

# 컬러 파인더

---

*ColorFinder*

**프로젝트명 :** 컬러 파인더

**기획 의도 및 소개 :**

- 퍼스널 컬러에 대한 관심도 급증에 따라 개발하게 됨.
- 사용자가 입력한 평가 값(화장품 어울림도) 분석에 따른 퍼스널 컬러 예측 및 어울리는 컬러의 화장품 추천 모바일 어플리케이션.

**프로젝트 기간 :** 2016.7 ~ 2017.8 (활동 당시 1,2학년 재학)

**팀 인원 :** 4명(성신여대 IT학부)

담당 역할 :


- 안드로이드 스튜디오를 활용한 앱 개발, 데이터베이스 구축
- 회원 가입, 로그인, 아이템 상세 창 구현, 데이터베이스 설계 및 데이터 입력

특이사항 :

- 오프라인 퍼스널 컬러 진단 업체 ‘컬러즈’에 찾아가 PT 후 약 800여 개 색조제품 컬러 분석 데이터를 직접 공수.
- 퍼스널 컬러 인식 및 관심도에 대한 자체 설문 조사 실시.
- 교내 소프트웨어 경진대회 금상 수상

# 개발 환경과 구축



 : 다루었던 부분

플랫폼 :  안드로이드 ( Target SDK : 7.1    Minimum SDK : 4.0.3 )

개발 툴 :  Android Studio  SublimeText  jupyter

디자인 툴 :  Ps  Ai

사용 언어 :  Java  python  php MySQL

서버 환경 :  php ver.5.5  MySQL ver.5.x UTF-8

## Register\_Activity.java - 회원가입 창 구현

```
private void registerUser(final String name, final String email, final String password) {

    HashMap<String, String> _params = new HashMap<>();
    _params.put("action", "user_register");
    _params.put("name", name);
    _params.put("email", email);
    _params.put("password", password);
    new HttpClass(HttpClass.REGISTER_PROCESS, mHandler, _params).start();

}
```

```
private void registerUser01(final String name, final String email,
                           final String password) {
    // Tag used to cancel the request
    String tag_string_req = "req_register";

    pDialog.setMessage("Registering ...");
    showDialog();

    StringRequest strReq = new StringRequest(Method.POST,
        AppConfig.URL_REGISTER, new Response.Listener<String>() {

            @Override
            public void onResponse(String response) {
                Log.d(TAG, "Register Response: " + response.toString());
                hideDialog();

                try {
                    JSONObject jsonObj = new JSONObject(response);
                    boolean error = jsonObj.getBoolean("error");
                    if (!error) {
                        // User successfully stored in MySQL
                        // Now store the user in sqLite
                        String uid = jsonObj.getString("uid");
                    }
                }
            }
        })
    ;
}
```

```
JSONObject user = jsonObj.getJSONObject("user");
String name = user.getString("name");
String email = user.getString("email");
String created_at = user
    .getString("created_at");

// Inserting row in users table
db.addUser(name, email, uid, created_at);

Toast.makeText(getApplicationContext(), "User successfully registered. Try login now!", Toast.LENGTH_LONG).show();

// Launch login activity
Intent intent = new Intent(
    RegisterActivity.this,
    LoginActivity.class);
startActivity(intent);
finish();
} else {

// Error occurred in registration. Get the error
// message
String errorMsg = jsonObj.getString("error_msg");
Toast.makeText(getApplicationContext(),
    errorMsg, Toast.LENGTH_LONG).show();
```

## Login\_Activity.java

### - 로그인 창 구현

```
private void checkLogin(String _email, String _password){
    HashMap<String, String> _params = new HashMap<>();
    _params.put("action", "user_login");
    _params.put("email", _email);
    _params.put("password", _password);
    new HttpClass(HttpClass.LOGIN_PROCESS, mHandler, _params).start();
}
```

```
// Create login session
session.setLogin(true);

// Now store the user in SQLite
String uid = jsonObj.getString("uid");

JSONObject user = jsonObj.getJSONObject("user");
String name = user.getString("name");
String email = user.getString("email");
String created_at = user
    .getString("created_at");

// Inserting row in users table
db.addUser(name, email, uid, created_at);

// Launch main activity
if(Common_Class.get_SharedPreferences_eval(LoginActivity.this).equals("empty")){
    Intent intent = new Intent(LoginActivity.this, Step01_Activity.class);
    startActivity(intent);
    finish();
}else {
    Intent intent = new Intent(LoginActivity.this, Main_VIew_Activity.class);
    startActivity(intent);
    finish();
}
```

```
private void checkLogin01(final String email, final String password) {
    // Tag used to cancel the request
    String tag_string_req = "req_login";

    pDialog.setMessage("Logging in ...");
    showDialog();

    StringRequest strReq = new StringRequest(Method.POST,
        AppConfig.URL_LOGIN, new Response.Listener<String>() {

        @Override
        public void onResponse(String response) {
            Log.d(TAG, "Login Response: " + response.toString());
            hideDialog();

            try {
                JSONObject jsonObj = new JSONObject(response);
                boolean error = jsonObj.getBoolean("error");

                // Check for error node in json
                if (!error) {
                    // user successfully logged in
                }
            } catch (JSONException e) {
                e.printStackTrace();
            }
        }
    });
}
```

## Product\_Detail.java

### - 아이템 상세 창 구현

```
private void setEvalValue(String _str){
    CfLog.i(TAG, "setEvalValue() / _str : "+_str);

    try{
        JSONObject _obj = new JSONObject(_str);
        mEval = new user_Collect_Value(_obj);
    }catch(Exception e){
        CfLog.e(TAG, e);
    }
}

private void completeEval(){
    CfLog.i(TAG, "completeEval()");
    HashMap<String, String> _params = new HashMap<>();
    _params.put("action", "get_eval_one");
    _params.put("user_id", ""+Main_View_Activity.mUserInfo.id);
    _params.put("item_id", ""+mValue.id);
    new HttpClass(HttpClass.GET_EVAL_ONE, mHandler, _params).start();
}
```

```
@Override
public void onClick(View view) {
    if(view == mBtn_eval){
        showEvalDialog();
    }else if(view == mBtn_like){
        if(Common_Class.save_SharedPreferences_Like(this, ""+mValue.id)){
            Toast.makeText(this, "워시리스트에 추가하였습니다.", Toast.LENGTH_SHORT).show();
        }else{
            Toast.makeText(this, "이미 워시리스트에 추가된 제품입니다.", Toast.LENGTH_SHORT).show();
        }
    }
}
```

```
_btn.setOnClickListener((view) -> {
    HashMap<String, String> _params = new HashMap<>();
    _params.put("action", "insert_eval");
    _params.put("user_id", Main_View_Activity.mUserInfo.id+"");
    _params.put("item_id", _temp.id+"");
    _params.put("fit_score", _seekIv.getText().toString());
    if(_cb.isChecked()){
        _params.put("expert_value", "1");
    }else{
        _params.put("expert_value", "0");
    }
    new HttpClass(HttpClass.INSERT_EVAL, mHandler, _params).start();

    mDialog_eval.dismiss();
});
```

User 정보 테이블

- name**  
 : 유저 이름을 저장
- email**  
 : 메일 주소를 저장
- encrypted password**  
 : 암호화된 패스워드를 저장
- pc\_season**  
 : 봄/여름/가을/겨울 중 사용자  
 의 퍼스널 컬러 계절을 저장
- pc\_detail\_tt**  
 : 라이트/뮤트/브라이트/딥 중  
 사용자의 퍼스널 컬러 디테일을  
 저장

←T→	id	unique_id	name	email	encrypted_password	salt	pc_season	pc_detail_t
<input type="checkbox"/>  	2	58c813dbd6dfa6.90053065	Yj	color@gmail.com	8K4KFmuD6qIrlp4lmZOcp0LRxj84MmY1ZWlyNTU0	82f5eb2554		
<input type="checkbox"/>  	3	590e9bf911e448.02101450	테스트유 저3	cf@colorfinder.com	o7nD2Od3eiRMGv3xSYtzhDVRehUwYWFIZWFmMDhm	0aaeeaf08f		
<input type="checkbox"/>  	4	590e9c07a353b0.01168825	Cf	colfa@colorfinder.com	HQScCdQd6X/2hVleCdHu5RlCpeZhOThiNTY1MWYy	a98b5651f2		
<input type="checkbox"/>  	5	5950a971df6a78.53037065	ㅇ 이아아	dd@gmail.com	U+l8F/YBFjsHk+nG6zSSqFUnZsQwM2VknDhjODRm	03ed48c84f		
<input type="checkbox"/>  	6	5950a97f9c59f6.29250405	ㅇ 이아아	ddd@gmail.com	IfYKL4k5o5JnDBnfompvJ0Sa4zo3ZmM2NDhmNDg5	7fc648f489		
<input type="checkbox"/>  	7	595cb9b309a9f6.49692110	R	r@r.com	ol8d/n/tUvNkXeNnCLOPHYuMu3oxMTJiInjE2Yjgy	112b616b82		
<input type="checkbox"/>  	8	5969d94a86ec63.97414408	테스트	a@a.a	vwEoh8Q+V7YwrKbFpbQm8MgWTN04NGE2NTizNDaw	84a6523400		
<input type="checkbox"/>  	9	5971ca4a263aa0.37212999	1	1@c.com	oBjJt/nsZWDqzvoTs1NxR+WlxbU3Y2NiMzRmNGU4	7ccb34f4e8		
<input type="checkbox"/>  	10	5971ca58d02f21.60279995	1	1@ca.com	U5hWblQMMyU475g+ia1a8c91EwM0M2E2NWlZWYWM0	43a65b3ac4		
<input type="checkbox"/>  	11	5971ca91014d95.41131433	ㅇ ㅈ	1@nav.com	7tAQ8RblPycaxTT0/bmRj/p0Q9RjOGRIyWQyYzMx	c8dbad2c31		
<input type="checkbox"/>  	12	5971ca99bf5895.16113498	ㅇ ㅈ	1@naver.com	9Q4QGfNr/cpEo2BDgHfjQsRtMR5lZWEzDQ0MzU4	eea1d44358		
<input type="checkbox"/>  	13	5972d87f7ce7a2.13155629	윤혜원	youns1102@naver.com	QwDzHxRABt/4uJ5+CmPf7drJfM0ZWJhYTY4ZTRk	4ebaa68e4d		
<input type="checkbox"/>  	14	59756de524bb88.31634772	ㅇ ㅈ	cf@cf.com	/IScLBBIKZWxD3Gj4/ZVKalNqqA4YTY4MDRjYjYw	8a6804cb60		
<input type="checkbox"/>  	15	598eadb82e1495.95950828	aa	aa@aa.aa	vdJL8r4tT3p2jGZbOnmnQYVanV4wMWRIZGYyYjFh	01dedf2b1a		
<input type="checkbox"/>  	16	598eb00de82733.09355733	h	h@h.h	xO56AjXWhM3rHIGT/jhx0/QRQcg0ZGE4ODM5YmFI	4da8839bae		
<input type="checkbox"/>  	17	598eb044e2b282.52058849	M	m@m.m	c80swSBaGrGgk13cn+zarH5QtB5jYmlxYjc1Mjhm	cb1b7528f		
<input type="checkbox"/>  	18	598eb0dee546a1.56024515	J	j@j.j	XrljkmpBaA0YBaylCjWx0JRgE384YWRmNDJiNDQx	8adf42b441		
<input type="checkbox"/>  	19	598ed174f1dad7.60474115	D	d@d.d	RNh3ashtb8zj06sRFwM59bhcr99iMmFmOTE4OGUy	b2af9188e2		
<input type="checkbox"/>  	20	599d44c113bd55.56426170	ace	ace@colorfinder.com	wVJ70iJOvUSHI9n+jwQqs1d93g0MDgzZTIkNWRk	4083e9d5dd		
<input type="checkbox"/>  	21	599e9b38cbd486.25774223	Sungshin	sungshin@colorfinder.com	mKIdAEAt7cjgLt1EtVflwCSZVN1iNzU4MmM3Zjji	b7582c7f2b		
<input type="checkbox"/>  	22	599ea030ba7253.09043961	Sung	sung@colorfinder.com	45AYsBcMEUI8SM0js/MN9zI86QhmZjk1NTFkNWI2	ff9551d5b6		
<input type="checkbox"/>  	23	599ea3516347a8.89169320	Sujung	sujujg@cf.com	Q3ztYeg5E2jFjYp4+0LwroPd3LMxNzg3ZjAxYzE3	1787f01c17		
<input type="checkbox"/>  	24	599ea534bfdea5.87613834	Sujung	sujujg@colorfinder.com	+4p+RdRitKcNlD576u966dWRD3U5NzIkNzgyZjlk	979d782f9d		
<input type="checkbox"/>  	25	599ebda4aecdc7.44511461	테스트유 저3	test@test.com	63zk1pIB8dtHlmoOv9Qm8AOML0RhNWRhZjY3ZTQw	a5daf67e40		



립, 섀도우 컬러 분석 테이블

id	part_id	item_season1	item_season2	item_detail_tone1	item_detail_tone2	item_color_name	brand_name	product_name	color_name
3	lip	SU		LI	MU	pink	에뛰드 하우스	디어마이애나 멜립스-톡(구)	PK003 고민 하는핑크
2	lip	SU		LI	BR	pink	에뛰드 하우스	디어마이애나 멜립스-톡(구)	PK002 애매 모호핑크
1	lip	WI		BR		pink	에뛰드 하우스	디어마이애나 멜립스-톡(구)	PK001 의심 가득핑크
7	lip	AU		DE		red	에뛰드 하우스	디어마이애나 멜립스-톡(구)	RD301 가우 똥 레드
4	lip	SP		LI	MU	pink	에뛰드 하우스	디어마이애나 멜립스-톡(구)	PK004 알쏭 달쏭핑크
5	lip	SP		BR		pink	에뛰드 하우스	디어마이애나 멜립스-톡(구)	PK005 뽀뽀 한핑크
8	lip	AU		DE		red	에뛰드 하우스	디어마이애나 멜립스-톡(구)	RD302 쭈뼛 쭈뼛 레드
9	lip	WI		DE		red	에뛰드 하우스	디어마이애나 멜립스-톡(구)	RD303 싱송 싱송 레드
10	lip	WI		DE		red	에뛰드 하우스	디어마이애나 멜립스-톡(구)	RD304 단호 한 레드
11	lip	AU		MU		red	에뛰드 하우스	디어마이애나 멜립스-톡(구)	RD305 갈팡 질팡 레드
12	lip	SU		LI	MU	red	에뛰드 하우스	컬러립스-핏(단 종)	PK001 매지 컬 핏 핑크
13	lip	SU	WI	BR		pink	에뛰드 하우스	컬러립스-핏(단 종)	PK002실루 엣 핏 핑크
14	lip	SP		BR	LI	pink	에뛰드 하우스	컬러립스-핏(단 종)	PK003 판타 지 핏 핑크
15	lip	SP		LI	MU	orange	에뛰드 하우스	컬러립스-핏(단 종)	OR201 드림 핏 코랄
16	lip	AU		MU		beige	에뛰드 하우스	컬러립스-핏(단 종)	BE101 시스 루 핏 베이

# 구현 부분

- User\_id  
: 사용자 id 저장
- Item\_id  
: 사용자가 평가한 아이템의 아이디 저장
- Fit\_score  
: 사용자가 입력한 어울림 정도 저장
- Expert\_values  
: 전문가 진단 여부를 저장

## User evaluation 테이블

←T→			_id	user_id	item_id	fit_score	expert_value	전문가체크여부
<input type="checkbox"/>			1	3	1130	3		0
<input type="checkbox"/>			2	3	920	3		0
<input type="checkbox"/>			3	3	1032	3		0
<input type="checkbox"/>			4	3	958	2		0
<input type="checkbox"/>			5	3	958	2		0
<input type="checkbox"/>			6	3	1016	0		0
<input type="checkbox"/>			7	3	183	3		0
<input type="checkbox"/>			8	3	261	4		1
<input type="checkbox"/>			9	3	261	4		1
<input type="checkbox"/>			10	3	166	4		1
<input type="checkbox"/>			11	3	149	3		0
<input type="checkbox"/>			12	3	192	3		0
<input type="checkbox"/>			13	3	75	3		0
<input type="checkbox"/>			14	3	219	4		1
<input type="checkbox"/>			15	3	275	4		0
<input type="checkbox"/>			16	3	66	1		0
<input type="checkbox"/>			17	3	247	1		0
<input type="checkbox"/>			18	3	258	3		0
<input type="checkbox"/>			19	3	252	1		0
<input type="checkbox"/>			20	3	161	2		0
<input type="checkbox"/>			21	3	182	2		0
<input type="checkbox"/>			22	3	231	2		0
<input type="checkbox"/>			23	3	159	1		0
<input type="checkbox"/>			24	3	44	1		0
<input type="checkbox"/>			25	3	88	1		0
<input type="checkbox"/>			26	3	214	1		0