

# 高级工程实训报告

14353002 白冰

## 目录

### 第一章 概述

### 第二章 背景

- 2.1 项目背景
- 2.2 项目需求
- 2.3 开发环境

### 第三章 项目说明

- 3.1 功能模块
- 3.2 逻辑流程
- 3.3 技术说明
  - 3.3.1 添加闹铃功能
  - 3.3.2 设置铃声，词库，单词查询模块
  - 3.3.3 答题模块
  - 3.3.4 Widget模块
  - 3.3.5 引导与欢迎界面
- 3.4 代码架构

### 第四章 应用效果

- 4.1 欢迎界面
- 4.2 添加闹钟提醒
- 4.3 铃声，词库设置，分享
- 4.5 单词查询
- 4.6 答题
- 4.7 Widget
- 4.8 项目源码

### 第五章 开发过程遇到的问题及解决办法

- 5.1 重复闹钟设置
- 5.2 设备重启后处理
- 5.3 App保活措施
- 5.4 QQ分享和语音输入

### 第六章 实验感想

## 第七章 参考资料

# 第一章 概述

本课题实现了一款集合了单词查询，测验，闹钟提醒功能的安卓应用，作为毕业设计《基于安卓》的子模块，目前实现了以下功能：a) 实现单词查询功能，支持中英文互译 b) 能通过界面设置单个或多个闹钟提醒（目前支持GMT+8 北京时间） c) 测验功能，闹钟提醒后开始单词测验，目前仅支持单选题。 d) Wigdet小部件背单词功能。

## 第二章 需求分析

### 2.1 项目背景

“E-learning”这一术语，自20世纪90年代中期开始流行起来，其目标是通过信息和通信技术建立一个独立于时间和地点的学习社区。由于近年来全球移动设备的普及，许多人每天都会携带智能手机，因此，他们可以随时轻松使用ICT技术，如电话，电子邮件，访问互联网和各种应用程序。因此，考虑到使用移动设备的电子学习环境，我认为使用智能手机对于英语词汇记忆等独立学习是有效的，因为学习者可以随时随地使用智能手机，充分利用零碎时间。在英语词汇学习中，延续是一个重要的因素，但很多学习者不善于坚持学习。在不善坚持的学习者中，有两种类型的学习者:(1)学习动机不大的学习者;(2)即使有学习动机，也无法抗拒诱惑的学习者。本选题以(2)学习者为研究对象，利用智能手机的特点，希望通过，添加提醒，游戏化等技术，为学习者可持续学习开发了英语词汇学习应用，重点讨论 TOEFL词汇和英语四六级词汇的学习。

### 2.2 项目需求

爱单词（LoveWords）是一款英文词汇学习应用，适合有学习动机，也无法抗拒诱惑，习惯拖延的学习者。本应用将背单词，词典，闹钟功能结合起来，小巧实用。

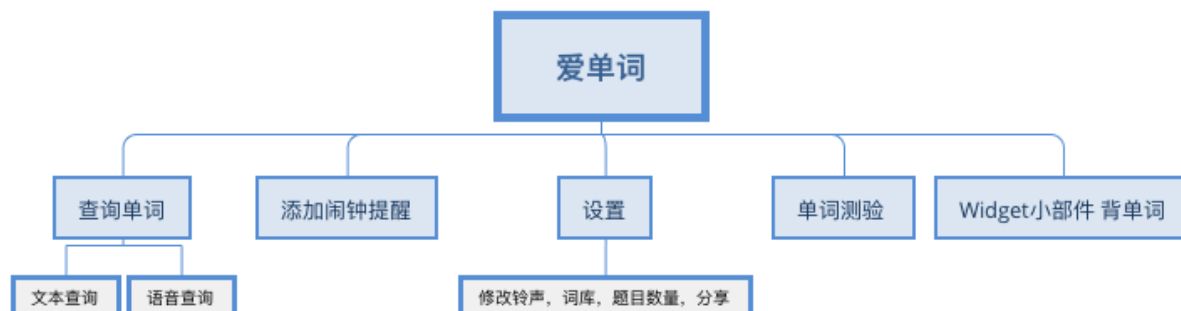
### 2.3 开发环境

MAC OS Java Development Kit 8 Android Studio 3.0.1 SQLite

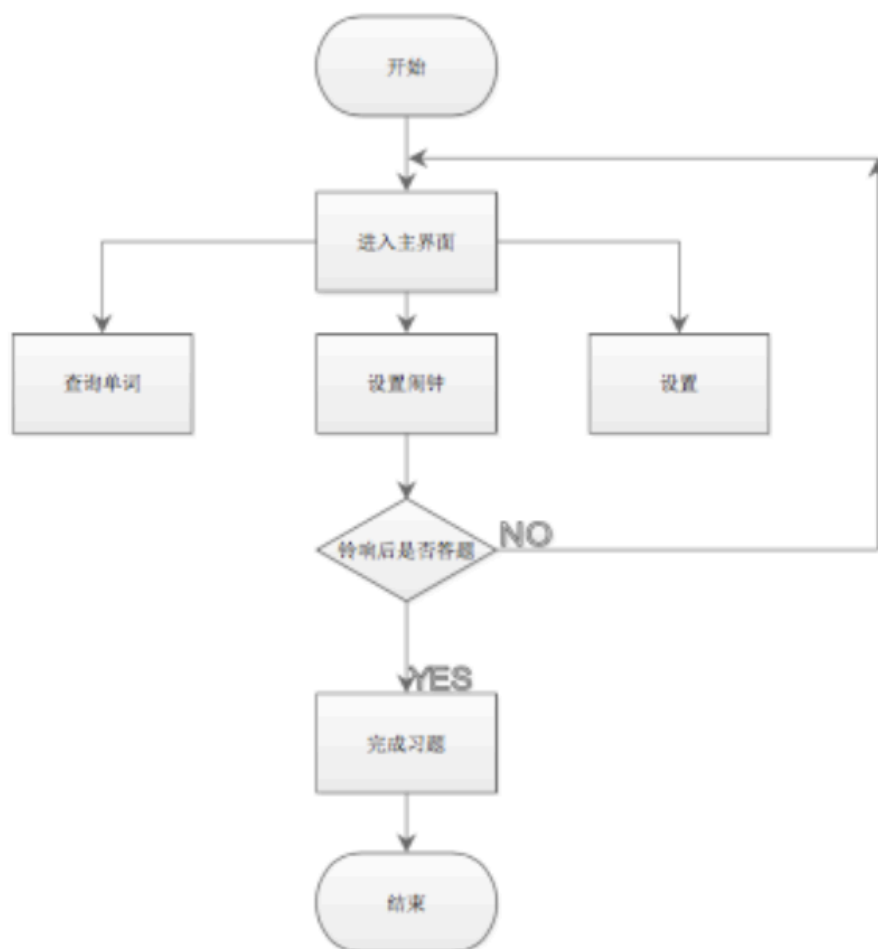
## 第三章 项目说明

### 3.1 功能模块

核心功能模块如图所示：



## 3.2 逻辑流程



## 3.3 技术说明

### 3.3.1 添加闹铃模块

**alarmsetting.java** 几个重要的函数如下： `getting()`函数：获取用户设置的时间

```
public void gettime(){
    if (Build.VERSION.SDK_INT >= Build.VERSION_CODES.M){
        hour = timePicker.getHour();
        minute = timePicker.getMinute();
    }
    else {
        hour = timePicker.getCurrentHour();
        minute = timePicker.getCurrentMinute();
    }
}
```

**savesomething()** 函数：查询数据库，检查设置时间是否与已有闹钟时间相距过近，若是，弹出提示消息，否则在数据库中存储该闹钟时间和重复时段 onCreate()函数：为各个button, imagebutton 设置监听器，调用各函数完成本activity的功能

```
@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.alarmsetting);
    findview();
    timePicker.setIs24HourView(true);
    View.OnClickListener onClickListener = new View.OnClickListener() {
        @Override
        public void onClick(View v) {
            switch (v.getId()){
                case R.id.save:
                    savesomething();
                    break;
                case R.id.back:
                    Alarmsettintg.this.finish();
                    break;
                case R.id.sunday:
                    repeatdate(0);
                    break;
                case R.id.monday:
                    repeatdate(1);
                    break;
                case R.id.tuesday:
                    repeatdate(2);
                    break;
                case R.id.wednesday:
                    repeatdate(3);
                    break;
                case R.id.thursday:
                    repeatdate(4);
                    break;
                case R.id.friday:
                    repeatdate(5);
                    break;
                case R.id.saturday:
                    repeatdate(6);
                    break;
            }
        }
    };
    save.setOnClickListener(onClickListener);
    back.setOnClickListener(onClickListener);
    button[0].setOnClickListener(onClickListener);
    button[1].setOnClickListener(onClickListener);
    button[2].setOnClickListener(onClickListener);
}
```

```

        button[3].setOnClickListener(onClickListener);
        button[4].setOnClickListener(onClickListener);
        button[5].setOnClickListener(onClickListener);
        button[6].setOnClickListener(onClickListener);
    }

```

### 3.3.2 设置铃声，词库，查询单词模块

#### MainActivity.java

功能：单词查询；闹钟显示，删除；各项设置

几个重要的函数如下：

**onCreate()**函数：为相应的button添加监听器，调用相应的函数来完成本activity的功能

**aftersearchfromdatabase()**函数：从网络查询（调用有道翻译的API）单词含义，并对返回的XML进行解析

**deletedata()**函数：删除闹钟信息

**shareevent()**函数：调用QQ分享API，与类ShareListener（自定义的类implements IUListener）和**onActivityResult()**函数共同完成分享到QQ或QQ空间的功能

```

public void shareevent(){
    Bundle params = new Bundle();
    params.putInt(QQShare.SHARE_TO_QQ_KEY_TYPE,
        QQShare.SHARE_TO_QQ_TYPE_DEFAULT);
    params.putString(QQShare.SHARE_TO_QQ_TITLE, "Love Words介绍");
    params.putString(QQShare.SHARE_TO_QQ_SUMMARY, "Love Words是一款好用
        小巧的单词学习应用，充分利用你零碎的时间来学习英语");

    //params.putString(QQShare.SHARE_TO_QQ_IMAGE_URL, "http://www.word");
    params.putString(QQShare.SHARE_TO_QQ_APP_NAME, "Love Words");
    tencent.shareToQQ(MainActivity.this, params, shareListener);
}

```

**initSpeech()**函数：调用科大讯飞语音听写API，与类Voice和parseVoice()共同完成语音输入的功能

```

public void initSpeech(final Context context) {
    RecognizerDialog mDialog = new RecognizerDialog(context,null);
    mDialog.setParameter(SpeechConstant.LANGUAGE,"en_us");
    mDialog.setParameter(SpeechConstant.ACCENT,null);
    mDialog.setListener(new RecognizerDialogListener(){
        @Override
        public void onResult(RecognizerResult recognizerResult,boolean isLast)
        {
            if (!isLast){
                String result = parseVoice(recognizerResult.getResultString());
                searchword.setText(result);
            }
        }
        @Override
        public void onError(SpeechError speechError){

        }
    });
    mDialog.show();
}

```

**whether\_first\_and\_something()**函数：判断是否第一次打开应用，若是，导入数据库

```

public void whether_first_and_something(){
    preferences = getSharedPreferences("sharedpreference",MODE_PRIVATE);
    first = preferences.getBoolean(first_start,true);
    if(first){
        importDatabase();
        editor = preferences.edit();
        editor.putBoolean(first_start,false).commit();
    }
}

```

**Network\_available()**函数：检测网络是否可访问

```
public boolean Network_available(){
    ConnectivityManager connectivityManager = (ConnectivityManager)
getSystemService(CONNECTIVITY_SERVICE);
    if (connectivityManager == null)
        return false;
    else{
        NetworkInfo networkInfos = connectivityManager.getActiveNetworkInfo();
        if (networkInfos != null && networkInfos.isConnected()){
            return true;
        }
    }
    return false;
}
```

**searchfromdatabase()**函数：从数据库中查询是否有该单词

```

public String[] searchfromdatabase() {
    String [] translation = {"", "", ""};
    File file = new File(getFilesDir()+"/databases/newdata.db");
    SQLiteDatabase database =
    SQLiteDatabase.openDatabase(file.getPath(), null, SQLiteDatabase.OPEN_READWRITE);
    Cursor cursor = database.rawQuery("select * from toeflcomplete where
english = ?", new String[]{searchword.getText().toString()});
    int column = cursor.getColumnCount();
    while (cursor.moveToNext()){
        for (int i = 0; i < column; i++) {
            String columnname = cursor.getColumnName(i);
            String columnvalue =
cursor.getString(cursor.getColumnIndex(columnname));
            translation[i] = columnvalue;
        }
    }
    if (translation[0].equals("")){
        cursor = database.rawQuery("select * from toeflcomplete where english =
?", new String[]{searchword.getText().toString()});
        column = cursor.getColumnCount();
        while (cursor.moveToNext()){
            for (int i = 0; i < column; i++){
                String columnname = cursor.getColumnName(i);
                String columnvalue =
cursor.getString(cursor.getColumnIndex(columnname));
                translation[i] = columnvalue;
            }
        }
    }
    if (database != null)
        database.close();
    if (translation[0].equals("")) {
        translation[0] = notFoundindb;
        return translation;
    }
    return translation;
}
}

```

**updateUI()**函数：从数据库获取闹钟信息，显示在alarm.xml中



```

public void updateUI(){
    String time,repeate;
    boolean hasvalue = false;
    File file = new File(getFilesDir()+"/databases/newdata.db");
    SQLiteDatabase database =
SQLiteDatabase.openDatabase(file.getPath(),null,SQLiteDatabase.OPEN_READWRIT
TE);

    Cursor cursor = database.rawQuery("select * from alarm", null);
    List<Map<String,Object>> data = new ArrayList<>();
    while (cursor.moveToNext()){
        hasvalue = true;
        time = cursor.getString(0);
        repeate = cursor.getString(1);
        Map<String,Object> temp = new LinkedHashMap<>();
        temp.put("time",time);
        temp.put("repeate",repeate);
        data.add(temp);
    }
    cursor.close();
    database.close();
    if (hasvalue) {
        simpleAdapter = new SimpleAdapter(this, data,
R.layout.alarm_item, new String[]{"time", "repeate"}, new int[]
{R.id.alarm_time, R.id.alarm_repeate});
        alarm_list.setAdapter(simpleAdapter);
    }
    else {
        alarm_list.setAdapter(null);
    }
}
}

```

alarmselectclick()函数：选择闹钟铃声，wordselectclick()函数选择测验词库，questionselectclick()函数选择题目数量的实现基本类似

```

public void alarmselectclick(){
    LayoutInflater inflater = LayoutInflater.from(MainActivity.this);
    alarmselectview =
inflater.inflate(R.layout.alarmselectdialog,null);
    AlertDialog.Builder builder = new
AlertDialog.Builder(MainActivity.this);
    builder.setView(alarmselectview);
    final AlertDialog dialog = builder.create();
    marriage = (RadioButton)
alarmselectview.findViewById(R.id.marriage);
    air = (RadioButton) alarmselectview.findViewById(R.id.air);
    original =(RadioButton)
alarmselectview.findViewById(R.id.original);
}

```

```

        preferences =
getSharedPreferences("sharedpreference",MODE_PRIVATE);
        String selectsong = preferences.getString(song,song1);
        if (selectsong.equals(song1)){
            marriage.setChecked(true);
        }
        else if (selectsong.equals(song2)){
            air.setChecked(true);
        }
        else {
            original.setChecked(true);
        }
        dialog.show();
        rgforalarm = (RadioGroup)
alarmselectview.findViewById(R.id.rgforalarm);
        rgforalarm.setOnCheckedChangeListener(new
RadioGroup.OnCheckedChangeListener() {
            @Override
            public void onCheckedChanged(RadioGroup group, int checkedId) {
                editor = preferences.edit();
                int songid = rgforalarm.getCheckedRadioButtonId();
                if (songid == R.id.marriage){
                    editor.putString(song,song1).commit();
                }
                else if (songid == R.id.air){
                    editor.putString(song,song2).commit();
                }
                else {
                    editor.putString(song,song3).commit();
                }
                //if语句改变歌曲
                dialog.dismiss();
            }
        });
    }

    public void wordselectclick(){
        LayoutInflater inflater = LayoutInflater.from(MainActivity.this);
        wordselectview = inflater.inflate(R.layout.wordselectdialog,null);
        AlertDialog.Builder builder = new
AlertDialog.Builder(MainActivity.this);
        builder.setView(wordselectview);
        final AlertDialog dialog = builder.create();
        cet = (RadioButton) wordselectview.findViewById(R.id.cet);
        toefl = (RadioButton) wordselectview.findViewById(R.id.toefl);
        preferences =
getSharedPreferences("sharedpreference",MODE_PRIVATE);
        String selectword = preferences.getString(worddatabase,database1);
        if (selectword.equals(database1)){
            cet.setChecked(true);

```

```

    }
    else {
        toefl.setChecked(true);
    }
    dialog.show();
    rgforword = (RadioGroup)
wordselectview.findViewById(R.id.rgforword);
    rgforword.setOnCheckedChangeListener(new
RadioGroup.OnCheckedChangeListener() {
        @Override
        public void onCheckedChanged(RadioGroup group, int checkedId) {
            int worddatabaseid = rgforword.getCheckedRadioButtonId();
            editor = preferences.edit();
            if (worddatabaseid == R.id.cet){
                editor.putString(worddatabase,database1).commit();
            }
            else {
                editor.putString(worddatabase,database2).commit();
            }
            //if语句改变词库
            dialog.dismiss();
        }
    });
}
}

```

### 3.3.3 测验模块

#### Alarming.java

功能：到达设定的时间后响铃，弹出测验页面，点击“GO!”button开始测验，点击“EXIT”退出，返回主界面。

较为简单，略去代码展示

#### Exercises.java

功能：进入做题界面，选择正确答案会跳到下一题

**setText()**函数 从单词数据库提取单词作为题目

```

public int setText() {
    if (whether_empty()) {
        return 5;
    } else {
        File file = new File(getFilesDir() + "/databases/newdata.db");
        SQLiteDatabase database =
        SQLiteDatabase.openDatabase(file.getPath(), null,
        SQLiteDatabase.OPEN_READWRITE);
        Cursor cursor = database.rawQuery("select * from " +
        getDatabase() + " order by RANDOM() limit 1", null);
        String word = "",table = "", explains = "";
    }
}

```

```

String[] randomexplains = new String[3];
while (cursor.moveToNext()) {
    word = cursor.getString(0);
    explains = cursor.getString(2);
}
wordquestion.setText(word);
if (getDatabase().equals("cet4")) {
    table = "cetcomplete";
} else {
    table = "toeflcomplete";
}
cursor = database.rawQuery("select * from " + table + " order
by RANDOM() limit 3", null);
int i = 0;
while (cursor.moveToNext()) {
    randomexplains[i] = cursor.getString(2);
    i++;
}
database.delete(getDatabase(), "english=?", new String[] {word});
database.close();
int position = (int) (3 * Math.random() + 1);
if (position == 1) {
    choice1.setText(explains);
    choice2.setText(randomexplains[0]);
    choice3.setText(randomexplains[1]);
    choice4.setText(randomexplains[2]);
} else if (position == 2) {
    choice1.setText(randomexplains[0]);
    choice2.setText(explains);
    choice3.setText(randomexplains[1]);
    choice4.setText(randomexplains[2]);
} else if (position == 3) {
    choice1.setText(randomexplains[0]);
    choice2.setText(randomexplains[1]);
    choice3.setText(explains);
    choice4.setText(randomexplains[2]);
} else {
    choice1.setText(randomexplains[0]);
    choice2.setText(randomexplains[1]);
    choice3.setText(randomexplains[2]);
    choice4.setText(explains);
}
return position;
}
}

```

## FinishExercise.java

功能：提示做题完成，点击“Good Job”退出APP

实现：较为简单，略去代码展示

### 3.3.4 Widget模块

#### WidgetOne.java

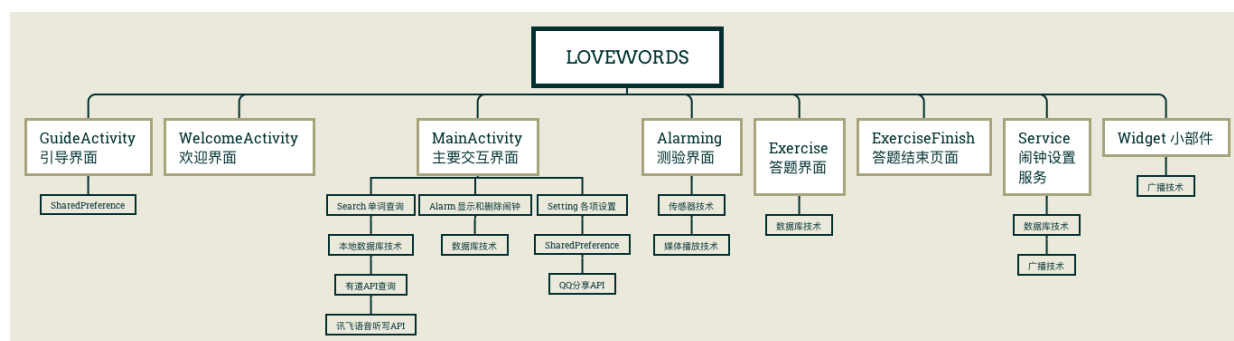
功能：用户点击widget的“Refresh”button或者过一段时间，widget的内容自动更新，充分利用用户零散时间，同时起着唤醒app的功能。

```
public void onUpdate(Context context, AppWidgetManager appWidgetManager,
int[] appWidgetIds) {
    // There may be multiple widgets active, so update all of them
    Intent intent1 = new Intent(context, MyService.class);
    intent1.setAction("来自widget");
    context.startService(intent1);
    Intent intent = new Intent();
    intent.setAction(UPDATE_CONDUCTION);
    context.sendBroadcast(intent);
    PendingIntent pendingIntent =
PendingIntent.getBroadcast(context, 0, intent, 0);
    RemoteViews remoteViews = new
RemoteViews(context.getPackageName(), R.layout.widget_one);
    remoteViews.setOnClickPendingIntent(R.id.newone, pendingIntent);
    appWidgetManager.updateAppWidget(appWidgetIds, remoteViews);
    for (int appWidgetId : appWidgetIds) {
        updateAppWidget(context, appWidgetManager, appWidgetId);
    }
}
```

### 3.3.5 欢迎界面

#### WelcomeActivity.java

## 3.4 代码架构



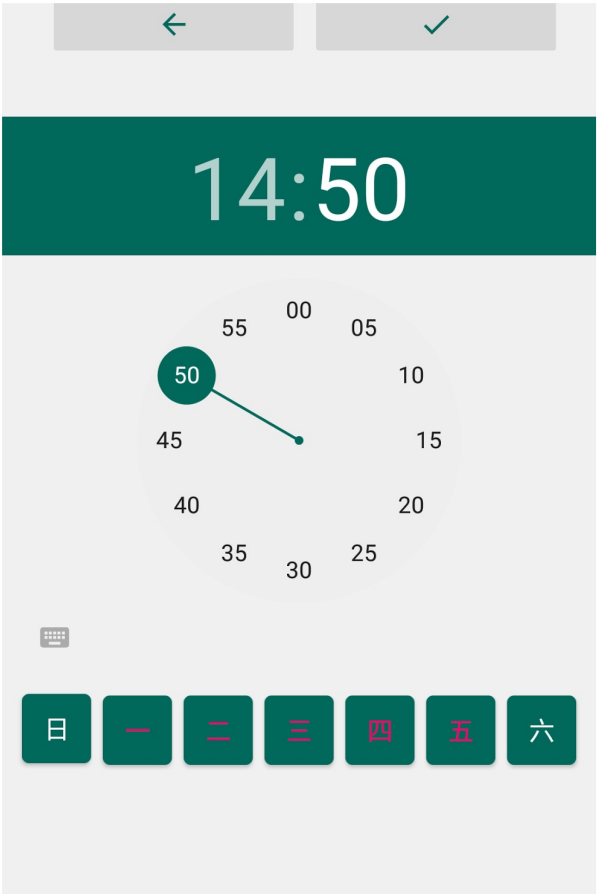
## 第四章 应用效果

### 4.1 欢迎界面（左） 主界面（右）



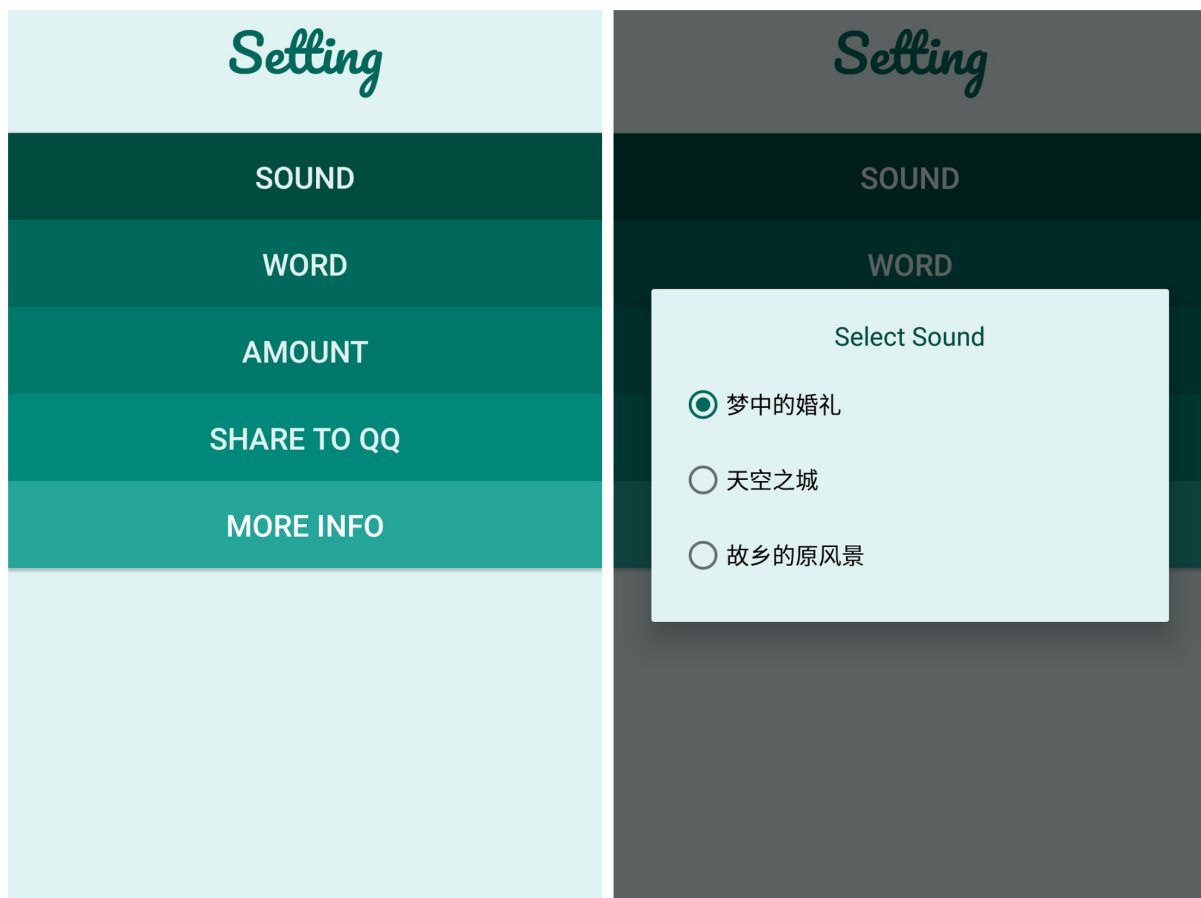
4.2 Alarm界面

添加闹钟并确认

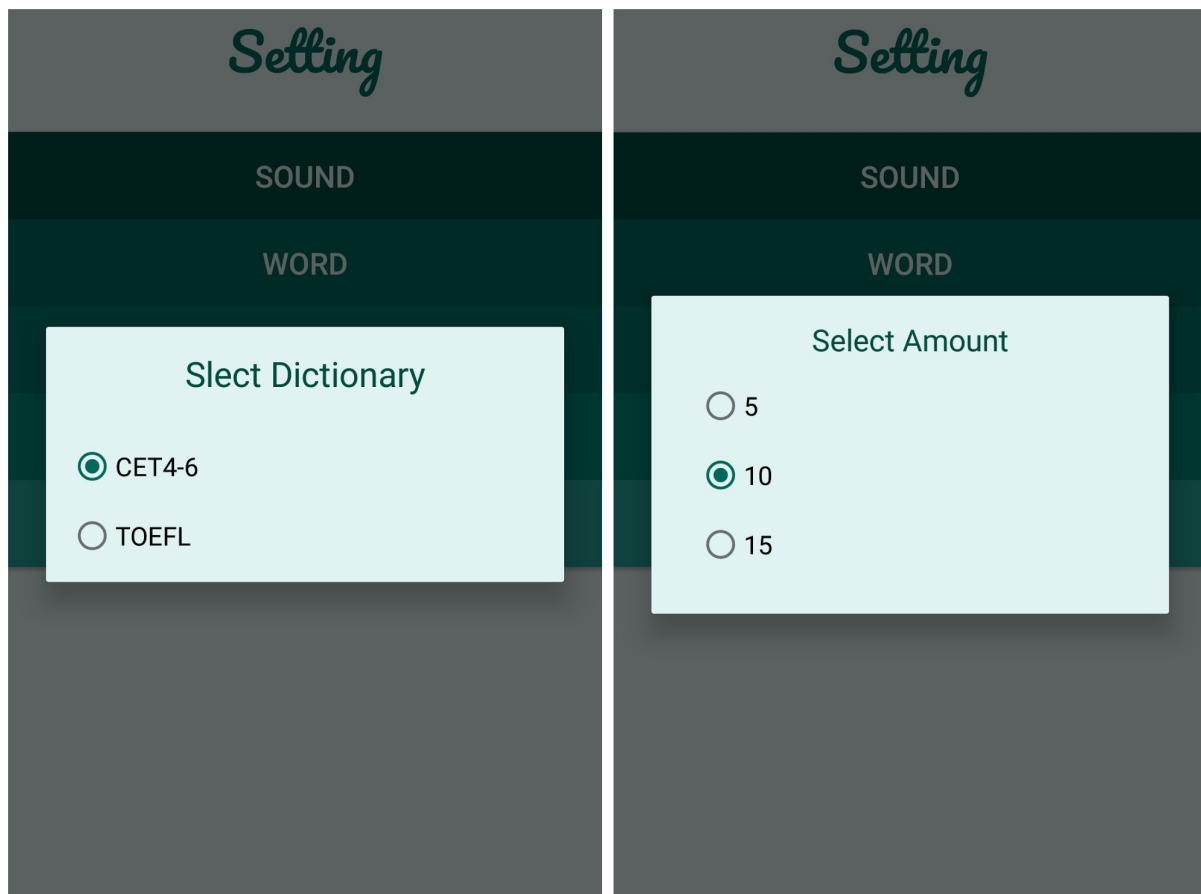


## 4.3 Setting界面

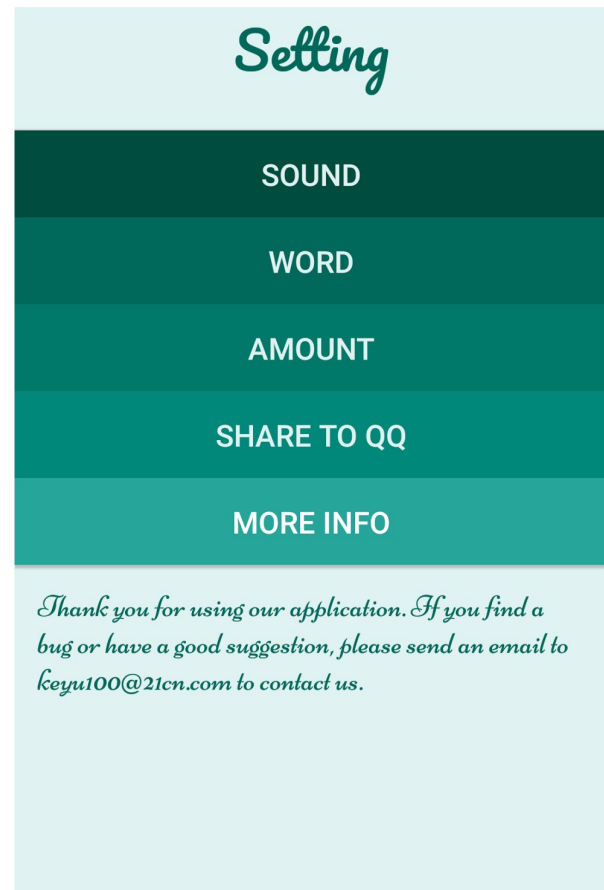
“设置”页面 修改铃声



修改词库（左） 修改每次测验的题数（右）

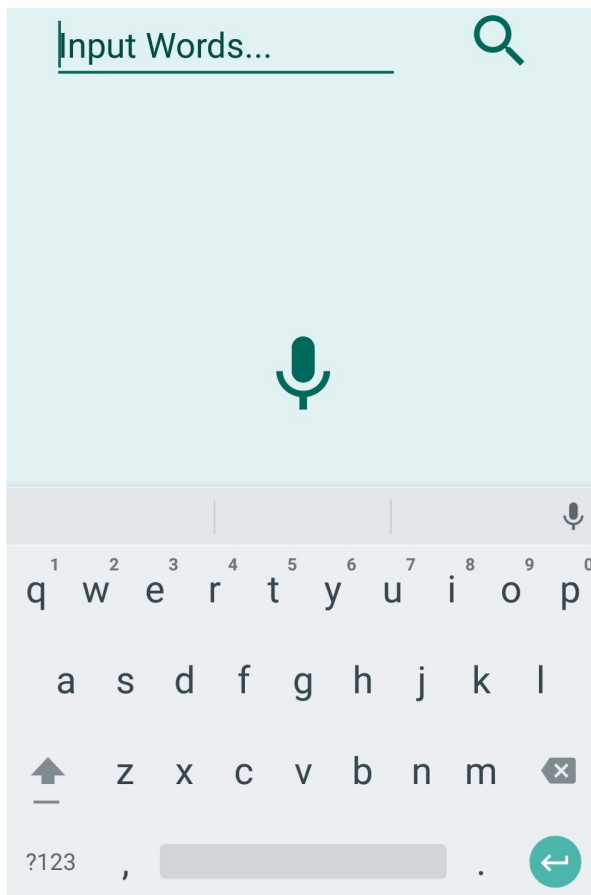
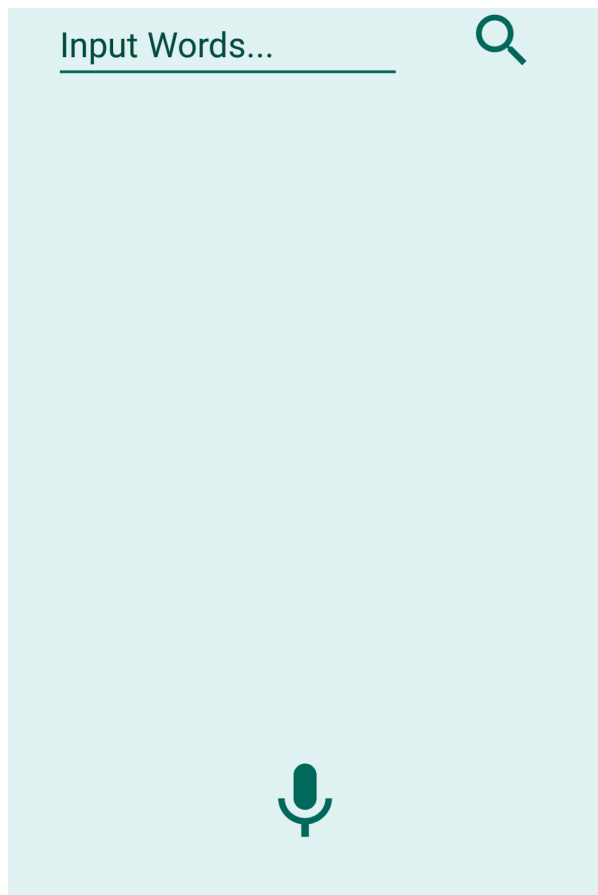


分享应用到QQ（左） 更多信息（右）（注：截图来自android模拟器，未成功安装QQ，在真机上可以成功分享，参见后面的视频演示部分）

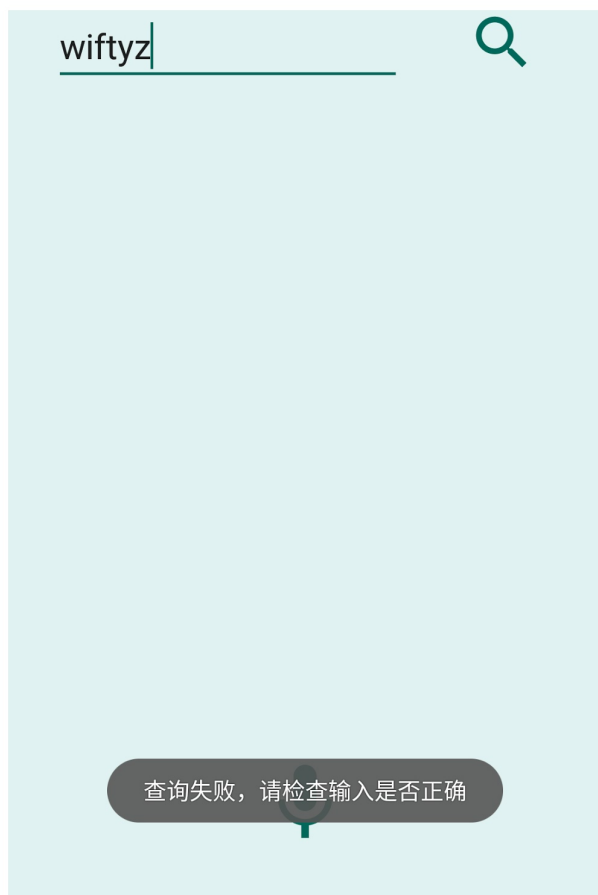


## 4.4 Search界面





查询失败（左） 查询成功（右）



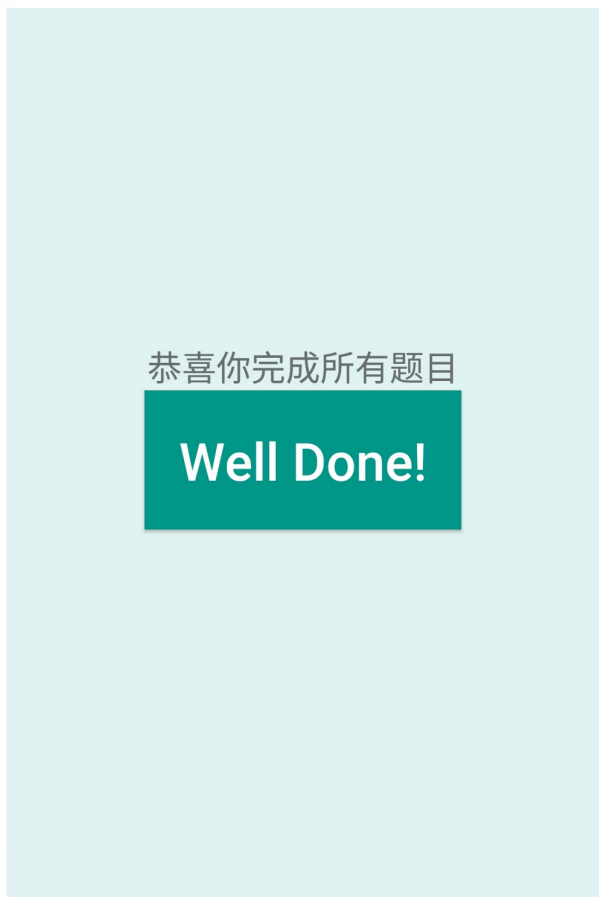
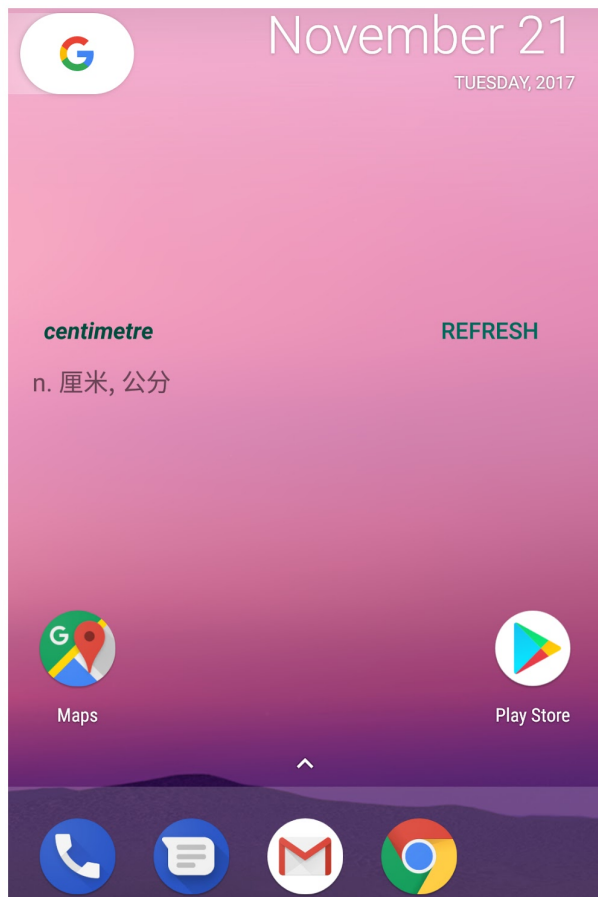
#### 4.5 答题界面

答题选择页面（左） 答题界面（右）



4.6 Widget界面

widget（左）完成答题后的界面（右）



注：以上截图使用了Android Studio 自带的模拟器：Android Emulator-Nexus\_5X\_API\_\_26

#### 4.7视频演示：

链接：<https://vimeo.com/243913060>

二维码：



注：以上视频演示使用的手机型号为：Vivo X7（Android 5.1.1，API22）

## 第五章 技术难点与优化

---

## 5.1 多次闹钟的设置

采用 `setwindow()`。

每次只设置一个闹钟。闹钟响后发广播寻找数据库中离当前时间最近的闹钟去设置。在API19中（本应用最低支持API21）后，Google提供了新的准确传递方式，我采用的是`setwindow()`。

在`AlarmManager`中，`set()`和`setRepeat()`在API19后都是非准确传递的，也就是闹钟可能会提前或延迟，而且`setRepeat()`只能设置间隔固定的重复闹钟，遇到不固定的无法处理。比如说设置周一，周二，周三的多次重复闹钟。比较容易想到的是使用`setRepeat()`来设置三个闹钟，每个闹钟重复间隔为7天。这样的方法有一个很大的弊端，在设置闹钟时，使用广播来触发响铃，使用广播来触发响铃，使用`pendingintent`，则`PendingIntent.getBroadcast(Context context, int requestCode, Intent intent, int flags)`，第二个`requestCode`一定要是唯一的。否则定时器之间会覆盖，导致先前设置的闹钟失效。而且在取消闹钟时也要找到相应的`requestCode`，及其麻烦。在网上没找到很好的解决思路，苦苦思索后想到了`setwindow()`方法的解决方案。

## 5.2 设备重启后处理

设置的闹钟会失效，可能已经过期。

解决方案描述：接受开机广播，SD卡挂载广播，widget更新发送广播，重新设置，过期的则从数据库删除。

这里难点不算大，不过很考验细心。因为一般能确定的是接收开机广播。在当前Android手机中，很多还是支持SD卡的。如果APP是安装在SD卡中，则APP无法接收到开机广播，所以还需要接收SD卡挂载广播。国内的手机系统大部分使用定制后的Android，可能会屏蔽掉开机广播或SD卡挂载广播。本人手机就是这样，但在测试中发现设备重启后widget还是会更新，所以想到可以在widget更新时发送广播来设置闹钟。加上开机广播和SD卡挂载广播，终于解决了设备重启后闹钟的设置。

## 5.3 APP服务保活措施

APP的服务容易被杀，管家类，清理类的APP可能会杀死我们的服务，Android系统自身的LMK（low memory killer）在低内存时也可能杀我们的服务。

解决方案：

1.重写Services 的 `onStartCommand()`函数，将flag设为`START_STICKY()`;

`onStartCommand()`保活只能在我们手机未获取root时有效。

2.在注册文件中将`persistent`设为`true`;

当用户将apk文件放在`system/app`文件夹下，APP将成为系统核心程序，第三方APP和LMK均无法杀死我们的APP。

## 5.4 QQ分享和语音输入

采用QQ分享和科大讯飞的语音听写API。

描述：注册腾讯开发者账号和科大讯飞开发者账号，下载相应SDK，集成到我们的app中。

应用分享功能是很多应用采用的引流方案，我采取了同样的方法，腾讯提供了比较简易的开发接口。语音输入我首选了百度语音的API，不过也正是这里耗费了较多时间，由于百度语音API的开发文档对应的是旧版API，最后选择了科大讯飞。科大讯飞的API功能较为复杂，阅读开发文档耗费了比较多的时间。

## 5.5 用户注册登录（计划中，暂未实现）

采用Android+PHP+MySQL

解决方案描述：Android设备将用户输入的账号密码发送到我们的服务器，运行在服务器的PHP代码连接MySQL数据库，将Android设备上传的数据与数据库的信息对比，向Android设备返回相应的信息（登陆成功，账号密码不匹配/账号不存在）；

采用这样的方式是一个主流选择。因为通过Android直接操作MySQL不仅消耗性能，而且数据安全得不到保障，由于本身没学过PHP，必须进行自学，数据返回格式为json，之前也未学过json格式的解析，需要花费一定时间自学。

用户数据库还有后续功能需要添加，早期关系型数据库的需求还不是很明确，暂时未构建。

## 第六章 总结

本次高级实训是作为毕业设计的一个子部分，实现了比较基础的功能，由于时间和目前的研究还不充分，用户数据库没能建立起来，希望在毕业设计项目的实施过程中作出改进，这次实验注重了基本功能的实现，对于交互流程，界面美观程度的设计规划比较粗糙。工程实训是毕业设计的初始版本，实现的功能比较少，后续将实现下列功能：

1.查询功能：除了基本的词语查询，加入同义词，反义词和例句。2.提醒功能：设置闹钟，到点后提醒用户去完成Quiz任务，四个级别（easy,medium,hard,very hard）根据每个单词的难度分配。通过估计学习者的技能来选择问题。3.推送：定期为用户推送单词，积少成多，帮助记忆。4.用户信息：注册账户，登陆，上传/更改头像，维护用户Quiz的得分状况，5.反馈：（1）时间审判挑战和排名系统，分享学习成果到社交网络。学习者可以满足被他人认同的愿望，并且与其他人产生竞争。学习者技能的可视化，游戏化的实施可以保持学习者的短期动机;然而，为了保持自己的长期动力，学习者最好能够发现学习本身的乐趣。学习者可以通过提高以前的分数来获得满足;进一步，他们可以感受到他们的技能和努力的提高。而且，这个得分也反映在排名中，通过竞争精神提高学习动机。（2）可视化学习者的努力价值和记忆程度，如回答问题的数量，正确答案的数量，以及学习问题的难度。统计学习者在一周内每天获得的积分总和。

## 第七章 参考资料

《Android 5.0 开发范例代码大全》ISBN：978-7-302-39621-5

<http://wiki.open.qq.com/wiki/API%E8%B0%83%E7%94%A8%E8%AF%B4%E6%98%8E>

<http://www.xfyun.cn/doccenter/awd> <https://developer.android.com/index.html>

<https://material.io/>