

디지털 세상 안에서 당신의 따뜻한 추억을 되살려 줄

Portrait



By CJM, HDH, KMK, SYA, YJH Dec 17, 2014



Contents

Prologue. Why Photo?

Chapter 1. Portrait Introduction

Chapter 2. Demonstration

Epilogue. Future Strategy

Q & A

Why Photo?

- 01. Beginning
- 02. Problem
- 03. Benchmark
- 04. Solution

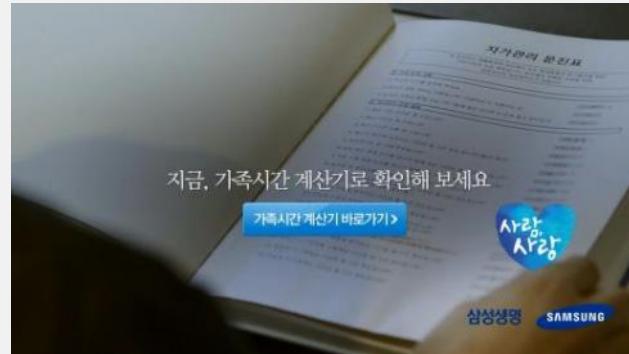


Beginning // Why Photo?

Brilliant memories || H자동차



가족시간 계산기 || 삼성생명



뜻밖의 퇴근 || M의류브랜드



오랜 시간을 가족과 함께 해온 자동차와의 새로운 시작이나
가족과 함께 할 남은 시간을 통해 다시 깨닫게 된 가족의 소중함과 같은

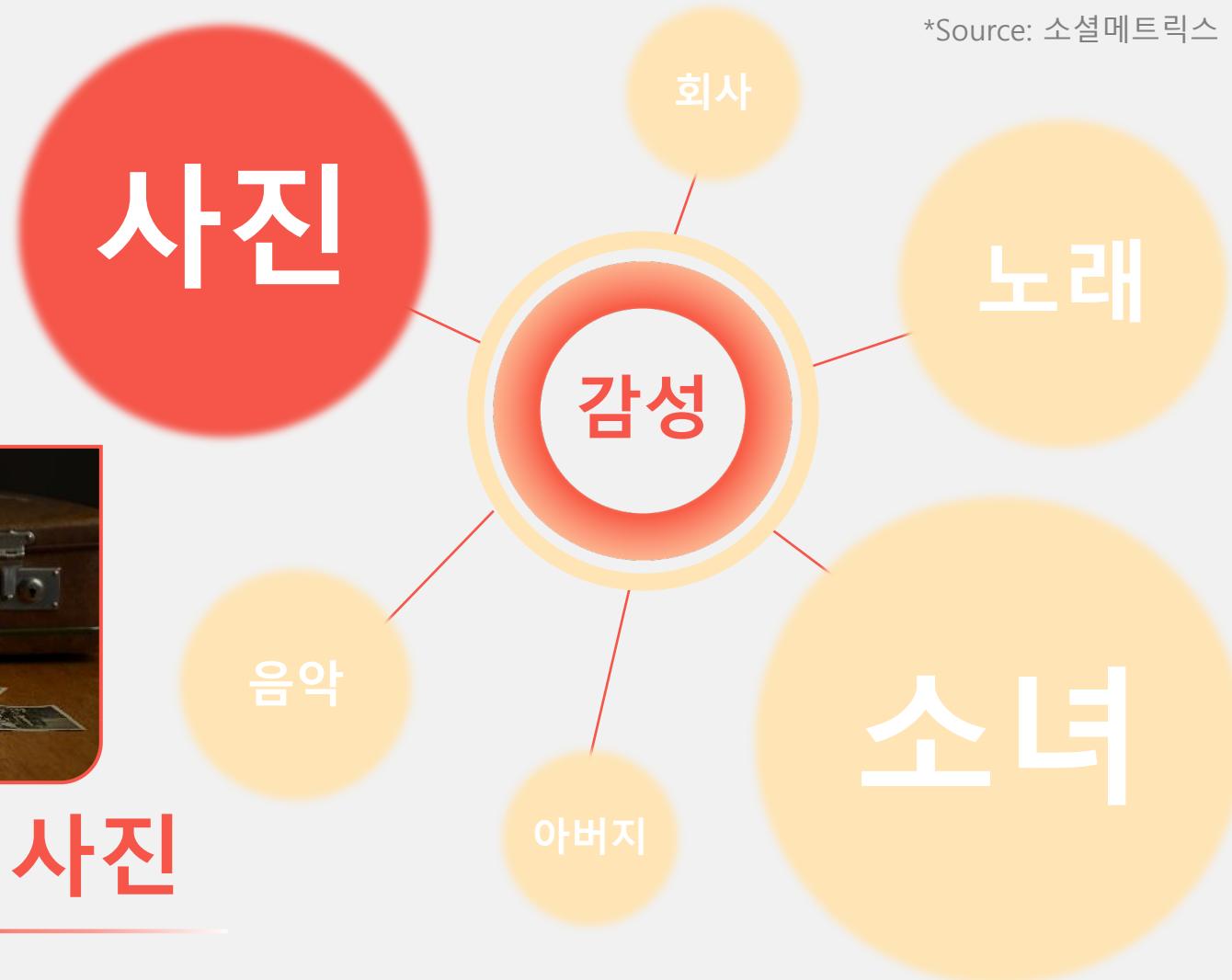
잊고 있었던 소중한 것에 대한 새로운 가치

지쳐있던 직장인에게 소원과 함께 주어진 뜻밖의 퇴근처럼
각박한 현실 속에서 얻을 수 있는 소소하고 따듯한 감성

Beginning // Why Photo?



가장 감성적인 아이템.. 사진



Problem // Lose sensibility

*Source: 미래창조과학부



*Source: eMarketer, PrivCo, Statista

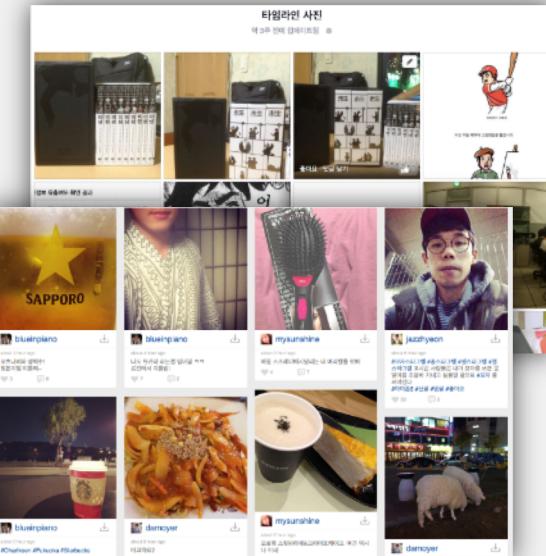
디지털 기기의 발달과 보급으로 인한
사진의 **범람**, 그리고 사진의 **희소성** 상실

Problem // Lose sensibility

0-2.	2011-04-04 오후...	파일 블더
1-0.블로그	2010-11-03 오후...	파일 블더
3-0. 디카원본zip	2011-04-23 오후...	파일 블더
7-0.디카원본2008	2009-05-31 오후...	파일 블더
7-1.디카원본2009	2010-09-09 오후...	파일 블더
7-2.디카원본2010	2010-09-09 오후...	파일 블더
7-4.디카원본2011	2011-01-29 오후...	파일 블더
7-5.핸드폰사진	2009-08-16 오후...	파일 블더
7-6.사진 인화	2010-04-29 오후...	파일 블더
8-0.월이풀간지	2009-10-03 오후...	파일 블더
8-1.월이안화동영상	2010-06-29 오후...	파일 블더
8-3.월이교복자료	2011-01-22 오후...	파일 블더
8-4.남우자로(이학서,월이)	2010-04-25 오후...	파일 블더
8-5.	2009-08-16 오후...	파일 블더
8-6.남우영화	2010-07-25 오후...	파일 블더
8-7.황남귀(취미,관심)	2011-01-08 오후...	파일 블더
9-0.	2009-08-16 오후...	파일 블더
9-1.	2009-09-16 오후...	파일 블더
9-2.	2009-10-03 오후...	파일 블더
10-0.밀린일	2010-05-12 오후...	파일 블더
11-1.	2011-02-09 오후...	파일 블더
15-0.자격증	2010-09-11 오후...	파일 블더
15-1.시간제	2011-01-25 오후...	파일 블더
15-2.동화사	2011-01-25 오후...	파일 블더
15-3.영어공부	2011-02-12 오후...	파일 블더
20-0.Fonts	2010-04-08 오후...	파일 블더
21.부산	2011-02-16 오후...	파일 블더
22.우주2층	2011-03-06 오후...	파일 블더
23.일	2009-10-03 오후...	파일 블더
24.프로그램	2011-02-11 오후...	파일 블더



중구난방식의 정리



저장소 다양화로 어려운 검색

많아진 사진이 얹히고 설킨 사진들로 인하여
추억이 아닌 **일**이 되어버린 사진

Problem // Rebirth Photo' Sensibility

REVERSE
&
REBIRTH

Benchmark // Analog Album

아날로그 앨범



주로 이벤트를 기준으로 분류

실물에 대한 물리적 희소성
인화를 위해 상대적으로 높은 비용
공유하기 위해 물리적인 장소/시간 필요

굵직한 이벤트를 중심으로 정리된 아날로그 앨범은
찾기는 어렵지만 **추억**을 되새기는 감성에 중점

Benchmark // Digital Application

디지털 어플



주로 SNS을 기반으로 공유에 초점

스마트폰을 통한 직관적인 UI/UX
GPS와 같은 센서 활용 가능
휴대전화 사진에 중점
사진 보관, 분류, 검색 불편

GPS, 직관적인 UI/UX를 통해서
쉽고 빠르게 정리 할 수 있지만 공산품의 느낌

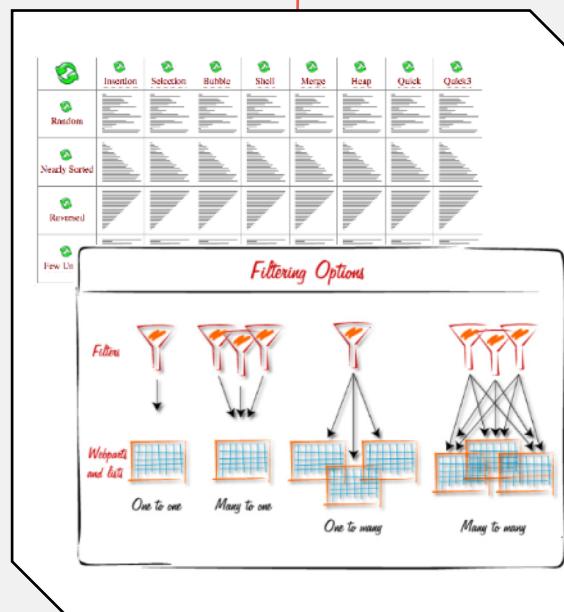
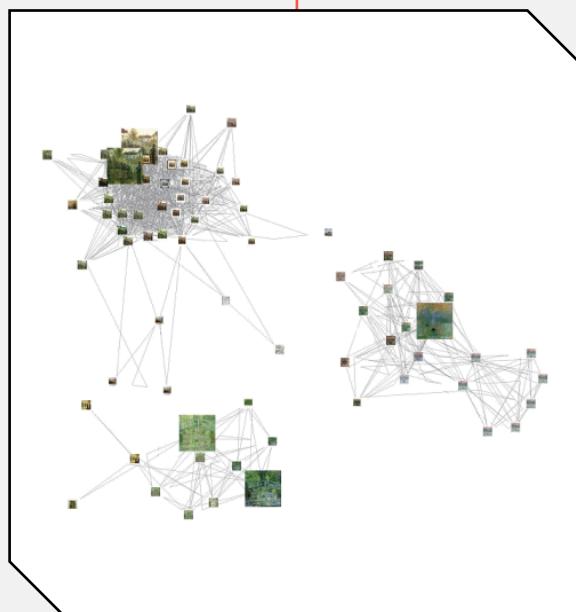
Solution // Digital Photo Management



효과적인 사진 관리

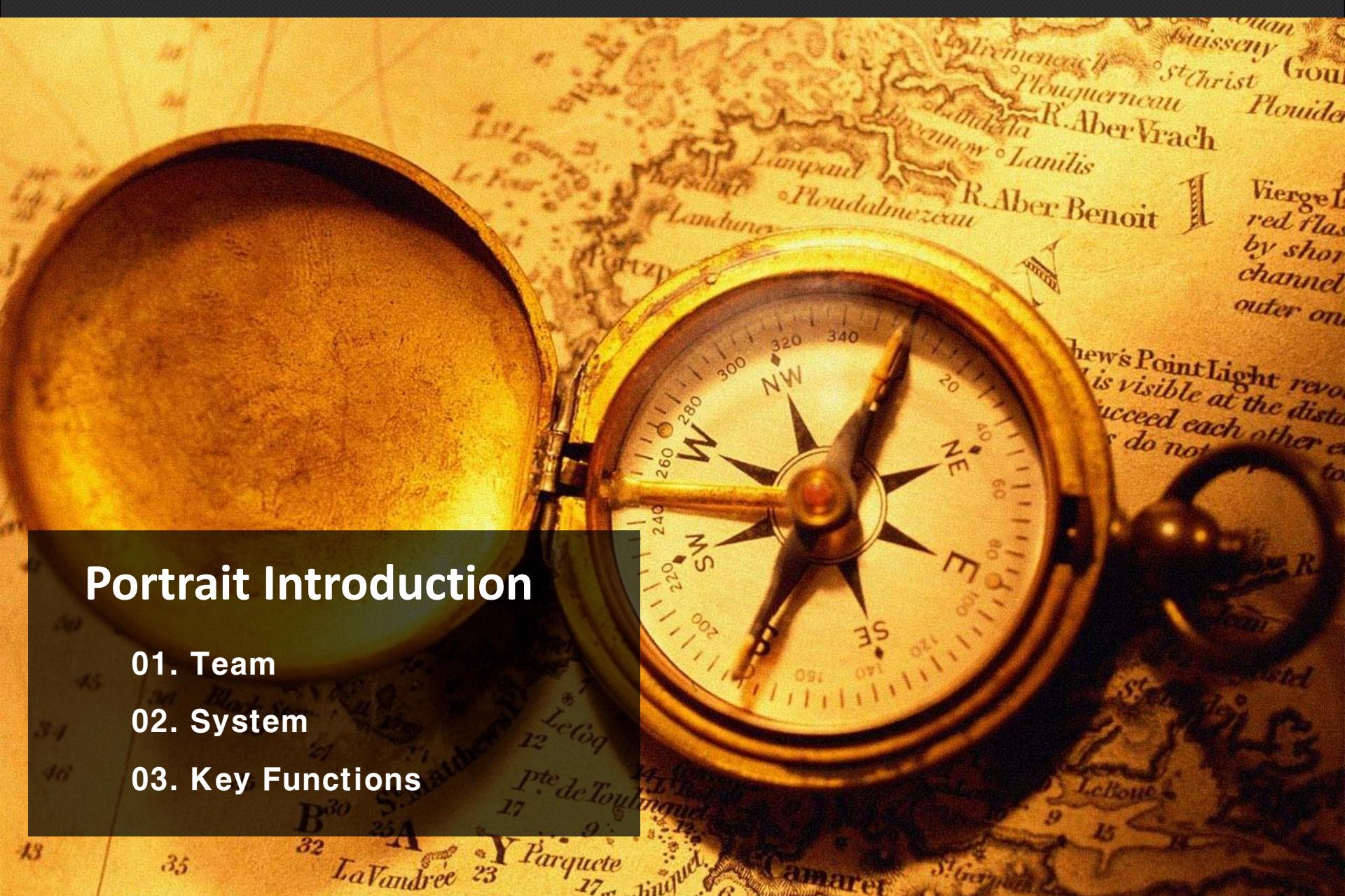
체계적인 사진 검색

On-Offline Publishing



Portrait Introduction

01. Team
02. System
03. Key Functions



Team // Team Members & Schedules

한동훈(AD)

- Android Developer 및 GPS



최준모(FE)

- Web 및 Android UX



서울아(ME)

- Web과 Server 간 조율
- Web UI 보조



김민경(PM)

- Project Manager
- Project 일정 관리
- Web UI 및 Design 담당



유주형(TM/BE)

- Technical Manager
- Server-side Java Client 및 Database

목표

초기개발

기능구현

통합 및 테스트

Web

Google API 분석
프로토타입

DB 모델링 및 UI 초안 구현

반응형 웹 구현,
안드로이드 연동
및 최종 테스트

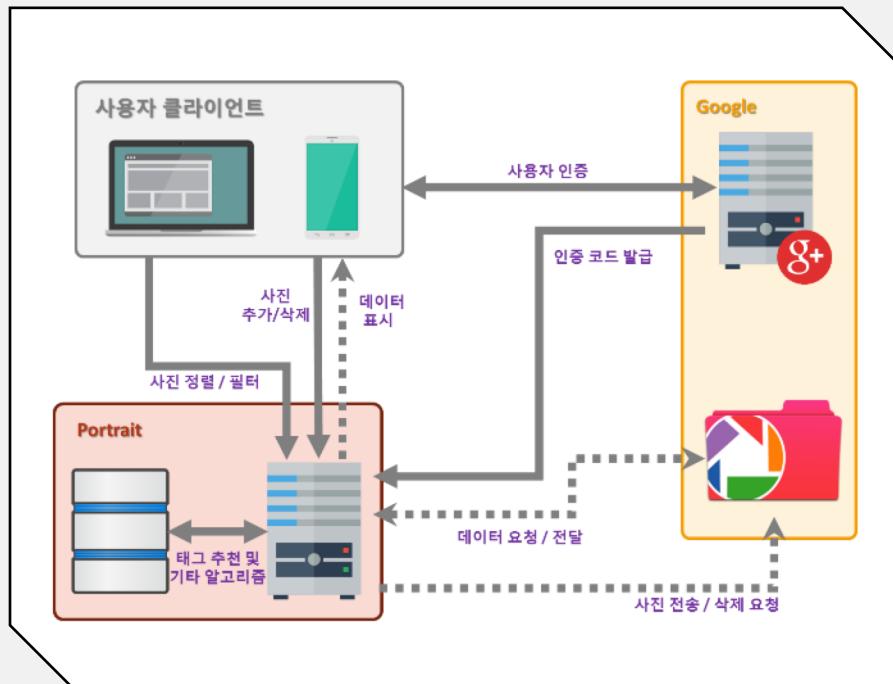
Android

GPS/업로드 모듈 구현

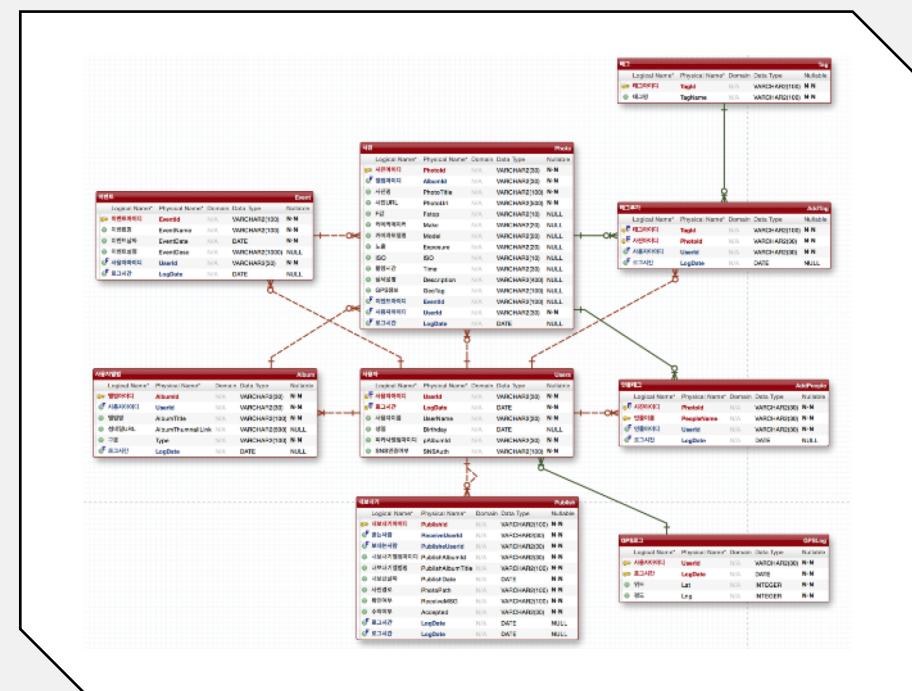
안드로이드 어플리케이션 구현

System // System Topology & DataBase

Server & Client Topology



Database Modeling



Google API를 적극적으로 활용하여

보안 및 데이터 저장소 아웃소싱

Key Functions // Core Algorithm

The screenshot shows the 'Photo Sort Guide' feature within the app. At the top, there's a navigation bar with links: HOME, ALBUM, TIMELINE, PEOPLE, LOCATION, UPLOAD, SHARE, SEARCH, and LOGOUT. Below the navigation, there's a header 'Photo Sort Guide' with a location pin icon and the text '10.0km로 모아보기'. To the right of this is a small preview of four photos. The main content area has two sections: one for location ('10.0km로 모아보기') and one for time ('1개월 단위로 모아보기'). The location section includes icons for location and calendar, and lists distances: 500m, 1km, 10km, 100km, and 1000km. The time section includes icons for location and calendar, and lists time periods: 1시간, 1일, 1개월, and 1년. Below these sections is a list of months from 2007 to 2014.

Period	Distance
1시간	
1일	
1개월	
1년	
2014년 12월	
2014년 11월	
2014년 08월	
2014년 01월	
2013년 07월	
2008년 11월	
2007년 07월	

Below the main content, there are three location suggestions: 대한민국 경기도 가평군, 대한민국 서울특별시, and 일본 오키나와 현 나카가미 군.

시간과 장소를 기반으로 한 편리한 분류 기능

사용자의 편한 사진 분류를 위해서 사진에 담긴 **시간 및 장소 정보를 이용하여 자동으로 분류**

장소의 경우 사용자가 선택한 범위를 기반으로 **500m에서 1000km**까지 범위 별로 사진을 그룹으로 표시

시간의 경우 **1시간부터 1년**까지 단위 시간별 그룹으로 표시

분류된 사진을 선택하여 **바로 앤범으로 생성**할 수 있는 기능 지원

Key Functions // Core Algorithm

PORTRAIT MEMORIES

HOME ALBUM TIMELINE PEOPLE LOCATION UPLOAD SHARE SEARCH LOGOUT

TIME LINE // WHOLE YOUR LIFE

남미여행

남미여행

호박이와

SCSA

MIN KYEONG KIM의 생일

2014/12/28

2014/12/25

일대기를 한 눈에 볼 수 있는 타임라인 기능

사용자의 편한 사진 분류를 위해서
사진에 담긴 **시간 및 장소 정보를**
이용하여 자동으로 분류

장소의 경우 사용자가 선택한 범위를
기반으로 **500m에서 1000km**까지 범위
별로 사진을 그룹으로 표시

시간의 경우 **1시간부터 1년**까지 단위
시간별 그룹으로 표시

분류된 사진을 선택하여 **바로**
앨범으로 생성할 수 있는 기능 지원

Key Functions // Core Algorithm

The screenshot shows a search results page for a person's location. At the top, there is a navigation bar with links for HOME, ALBUM, TIMELINE, PEOPLE, LOCATION, UPLOAD, SHARE, SEARCH, and LOGOUT. Below the navigation bar, the text "ABOUT PEOPLE // AND WHO MAKE YOU BRILLIANT" is displayed. The main content area shows several thumbnail images of people and a map of South Korea with a highlighted location. A large image of a dog wearing glasses is prominently displayed. At the bottom, there is a horizontal strip of smaller thumbnail images.

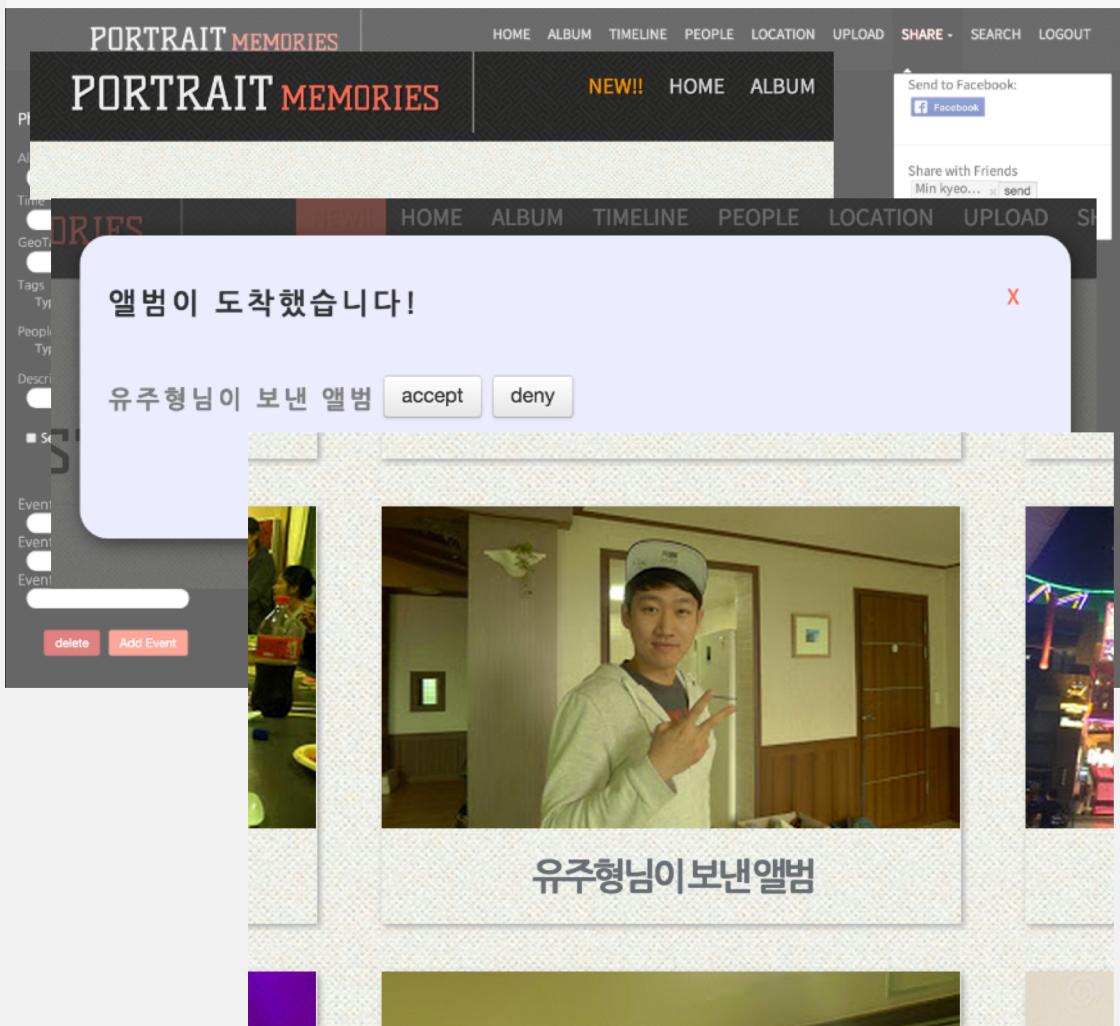
인물과 장소를 기반으로 한 사진 보기 기능

사용자에 의해서 입력된 **인물 태그 및 사진의 장소**를 기반으로 한 별도의 보기 페이지를 제공

사용자가 입력한 인물 태그를 기반으로 **대상 인물의 사진 개수**에 맞추어 자동으로 크기가 조절되어 표시

사진 내 장소를 기반으로 **지도에서 사진을 선택**하여 볼 수 있으며, 대상 사진의 주변 사진도 함께 볼 수 있도록 제공

Key Functions // Core Algorithm



옛 추억을 되살릴 수 있는 온라인 공유 기능

사용자에 의해서 입력된 **인물 태그 및 사진의 장소**를 기반으로 한 별도의 보기 페이지를 제공

사용자가 입력한 인물 태그를 기반으로 **대상 인물의 사진 개수**에 맞추어 자동으로 크기가 조절되어 표시

사진 내 장소를 기반으로 **지도에서 사진을 선택**하여 볼 수 있으며, 대상 사진의 주변 사진도 함께 볼 수 있도록 제공

Key Functions // Core Algorithm



검색할 단어를 하단에서 체크해주세요

DATE

PEOPLE TAGS LOCATION

이름을 입력하세요 태그를 입력하세요 장소를 입력하세요

PEOPLE

- 최충보
- yoolaseo
- 호박이
- 유주형

TAGS

- 같다
- 고양이
- 상의중
- 글
- 브이

LOCATION

- 대한민국 경기도 가평군 청평면 동
- 대한민국 서울특별시 강남구 역삼동

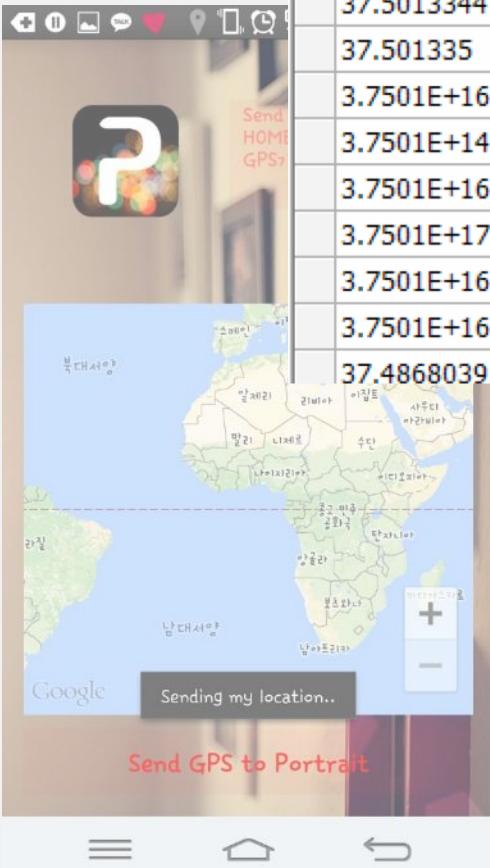
다양한 조건을 통한 사진 검색 기능

사용자에 의해서 입력된 **사용자 태그, 인물 태그 및 사진의 시간, 장소**를 조건으로 사진을 검색

다수의 사진 중에서 원하는 사진을 빠르게 검색할 수 있도록 **다양한 차원으로 검색 가능한 알고리즘**을 구현

사용자가 입력한 사용자 태그, 인물 태그 및 사진의 장소 정보를 기본적으로 출력해줌으로써 **사용자가 검색을 위한 키워드 입력도 편리하게 할 수 있도록 기능** 구현

Key Functions // Core Algorithm



안드로이드의 GPS를 활용한 GPS Logging 기능

DSLR을 사용하는 대다수의 사용자들의 경우 **별도의 장비가 필요없이** 사진을 찍은 시점의 GPS 정보가 업로드 시 입력되도록 구현

안드로이드 어플리케이션을 통해
사용자의 GPS 정보를 확인, DB에 저장

업로드되는 사진 중 GPS 정보가 없는 사진에 한하여 촬영 시간을 기준으로
저장된 GPS 정보를 자동으로 입력

카메라 **GPS Logging 장비가 20~30만원**의 고가에 책정되어 있는 점에서 사용자에게 큰 메리트

Key Functions // Core Algorithm

```

42         flow = new GoogleAuthorizationCodeFlowBuilder(HTTP_TRANSPORT,
43             JSON_FACTORY, CLIENT_ID, CLIENT_SECRET, SCOPE).build();
44
45     generateStateToken();
46 }
47
48 public String buildLoginUrl() {
49     final GoogleAuthorizationCodeRequestUrl url = flow.newAuthorizationUrl();
50     return url.setRedirectUri(CALLBACK_URI).setState(stateToken).build();
51 }
52
53 public String buildTokenUrl() {
54     String authSubUrl = AuthSubUtil.getRequestUrl(hostedDomain, nextUrl, scope, secure,
55     System.out.println(authSubUrl);
56     return authSubUrl;
57 }
58
59 private void generateStateToken() {
60     SecureRandom sr1 = new SecureRandom();
61     stateToken = "google;" + sr1.nextInt();
62 }
63
64 public String getStateToken() {
65     return stateToken;
66 }
67
68 public String authorizationUrl() {
69     return AuthSubUtil.getRequestUrl();
70 }
71
72 public String getUserInfoJson() {
73
74     final GoogleTokenResponse tokenResponse = tokenService.getAccessToken();
75     final Credential credential = tokenResponse.getCredential();
76     final HttpRequestFactory requestFactory = tokenResponse.getRequestFactory();
77     final GenericUrl url = new GenericUrl(tokenResponse.getAccessTokenUrl());
78     final HttpRequest request = requestFactory.buildGetRequest(url);
79     System.out.println(request.getUrl());
80     request.getHeaders().set("User-Agent", "Portrait-Memories");
81     System.out.println(request.getUrl());
82     final String jsonIdentity = tokenService.getUserInfo();
83
84     return jsonIdentity;
85 }
86
87 }
88 }

1924 */
1925 * 사진 JSONObject를 Parsing 하여 Photo 객체 생성
1926 */
1927 private Photo parseJSON(JSONObject photo, String userId) throws Exception {
1928     Photo temp = new Photo();
1929     temp.setPhotoId(URLDecoder.decode(photo.getJSONObject("title")
1930         .getString("content"), "UTF-8"));
1931     temp.setPhotoId(String.valueOf(photo.getLong("gphoto:id")));
1932     temp.setTime(String.valueOf(photo.getLong("gphoto:timestamp")));
1933     if (photo.getJSONObject("exif:tags").has("exif:fstop")) {
1934         temp.setFstop(String.valueOf(photo.getJSONObject("exif:tags")
1935             .getDouble("exif:fstop")));
1936     }
1937     if (photo.getJSONObject("exif:tags").has("exif:make")) {
1938         temp.setMake(photo.getJSONObject("exif:tags")
1939             .getString("exif:make"));
1940     }
1941     if (photo.getJSONObject("exif:tags").has("exif:model")) {
1942         temp.setModel(photo.getJSONObject("exif:tags").getString(
1943             "exif:model"));
1944     }
1945     if (photo.getJSONObject("exif:tags").has("exif:exposure")) {
1946         temp.setExposure(String.valueOf(photo.getJSONObject("exif:tags")
1947             .getDouble("exif:exposure")));
1948     }
1949     if (photo.getJSONObject("exif:tags").has("exif:iso")) {
1950         temp.setIso(String.valueOf(photo.getJSONObject("exif:tags").getInt(
1951             "exif:iso")));
1952     }
1953     temp.setPhotoUrl(photo.getJSONObject("content").getString("src"));
1954     if (photo.getJSONObject("media:group")
1955         .getJSONObject("media:description").has("content")) {
1956         temp.setDescription(URLDecoder.decode(
1957             photo.getJSONObject("media:group")
1958                 .getJSONObject("media:description")
1959                 .getString("content"), "UTF-8"));
1960     } else {
1961         temp.setDescription("{ 'album':'', 'event':'', 'comments':'', 'tag':''}");
1962     }
1963     if (photo.has("georss:where")) {
1964         temp.setGeoTag(photo.getJSONObject("georss:where")
1965             .getJSONObject("gml:Point").getString("gml:pos"));
1966     }
1967     temp.setUserId(userId);
1968
1969     return temp;
1970 }

```

Google API를 활용하여
보안성 및 저장공간 확보

Google Open API로 제공되는 로그인,
구글 웹 앨범, 구글 맵 등의 기능을
적극적으로 활용하여 웹 어플리케이션에
반영

구글의 OAuth 2.0을 통해 로그인하여
사용자 정보와 관련된 내용을 프로젝트
서버에 보관하지 않음으로써 사용자
계정에 대한 **보안을 구글에 위임**

구글 웹 앨범을 사진 저장소로
활용함으로써 특정 조건(사진 크기
2048*2048)을 충족하는 경우에 한하여
무제한의 저장 공간을 확보

사진과 관련하여 구글에서 제공하지 않는
정보를 저장하는 별도의 방식을 개발하여 더욱 다양한 정보를 저장할 수
있도록 기능 개발

Demonstration

- 01. Photo Sorting
- 02. Timeline
- 03. People View
- 04. Publish
- 05. Search
- 06. Android Upload



Demonstration // Photo Sorting

PORTRAIT MEMORIES

HOME ALBUM TIMELINE PEOPLE LOCATION UPLOAD SHARE SEARCH LOGOUT

~WELCOME TO PORTRAIT~



< >

Demonstration // Timeline

PORTRAIT MEMORIES

HOME ALBUM TIMELINE PEOPLE LOCATION UPLOAD SHARE SEARCH LOGOUT

~WELCOME TO PORTRAIT~

A close-up photograph of a person's hands holding a Polaroid Land Camera Model 180. The camera is black with a silver base and lens. The lens has text on it, including "Tessar Opt. 50mm f=1:4.5", "No 54602", and "POLAROID LAND CAMERA MODEL 180". In the background, another Polaroid camera is visible, slightly out of focus. The background is dark.

< >

localhost:9090/Portrait_Dev/home.do

Demonstration // People View

PORTRAIT MEMORIES

HOME ALBUM TIMELINE PEOPLE LOCATION UPLOAD SHARE ▾ SEARCH LOGOUT

ALBUM LIST //FOR YOUR MEMORIES



미분류 앨범



전체 앨범



휴지통

Demonstration // Publish

PORTRAIT MEMORIES

HOME ALBUM TIMELINE PEOPLE LOCATION UPLOAD SHARE SEARCH LOGOUT

~WELCOME TO PORTRAIT~

A close-up photograph of a person's hands holding a Polaroid Land Camera Model 180. The camera is black with a silver lens and a flash unit attached. The brand name "POLAROID" and model number "LAND CAMERA MODEL 180" are printed on the front. In the background, another camera is visible, slightly out of focus. The entire image is framed by a white border, which is itself set against a dark background.

< >

Demonstration // Search

PORTRAIT MEMORIES

HOME ALBUM TIMELINE PEOPLE LOCATION UPLOAD SHARE SEARCH LOGOUT

ALBUM LIST //FOR YOUR MEMORIES



미분류 앨범



전체 앨범

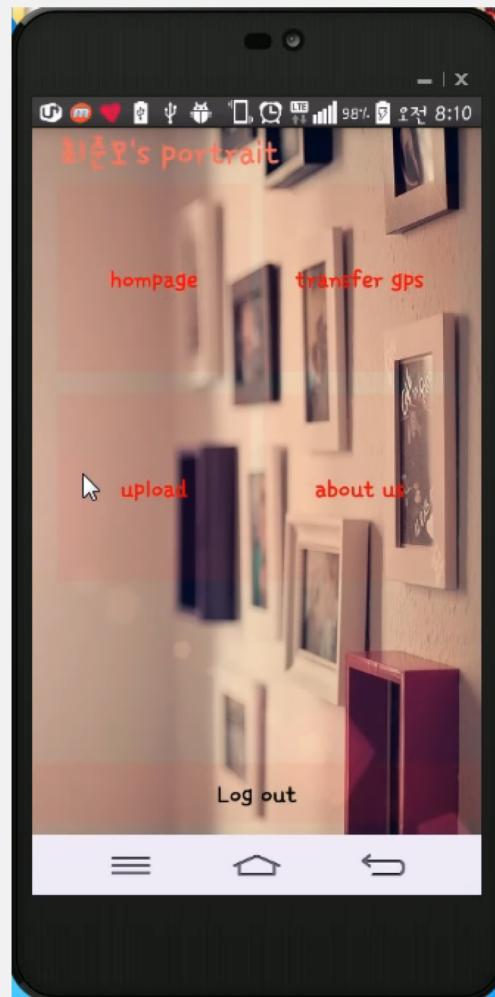


귀여운 동물들



휴지통

Demonstration // Android Upload





Future Strategy

01. Business Model

02. Strategy Plan

Biz Model // Business Model Canvas

Key Partner



Google Open API



오프라인 인화 사이트



온라인 사진 커뮤니티

Key Activities



사진 분류 /
검색 / 가치
발견



온라인/오프라
인 퍼블리싱
(SNS, 포토북 등)

Key Resources



물적: 온라인
플랫폼
지적: 검색
알고리즘
개발/유지보수
금융: 소셜펀딩



Value Proposition

체계적인
디지털
사진정리

내가 원하는
사진을 쉽고
빠르게 검색



신뢰성있는
클라우드를
통해 보다
안전한
사진 보관

자신의
추억을
차근차근 볼
수 있는
타임라인

Customer Relationship



사진의 분류 방식에 대한
지속적인 피드백
(주로 SNS와 메일을 활용)

Channel



웹 페이지 및 앱 마켓

Customer Segmentation



과거 사진 앨범을
추억하고, 자식에게
앨범을 물려주고 싶은
30대 젊은 부모



사진을 많이 찍고 있지만
정작 제대로 관리하지는
못하고 있는 20대

Cost Structure



도메인 호스팅 비용



서버/디비 등 H/W 비용



홈페이지 및
어플리케이션 광고 수익

Revenue Stream



오프라인 인화
사이트와의
연계를 통한 커미션

Biz Model // Business Revenue Strategy



Step One Advertisement Revenue

- 모바일과 홈페이지의 광고 수익
- 웹페이지: Google Adsense
- 어플리케이션: Android Admob, iOS iAd



Step Two Collaboration with Offline Printing Sites

- 오프라인 인화 사이트 연계
- 자연스러운 인화 유도 및 유입 고객에 따른 부가 수익
- 어플리케이션이 부재한 인화 사이트 모바일 유통 채널 제공



Step Three Social Funding

- 기존 수익 구조를 토대로 Social Funding
- Portrait를 하나의 Brand화
- 추가적인 비즈니스 모델 도출

Strategy Plan // Business Revenue Strategy

인물 얼굴 인식 기능

오프라인 인화 사이트 연계

 <p>596 x 405 [4:3]</p> <p>퀵미리보기/[트리밍]</p> <ul style="list-style-type: none"> <input checked="" type="radio"/> 이미지풀 <input type="radio"/> 이미지풀 <input type="radio"/> 이미지풀-유대 <input type="radio"/> 이미지풀-유대 <p>■ 밖에 노경하여 인화하기 ■ 활엽들자 넣기</p>	 <p>300 x 368 [4:3]</p> <p>퀵미리보기/[트리밍]</p> <ul style="list-style-type: none"> <input checked="" type="radio"/> 이미지풀 <input type="radio"/> 이미지풀 <input type="radio"/> 이미지풀-유대 <input type="radio"/> 이미지풀-유대 <p>■ 밖에 노경하여 인화하기 ■ 활엽들자 넣기</p>
 <p>763 x 960 [4:3]</p> <p>퀵미리보기/[트리밍]</p> <ul style="list-style-type: none"> <input checked="" type="radio"/> 이미지풀 <input type="radio"/> 이미지풀 <input type="radio"/> 이미지풀-유대 <input type="radio"/> 이미지풀-유대 <p>■ 밖에 노경하여 인화하기 ■ 활엽들자 넣기</p>	 <p>672 x 960 [4:3]</p> <p>퀵미리보기/[트리밍]</p> <ul style="list-style-type: none"> <input checked="" type="radio"/> 이미지풀 <input type="radio"/> 이미지풀 <input type="radio"/> 이미지풀-유대 <input type="radio"/> 이미지풀-유대 <p>■ 밖에 노경하여 인화하기 ■ 활엽들자 넣기</p>

온라인 앨범 기능

The image consists of two main parts. The top portion is a collage of six photographs: a detailed watercolor-style drawing of a large, multi-story house with a prominent front porch; a black and white portrait of Martin Luther King Jr. wearing a fedora; a black and white photograph of King smiling; a color photograph of King in profile; a black and white portrait of King looking slightly to the side; and a black and white photograph of King smiling broadly. The bottom portion shows a screenshot of a digital photo stream interface. On the left, there's a vertical navigation bar with a 'Previous' button at the top, followed by a list of photo albums and a search bar. The main area displays a grid of thumbnail images, each representing a different photo album. One album in the center is titled 'May 2012 Photo Stream' and features a large thumbnail of a park scene with people sitting under trees. At the bottom of the interface, there's a footer with text and small thumbnail images.

얼굴 자동 인식 관련 Open API(ex., Cloudinary 등)를 통해 사진 등록 시 자동으로 인물 얼굴을 인식 하는 기능 구현

오프라인 사진 인화 사이트(Zixx, OkPhotoLink 등)와의 연계를 통해서 Portrait에서 바로 사진 인화 사이트에 주문을 등록할 수 있도록 연계

온라인에서도 앨범과 같은 UI를 구현하여 사진을 마치 앨범처럼 보여주며, 슬라이드 쇼를 동영상으로 저장하여 보관할 수 있도록 기능 구현

*Learn from yesterday,
Live for today,
Hope for tomorrow*

