



ㄱ ㄴ ㄷ ㄹ ㅁ ㅂ ㅅ ㅇ ㅈ ㅊ ㅋ ㅌ ㅍ ㅎ ㅣ ㅑ ㅓ ㅕ ㅗ ㅛ ㅜ ㅠ ㅡ ㅣ ㄱ ㄴ ㄷ ㄹ ㅁ ㅂ ㅅ ㅇ ㅈ ㅊ ㅋ ㅌ ㅍ ㅎ ㅣ ㅑ ㅓ ㅕ ㅗ ㅛ ㅜ ㅠ ㅡ

Y Z A B C D E F G H I J K L M N O P Q R S T U V W X Y Z A B C D E F G H I J K L M N O P Q R S

KoreanPocket

Check your Pocket right now!

IT학부
주혜원



목차

- 작품 목표
- 개발 환경
- 추진 일정
- 작품의 차별점
- 작품 내용 및 코드
- 작품 기대효과 및 향후 계획



작품 목표

- 한국어를 배우고자 하는 외국인들을 위한 ‘한국어 공부’ 안드로이드 앱
- 한국어의 가장 기본인 ‘한글’부터, 실생활에 많이 사용되는 어휘 및 문장을 위주로 소개하여 핵심적이고, 효율적인 공부 환경을 제공
- 41개의 다양한 주제, 1600개 이상의 단어/문장 데이터 보유(지속적으로 추가 업데이트)
- 한국어/영어/발음/품사/목소리를 동시에 제공하여 사용자들이 다방면으로 한국어를 배우기 적합합니다.
- 다양한 스터디 방식(1. Category, 2. Flip Card, 3. Screen Slide, 4. Game)
- 새로운 사용자에게도 쉽고 편리한 앱 조작을 위해 UI/UX를 고려한 심플하고, 깔끔한 디자인과 더불어 서비스를 사용자들에게 제공합니다.



개발 환경

- Android Studio 3.0.1
- Gradle version 4.1
- Compile Sdk Version API 26 : Android 8.0 (Oreo)

- 프로그래밍 언어 : JAVA, XML, SQL
- DB : Android SQLiteDatabase
- 디자인 툴 : Illustrator CS6



추진 일정

연	월	내용	
2017	11	Android Studio 및 프로그래밍 언어 습득 및 코딩 복습	
	12		
2018	1	프로젝트 아이디어 기획 및 레이아웃 설계	
	2		
	3	앱 기능 개발	
	4		
	5		앱 추가 기능 개발
	6	UI/UX 디자인 구성	디자인 및 기능 보완
	7	디버깅 및 테스트	
	8		앱 라이선스 및 구글 플레이스토어 앱 등록, 작품 제출



작품의 차별점

- 자체적인 DB 운영

- 단어 DB를 외부에서 가져오지 않고, 필요한 DB 직접 제작.

- 단어(데이터) 하나하나 입력 후 전체적으로 여러 번 검토하여 사용자에게 정확한 정보 서비스 전달.

- 실생활 밀착형 주제 및 내용

- 실생활에 자주 사용되는 단어/문장, 필수 어휘 등을 중심으로 제공하여, 핵심적인 공부 환경 제공.

- Google TTS(Text To Speech) API

- 타사 TTS 프로그램과 비교하여, 한국어 목소리 전달력이 가장 뚜렷하고, 확실한 Google TTS API 채택.

- 원인

- 동종 서비스 앱들을 분석해본 결과, 한국어에 대한 지식이 없는 개발자가 한국어 DB를 가져와 '한국어 배우기' 앱을 개발한 경우가 많음. 그 앱들을 직접 이용해 봤을 때, 영어를 한국어로 해석한 부분이 어색하거나 정확성과 전달력이 떨어지는 경우가 있음.



작품 내용 및 코드

- Splash 화면
- 로그인 (1. 구글 계정, 2. 이메일)
- 회원 가입
- 이메일 로그인
- 비밀번호 찾기
- 메인 화면
- Navigation Drawer
- My Page
- Grid Layout
 - Category
 - Flip Cards
 - Screen Slide
 - Game
- DB
- Category 단어 리스트
- Category 단어 뷰
- TTS API (Text To Speech)
- Flip Card 단어 뷰
- Screen Slide 단어 뷰
- SharedPreferences
- Game 단어 뷰
- My Score
- Share
- My Pocket
- 단어 뷰 Intent
- Search



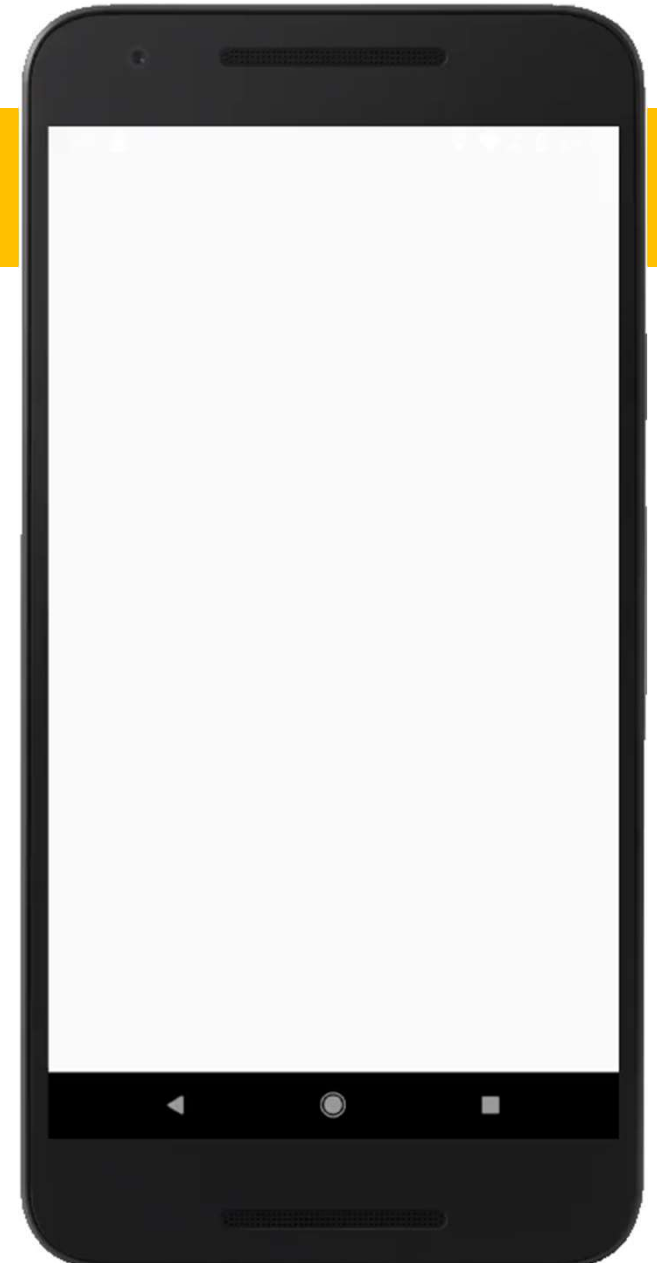
Splash 화면

XML

```
<LinearLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    tools:context="com.example.project1.LoadingActivity"
    android:background="@color/colorPrimary"
    android:gravity="center">

    <ImageView
        android:layout_width="90dp"
        android:layout_height="90dp"
        android:src="@mipmap/ic_koreanpocket"/>

</LinearLayout>
```



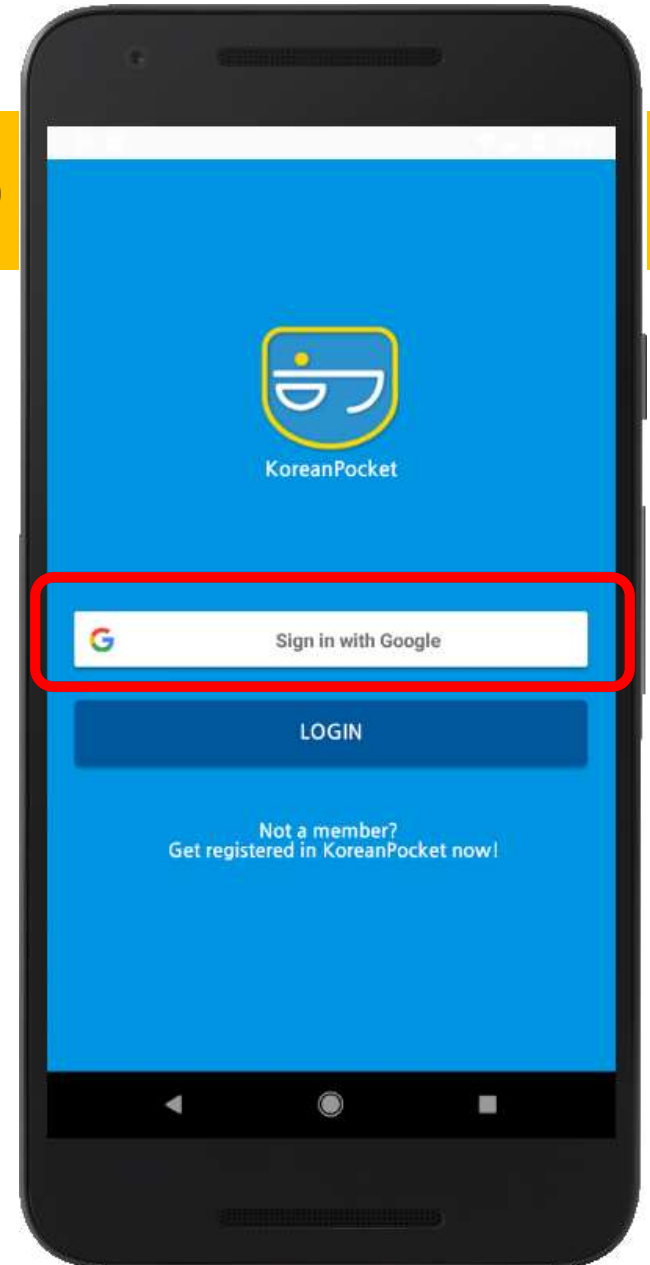


로그인 (1.구글 계정, 2.이메일)

1. 구글 계정

XML

```
<!-- Google Login Button -->  
<com.google.android.gms.common.SignInButton  
    android:id="@+id/login_with_google"  
    android:layout_width="match_parent"  
    android:layout_height="wrap_content" />
```



JAVA

```
public class LoginActivity extends BaseActivity implements GoogleApiClient.ConnectionCallbacks,
    GoogleApiClient.OnConnectionFailedListener, View.OnClickListener {

    private GoogleApiClient mGoogleApiClient;

    public void configureSignIn(){
        // 사용자의 기본 정보(이름, 이메일 etc.) 설정
        GoogleSignInOptions options = new GoogleSignInOptions.Builder(GoogleSignInOptions.DEFAULT_SIGN_IN)
            .requestIdToken(LoginActivity.this.getResources().getString(R.string.web_client_id))
            .requestEmail()
            .build();

        // GoogleSignIn.API 와 위의 옵션에 접근 허용하기 위한 GoogleApiClient 빌드
        mGoogleApiClient = new GoogleApiClient.Builder(mContext)
            .enableAutoManage(this /* FragmentActivity */, this /* OnConnectionFailedListener */)
            .addApi(Auth.GOOGLE_SIGN_IN_API, options)
            .build();
        mGoogleApiClient.connect();
    }
}
```

로그인 (1.구글 계정)

```
@Override
public void onClick(View view) {

    Utils utils = new Utils(this);
    int id = view.getId();

    if (id == R.id.Login_with_google){

        if (utils.isNetworkAvailable()){
            signIn();
        }else {
            Toast.makeText(LoginActivity.this, "Oops! no internet connection!",
                                                                    Toast.LENGTH_SHORT).show();
        }
    }
}

private void signIn() {
    Intent signInIntent = Auth.GoogleSignInApi.getSignInIntent(mGoogleApiClient);
    startActivityForResult(signInIntent, RC_SIGN_IN);
}
```

로그인 (1.구글 계정)

```
@Override
protected void onActivityResult(int requestCode, int resultCode, Intent data) {
    super.onActivityResult(requestCode, resultCode, data);
    if (requestCode == RC_SIGN_IN) {
        GoogleSignInResult result = Auth.GoogleSignInApi.getSignInResultFromIntent(data);
        if (result.isSuccess()) {
            // 구글 로그인 성공 시, 토큰 저장 후 인증
            GoogleSignInAccount account = result.getSignInAccount();

            idToken = account.getIdToken();
            name = account.getDisplayName();
            email = account.getEmail();
            photoUri = account.getPhotoUrl();
            photo = photoUri.toString();

            // SharedPreferences 에 데이터 저장
            sharedPrefManager = new SharedPreferences(mContext);
            sharedPrefManager.saveIsLoggedIn(mContext, true);

            sharedPrefManager.saveEmail(mContext, email);
            sharedPrefManager.saveName(mContext, name);
            sharedPrefManager.savePhoto(mContext, photo);

            sharedPrefManager.saveToken(mContext, idToken);

            AuthCredential credential = GoogleAuthProvider.getCredential(idToken, null);
            firebaseAuthWithGoogle(credential);
        } else {
            Log.e(TAG, "Login Unsuccessful. ");
        }
    }
}
```

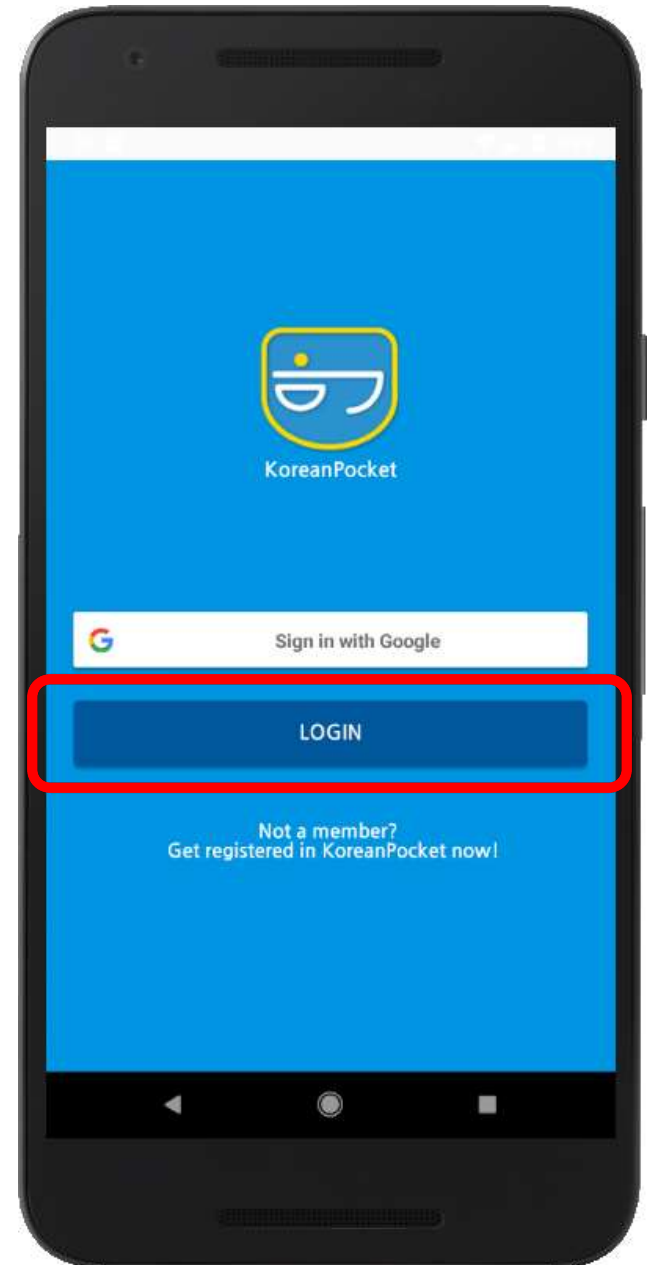
2. 이메일

XML

```
<Button
    android:id="@+id/first_login"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_marginLeft="4dp"
    android:layout_marginRight="4dp"
    android:layout_marginTop="20dp"
    android:background="@drawable/login_rec_button"
    android:text="@string/btn_login"
    android:textSize="15dp"
    android:textColor="@android:color/white" />
```

JAVA

```
// Login Button
first_login = (Button) findViewById(R.id.first_login);
first_login.setOnClickListener(new View.OnClickListener(){
    @Override
    public void onClick(View view) {
        Intent intent = new Intent(getApplicationContext(),
LoginBasicActivity.class);
        startActivity(intent);
    }
});
```

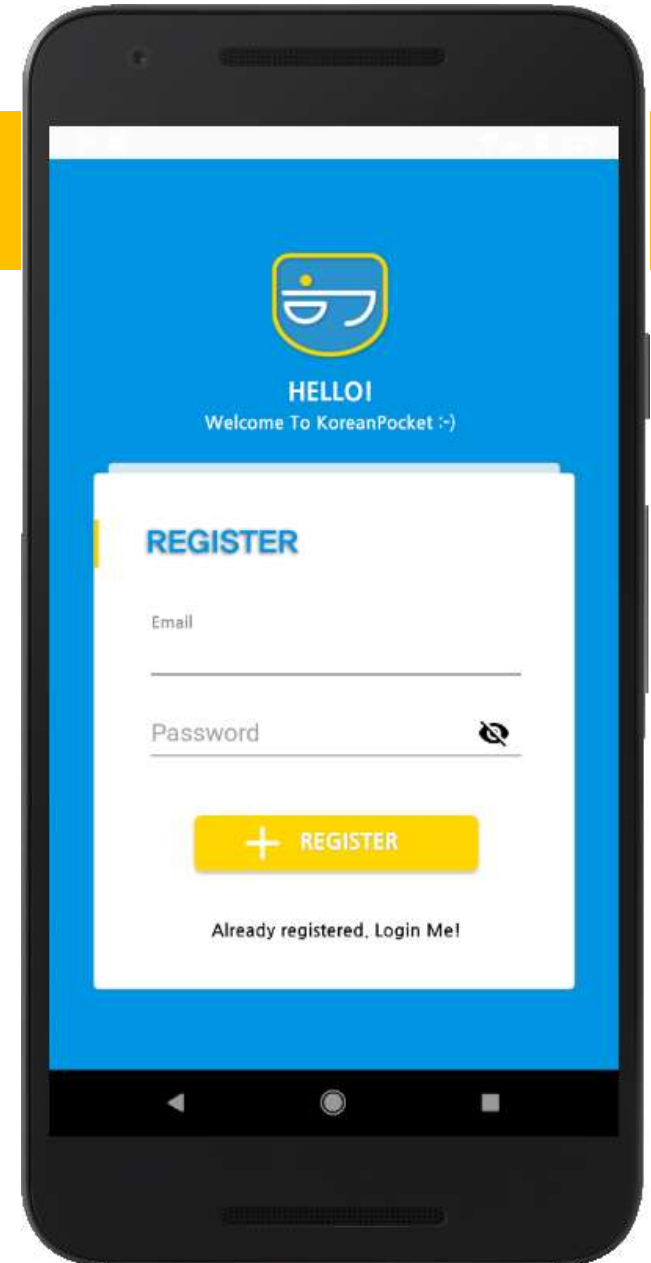




회원 가입

XML

```
<Button
    android:id="@+id/sign_up_button"
    android:layout_width="match_parent"
    android:layout_height="40dp"
    android:layout_gravity="center"
    android:gravity="center"
    android:layout_marginLeft="50dp"
    android:layout_marginRight="50dp"
    android:layout_marginTop="35dp"
    android:paddingLeft="30dp"
    android:drawablePadding="-50dp"
    android:drawableLeft="@drawable/register_rec_cross2"
    android:background="@drawable/register_rec_button"
    android:text="@string/action_sign_in_short"
    android:textColor="@android:color/white"
    android:textSize="16dp"
    android:shadowColor="@color/colorGrayText"
    android:shadowDx="1"
    android:shadowDy="1"
    android:shadowRadius="2"
    android:textStyle="bold"/>
```



JAVA

```
public class SignupActivity extends AppCompatActivity {

    private EditText inputEmail, inputPassword;
    private Button btnSignIn, btnSignUp, btnResetPassword;
    private FirebaseAuth auth;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_signup);

        //Get Firebase auth instance
        auth = FirebaseAuth.getInstance();

        btnSignUp.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {

                String email = inputEmail.getText().toString().trim();
                String password = inputPassword.getText().toString().trim();
```

회원가입

// 사용자 생성

```
auth.createUserWithEmailAndPassword(email, password)
    .addOnCompleteListener(SignupActivity.this, new OnCompleteListener<AuthResult>() {

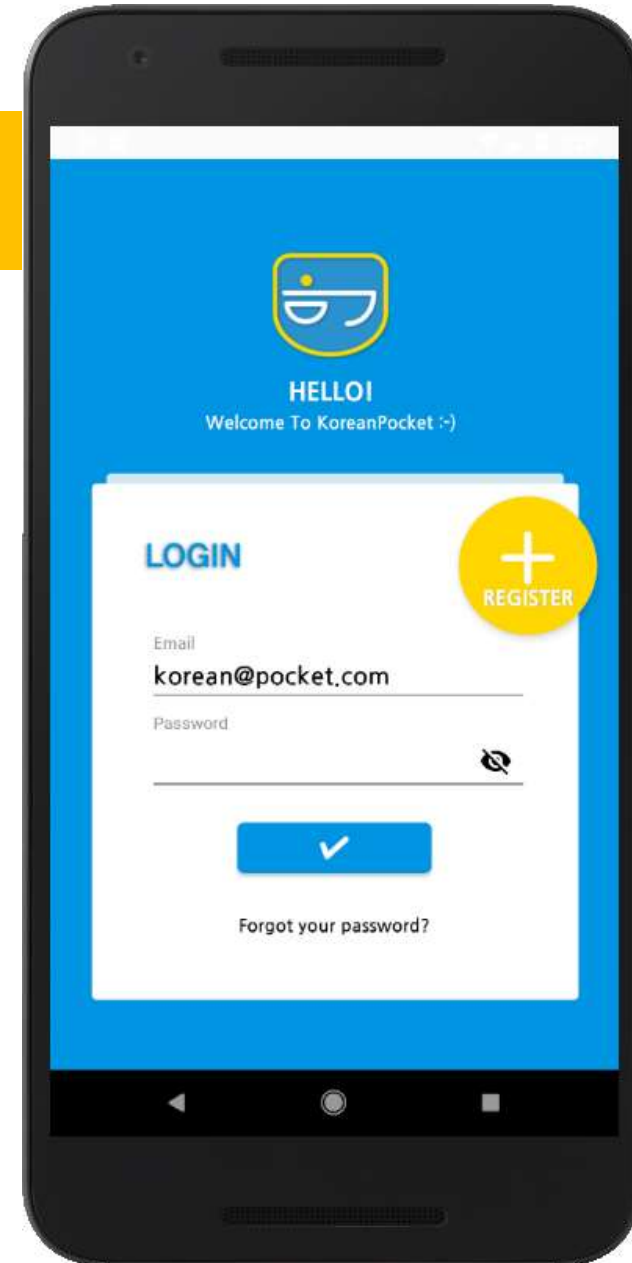
        @Override
        public void onComplete(@NonNull Task<AuthResult> task) {
            progressBar.setVisibility(View.GONE);

            if (!task.isSuccessful()) {
                Toast.makeText(SignupActivity.this, getString(R.string.appropriate_email_format),
                    Toast.LENGTH_SHORT).show();
            } else {
                startActivity(new Intent(SignupActivity.this, MainActivity.class));
                finish();
            }
        }
    });
});
```




이메일 로그인

```
btnLogin.setOnClickListener(new View.OnClickListener() {  
    @Override  
    public void onClick(View v) {  
        String email = inputEmail.getText().toString();  
        final String password = inputPassword.getText().toString();  
  
        // 사용자 인증 절차  
        auth.signInWithEmailAndPassword(email, password)  
            .addOnCompleteListener(LoginBasicActivity.this, new OnCompleteListener<AuthResult>()  
{  
            @Override  
            public void onComplete(@NonNull Task<AuthResult> task) {  
                if (!task.isSuccessful()) {  
                    // there was an error  
                    if (password.length() < 6) {  
                        Toast.makeText(LoginBasicActivity.this,  
                            getString(R.string.minimum_password), Toast.LENGTH_LONG).show();  
                    } else {  
                        Toast.makeText(LoginBasicActivity.this,  
                            getString(R.string.auth_failed), Toast.LENGTH_LONG).show();  
                    }  
                } else {  
                    Intent intent = new Intent(LoginBasicActivity.this,  
                                                MainActivity.class);  
                    startActivity(intent);  
                    finish();  
                }  
            }  
        }  
    }  
});
```





비밀번호 찾기

Reset your password for KoreanPocket



Inbox x



noreply@project1-200601.firebaseio.com

to me ▾

Hello!

Forgot your password? No problem!

Follow this link to reset your KoreanPocket password for your joo7875@gmail.com account.

https://project1-200601.firebaseio.com/_/auth/action?mode=resetPassword&oobCode=J6lgYZG6Jc00py5EBtrk0&lang=en

If you didn't ask to reset your password, you can ignore this email.

We hope you're enjoying the KoreanPocket experience!

Thanks,

Your KoreanPocket team

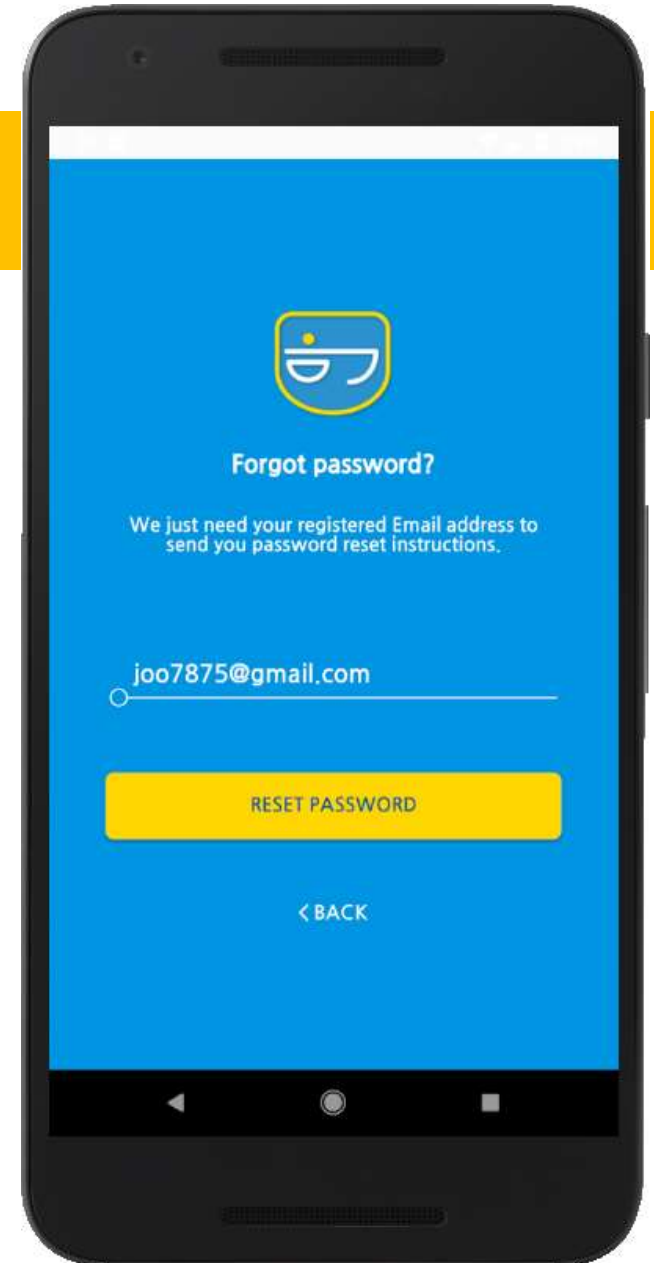


Forgot password?

We just need your registered Email address to send you password reset instructions.

RESET PASSWORD

< BACK





메인 화면

JAVA

```
long date = System.currentTimeMillis();
SimpleDateFormat sdf = new SimpleDateFormat("dd MMMM, yyyy");
// "MMM dd MM, yyyy h:mm a"
String dateString = sdf.format(date);
mainDate.setText(dateString);
```

// SharedPreferences 에 저장된 사용자 데이터를 가져옴

```
sharedPrefManager = new SharedPreferences(mContext);
```

```
mUsername = sharedPrefManager.getName();
```

```
mEmail = sharedPrefManager.getUserEmail();
```

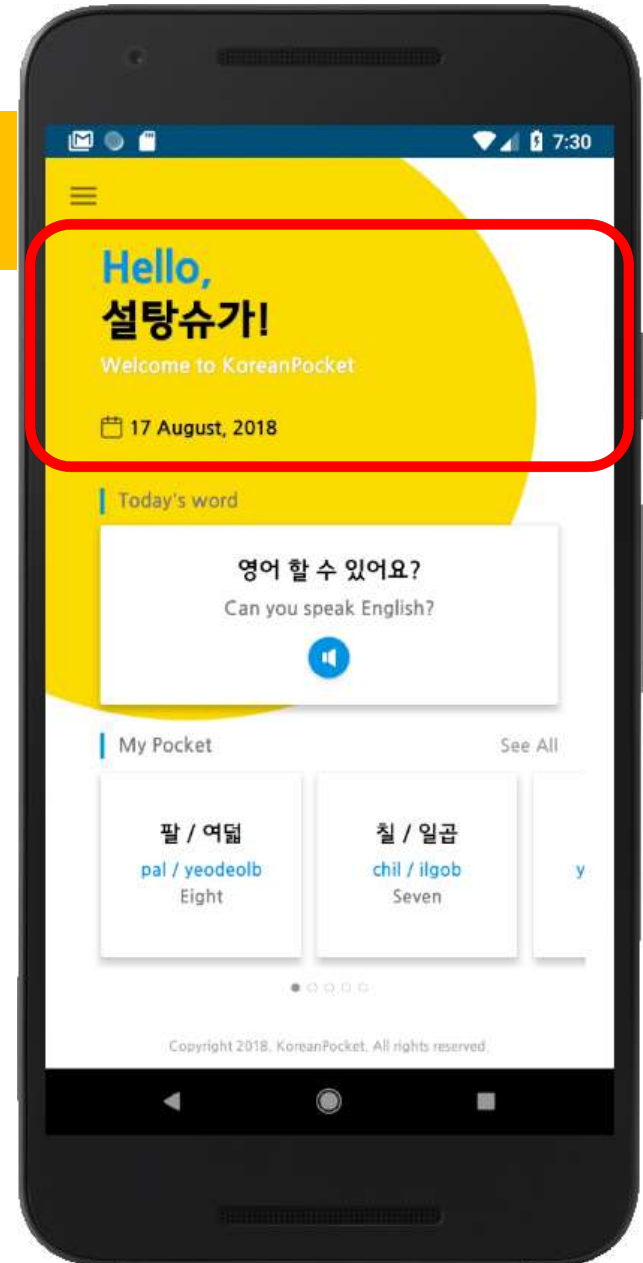
```
String uri = sharedPrefManager.getPhoto();
```

```
if(uri != null) {
    mPhotoUri = Uri.parse(uri);
}
```

```
mFullNameTextView.setText(mUsername);
```

```
mEmailTextView.setText(mEmail);
```

```
mainNickname.setText(mUsername + "!");
```



```

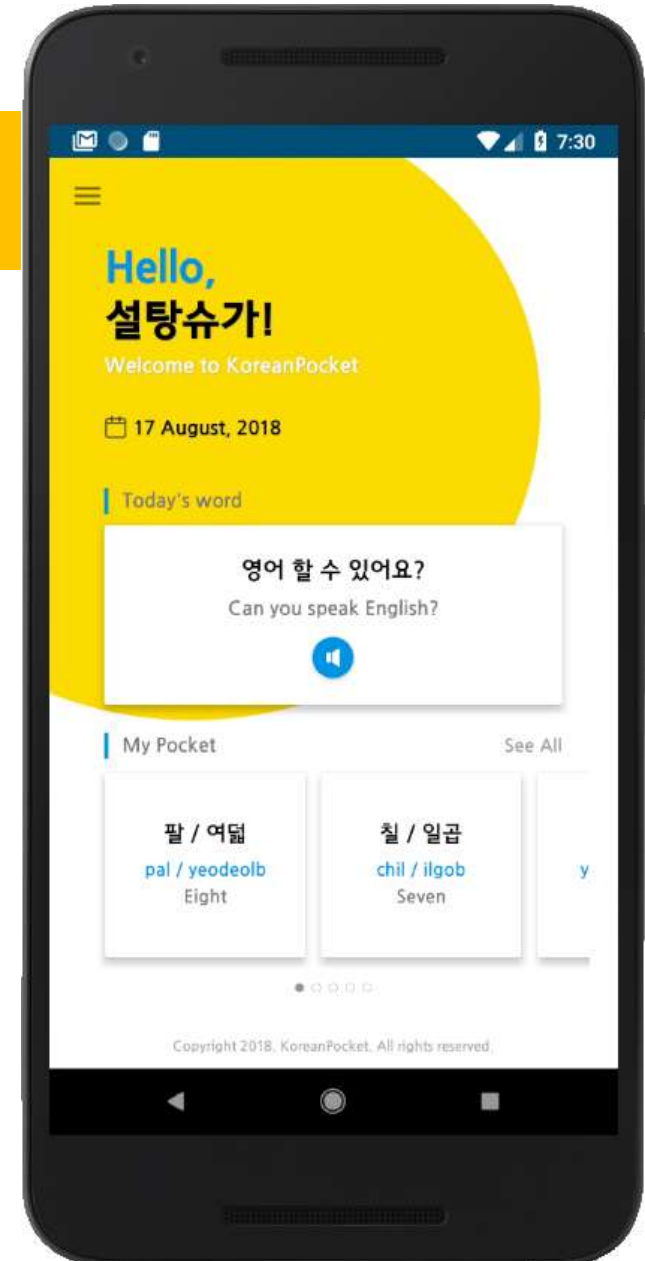
@SuppressLint("SetTextI18n")
private void setDataToView(FirebaseUser user) {

    str_name = "";
    str_email = user.getEmail();

    for(int i = 0; i < user.getEmail().indexOf('@'); i++) {
        char char_name = user.getEmail().charAt(i);
        str_name += Character.toString(char_name);
    }

    if(user.getDisplayName() == null) {
        mFullNameTextView.setText(str_name);
        mainNickname.setText(str_name + "!");
    }
    mEmailTextView.setText(user.getEmail());
}

```

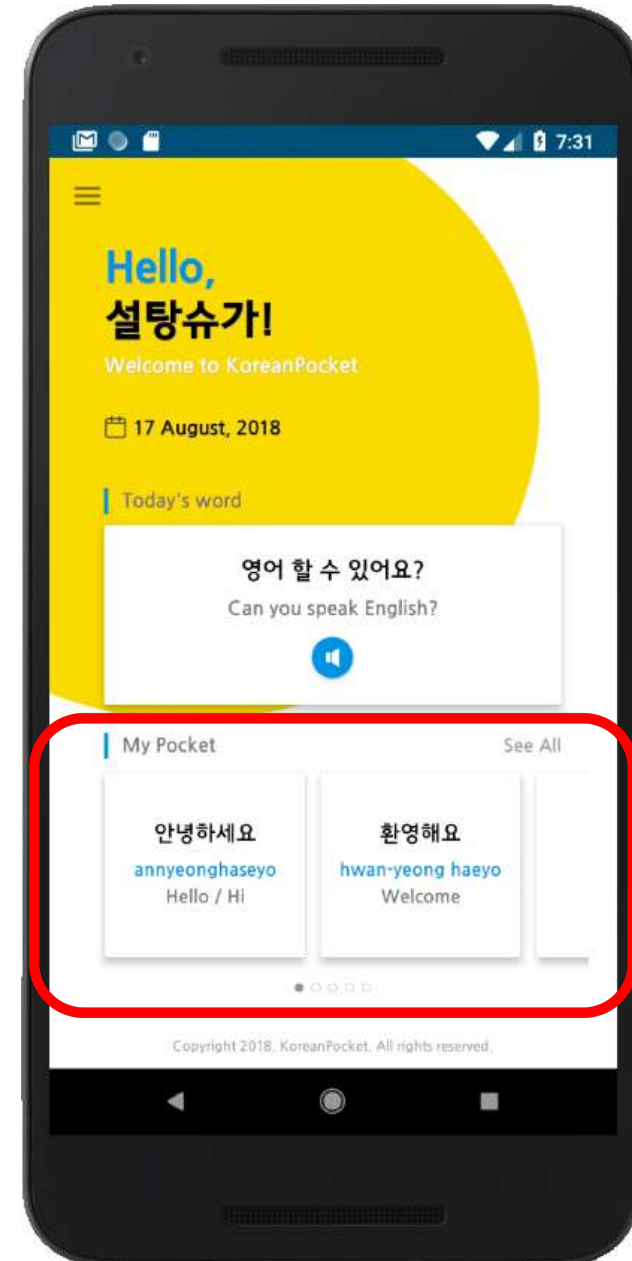


```
// My Pocket Card
sharedPreference = new SharedPreferences();
mViewPager = (ViewPager) findViewById(R.id.viewPager);
List<Item> favor = sharedPreference.getFavorites(this);
mCardAdapter = new CardPagerAdapter();

if(favor.size() != 0){
    for(int i = 0; i < favor.size(); i++){
        mCardAdapter.addCardItem(new Item(favor.get(i).getKor(),
            favor.get(i).getPro(), favor.get(i).getEng()));
    }
} else {
    mCardAdapter.addCardItem(new Item("안녕하세요", "annyeonghaseyo", "Hello / Hi"));
    mCardAdapter.addCardItem(new Item("환영해요", "hwan-yeong haeyo", "Welcome"));
    mCardAdapter.addCardItem(new Item("이 / 둘", "i / dul", "2"));
    mCardAdapter.addCardItem(new Item("한국어", "hangug-eo", "Korean"));
    mCardAdapter.addCardItem(new Item("호주머니", "hojumeoni", "Pocket"));
}

mViewPager.setAdapter(mCardAdapter);
mViewPager.setOffscreenPageLimit(3);

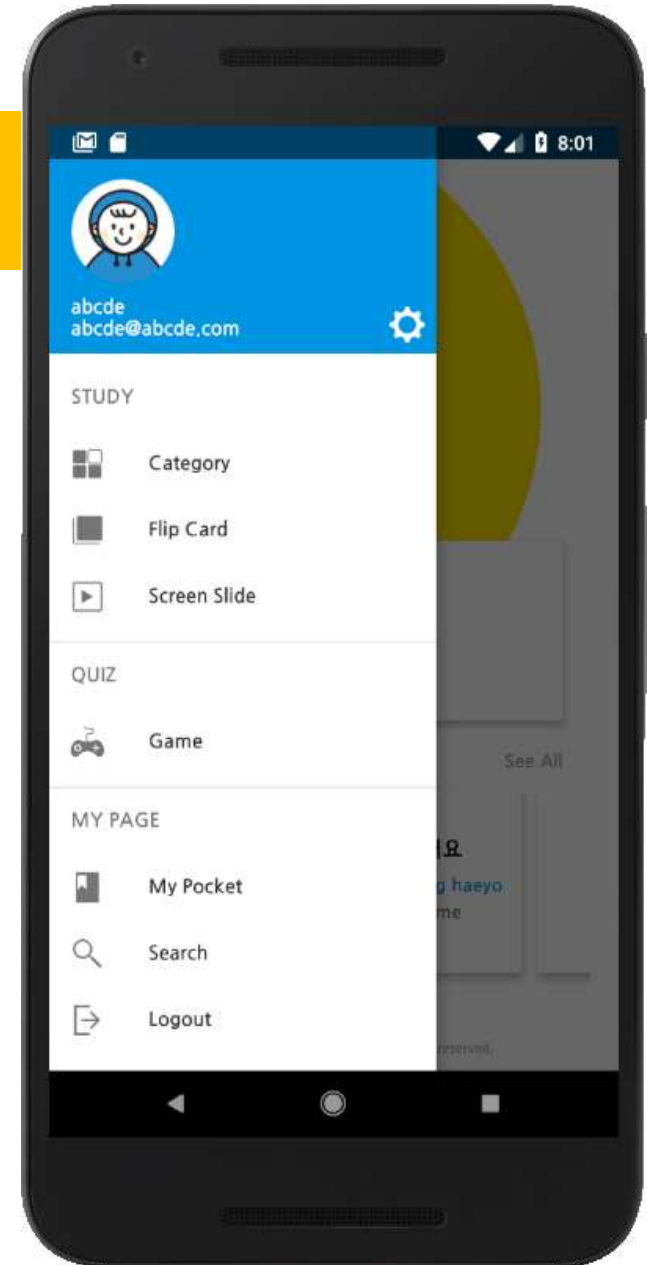
CircleIndicator circleIndicator = (CircleIndicator)
findViewById(R.id.circle_indicator);
circleIndicator.setupWithViewPager(mViewPager);
```





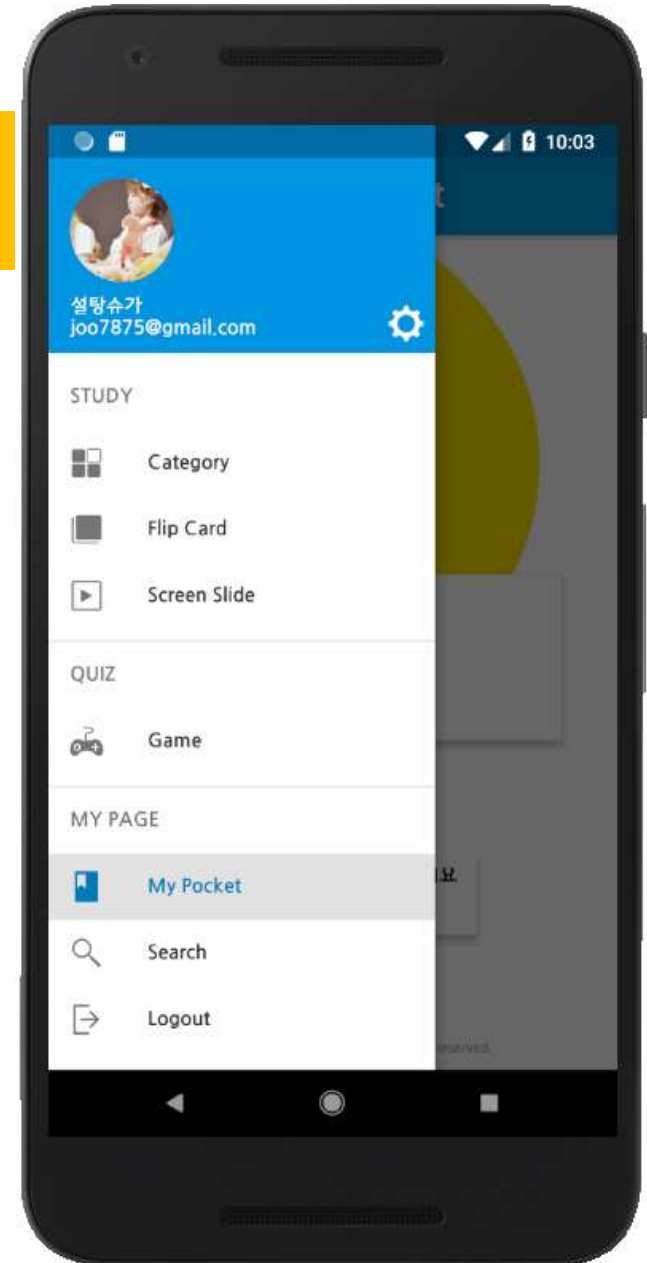
Navigation Drawer

- **Categories** 주제별, 단어/문장별로 선택할 수 있고, 특정 단어 또는 문장을 찾고 싶을 때 유용.
- **Flip Cards** Flip! 버튼을 클릭하면, 앞면(한국어/품사), 뒷면(한국어/영어/품사/발음)을 뒤집거나 슬라이드 하며 공부할 수 있는 단어장 기능
- **Screen Slide** 카드(한국어/영어/발음/품사)를 슬라이드하며 공부할 수 있는 단어 암기장 형식
- **Game** 주제별 4지선다형 게임. 틀릴 경우 대화상자로 정답 표기. 문제를 모두 다 풀거나 중간에 QUIT 버튼을 클릭 시, 게임 점수 나타남.
- **My Pocket** 단어마다 있는 별(즐거찾기) 버튼을 탭 할 경우, 그 단어들은 My Pocket에 저장됨. 사용 자가 즐겨찾기한 단어들을 복습할 수 있음.
- **Search** 단어 혹은 문장 검색 기능



XML

```
<android.support.design.widget.NavigationView
    android:id="@+id/nav_view"
    android:layout_width="wrap_content"
    android:layout_height="match_parent"
    android:layout_gravity="start"
    android:fitsSystemWindows="true"
    android:background="@color/white"
    app:headerLayout="@layout/nav_header_main"
    app:menu="@menu/activity_main_drawer"
    app:itemIconTint="@drawable/nav_item_icon_color"
    app:itemTextColor="@drawable/nav_item_text_color"
    app:itemBackground="@drawable/nav_item_background" />
```



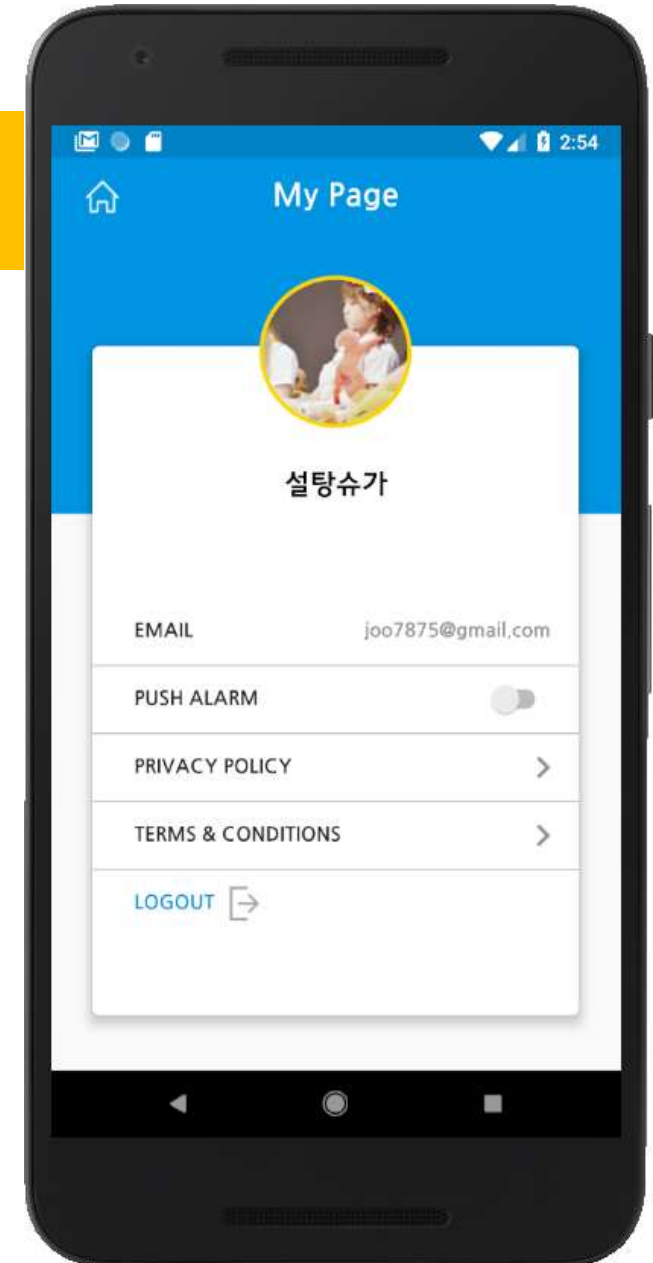
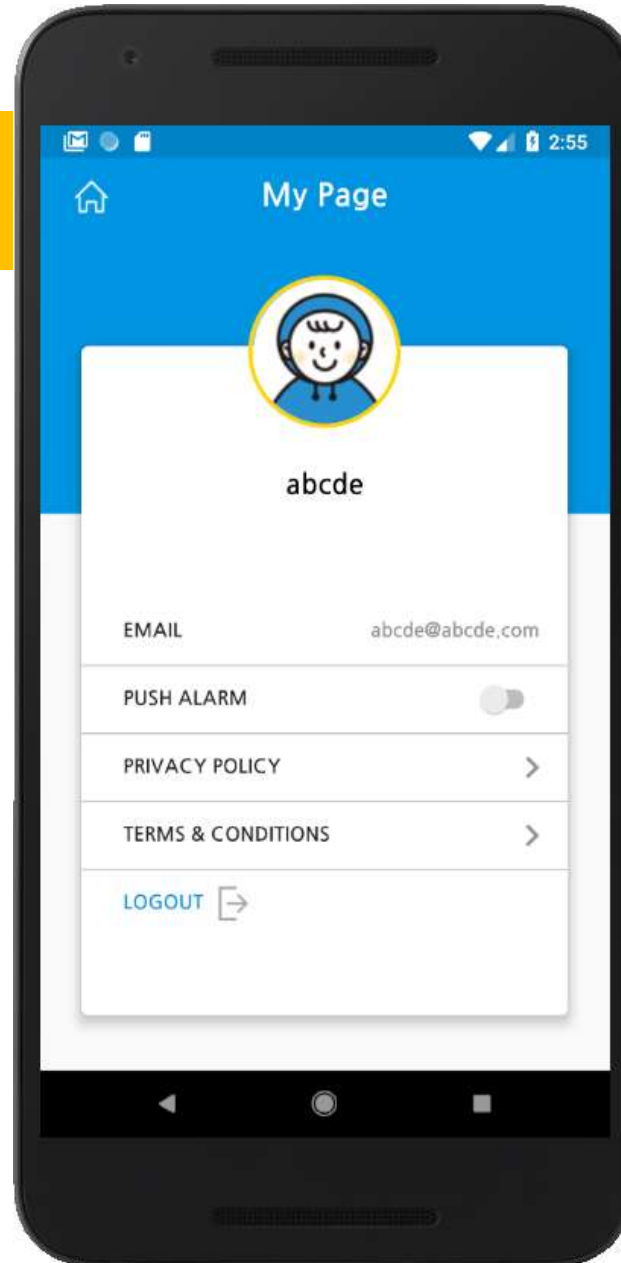
JAVA

```
navigationView.setOnItemClickListener(new NavigationView.OnItemClickListener() {  
    @Override  
    public boolean onItemClick(MenuItem item) {  
        switch (item.getItemId()) {  
            case R.id.nav_categories:  
                startActivity(new Intent(getApplicationContext(), Grid_Category.class));  
                break;  
            case R.id.nav_flipcards:  
                startActivity(new Intent(getApplicationContext(), Grid_Flipcard.class));  
                break;  
            case R.id.nav_screenslide:  
                startActivity(new Intent(getApplicationContext(), Grid_Screenslide.class));  
                break;  
            case R.id.nav_game1:  
                startActivity(new Intent(getApplicationContext(), Grid_Game.class));  
                break;  
            case R.id.nav_mystudy:  
                startActivity(new Intent(getApplicationContext(), MyStudy.class));  
                break;  
            case R.id.nav_search:  
                startActivity(new Intent(getApplicationContext(), Search.class));  
                break;  
            case R.id.nav_logout:  
                signOut();  
                drawerLayout.closeDrawers();  
                break;  
        }  
    }  
});
```




My Page

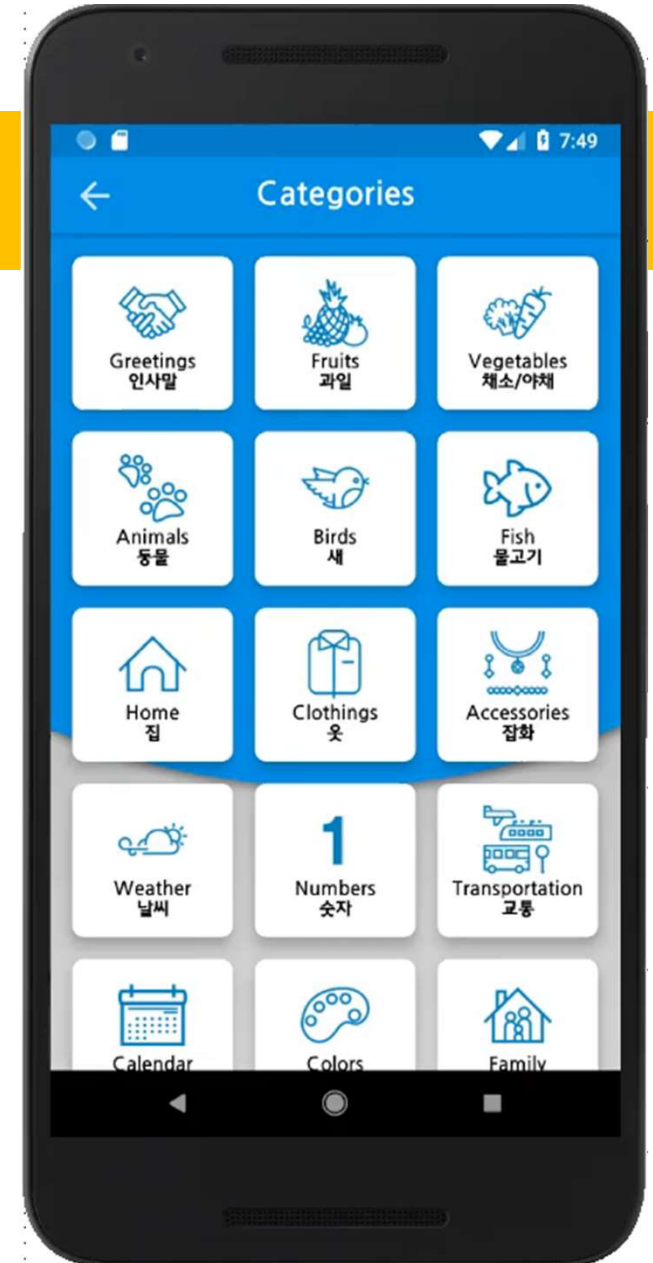
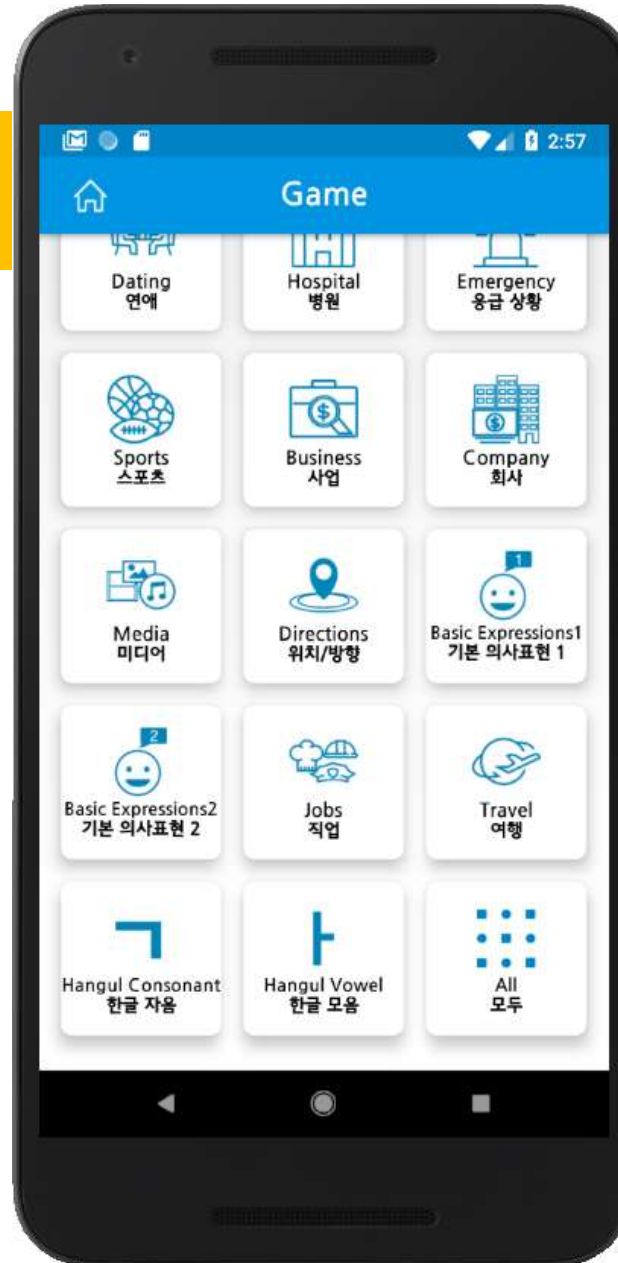
- 이메일
- 푸시 알림
- 프라이버시 정책
- 서비스 약관
- **로그아웃**





Grid Layout

- Category
- Flip Card
- Screen Slide
- Game



XML

```
<ScrollView xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context="com.example.project1.Grid_Category">
```

```
<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="@drawable/bg_gridlayout"
    android:orientation="vertical">
```

```
<GridLayout
    android:id="@+id/mainGrid"
    android:layout_width="match_parent"
    android:layout_height="0dp"
    android:layout_weight="1"
    android:alignmentMode="alignMargins"
    android:rowCount="14"
    android:columnCount="3"
    android:columnOrderPreserved="false"
    android:padding="0dp">
```

```
<!-- Row 1 -->
<!-- Column 1_1 -->
<android.support.v7.widget.CardView
    android:layout_width="0dp"
    android:layout_height="0dp"
    android:layout_columnWeight="1"
    android:layout_marginBottom="8dp"
    android:layout_marginLeft="16dp"
    android:layout_marginRight="8dp"
    android:layout_marginTop="16dp"
    android:layout_rowWeight="1"
    app:cardCornerRadius="8dp"
    app:cardElevation="8dp">
    android:textStyle="normal" />
```

JAVA

```
@SuppressWarnings("ClickableViewAccessibility")
private void setSingleEvent(GridLayout mainGrid) {
    for (int i = 0; i < mainGrid.getChildCount(); i++) {
        final CardView m_cardView = (CardView) mainGrid.getChildAt(i);
        final int finalI = i;

        m_cardView.setOnTouchListener(new View.OnTouchListener() {
            @Override
            public boolean onTouch(View view, MotionEvent motionEvent) {
                switch (motionEvent.getAction()){
                    case ACTION_DOWN:
                        m_cardView.setCardBackgroundColor(Color.parseColor("#E1F5FE"));

                        switch(finalI){
                            case 0:
                                startActivity(new Intent(getApplicationContext(), Categories_wordview1.class));
                                break;
                                ...
                        }
                        break;
                    case ACTION_UP:
                        m_cardView.setCardBackgroundColor(Color.parseColor("#FFFFFF"));
                        break;
                }
                return true;
            }
        });
    }
}
```



DB

CATEGORY	DATA TYPE	PK	FK	JOIN	CATEGORY(KOR)
category_id	integer	TRUE			카테고리ID
category_type	text				카테고리_종류

WORD	DATA TYPE	PK	FK	JOIN	WORD(KOR)
word_id	integer	TRUE			단어ID
category_id_fk	integer		TRUE	category. category_id	카테고리ID
word_kor	text				단어_한국어
word_eng	text				단어_영어
word_pro	text				단어_발음
word_gram	text				단어_품사

JAVA

```
public class DB_Category extends SQLiteOpenHelper {
    public DB_Category(Context context){
        super(context, "category.db", null, 1);
    }

    @Override
    public void onCreate(SQLiteDatabase db) {
        db.execSQL("create table if not exists category (category_id integer primary key, category_type text not null);");
    }
}

public class DB_Word extends SQLiteOpenHelper {
    public DB_Word(Context context){
        super(context, "word.db", null, 1);
    }

    public void onCreate(SQLiteDatabase db) {
        db.execSQL("create table if not exists word (word_id integer primary key, category_id_fk integer, word_kor text, word_eng text, word_pro text, word_gram text, foreign key(category_id_fk) references category(category_id));");
    }
}
```



word_db.csv

```
1 |1,1,안녕하세요,Hi / Hello,annyeonghaseyo,,인사말
2 2,1,잘 지냈어요?,How are you?,jarl geeness-eoyo?,
3 3,1,저는 잘 지냈어요,I am fine. Thanks!,jeoneun jal jinaess-eoyo,,
4 4,1,저는 잘 못 지냈어요,I am not well. Thanks!,jeoneun jarl mot geenaess-eoyo,,
5 5,1,만나서 반갑습니다,Nice to meet you,mannaseo bangabseubnida,,
6 6,1,오랜만이에요,Long time no see,oraenman-ieyo,,
7 7,1,좋은 아침입니다,Good morning,joh-eun achim-ibnida,,
8 8,1,좋은 밤 되세요,Have a Good night,joh-eun bam doeseyo,,
9 9,1,몇 살 이세요?,How old are you?,myeot sal iseyo?,
10 10,1,저는 - 살 입니다,I am - years old,jeoneun - sal ibnida,,
11 11,1,저는 - 에서 왔어요,I am from -,jeoneun - eseo wass-eoyo,,
12 12,1,저는 한국말을 조금 할 수 있어요,I can speak Korean a little bit,jeoneun hangukmal-eul jogeum hal su iss-eoyo,,
13 13,1,저는 한국말을 잘 못해요,I cannot speak Korean very well,jeoneun hangugmal-eul jal mothaeyo,,
14 14,1,안녕히가세요 / 잘가요,Good bye / bye,annyeonghi gaseyo / jalgayo,,
15 15,1,또 봐요,See you again,tto bwayo,,
16 16,1,곧 봐요,See you soon,got bwayo,,
17 17,1,이름이 뭐예요?,What is your name?,ireum-i mwo-eyo?,
18 18,1,저는 - 입니다,I am -,jeoneun - imbnida,,
19 19,1,저는 영어 할 수 있어요,I can speak English,jeoneun yeong-eo hal su iss-eoyo,,
20 20,1,영어 할 수 있어요?,Can you speak English?,yeong-eo harl su iss-eoyo?,
21 21,1,감사합니다,Thank you,gamsahabnida,,
22 22,1,정말 감사합니다,Thank you so much,jeongmal gamsahabnida,,
23 23,2,과일,Fruits,gwa-il,noun,과일
24 24,2,사과,Apple,sa-gwa,noun,
25 25,2,딸기,Strawberry,ttagi,noun,
26 26,2,포도,Grape,podo,noun,
27 27,2,수박,Watermelon,subak,noun,
28 28,2,바나나,Banana,banana,noun,
29 29,2,망고,Mango,mango,noun,
30 30,2,복숭아,Peach,boksung-a,noun,
```



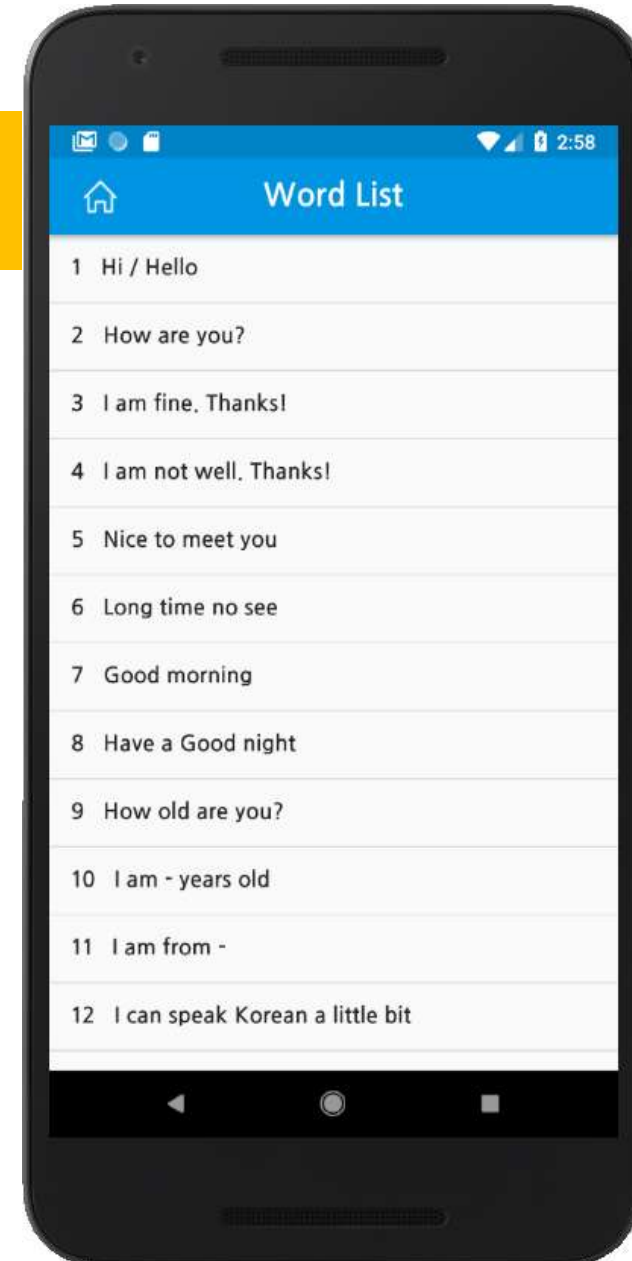
Category 단어 리스트

JAVA

```
public void select() {
    try {
        db = dbc.getReadableDatabase();
        sql = "select word_id, word_kor, word_eng from word where category_id_fk = 1;";
        cursor = db.rawQuery(sql, null);
        int count = cursor.getCount();
        result = new String[count];

        for (int i = 0; i < count; i++) {
            cursor.moveToNext();
            String str_id = cursor.getString(0);
            String str_kor = cursor.getString(1);
            String str_eng = cursor.getString(2);
            result[i] = i+1 + " " + str_eng;
        }

        Adapter = new ArrayAdapter<String>(this, android.R.layout.simple_list_item_1, result);
        wordlist.setAdapter(Adapter);
        wordlist.setChoiceMode(ListView.CHOICE_MODE_SINGLE);
    }
}
```

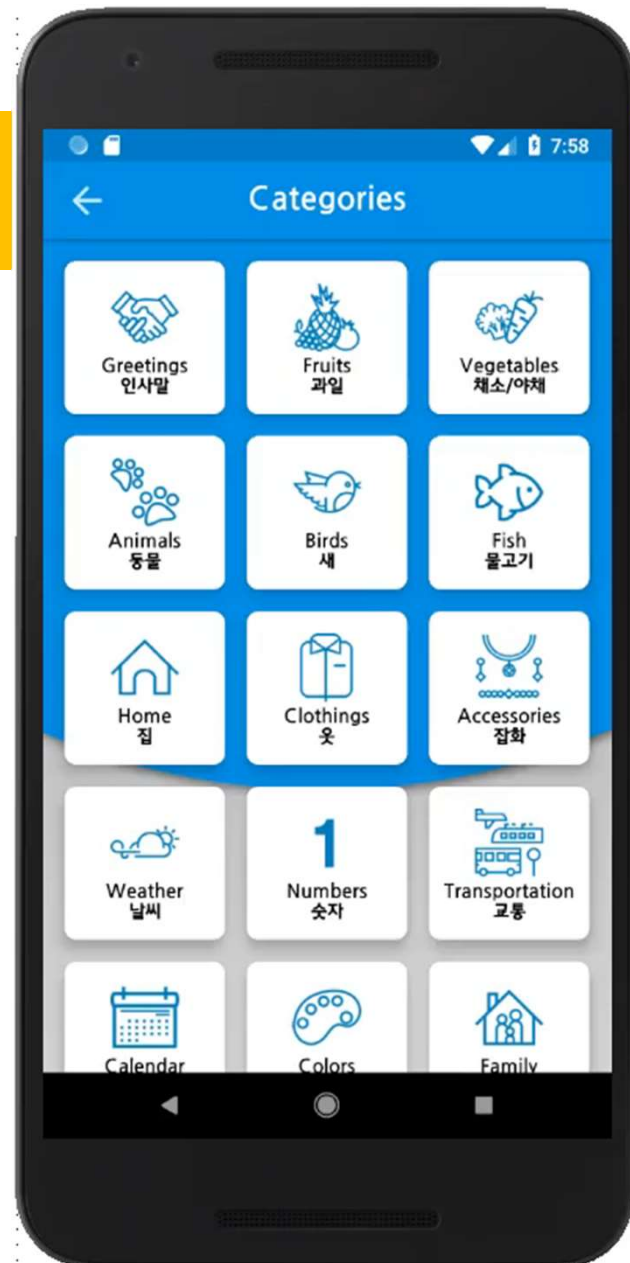




Category 단어 뷰

JAVA

```
wordlist.setOnItemClickListener(new AdapterView.OnItemClickListener() {  
    @Override  
    public void onItemClick(AdapterView<?> parent, View view, int position, long id) {  
        dialog_pos = position;  
  
        if(position == 0) alertDialog(1);  
        else if(position == 1) alertDialog(2);  
        else if(position == 2) alertDialog(3);  
        else if(position == 3) alertDialog(4);  
        else if(position == 4) alertDialog(5);  
        else if(position == 5) alertDialog(6);  
        else if(position == 6) alertDialog(7);  
        else if(position == 7) alertDialog(8);  
        else if(position == 8) alertDialog(9);  
        else if(position == 9) alertDialog(10);  
        else if(position == 10) alertDialog(11);  
        ...  
    });
```



Category 단어 뷰

```
public void alertDialog(int b) {
    try {
        db = dbc.getReadableDatabase();
        sql = "select word_kor, word_eng, word_pro, word_gram from word where category_id_fk = 1 and word_id= " + b + " ; ";
        cursor = db.rawQuery(sql, null);
        int count = cursor.getCount();

        for (int I = 0; I < count; i++) {
            cursor.moveToNext();
            str_kor = cursor.getString(0);
            str_eng = cursor.getString(1);
            str_pro = cursor.getString(2);
            str_gram = cursor.getString(3);
        }

    } catch (Exception e) {
        e.printStackTrace();
    }

    AlertDialog.Builder mBuilder = new AlertDialog.Builder(Categories_wordview1.this);
    View mView = getLayoutInflater().inflate(R.layout.categories_dialog, null);
    TextView txt_id = (TextView) mView.findViewById(R.id.txt_id);
    TextView quote = (TextView) mView.findViewById(R.id.quote);
    TextView txt_kor = (TextView) mView.findViewById(R.id.txt_kor);
    TextView txt_eng = (TextView) mView.findViewById(R.id.txt_eng);
    TextView txt_pro = (TextView) mView.findViewById(R.id.txt_pro);
    TextView txt_gram = (TextView) mView.findViewById(R.id.txt_gram);
    ImageView dialog_cancel = (ImageView) mView.findViewById(R.id.dialog_cancel);
```

Category 단어 뷰

```
//tts
tts = new TextToSpeech(this, this);
btnSpeak = (FloatingActionButton) mView.findViewById(R.id.btnSpeak);

txt_id.setText(Integer.toString(dialog_pos + 1));
String base_white = "
quote.setText(Html.fromHtml("&ldquo;") + base_white.substring(0, (str_kor.length()*2) + Html.fromHtml("&rdquo;"));
txt_kor.setText(str_kor);
txt_eng.setText(str_eng);
txt_pro.setText(str_pro);
txt_gram.setText(str_gram);

// button on click event
btnSpeak.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View arg0) {
        speakOut();
    }
});

mBuilder.setView(mView);
final AlertDialog dialog = mBuilder.create();

dialog_cancel.setOnClickListener(new View.OnClickListener(){
    @Override
    public void onClick(View view) {
        dialog.dismiss();
    }
});
dialog.show();
}
```



TTS API (Text To Speech)

```
@Override
public void onInit(int status) {

    if (status == TextToSpeech.SUCCESS) {

        int result = tts.setLanguage(Locale.KOREA);

        // tts.setPitch(5); // set pitch level
        // tts.setSpeechRate(2); // set speech speed rate

        if (result == TextToSpeech.LANG_MISSING_DATA
            || result == TextToSpeech.LANG_NOT_SUPPORTED) {
            Log.e("TTS", "Language is not supported");
        }
        else {
            btnSpeak.setEnabled(true);
        }
    }
    else {
        Log.e("TTS", "Initialization Failed");
    }
}

//tts
@Override
public void onDestroy() {
    // Don't forget to shutdown!
    if (tts != null) {
        tts.stop();
        tts.shutdown();
    }
    super.onDestroy();
}

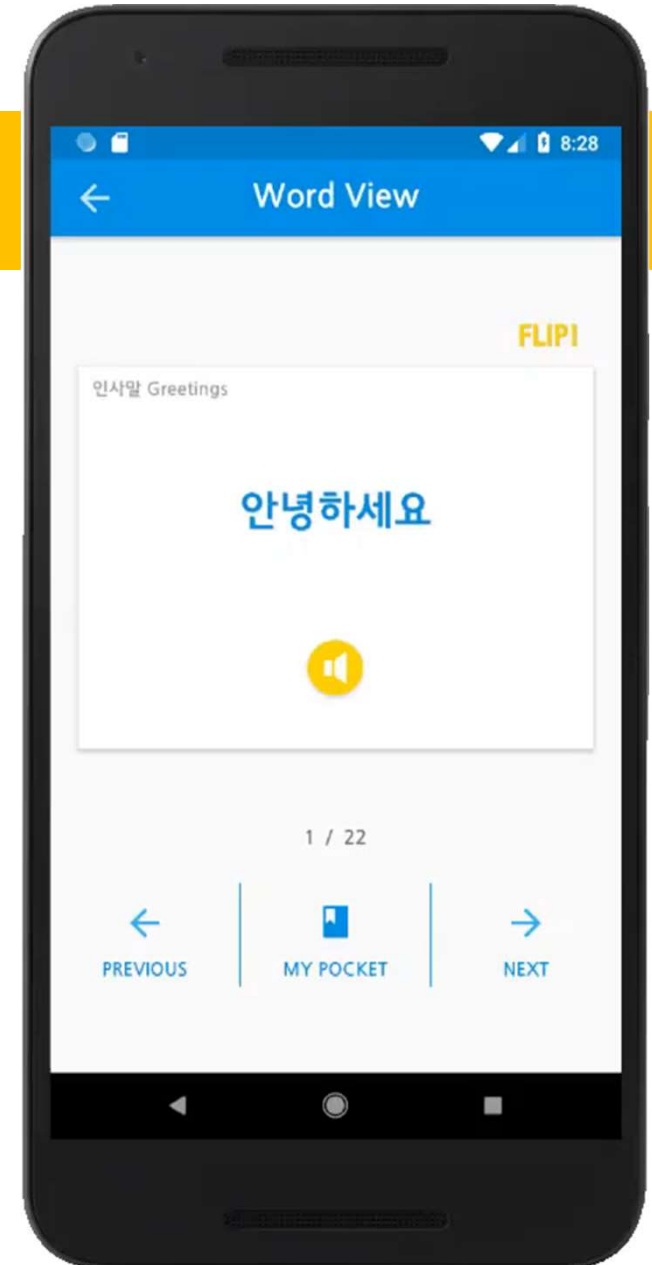
private void speakOut() {
    tts.speak(str_kor, TextToSpeech.QUEUE_FLUSH, null, null);
}
```



Flip Card 단어 뷰

```
adapter = new MyCustomPagerAdapterF(getFragmentManager(), 1, 22);
viewPager = (ViewPager) findViewById(R.id.view_pager);
viewPager.setAdapter(adapter);

@Override
public void onClick(View v) {
    switch (v.getId()) {
        case R.id.flipcard_bookmark:
            startActivity(new Intent(this, MyStudy.class));
            break;
        case R.id.flipcard_next:
            viewPager.setCurrentItem(viewPager.getCurrentItem() + 1, true);
            break;
        case R.id.flipcard_previous:
            viewPager.setCurrentItem(viewPager.getCurrentItem() - 1, true);
            break;
        default:
            showUnsupportedSnackBar();
            break;
    }
}
```



```

public class MyCustomPagerAdapterF extends
    android.support.v13.app.FragmentStatePagerAdapter {

    public CardContainerFragment new_frag;
    public int new_frag_id;
    public int new_frag_count;
    public String new_frag_kor;

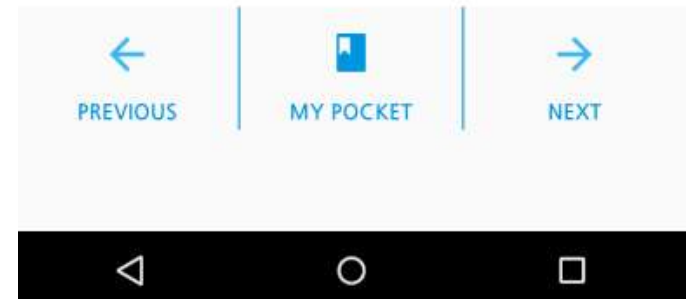
    public MyCustomPagerAdapterF(FragmentManager fm, int id, int count) {
        super(fm);
        new_frag_id = id;
        new_frag_count = count;
    }

    @Override
    public Fragment getItem(int position) {

        new_frag = new CardContainerFragment();
        new_frag.position = position;
        new_frag.id = new_frag_id;
        new_frag.str_kor = new_frag_kor;
        return new_frag;
    }

    @Override
    public int getCount() {
        new_frag = new CardContainerFragment();
        return new_frag_count;
    }
}

```



```

public class CardContainerFragment extends Fragment {

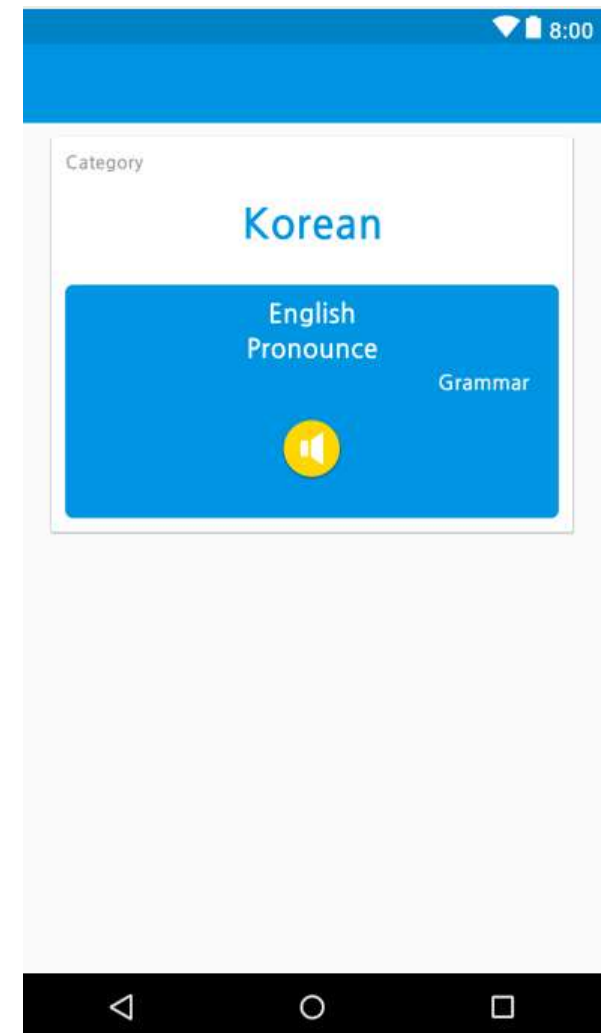
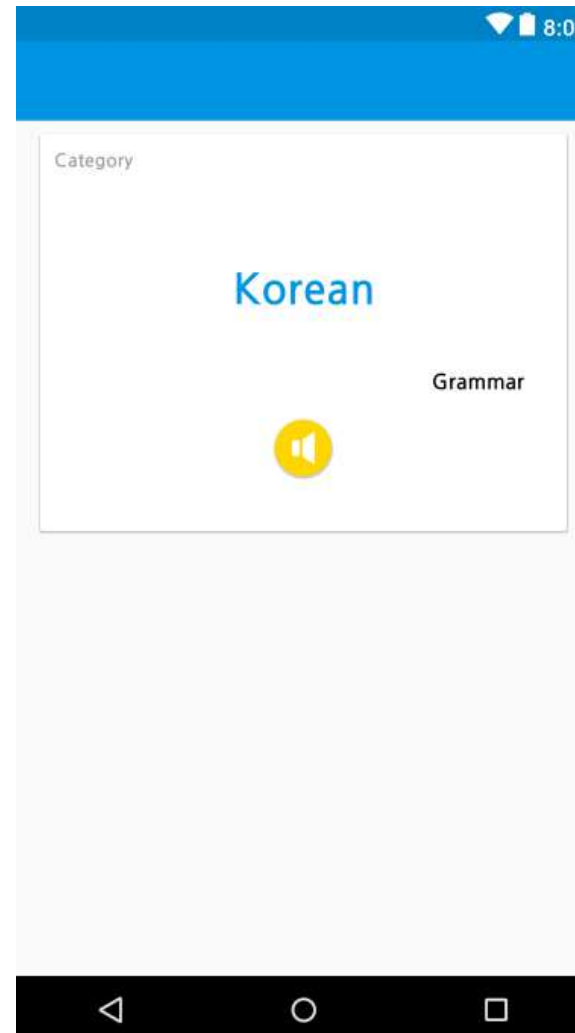
    public String str_kor, str_eng, str_pro, str_gram;
    DB_Word dbc;
    SQLiteDatabase db;
    ...
    private boolean cardFlipped = false;

    public void flipCard() {
        Fragment newFragment;
        if (cardFlipped) {
            newFragment = getFrontFrag();
        } else {
            newFragment = getBackFrag();
        }

        newFragment.setArguments(args);
        getChildFragmentManager()
            .beginTransaction()
            .setCustomAnimations(
                R.animator.card_flip_right_in,
                R.animator.card_flip_right_out,
                R.animator.card_flip_left_in,
                R.animator.card_flip_left_out)
            .replace(R.id.container, newFragment)
            .commit();

        cardFlipped = !cardFlipped;
    }
}

```



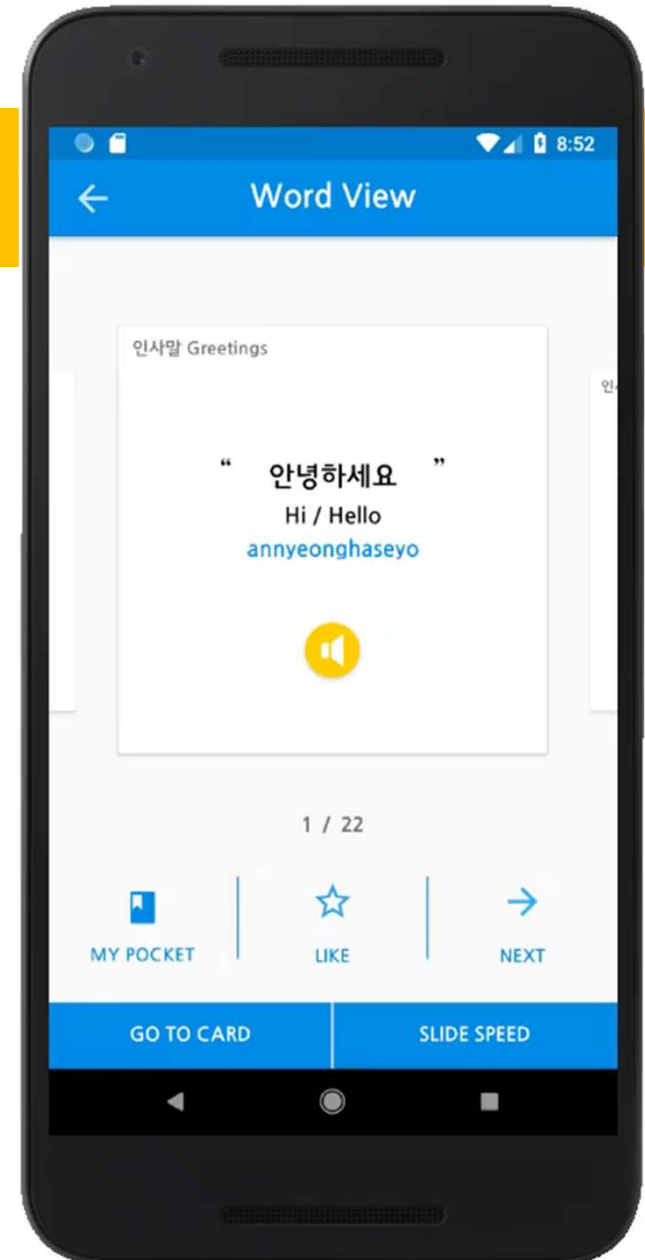


Screen Slide 단어 뷰

```
@Override
public void onClick(View v) {
    switch (v.getId()) {
        case R.id.btn_transition_time:
            DiscreteScrollViewOptions.configureTransitionTime(itemPicker);
            break;
        case R.id.btn_smooth_scroll:
            DiscreteScrollViewOptions.smoothScrollToUserSelectedPosition(itemPicker, v);
            break;
        case R.id.item_btn_mypocket:
            startActivity(new Intent(this, MyStudy.class));
            break;
        case R.id.item_btn_next:
            final RecyclerView.Adapter adapter = itemPicker.getAdapter();
            int position = positionInDataSet + 1;

            if(position == SActivity.total_count){
                position = 0;
            }

            ...
            break;
        default:
            showUnsupportedSnackBar();
            break;
    }
}
```





SharedPreferences

```
public class SharedPreferences {  
  
    public static final String PREFS_NAME = "PRODUCT_APP";  
    public static final String FAVORITES = "Product_Favorite";  
  
    public SharedPreferences() {  
        super();  
    }  
  
    public void saveFavorites(Context context, List<Item> favorites) {  
        SharedPreferences settings;  
        Editor editor;  
  
        settings = context.getSharedPreferences(PREFS_NAME, Context.MODE_PRIVATE);  
        editor = settings.edit();  
  
        Gson gson = new Gson();  
        String jsonFavorites = gson.toJson(favorites);  
  
        editor.putString(FAVORITES, jsonFavorites);  
  
        editor.commit();  
    }  
    ...  
}
```

SharedPreferences

```
public void addFavorite(Context context, Item item) {  
    List<Item> favorites = getFavorites(context);  
    if (favorites == null)  
        favorites = new ArrayList<Item>();  
    favorites.add(item);  
    saveFavorites(context, favorites);  
}
```

```
public void removeFavorite(Context context, Item item) {  
    ArrayList<Item> favorites = getFavorites(context);  
    if (favorites != null) {  
        favorites.remove(item);  
        saveFavorites(context, favorites);  
    }  
}
```

```
public ArrayList<Item> getFavorites(Context context) {  
    SharedPreferences settings;  
    List<Item> favorites;  
  
    settings = context.getSharedPreferences(PREFS_NAME,  
Context.MODE_PRIVATE);  
  
    if (settings.contains(FAVORITES)) {  
        String jsonFavorites = settings.getString(FAVORITES, null);  
        Gson gson = new Gson();  
        Item[] favoriteItems = gson.fromJson(jsonFavorites,  
            Item[].class);  
  
        favorites = Arrays.asList(favoriteItems);  
        favorites = new ArrayList<Item>(favorites);  
    } else  
        return null;  
  
    return (ArrayList<Item>) favorites;  
}
```

SharedPreferences

```
private void changeRateButtonState(Item item) {
    List<Item> favorites = SActivity.sharedPreference.getFavorites(this);

    if (SActivity.isRated(item.getKor())) { // Like: star -> ~star

        if (favorites != null) {

            if (contains(favorites, item.getKor())) {
                System.out.println("Data Exist(s)");
            } else {
                SActivity.sharedPreference.addFavorite(this, SActivity.arr.get(positionInDataSet));
                System.out.println("Not Exist(s)");
            }

        } else if (favorites == null) {
            SActivity.sharedPreference.addFavorite(this, SActivity.arr.get(positionInDataSet));
            System.out.println("Favorite null, Not Exist(s)");
        }

         rateItemButton.setImageResource(R.drawable.ic_star_black_24dp);
         rateItemButton.setColorFilter(ContextCompat.getColor(this, R.color.colorAccent_yellow));

    } else { // Like: ~star -> star
        if (contains(favorites, item.getKor())) {
            SActivity.sharedPreference.removeFavorite(this, arr.get(positionInDataSet));
             rateItemButton.setImageResource(ic_star_border_black_24dp);
             rateItemButton.setColorFilter(ContextCompat.getColor(this, colorPrimary));
        } else {
             rateItemButton.setImageResource(ic_star_border_black_24dp);
             rateItemButton.setColorFilter(ContextCompat.getColor(this, colorPrimary));
        }
    }
}
```



SQL Insert data

```
public void insertAll() {
    db = dbc.getWritableDatabase();
    db.beginTransaction();
    try {
        BufferedReader buffer = new BufferedReader(new InputStreamReader(is, "UTF-8"));

        String tableName = "word";
        String columns = "word_id, category_id_fk, word_kor, word_eng, word_pro, word_gram";
        String str1 = "insert into " + tableName + "(" + columns + ") values(";
        String str2 = ");";
        String line = "";

        while ((line = buffer.readLine()) != null) {
            StringBuilder sb = new StringBuilder(str1);
            String[] str = line.split(",");
            sb.append(str[0] + ", ");
            sb.append(str[1] + ", '");
            sb.append(str[2] + "', '");
            sb.append(str[3] + "', '");
            sb.append(str[4] + "', '");
            sb.append(str[5] + "'");
            sb.append(str2);
            db.execSQL(sb.toString());
        }
        db.setTransactionSuccessful();
        db.endTransaction();
    } catch (IOException ex) {
        ex.printStackTrace();
    } finally {
        try {
            is.close();
        } catch (IOException e) {
            e.printStackTrace();
        }
    }
}
```



SQL Select data

```
public void select(int select_category) {
    category_type = select_category;

    try {
        db = dbc.getReadableDatabase();
        sql = "select word_id, word_kor, word_eng, word_pro, word_gram from word where category_id_fk = " + select_category + ";";
        cursor = db.rawQuery(sql, null);
        total_count = cursor.getCount();
        result = new String[total_count][4];

        for (int i = 0; i < total_count; i++) {
            cursor.moveToNext();
            String str_id = cursor.getString(0);
            String str_kor = cursor.getString(1);
            String str_eng = cursor.getString(2);
            String str_pro = cursor.getString(3);
            String str_gram = cursor.getString(4);
            result[i][0] = str_kor;
            result[i][1] = str_eng;
            result[i][2] = str_pro;
            result[i][3] = str_gram;
        }
    } catch (Exception e) {
        e.printStackTrace();
    }
} //select()
```



SQL Delete data

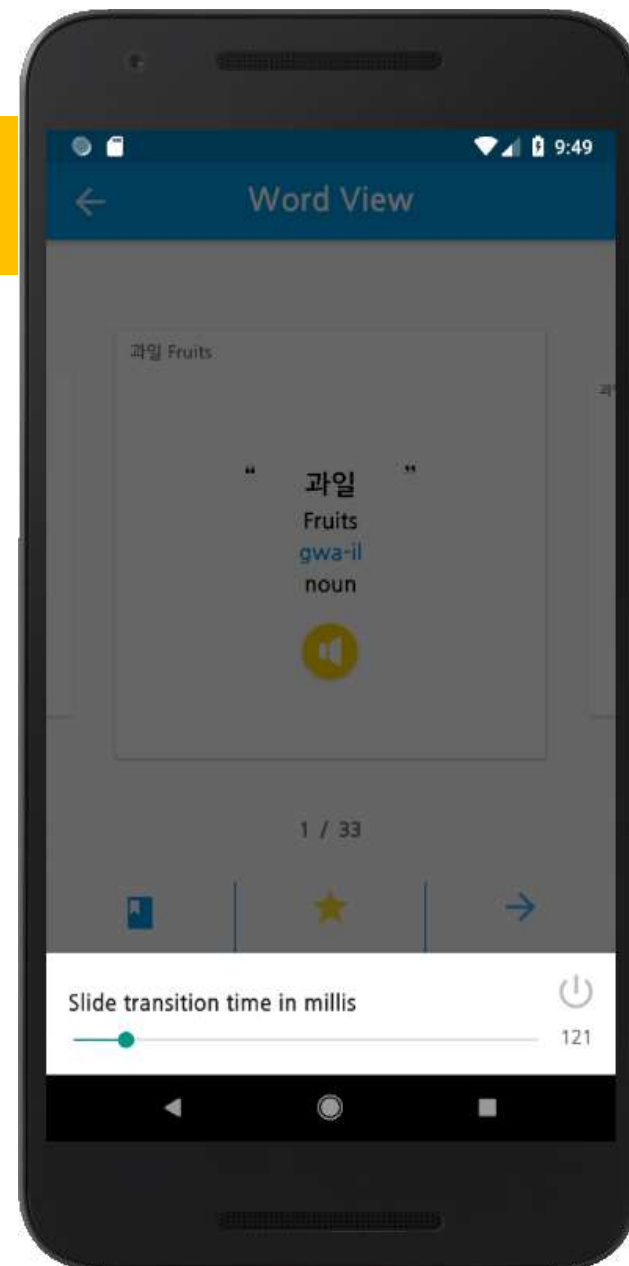
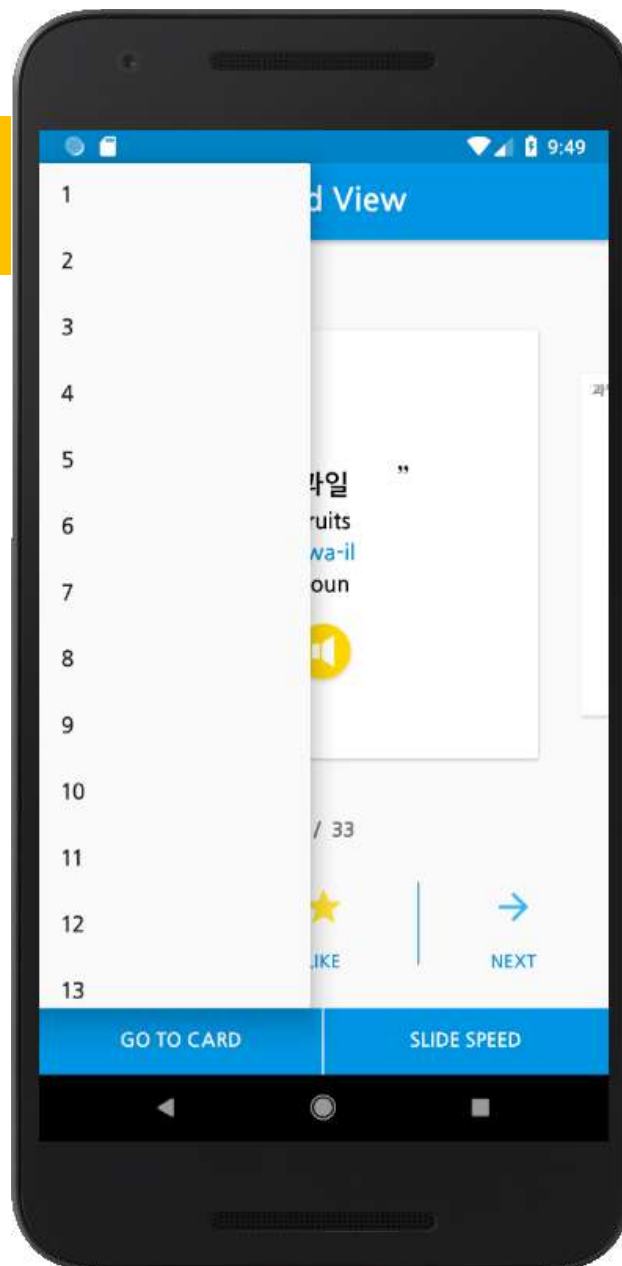
```
public void delete() {
    try {
        db = dbc.getWritableDatabase();
        db.execSQL("delete from word;");
    } catch (Exception e) {
        e.printStackTrace();
    }
}
//delete()

public void getData() {
    Item[] new_item_array = new Item[total_count];
    for (int i = 0; i < total_count; i++) {
        //arr = Arrays.asList(
        new_item_array[i] = new Item(i + 1,
            category_type_kor[category_type-1], category_type_eng[category_type-1],
            result[i][0], result[i][1], result[i][2], result[i][3], R.drawable.white_card);
    }
    arr = Arrays.asList(new_item_array);
}
```



추가 기능

- GO TO CARD
: n번째 카드로 이동
- SLIDE SPEED
: 카드 넘기기 속도 조절





Game 단어 뷰

```
public void select(int num) {
    try {
        db = dbc.getReadableDatabase();

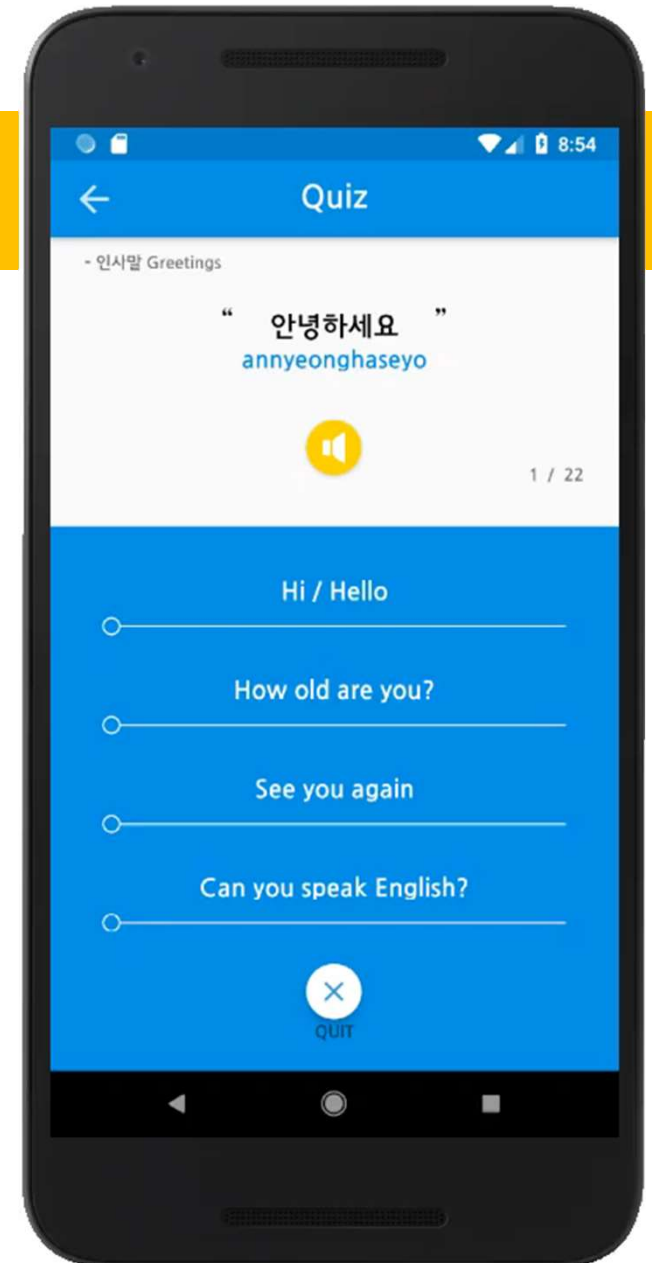
        // 질문: 한국어, 답변: 영어1, 영어2, 영어3, 영어4
        sql = "select word_kor, word_eng, word_pro, word_gram from word;";

        cursor = db.rawQuery(sql, null);
        count = cursor.getCount();

        for (int i = 0; i < count; i++) {
            cursor.moveToNext();
            String str_kor = cursor.getString(0);
            String str_eng = cursor.getString(1);
            String str_pro = cursor.getString(2);
            String str_gram = cursor.getString(3);

            str_answer[i] = str_eng;
            kor[i] = str_kor;
            str_question_kor[i] = str_kor;
            str_question_pro[i] = str_pro;
            str_question_gram[i] = str_gram;

            array.add(str_answer[i]);
        }
    }
}
```




```

q = str_question[num];
a = str_answer[num];
k = kor[num];
q_k = str_question_kor[num];
p = str_question_pro[num];
g = str_question_gram[num];

```

```

array.remove(num);
shuffle(array);
String wa1 = array.get(0);
String wa2 = array.get(1);
String wa3 = array.get(2);

```

```

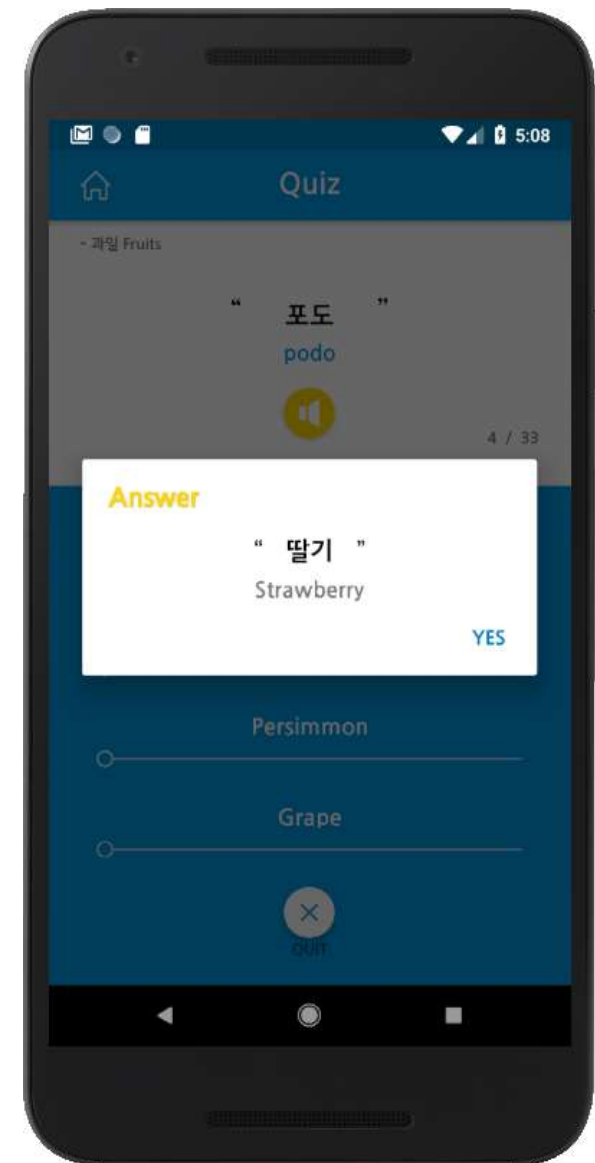
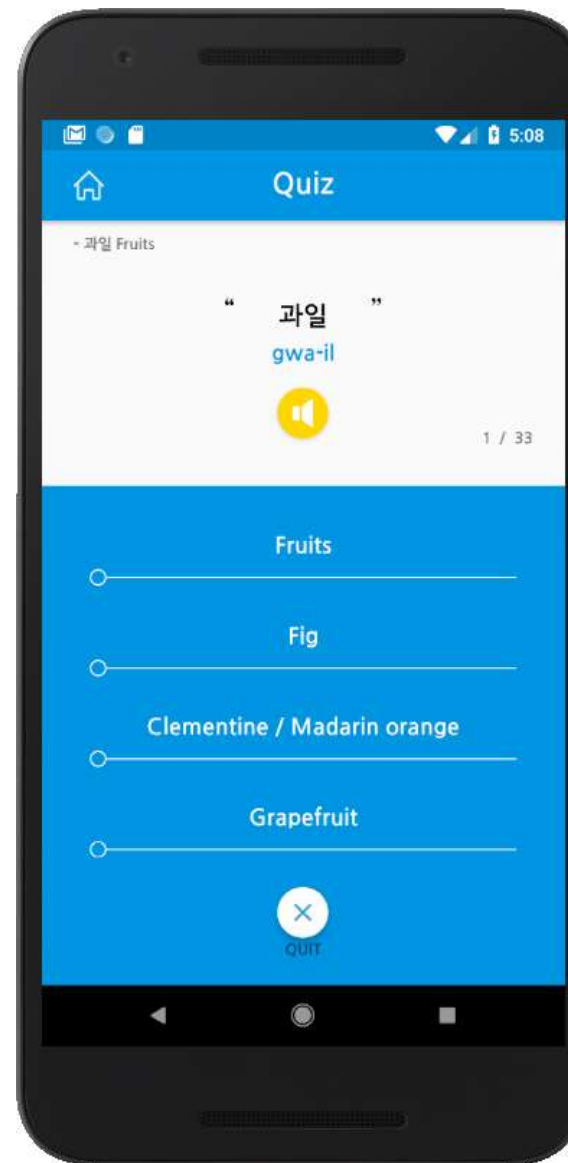
finalAnswerArray.add(a);
finalAnswerArray.add(wa1);
finalAnswerArray.add(wa2);
finalAnswerArray.add(wa3);
shuffle(finalAnswerArray);

```

```

question.setText(q);
question_kor.setText(q_k);
question_pro.setText(p);
question_gram.setText(g);
tv_answer1.setText(finalAnswerArray.get(0));
tv_answer2.setText(finalAnswerArray.get(1));
tv_answer3.setText(finalAnswerArray.get(2));
tv_answer4.setText(finalAnswerArray.get(3));

```





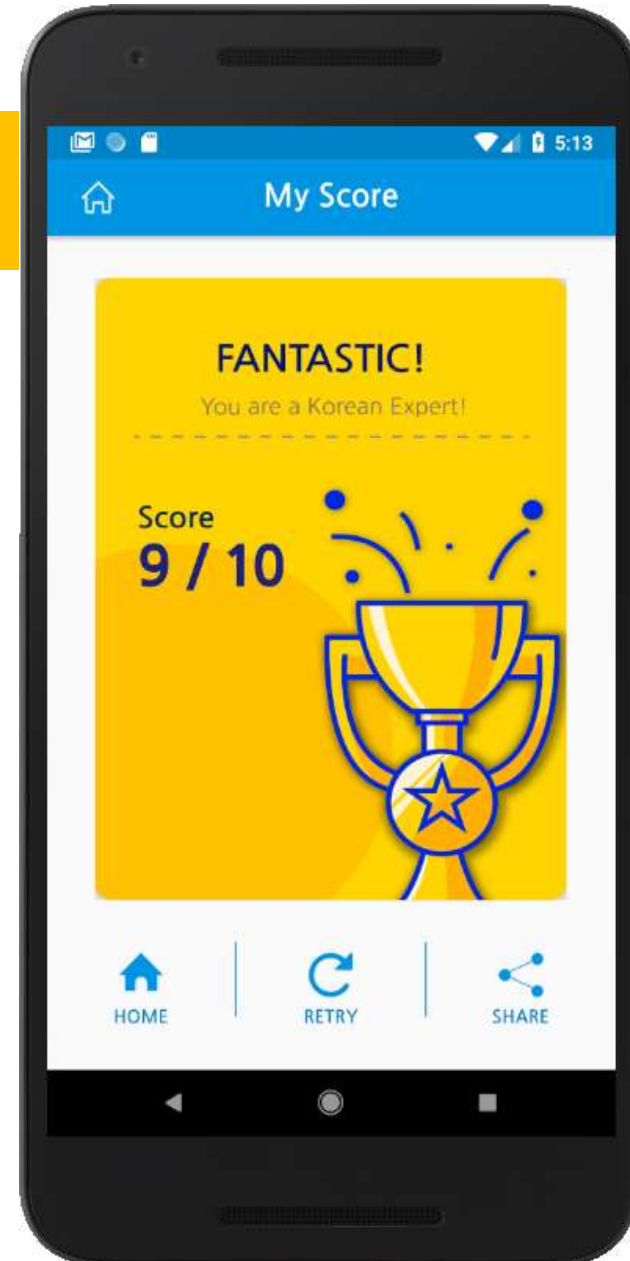
My Score

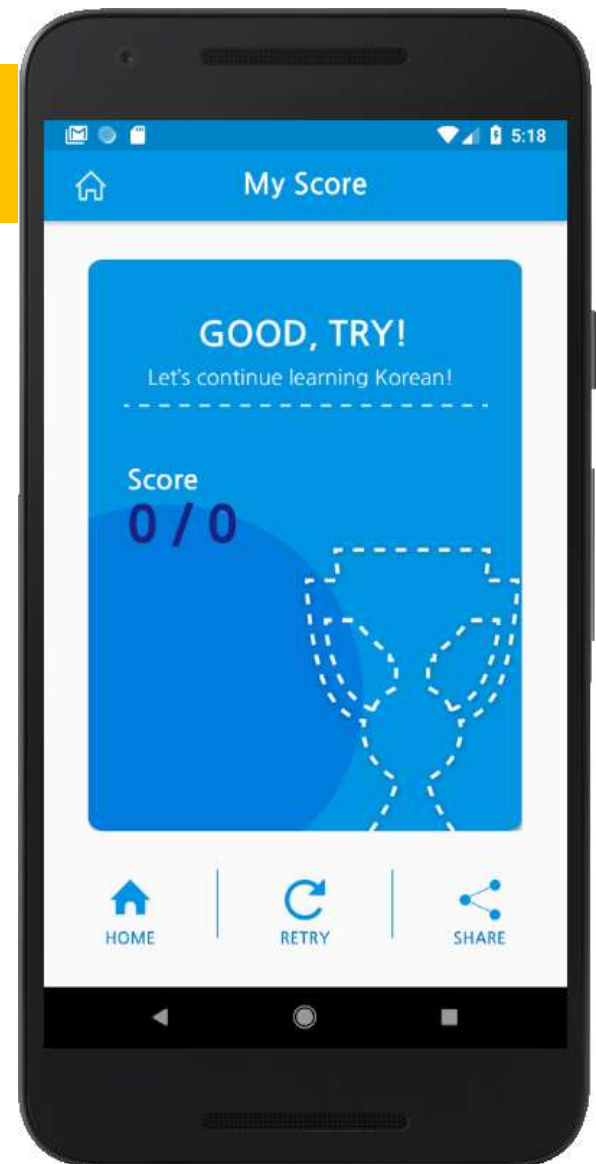
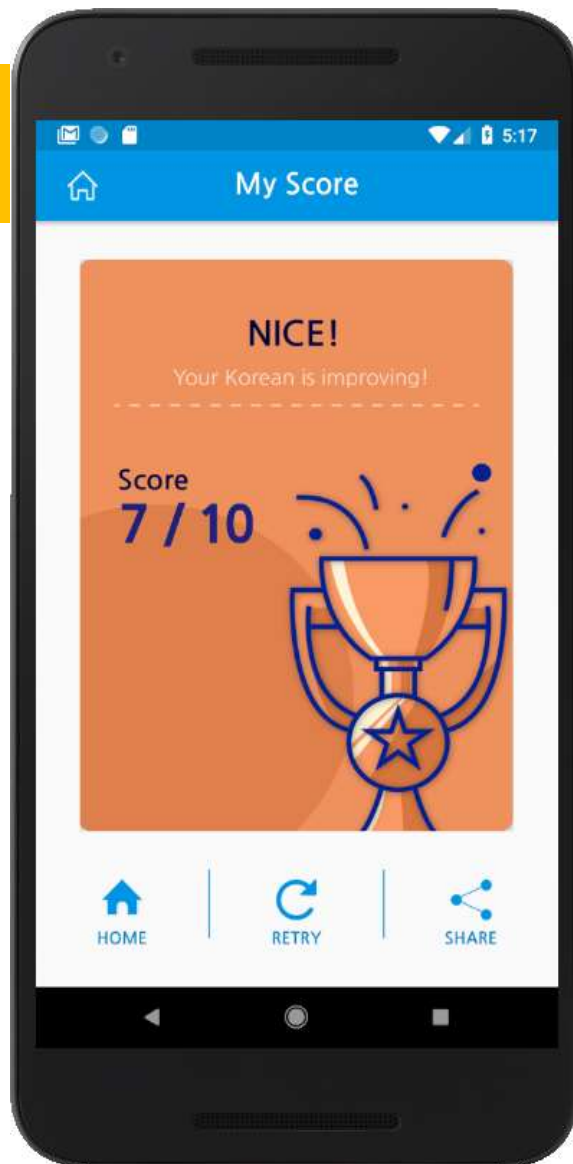
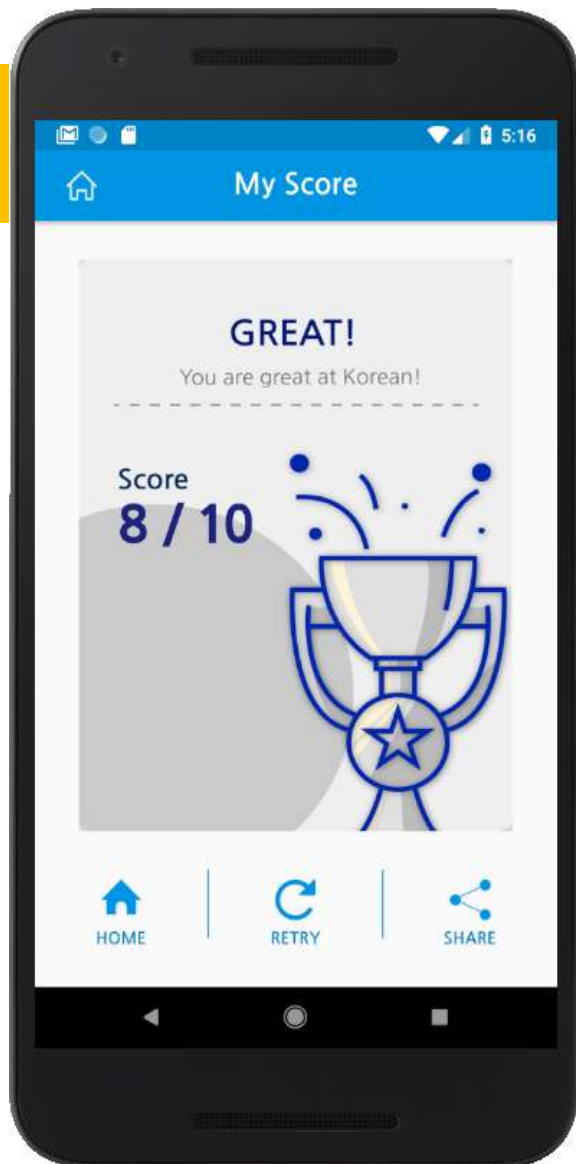
```
Intent intent = getIntent();
int point = intent.getIntExtra("str_gamescore", 1);
int QCount = intent.getIntExtra("QCount", 1);

gamescore.setText(point + " / " + QCount);

int cal_trophy = Math.round((((float)point/QCount) * 100));

if(cal_trophy >= 90){
    trophy_bg.setBackgroundResource(trophyImage[0]);
} else if(cal_trophy >= 80){
    trophy_bg.setBackgroundResource(trophyImage[1]);
} else if(cal_trophy >= 70){
    trophy_bg.setBackgroundResource(trophyImage[2]);
} else if(cal_trophy < 70){
    trophy_bg.setBackgroundResource(trophyImage[3]);
}
```

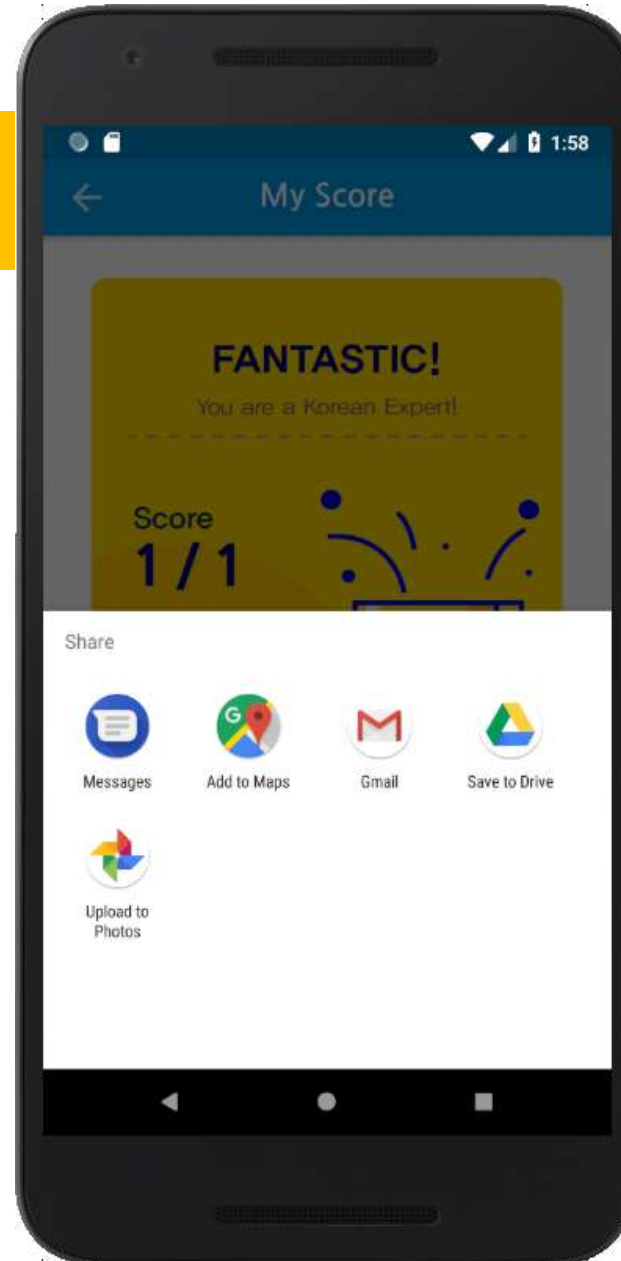






Share

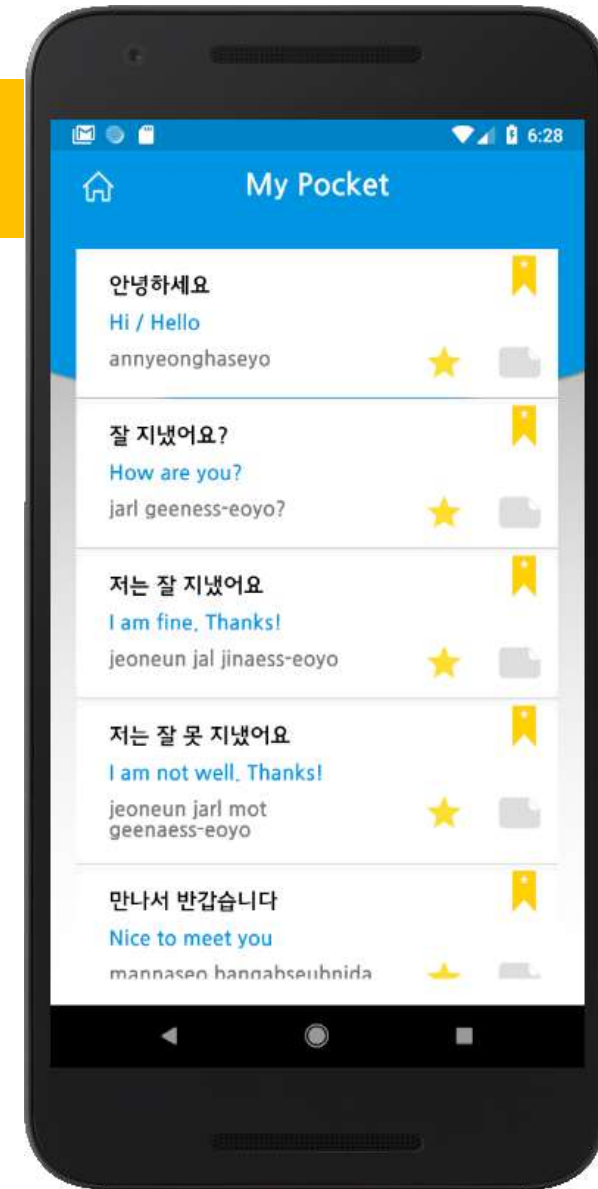
- Messages
- Google Maps
- Gmail
- Google Drive
- Google Photos
- Facebook
- KakaoTalk
- Twitter





My Pocket

```
if (favorites != null) {  
    favoriteList.setAdapter(myStudyListAdapter);  
  
    favoriteList.setOnItemClickListener(new AdapterView.OnItemClickListener() {  
  
        public void onItemClick  
            (AdapterView<?> parent, View arg1, final int position, long arg3) {  
  
            imgbtn_favorite = (ImageView) arg1.findViewById(R.id.imgbtn_favorite);  
            imgbtn_favorite.setOnClickListener(new View.OnClickListener(){  
  
                @Override  
                public void onClick(View view) {  
                    SActivity.setRated(favorites.get(position).getKor(), false);  
                    sharedPreferences.removeFavorite(getBaseContext(), favorites.get(position));  
                    myStudyListAdapter.remove_pocket_item(favorites.get(position));  
                }  
            });  
        }  
    });  
}
```





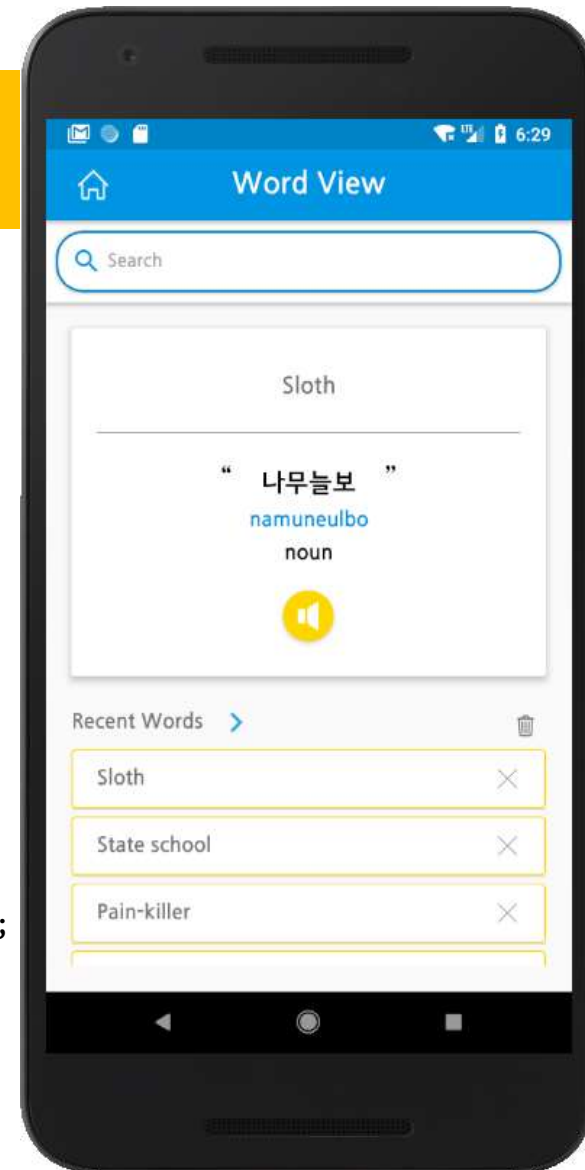
단어 뷰 Intent

```
searchSharePreference = new Search_SharePreference();
searchList = (ListView) findViewById(R.id.searchListView);
searchsaveFav = searchSharePreference.getFavorites(this);
Collections.reverse(searchsaveFav);
searchSaveListAdapter = new SearchSaveListAdapter(this, searchsaveFav);

if(searchsaveFav != null){
    searchList.setAdapter(searchSaveListAdapter);

    searchList.setOnItemClickListener(new AdapterView.OnItemClickListener() {
        public void onItemClick(AdapterView<?> adapterView, View view, final int I, long l) {

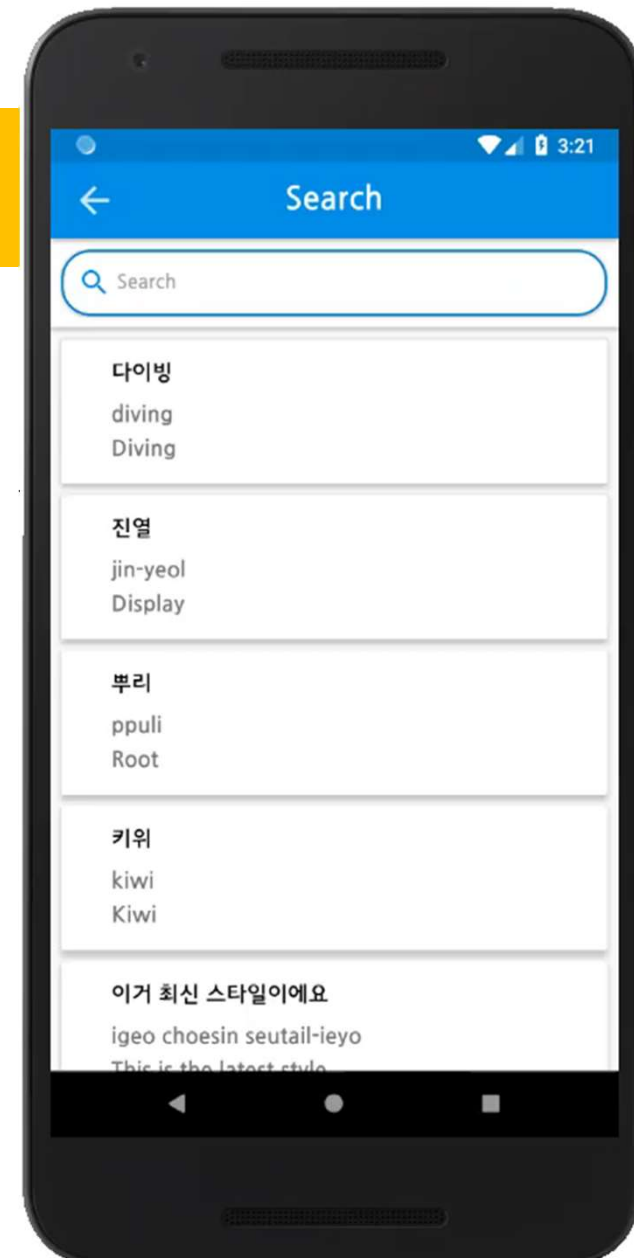
            btnRemove = (ImageView) view.findViewById(R.id.btnRemove);
            btnRemove.setOnClickListener(new View.OnClickListener(){
                @Override
                public void onClick(View view) {
                    searchSharePreference.removeFavorite(getBaseContext(), searchsaveFav.get(i));
                    searchSaveListAdapter.remove_pocket_item(searchsaveFav.get(i));
                }
            });
        }
    });
}
```





Search

```
search_expanded_edit_text.addTextChangedListener(new TextWatcher() {  
  
    @Override  
    public void beforeTextChanged(CharSequence charSequence, int i, int i1, int i2) {  
    }  
  
    @Override  
    public void onTextChanged(CharSequence charSequence, int i, int i1, int i2) {  
    }  
  
    @Override  
    public void afterTextChanged(final Editable editable) {  
        Log.d(TAG, "afterTextChanged: " + editable);  
  
        String filterText = editable.toString();  
        ((ListViewAdapter) listView.getAdapter()).getFilter().filter(filterText);  
  
        Log.d(TAG, "filterText: " + filterText);  
    }  
});
```





작품 기대효과 및 향후 계획

- 다국적언어 지원 기능

: 사용자 타겟층을 국제적으로 아울러 지정하여, 중국어, 스페인어, 러시아어, 아랍어 등의 다국적언어 기능을 DB에 추가할 계획입니다.

결과 - 전세계적으로 많은 사용자들에게 서비스 제공 가능

- 단어에 대한 문장/대화 예시 기능 지원

: 문장 또는 간단한 대화를 저장하는 DB를 추가하여, 특정 단어에 관련된 문장 예시들을 제시합니다.

결과 - 사용자들의 공부 환경 및 단어에 대한 이해력 증진 도움



감사합니다
KoreanPocket