

## 종합설계 프로젝트 수행 보고서

프로젝트명	다중생체인식을 활용한 금고
팀번호	S5-11
문서제목	수행계획서( ) 2차발표 중간보고서( ) 3차발표 중간보고서( ) 최종결과보고서( O )

2020.11.20

팀원 : 임 현성 (팀장)  
김 경석  
김 현정

지도교수 : 최 종필 교수 (인)  
지도교수 : 정 의훈 교수 (인)

## 문서 수정 내역

작성일	대표작성자	버전(Revision)	수정내용	
2020.01.20	임현성(팀장)	1.0	수행계획서	최초작성
2020.02.27	임현성(팀장)	2.0	2차발표자료	설계서추가
2020.04.20	임현성(팀장)	3.0	3차발표자료	시험결과추가
2020.06.27	임현성(팀장)	4.0	최종발표자료	시험결과 수정
2020.11.20.	임현성(팀장)	5.0	최종결과보고서	최종작성

## 문서 구성

진행단계	프로젝트 계획서 발표	중간발표1 (2월)	중간발표2 (4월)	학기말발표 (6월)	최종발표 (10월)
기본양식	계획서 양식	계획서 양식	계획서 양식	계획서 양식	계획서 양식
포함되는 내용	I. 서론 (1~6) II. 본론 (1~3) 참고자료	I. 서론 (1~6) II. 본론 (1~4) 참고자료	I. 서론 (1~6) II. 본론 (1~5) 참고자료	I. 서론 (1~6) II. 본론 (1~7) 참고자료	I II III

이 문서는 한국산업기술대학교 컴퓨터공학부의  
“종합설계” 교과목에서 프로젝트  
“다중생체인식을 활용한 금고”를 수행하는  
(S5-11, 임 현성, 김 경석, 김 현정)들이 작성한 것으로  
사용하기 위해서는 팀원들의 허락이 필요합니다.

# 목 차

## I. 서론

1. 작품선정 배경 및 필요성 .....	1
2. 기존 연구/기술동향 분석 .....	1
3. 개발 목표 .....	2
4. 팀 역할 분담 .....	2
5. 개발 일정 .....	3
6. 개발 환경 .....	3

## II. 본론

1. 개발 내용 .....	3
2. 문제 및 해결방안 .....	3
3. 시험시나리오 .....	4
4. 상세 설계 .....	5
5. Prototype 구현 .....	11
6. 시험/ 테스트 결과 .....	14
7. Coding & DEMO .....	15

## III. 결론

1. 연구 결과 .....	22
2. 작품제작 소요재료 목록 .....	23

참고자료 .....	24
------------	----

# 1. 서론

## 가. 작품선정 배경 및 필요성

### 1) 작품선정 배경

과거에는 사용자 인증방법으로 비밀번호, 패턴 등이 사용되었다. 그러나 이러한 인증 방법은 타인이 도용하기 쉬워 보안에 취약하고 암기를 요구하기 때문에 편리성이 낮았다. 이러한 상황에서 생체인식을 통한 인증은 보안과 편의의 측면에서 과거의 인증방법에 비해 뛰어나기 때문에 새로운 인증 수단으로 대두되었다. 하지만, 단일 생체 인식은 환경에 따라 쉽게 인식률의 변동 폭이 크다. 이 점을 보완하기 위해서 효율적으로 인식률을 높이는 방법인 멀티 모달 인식(단일 생체 인식의 융합) 기술을 채택하여 생체 인식 기술의 인식률을 높이고자 한다.

### 2) 작품선정 필요성

단일 생체 인식만으로 사용자 인식률을 높이는 데에 한계가 존재한다. 인식률을 올리기 위해서는 많은 비용과 시간이 소요된다. 이처럼 단일 생체 인식의 한계점이나, 생체 인식 부위의 변질, 조명 등 환경적인 요인으로 인해 인식률이 저하되는 상황에서 서로 다른 인식기술 병행을 통한 사용자 인증인 ‘멀티 모달 생체 인식’을 통한 인식률 증대로 단일 생체 인식 기술의 단점을 보완할 수 있다. 위에서 기술한 멀티 모달 생체 인식 기술을 금고에 접목시켜 보안성을 강화시키고자 한다.

## 나. 기존 연구/기술동향 분석

### 1) 단일 인식의 특징

< High - 좋음 / Low - 나쁨 >

본인 거부율(FRR): 생체 인식 시스템에서 등록된 사용자를 거부하는 확률  
타인 수락률(FAR): 비등록자를 등록자로 잘못 수락하는 확률

	지속성	수집성	위조/변조	본인 거부율	타인 수락률
지문	High	Medium	High	0.1% ~ 0.5%	0.001% ~ 0.01%
얼굴	Medium	High	Low	1% ~ 2.6%	1% ~ 1.3%
홍채	High	Medium	High	0.0001% ~ 0.1%	0.000083% ~ 0.0001%

< 표 1 >

가) 지문인식의 데이터인 지문은 동일한 형태가 발견된 적이 없고 지문의 형태는 평생 변하지 않는다는 점에서 위조 및 변조가 어렵다. 또한 지문 인식 기술은 비용이 저렴하고 안정성이 우수하다. 그러나 지문이 손상되어 데이터가 미 추출될 가능성이 존재한다. 현재 일반적인 지문인식은 본인 거부율이 0.1% ~ 0.5%이고 타인 수락률은 0.001% ~ 0.01%로 알고리즘에 따라 차이가 존재한다.

나) 얼굴 인식의 데이터인 얼굴은 수집이 간편하고 빠르며 인식 기술 비용이 적게 들지만 사진이나 영상 등 위조가 쉽고 노화, 상처, 성형 등 변화가 잦은 생체 데이터라는 단점이 있다. 조명과 같은 외부 환경에 영향을 받을 정도로 인식률이 낮다. 현재 일반적인 얼굴인식은 본인 거부율이 1% ~ 2.6%이고 타인 수락률은 1% ~ 1.3%로 알고리즘에 따라 차이가 존재한다.

다) 홍채 인식의 데이터인 홍채는 변함이 없어 위조가 불가능하지만 빛에 크게 영향을 받는다는 단점이 있다. 현재 일반적인 홍채인식은 본인 거부율이 0.0001% ~

0.1%이고 타인 수락률은 0.000083% ~ 0.0001%로 알고리즘에 따라 차이가 존재한다.

## 2) 구현할 멀티 모달

단일 생체 인식만으로 사용자를 완벽하게 인증하는 인식 기술을 구현하기 어렵다. 예시로 지문 인식은 지문이 훼손된 사람이 본인 인증을 할 수 없는 신뢰성의 한계가 있다. 이러한 단점들을 다른 생체 인식 기술을 통해 신뢰성 한계를 극복하여 상호 보완이 가능하다. 다중으로 생체 특징을 추출하여 판단하면 보다 정확하게 사용자를 인증하는 인식 기술을 구현할 수 있다. 이러한 멀티 모달의 방법에는 다중 센서, 다중 생체 특징, 동일 생체 특징의 다중 유닛, 동일 생체특징을 여러 번 획득, 동일 입력 생체 특징 신호에 대한 다중 표현과 매칭 알고리즘 등이 있다. 구현예정인 금고에 접목할 멀티 모달 방법 중 다수의 서로 다른 생체 정보를 사용하는 ‘다중 생체 특징’ 방법을 채택했다. 각 단일 생체인식 시스템이 독립적으로 인식하여 판단한 값을 융합하여 최종적으로 인식하는 ‘판단 융합 다중 생체 인식 방식’ 활용하여 구현할 계획이다.

## 다. 개발 목표

기존 생체 인식 API를 기반으로 ‘판단 융합 다중 생체 인식 방식’ 활용하여 멀티 모달 알고리즘을 설계해 인식률을 증가시키는 멀티 모달을 구현한다. 이렇게 구현한 멀티 모달을 이용하여 금고의 보안성을 강화하고, 더 나아가 금고 폐쇄장치를 어플에서만 제어함으로써 방도성을 강화한다. 결과적으로는 금고, 어플, 서버로 구성된 3-Tier를 통한 체계적인 시스템 구축을 목표로 한다.

## 라. 팀 역할 분담

### 1) 아두이노를 통한 금고 개발

- 가) 임현성 : 회로 제작, 진동 센서, 자이로센서 구현, 데모 테스트
- 나) 김경석 : 모듈 연동, 서브모터, LCD 구현
- 다) 김현정 : 부품별 물리적 연결, 적외선센서, 와이파이 모듈 구현

### 2) 금고 제어를 위한 어플리케이션

- 가) 임현성 : 금고 관리 기능 구현
- 나) 김경석 : 금고 조회 기능 구현
- 다) 김현정 : 금고 해제 기능 구현, UI 구현

### 3) 판단 융합 다중 생체 인식 방식의 멀티 모달 구현

- 가) 임현성 : 홍채인식
- 나) 김경석 : 얼굴인식
- 다) 김현정 : 지문인식

### 4) 프로젝트 테스트

- 가) 임현성 : DB 구현 테스트, 통합 테스트
- 나) 김경석 : 멀티모달 구현 테스트, 연동 테스트

다) 김현정 : 어플 구현 테스트, 클라이언트 테스트

## 마. 개발 일정

추진 사항	12월	1월	2월	3월	4월	5월	6월	7월	8월	9월	10월
기초 자료 조사											
요구사항 분석											
시스템 설계											
멀티 모달 개발											
어플 개발											
아두이노 개발											
DB 구축 및 개발											
데모 및 디버깅											
시연/논문 작성											

< 표 2 >

## 바. 개발 환경

- 1) 어플리케이션: Android Studio / 개발 언어: JAVA
- 2) 아두이노: Arduino / 개발 언어: C/C++
  - 가) 메인보드: 아두이노 UNO wifi Rev2
  - 나) 서브모터: SG90
  - 다) 모션 센서: PIR Motion Sensor
  - 라) 진동 및 충격 센서: SW-420
  - 마) 자이로 및 가속도 센서: MPU-6050
- 3) DB: MySQL
- 4) Web Server: Apache
- 5) Web Application: Tomcat
- 6) Web: PHP

## 2. 본론

### 가. 개발 내용

- 1) 서버는 금고의 데이터를 받아와 금고 상태 저장, 어플리케이션 데이터 저장, 어플리케이션과 금고 사이의 데이터 송수신 기능을 구현한다.
- 2) 어플리케이션은 멀티 모달을 이용한 금고 해제 기능, 금고의 내/외부 및 상태 조회 기능, 금고 관리 기능, 어플리케이션 데이터를 서버로 송수신하는 기능을 구현한다.
- 3) 금고는 금고의 기울임 감지 기능, 진동 감지 기능, 금고 개폐 기능, 금고의 데이터를 서버로 송수신하는 기능을 구현한다.

### 나. 문제 및 해결방안

#### 1) 멀티 모달의 인식률 문제

기존에 제공되는 API들을 활용하여 인식 결과 값을 통해 최적의 인식률을 도출할 수 있는 알고리즘 설계, 결과 값들의 허용 범위를 조절하여 몇 가지 모델을 구현 한 후 가장 이상적인 인식률을 가지는 모델을 채택

#### 2) 어플리케이션 - 서버 - 금고의 데이터 송수신시 보안성 문제

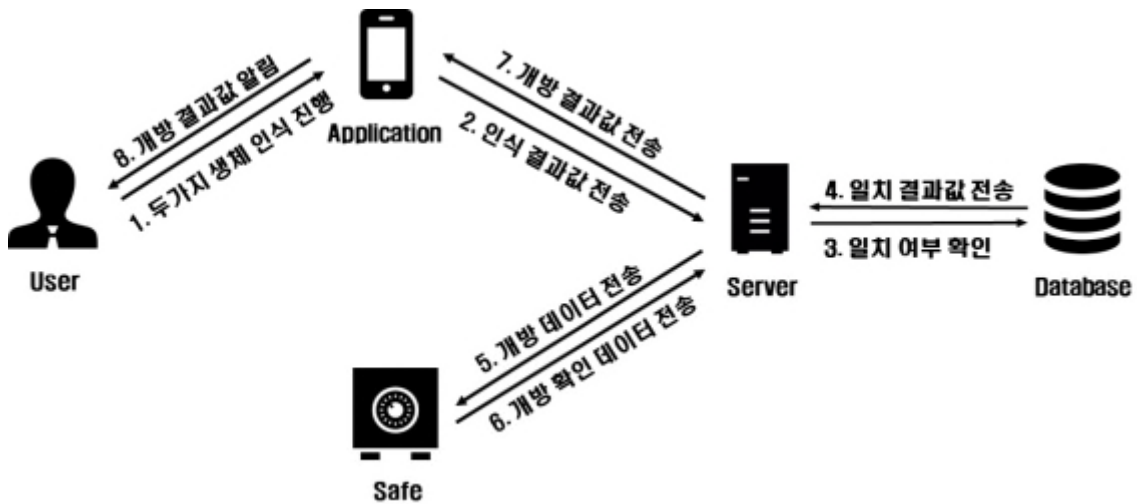
데이터를 서버를 통해 송수신 되는 것이기에, 송수신 신호를 가로채거나 임의로 신호를 보낼 수 있는 위험이 존재하기 때문에 송수신 되는 신호를 암호화해야하는 필요성이 있음

#### 3) 개별 금고의 다중 사용자 문제

금고별 대표 사용자를 두고 대표 사용자의 승인 하에 추가 사용자의 금고 제어 허용

### 다. 시험 시나리오

#### 1) 금고 잠금 해제



< 그림 1 >

가) 사용자가 어플리케이션에 멀티 모달 생체 인식을 진행한다.

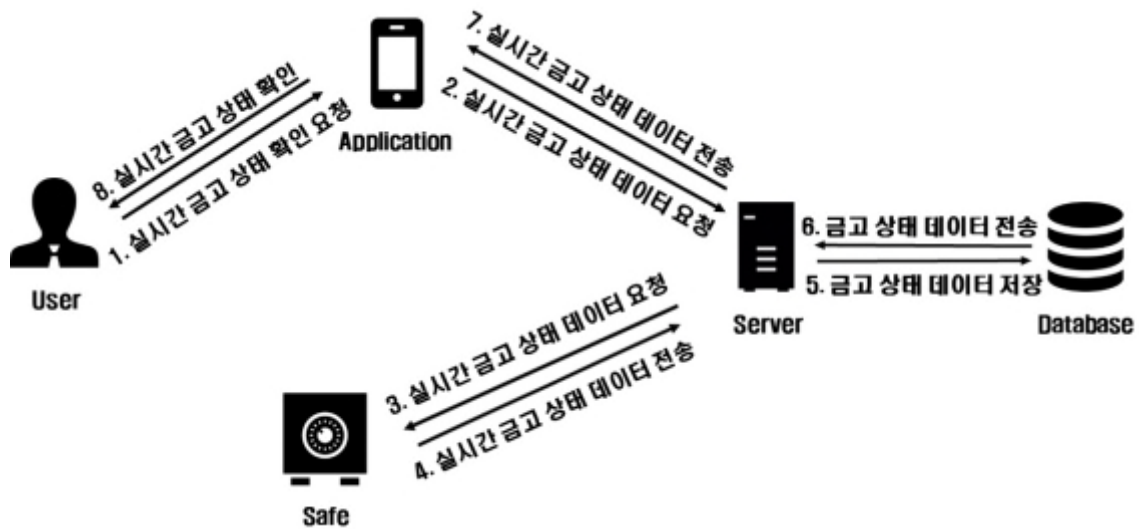
나) 어플리케이션은 인식된 결과 값을 서버에 전송한다.

다) 서버는 수신된 결과 값을 데이터베이스에서 생체 인식의 일치 여부를 확인하고, 일치 결과 값을 서버에 전송한다.

라) 서버는 어플리케이션과 금고에 개방 데이터를 전송하고, 금고를 개방시킨다. 금고는 개방될 시, 개방 확인 데이터를 서버에 전송한다.

마) 어플리케이션은 수신한 개방 데이터 값을 사용자에게 알린다. 사용자는 알림을 받고 금고를 개방한다.

#### 2) 금고 상태 확인



< 그림 2 >

- 가) 사용자가 어플리케이션으로 실시간 금고 상태를 확인한다.
- 나) 어플리케이션은 전달 받은 요청을 서버에 전송하고 금고를 통해 요청하고 데이터를 전달 받는다.
- 다) 서버는 금고로부터 전달 받은 실시간 금고 상태를 데이터베이스에 저장한다.
- 라) 서버는 데이터베이스의 실시간 금고 상태 정보를 어플리케이션에 전송하여 사용자가 확인한다.

### 3) 멀티 모달 인증



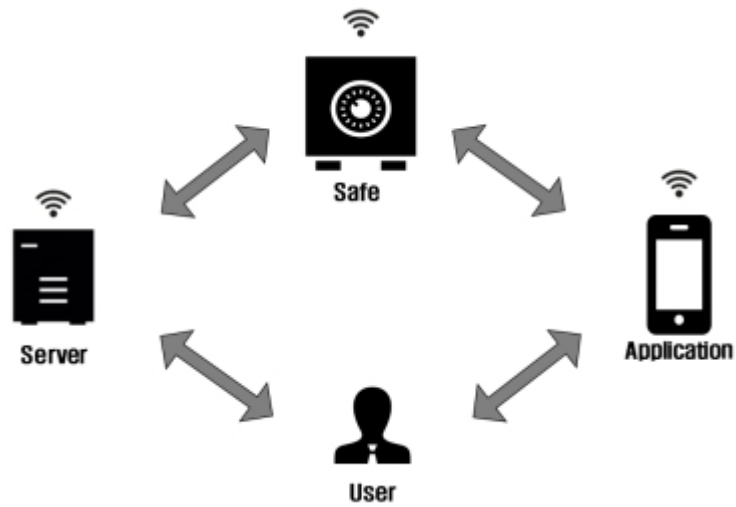
< 그림 3 >

- 가) 사용자는 두 가지 생체인식(얼굴, 지문)을 진행한다.
- 나) 인증에 대한 결과 데이터를 받아 멀티 모달 인증을 진행한다.
- 다) 멀티 모달 인증을 진행하여 나온 결과 값으로 접근 권한을 허용 받아 어플리케이션에 접속한다.

## 라. 상세 설계

### 1) 시스템 구성도





< 그림 4 >

가) 서버

(1) 금고 데이터

- (가) 금고 이벤트 발생 확인
- (나) 금고 실시간 모니터링
- (다) 금고 잠금장치 관리
- (라) 데이터 송수신

(2) 어플리케이션 데이터

- (가) 생체 데이터 관리
- (나) 멀티 모달 데이터 확인
- (다) 데이터 송수신

나) 어플리케이션 구성도

(1) 금고 상태 조회

- (가) 금고 이벤트 발생 확인
- (나) 금고 실시간 모니터링
- (다) 데이터 송수신

(2) 금고 관리

- (가) 생체 인식 등록 및 관리
- (나) 멀티 모달을 이용한 해제
- (다) 데이터 송수신

다) 금고 구성도

(1) 아두이노

- (가) 금고 상태 조회
- (나) 금고 관리

- (다) LCD 상태 표시
- (라) 데이터 송수신

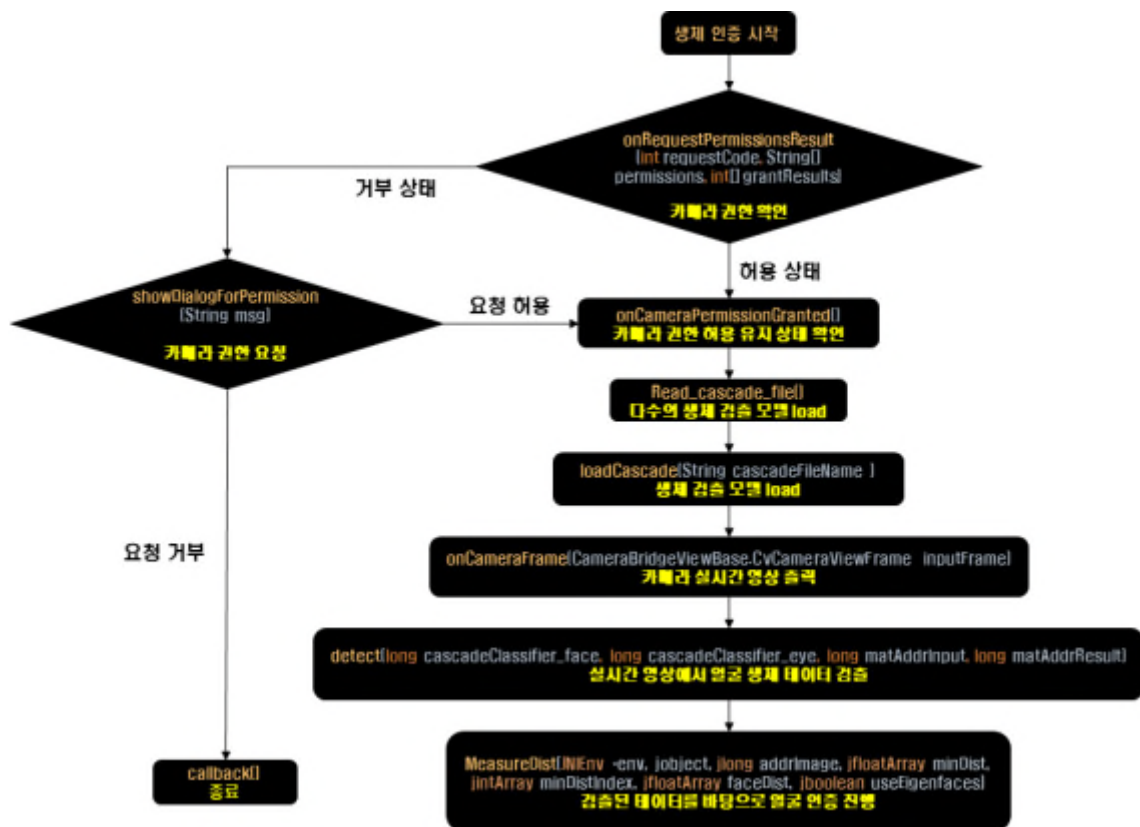
- (2) 금고 상태 조회
  - (가) 적외선 센서
  - (나) 진동 센서
  - (다) 자이로 센서
  - (라) 데이터 송수신

- (3) 금고 관리
  - (가) 서보 모터
  - (나) 데이터 송수신

## 2) 시스템 모듈 상세 설계

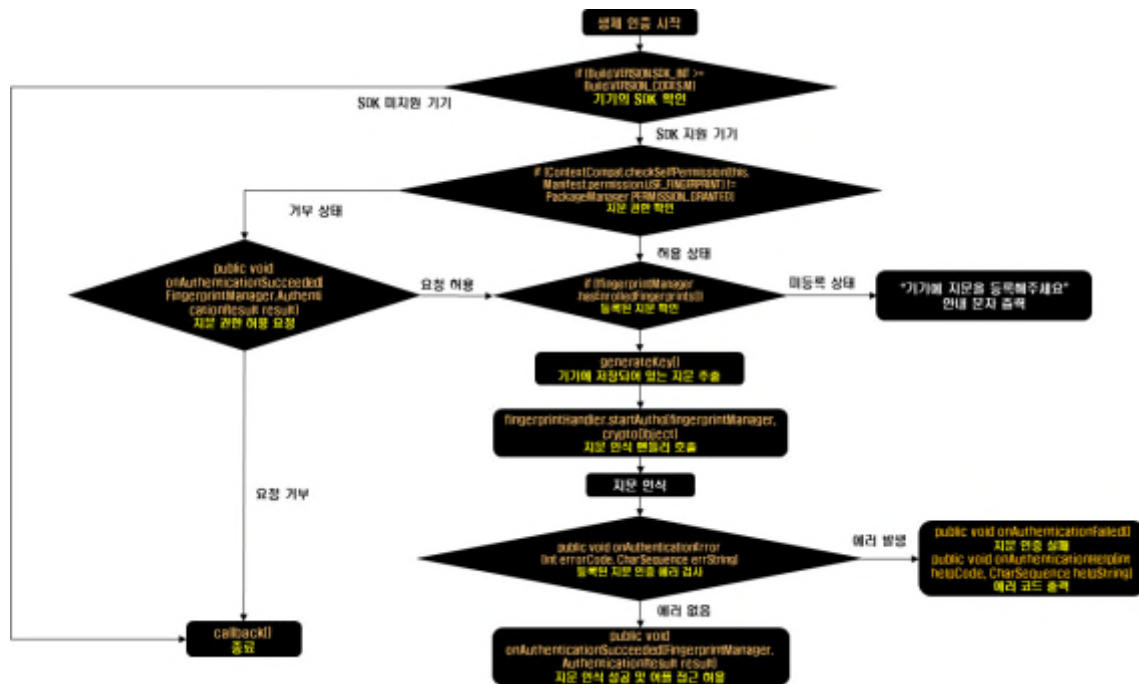
### 가) 어플리케이션

#### (1) 얼굴 인식



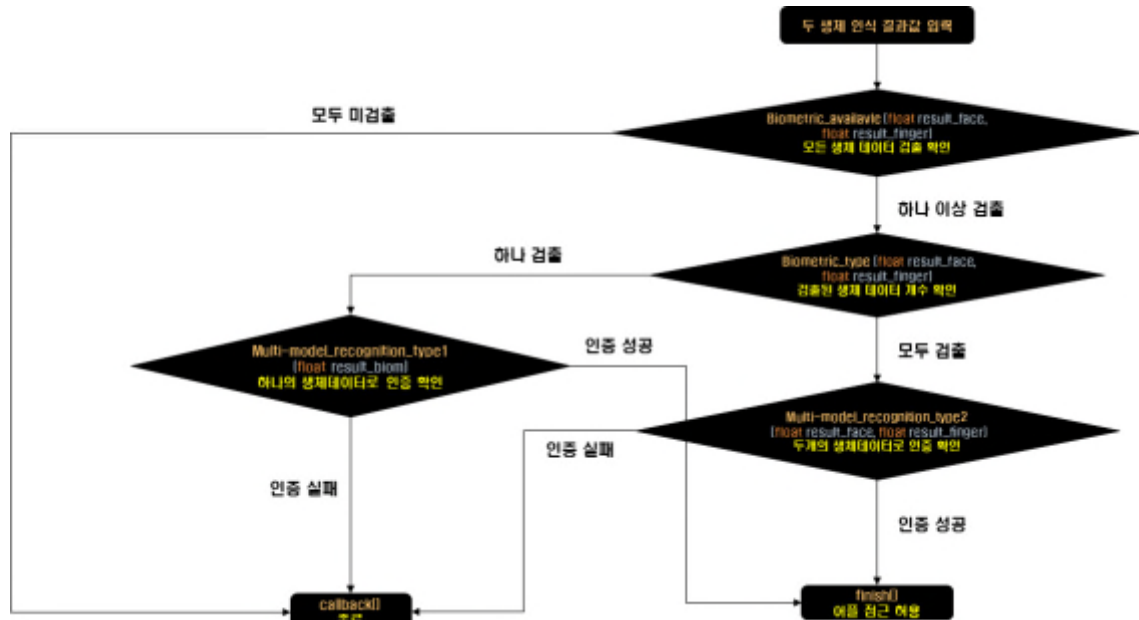
< 그림 5 >

#### (2) 지문 인식



< 그림 6 >

### (3) 멀티 모달 인식



< 그림 7 >

### 나) 서버

#### (1) 데이터베이스

테이블명	자이로센서	Gyro_Sensor		
논리 이름	물리 이름	도메인	데이터 타입	널 허용
* 자이로센서 상태 순서	* state_seq	시퀀스	INT UNSIGNED	NOT NULL
X평형 값	x_index_value	숫자	INTEGER	NOT NULL
Y평형 값	y_index_value	숫자	INTEGER	NOT NULL
Z평형 값	z_index_value	숫자	INTEGER	NOT NULL
이상상태	state	숫자	INT UNSIGNED	NOT NULL
발생일시	occur_time	일시	DATETIME	NOT NULL

< 표 6 >

테이블명	진동센서	Oscillation_Sensor		
논리 이름	물리 이름	도메인	데이터 타입	널 허용
* 진동센서 상태 순서	* state_seq	시퀀스	INT UNSIGNED	NOT NULL
진동센서 값	value	숫자	INTEGER	NOT NULL
이상상태	state	숫자	INT UNSIGNED	NOT NULL
발생일시	occur_time	일시	DATETIME	NOT NULL

< 표 7 >

테이블명	적외선센서	Infrared_Sensor		
논리 이름	물리 이름	도메인	데이터 타입	널 허용
* 적외선센서 상태 순서	* state_seq	시퀀스	INT UNSIGNED	NOT NULL
적외선센서 값	value	숫자	INTEGER	NOT NULL
이상상태	state	숫자	INT UNSIGNED	NOT NULL
발생일시	occur_time	일시	DATETIME	NOT NULL

< 표 8 >

테이블명	전원상태	Power_status		
논리 이름	물리 이름	도메인	데이터 타입	널 허용
* 전원 상태 순서	* state_seq	시퀀스	INT UNSIGNED	NOT NULL
전원상태 값	value	숫자	INTEGER	NOT NULL
이상상태	state	숫자	INT UNSIGNED	NOT NULL
발생일시	occur_time	일시	DATETIME	NOT NULL

< 표 9 >

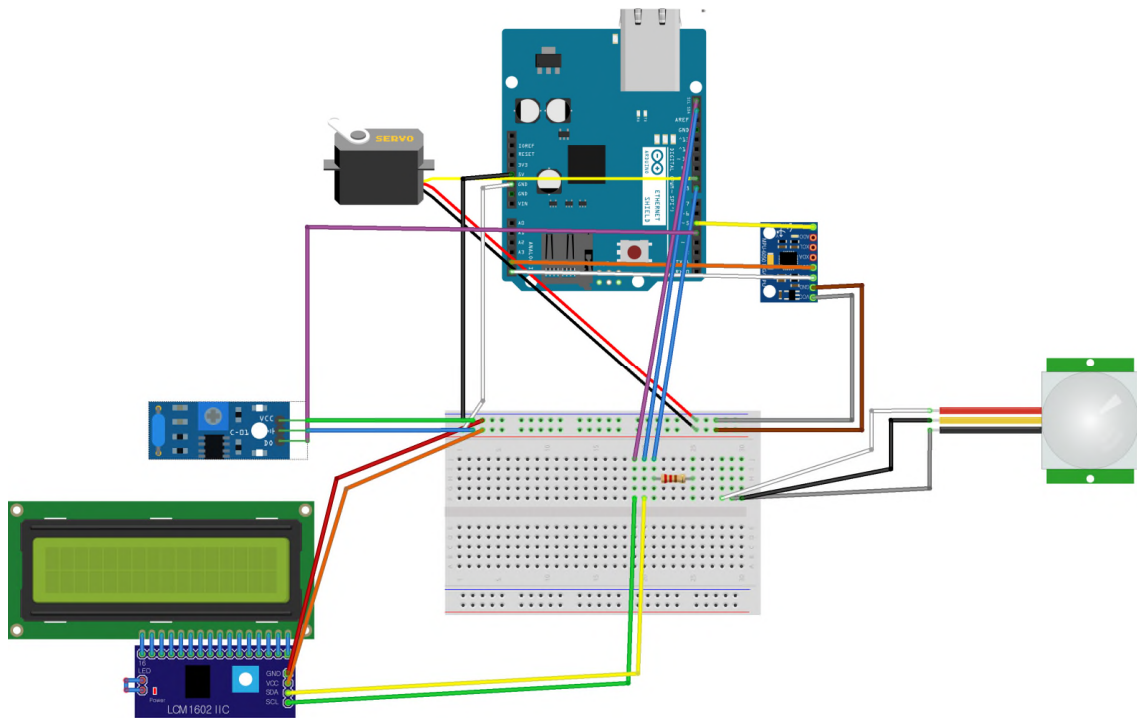
테이블명	개폐여부	Open_Close_status		
논리 이름	물리 이름	도메인	데이터 타입	널 허용
* 개폐여부 순서	* state_seq	시퀀스	INT UNSIGNED	NOT NULL
개폐여부 값	value	숫자	INTEGER	NOT NULL
이상상태	state	숫자	INT UNSIGNED	NOT NULL
열림일시	open_time	일시	DATETIME	NULL
닫힘일시	close_time	일시	DATETIME	NULL
발생일시	occur_time	일시	DATETIME	NOT NULL

< 표 10 >

테이블명	이상여부			state_change
논리 이름	물리 이름	도메인	데이터 타입	널 허용
* 이상여부 순서	* state_seq	시퀀스	INT UNSIGNED	NOT NULL
자이로 센서 이상상태	Gyro_state	숫자	INT UNSIGNED	NOT NULL
적외선 센서 이상상태	Infrared_state	숫자	INT UNSIGNED	NOT NULL
진동 센서 이상상태	Oscillation_state	숫자	INT UNSIGNED	NOT NULL
이상상태	state	숫자	INT UNSIGNED	NOT NULL
발생일시	occur_time	일시	DATETIME	NOT NULL

< 표 11 >

다) 금고 - 회로도



< 그림 8 >

## 마. Prototype 구현

### 1) 어플리케이션

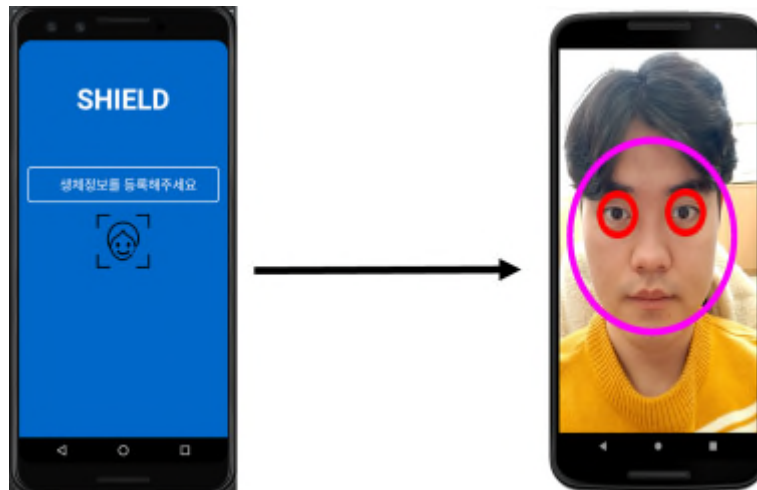
#### 가) Intro



< 그림 9 >

- (1) 회원가입을 통해서 사용자 등록
- (2) 등록된 사용자와 생체정보의 일치함을 확인하기 위함(예시 - 은행 어플리케이션)
- (3) 기기변경 및 타 기기에서의 로그인을 위한 회원 기능

#### 나) 생체 정보 얼굴 인식



< 그림 10 >

- (1) 얼굴 인식
  - (가) 대상의 얼굴을 우선적으로 검출
  - (나) 대상의 검출된 얼굴에서 눈 탐색
  - (다) 탐색된 데이터들을 통해서 얼굴 인식
  - (라) 생체 정보 저장

다) 생체 정보 지문 인식



< 그림 11 >

(1) 지문인식

- (가) 대상의 지문을 검출
- (나) 탐색된 데이터들을 통해서 지문 인식
- (다) 생체 정보 저장

라) 다중 인식



- (1) 얼굴인증 : 인식값 도출
- (2) 지문인증 : 인식값 도출
- (3) 다중인식
  - (가) 얼굴 인식값과 지문 인식값을 이용하여 인증 판단
  - (나) 인증이 승인될 경우 금고 개방 (아두이노 서브모터 작동)

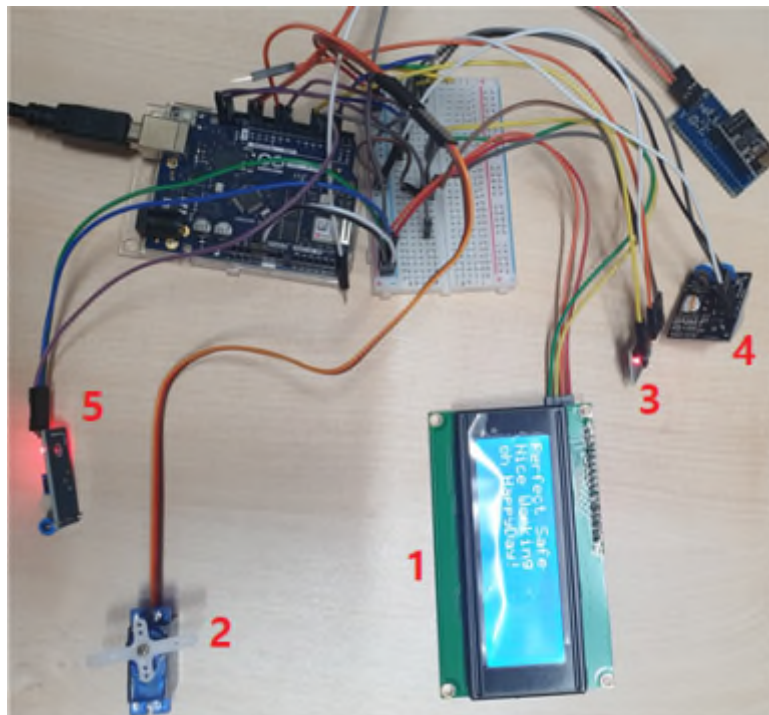
마) 금고 관리 메인 페이지



< 그림 13 >

- (1) 어플리케이션을 통한 금고 동작 제어
- (2) 금고 등록 현황 확인
- (3) 이벤트 발생 리스트 확인
- (4) Information - Detail 확인
- (5) 여러 개의 다중 금고를 원격으로 확인 가능

바) 금고



- (1) LCD : 금고 상태 확인
- (2) 서브모터 : 금고 개폐 기능
- (3) 자이로 센서 : 금고 기울임 감지 기능
- (4) PIR 센서 : 금고 개폐 여부 확인 기능



(5) 충격 센서 : 금고 충격 감지 기능

사) 금고의 센서값 확인

```
COM8

<<PIR 값>>
1 = Block

-----

<<자이로센서 값>>
AcX = -1 | AcY = -1 | AcZ = -1 | Tmp = 36.53 | GyX = -1 | GyY = -1 | GyZ = -1

-----

<<진동 값>>
measurment(Vibration) = 0

-----

<<서보 모터 값>>
서보모터: 0도
서보모터: 90도

-----
```

## 바. 시험/ 테스트 결과

대상	기능	구현
Android App	회원가입 기능	○
	로그인 / 로그아웃 기능	○
	아이디 / 비밀번호 찾기 기능	○
	사용자 금고 목록 조회	○
	금고 정보 조회	○
	금고 이벤트 조회	○
	금고 개폐 기능	○
	얼굴 등록 기능	○
	얼굴 인식 기능	○
	지문 인식 기능	○
	멀티 모달 기능	○
	개인정보 수정 기능	○
	금고 수정 기능	○
	Validation	△
	서버와 데이터 송수신 기능	○
	Database와 연동 기능	○
Arduino	금고 상태 전송 기능	○
	금고 데이터 전송 기능	○
	LCD 표시 기능	○
	서버와 데이터 송수신 기능	○

	적외선 센서 동작 기능	○
	자이로 센서 동작 기능	○
	서보 모터 동작 기능	○
	진동 센서 동작 기능	○
	Database와 연동 기능	○
Server	금고 이벤트 발생 확인	○
	금고 실시간 모니터링	○
	금고 잠금장치 관리	○
	데이터 송수신	○
	생체 데이터 관리	○
	멀티 모달 데이터 확인	○
DB	사용자 정보 저장	○
	금고 정보 저장	○
	적외선 센서 정보 저장	○
	금고 개폐 여부 정보 저장	○
	진동 센서 정보 저장	○
	자이로 센서(기울임 감지) 정보 저장	○
	금고 전원 상태 정보 저장	○
	금고 이상 여부 판단 정보 확인	○

## 사. Coding & DEMO

### 1) 어플리케이션

#### 가) 얼굴인식

```

Mat imgRgba = inputFrame.rgba();
Mat img = new Mat();
imgRgba.copyTo(img);
List<Mat>          images          =          ppF.getProcessedImage(img,
PreProcessorFactory.PreprocessingMode.RECOGNITION);
Rect[] faces = ppF.getFacesForRecognition();

// Selfie / Mirror mode
if(front_camera){
    Core.flip(imgRgba, imgRgba, 1);
}
if(images == null || images.size() == 0 || faces == null || faces.length == 0 || !
(images.size() == faces.length)){
    // skip
    return imgRgba;
} else {
    faces = MatOperation.rotateFaces(imgRgba, faces, ppF.getAngleForRecognition());
    for(int i = 0; i<faces.length; i++){

```

```

        MatOperation.drawRectangleAndLabelOnPreview(imgRgba, faces[i],
rec.recognize(images.get(i), ""), front_camera);
        if(rec.recognize(images.get(i), "").equals("rla99tjr")){
            try{
                Thread.sleep(2000);
            } catch (InterruptedException e) {
                e.printStackTrace();
            }
            startActivity(new Intent(getApplicationContext(),
FingerInterface.class));
        }
        //Log.e(this.getClass().getName(), rec.recognize(images.get(i), "")+"와우");
    }
    //startActivity(new Intent(getApplicationContext(), finger_inter.class));
    return imgRgba;
}

```



#### 나) 지문인식

```

private KeyPair generateKeyPair(String keyName, boolean
invalidatedByBiometricEnrollment) throws Exception {
    KeyPairGenerator keyPairGenerator =
KeyPairGenerator.getInstance(KeyProperties.KEY_ALGORITHM_EC, "AndroidKeyStore");

    KeyGenParameterSpec.Builder builder = new KeyGenParameterSpec.Builder(keyName,
KeyProperties.PURPOSE_SIGN)

```

```

        .setAlgorithmParameterSpec(new ECGenParameterSpec("secp256r1"))
        .setDigests(KeyProperties.DIGEST_SHA256,
                    KeyProperties.DIGEST_SHA384,
                    KeyProperties.DIGEST_SHA512)
        // Require the user to authenticate with a biometric to authorize every
use of the key
        .setUserAuthenticationRequired(true)
        .setInvalidatedByBiometricEnrollment(invalidatedByBiometricEnrollment);

keyPairGenerator.initialize(builder.build());

return keyPairGenerator.generateKeyPair();
}

```



## 2) 아두이노(금고)

### 가) Gyro 센서 동작 함수

```

void Gyro() {
    Wire.beginTransmission(MPU_addr);
    Wire.write(0x3B); // starting with register 0x3B (ACCEL_XOUT_H)
    Wire.endTransmission(false);
    Wire.requestFrom(MPU_addr, 14, true); // request a total of 14 registers
    AcX = Wire.read() << 8 | Wire.read(); // 0x3B (ACCEL_XOUT_H) & 0x3C (ACCEL_XOUT_L)
    AcY = Wire.read() << 8 | Wire.read(); // 0x3D (ACCEL_YOUT_H) & 0x3E (ACCEL_YOUT_L)
    AcZ = Wire.read() << 8 | Wire.read(); // 0x3F (ACCEL_ZOUT_H) & 0x40 (ACCEL_ZOUT_L)
}

```

```

GyX = Wire.read() << 8 | Wire.read(); // 0x43 (GYRO_XOUT_H) & 0x44 (GYRO_XOUT_L)
GyY = Wire.read() << 8 | Wire.read(); // 0x45 (GYRO_YOUT_H) & 0x46 (GYRO_YOUT_L)
GyZ = Wire.read() << 8 | Wire.read(); // 0x47 (GYRO_ZOUT_H) & 0x48 (GYRO_ZOUT_L)

Serial.println("<<자이로 센서 값>>");
Serial.print("AcX = "); Serial.print(AcX);
Serial.print(" | AcY = "); Serial.print(AcY);
Serial.print(" | AcZ = "); Serial.print(AcZ);
Serial.print(" | GyX = "); Serial.print(GyX);
Serial.print(" | GyY = "); Serial.print(GyY);
Serial.print(" | GyZ = "); Serial.println(GyZ);

delay(1000);
}

```

나) Mootor 동작 함수

```

void Motor_OPEN() {
    Serial.println("<<서보 모터 값>>");
    myservo.write(0); // myservo를 0도가 되도록 움직입니다.
    delay(1000);      // 1초동안 기다립니다.
    Serial.println("서보모터: 0도(금고 열림)");
}

void Motor_CLOSE() {
    Serial.println("<<서보 모터 값>>");
    myservo.write(90); // myservo를 90도가 되도록 움직입니다.
    delay(1000);      // 1초동안 기다립니다.
    Serial.println("서보모터: 90도(금고 닫힘)");
}

```

다) PIR 동작 함수

```

void PIR() {
    Serial.println("<<PIR 값>>");

    // 적외선 인체감지 센서에서 값을 읽는다
    // 막으면 1 , 없으면 0
    int sensor = digitalRead(PIR_pin);

    if (sensor == 1 ) // 막혔을 시
    {
        Serial.print(sensor);
    }
}

```

```

        Serial.println(" = Block");
    }
    else {
        Serial.print(sensor);
        Serial.println(" = Open");
    }
    delay(1000);
}

```

라) 진동 동작 함수

```

void Vibration() {
    Serial.println("<<진동 값>>");
    long measurement = TP_init();

    Serial.print("measurment(Vibration) = ");
    Serial.println(measurement);

    delay(1000);
}

long TP_init() {
    delay(10);
    long measurement = pulseIn (Vib_pin, HIGH);
    return measurement;
}

```

3) 서버

가) Mariadb - DBeaver : 데이터베이스 값 연동 확인

```

root@kali:~# mysql -u root -p
Enter password:
ERROR 1045 (28000): Access denied for user 'root'@'localhost' (using password: YES)
root@kali:~# mysql -u root -p
Enter password:
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 15
Server version: 10.2.32-MariaDB-10.2.32+maria-serial-log mariadb.org binary distribution
Copyright (c) 2000, 2019, Oracle, MariaDB Corporation AB and others.
Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

MariaDB [(none)]>
MariaDB [(none)]>
MariaDB [(none)]>
MariaDB [(none)]> show databases
+-----+
| Database |
+-----+
| information_schema |
| mysql |
| performance_schema |
+-----+
4 rows in set (0.00 sec)

MariaDB [(none)]>

```

Table	Engine	Auto Increment	Data Length	Description
equ_info	InnoDB	1	19KB	물고
equ_sensor	InnoDB	1	19KB	자외선 센서
confused_sensor	InnoDB	1	19KB	진동 센서
equ_sensor_status	InnoDB	1	19KB	계측 여부
equ_sensor_status	InnoDB	1	19KB	진동 센서
power_status	InnoDB	1	19KB	전원 상태
status_change	InnoDB	1	19KB	이상 여부
user_info	InnoDB	4	19KB	사용자 정보

#### 나) Apache 2.4 설치 버전 확인

```
apache2.conf    conf-enabled  magic          mods-enabled  sites-available
conf-available  envvars        mods-available ports.conf    sites-enabled
root@kgs:/etc/apache2# cat apache2.conf
# This is the main Apache server configuration file.  It contains the
# configuration directives that give the server its instructions.
# See http://httpd.apache.org/docs/2.4/ for detailed information about
# the directives and /usr/share/doc/apache2/README.Debian about Debian specific
# hints.
#
#
# Summary of how the Apache 2 configuration works in Debian:
# The Apache 2 web server configuration in Debian is quite different to
# upstream's suggested way to configure the web server. This is because Debian's
# default Apache2 installation attempts to make adding and removing modules,
# virtual hosts, and extra configuration directives as flexible as possible, in
# order to make automating the changes and administering the server as easy as
# possible.
#
# It is split into several files forming the configuration hierarchy outlined
# below, all located in the /etc/apache2/ directory:
#
#     /etc/apache2/
#     |-- apache2.conf
#     |   |-- ports.conf
#     |-- mods-enabled
#     |   |-- *.load
#     |   |-- *.conf
#     |-- conf-enabled
#     |   |-- *.conf
#     |-- sites-enabled
#         |-- *.conf
#
#
# * apache2.conf is the main configuration file (this file). It puts the pieces
# together by including all remaining configuration files when starting up the
# web server.
#
```

다) Java8(JDK 1.8) 설치 버전 확인

```
root@kgs: ~  
update-alternatives: using /usr/lib/jvm/java-8-openjdk-amd64/jre/bin/policytool to provide /usr/bin/java-policytool  
Setting up openjdk-8-jdk-headless:amd64 (8u252-b09-1-16.04) ...  
update-alternatives: using /usr/lib/jvm/java-8-openjdk-amd64/bin/ldlj to provide /usr/bin/ldlj  
update-alternatives: using /usr/lib/jvm/java-8-openjdk-amd64/bin/jdeps to provide /usr/bin/jdeps  
update-alternatives: using /usr/lib/jvm/java-8-openjdk-amd64/bin/wsimport to provide /usr/bin/wsimport  
update-alternatives: using /usr/lib/jvm/java-8-openjdk-amd64/bin/rmic to provide /usr/bin/rmic  
update-alternatives: using /usr/lib/jvm/java-8-openjdk-amd64/bin/jinfo to provide /usr/bin/jinfo  
update-alternatives: using /usr/lib/jvm/java-8-openjdk-amd64/bin/jsadebugd to provide /usr/bin/jsadebugd  
update-alternatives: using /usr/lib/jvm/java-8-openjdk-amd64/bin/native2ascii to provide /usr/bin/native2ascii  
update-alternatives: using /usr/lib/jvm/java-8-openjdk-amd64/bin/jstat to provide /usr/bin/jstat  
update-alternatives: using /usr/lib/jvm/java-8-openjdk-amd64/bin/javadoc to provide /usr/bin/javadoc  
update-alternatives: using /usr/lib/jvm/java-8-openjdk-amd64/bin/javah to provide /usr/bin/javah  
update-alternatives: using /usr/lib/jvm/java-8-openjdk-amd64/bin/clhsdb to provide /usr/bin/clhsdb  
update-alternatives: using /usr/lib/jvm/java-8-openjdk-amd64/bin/jstack to provide /usr/bin/jstack  
update-alternatives: using /usr/lib/jvm/java-8-openjdk-amd64/bin/jrunscript to provide /usr/bin/jrunscript  
update-alternatives: using /usr/lib/jvm/java-8-openjdk-amd64/bin/javac to provide /usr/bin/javac  
update-alternatives: using /usr/lib/jvm/java-8-openjdk-amd64/bin/javap to provide /usr/bin/javap  
update-alternatives: using /usr/lib/jvm/java-8-openjdk-amd64/bin/jar to provide /usr/bin/jar  
update-alternatives: using /usr/lib/jvm/java-8-openjdk-amd64/bin/extcheck to provide /usr/bin/extcheck  
update-alternatives: using /usr/lib/jvm/java-8-openjdk-amd64/bin/hsdb to provide /usr/bin/hsdb  
update-alternatives: using /usr/lib/jvm/java-8-openjdk-amd64/bin/schemagen to provide /usr/bin/schemagen  
update-alternatives: using /usr/lib/jvm/java-8-openjdk-amd64/bin/jps to provide /usr/bin/jps  
update-alternatives: using /usr/lib/jvm/java-8-openjdk-amd64/bin/xjc to provide /usr/bin/xjc  
update-alternatives: using /usr/lib/jvm/java-8-openjdk-amd64/bin/jmap to provide /usr/bin/jmap  
update-alternatives: using /usr/lib/jvm/java-8-openjdk-amd64/bin/jstatd to provide /usr/bin/jstatd  
update-alternatives: using /usr/lib/jvm/java-8-openjdk-amd64/bin/jhat to provide /usr/bin/jhat  
update-alternatives: using /usr/lib/jvm/java-8-openjdk-amd64/bin/jdb to provide /usr/bin/jdb  
update-alternatives: using /usr/lib/jvm/java-8-openjdk-amd64/bin/serialver to provide /usr/bin/serialver  
update-alternatives: using /usr/lib/jvm/java-8-openjdk-amd64/bin/wsgen to provide /usr/bin/wsgen  
update-alternatives: using /usr/lib/jvm/java-8-openjdk-amd64/bin/jcmd to provide /usr/bin/jcmd  
update-alternatives: using /usr/lib/jvm/java-8-openjdk-amd64/bin/jarsigner to provide /usr/bin/jarsigner  
Setting up openjdk-8-jdk:amd64 (8u252-b09-1-16.04) ...  
update-alternatives: using /usr/lib/jvm/java-8-openjdk-amd64/bin/appletviewer to provide /usr/bin/appletviewer  
update-alternatives: using /usr/lib/jvm/java-8-openjdk-amd64/bin/jconsole to provide /usr/bin/jconsole  
Processing triggers for libc-bin (2.23-0ubuntu5) ...  
Processing triggers for systemd (229-4ubuntu13) ...  
Processing triggers for ureadahead (0.100.0-19) ...  
Processing triggers for ca-certificates (20160104ubuntu1) ...  
Updating certificates in /etc/ssl/certs...  
0 added, 0 removed; done.  
Running hooks in /etc/ca-certificates/update.d...  
  
done.  
done.  
root@kgs:~# java -version  
openjdk version "1.8.0_252"  
OpenJDK Runtime Environment (build 1.8.0_252-8u252-b09-1-16.04-b09)  
OpenJDK 64-Bit Server VM (build 25.252-b09, mixed mode)  
root@kgs:~#
```



## 라) Tomcat8 설치 버전 확인

```
root@kgs: /usr/share/tomcat8
while [ -h "$PRG" ] ; do
  ls=`ls -ld "$PRG"`
  link=`expr "$ls" : '.*-> \(.*)$'`
  if expr "$link" : '/.*' > /dev/null; then
    PRG="$link"
  else
    PRG=`dirname "$PRG"`/"$link"
  fi
done

PRGDIR=`dirname "$PRG"`
EXECUTABLE=catalina.sh

# Check that target executable exists
if $os400; then
  # -x will Only work on the os400 if the files are:
  # 1. owned by the user
  # 2. owned by the PRIMARY group of the user
  # this will not work if the user belongs in secondary groups
  eval
else
  if [ ! -x "$PRGDIR"/"$EXECUTABLE" ]; then
    echo "Cannot find $PRGDIR/$EXECUTABLE"
    echo "The file is absent or does not have execute permission"
    echo "This file is needed to run this program"
    exit 1
  fi
fi

exec "$PRGDIR"/"$EXECUTABLE" version "$@"
root@kgs:/usr/share/tomcat8/bin# cd ..
root@kgs:/usr/share/tomcat8# sudo /usr/share/tomcat8/bin/version.sh
sudo: unable to resolve host kgs
Using CATALINA_BASE:   /usr/share/tomcat8
Using CATALINA_HOME:   /usr/share/tomcat8
Using CATALINA_TMPDIR: /usr/share/tomcat8/temp
Using JRE_HOME:        /usr
Using CLASSPATH:       /usr/share/tomcat8/bin/bootstrap.jar:/usr/share/tomcat8/bin/tomcat
i.jar
Server version: Apache Tomcat/8.0.32 (Ubuntu)
Server built:   Jan 24 2020 16:24:30 UTC
Server number:  8.0.32.0
OS Name:       Linux
OS Version:    4.4.0-59-generic
Architecture:  amd64
JVM Version:   1.8.0_252-8u252-b09-1~16.04-b09
JVM Vendor:    Private Build
root@kgs:/usr/share/tomcat8#
```

## 3. 결론

### 가. 연구 결과

- 1) 사용자 인식률을 높이는데 단일 생체인식만으로는 한계가 있어, '멀티 모달 생체 인식'을 통해서 단일 생체인식의 단점을 보완하였다.
- 2) 또한, 멀티모달 생체 인식기능을 통하여 단일 생체인식의 인식 오류를 해결함으로써 금고에 접근할 수 있는 사용자를 더욱더 정확하게 인식하여 금고의 보안성을 강화하였다.

- 3) 서버를 통해 금고와 애플리케이션 간에 Wifi 통신과 FCM을 활용하여, 금고 원격 접근을 통해 시간과 장소에 제한적이지 않고 사용자가 금고의 상태를 실시간으로 확인 및 관리 할 수 있도록 하여 접근성과 효율성을 증대시켰다.
- 4) 금고의 센서 이상 감지 여부를 서버와 실시간으로 통신하며, 이상징후 발생 시 즉시 사용자에게 알림을 전송하고 사용자 인증을 받은 애플리케이션으로만 금고의 개폐를 조작할 수 있게 함으로써 금고 방도성을 강화하고 관리의 효율성 및 편의성을 증대시켰다.

#### 나. 작품제작 소요재료 목록

아두이노 우노 R3	20x4 LCD 모듈 + 어댑터	BAP- FT90B 마이크로 서보 모터 180도
		
MPU-6050 3축 자이로 센서 + GY521 3축 가속도 센서	인체 감지 모션 센서	SW-420 진동, 충격 센서모듈
		
ESP-01 WiFi 모듈 + 어댑터	커넥터 M-M	막대저항 R1/4W-1K옴
		
USB 2.0 A-B 케이블	전원 어댑터	



#### 4. 참고 자료

[https://ettrends.etri.re.kr/ettrends/91/0905000569/20-1\\_084\\_092.pdf](https://ettrends.etri.re.kr/ettrends/91/0905000569/20-1_084_092.pdf)

김연희 외 3명(2005). 다중생체 인식 기술 동향. 유비쿼터스 사회와 정보보호 특집 논문. 20(1), 84 - 92