

과제 최종보고서

과제명	스마트 거울			
지도교수	배 종 우 교수님			
연구원	이 름	학 번	전화번호	E-mail
	방윤환	60091998	010 2003 0582	kebin1104@nate.com

과제에 대한 최종보고서를 첨부와 같이 제출합니다.

2016 년 6 월 7 일

제출자 : 방 윤 환

목차

1. 과제 개요

2. 내용

2.1 목표설정

- 2.1.1 목표설정 전 요구사항 파악
- 2.1.2 요구사항 파악을 토대로 목표설정

2.2 분석

2.3 제작

3. 참고 문헌

(소스코드 별첨)

1. 과제 개요

‘하루에 한번 이상 보게 되는 거울을 스마트하게 만들어 삶의 편리함을 더해주면 좋겠다.’라는 생각에서 아이디어가 시작되었다. 거울에 비치는 자신의 모습을 보는 동시에 자신에게 필요한 정보들까지 확인할 수 있게 하는 것을 구상하였다. 경찰서 취조실에 있는 거울과 같이 한쪽 면은 거울, 다른 한쪽 면은 유리처럼 보이는 원웨이 미러 필름과 모니터를 이용하여 제작하는 방법을 기획하였다. 또한 카메라를 이용하여 영상처리 기술을 통한 스마트 거울의 기능 제어도 추가적으로 기획하게 되었다.

2. 내용

2.1 목표설정

2.1.1 목표설정 전 요구사항 파악

요구사항 ID	상세	요구사항 설명	수용여부	미수용 사유	반영 방법
1	날짜확인	사용자는 스마트 거울 앞에 있을 경우 해당 날짜를 확인할 수 있다.	Y		안드로이드 자체 날짜 정보를 받아와서 출력한다.
2	날씨정보 확인	사용자가 원하는 지역(대한민국 내)의 오전, 오후, 저녁, 밤 날씨 정보를 확인할 수 있다.	Y		지역설정은 Daum Local API를 이용하여 검색 후 반영한다. (위도, 경도, 주소 명칭을 알 수 있음) 날씨정보는 Openweathermap API를 이용하여 파악한다.

요구사항 ID	상세	요구사항 설명	수용여부	미수용 사유	반영 방법
3	노래듣기	사용자가 원하는 노래를 스마트거울을 통해 들을 수 있다.	Y		MP3 파일을 이용하여 서버의 용량을 부담시키지 않고, Soundcloud API를 이용하여 스트리밍 음악을 재생한다.
4	메모작성과 확인	사용자가 스마트거울 사용자 안드로이드 어플리케이션을 이용하여 원하는 날짜에 메모를 작성하면 해당날짜에 메모를 확인할 수 있다.	Y		서버와 데이터베이스를 이용하여 반영한다.
5	수면패턴 확인	수면패턴을 분석하여 사용자의 수면정보를 스마트거울로 확인할 수 있다.	N	수면패턴을 분석하기 위한 알고리즘 구현이 현실적으로 불가능하다고 판단함.	

요구사항 ID	상세	요구사항 설명	수용여부	미수용 사유	반영 방법
6	거울의 개인화	모든 사용자는 자신만의 정보를 노출시킬 수 있는 스마트 거울 기기를 가질 수 있어야 한다.	Y		회원가입 시스템을 구축하여 정보를 개인화 한다. 서버와 데이터베이스를 이용하여 반영한다.
7	원웨이 미러 (취조실 거울)	사용자는 스마트 거울 앞에서 자신의 모습과 동시에 스마트 거울 기기 안드로이드 어플리케이션이 출력하는 정보를 확인할 수 있다.	Y		유리, 원웨이 미러 필름, 모니터와 안드로이드 OS 기기를 이용하여 반영한다.
8	터치 제어	사용자가 스마트 거울을 직접 터치하여 기능을 제어할 수 있다.	N	모니터 이상의 크기를 가진 터치 가능한 부품이 현실적으로 구매 불가하다고 판단함.	
9	모션인식 제어	사용자가 직접적인 제어(터치, 버튼 등) 없이 모션인식으로 스마트 거울 기능을 제어할 수 있다.	N	모션인식을 위한 알고리즘 구현이 힘들다고 판단함.	

요구사항 ID	상세	요구사항 설명	수용여부	미수용 사유	반영 방법
10	음성인식 제어	사용자가 직접적인 제어(터치, 버튼 등) 없이 음성인식으로 스마트 거울 기능을 제어할 수 있다.	Y		구글 TTS 라이브러리를 이용하여 반영
11	정보 디스플레이	스마트 거울에서 불필요한 디스플레이를 없애기 위하여 사용자가 거울 앞에 위치했을 경우에만 정보를 디스플레이 한다.	Y		안드로이드 OpenCV 라이브러리를 이용하여 사용자 얼굴을 탐지한 후 반영한다.
12	음성 인식 제어 진입	음성인식을 시작하기 위하여 사용자 얼굴의 윙크를 인식한 후 작동하는 기능.	Y		음성인식(구글 TTS 라이브러리)은 시작과 종료의 있기 때문에, 시작을 알리는 것이 필요하다. 안드로이드 OpenCV 라이브러리를 이용하여 한쪽 눈 인식(윙크) 알고리즘을 설계하여 반영한다.

2.1.2 요구사항 파악을 토대로 목표설정

- 시스템 구성 목표

1. 사용자 안드로이드 어플리케이션

- > 스마트 거울을 개인화(메모, 날씨, 노래)하기 때문에 사용자가 정보를 작성하고 수정할 수 있는 어플리케이션 필요.
스마트폰 종류 3가지(ios, Window, Android) 중 무료이며 쉽게 개발이 가능한 Android 어플리케이션 개발로 선택

2. 서버와 데이터 베이스

- > 스마트 거울 사용자 안드로이드 어플리케이션과 스마트 거울 기기 안드로이드 어플리케이션과 통신을 하기 위한 웹 서버와 정보를 저장하고 동기화 할 수 있는 데이터베이스가 필요.

3. 스마트 거울 기기 안드로이드 어플리케이션

- > 사용자 안드로이드 어플리케이션에서 설정한 정보를 서버와 통신하여 정보를 출력해 줄 수 있는 어플리케이션이 필요.
파이썬, 자바스크립트 등 웹 어플리케이션 보다 안드로이드 어플리케이션 개발에 경험이 많기 때문에 안드로이드 어플리케이션으로 선택.

4. 스마트 거울 기기

- > 사용자는 스마트 거울을 보았을 때 자신의 모습과 스마트 거울 기기 안드로이드 어플리케이션이 출력하는 정보를 확인할 수 있게 제작한다.
원웨이 필름(취조실 거울처럼)과 유리, 모니터를 이용하여 제작한다. 모니터에 출력, 서버와의 통신, 사운드 출력, 음성을 입력, 영상을 입력받을 수 있는 카메라와 연결, 안드로이드 OS 실행 이 모든 것이 가능한 HW를 선택한다.

- 시스템 별 기능 구현 목표

1. 사용자 안드로이드 어플리케이션

- 회원 시스템 : 회원 가입, 회원 로그인, 로그아웃 기능을 구현한다.
- 위치 설정 : 동, 읍, 면 단위의 위치 검색을 한 후 위도, 경도, 주소 이름 정보를 저장한다. (Daum Local API 연동)
- 메모 : 날짜순의 등록했던 메모 리스트를 화면에 표시한다. 지난 날짜에 대한 정보는 화면에 출력하지 않도록 구현한다.
원하는 메모 내용, 날짜를 작성할 수 있도록 구현한다.

- 노래 : 등록했던 노래 리스트를 화면에 표시한다.
 SoundCloud 공유 Url을 이용하여 원하는 제목의 노래를 등록할 수 있도록 구현한다.
 등록된 노래를 확인할 수 있도록 미리듣기 기능을 구현한다.
 (SoundCloud API 연동 -> 음악 링크 이미지, 스트리밍 음악 Url 제공)
- 스마트 거울 기기 등록 : 스마트 거울 기기의 시리얼 번호를 이용하여 등록할 수 있는 기능을 구현한다.

2. 스마트 거울 기기 어플리케이션

- 날짜 정보 출력 : 스마트 거울 기기 안드로이드 자체의 날짜 정보
 (연도, 월, 일, 요일)를 받아 화면으로 출력하는 기능을 구현한다.
- 날씨 정보 출력 : 사용자가 사용자 어플리케이션으로 설정한 위치의 이름을
 출력하는 기능을 구현한다.
 해당 날짜의 날씨를 그림으로 출력하는 기능을 구현한다.
 (맑음, 구름 조금, 구름 많음, 비, 눈, 번개, 안개)
 해당 날짜의 날씨를 온도로 출력하는 기능을 구현한다.
 -> Openweathermap API를 연동하여 정보를 얻는다.
- 메모 리스트 출력 : 사용자가 사용자 어플리케이션으로 등록한 메모정보를
 해당날짜에 맞는 것만 화면에 출력하는 기능을 구현한다.
- 화면 출력 : 얼굴 탐지를 이용하여 탐지가 되었을 경우 일정 시간동안 정보화면을
 출력하는 기능을 구현한다.
 -> 안드로이드 OpenCV 라이브러리를 이용하여 얼굴을 탐지한다.
- 음성 인식 : 음성인식기능을 이용하여 스마트 거울 기기 어플리케이션에 동작할
 기능을 제어할 수 있게 구현한다.
 -> 구글 TTS 라이브러리를 이용하여 음성인식 구현
- 음성 인식 기능 진입 : 구글 TTS 라이브러리 특성 상 시작과 끝이 존재하기 때문에
 음성인식의 시작을 알리는 부분이 필요하다.
 안드로이드 OpenCV 라이브러리를 이용하여 눈을 인식하고
 한쪽눈을 탐지하는 알고리즘을 구현하여 음성 인식 진입을
 할 수 있는 과정을 구현한다.

- 노래 듣기 : 음성 인식 기능으로 사용자가 사용자 안드로이드 어플리케이션으로 등록된 노래를 재생할 수 있는 알고리즘을 구현한다.
사용자가 연동한 노래 리스트 정보를 서버에서 수신하여 스트리밍 Url로 노래를 재생할 수 있게 구현한다.

3. 서버와 데이터 베이스

- 서버 구축 : 365일 24시간 돌아가야 한다는 조건을 살펴보았을 때 추가적인 PC를 보유하고 있지 않으며, 학교 인터넷 보안 특성 상 같은 Wifi 연결에만 접속 가능한 제한이 있다. 이를 해결하기 위해 닷홈 (<http://www.dothome.co.kr/>)에서 웹 호스팅을 사용하여 서버를 구축한다.
- 데이터베이스 구축 : 사용자 안드로이드 어플리케이션, 스마트 거울 기기 안드로이드 어플리케이션 기능에 필요한 요구사항을 파악한 후 데이터베이스 테이블을 구축한다.
-> MySQL 사용
- API 구현 : Middle Ware인 PHP로 안드로이드 어플리케이션과 데이터베이스 통신에 필요한 API를 개발한다.
- 정보의 암호화 : 스마트 거울을 사용하는 사용자의 정보를 암호화 하여 보호하도록 구현한다. 사용자의 비밀번호는 관리자도 못 알아봐야 한다.
이를 위해 단방향 해쉬 알고리즘인 SHA-256을 이용하여 암호화 한다.

4. 요구사항 파악 후 추가적인 목표

- 알림 연동기능 구현 : 사용자 안드로이드 어플리케이션에서 알림연동을 할 수 있게 구현한다. 사용자가 로그인한 스마트폰의 알림 (카카오톡, MMS, Facebook알림)을 스마트 거울 기기 안드로이드 어플리케이션에서 확인할 수 있게 구현한다.
- 음성인식 설명 기능 : 사용자가 스마트 거울의 음성인식 기능을 사용할 수 있게끔 음성인식 설명서를 화면에 출력할 수 있게 구현한다.
- 페이스북 계정 연동 : 사용자가 회원가입과 로그인을 편리하게 하기 위해 페이스북 API를 이용하여 계정을 연동하는 기능을 구현한다.

2.2 분석

- 데이터 베이스 요구사항 분석

- 회원가입
 - 사용자는 안드로이드 어플리케이션을 이용하여 회원가입을 할 수 있다.
 - 사용자의 email, 사용자 이름, 비밀번호, 위치정보(위도, 경도, 주소)는 회원가입을 위한 필수적인 요소이다.
 - 사용자의 'email'은 이미 회원가입 되어 있는 사용자의 'email'과 중복되지 않는다.
 - 사용자는 Facebook 계정으로 회원가입을 할 수 있다.
 - Facebook 계정으로 스마트 미러 서비스의 회원가입을 수행하려면 Facebook 계정의 email 정보 제공 동의를 하여야 한다.
- 로그인
 - 사용자는 회원가입 되어 있는 'email'과 '비밀번호'로 로그인 기능을 수행할 수 있다.
 - 사용자는 사용자가 입력한 'email'과 비밀번호가 회원가입 되어 있는 사용자의 'email'과 '비밀번호'가 일치할 경우 로그인에 성공한다.
 - Facebook 계정으로 회원 가입한 사용자는 Facebook 계정으로 로그인에 성공할 경우, 스마트 미러 서비스의 로그인 기능을 수행할 수 있다.
- 메모
 - 사용자는 로그인에 성공한 계정으로 메모를 작성할 수 있다.
 - 메모의 개수는 없을 수 있으며, 여러 개일 수 있다.
 - 메모의 정보는 내용, 작성날짜 가 있으며 메모정보의 '내용'은 50자를 넘지 못한다.
 - 사용자는 로그인에 성공한 계정으로 작성한 기존의 메모를 삭제, 수정할 수 있다.
- 노래
 - 사용자는 로그인에 성공한 계정으로 노래 리스트를 작성할 수 있다.
 - 노래의 개수는 없을 수 있으며, 여러 개일 수 있다.
 - 노래의 정보는 제목, soundcloud 링크 url, 작성날짜 가 있으며 '제목'은 40자를 넘지 못한다.
 - 노래의 정보 중 '제목', 'soundcloud 링크 url'의 내용은 없을 수 없다.
- 비밀번호 변경
 - 사용자는 로그인에 성공하여 회원 가입한 계정의 비밀번호를 수정할 수 있다.
 - 사용자는 회원 가입한 계정의 비밀번호를 수정하기 위해 현재의 비밀번호를 입력하고, 그 비밀번호 정보가 일치해야 한다.
- 위치 변경
 - 사용자는 로그인에 성공한 계정으로 위치를 변경할 수 있다.

- 사용자는 동명(읍, 면)을 입력하여 위치를 검색할 수 있고 그 위치를 설정할 수 있다.
- 스마트 거울 기기 설정
 - 사용자는 로그인에 성공한 계정으로 스마트 거울 기기를 설정할 수 있다.
 - 사용자는 로그인에 성공한 계정 하나 당 한 개의 스마트 거울 기기를 설정할 수 있다.
 - 사용자는 스마트 거울 기기의 일련번호를 입력하여 설정한다.

- API 분석

- Daum Local API (위치 검색)

요청 방법	기능	요청 주소
GET	주소→좌표 변환	/local/geo/addr2coord
GET	좌표→주소 변환	/local/geo/coord2addr
GET	좌표계 변환	/local/geo/transcoord
GET	좌표→상세주소 변환	/local/geo/coord2detailaddr
GET	키워드로 장소검색	/local/v1/search/keyword.{format}
GET	카테고리로 장소 검색	/local/v1/search/category.{format}

[그림 1. 요청 방법]

- 주소를 좌표로 변환하는 <https://apis.daum.net/local/geo/addr2coord> 사용

요청 변수 (Request Parameters)				
파라미터	타입	설명	값의 범위	예제
q*	string	검색을 원하는 질의어		
pageno	integer	검색 결과의 페이지 번호		
page_size	integer	페이지당 결과건수	기본값 10 최소값 1 최대값 30	
output	string	포맷		json, xml

[그림 2. 요청 변수]

- q(검색을 원하는 질의어)에 사용자가 검색할 읍, 면, 동의 이름값을 전달하여 위치 정보를 받는다.
- output(포맷)을 json으로 고정하여 json 포맷의 데이터를 받아 파싱한다.

출력 결과 (Response Element)

출력 변수	값	설명
title	string	검색 제목
description	string	검색 결과의 간략한 소개
totalCount	integer	전체 검색 결과의 수
result	integer	한 페이지에 보여질 결과 수
q	string	검색어
item	개별 검색 결과 정보	
title	string	전체 주소
mountain	string	산
localName_1	string	지역명1
localName_2	string	지역명2
localName_3	string	지역명3
mainAddress	string	주 번지
subAddress	string	부 번지
buildingAddress	string	건물 번지
isNewAddress	string	신주소 여부
newAddress	string	신주소
lng	string	경도
lat	string	위도

[그림 3. 출력 결과]

- [그림 3]에 표시된 출력 변수 중 totalCount(전체 검색 결과의 수), title(전체 주소), localName_1(지역명1), localName_2(지역명2), localName_3(지역명3), lng(경도), lat(위도)을 안드로이드 어플리케이션과 서버에서 JSON 파싱을 이용하여 정보를 저장한다.

- OpenWeatherMap API (날씨 정보)

The screenshot shows the OpenWeatherMap API website. The header includes the OpenWeatherMap logo and navigation links: Weather, Maps, API, Price, Partners, Stations, News, and About. The main heading is "Weather API". Below this, a paragraph states: "Our weather API is simple, clear and free. We also offer higher levels of support, please see our **paid plan options**. To access the API you need to sign up for an **API key** if you are on a free or paid plan."

The website features six main API categories, each with a list of features and links to "API doc" and "Subscribe":

- Current weather data** (highlighted with a red border):
 - Access current weather data for any location including over 200,000 cities
 - Current weather is frequently updated based on global models and data from more than 40,000 weather stations
 - Data is available in JSON, XML, or HTML format
 - Available for Free and all other paid accounts
- 5 day / 3 hour forecast**:
 - 5 day forecast is available at any location or city
 - 5 day forecast includes weather data every 3 hours
 - Forecast is available in JSON, XML, or HTML format
 - Available for Free and all other paid accounts
- 16 day / daily forecast**:
 - 16 day forecast is available at any location or city
 - 16 day forecasts includes daily weather
 - Forecast is available in JSON, XML, or HTML format
 - Available for Developer, Professional and Enterprise accounts
- Historical data**:
 - Through our API we provide city historical weather data for 20,000+ cities
 - Historical data is available for 1 month previous in Starter account, for 1 year previous in Medium accounts, and is 5 and more year previous in Advanced accounts
- UV Index**:
 - Current UV index (Clear Sky) and historical data are available for any geo location (lat/lon)
 - Interpreting of the UV Index and recommended protection are provided
 - Data is available in JSON
 - Available for Professional and Enterprise accounts
- Weather map layers**:
 - Weather maps include precipitation, clouds, pressure, temperature, wind, and more
 - Connect our weather maps to your mobile applications and websites
 - Use as layers in Direct Tiles, OpenLayers, Leaflet, and Google Maps
 - Available for Free and all other paid accounts

[그림 4. OpenWeatherMap API 종류]

- 해당 날짜 하루의 자세한 날씨 정보를 취득하기 위해 API 종류 중 Current weather data를 사용한다.

By geographic coordinates

API call:

`api.openweathermap.org/data/2.5/weather?lat={lat}&lon={lon}`

Parameters:

lat, lon coordinates of the location of your interest

Examples of API calls:

`api.openweathermap.org/data/2.5/weather?lat=35&lon=139`

API respond:

```
{
  "coord": {
    "lon": 139,
    "lat": 35
  },
  "sys": {
    "country": "JP",
    "sunrise": 1369769524,
    "sunset": 1369821049
  },
  "weather": [
    {
      "id": 804,
      "main": "clouds",
      "description": "overcast clouds",
      "icon": "04n"
    }
  ],
  "main": {
    "temp": 289.5,
    "humidity": 89,
    "pressure": 1013,
    "temp_min": 287.04,
    "temp_max": 292.04
  },
  "wind": {
    "speed": 7.31,
    "deg": 187.002
  },
  "rain": {
    "3h": 0
  },
  "clouds": {
    "all": 92
  },
  "dt": 1369824698,
  "id": 1851632,
  "name": "Shuzenji",
  "cod": 200
}
```

[그림 5. 위도와 경도를 이용한 날씨 API 사용법]

- Daum Local API를 이용하여 위도와 경도를 알아냈기 때문에 By geographic coordinates API 요청을 사용한다.

Other features

Format

Description:

JSON format is used by default. To get data in XML or HTML formats just set up mode = xml or html.

Parameters:

mode - possible values are xml and html. If mode parameter is empty the format is JSON by default.

Examples of API calls:

JSON api.openweathermap.org/data/2.5/weather?q=London

XML api.openweathermap.org/data/2.5/weather?q=London&mode=xml

HTML api.openweathermap.org/data/2.5/weather?q=London&mode=html

[그림 6. 날씨 API 응답 형식]

- JSON, XML, HTML 의 응답 형식 중 JSON 포맷 응답 형식을 선택하여 요청한다.

Units format

Description:

Standard, metric, and imperial units are available.

Parameters:

units metric, imperial. When you do not use units parameter, format is Standard by default.

Temperature is available in Fahrenheit, Celsius and Kelvin units.

- For temperature in Fahrenheit use units=imperial
- For temperature in Celsius use units=metric
- Temperature in Kelvin is used by default, no need to use units parameter in API call

List of all API parameters with units openweathermap.org/weather-data

Examples of API calls:

standard api.openweathermap.org/data/2.5/find?q=London



















metric api.openweathermap.org/data/2.5/find?q=London&units=metric

imperial api.openweathermap.org/data/2.5/find?q=London&units=imperial

[그림 7. 온도 형식 응답 요청]

- Fahrenheit, Celsius, Kelvin 온도 형식 응답 방법 중 사용자가 읽기 쉬운 Celsius(섭씨)를 요청하여 응답받는다.

Icon list

Day icon	Night icon	Description
01d.png 	01n.png 	clear sky
02d.png 	02n.png 	few clouds
03d.png 	03n.png 	scattered clouds
04d.png 	04n.png 	broken clouds
09d.png 	09n.png 	shower rain
10d.png 	10n.png 	rain
11d.png 	11n.png 	thunderstorm
13d.png 	13n.png 	snow
50d.png 	50n.png 	mist

[그림 8. 날씨 별 아이콘 응답 표]

- 날씨에 따라 응답받는 아이콘을 분석하여 스마트 거울에 표시할 이미지를 정한다.


```
Formatted JSON Data
{
  "city": {
    "id": 1851632,
    "name": "Shuzenji",
    "coord": {
      "lon": 138.933334,
      "lat": 34.966671
    },
    "country": "JP",
    "population": 0,
    "sys": {
      "population": 0
    }
  },
  "cod": "200",
  "message": 0.0129,
  "cnt": 40,
  "list": [
    {
      "dt": 1465171200,
      "main": {
        "temp": 18.61,
        "temp_min": 18.61,
        "temp_max": 19.45,
        "pressure": 1020.48,
        "sea_level": 1029.46,
        "grnd_level": 1020.48,
        "humidity": 100,
        "temp_kf": -0.84
      },
      "weather": [
        {
          "id": 800,
          "main": "Clear",
          "description": "clear sky"
        }
      ]
    }
  ]
}
```


[그림 9. 날씨 API 실제 응답 값]

- 날씨 API 실제 응답값의 "dt" 속성이 대한민국 시간이 아니기 때문에 GMT+9를 이용하여 9시간 더한 값으로 시간을 계산하여 사용한다.

- SoundCloud API (음악 정보)


SoundCloud for Developers

Discover, connect and build



SoundCloud API

Build applications that allow users to upload and share sounds across the web




SoundCloud SDK

The SDKs will make it easier to access the SoundCloud API on your framework of choice


Developers

- Your Apps
- Register a new app
- HTTP API Guide
- HTTP API Reference
- HTTP API Wrappers
- Widget API
- Rate Limits
- Support
- Backstage Blog




Embedded Player

Learn how to control the widget from the page the widget is inserted into




Branding kit

It's always nice to give people credit for their work. If you're building an app that uses data provided by SoundCloud and our community it's best to give attribution.



Twitter

Follow @SoundCloudDev

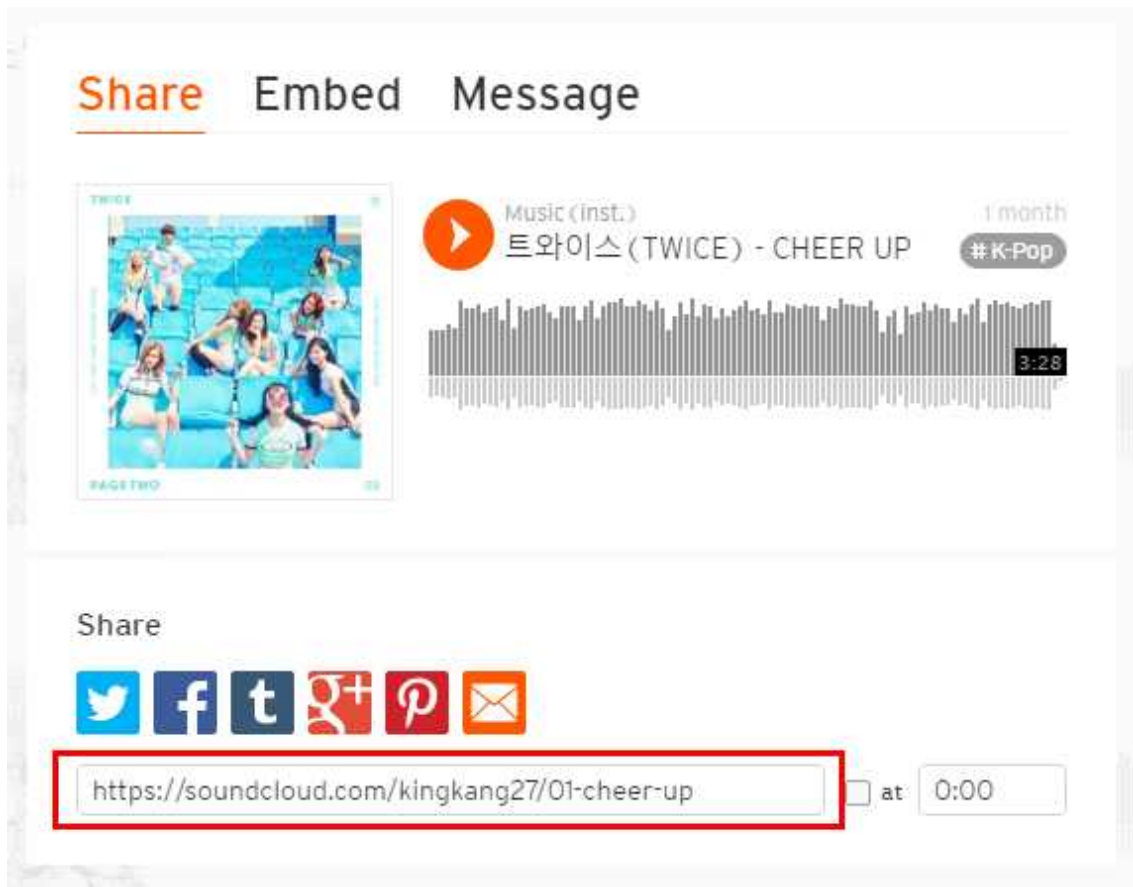


Backstage blog

Latest updates

[그림 10. SoundCloud API 종류]

- SoundCloud API 종류 중 [그림 10]에 나온 첫 번째 API를 사용한다.



[그림 11. SoundCloud 공유 링크]

- SoundCloud 노래 정보를 API에 요청하기 위해 공유링크를 이용한다.

/resolve

The resolve resource allows you to lookup and access API resources when you only know the SoundCloud.com URL.

Example

```
$ curl -v 'http://api.soundcloud.com/resolve?
url=http://soundcloud.com/matas/hobnotropic&client_id=YOUR_CLIENT_ID'

< HTTP/1.1 302 Moved Temporarily
< Location: http://api.soundcloud.com/tracks/49931
```

This request will resolve and redirect to the API resource URL for the track <http://soundcloud.com/matas/hobnotropic>. Just follow the redirect and you will get the representation you want. The resolver supports URLs for:

- users
- tracks
- playlists (sets)
- groups
- apps

[그림 12. SoundCloud /resolve API 요청 방법 및 응답 값]

- <http://api.soundcloud.com/resolve> API에 등록할 노래 SoundCloud 공유링크로 요청한다.
- 자세한 정보를 알기 위해 응답으로 받은 tracks값을 다시 /tracks API에 요청한다.

/tracks

A SoundCloud Track

Example

```
$ curl "http://api.soundcloud.com/tracks/13158665?client_id=YOUR_CLIENT_ID"
```

```
{
  "id": 13158665,
  "created_at": "2011/04/06 15:37:43 +0000",
  "user_id": 3699101,
  "duration": 18109,
  "commentable": true,
  "state": "finished",
```

Properties


name		description	example value
id	RO	integer ID	123
created_at	RO	timestamp of creation	"2009/08/13 18:30:10 +0000"
		user-id of the	

[그림 13. SoundCloud /tracks API 요청 방법 및 응답 값]

- <http://api.soundcloud.com/tracks> API로 요청한 응답 값을 선택하여 저장 후 사용한다.
- title(제목), artwork_url(음악 이미지), streamable(스트림 음악 가능 여부), stream_url(스트림 음악 Url) 응답 값을 전달 받은 후 데이터 베이스에 저장한다.

- Facebook API (페이스북 계정 연동)

대시보드



SmartMirror

이 앱은 개발 모드 상태이므로 앱 관리자, 개발자, 테스터만 사용할 수 있습니다. [?]

API 버전 [?]


앱 ID

v2.5

앱 시크릿 코드

.....

보기



SmartMirror 앱을 수익화하는 방법

타겟 네트워크를 사용하여 SmartMirror에 광고를 표시하고 3백만 광고 주로부터 수익을 얻어보세요.

시작하기

Facebook SDK 시작하기

빠른 시작 가이드를 사용하여 iOS 또는 Android 앱, 캔버스 게임 또는 웹사이트에 맞게 Facebook SDK를 설정하세요.

플랫폼 선택

Facebook 앱 분석

[그림 14. Facebook Developer에 스마트 거울 앱 등록 화면]

- 페이스북 계정 연동 기능을 위해 Facebook Developer에 스마트 거울 어플리케이션을 등록한 후 사용한다.

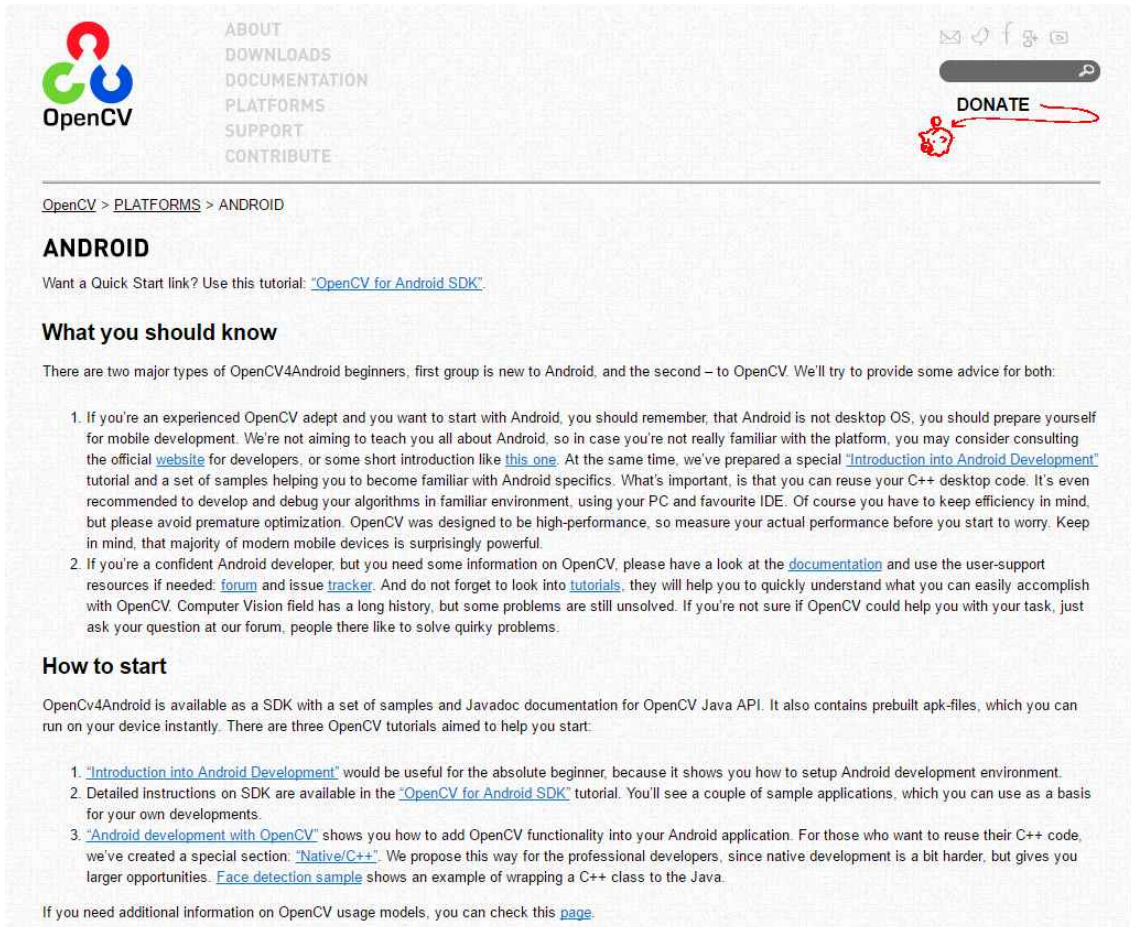
Field	설명
id numeric string	The id of this person's user account. This ID is unique to each app and cannot be used across different apps. Our upgrade guide provides more information about app-specific IDs 코어
email string	The person's primary email address listed on their profile. This field will not be returned if no valid email address is available 코어
name string	The person's full name 코어 기본

[그림 15. Facebook API를 이용하여 얻을 수 있는 값]

- 페이스북 API를 이용하여 사용자 스마트 거울 어플리케이션 회원정보에 필요한 id (페이스북 고유 id), email(페이스북 가입 시 입력한 email 주소), name (페이스북에서 사용하는 이름) 정보를 받아온다.

- 영상처리 분석

- Android OpenCV 라이브러리



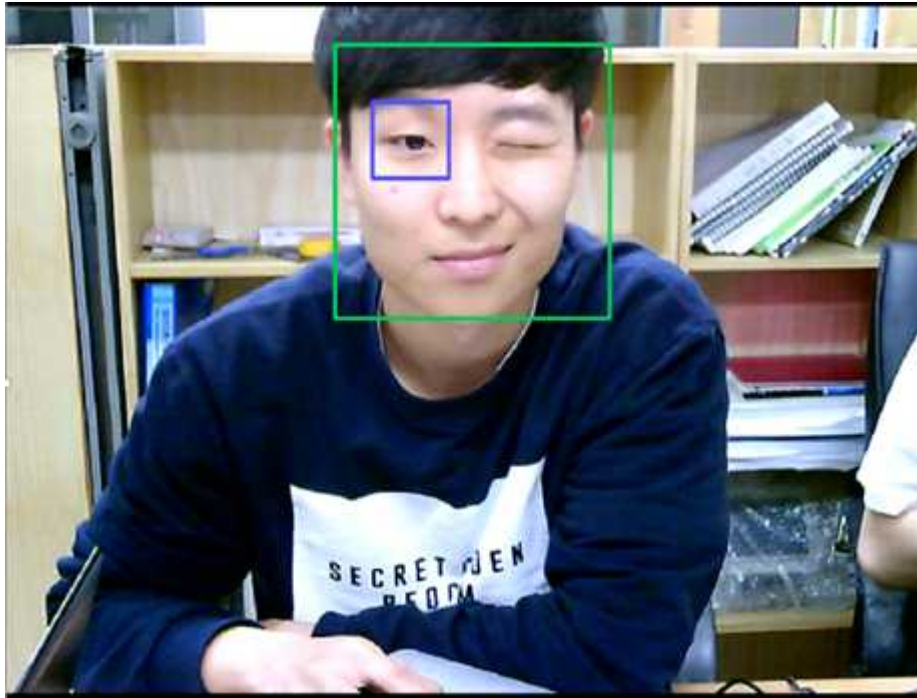
[그림 16. Android OpenCV 홈페이지]



[그림 17. Android에서 OpenCV 라이브러리 사용법]

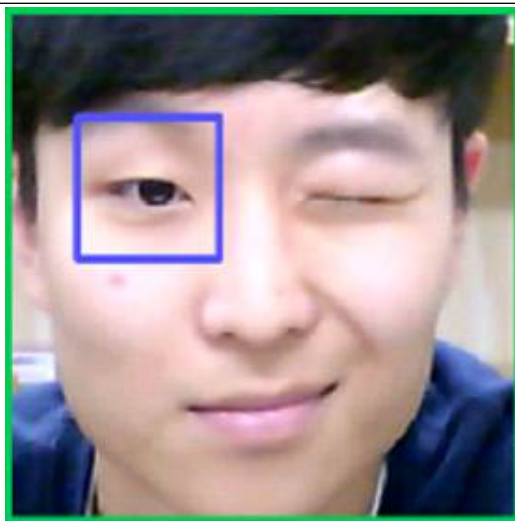
- Android OpenCV 홈페이지(<http://opencv.org/platforms/android.html>)에 기재되어 있는 사용법을 분석한다.
- [그림 17]의 How to start를 참고하여 OpenCV for Android SDK를 다운로드하여 사용한다.

- 얼굴인식, 눈 탐지 알고리즘 분석



[그림 18. 얼굴 인식, 눈 인식 알고리즘을 적용한 테스트 화면]

- 얼굴 인식 알고리즘은 lbpcascade_frontalface을 사용하여 탐지한다.
lbpcascade_frontalface 얼굴인식 알고리즘은 haarcascade_frontalface 얼굴인식 보다 빠르게 처리되며 가볍기 때문에 Mobile 기기에 적용하기에 적절한 알고리즘이라고 판단함.



[그림 19. 탐지된 얼굴 영역]



[그림 20. 얼굴 상단 3분의 2영역]

- 눈 탐지 알고리즘은 haarcascade_eye을 사용하여 탐지한다.
불필요한 영역의 눈 탐지를 제거하고 정확성을 높이기 위해 탐지된 얼굴 영역에서만 이 알고리즘을 실행하도록 적용한다.
사람의 눈 위치 특성상 얼굴 상단에 위치하고 있기 때문에 탐지된 얼굴 영역에서도 상단 3분의 2영역에서 적용시킨다.

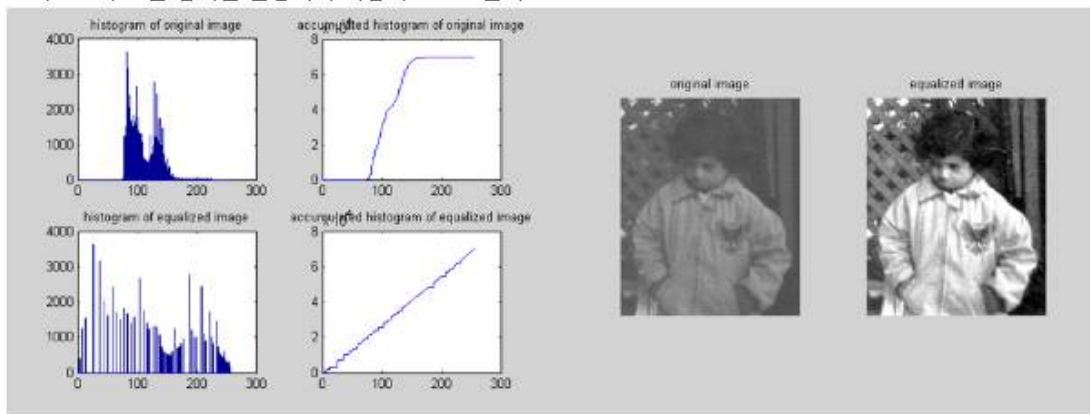
- 얼굴인식 영상 개선

히스토그램 균일화(Histogram Equalization)

- 히스토그램 확장(Histogram Stretching)의 일종.
- 영상의 통계 정보를 이용하는 histogram stretching
- 영상의 밝기값의 분포를 균일하게 만드는 영상 개선 작업
- 콘트라스트가 나쁜 영상의 개선에 사용(High Key/Low Key)

특징

- 히스토그램의 양 끝값을 누적분포(빈도)를 이용하여 비선형적으로 확장
- 계조도의 모든 영역을 균등하게 이용하므로 효율적



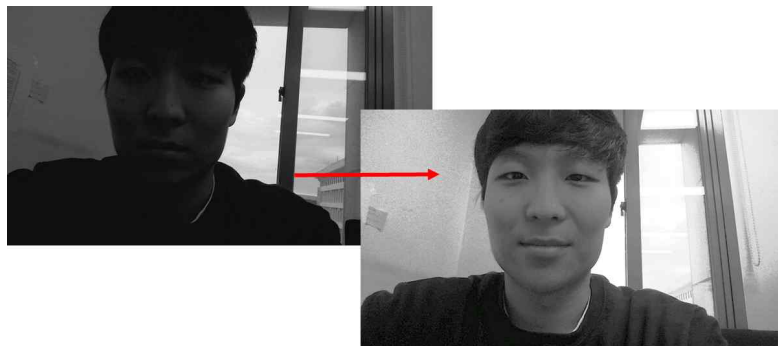
알고리즘

- 영상의 히스토그램을 구한다.
- 누적 히스토그램의 값을 정규화한다.
- 정규화된 누적 히스토그램을 사상함수(맵핑 함수)로 하여 밝기값을 0~Gmax까지 선형증가하도록 맵핑.

$$k_i = \frac{g_{\max}}{n_t} H(i)$$

[그림 21. Histogram Equalization 알고리즘]

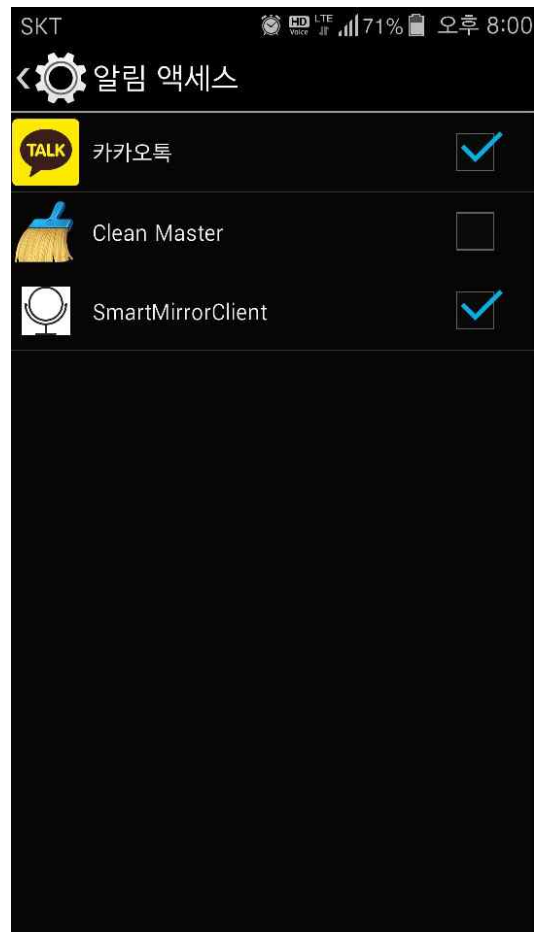
- 얼굴인식 알고리즘 lbpcascade_frontalface은 조명(명암) 변화에 매우 취약하기 때문에 영상의 밝기를 균일하게 만들어줄 Histogram Equalization 알고리즘을 분석하여 적용한다.



[그림 22. Histogram Equalization 알고리즘 적용 테스트 화면]

- 페이스북, MMS, 카카오톡 알림 연동 분석

- 안드로이드 스마트폰 알림 액세스 분석

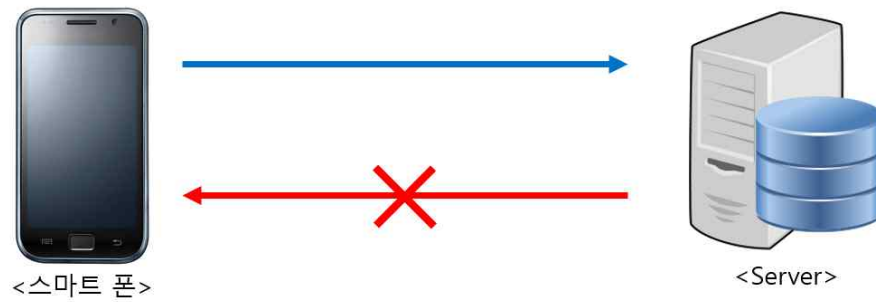


[그림 23. 스마트폰 알림에 접근하기 위한 권한 승인 화면]

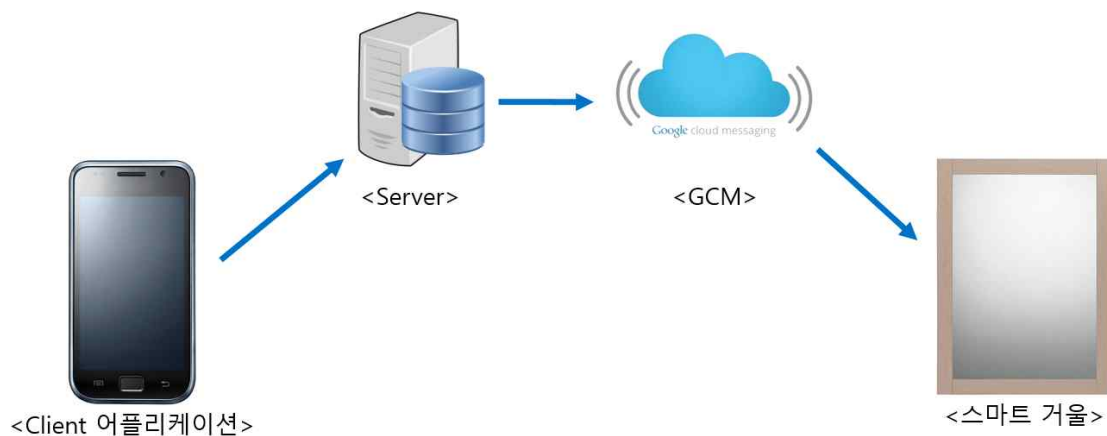
- 사용자 스마트폰으로 오는 알림을 알기 위해서 사용자 권한 승인이 필요함.
- 안드로이드 보안 정책상 피할 수 없는 과정.

- Google Cloud Messaging 서비스 분석

- 문제점 : 스마트폰은 서버에 요청을 할 수 있지만, 서버는 스마트폰으로 요청을 하지 못함 (서버는 1대, 스마트폰은 다수)



[그림 24. 알림 연동을 위한 문제점]



[그림 25. Google Cloud Messaging 서비스를 이용하여 문제 해결]

- GCM(Google Cloud Messaging 서비스)을 이용하면 원하는 안드로이드 스마트폰 기기에 데이터를 전송할 수 있다.
- 데이터에 알림의 종류(페이스북, MMS, 카카오톡)와 알림 수신/삭제 여부를 포함하여 스마트 거울 기기 안드로이드 어플리케이션에 전송하도록 한다.

- 비밀번호 암호화 분석

안전한 패스워드 저장

📅 2013.03.25 | 👤 77365

"보안 시스템은 가장 약한 연결 고리만큼만 강하다."

보안 시스템은 여러 부분으로 이뤄집니다. 공격자(attacker)는 이 중에서 가장 취약한 부분을 공격할 것이라고 가정해야 합니다. 보안 시스템이라는 사슬에서 가장 약한 고리가 끊어지면 다른 고리가 얼마나 강한지는 문제가 되지 않습니다. 즉, 보안 시스템의 안정성은 '강한 부분이 얼마나 강한가'보다는 '약한 부분이 얼마나 약한가'에 따라서 좌우됩니다.

지난해 6월 세계 최대 비즈니스 전문 소셜 네트워크 서비스(SNS) LinkedIn은 사용자데이터 해킹 사고로 650만 명의 아이디와 패스워드 정보가 유출된 후 집단 소송을 당했습니다. 취약한 암호화 알고리즘인 SHA-1을 사용했다는 것이 그 이유였습니다. 이제 보안 시스템의 한 부분인 암호화 알고리즘으로 어떤 알고리즘을 선택했는지도 보안의 책임을 다했는지 판단할 때 중요한 요소입니다.

이 글에서는 보안 시스템의 여러 부분 중, 패스워드를 저장할 때 사용되는 해시 함수(hash function)의 개념을 설명하고 대부분의 웹 사이트에서 사용하고 있는 암호화 알고리즘의 안정성을 검토하겠습니다. 그리고 어떤 암호화 알고리즘을 사용해야 안전한지 설명하겠습니다.

[그림 26. Naver D2의 암호화 알고리즘 선택의 중요성]

단방향 해시 함수

보통 프로그래머는 아래의 두 가지 중 한 가지로 사용자의 패스워드를 저장한다.

- 단순 텍스트(plain text)
- 단방향 해시 함수(one-way hash function)의 다이제스트(digest)

단순 텍스트로 패스워드를 저장하는 것은 범죄를 저지르는 것이나 다름없다. 아직도 이런 방법을 사용하고 있다면 지금 당장 변경해야 한다.

단방향 해시 함수는 수학적 연산을 통해 원본 메시지를 변환하여 암호화된 메시지인 다이제스트를 생성한다. 원본 메시지를 알면 암호화된 메시지를 구하기는 쉽지만 암호화된 메시지는 원본 메시지를 구할 수 없어야 하며 이를 '단방향성'이라고 한다.

예를 들어 사용자의 패스워드가 "hunter2"라면 이 문자열을 흔히 사용하는 해시 알고리즘인 SHA-256으로 인코딩하여 아래와 같은 값을 얻을 수 있다.

```
f52fbd32b2b3b86ff88ef6c490628285f482af15ddcb29541f94bcf526a3f6c7
```

위의 값을 저장하면 사용자의 패스워드를 직접 저장하는 위험을 피할 수 있다. 그리고 사용자가 로그인할 때 패스워드를 입력하면, 이를 해시한 값을 저장된 값과 비교하여 일치 여부를 확인할 수 있다.

대부분의 해시 함수는 입력 값의 일부가 변경되었을 때 다이제스트가 완전히 달라지도록 설계되어 있다. "hunter3"라는 값의 SHA-256 다이제스트는 아래와 같으며 위의 "hunter2"와는 완전히 달라진 것을 확인할 수 있다.

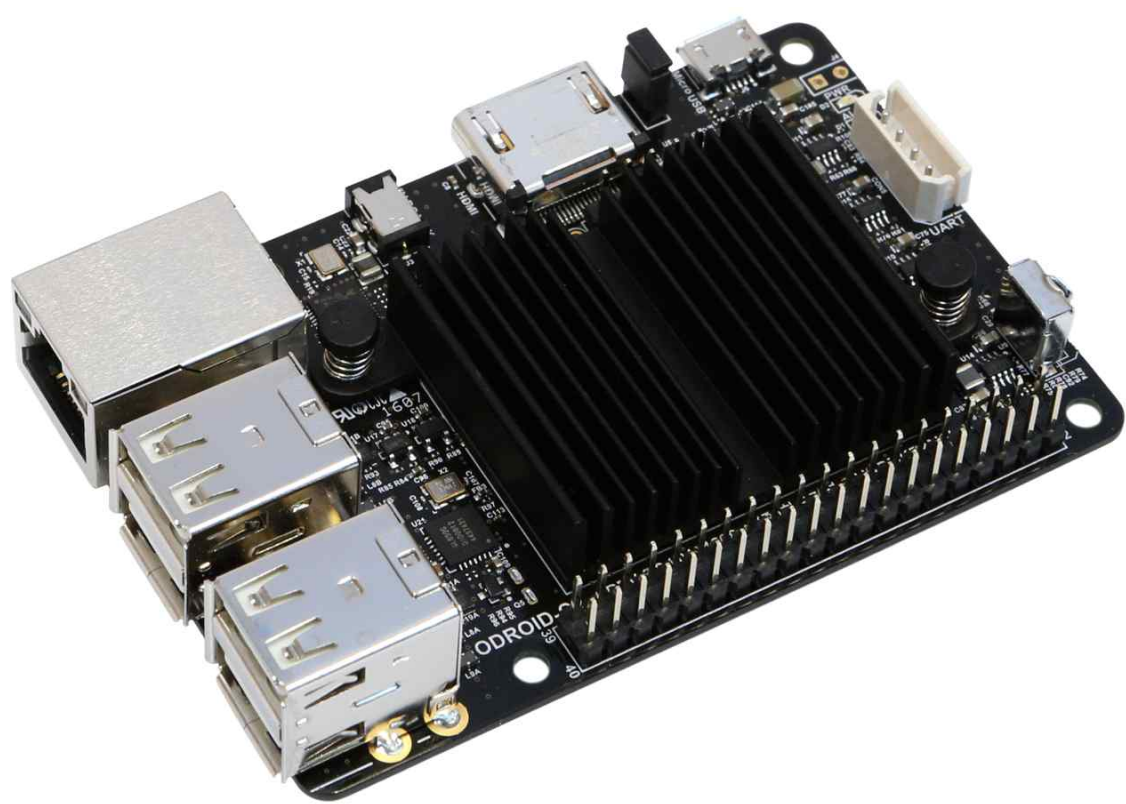
```
fb8c2e2b85ca81eb4350199fadd983cb26af3064614e737ea9f479621cfa57a
```

[그림 27. 단방향 해시 알고리즘인 SHA-256]

- '단순 텍스트로 패스워드를 저장하는 것은 범죄를 저지르는 것이나 다름없다. 아직도 이런 방법을 사용하고 있다면 지금 당장 변경해야 한다.' 라는 말이 적혀있는 것처럼 비밀번호를 그대로 저장하는 것에 있어서 문제점을 파악
- '단방향 해시 함수는 수학적 연산을 통해 원본 메시지를 변환하여 암호화된 메시지인 다이제스트를 생성한다. 원본 메시지를 알면 암호화된 메시지를 구하기는 쉽지만 암호화된 메시지는 원본 메시지를 구할 수 없어야 하며 이를 '단방향성'이라고 한다.' 라는 말처럼 원본 메시지를 구할 수 없는 알고리즘을 선택해야 함.

- 스마트 거울 기기 HW 스펙 분석

- ODROID-C2



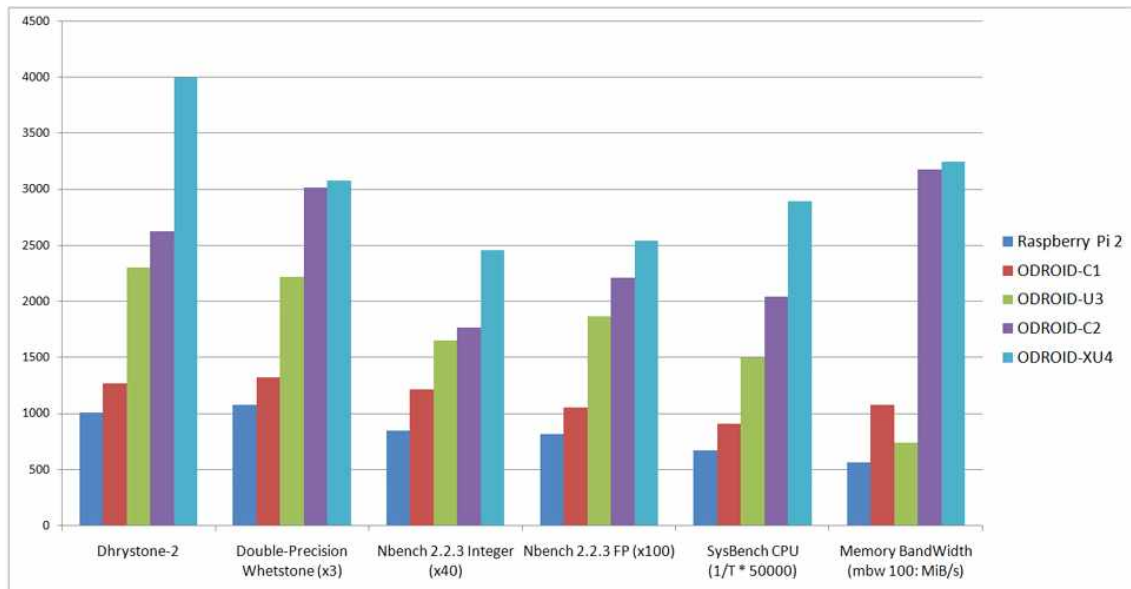
[ODROID-C2 모습]

SPECIFICATIONS

Processor	Amlogic S905 SoC ARM® Cortex®-A53 (ARMv8) 2.0GHz Quad Core ARMv8 architecture @28nm wafer
Memory	2Gbyte DDR3 SDRAM
3D Accelerator	ARM® Mali™-450 OpenGL ES 2.0 / 1.1 (3 x Pixel processors and 2 x Vertex shader processors)
Flash Storage	eMMC 5.0 Module Socket : 8~64GB eMMC module (option) MicroSD Card Slot : 8 ~128GB MicroSD UHS-1 (option)
USB2.0 Host	High speed standard A type connector x 4 ports
USB2.0 Device/OTG	High Speed USB standard A type connector x 1 port
Ethernet/LAN	10/100/1000Mbps Ethernet with RJ-45 Jack (Auto-MDIX support)
Video Output	HDMI2.0
Audio Output	HDMI / I2S
Camera Input	USB 720p(option)
IO Expansion	40pin port (GPIO / UART / I2C / ADC) 7pin port (I2S)
WiFi	USB IEEE 802.11b/g/n WLAN with Antenna (USB module) (option)
Power	5V 2A Power (option)
System Software	Ubuntu 16.04 on Kernel 3.14 Android 5.1.x on Kernel 3.14 Full source code is accessible via our Github.
Board Size	85 x 56 x 18 mm approx. (Weight: 40 gram w/o heat sink, 56 gram with heat sink) PCB Thickness : 1.0mm

[ODROID-C2 사양]

- ODROID-C2의 System Software인 Ubuntu 16.04 on Kernel 3.14 와 Android 5.1.x on Kernel 3.14 중 Android 5.1.x on Kernel 3.14를 선택하여 스마트 거울 기기 안드로이드 어플리케이션이 실행 가능하게 한다.



[Raspberry Pi 2 와 ODROID-C2의 성능 비교]

- Raspberry Pi 2 와 ODROID-C2의 성능을 비교 분석해보았을 때 ODROID-C2의 성능이 우월하다는 것을 알 수 있다. 안드로이드 OpenCV 라이브러리를 이용하여 영상처리 과정이 필요하기 때문에 성능이 좋을수록 수월하게 어플리케이션이 실행 될 수 있기 때문에 ODROID-C2를 선택.

- 원웨이 필름 분석



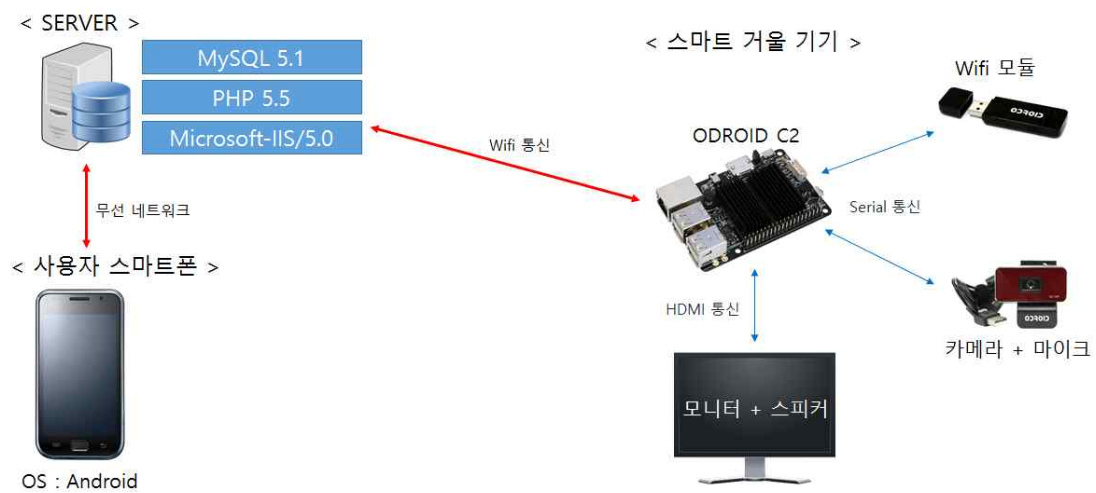
[원웨이 필름의 시공 예시]

- 원웨이 필름의 시공 예시를 분석하였을 때 실현가능성이 있다고 판단.

2.3 제작

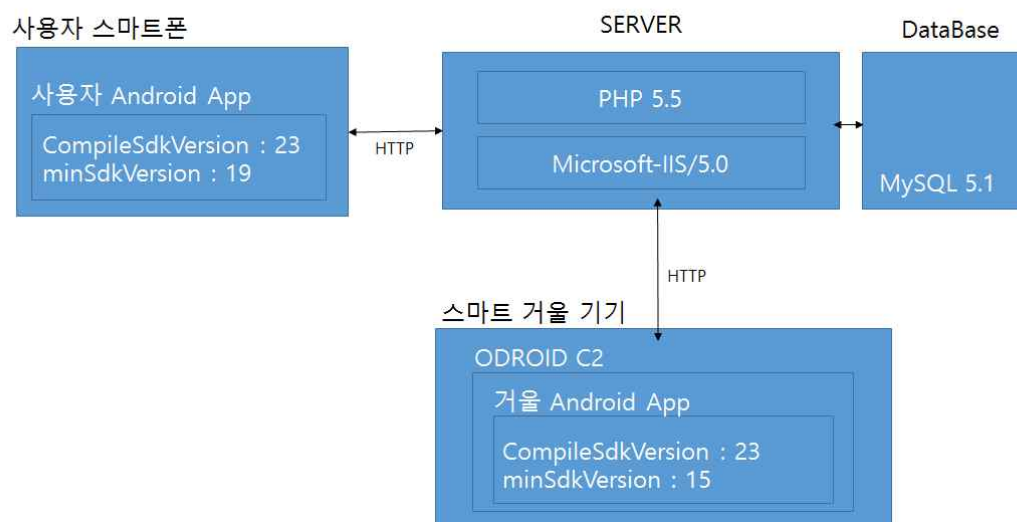
- 전체 시스템 구성도

시스템 구성도



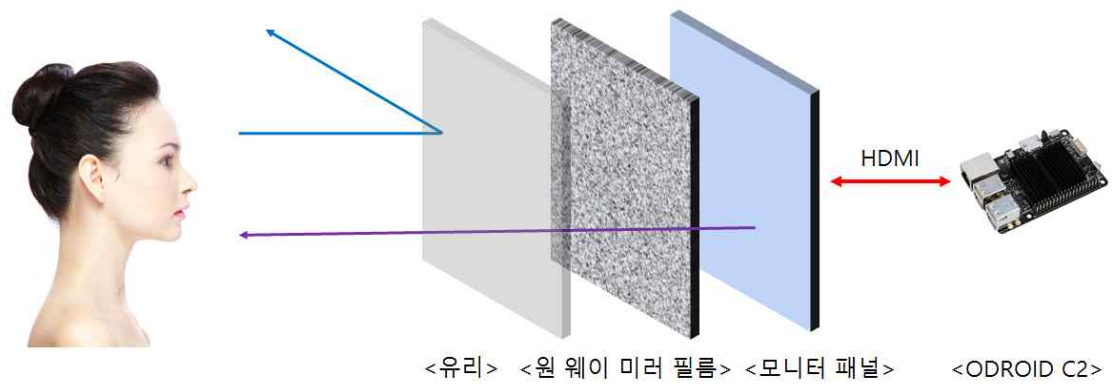
- Software 시스템 구성도

SW 구성도



- 스마트 거울 구성도

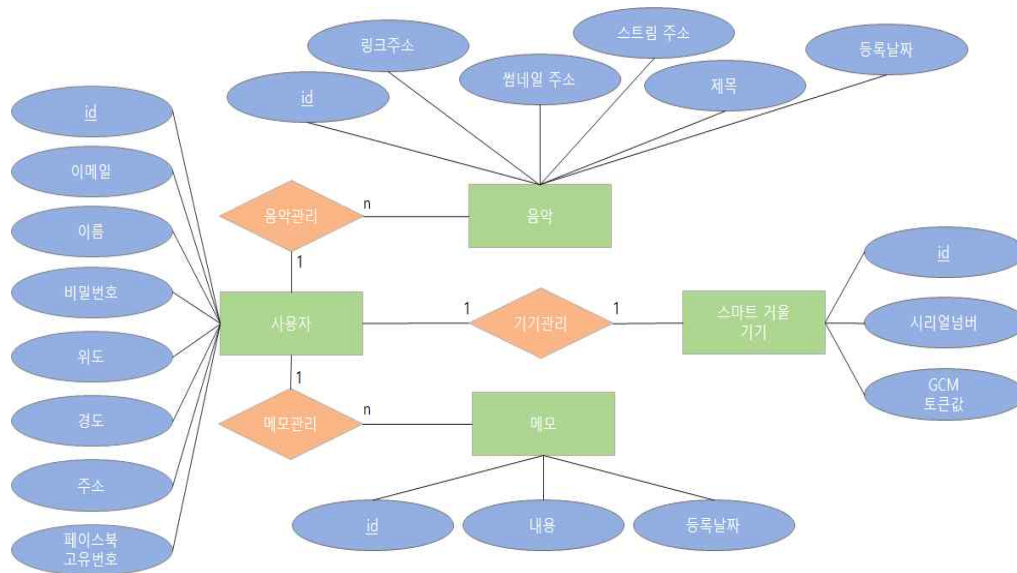
스마트 거울 구성도



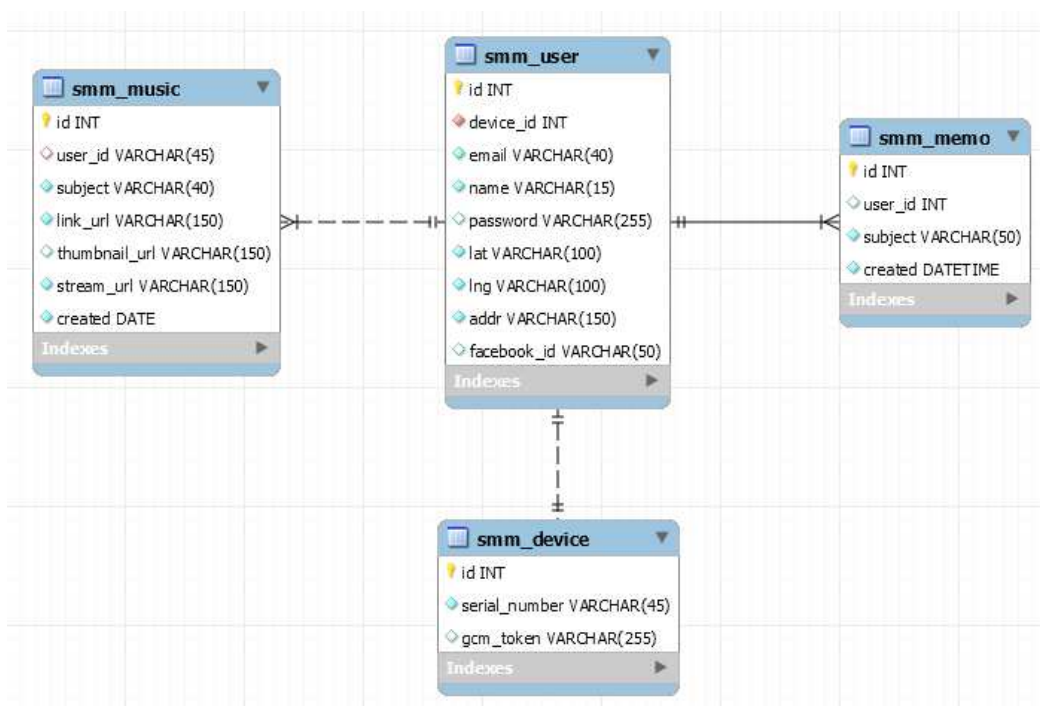
- 데이터 베이스 테이블 제작

-> 데이터 베이스 요구사항을 바탕으로 테이블을 설계

• E-R Diagram 설계



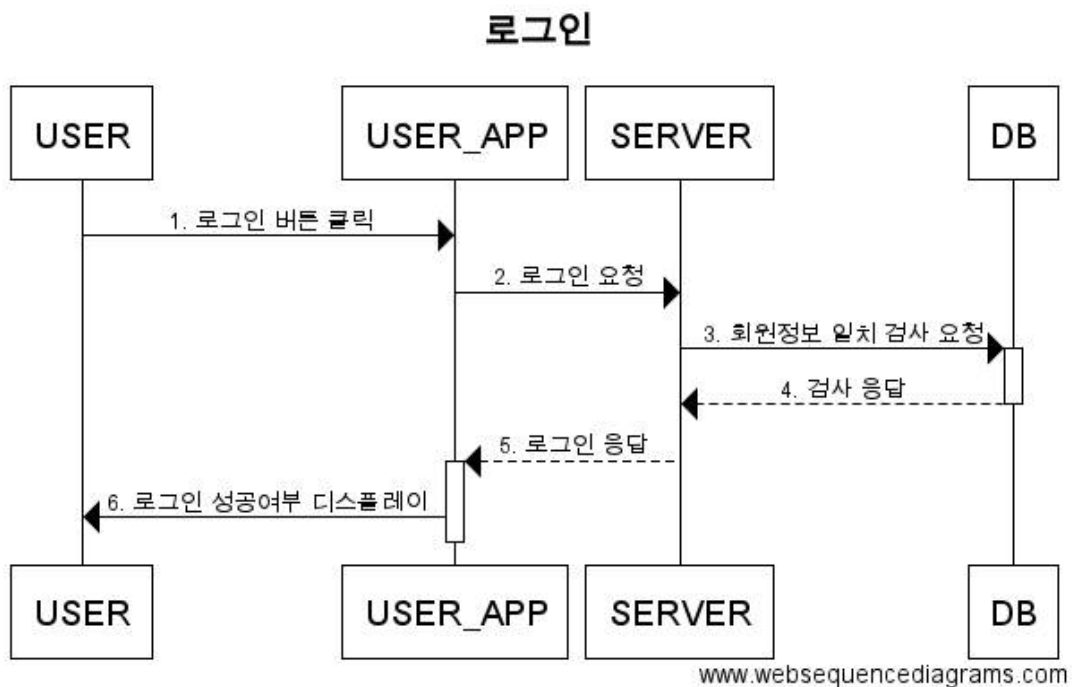
• 데이터 베이스 테이블 구현



- 안드로이드 어플리케이션 주요 기능 시퀀스 다이어그램 제작

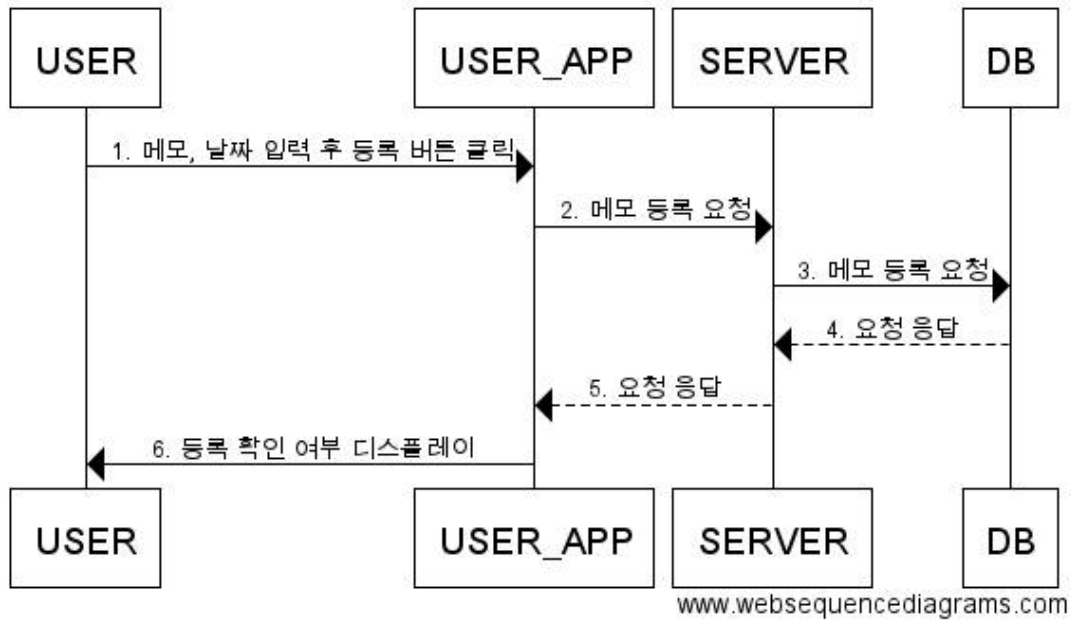
-> 주요 기능 별 시퀀스 다이어그램을 작성하여 스마트 거울 안드로이드 어플리케이션에 필요한 변수와 프로세싱 스케줄을 정확히 파악한다.

또한 안드로이드 어플리케이션, 서버, 데이터베이스 등 서로 통신하면서 필요한 데이터를 정의할 수 있다.



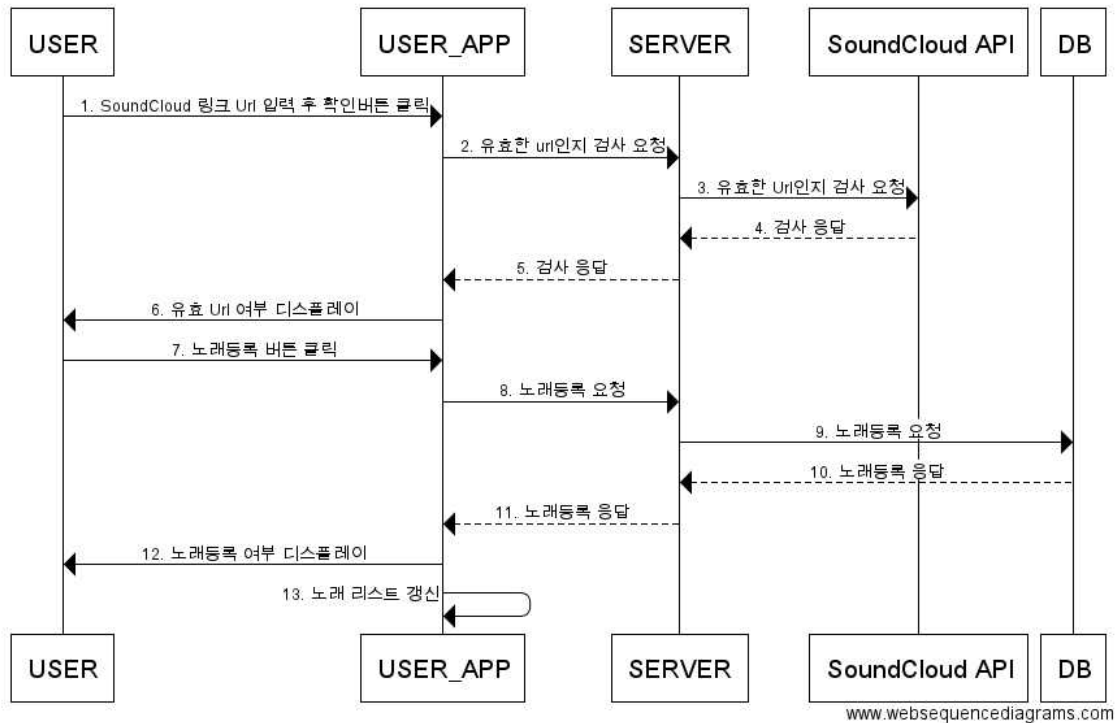
[로그인 시퀀스 다이어그램]

메모등록



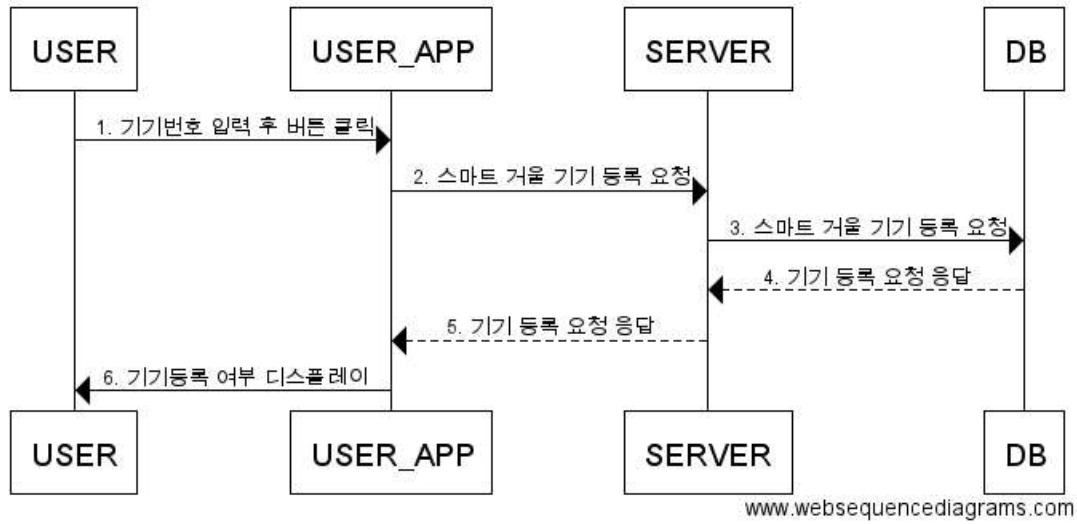
[메모등록 시퀀스 다이어그램]

노래등록



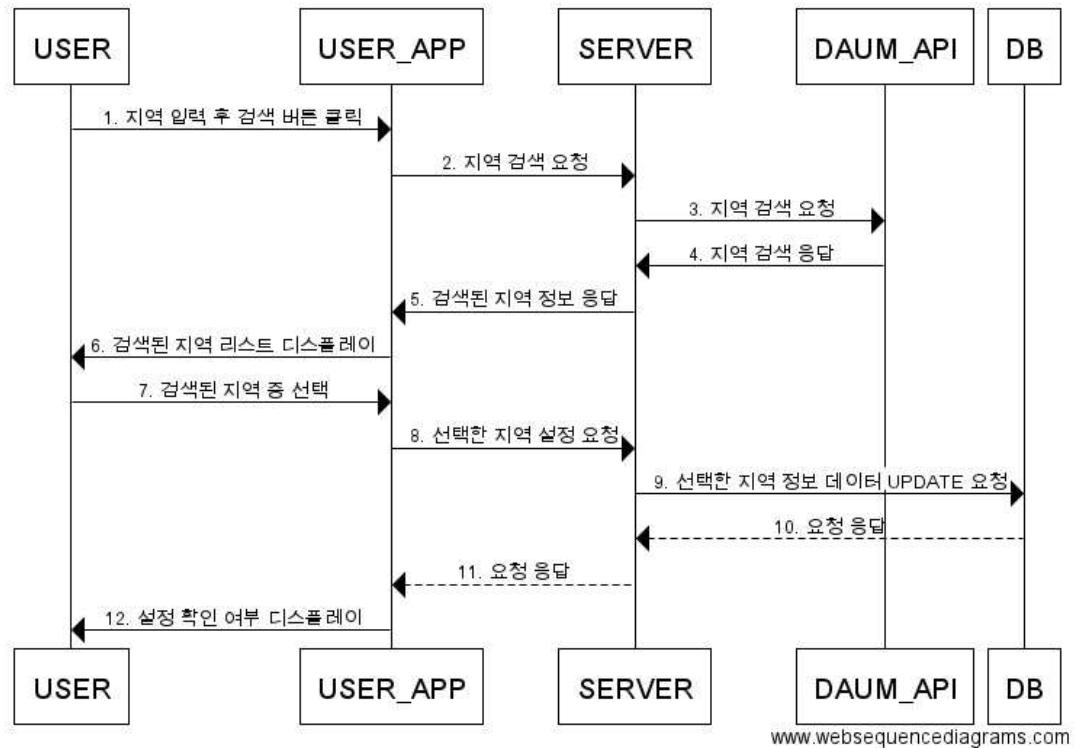
[노래등록 시퀀스 다이어그램]

스마트 거울 기기 등록

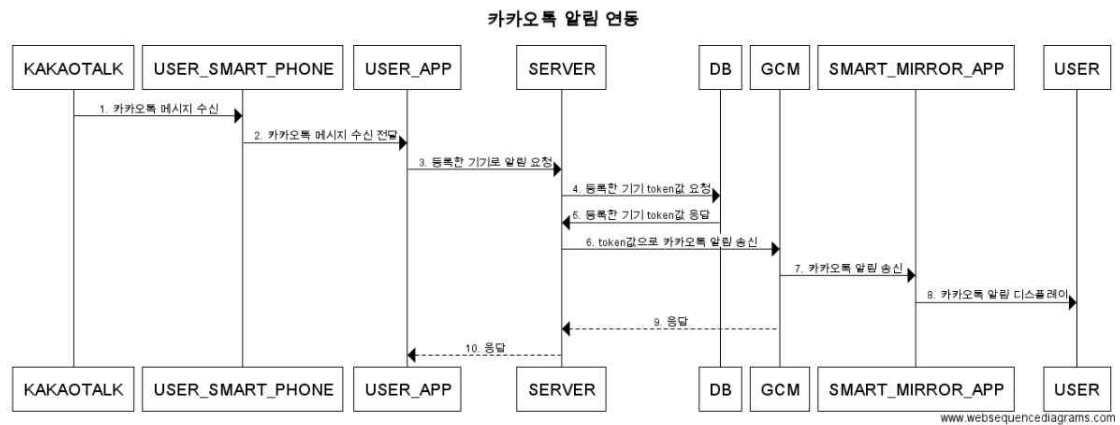


[스마트 거울 기기 등록 시퀀스 다이어그램]

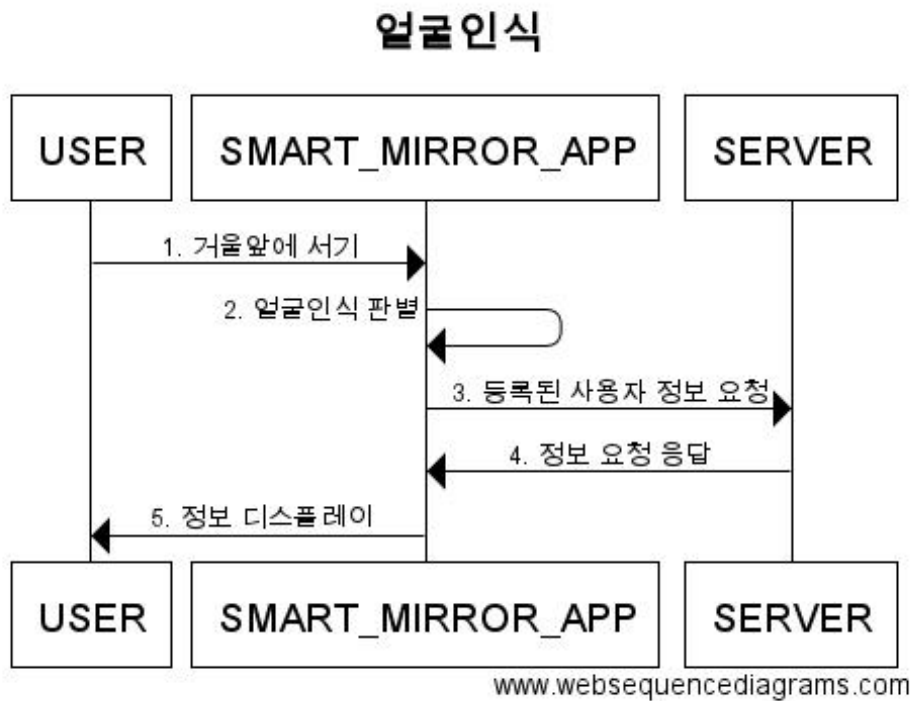
지역 설정



[지역 설정 시퀀스 다이어그램]

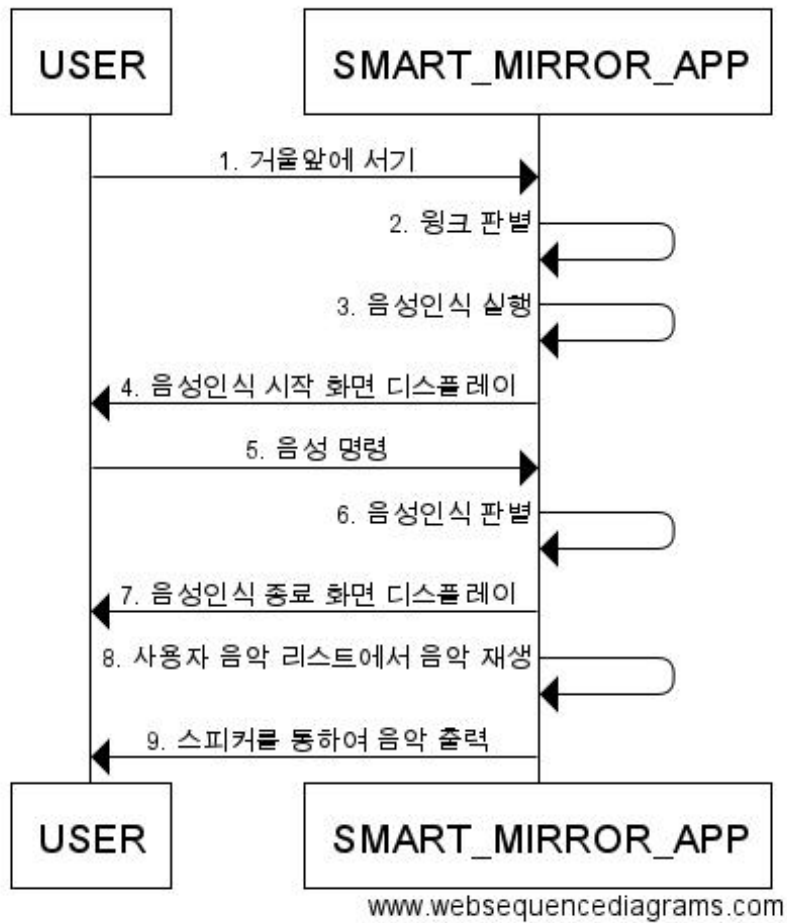


[카카오톡 알림 연동 시퀀스 다이어그램]



[얼굴인식 시퀀스 다이어그램]

음성명령 인식



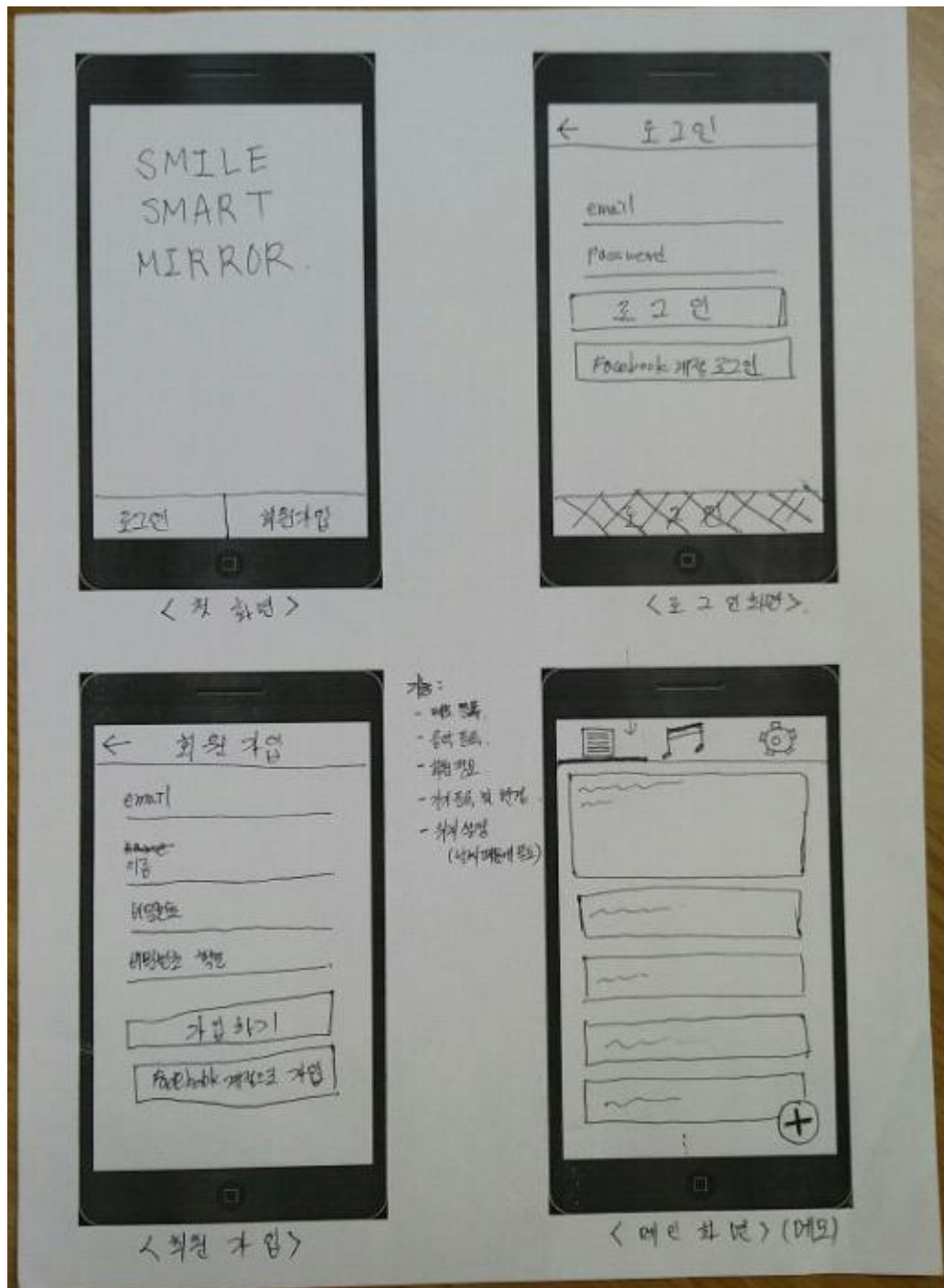
www.websequencediagrams.com

[음성명령 인식 시퀀스다이어그램]

- 스마트 거울 사용자 안드로이드 어플리케이션 제작

• UI 구상 및 도안 제작

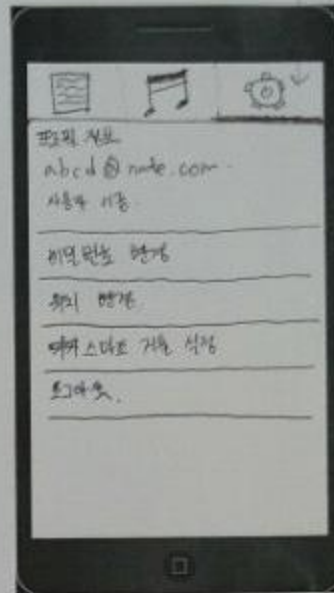
- 설정한 기능을 목표에 맞게 버튼과 화면을 구성한다.
- UI를 미리 구상 및 제작함으로써 누락된 기능이 없는지 확인할 수 있다.



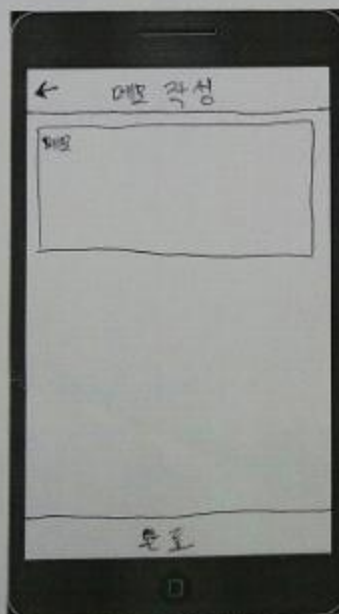


<메인 화면> (음악)

- 상단:
- 현재 화면
 - 이전 화면
 - 오디오
 - 오디오 설정

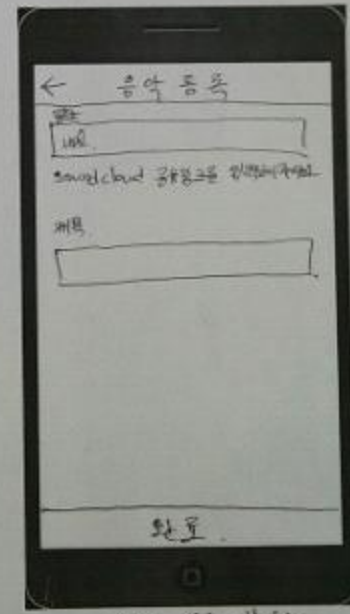


<메인 화면> (설정)



<메모 작성 화면>

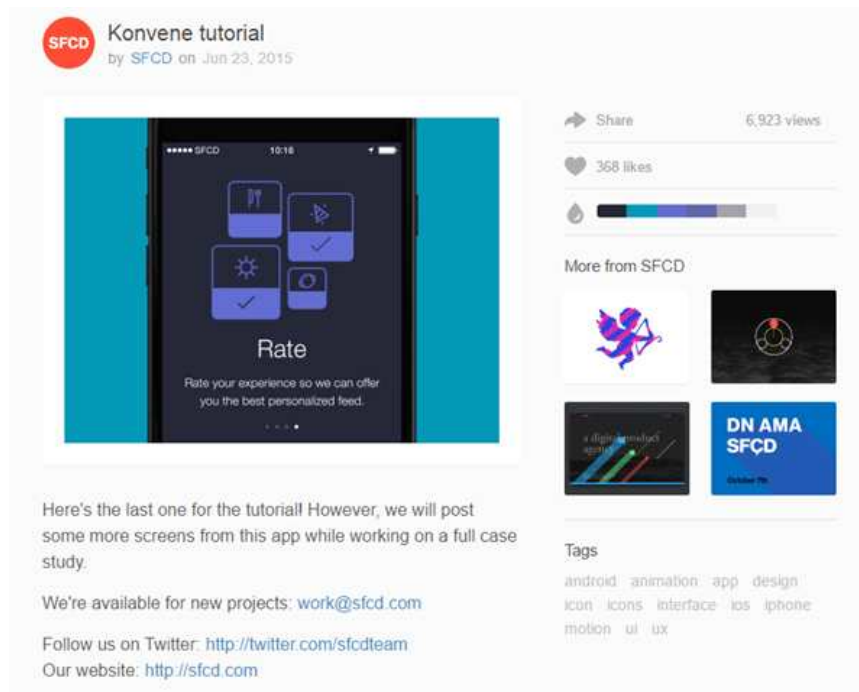
공격자 제한 (50)



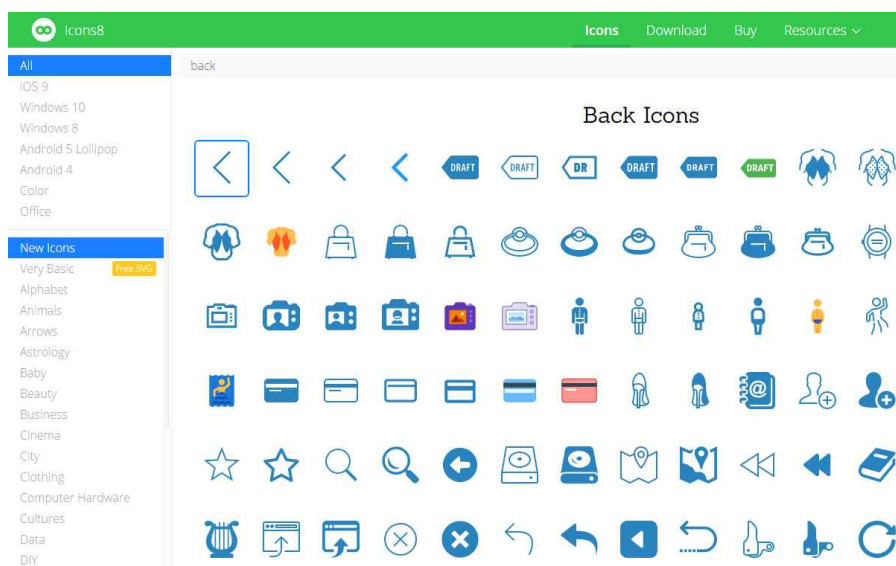
<음악 등록 화면>

- 테마 및 아이콘 설정

- 제작한 UI 구상도를 바탕으로 안드로이드 어플리케이션 테마를 제작한다.
- 웹, 어플리케이션 디자이너가 아니고 또 없기 때문에 안드로이드 어플리케이션 디자인 프로토타입 홈페이지를 적극 활용하여 참고한다.
- 안드로이드 어플리케이션에 사용할 아이콘 역시 무료 제공 홈페이지를 적극 활용한다.



[설정한 테마]



[무료 아이콘 다운로드 홈페이지]

- 안드로이드 어플리케이션 레이아웃 제작

	
<p>홈 화면</p>	<p>로그인 화면</p>
	
<p>회원가입 화면</p>	<p>메모 리스트 화면</p>

	
노래 리스트 화면	설정 화면
	
위치설정 화면	음악 등록 화면

 <p>메모 등록 화면</p>	 <p>기기 등록 화면</p>
--	---

- 스마트 거울 기기 안드로이드 어플리케이션 제작

- 안드로이드 어플리케이션 레이아웃 제작

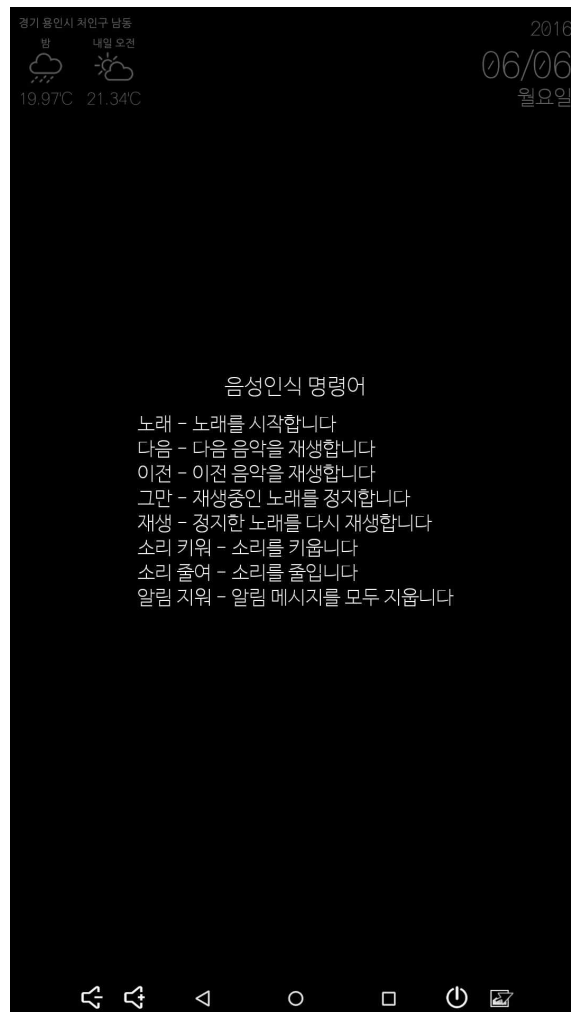


[스마트 거울 기기 메인 화면 레이아웃]



[스마트 거울 기기 음성인식 레이아웃]

- 상단에 위치, 날씨, 메모, 날짜 정보를 디스플레이 한다.
- 하단에 알림, 재생 중인 노래 정보를 디스플레이 한다.
- 검은색 배경으로 제작해야 원웨이 필름이 거울처럼 보일 수 있다.
- 링크를 인식하여 음성인식이 시작되는 것을 알리기 위해 초록색 배경으로 제작.



[스마트 거울 기기 음성인 명령어 화면 레이아웃]

- 음성인식 명령어에 집중할 수 있게 다른 정보 화면을 어둡게 제작.
- 일정 시간이 지나면 다시 원래의 화면으로 돌아오게 제작.

• 음성인식 명령어 기능 제작

- "다음" : 다음 노래 재생
- "이전" : 이전 노래 재생
- "그만" : 재생 중인 노래 중지
- "재생" : 중지한 노래 다시 재생
- "소리 키워" : 소리를 키움
- "소리 줄여" : 소리를 줄임
- "도와줘" : 사용자 음성인식 매뉴얼 제공

• 스마트 거울 기기 카메라를 이용한 영상처리 테스트

- 얼굴, 눈 탐지 알고리즘 테스트

	<ol style="list-style-type: none"> 1. 카메라로 영상을 입력받는다. 2. 입력받은 영상에 Histogram Equalization 알고리즘을 적용하여 명암을 균일화 한다. 3. 균일화 된 영상에서 lbpcascade_frontalcatface 알고리즘을 이용하여 얼굴을 탐지한다. 4. 탐지한 얼굴 영역(초록색 사각형)에서 haarcascade_eye 알고리즘을 이용하여 눈을 탐지한다. 5. 적용 확인을 위해 원본 영상에 사각형 테두리를 씌워 화면으로 출력한다.
<p>[얼굴탐지, 눈 탐지 테스트 화면]</p>	

- 한쪽 눈(윙크) 인식 알고리즘 테스트

	<ol style="list-style-type: none"> 1. 위의 얼굴 , 눈 탐지 알고리즘을 이용하여 한쪽 눈(윙크)을 탐지한다. 2. 한쪽 눈이 탐지 되었을 경우 다음 5개의 영상 프레임 모두 한쪽 눈만 탐지 되었을 때 윙크로 간주한다. 3. 윙크로 간주되면 구글 TTS(음성인식)를 실행한다. 4. 구글 TTS가 작동하는 동안은 어떠한 영상처리 알고리즘을 적용하지 않는다.
<p>[한쪽 눈(윙크) 인식 알고리즘 테스트 화면]</p>	

• 스마트 거울 제작

	
<p>1. 좌측 ODRROID-C2 와 우측 모니터</p>	<p>2. 모니터 분해</p>
	
<p>3 모니터 패널과 모니터 드라이브, ODRROID-C2 일체화</p>	<p>4. 스마트 거울에 사용할 유리 준비</p>
	
<p>5. 유리에 원웨이 필름을 부착</p>	<p>6. 합판과 골판지, 하드보드지로 틀 제작 준비</p>
	
<p>7. 유리, 모니터, 틀 가 조립</p>	<p>8. 조립 후 테스트</p>

- 스마트 거울 완성 모습



완성한 스마트 거울의 전면



완성한 스마트 거울의 후면

3. 참고문헌

SoundCloud API (노래) – <https://developers.soundcloud.com/>

Daum Local API (위치 정보) – <http://developers.daum.net/services/apis/local>

OpenWeatherMap API (날씨 정보) – <http://openweathermap.org/api>

안드로이드 OpenCV – <http://opencv.org/platforms/android.html>

안드로이드 OpenCV Cascade Classifier –
http://docs.opencv.org/2.4/doc/tutorials/objdetect/cascade_classifier/cascade_classifier.html

Histogram equalization 알고리즘 –
<http://warmz.tistory.com/entry/%ED%9E%88%EC%8A%A4%ED%86%A0%EA%B7%B8%EB%9E%A8-%EA%B7%A0%EC%9D%BC%ED%99%94Histogram-Equalization>

안전한 비밀번호 저장 – <http://d2.naver.com/helloworld/318732>

ODROID-C2 –
http://www.hardkernel.com/main/products/prdt_info.php?g_code=G145457216438

Dribbble (안드로이드 어플리케이션 디자인 프로토타입) – <https://dribbble.com/>

Icons8 (무료 모바일 아이콘) – <https://icons8.com/>

Websequencediagram (시퀀스 다이어그램) – <https://www.websequencediagrams.com/>