



移动终端程序设计

结课报告

班 级：	计算机科学与技术 1802
学 号：	20188068
姓 名：	孔天欣

序号	要求	得分
1	界面要求：美观，友好，不能过于粗糙，具备一定设计美感（20 分）	
2	功能要求：基本功能实现，需要可以对功能进行扩充（20 分）	
2	对数据的持久化保存方式：可以使用文件，也可以使用数据库（20 分）	
4	代码要求：书写规范，结构合理，结合必要的注释（20 分）	
5	文档要求：内容详细，结构合理，格式正确（20 分）	
合计		

评阅人：

东北大学秦皇岛分校计算机与通信工程学院

基于移动端的单词背诵 APP

1 概述

“单词背诵 APP”是一款基于 Android studio 平台，使用 Java 语言开发的，在 Android 系统运行的移动端 APP，它的数据库使用了轻量级的 SQLite。用户可以根据此 APP 实现简单的单词录入、背诵和查阅单词库等功能。该单词背诵 APP 暂未在实际安卓机器中运行，但能够在基于 Windows 10 的 AVD 仿真模拟器 Pixel 2 API 29 编译并运行通过，初步显示了其可行性。此外，该 APP 当前仅实现了基本功能，尚有许多待扩展空间，如单词背诵的用户切换等。

2 总体设计

2.1 功能设计

该单词背诵 APP 主要包含记录单词、背诵单词和查阅单词库三个功能。

1. 记录单词

用户需要通过记录单词功能为单词背诵 APP 的词库中增添新的单词以达到背诵的目的，其中，要求用户记录词汇的格式是单词英文和翻译的组合，单词英文不得包含中文内容，也不得有其他符号。对应的翻译可以为任意文本格式。记录的单词将通过轻量级的数据库 SQLite 存入，并在用户查阅单词库的时候调用数据库中的相关表进行查询和显示。

2. 背诵单词

用户在完成单词的记录后，单词加入到数据库中，这时用户可以使用背诵单词功能来完成单词的记忆。用户背诵的时候可以选择“认识”，“提示”和“不认识”三个选项，其中“认识”选项意为用户已经对这个单词的释义有所了解，而且不需要再次记忆，当用户点击认识之后，该单词会从数据库中删去，以避免用户在后续背诵中再次遇到该单词。“提示”选项意为用户对单词不太了解，因此点击提示后会出现单词的中文释义，此后，用户再点击“不认识”进行下一个单词的记忆，而当前不认识的单词将加入到下一轮的背诵当中，直到用户点击完“认识”后才不会在后续的背诵中再次出现。

3. 查阅单词库

用户可以通过点击查阅单词库的按钮来完成对当前词库中已经记录的单词

数量以及内容的查询。查询的内容有单词的英文和中文释义组成的列表。列表中显示的内容会跟随用户背诵的情况进行动态更新，方便用户了解当前需要背诵以及暂未完全掌握的单词的情况。

1.2 界面设计

该单词背诵 APP 的设计主要包括三个界面，分别是单词记录界面、单词背诵界面和单词库查阅界面。当用户启动 APP 时，首先显示的是单词记录界面。

1. 单词记录界面



图 2.1 单词记录界面

该单词背诵 APP 的记录界面如图 2.1 所示。可见，该界面主要包含下侧的主按钮“记录单词”，“背诵单词”和“查询词库”。用户在点击对应的主按钮时，该按钮会将形态和里面的字体放大，使得用户了解自己当前正在访问的界面。此外，还有单词记录界面主体的按钮“保存”，用户需要在“输入单词”框中输入对应的单词，并在“输入翻译”框中输入对应的翻译，然后点击“保存”按钮，

将单词和翻译的数据发送给后台进行数据库插入记录处理。此外，该界面的最上侧所带有的“书山有路勤为径，学海无涯苦作舟”的横幅，在所有界面均处于置顶的优先级，用以激励用户持之以恒完成单词的学习任务。

2. 单词背诵界面



图 2.2 单词背诵界面

如图 2.2 所示，单词背诵界面主要包含显示单词和显示翻译的视图框，其中，显示单词的视图框并没有设置特殊的背景，同时单词的字体特意进行了放大处理，方便用户第一眼就能看到单词主体。此外，还将“认识”，“提示”和“不认识”三个按钮放在整体区域的下侧，位于主按钮的上侧，方便用户进行点击。通过上述操作，即可加快用户的单词背诵速度和软件使用舒适度。

此外，当用户点击提示后，翻译的视图框便会显示翻译的内容，如图 1.3 所示。



图 2.3 用户点击提示界面

3. 词库查询界面

用户通过点击“查询词库”按钮，即可进入单词库查询界面，该界面设计简洁直观，直接一行行展示了用户此前输入的单词词库列表，如图 2.4 所示。

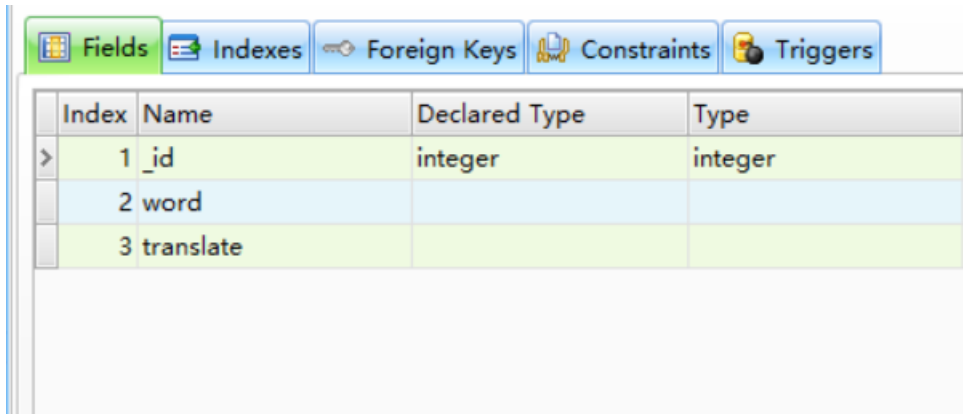


图 2.4 查阅单词库界面

3 详细设计

3.1 持久化设计

该单词 APP 包含的数据库表主要有 `tb_wd`，如图 3.1 所示，该表中包含了三个字段，`_id`，`word` 和 `translate`，其中 `word` 字段是单词的英文词汇，`translate` 字段是单词的中文释义，由于数据格式合法性的判断已经在业务层完成，因此无需在数据库中增添更多的格式约束，也有利于提高整体性能。



Index	Name	Declared Type	Type
1	_id	integer	integer
2	word		
3	translate		

图 3.1 数据库单词设计表

3.2 功能设计

1. 持久化代码

通过使用 `sqlite` 数据库即可进行单词的添加和删除操作，如代码清单 3.1 和 3.2 所示，可见将安卓前端发送至服务端的词汇数据 `word` 和 `translate` 进行了数据库的添加和删除操作处理。

代码清单 3.1 数据增加代码

```
public void writeData(SQLiteDatabase sqLiteDatabase,String word,String translate){  
    ContentValues values = new ContentValues();  
    values.put("word",word);  
    values.put("translate",translate);  
    sqLiteDatabase.insert("tb_wd",null,values);  
}
```

代码清单 3.2 词汇删除代码

```
public void deleteData(SQLiteDatabase sqLiteDatabase,String word,String translate){  
    ContentValues values = new ContentValues();
```

```
        values.put("word",word);

        values.put("translate",translate);

        String[] wordsArr = {word};

        sqLiteDatabase.delete("tb_dict","word=?",wordsArr);

    }
```

2. 单词的保存工作

用户在单词记录界面输入词汇后点击保存按钮，在后台即触发 `click` 监听器捕获，并调用对应的 `onClick` 方法，如代码清单 3.3 所示，该方法收集了前台传输的 `word` 和 `translate` 字段，并进行了格式检查（非空），如果发现格式不符，则在调用 `print` 方法在安卓前台弹出提示，提醒用户进行正确格式的输入。如果用户输入的格式通过了检查，那么会调用 `wordMapper`（即 `word` 表对应的持久层交互类，以进行和 `word` 表相关的增删查改操作）的 `writeData` 方法（见代码清单 2.1），其中 `wordMapper` 还调用了 `getReadableDatabase` 方法，该方法返回可读数据库的句柄。

代码清单 3.3 “保存”按钮的点击触发方法

```
button.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {
        String word = wordText.getText().toString();
        String translate = translateText.getText().toString();
        if (word.isEmpty() || translateText.getText().toString().isEmpty()){
            ToastUtil.print(getActivity(),"空");
        }else{
            wordMapper.writeData(wordMapper.getReadableDatabase(),word,translate);

            ToastUtil.print(getActivity(),"保存");
            translateText.setText("");
            wordText.setText("");
            wordText.requestFocus();
            wordText.setHeight(WRAP_CONTENT);
        }
    }
});
```

3. 单词背诵算法

本单词背诵 APP 的单词背诵算法如代码清单 3.4 所示，该背诵算法在用户点

击单词背诵按钮时触发。首先先从视图中获取用户点击的情形。如果是 **rk**，即用户点击了认识按钮，那么先检查词书当前的背诵情况，如果已经背诵完毕（即从数据库中获得的 **wordList** 为空），那么弹出提示“背诵完毕”；如果尚未背诵完毕，就从词书中移除当前单词，并置空翻译框和单词框内的文本，然后进行数据库的更新操作。如果用户点击“提示”，便会触发 **rpt**，那么便调用先前从数据库提取记录的翻译字段，通过调用 **setText** 方法在翻译框内显示该单词对应的中文释义。如果用户点击不认识，即 **rdn**，那么便显示下一个单词，该单词继续加入重复记忆列表中。

代码清单 3.4 单词背诵算法

```
@Override
public void onClick(View view) {
    if(wordsList.isEmpty()){
        ToastUtil.print(getActivity(), "词书是空的 请录入单词");
        return;
    }
    switch (view.getId()) {
        case R.id.rk: {
            if (wordsList.size() == 0){
                ToastUtil.print(getActivity(), "背诵完毕");
            }else{
                wordsList.remove(i);
                word.setText("");
                translate.setText("");
            }
            wordMapper.onUpgrade(wordMapper.getReadableDatabase(), 0, 0);
            for (Word word : wordsList) {
                wordMapper.writeData(wordMapper.getReadableDatabase(),
word.word, word.translate);
            }
            break;
        }
        case R.id.rpt:{
            if(wordsList.size()==0){
                ToastUtil.print(getActivity(), "背诵完毕");
            }else{
                translate.setText(current_translation);
                return;
            }
            break;
        }
    }
}
```

```
        case R.id.rdn:{
            i == wordsList.size()-1?0:i+1;
            word.setText(getCurrent_txt(i));
            translate.setText("");
            break;
        }
    }
}
```

4 总结

通过本次结课项目的编写，本人成功掌握了移动端开发的基本流程和方法，填补了本人对于移动端开发方面了解的空白。并能够根据掌握到的技术开发出功能简单、实用的单词背诵 APP，尽管当前 APP 功能较少，但仍然具有较大的可拓展空间。同时本人可能在未来深入学习移动端开发技术栈后，对其进行更为优质的迭代。