 지문인식 센서를 이용한 출석 체크 시스템 및 방법에서는 개개인의 스마트 디바이스에 출석을 위한 어플리케이션을 제작하여 이를 활용함으로써 비용을 절감하고 관리를 용이하게 하며 지문인식 기능을 사용하도록 하여 기존의 출석 시스템보다 보안성을 강화하였다.
- [0054] BluetoothLowEnergy(BLE) : 비콘 제작의 핵심이 되는 기술이며 해당 구역에 애드버타이징(Advertising) 신호를 지속적으로 송출한다. 따라서 마스터측에서 페어링을 하지 않고도 정보 수신이 가능하다. 라즈베리파이에서 이 기술을 이용해 비콘으로 활용하며 스마트폰과 같은 스마트 디바이스에 출석을 위한 키값(UUID)을 전송하여 부정 인증 등을 방지한다.
- [0055] 웹소켓(WebSocket) : 서버와 라즈베리파이 간의 통신에 사용된다. TCP용을 사용하고, http보다 데이터의 길이가 짧아 속도가 빠르며 http와 달리 동시에 양방향 통신이 가능하다.
- [0056] MySql : 본 발명의 일 실시예에 의한 블루투스 스마트 디바이스 지문인식 센서를 이용한 출석 체크 시스템 및 방법에서 활용한 데이터베이스이다. 실제 대학교나 기업 등에서 병합하기 쉽도록 최대한 작고 효율적이게 데이터베이스를 디자인 하였다.
- [0057] Node.js : 웹 서버를 구성하고 동작하게 하는 어플리케이션이다. 자바 스크립트 언어로 되어있어 사용법의 습득 및 사용이 쉽다. 구성이 간단하여 메모리 및 CPU를 적게 사용하고 따라서 하드웨어적 부담이 매우 적다. 스마트 디바이스 어플리케이션용 서버나 IoT 관련 디바이스인 라즈베리파이 등에서 널리 사용되는 어플리케이션이다.

[0059] **본 발명의 일 실시예에 의한 블루투스 스마트 디바이스 지문인식 센서를 이용한 출석 체크 시스템 및 방법**

- [0060] 도 1은 본 발명의 일 실시예에 의한 블루투스 스마트 디바이스 지문인식 센서를 이용한 출석 체크 시스템의 블록도를 도시한 것이다.
- [0061] 도 1을 참조하면, 본 발명의 일 실시예에 의한 블루투스 스마트 디바이스 지문인식 센서를 이용한 출석 체크 시스템은, 서버(102)에 강의실과 관련된 고유값을 전송하고 상기 서버(102)로부터 상기 고유값에 해당하는 강의 UUID(universal unique identification)를 수신하며, 수신된 강의 UUID에 기반하여 애드버타이징 신호를 송출하는 비콘(100), 출석자가 소지하는 스마트 디바이스로서, 지문인식 센서(미도시)를 포함하여 출석자의 지문인증을 수행하며, 상기 비콘(100)으로부터 수신된 강의 UUID에 기반하여 상기 서버(102)에 접속하는 제1 내지 제n 스마트 디바이스(104_1 내지 104_n), 및 상기 강의 UUID, 강의 정보, 출석 정보, 및 출석자 ID를 저장하는 데이터베이스(미도시)를 포함하며, 상기 스마트 디바이스(104_1 내지 104_n)에서 출석자의 지문 인증이 성공하는 경우 상기 스마트 디바이스(104_1 내지 104_n)로부터 수신한 출석자 ID 및 강의 UUID에 기반하여 출석 체크를 수행하는 서버(102)를 포함한다.
- [0062] 상기 스마트 디바이스(104_1 내지 104_n)는 상기 비콘(100)으로부터 수신된 강의 UUID 및 상기 강의실과 관련된 고유값을 결합하여 상기 서버(102)에 전송하고, 상기 서버(102)는 인터넷주소를 상기 스마트 디바이스(104_1 내지 104_n)에 제공하여 상기 스마트 디바이스(104_1 내지 104_n)에서 웹브라우저에 기반한 출석 체크 어플리케이션이 실행되도록 함으로써, 피지컬 웹(physical web)에 기반하여 스마트 디바이스(104_1 내지 104_n)에서 출석 체크 어플리케이션이 실행된다. 예를 들어, 제1 스마트 디바이스(104_1)가 강의 UUID와 강의실 번호를 결합한 값인 "X354"를 상기 서버(102)로 전송하면, 상기 서버(102)는 출석 체크 어플리케이션이 상기 제1 스마트 디바이스(104_1)의 웹 브라우저에서 실행될 수 있도록 상기 제1 스마트 디바이스(104_1)로 관련된 웹 주소를 전송한다.
- [0063] 또한, 상기 어플리케이션이 처음 실행되는 경우, 상기 어플리케이션의 고유한 어플리케이션 UUID가 랜덤하게 생성되어 상기 스마트 디바이스(104_1 내지 104_n)에 저장되고, 상기 어플리케이션 UUID는 상기 서버(102)로 전송되어 상기 데이터베이스에 저장되며, 상기 서버(102)는, 상기 스마트 디바이스(104_1 내지 104_n)로부터 수신된 어플리케이션 UUID 및 출석자 ID가 상기 데이터베이스에 저장된 어플리케이션 UUID 및 출석자 ID와 일치하는 경우에만, 상기 스마트 디바이스(104_1 내지 104_n)에서 상기 출석 체크 어플리케이션이 실행되도록 하고, 상기 스마트 디바이스(104_1 내지 104_n)로부터 수신된 어플리케이션 UUID 및 출석자 ID가 상기 데이터베이스에 저장

된 어플리케이션 UUID 및 출석자 ID와 일치하지 않는 경우에는 상기 출석자를 불량 사용자로 등록하고 상기 어플리케이션을 이용할 수 없게 한다.

- [0064] 또한, 상기 어플리케이션의 삭제 및 변경 시 상기 어플리케이션 UUID가 삭제되고, 상기 서버(102)는 처음 가동 시 상기 강의 UUID 값을 랜덤하게 생성하여 상기 데이터베이스에 저장한다.
- [0065] 또한, 상기 서버(102)는 상기 강의 UUID를 소정 시간마다 랜덤하게 재생성하여 상기 강의 UUID를 초기화한다.
- [0066] 상기와 같이 구성된 도 1은 본 발명의 일 실시예에 의한 블루투스 및 스마트 디바이스 지문인식 센서를 이용한 출석 체크 시스템의 동작을 하기에 상세히 설명하기로 한다.
- [0067] 출석자는 개인의 스마트폰과 같은 스마트 디바이스(104_1 내지 104_n)를 소지하고 있다.
- [0068] 먼저 도 2를 참조하여 통신 구성을 보면, 라즈베리 파이 iBeacon(100, 이하 비콘이라 함)은 어플리케이션에 블루투스(BLE)를 통한 애드버타이징 신호를 지속적으로 송출하며 어플리케이션과 따로 데이터를 주고 받지 않는다.
- [0069] 어플리케이션은 서버(102) Restful Api를 통한 http 통신을 하며, 비콘(100)과 서버(102)는 웹 소켓을 통한 통신을 한다. 다음으로 시스템에 대해 설명한다.
- [0070] 출석자가 출석 체크를 하기 위해선 어플리케이션에 미리 자신의 지문을 등록하여야 하며 1개의 지문만 등록 가능하다. 등록한 지문이 아니면 출석 인증을 받을 수 없다.
- [0071] 비콘(100)은 가동 시 서버(102)에 비콘의 고유 번호를 전송하고 서버(102)는 해당 고유번호에 맞는 시간대의 강의 UUID(universal unique identification)를 전송한다. 비콘(100)은 서버(102)로부터 전송받은 강의 UUID로 애드버타이징 신호 송출을 시작한다.
- [0072] 스마트 디바이스(104_1 내지 104_n)를 소지하고 있는 출석자들이 비콘(100) 근처에 접근하면 스마트 디바이스(104_1 내지 104_n)는 비콘(100)으로부터 자동으로 강의 UUID 정보를 받게 되며 해당 강의 UUID값으로 서버(102)에 접근할 권한을 얻게 된다. 서버(102)에 접근할 권한을 얻은 출석자들은 등록해놓은 지문으로 인증이 완료되면 서버(102)에 출석 체크 요청을 하게 된다.
- [0073] 서버(102)는 데이터베이스에 저장되어 있는 시간과 비교하여, 일정 시간 이상이면 지각 혹은 결석 체크를 자동으로 실시한다. 어플리케이션은 각각 고유한 값을 가지고 있으며 출석자가 처음으로 서버(102)에 접속할 때 출석자 ID와 어플리케이션의 고유한 값인 어플리케이션 UUID를 묶어서 저장한다.
- [0074] 서버(102)는 출석자가 접속할 때마다 이 값들을 검사하며, 한 개라도 일치하지 않을 시 해당 사용자를 불량 사용자로 등록하고 하루 동안 시스템을 이용할 수 없게 한다. 이는 시스템의 부정한 사용을 방지하기 위한 조치이다.
- [0075] 도 2는 본 발명의 일 실시예에 의한 블루투스 및 스마트 디바이스 지문인식 센서를 이용한 출석 체크 방법의 흐름도이다. 본 실시예에서는 제1 스마트 디바이스(104_1)를 소지하고 있는 제1 출석자가 강의실에 들어간다고 가정한다.
- [0076] 도 1 및 도 2를 참조하면, 단계 S200에서, 강의실에 설치된 비콘(100)이 서버(102)에 상기 강의실과 관련된 고유값을 전송한다.
- [0077] 단계 S202에서, 상기 서버(102)가 상기 비콘(100)으로 상기 고유값에 해당하는 강의 UUID를 전송한다.
- [0078] 단계 S204에서, 상기 비콘(100)이 상기 서버(102)로부터 수신된 상기 고유값에 해당하는 강의 UUID에 기반하여 애드버타이징 신호를 송출한다.
- [0079] 단계 S206에서, 제1 출석자가 상기 강의실에 들어가는 경우, 상기 제1 출석자가 소지하는 제1 스마트 디바이스(104_1)가 상기 비콘(100)으로부터 상기 강의 UUID를 수신한다.
- [0080] 단계 S208에서, 상기 제1 스마트 디바이스(104_1)에서 지문 인증을 실시한다.
- [0081] 단계 S210에서, 상기 지문 인증이 성공하는 경우, 상기 제1 스마트 디바이스(104_1)가 출석자 ID 및 상기 강의 UUID를 상기 서버(102)로 전송하여 출석 체크를 요청한다.
- [0082] 단계 S212에서, 상기 서버(102)가 상기 제1 스마트 디바이스(104_1)로부터 수신한 출석자 ID 및 강의 UUID에 기반하여 출석 체크를 수행한다.

- [0083] 한편, 상기 단계 S204 이후에, 상기 제1 스마트 디바이스(104_1)가 상기 비콘(100)으로부터 수신된 강의 UUID 및 상기 강의실과 관련된 고유값을 결합하여 상기 서버(102)에 전송하는 단계와, 상기 서버(102)가 인터넷주소를 상기 제1 스마트 디바이스(104_1)에 제공하여 상기 제1 스마트 디바이스(104_1)에서 웹브라우저에 기반한 출석 체크 어플리케이션이 실행되도록 하는 단계가 더 수행될 수 있다. 즉, 피지컬 웹(physical web)에 기반하여 제1 스마트 디바이스(104_1)에서 출석 체크 어플리케이션이 실행될 수 있다.
- [0084] 이 경우, 상기 단계 S206은 제1 출석자가 상기 강의실에 들어가는 경우, 상기 출석 체크 어플리케이션을 통해 제1 스마트 디바이스(104_1)가 상기 비콘(100)으로부터 상기 강의 UUID를 수신하는 단계를 포함할 수 있고, 상기 단계 S208은 상기 출석 체크 어플리케이션을 통해 상기 제1 스마트 디바이스(104_1)에서 지문 인증을 실시하는 단계를 포함할 수 있다.
- [0085] 또한, 상기 어플리케이션이 처음 실행되는 경우, 상기 어플리케이션의 고유한 어플리케이션 UUID가 랜덤하게 생성되어 제1 스마트 디바이스(104_1)에 저장되고, 상기 어플리케이션 UUID는 상기 서버(102)로 전송되어 상기 데이터베이스에 저장되며, 상기 서버(102)는, 상기 제1 스마트 디바이스(104_1)로부터 수신된 어플리케이션 UUID 및 출석자 ID가 상기 데이터베이스에 저장된 어플리케이션 UUID 및 출석자 ID와 일치하는 경우에만, 상기 제1 스마트 디바이스(104_1)에서 상기 출석 체크 어플리케이션이 실행되도록 하고, 상기 제1 스마트 디바이스(104_1)로부터 수신된 어플리케이션 UUID 및 출석자 ID가 상기 데이터베이스에 저장된 어플리케이션 UUID 및 출석자 ID와 일치하지 않는 경우에는 상기 출석자를 불량 사용자로 등록하고 상기 어플리케이션을 이용할 수 없게 한다. 또한, 상기 어플리케이션의 삭제 및 변경 시 상기 어플리케이션 UUID가 삭제된다.
- [0086] 또한, 상기 서버(102)는 처음 가동 시 상기 강의 UUID 값을 랜덤하게 생성하여 상기 데이터베이스에 저장하고, 상기 서버(102)는 상기 강의 UUID를 소정 시간마다 랜덤하게 재생성하여 상기 강의 UUID를 초기화함으로써, 불량 사용자가 강의 UUID를 탈취하여 비콘(100)에 접근하지 않고 출석 인증을 시도하는 것을 방지할 수 있다.
- [0088] **본 발명의 일 실시예에 의한 블루투스 및 스마트 디바이스 지문인식 센서를 이용한 출석 체크 시스템 및 방법의 흐름 구성**
- [0089] 본 발명의 일 실시예에 의한 블루투스 및 스마트 디바이스 지문인식 센서를 이용한 출석 체크 시스템 및 방법을 흐름순대로 나열하고 동작 과정을 설명한다.
- [0090] 1) 서버 가동 - 서버(102)를 처음 가동 시 데이터베이스 내의 강의 UUID값을 전부 랜덤하게 생성한 후 저장한다. 서버(102) 가동 후 일정시간, 예를 들어 1시간이 경과할 때마다 강의 UUID를 랜덤하게 재생성한다.
- [0091] 2) 비콘 가동 - 비콘(100) 가동시 서버(102)에 강의실 번호 등과 같은 고유값을 전송한다. 그 후 서버(102)는 해당 고유값에 해당하는 강의 UUID를 라즈베리파이로 전송하고 라즈베리파이는 해당 UUID값을 사용하여 비콘(100)으로 동작한다.
- [0092] 3) 어플리케이션 설정 - 출석자의 스마트폰과 같은 스마트 디바이스(104_1 내지 104_n)에서 지문 인식과 BLE가 사용 가능한지 여부를 체크한다. 두 기능 중 하나라도 사용 불가능하면 비콘(100)에 연결된 별도의 아두이노 지문인식 센서(미도시)를 통해 출석을 하도록 한다. 두 기능이 모두 사용 가능하면 어플리케이션에 등록된 지문이 있는지 검사한다. 지문이 등록되어 있지 않으면 지문을 등록하도록 유도한다.
- [0093] 4) 불량 사용자 검증 - 어플리케이션은 실행 후 처음 사용자 인증 시 어플리케이션의 고유한 UUID값과 함께 사용자 ID를 서버(102)에 전송하고, 서버(102)는 이를 데이터베이스에 저장한다. 그 후 어플리케이션 실행 시마다 어플리케이션의 UUID값과 사용자 ID를 통해 서버(102)에 인증을 요구하고 인증에 성공해야만 출석자는 어플리케이션을 이용할 수 있다.
- [0094] 어플리케이션 UUID값은 어플리케이션을 삭제하거나 변경시 같이 삭제되며, 그 후 어플리케이션을 실행하면 서버(102)에서 해당 사용자를 불량 사용자로 등록하고 당일엔 어플리케이션을 이용할 수 없다. 따라서 별도의 아두이노 지문인식 등의 과정을 거쳐야 한다. 다음날이 되면 불량 사용자는 모두 리셋되며 어플리케이션을 다시 등록하고 사용할 수 있다.
- [0095] 5) 어플리케이션 사용 - 어플리케이션이 비콘(100)의 범위 내에 접근하면 비콘(100)의 UUID 값을 전송받게 된다. 어플리케이션은 서버(102)에 해당 UUID 값이 데이터베이스에 존재하는지를 검사받고 존재하면 UUID 값에 대응되는 장소(강의실 등)의 이름을 수신한다. 스마트 디바이스(104_1 내지 104_n)에서 비콘 재탐색은 언제나

가능하다.

- [0096] 6) 출석 인증 - 스마트 디바이스(104_1 내지 104_n)가 서버(102)로부터 UUID값을 검증받고 대응되는 장소가 출석자가 출석을 위한 공간이 맞으면 출석체크를 위한 지문 인증을 시도할 수 있다. 어플리케이션에 등록되어있는 지문과 일치하는 경우 서버(102)에 출석 체크 요청을 하게 되고, 서버(102)는 출석 시간에 맞춰 출석, 지각, 결석 등을 판단하고 판단 결과를 자동으로 데이터베이스에 저장하고, 필요한 경우 해당 스마트 디바이스에 판단 결과를 제공한다.
- [0097] 7) 과정 반복 - 서버(102)가 멈추기 전까지 해당 과정은 계속 되풀이되며, 매일 지정된 시간까지 출석자가 출석 인증을 하지 않으면 자동으로 모두 결석처리가 된다.
- [0098] 도 3은 비콘(100)의 동작을 설명하기 위한 흐름도이다. 도 3을 참조하면, 단계 S300에서 비콘이 온(On)된다. 단계 S302에서, 비콘(100)이 서버에 연결되고 오류 여부가 확인된다. 오류가 있는 경우, 단계 S304에서 오류가 점검되고, 오류가 없는 경우, 단계 S306에서 강의실 번호와 같은 비콘(100)의 고유값을 서버(102)에 전송하고, 서버(102)로부터 고유값에 해당하는 강의 UUID를 수신하며, 이에 기반하여 비콘(100)이 가동된다.
- [0099] 도 4는 스마트 디바이스(104_1 내지 104_n)에서 실행되는 어플리케이션의 동작과 서버(102)의 동작을 설명하기 위한 도면이다.
- [0100] 단계 S400에서 스마트 디바이스(104_1 내지 104_n)의 어플리케이션이 온(On)된다. 단계 S402에서 지문이 등록되어 있는지가 판단되고, 지문이 등록되어 있지 않은 경우, 단계 S404에서 출석자가 어플리케이션을 통해 지문을 등록한다.
- [0101] 단계 S406에서는 어플리케이션 ID에 기반하여 사용자가 불량 사용자인지가 판단된다. 불량 사용자인 경우 어플리케이션이 종료되고, 불량 사용자가 아닌 경우, 단계 S408에서 비콘 검색을 시작한다.
- [0102] 단계 S410에서 비콘을 인식하였는지가 판단되고, 비콘을 인식한 경우, 단계 S412에서 지문 인증을 실시한다. 단계 S414에서는 지문 인증이 성공하였는지가 판단되고, 지문 인증이 실패한 경우 어플리케이션을 종료하고, 지문 인증이 성공한 경우, UUID와 출석자 ID를 서버(102)로 전송한다.
- [0103] 단계 S416에서는 비콘 재검색 요청이 있는 경우, 단계 S408로 진행하고, 재검색 요청이 없는 경우 어플리케이션을 종료한다.
- [0104] 한편, 단계 S418에서 서버(102)가 온(On)된다. 단계 S420에서 서버(102)를 처음 가동하는지가 판단된다. 서버(102)를 처음 가동하는 경우 단계 S424로 진행하여 강의 UUID를 랜덤하게 생성하여 서버(102)의 데이터베이스에 저장된 강의 UUID를 초기화한다.
- [0105] 서버(102)를 처음 가동하는 것이 아닌 경우, 단계 S422에서 강의 UUID를 초기화한 후 1시간 이상 경과되었는지가 판단된다.
- [0106] 강의 UUID를 초기화한 후 1시간 이상 경과된 경우, 단계 S424에서 강의 UUID를 랜덤하게 재생성하여 서버(102)의 데이터베이스에 저장된 강의 UUID를 초기화한다.
- [0107] 단계 S426에서는 스마트 디바이스(104_1 내지 104_n)의 어플리케이션으로부터 강의 UUID가 수신되는지가 판단된다.
- [0108] 스마트 디바이스(104_1 내지 104_n)의 어플리케이션으로부터 강의 UUID가 수신되는 경우, 단계 S428에서 수신한 강의 UUID를 검증하고 검증 성공 여부가 판단된다.
- [0109] 강의 UUID 검증이 성공한 경우, 단계 S430에서 서버(102)는 강의 UUID와 출석자의 ID에 기반하여 출석, 지각, 결석 여부를 판단한 후, 판단 결과를 데이터베이스에 출석 정보로서 저장하고, 필요한 경우 판단 결과를 해당 스마트 디바이스에 제공한다.

[0111] 본 발명의 일 실시예에 의한 블루투스 스마트 디바이스 지문인식 센서를 이용한 출석 체크 시스템 및 방법의 동작

[0112] 어플리케이션

[0113] 어플리케이션 제작에는 윈도우10 환경에서 안드로이드 스튜디오(Android Studio)를 사용하였으며 자바(java) 언어

어와 그레이들(Gradle) 툴을 사용하였다. 지문인식 기능을 활용하기 위해 삼성 SDK의 SPASS 라이브러리를 사용하였고, BLE 기능을 활용하기 위해 AltBeacon 라이브러리를 사용하였으며 서버 연결을 위한 Restful api로 loopj의 android-async-http 라이브러리를 활용하였다.

[0114] 어플리케이션의 주요 기능으로 비콘 탐색, 지문 인식, 서버 연결이 있으며 하기에 이를 설명한다.

[0115] 1) 비콘 탐색

[0116] 도 5는 어플리케이션 가동시 나타나는 첫 화면이다. 비콘(100)을 탐색하기 전이기 때문에 출석 체크는 사용할 수 없으며, 강의실 찾기 외에 기타 기능을 사용 할 수 있다.

[0117] 도 6은 비콘(100)을 탐색 완료한 상태이다. 이 상태에서 출석 인증을 시도할 수 있으며 출석 체크 버튼을 누르면 지문 인증 상태를 알려주는 다이얼로그 창이 하나 생성된다.

[0118] 도 7은 비콘 탐색을 위한 코드 중 일부이다. doscan 메소드에서 비콘 탐색을 시작하며, logGenericBeacon 메소드에서 비콘 정보를 추출하고 후에 서버(102)에 이 정보를 전송하고 강의실 이름을 받아온다.

[0119] 2) 지문 인식

[0120] 도 8은 지문 인식 다이얼로그 기본 화면이며 도 9는 현재 지문 인식이 진행되는 상황에 따라 자동으로 다이얼로그의 메시지가 바뀐다는 것을 보여주는 예이다. 지문 인식과 관련하여서는 앞에 서술한 대로 삼성 SDK의 api를 사용하였다. 안드로이드 자체에 지문 인식을 위한 라이브러리가 존재하지만, 지문을 구별하지는 않고 등록된 지문들 중 1개라도 인식이 되었는지 여부만 알 수 있다. 따라서 본 시스템에 이를 사용할 경우, 한 사용자가 여러 사용자들의 디바이스에 지문을 인식해 놓고 출석하는 등으로 악용할 가능성이 있기 때문에 삼성 SDK의 api를 사용하였다. 삼성 S-PASS는 지문을 인식하면 해당 지문이 몇 번인지를 리턴할 수 있기 때문에 안드로이드 라이브러리의 문제점을 해결할 수 있다.

[0121] 도 10은 지문 인식 관련 코드이다. startIdentify는 지문 인증을 시작하는 메소드이다. onDoingIdentify은 현재 인증이 진행중임을, onReadyIdentify은 인증을 할 준비가 되었음을 각각 표시하는 변수이다. mSpassFingerprint은 지문인식을 준비, 진행, 결론 도출 등을 총괄하는 클래스의 객체이다. mIdentifyListener은 mSpassFingerprint에서 지문 인식을 통해 결과값을 받고 작업을 수행 할 수 있는 메소드이다. mIdentifyListener를 보면 인증에 성공하였을 경우 setting이 false이면 인증에 사용할 지문의 id번호를 어플리케이션에 저장하며 setting이 true일 경우 서버에 출석을 요청하여 출석, 지각, 결석 등의 결과를 서버(102)에 서 받아 화면에 출력한다. setting은 현재 어플리케이션에 인증에 사용할 지문 ID가 존재하는지 여부를 체크하는 변수이다. 지문 ID는 지문을 등록할 때마다 0에서부터 1씩 증가하며 자동으로 부여되는 방식이므로, 사용자의 지문 정보 자체가 노출될 염려는 없다.

[0122] 3) 서버 연결

[0123] 어플리케이션과 서버(102)는 http 방식을 사용하며 이를 위해 Restful api가 필요하다. 이를 직접 구현할 수도 있으나, 개발의 편의를 위해 android-async-http 라이브러리를 사용하였다. 도 11은 해당 코드의 설정과 사용법을 보여주고 있다. RestClient 클래스에서 서버(102)의 기본 URL을 입력하고 get, post 방식의 메소드를 정의함으로써 이를 사용할 수 있게 한다. 사용 시에는 RestClient 클래스로 객체를 생성하여 get, post 메소드를 호출하여 사용한다.

[0124] 위에서 설명한 기능 외에 불량 사용자 인증을 위한 어플리케이션 고유값 생성 기능이 있으며 이는 아래의 서버(102) 기능에서 자세히 설명하도록 한다.

[0126] 서버

[0127] 서버에는 Node.js를 통해 구현한 메인 서버, 데이터베이스인 Mysql이 사용되고 있다. Node.js에서는 크게 사용자와 http 통신의 get, post 요청을 받은 후 데이터베이스에서 검색하여 값을 전송하거나 데이터베이스를 수정하는 역할, 비콘(100)과 소켓(Socket) 통신을 통해 비콘(100)의 고유값을 수신하고 이에 맞는 강의 정보를 데이터베이스에서 검색하여 송신하는 역할 2가지가 있다.

[0128] 도 12에서 app은 http 통신을 실행하는 변수이며, .get과 .post 등의 함수를 통해 동작한다. dbmodule은 데이터베이스와 통신하고 작업을 수행하는 함수를 모아놓은 모듈이다. 해당 모듈에서 상황에 맞는 함수를 호출하고,

결과 값을 send 함수를 통해 요청자에게 송신한다.

[0129] 메인 서버(102)는 소켓(Socket) 통신의 마스터(서버) 측을 담당하고 있다. 소켓(Socket)의 서버는 클라이언트와 통신할 때 클라이언트의 ip를 알 필요가 없으며 클라이언트에서 서버의 ip만 알고 있으면 서로 통신이 가능하다. 소켓(Socket) 통신은 접속(connection) 값을 가지고 있는 on 함수 안에서 동작한다. on 함수 안의 명령 값이 메시지(message)이면 클라이언트로부터 값을 수신할 때, 접속해제(disconnection)는 클라이언트와 연결이 끊어졌을 때 동작을 지정한다. 클라이언트로 값을 전송할 때는 send 함수를 사용한다.

[0130] 도 14는 Mysql의 테이블 설계도이다. classroom, class, student 테이블이 상위 테이블이며 하위 테이블들은 상위 테이블의 기본키를 외래키로 받아 기본키로 사용한다. attendancelist 테이블은 학생이 어떠한 강의를 듣는지의 정보를, attendance 테이블은 모든 학생의 출석 정보를 저장한다. classlist 테이블은 강의가 시작하는 요일, 시간, 강의실 정보를 저장하고 있다. stukeys는 어플리케이션의 고유 UUID정보와 학생 ID를 묶어서 저장하고 있다.

[0131] 학생이 어플리케이션을 처음 가동하면 자동으로 어플리케이션 고유 UUID키가 생성되고, 서버(102)에 접속 시 이 정보를 저장한다. 그 후 어플리케이션 가동 시마다 서버(102)에서 UUID 값과 학생 ID를 검증받으며 하나라도 일치하지 않으면 해당 학생을 불량 사용자로 등록하고 시스템을 하루 동안 이용할 수 없다. 이 기능을 통해 시스템의 악용을 어느 정도 방지할 수 있다.

[0132] 서버(102)에는 위에 설명한 기능 외에 강의의 고유 UUID 값을 일정 시간마다 랜덤하게 리셋하는 기능이 있다. 비콘(100)과 통신 시 비콘(100)에 해당 시간대에 맞는 강의 UUID를 전송하는데, 일정시간마다 강의 UUID 값을 리셋하고 비콘(100)에 자동으로 적용시켜 해킹을 통해 강의 UUID 값이 갈취되더라도 계속 해당 값이 바뀌기 때문에, 서비스의 악용 가능성을 현저히 낮출 수 있다.

[0134] 비콘

[0135] 도 15는 비콘(100)의 동작 코드를 도시한 도면이다. 비콘(100)은 비교적 간단하다. 비콘(100) 또한 Node.js로 구현되어 있다. 여기서 비콘(100)은 Node.js를 사용하더라도 서버로서 동작하지 않으며 메인 서버(102)와의 통신을 소켓(Socket)으로 구현하기 용이하도록 하였다.

[0136] roomid는 비콘(100)의 고유 값이며 여기서 'c101'은 강의실 이름이다. 전체적으로 서버(102)의 소켓(Socket) 구현 코드와 유사하다. 비콘(100)이 가동되면 서버(102)로 먼저 연결을 시도하며 서버(102)로부터 수신한 값을 통해 비콘(100)의 동작 여부가 결정된다.

[0137] Blecon.startAdvertising은 비콘으로 동작하여 애드버타이징(Advertising)을 시작하게 하는 함수이다. Blecon.startAdvertising가 실행되고 나면 서버(102)와의 연결이 해제될 때까지 계속 비콘으로 동작하며 어플리케이션과는 상호 통신하지 않는다. 애드버타이징이 단순히 주위에 일정한 신호를 송출하는 역할이기 때문에 스마트폰 디바이스(104_1 내지 104_n)의 어플리케이션은 비콘(100) 근처로 접근만 하여도 애드버타이징 신호를 받을 수 있다. 또한 위에서 서술한 대로 일정 시간마다 강의의 UUID값이 바뀌는데, 이때마다 자동으로 바뀐 강의 UUID값으로 비콘(100)이 동작하게 된다.

[0139] 보안

[0140] 상기에서 어플리케이션, 서버(102), 비콘(100)의 기능을 설명하며 언급했던 보안적인 측면을 다시 한 번 정리하도록 한다.

[0141] 1) 어플리케이션의 고유 UUID - 어플리케이션이 처음 실행될 때 어플리케이션만의 고유 UUID를 랜덤하게 생성하고 어플리케이션에 저장한다. UUID는 랜덤 생성 시 중복 확률이 0%에 수렴하므로 고유값으로 사용하기에 적합하다. 어플리케이션 처음 실행 시 사용자의 ID값을 입력받고 다시는 수정할 수 없게 구성되어 있으며 입력받은 ID 값과 어플리케이션 UUID 값을 서버(102)에 전송하고 데이터베이스에 등록한다.

[0142] 만약 사용자가 시스템을 악용하기 위해 사용자 ID값을 변경하는 작업 또는 어플리케이션 데이터 삭제 등을 시도할 시 어플리케이션에 저장되어 있는 어플리케이션 UUID 값도 같이 삭제된다. 따라서 삭제된 어플리케이션 UUID와 같이 저장되어 있는 사용자 ID는 어떠한 방법으로도 서버(102)에서 인증을 받을 수 없게 되며 해당 사용자는 서버(102)에 접속 시 바로 불량 사용자로 등록되어 시스템을 하루동안 이용할 수 없게 된다.

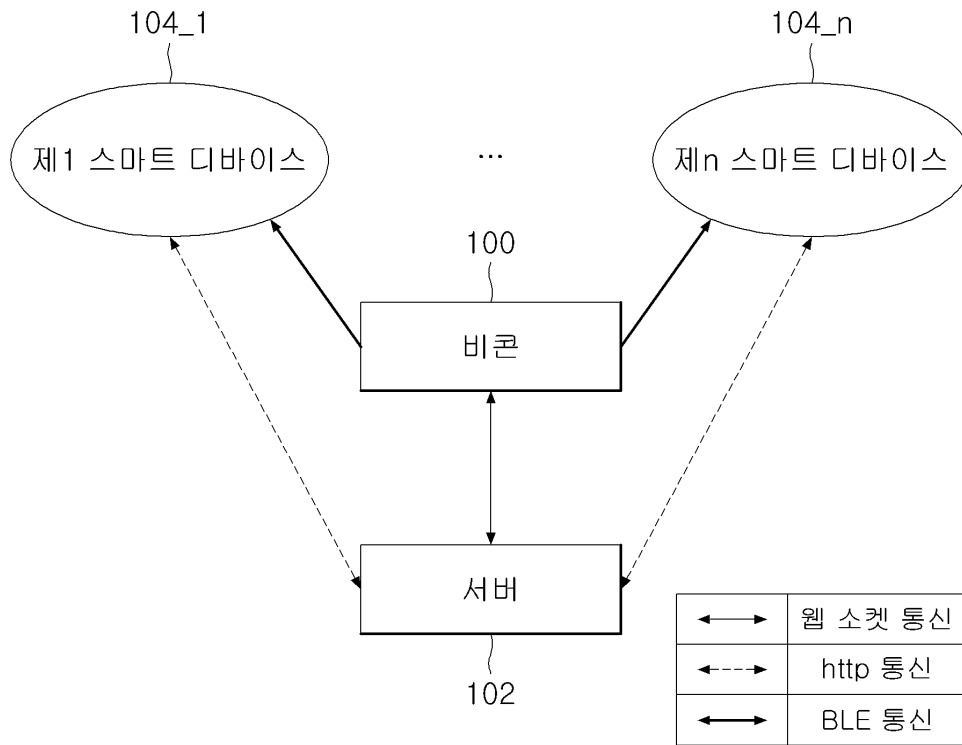
- [0143] 2) 강의의 고유 UUID - 어플리케이션과 서버(102) 간 UUID값이 포함된 통신 시 post 방식을 사용하지만 해당 UUID 값이 노출될 수 있다. 또한 비콘(100)과 서버(102) 간 소켓(Socket) 통신 시에도 UUID 값이 노출될 수 있다. 각각의 통신에 TLS 방식을 적용하여 보안성을 강화할 수 있지만, 본 실시예에서는 UUID 값을 일정시간마다 리셋하는 방식으로 간단하게 보안성을 높인다.
- [0144] 강의마다 고유한 UUID값이 존재하며 이 정보는 서버(102)의 데이터베이스에 저장되어있다. 만약 UUID 값이 계속 변함이 없다면, 불량 사용자가 시스템을 악용하기 위해 UUID값 탈취에 성공했을 경우 비콘(100)에 접근하지 않고도 출석 인증을 시도할 수 있다.
- [0145] 이를 방지하기 위해 강의의 UUID 값을 일정시간마다 랜덤하게 리셋한다. 위에서 언급한 대로 UUID는 랜덤 생성 시 중복 확률이 0%에 수렴하기 때문에 랜덤하게 리셋할 시 같은 UUID의 값이 저장될 가능성은 없다고 볼 수 있다. 따라서 만약 UUID 값을 갈취하였다 하더라도 강의의 UUID 값은 계속 변하기 때문에 이를 악용하여 사용할 수 없게 된다.
- [0147] **결론**
- [0148] 스마트 디바이스를 사용한 출석 체크 시스템에는 QR코드를 활용한 방식, 비콘을 사용한 방식 등 여러 가지가 존재한다. 그러나 스마트 디바이스의 지문 인식을 사용한 시스템은 없으며 본 발명의 실시예에서는 비콘까지 사용함으로써 보안성 및 안정성을 높였다.
- [0149] 지문 인식은 생체 인식 기술로서 보안성과 사용자 검증에서 타 기술보다 뛰어난 이미 보편화된 사실이다. 또한 스마트 디바이스와 같은 개인 디바이스를 활용함으로써 시스템 도용에 필요한 비용을 줄일 수 있으며 사용자들의 편의성을 높일 수 있다. 해당 시스템의 도입 시 출결 처리를 자동화 시킬 수 있어 전산 처리의 편의성 또한 증대시킬 수 있다. 데이터베이스 테이블도 구조가 간단하여 이미 적용되어있는 출석 체크 시스템에도 쉽게 추가하여 사용할 수 있다.
- [0150] 본 발명의 일 실시예에 의한 블루투스나 스마트 디바이스 지문인식 센서를 이용한 출석 체크 시스템은 하나의 예시일 뿐이며 이를 활용하면 다양한 분야에서 응용될 수 있다. 예를 들면 도어락과 같이 높은 보안성이 요구되는 기존의 IoT 시스템에 지문 인식을 통한 동작을 추가하여 보안성을 강화할 수 있다. IoT가 아니더라도 원격에서의 보안성이 높은 사용자의 인증이 필요한 분야에서 다양한 사용을 기대할 수 있다.
- [0151] 이상 본 발명을 구체적인 실시예를 통하여 상세하게 설명하였으나, 이는 본 발명을 구체적으로 설명하기 위한 것으로, 본 발명은 이에 한정되지 않으며, 본 발명의 기술적 사상 내에서 당 분야의 통상의 지식을 가진 자에 의해 그 변형이나 개량이 가능함은 명백하다고 할 것이다.
- [0152] 본 발명의 단순한 변형 내지 변경은 모두 본 발명의 영역에 속하는 것으로, 본 발명의 구체적인 보호 범위는 첨부된 청구범위에 의하여 명확해질 것이다.

부호의 설명

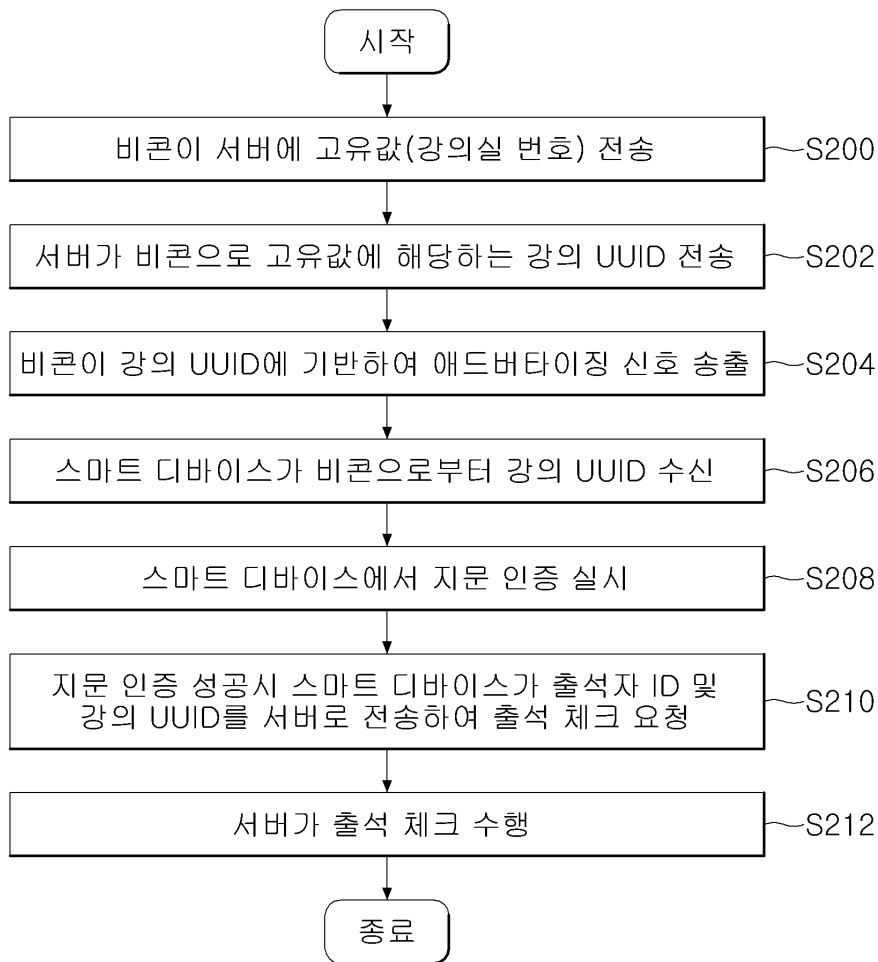
- [0153] 100 : 비콘 102 : 서버
104_1 내지 104_n : 제1 내지 제n 스마트 디바이스

도면

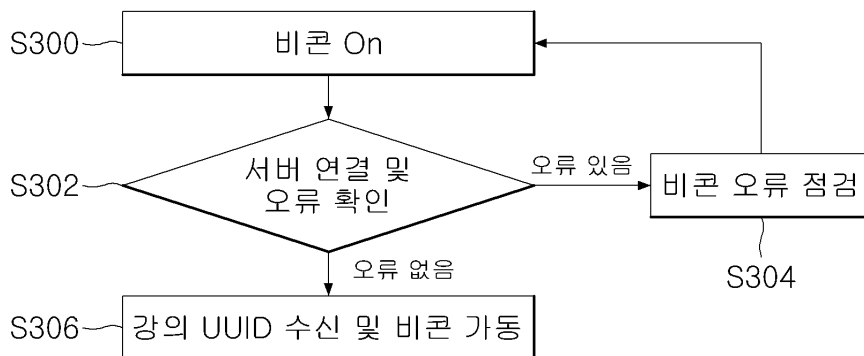
도면1



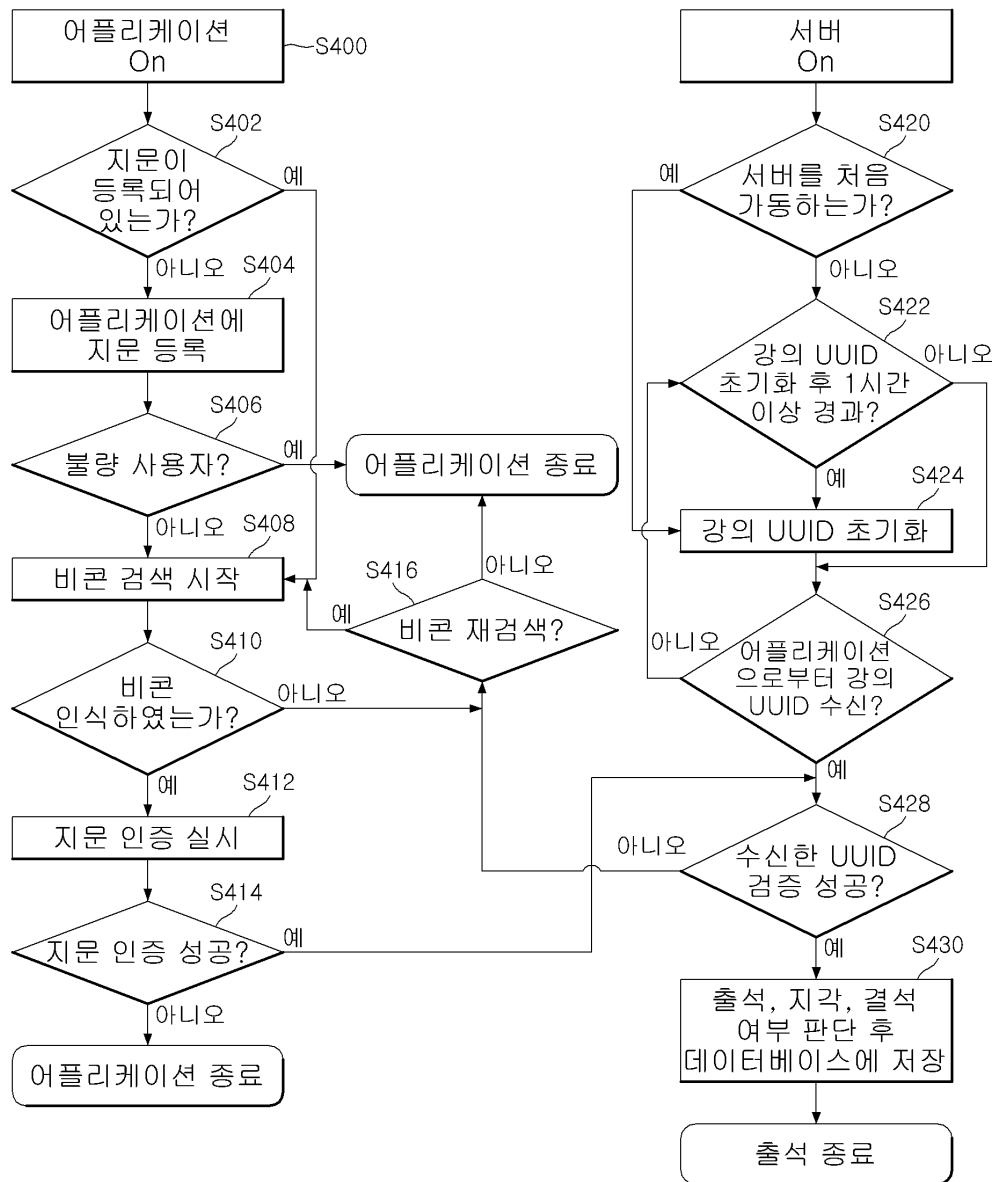
도면2



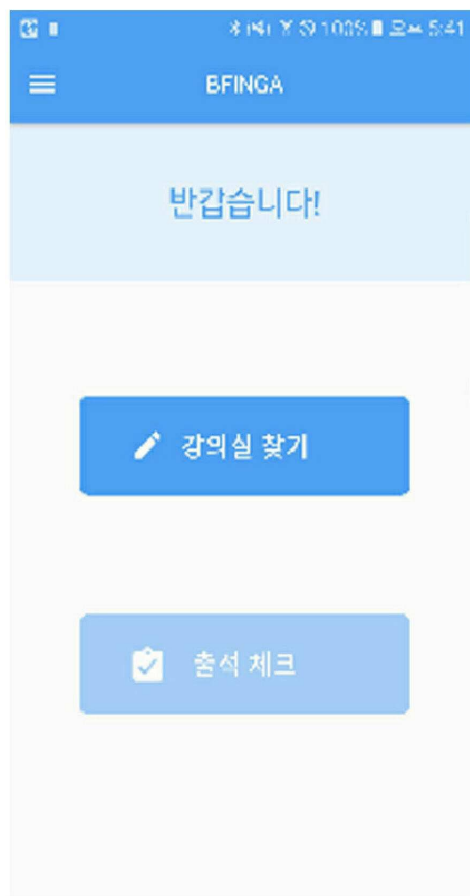
도면3



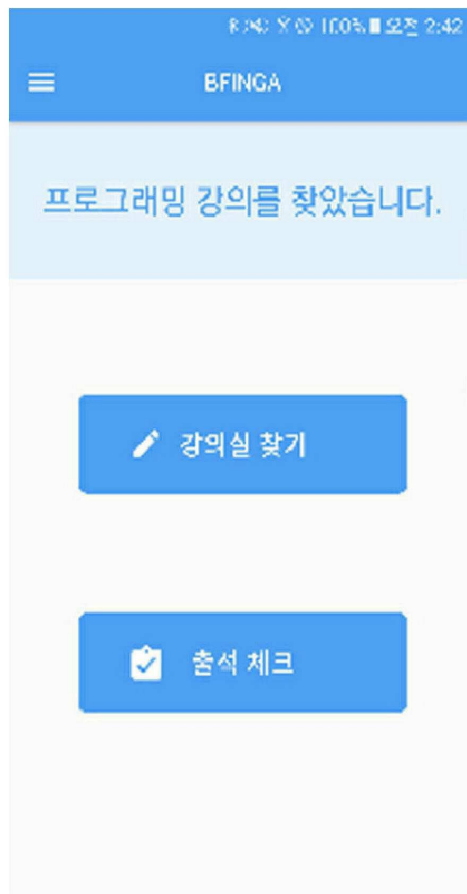
도면4



도면5



도면6



도면7

```

private void doscan(){
    isbeaconscan = true;
    AppStatus.getbm().setRangeNotifier((beacons, region) -> {
        if (beacons.size() > 0) { // 현재 한번만 탐지하고 중지
            Iterator<Beacon> beaconIterator = beacons.iterator();
            stopscan();
            Beacon beacon = beaconIterator.next();
            logGenericBeacon(beacon); // 탐지한 비콘 정보 분류
            while (beaconIterator.hasNext()) { // 계속 비콘 탐지하는 코드

                // Debug - logging a beacon - checking background logging is working.
                System.out.println("Logging another beacon.");
            }
        }
    });
    try {
        AppStatus.getbm().startRangingBeaconsInRegion(region);
    } catch (RemoteException e){
    }
}

private void logGenericBeacon(Beacon beacon) {
    StringBuilder scanString = new StringBuilder();

    scanString.append(" UUID: ").append(beacon.getId1());

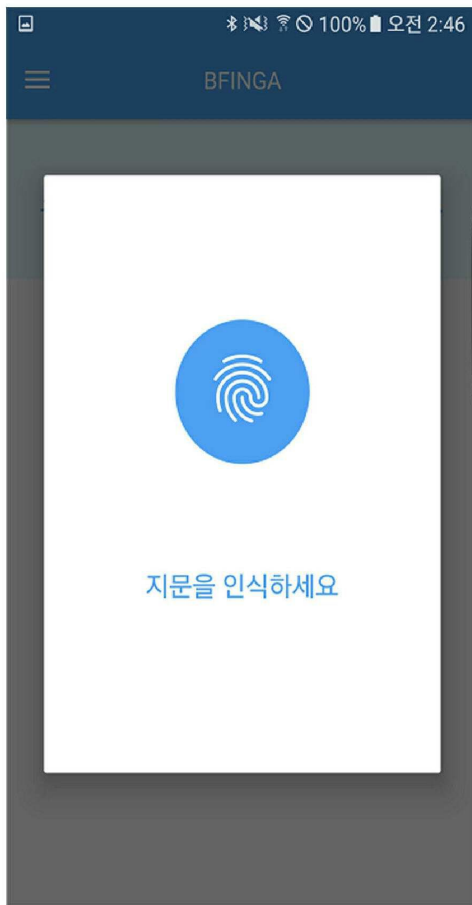
    scanString.append(" Maj. Mnr.: ");
    if (beacon.getId2() != null) {
        scanString.append(beacon.getId2());
    }
    scanString.append("-");
    if (beacon.getId3() != null) {
        scanString.append(beacon.getId3());
    }
    scanString.append(" RSSI: ").append(beacon.getRssi());

    beaconText = scanString.toString();
    uuid = beacon.getId1().toString();

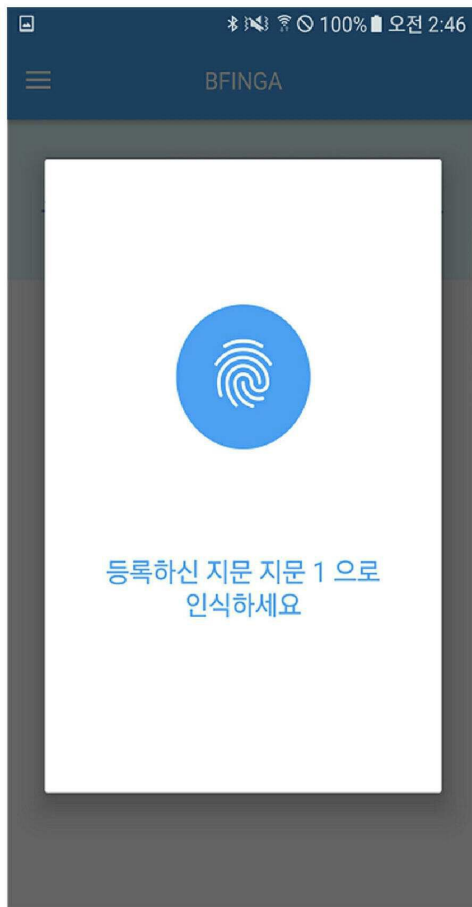
    if(!scanString.toString().equals("")){
        isbeaconscan = false;
        isfindclass = true;
        AppStatus.getattnbtn().setClickable(true);
        AppStatus.getattnbtn().setAlpha((float)1);
        dialog.cancel();
    }
}
}

```

도면8



도면9



도면10

```

private void startIdentify() {
    onDoingIdentify = true;
    if (onReadyIdentify == false) {
        try {
            onReadyIdentify = true;
            if (mSpssFingerprint != null) {
                setIdentifyIndex();
                mSpssFingerprint.startIdentify(mIdentifyListener);
            }
            if (designatedFingers != null) {
                Log.i("I", "Please identify finger to verify you with " + designatedFingers.toString() + " finger");
                if (setting) {
                    setDialogTitle("등록하신 지문 " + fingername + " 으로 인식하세요");
                }
            } else {
                Log.i("I", "Please identify finger to verify you");
            }
        } catch (SpssInvalidStateException ise) {
            onReadyIdentify = false;
            resetIdentifyIndex();
            if (ise.getType() == SpssInvalidStateException.STATUS_OPERATION_DENIED) {
                Log.i("I", "Exception: " + ise.getMessage());
                setDialogTitle("오류가 발생하였습니다. 잠시 후 다시 시도해주세요.");
            }
        } catch (IllegalStateException e) {
            onReadyIdentify = false;
            resetIdentifyIndex();
            Log.i("I", "Exception: " + e);
        }
    } else {
        Log.i("I", "The previous request is remained. Please finished or cancel first");
    }
}

private SpssFingerprint.IdentifyListener mIdentifyListener = new SpssFingerprint.IdentifyListener() {
    @Override
    public void onFinished(int eventStatus) {
        Log.i("I", "Identify finished : reason = " + getEventStatusName(eventStatus));
        int FingerprintIndex = 0;
        String FingerprintGuideText = null;
        try {
            FingerprintIndex = mSpssFingerprint.getIdentifiedFingerprintIndex();
        } catch (IllegalStateException ise) {
            Log.e("E", ise.getMessage());
        }
        if (eventStatus == SpssFingerprint.STATUS_AUTHENTICATION_SUCCESS) {
            Log.i("I", "onFinished() : Identify authentication Success with FingerprintIndex : " + FingerprintIndex);
            onDoingIdentify = false;
            if (!setting) {
                editor.putInt("mainfinger", FingerprintIndex);
                getFingerprintName(FingerprintIndex);
                editor.putBoolean("setting", true);
                editor.commit();
                AlertDialog.Builder builder = new AlertDialog.Builder(mContext);
                builder.setMessage("등록하신 지문은 " + fingername + " 입니다. 꼭 기억하세요 수정할 수 없습니다.");
                builder.setPositiveButton("OK", new DialogInterface.OnClickListener() {
                    @Override
                    public void onClick(DialogInterface dialog, int which) {
                    }
                }).setOnDismissListener((dialogInterface) -> {
                    userDialog.dismiss();
                });
                AlertDialog alertDialog = builder.create();
                alertDialog.setTitle("");
                alertDialog.show();
            }
        } else {
    }
}

```

도면11

```

public class RestClient {
    private static Context context;
    private static final String BASE_URL = "http://192.168.0.109:8107/";
    private static AsyncHttpClient client = new AsyncHttpClient();

    public RestClient(Context context){
        if(this.context == null) {
            this.context = context;
        }
    }

    public static void get(String url, RequestParams params, AsyncHttpResponseHandler responseHandler) {
        client.get(getAbsoluteUri(url), params, responseHandler);
        PersistentCookieStore cookieStore = new PersistentCookieStore(context);
        client.setCookieStore(cookieStore);
    }

    public static void post(String url, RequestParams params, AsyncHttpResponseHandler responseHandler) {
        client.post(getAbsoluteUri(url), params, responseHandler);
    }

    private static String getAbsoluteUri(String relativeUri) { return BASE_URL + relativeUri; }

    public CookieStore getCookie(){
        HttpContext httpContext = client.getHttpContext();
        CookieStore cookieStore = (CookieStore) httpContext.getAttribute(ClientContext.COOKIE_STORE);
        return cookieStore;
    }
}

RequestParams params = new RequestParams();
params.put("AUDIO", preferences.getString("AUDIO", ""));
params.put("studentid", userid);
restClient.post("checkclientid", params, new TextHttpResponseHandler() {
    @Override
    public void onFailure(int statusCode, Header[] headers, String responseString, Throwable throwable) {

    }

    @Override
    public void onSuccess(int statusCode, Header[] headers, String responseString) {

```

도면12

```

app.get('/getuserclass/:userid',function(req,res,next) { //학 생 출 석 현 황
  var userid = req.params.userid;
  dbmodule.getuserclasslist(userid,function(msg) {
    if(msg == '[]'){
      log('user ' + userid + ' is no one class list');
      res.set('Content-Type', 'application/json; charset=utf-8');
      res.send(msg);
      res.end();
    }
    else {
      res.set('Content-Type', 'application/json; charset=utf-8');
      res.send(msg);
      log('to user ' + userid + ' send class list');
      res.end();
    }
  });
});
app.post('/attendance', function(req,res,next){ // 출 석 제 크
  var useruuid = req.body.uuid;
  var userinfo = req.body.studentid;
  var result;
  log('student ' + userinfo + ' connection to ' + useruuid,id);
  res.set('Content-Type', 'text/plain');
  checkbaduser(userinfo,function(msg) {
    if(msg == 'bad'){
      res.send('bad');
      log('bad user');
      res.end();
    }
    else {
      dbmodule.attendance(useruuid,userinfo,function(msg){
        result = msg;
        log(result);
        res.send(result);
        res.end();
      });
    }
  });
});
});

```

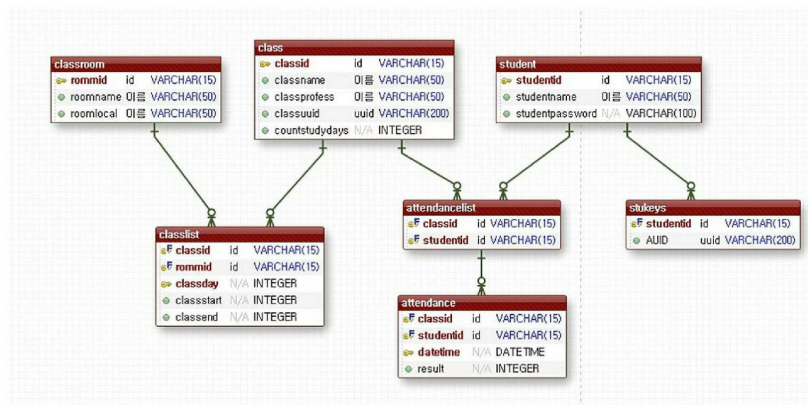
도면13

```

io.on('connection',function(socket){ //비 콘 과 소 켓 톨 션
  socket.on('message',function(message){
    roomid = message;
    setTimeout(function(){
      if(dointerval == false){
        socketback();
        interval = setInterval(function(){socketback()},1000);
        dointerval == true;
      }
    },100);
  });
  socket.on('disconnect', function() {
    clearTimeout(interval);
    if(roomid != null) {
      log('socketid ' + socket.id + ' classroom ' + roomid + ' disconnecting');
    }
    else {
      log('socketid ' + socket.id + ' disconnect');
    }
  });
});

```

도면14



도면15

```

var io = require('socket.io-client');
var socket = io('http://192.168.0.109:8107');
var Bleacon = require('bleacon');
var uuid = require('node-uuid');
var timelog = require('./timelog');
var String;
var classuuid;
var classid;
var roomid = 'c101';

log('classroom ' + roomid + ' online');

socket.on('connect',function() {
    log('main server connected');
    socket.send(roomid);
});
socket.on('message',function(data) {
    if(data == 'incorrect'){
        Bleacon.stopAdvertising();
        log('incorrect class err');
    }
    else if(data == 'nottoday'){
        Bleacon.stopAdvertising();
        log("this room don't have class today");
    }
    else if(data == 'endtoday'){
        Bleacon.stopAdvertising();
        log('end today schedule');
    }
    else if (data != undefined){
        classid = data.classid;
        log('class ' + classid + ' UUID : ' + data.uuid);
        //var bytes = uuid.parse(data);
        //String = uuid.unparse(bytes);
        Bleacon.startAdvertising(data.uuid,0,0,-59);
        log('iBeacon Start');
    }
});

socket.on('disconnect',function() {
    Bleacon.stopAdvertising();
    log('main server disconnect and ibeacon stop');
});

function log(data) {
    timelog.mlog(data);
}
    
```