

# 모바일 기반 개발 관련 웹사이트 추천 및 사용자화 서비스

201233444 최한동

# 목차

- 서비스 제안 배경
- 서비스 개요
- 서비스 개발
  - 개발 목표
  - 개발 내용
  - 개발 전략
  - 개발 과정

# 서비스 제안 배경

개발자의  
정보  
탐색/습득  
필요

정보 시각화  
및 접근성  
용이한 App

개발 관련  
정보 파편화

# 서비스 제안 배경

## 1. 개발자 관련 웹사이트 증가

국내 개발자 수 꾸준히 증가

→ 하지만 참고할 개발 관련 자료가 대부분 외국어

→ 자료에 쉽게 접근, 정보 공유를 위해 국내 다수의 커뮤니티 증가

## 2. 개발자 관련 웹사이트 파편화

다수의 커뮤니티 생성

→ 동일한 카테고리 내 웹사이트가 너무 많아서 정보 분산화

→ 사용자 입장에서 어떤 웹사이트를 어떻게 참고할 지 혼란

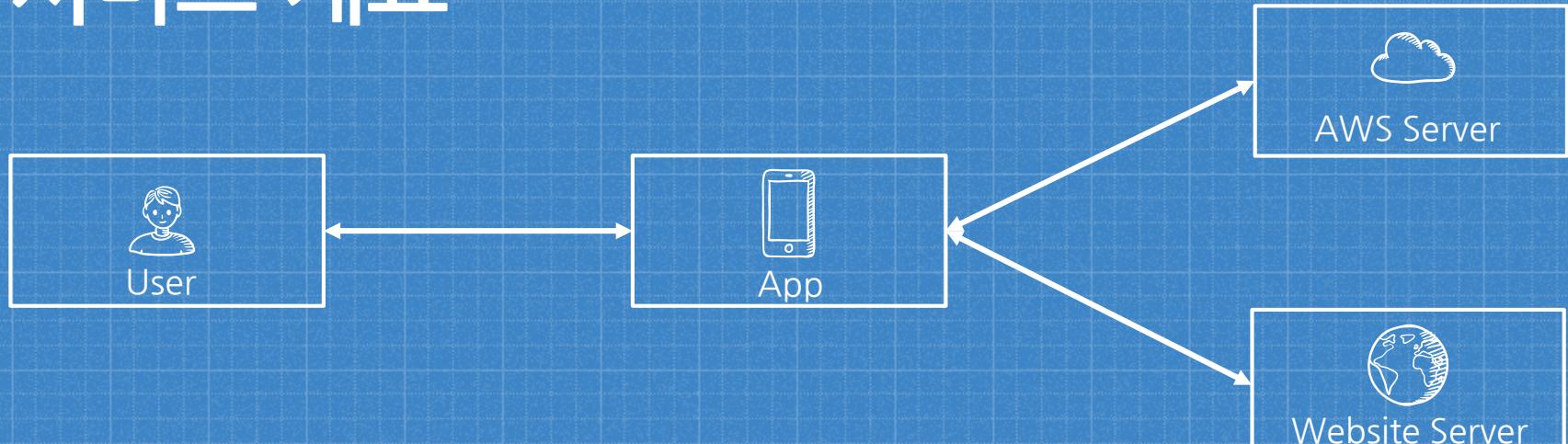
## 3. 정보의 단편화 필요

개발자 관련 웹사이트 안내 지침서가 필요

→ 다수의 카테고리 리스트화

→ 한 카테고리 내 다양한 웹사이트의 정보 시각화 및 접근성 용이

# 서비스 개요



User (사용자)

- 회원가입, 로그인
- 웹사이트 북마크 기능
- 북마크 기능 사용자화
- 콘텐츠 검색 및 필터링 기능



App (안드로이드 앱)

- 자체 계정 로그인 / 회원가입
- 최신 콘텐츠 Main에 표시
- 카테고리 리스트화 및 Navigation View에 표시
- 카테고리 북마크 후 북마크한 카테고리를 Fragment에 추가
- 카테고리 선택 후 콘텐츠 표시
- 콘텐츠 북마크 후 북마크한 콘텐츠 Main에 우선 표시

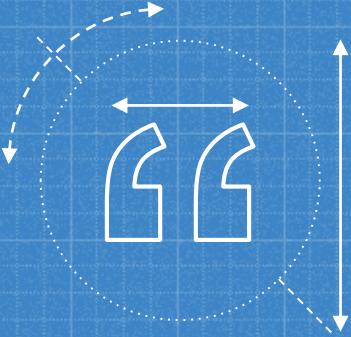


AWS Server (자체 서버)

- 자체 서버(AWS)에 사용자 등록 및 로그인
- 사용자 계정 내 북마크 정보를 통한 Fragment 화면 구성
- 전체 카테고리 정보를 통한 Navigation View 구축

Website Server (외부 서버)

- 외부 서버에서 웹사이트 콘텐츠 제공



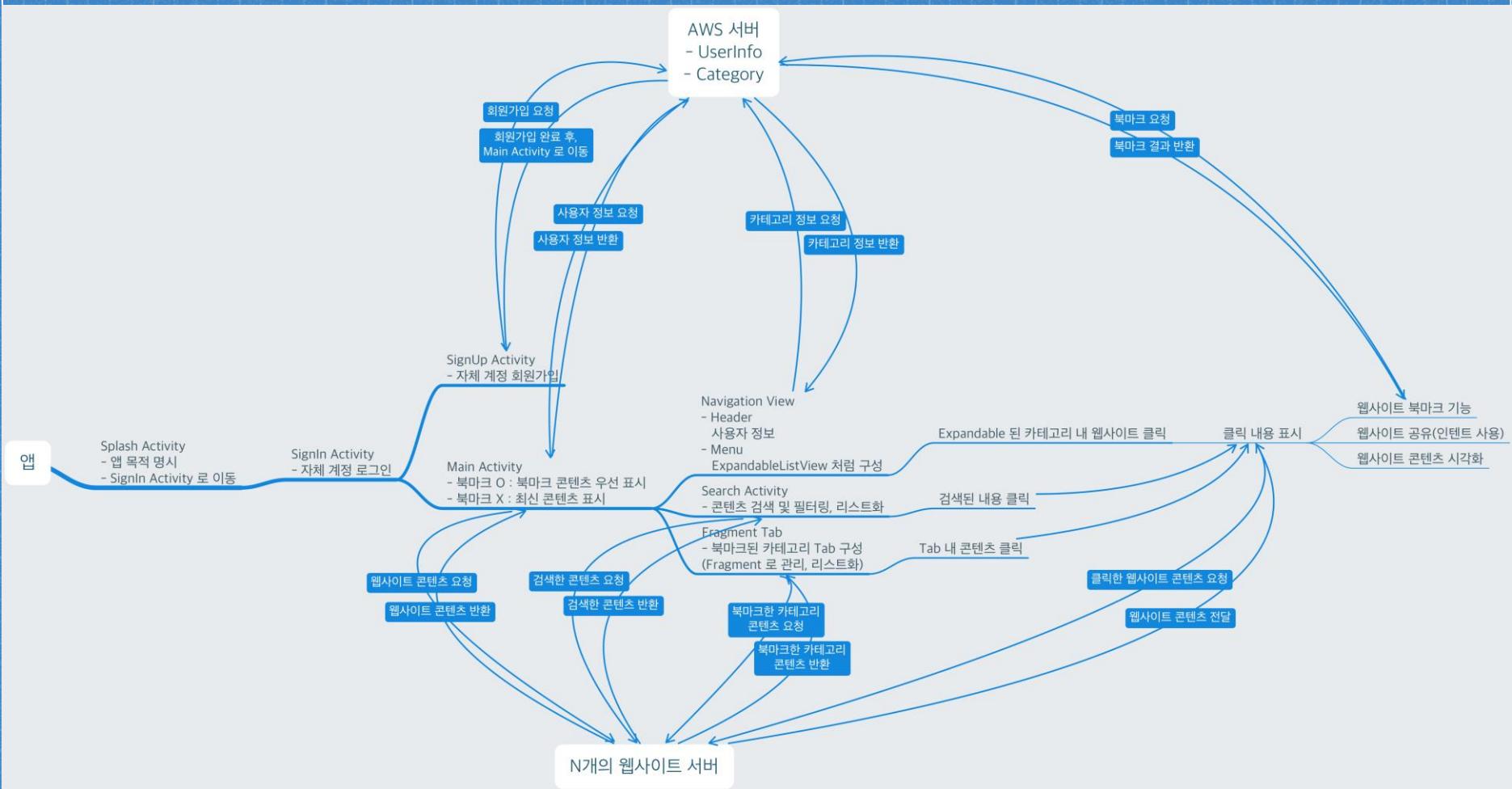
# 서비스 개발

# 개발 목표

- 개발 관련 정보가 담긴 웹사이트들을 한 곳에 모아 정리해서 접근성이 좋은 앱을 개발하자!
- 9XD, 온오프믹스, 이상한모임, GDG, 생활코딩, OKKY, 안드로이드사이드 등…



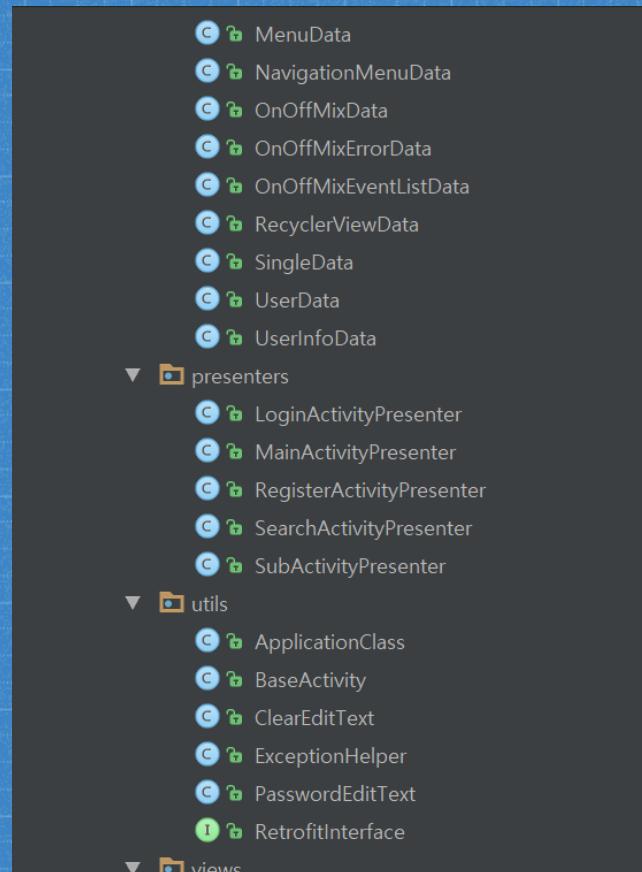
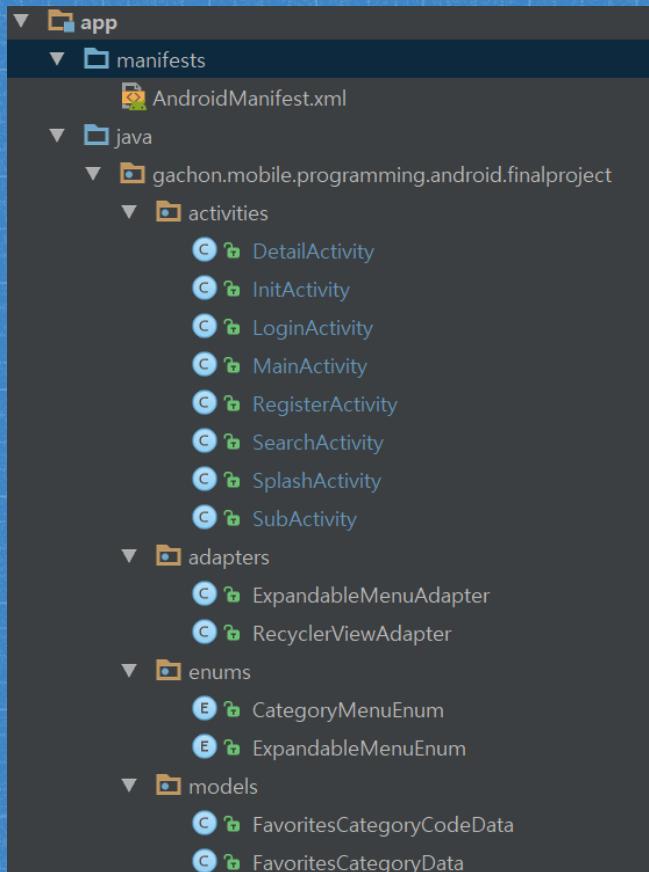
## 개발 내용



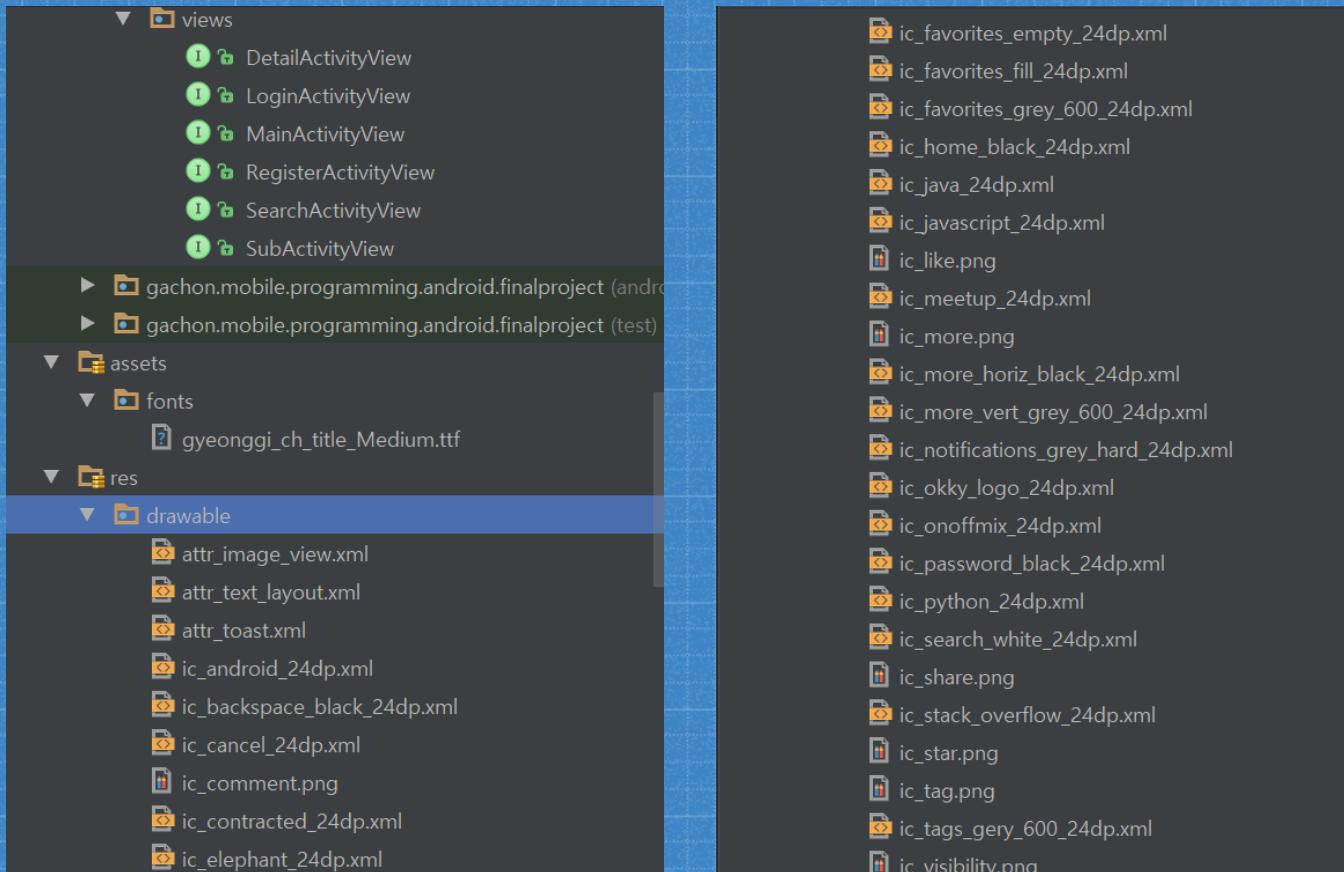
# 개발 전략

- 외부 서버 콘텐츠 파싱 구조 설계
  - (아래 사항 제외)
  - 자체 서버 구축
  - 외부 서버 콘텐츠 파싱 구조 설계
  - 안드로이드 구조 설계
  - Splash, Login 화면 구축 (서버 통신)
  - Main 화면 구축 (Fragment, Navigation View)
- 안드로이드 구조 설계
- 회원가입 기능 구축 (서버 통신)
- 검색 화면 기능 구축 (서버 통신)
- 콘텐츠 내부 화면 구축 (콘텐츠 공유 기능 및 내용 시각화)

# 개발 과정 - 안드로이드 구조 설계



# 개발 과정 - 안드로이드 구조 설계



# 개발 과정 - 회원가입 기능 (서버 통신)

LTE 89% 6:54

가입하기

Username  
영문, 숫자만 허용(4~20자)  
0 / 20

Email  
이메일 주소  
0 / 30

Password  
영문, 숫자, 특수문자(6자 이상)  
0 / 60

Password Confirm  
비밀번호 재확인  
0 / 60

Phone  
휴대폰 번호(-자 제외)

- 회원가입 화면은 사용자의 이름, 로그인, 이메일 주소, 비밀번호, 비밀번호 확인, 휴대폰 번호를 입력 받아 AWS-EC2 서버와 통신하여 회원가입을 하는 화면입니다.

# 개발 과정 - 회원가입 기능 (서버 통신)

```
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_register);

    final Toolbar toolbarRegister = (Toolbar) findViewById(R.id.toolbar_register);
    setSupportActionBar(toolbarRegister);

    final ActionBar actionBar = getSupportActionBar();
    if (actionBar != null) {
        actionBar.setDisplayShowTitleEnabled(false);
        actionBar.setDisplayHomeAsUpEnabled(true);
    }

    mRegisterActivityPresenter = new RegisterActivityPresenter(context: RegisterActivity.this, registerActivityView: this);

    mRegisterUsernameView = (ClearEditText) findViewById(R.id.username);
    final TextView usernameTextView = (TextView) findViewById(R.id.username_text_edited);
    mRegisterUsernameView.addTextChangedListener(new TextWatcher() {...});

    mRegisterEmailView = (ClearEditText) findViewById(R.id.email);
    final TextView emailTextView = (TextView) findViewById(R.id.email_text_edited);
    mRegisterEmailView.addTextChangedListener(new TextWatcher() {...});

    mRegisterPasswordView = (PasswordEditText) findViewById(R.id.password);
    final TextView passwordTextView = (TextView) findViewById(R.id.password_text_edited);

    mRegisterPasswordView.addTextChangedListener(new TextWatcher() {...});

    mRegisterPasswordConfirmView = (PasswordEditText) findViewById(R.id.password_confirm);
    final TextView passwordConfirmTextView = (TextView) findViewById(R.id.password_confirm_text_edited);

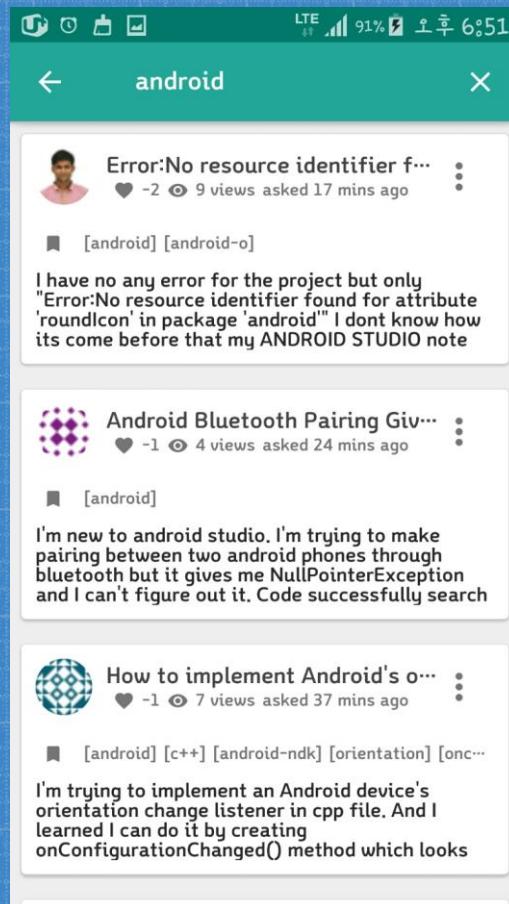
    mRegisterPasswordConfirmView.addTextChangedListener(new TextWatcher() {...});

    mRegisterPhoneView = (ClearEditText) findViewById(R.id.phone);
    final TextView phoneTextView = (TextView) findViewById(R.id.phone_text_edited);
    mRegisterPhoneView.setOnEditorActionListener((textView, id, keyEvent) -> {...});
    mRegisterPhoneView.addTextChangedListener(new TextWatcher() {...});

    // 회원가입 버튼을 통한 서버와의 통신
    findViewById(R.id.register_sign_up_button).setOnClickListener(view -> mRegisterActivityPresenter.attemptSignUp());
}
```

RegisterActivity.java

# 개발 과정 - 검색 화면 기능 (서버 통신)



- 검색 화면은 검색 메뉴에 입력한 값을 바탕으로 Stack Overflow, OKKY-Tech, OKKY-Q&A에 자료를 요청하고, 해당하는 내용을 RecyclerView를 사용해 리스트화 시켜주었습니다.

# 개발 과정 - 검색 화면 기능 (서버 통신)

```
// 검색을 통한 자료 불러오기 요청 작업
@Override
public boolean onCreateOptionsMenu(final Menu menu) {
    getMenuInflater().inflate(R.menu.menu_search, menu);
    final MenuItem searchItem = menu.findItem(R.id.action_search);
    SearchView searchView = (SearchView) MenuItemCompat.getActionView(searchItem);
    searchView.setQueryHint("검색어를 입력해 주세요.");
    searchView.setOnQueryTextListener(new SearchView.OnQueryTextListener() {
        @Override
        public boolean onQueryTextSubmit(final String searchValue) {
            mSearchActivityPresenter.refreshDisplay(searchValue);
            return false;
        }

        @Override
        public boolean onQueryTextChange(String searchValue) { return false; }
    });
    return true;
}
```

SearchActivity.java

# 개발 과정 - 검색 화면 기능 (서버 통신)

```
// 검색을 통한 자료 불러온것 화면에 처리 작업
@Override
public void setDisplayRecyclerView(final ArrayList<RecyclerViewData> recyclerViewdataArrayList) {
    final TextView textViewSearchContent = (TextView) findViewById(R.id.text_view_search_content);
    final RecyclerView recyclerView = (RecyclerView) findViewById(R.id.recycler_view_search);

    if (recyclerViewdataArrayList.size() == 0) {
        textViewSearchContent.setVisibility(View.VISIBLE);
        textViewSearchContent.setText("자료가 존재하지 않습니다.");
        recyclerView.setVisibility(View.GONE);
        return;
    } else {
        textViewSearchContent.setVisibility(View.GONE);
        recyclerView.setVisibility(View.VISIBLE);
    }

    recyclerView.setItemAnimator(new SlideInUpAnimator());
    recyclerView.setHasFixedSize(true);
    recyclerView.setLayoutManager(new LinearLayoutManager(getApplicationContext()));

    final RecyclerViewAdapter recyclerViewAdapter = new RecyclerViewAdapter(getApplicationContext(), recyclerViewdataArrayList);
    ScaleInAnimationAdapter scaleInAnimationAdapter = new ScaleInAnimationAdapter(recyclerViewAdapter);
    scaleInAnimationAdapter.setFirstOnly(true);
    scaleInAnimationAdapter.setDuration(500);
    scaleInAnimationAdapter.setInterpolator(new OvershootInterpolator( tension: 1f));
    recyclerView.setAdapter(scaleInAnimationAdapter);
}
```

SearchActivity.java

# 개발 과정 - 검색 화면 기능 (서버 통신)



- 검색어 입력 화면은 검색 메뉴를  
app:actionViewClass="android.support.v7.widget.Se  
archView"로 두어 검색 가능하게 설정하였습니다.

# 개발 과정 - 검색 화면 기능 (서버 통신)

```
<?xml version="1.0" encoding="utf-8"?>
<menu xmlns:android="http://schemas.android.com/apk/res/android"
      xmlns:app="http://schemas.android.com/apk/res-auto">
    <item
        android:id="@+id/action_search"
        android:icon="@drawable/ic_search_white_24dp"
        android:title="검색"
        app:showAsAction="always"
        app:actionViewClass="android.support.v7.widget.SearchView" />
</menu>
```

menu\_search.xml

# 개발 과정 - 콘텐츠 내부 화면 (내용 시각화)



- 세부 콘텐츠 내용 표시 화면은 세부 컨텐츠의 타이틀, 좋아요 회수, 조회수, 글등록시간, 태그들, 내용(텍스트, 이미지)을 표시해줍니다.

# 개발 과정 - 콘텐츠 내부 화면 (내용 시각화)

```
protected void onCreate(Bundle savedInstanceState) {  
    super.onCreate(savedInstanceState);  
    setContentView(R.layout.activity_detail);  
  
    final Toolbar toolbarLogin = (Toolbar) findViewById(R.id.toolbar_detail);  
    setSupportActionBar(toolbarLogin);  
  
    final ActionBar actionBar = getSupportActionBar();  
    if (actionBar != null) {  
        actionBar.setDisplayShowTitleEnabled(false);  
        actionBar.setDisplayHomeAsUpEnabled(true);  
    }  
    final TextView detailTitle = (TextView) findViewById(R.id.text_view_detail_title);  
  
    SharedPreferences sharedPreferences = getApplicationContext().getSharedPreferences(PREF_ID, Activity.MODE_PRIVATE);  
  
    final Intent selectedIntent = getIntent();  
    if (selectedIntent != null) {  
        mRecycler ViewData = selectedIntent.getParcelableExtra(name: "selectedData");  
        detailTitle.setText(mRecycler ViewData.getTitle());  
        setWebViewFromHtml(mRecycler ViewData.getContentUrl());  
    }  
}
```

DetailActivity.java

# 개발 과정 - 콘텐츠 내부 화면 (내용 시각화)

```
public void setWebViewFromHtml(final String baseUrl) {  
    final WebView htmlWebView = (WebView) findViewById(R.id.html_web_view);  
  
    final WebSettings webSettings = htmlWebView.getSettings();  
    // 자바스크립트 허용  
    webSettings.setJavaScriptEnabled(true);  
    webSettings.setDomStorageEnabled(true);  
    webSettings.setDatabaseEnabled(true);  
  
    // 캐시추가  
    final File cacheDir = getCacheDir();  
    if (!cacheDir.exists()) {  
        cacheDir.mkdirs();  
    }  
    webSettings.setAppCachePath(cacheDir.getPath());  
    webSettings.setAppCacheEnabled(true);  
  
    //배경색  
    htmlWebView.setBackgroundColor(0);  
    //웹뷰 컨텐츠 사이즈 맞추기  
    htmlWebView.getSettings().setLayoutAlgorithm(WebSettings.LayoutAlgorithm.SINGLE_COLUMN);  
    //Zoom 허용  
    htmlWebView.getSettings().setBuiltInZoomControls(true);  
    //Zoom 허용  
    htmlWebView.getSettings().setSupportZoom(true);  
    //웹뷰 확대된것 비율맞추기 %  
    htmlWebView.setInitialScale(1);  
    //meta 태그의 viewport 사용 가능  
    htmlWebView.getSettings().setLoadWithOverviewMode(true);  
    //html 스크린크기를 웹뷰에 맞춰줌  
    htmlWebView.getSettings().setUseWideViewPort(true);  
    //웹뷰의 뒤로가기  
    htmlWebView.goBack();  
  
    //콘텐츠 URL로 이동  
    htmlWebView.loadUrl(baseUrl);  
  
    // 현재 타겟 API가 17이므로 안써도됨.  
    /*if (Build.VERSION.SDK_INT < Build.VERSION_CODES.ICE_CREAM_SANDWICH) {  
        // 버전 4.0미하  
        htmlWebView.loadData(htmlSources, "text/html", "UTF-8");  
    } else {  
        // 버전 4.1미상  
        htmlWebView.loadData(htmlSources, "text/html; charset=UTF-8", null);  
    }*/
}
```

DetailActivity.java

# 개발 과정 - 콘텐츠 내부 화면 구축 (콘텐츠 공유 기능)



- 세부 콘텐츠 URL 복사/공유 선택 화면은 세부 컨텐츠의 URL을 클립보드에 복사해주는 기능과 공유하기 기능이 있습니다. 공유하기 기능은 앱의 이름과 사용자 이름, 세부 컨텐츠의 타이틀과 URL을 포함하여 외부 앱에 공유할 수 있도록 설정했습니다. (외부 앱 ex) 카카오톡, 페이스북, 트위터 등)

# 개발 과정 - 콘텐츠 내부 화면 구축 (콘텐츠 공유 기능)

```
public boolean onOptionsItemSelected(MenuItem item) {
    switch (item.getItemId()) {
        case android.R.id.home:
            finish();
            break;
        case R.id.action_content_favorites:
            final String email = mSharedPreferences.getString(key: "email", defaultValue: null);
            final String password = mSharedPreferences.getString(key: "password", defaultValue: null);
            final JSONObject jsonObject = new JSONObject();
            try {
                jsonObject.put(name: "email", value: email);
                jsonObject.put(name: "password", value: password);
                jsonObject.put(name: "content_title", value: mRecycler ViewData.getTitle());
                jsonObject.put(name: "content_url", value: mRecycler ViewData.getContentUrl());
                jsonObject.put(name: "content_tag", value: mRecycler ViewData.getTags());
                jsonObject.put(name: "content_summary", value: mRecycler ViewData.getContent());
                jsonObject.put(name: "content_watch_count", value: mRecycler ViewData.getWatchCount());
                jsonObject.put(name: "contentFavoritesCount", value: mRecycler ViewData.getFavoritesCount());
            } catch (final JSONException e) {
                DisplayCustomToast(getApplicationContext(), ExceptionHelper.getApplicationExceptionMessage(e));
                break;
            }
            final Observable<SingleData> storeCategoryRx = RETROFIT_INTERFACE.StoreContentRx(RequestBody.create(MEDIA_TYPE_JSON, jsonObject.toString()));
            storeCategoryRx.subscribeOn(Schedulers.io())
                .observeOn(AndroidSchedulers.mainThread())
                .subscribe(t -> DisplayCustomToast(getApplicationContext(), t.getData()),
                           e -> DisplayCustomToast(getApplicationContext(), ExceptionHelper.getApplicationExceptionMessage((Exception) e)));
            break;
        case R.id.action_copy:
            final ClipboardManager clipboardManager = (ClipboardManager) getApplicationContext().getSystemService(Context.CLIPBOARD_SERVICE);
            clipboardManager.setPrimaryClip(ClipData.newPlainText(label: "text", value: mRecycler ViewData.getContentUrl()));
            DisplayCustomToast(getApplicationContext(), "해당 게시물의 주소 복사에 성공하였습니다.");
            break;
        case R.id.action_share:
            final String userName = mSharedPreferences.getString(key: "name", defaultValue: null);

            final Intent intentForShare = new Intent(Intent.ACTION_SEND);
            intentForShare.putExtra(Intent.EXTRA_SUBJECT, "'개발다방'에서 " + userName + "님이 자료를 공유하였습니다." + "\n");
            intentForShare.putExtra(Intent.EXTRA_TEXT, "제목" + ":" + mRecycler ViewData.getTitle() + "\n\n" + mRecycler ViewData.getContentUrl());
            intentForShare.setType("text/plain");

            getApplicationContext().startActivity(Intent.createChooser(intentForShare, "공유").addFlags(Intent.FLAG_ACTIVITY_NEW_TASK));
            break;
    }
    return super.onOptionsItemSelected(item);
}
```

DetailActivity.java