**实 验 报 告**

**课程名称**  移 动 应 用 系统

**实验项目** 移动平台数据持久化应用设计与开发

**实验仪器** 计算机一台、Android手机一部

**系 别**  软件工程

**专 业** 软件工程

**班级/学号**  软工1801 /2018011192

**学生姓名** 刘宏远

**实验日期**  2020/11/15

**成 绩**

**指导教师**  郝 保 水

# 实验二 移动平台数据持久化应用设计与开发

1. 实验目的
   1. 了解Android数据存储的基本概念；
   2. 理解并掌握SQLite数据库的使用方法;
   3. 进一步熟悉与掌握Fragment使用。
   4. 进一步熟悉各组件、适配器、菜单、对话框等相关概念和技术等。
   5. 掌握ContentProvider相关概念和技术。
   6. 掌握Handler相关概念和技术。

2实验要求

1. 课前预习实验内容，并查找相关资料。
2. 按照实验步骤完成各个相关内容。
3. 撰写实验报告。
   1. 实验报告格式必须符合学校要求（例如必须采用学校规定的实验封面）；
   2. 写出实验详细步骤，包括主要采用的技术方案、相关分析和核心代码。注意：不要简单地近包括截屏和代码，完整代码可以作为附录放在实验报告结尾；
   3. 总结实验中遇到的问题、分析和解决方法。
   4. 写出心得体会与收获等。

3实验步骤

主要分为几个大步骤进行实现。

1. 需求分析，调研现有单词类应用具有哪些功能，确定界面。
2. 系统设计，对系统进行模块划分，确定技术方案。
3. 编码，编写代码实现各项功能。
4. 测试。

3.1需求分析

这里列出主要功能，也可参考其他单词类软件功能。

1. 使用Fragment实现英语词典，当竖屏时显示单词释义，当横屏时显示左右两列，右侧是释义，左侧是单词列表。
2. 单词存放在数据库中，系统预先预置一系列单词，用户可以对单词库进行增加、删除、修改等，并支持查询，包括模糊查询等。
3. 作为ContentProvider，供其它应用程序访问（可以编写一个测试程序验证之）。
4. 用户添加单词或者查询单词时，单词含义和示例等信息可以从网络上获得（推荐有道等）。
5. 可以网络上获取英语新闻，用户可以针对不认识的单词进行查询和添加到单词库（例如在单词上长按，弹出上下文菜单供用户操作）等。
6. 对于不认识的单词，增加到生词本，可以浏览生词本，并提供多种方式背诵这些生单词。

3.2系统设计

模块分为：SQLite数据库的使用，Fragment的使用和选项菜单、上下文菜单。

SQLite数据库的使用：

数据库相关类

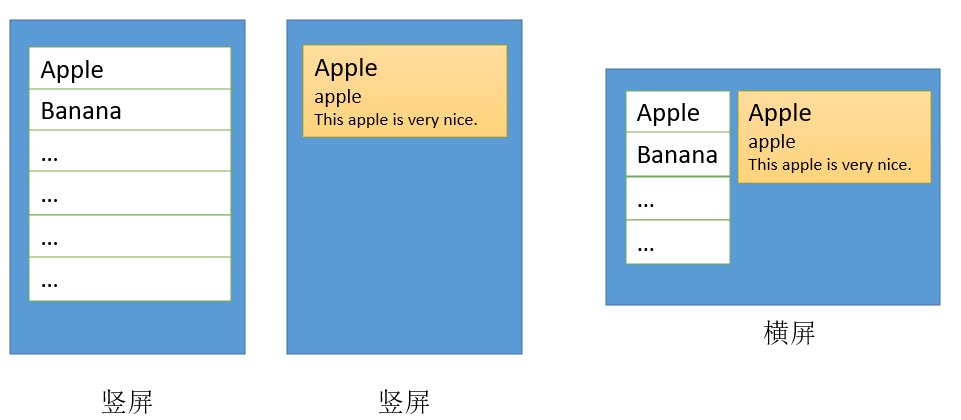
Words：合约类(contract)

WordsDBHelper：数据库辅助类

WordsDB：数据库操作类

单例模式

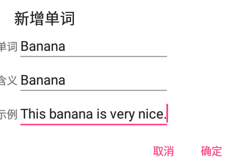
Fragment的使用和选项菜单：



public class WordItemFragment extends ListFragment {  
 private OnFragmentInteractionListener mListener;  
  
 @Override  
 public void onAttach(Context context) {  
 super.onAttach(context);  
 mListener = (OnFragmentInteractionListener) getActivity();  
 }  
  
 @Override  
 public void onDetach() {  
 super.onDetach();  
 mListener = null;  
 }  
   
 /\*\*  
 \* Fragment所在的Activity必须实现该接口，通过该接口Fragment和Activity可以进行通信  
 \*/  
 public interface OnFragmentInteractionListener {  
 public void onWordItemClick(String id);  
 public void onDeleteDialog(String strId);  
 public void onUpdateDialog(String strId);  
 }  
}

//更新单词列表，从数据库中找到同strWord向匹配的单词，然后在列表中显示出来  
public void refreshWordsList(String strWord) {  
 WordsDB wordsDB=WordsDB.getWordsDB();  
 if (wordsDB != null) {  
 ArrayList<Map<String, String>> items = wordsDB.SearchUseSql(strWord);  
 if(items.size()>0){  
  
 SimpleAdapter adapter = new SimpleAdapter(getActivity(), items, R.layout.item,  
 new String[]{Words.Word.\_ID, Words.Word.COLUMN\_NAME\_WORD},  
 new int[]{R.id.textId, R.id.textViewWord});  
  
 setListAdapter(adapter);  
 }else{  
 Toast.makeText(getActivity(),"Not found",Toast.LENGTH\_LONG).show();  
 }  
 }  
}

上下文菜单：



3.3编码实现

对话框：

//删除对话框  
private void DeleteDialog(final String strId) {  
 new AlertDialog.Builder(this).setTitle("删除单词")

.setMessage("是否真的删除单词?")

.setPositiveButton("确定", new DialogInterface.OnClickListener() {  
 @Override  
 public void onClick(DialogInterface dialogInterface, int i) {  
 //既可以使用Sql语句删除，也可以使用使用delete方法删除  
 WordsDB wordsDB=WordsDB.getWordsDB();  
 wordsDB.DeleteUseSql(strId);  
  
 //单词已经删除，更新显示列表  
 RefreshWordItemFragment();  
 }  
 }).setNegativeButton("取消", new DialogInterface.OnClickListener() {  
 @Override  
 public void onClick(DialogInterface dialogInterface, int i) {  
  
 }  
 }).create().show();  
}

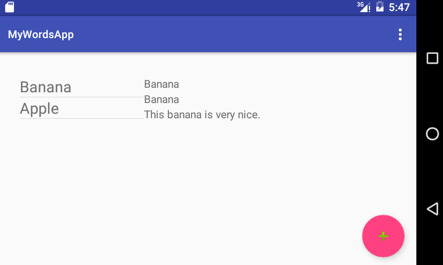
数据库：

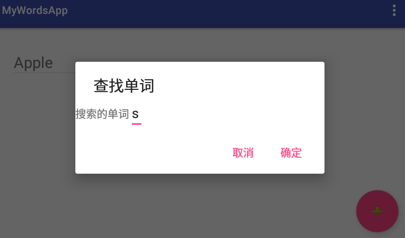
public class WordsDBHelper extends SQLiteOpenHelper {  
 public WordsDBHelper(Context context) {  
 super(context, DATABASE\_NAME, null, DATABASE\_VERSION);  
 }  
  
 @Override  
 public void onCreate(SQLiteDatabase sqLiteDatabase) {  
 //创建数据库  
 sqLiteDatabase.execSQL(SQL\_CREATE\_DATABASE);  
 }  
  
 @Override  
 public void onUpgrade(SQLiteDatabase sqLiteDatabase, int oldVersion, int newVersion) {  
 //当数据库升级时被调用，首先删除旧表，然后调用OnCreate()创建新表  
 sqLiteDatabase.execSQL(SQL\_DELETE\_DATABASE);  
 onCreate(sqLiteDatabase);  
 }  
}

单词添加：

//新增对话框  
private void InsertDialog() {  
 final TableLayout tableLayout = (TableLayout) getLayoutInflater().inflate(R.layout.insert, null);  
 new AlertDialog.Builder(this)  
 .setTitle("新增单词")//标题  
 .setView(tableLayout)//设置视图  
 //确定按钮及其动作  
 .setPositiveButton("确定", new DialogInterface.OnClickListener() {  
 @Override  
 public void onClick(DialogInterface dialogInterface, int i) {  
 String strWord = ((EditText) tableLayout.findViewById(R.id.txtWord)).getText().toString();  
 String strMeaning = ((EditText) tableLayout.findViewById(R.id.txtMeaning)).getText().toString();  
 String strSample = ((EditText) tableLayout.findViewById(R.id.txtSample)).getText().toString();  
  
 //既可以使用Sql语句插入，也可以使用使用insert方法插入  
 // InsertUserSql(strWord, strMeaning, strSample);  
 WordsDB wordsDB=WordsDB.getWordsDB();  
 wordsDB.Insert(strWord, strMeaning, strSample);  
  
 //单词已经插入到数据库，更新显示列表  
 RefreshWordItemFragment();  
 }  
 })  
 //取消按钮及其动作  
 .setNegativeButton("取消", new DialogInterface.OnClickListener() {  
 @Override  
 public void onClick(DialogInterface dialogInterface, int i) { }  
 })  
 .create()//创建对话框  
 .show();//显示对话框  
}

3.4系统测试





4.问题分析（也可以放在实验步骤中写，即写了相关技术或代码后，运行时发现问题，对问题进行分析讨论）

写出问题涉及的相关技术或知识点、问题分析、解决办法等。

数据库导入失败，更换虚拟机解决。

5.实验心得与体会

实验收获、心得、总结等。

内容不少于500字。

收获：学会了Fragment的使用，掌握了android数据库的基本用法。数据库操作类采用单例模式。

心得：通过老师PPT的代码，加上自己的修改和百度，基本完成老师的任务。

总结：要模型分块，将前后端分离，使用MVC构建。

实验报告一定要注意：

1.不要抄袭雷同，一旦发现，不区分抄袭者和被抄袭者，一律0分；

注意，一定不要雷同，不要抄袭！

注意，一定不要雷同，不要抄袭！

注意，一定不要雷同，不要抄袭！

2.格式一定要正确，不要缺少封皮、实验目的、实验内容步骤、心得等；

3，不要代码截屏，我通过github看大家代码，不用为了显得篇幅长而在实验报告中大量粘贴代码；

4.按时提交，过了时间点后系统不允许提交。