2018 －2019 学年第 二 学期

组号：G02



霸笔笔记APP开发

代码清单



实验课程名称 软件工程导论

专 业 班 级 软工1701 软工1702

学 号 31701401 31701257 31701371

学 生 姓 名 高兴欣 王晨旭 倪嘉玲

实验指导老师 杨枨

|  |  |  |
| --- | --- | --- |
| 文件状态： | 文件标识： | SE-2019-G02 |
| [ ]:草稿 | 当前版本： | 1.5 |
| [ ]:正式发布 | 作 者： | 高兴欣、王晨旭、倪嘉玲 |
| [ ]:正在修改 | 完成日期： | 2019年5月25日 |
| [ √ ]:正在修改 | 完成日期： | 2019年5月25日 |

# 版本历史

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **版本/状态** | **作者** | **参与者** | **起止日期** | **备注** |
| 1.0 | 高兴欣、王晨旭、倪嘉玲 | 高兴欣、王晨旭、倪嘉玲 | 2019-05-17  至  2019-05-25 | 初步完成代码清单 |
| 2.0 | 高兴欣、王晨旭、倪嘉玲 | 高兴欣、王晨旭、倪嘉玲 | 2019-05-31  至  2019-06-25 | 修订代码清单 |

目录

[版本历史 2](#_Toc10317897)

[1. 数据库端代码 11](#_Toc10317898)

[1.1功能函数 11](#_Toc10317899)

[①查询用户密码模块 11](#_Toc10317900)

[②注册账号模块 12](#_Toc10317901)

[③修改用户信息模块 14](#_Toc10317902)

[④修改用户密码模块 15](#_Toc10317903)

[⑤查询用户信息模块 16](#_Toc10317904)

[⑥初始化表格 18](#_Toc10317905)

[⑧清空表格数据 21](#_Toc10317906)

[⑨添加数据 21](#_Toc10317907)

[⑬获取所有数据 30](#_Toc10317908)

[1.2数据库链接函数 34](#_Toc10317909)

[①链接数据库 34](#_Toc10317910)

[②数据库驱动函数-查询 35](#_Toc10317911)

[③数据库驱动函数-登录和注册 39](#_Toc10317912)

[1.3后端数据传输函数 45](#_Toc10317913)

[①登录数据函数 45](#_Toc10317914)

[②注册数据函数 49](#_Toc10317915)

[⑭修改用户信息模块 54](#_Toc10317916)

[⑮修改用户密码模块 59](#_Toc10317917)

[1.3定义函数 64](#_Toc10317918)

[①添加笔记定义 64](#_Toc10317919)

[②添加反馈定义 67](#_Toc10317920)

[③添加用户定义 68](#_Toc10317921)

[2. 前端代码 70](#_Toc10317922)

[2.1连接服务器配置代码 70](#_Toc10317923)

[2.2功能函数 74](#_Toc10317924)

[①添加笔记 74](#_Toc10317925)

[③删除单词函数 83](#_Toc10317926)

[③登录函数 88](#_Toc10317927)

[⑤改密码函数 93](#_Toc10317928)

[⑥个人中心函数 99](#_Toc10317929)

[⑨反馈函数 109](#_Toc10317930)

[⑩编辑笔记函数 120](#_Toc10317931)

[⑪**分**类管理函数 131](#_Toc10317932)

[⑫**list操作** 138](#_Toc10317933)

[⑬分类转换函数 151](#_Toc10317934)

[⑭**笔记转换模块** 155](#_Toc10317935)

[⑲当前用户信息函数 157](#_Toc10317936)

[2.3 UI界面代码 158](#_Toc10317937)

[① 分类编辑界面 158](#_Toc10317938)

[② 帮助界面 158](#_Toc10317939)

[③ 删除模式界面 160](#_Toc10317940)

[⑤ 登录界面 161](#_Toc10317941)

[⑥ 改密码界面 166](#_Toc10317942)

[⑦ 个人中心界面 171](#_Toc10317943)

[⑧ 管理员界面 178](#_Toc10317944)

[⑪菜单管理界面 182](#_Toc10317945)

[⑫笔记列表管理界面 182](#_Toc10317946)

[⑬文件夹添加页面 183](#_Toc10317947)

[⑭文件夹编辑界面 185](#_Toc10317948)

[⑮笔记输入界面 186](#_Toc10317949)

[⑯主界面 189](#_Toc10317950)

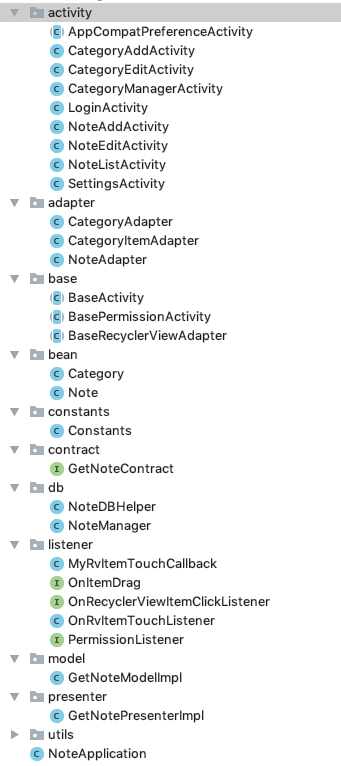
[⑱设置界面 195](#_Toc10317951)

[⑳用户反馈界面 199](#_Toc10317952)

[㉑注册界面 201](#_Toc10317953)

[㉒主菜单界面 211](#_Toc10317954)

**一、设计说明**



activity下存放着各个页面的操作函数。

adapter下存放着各个接口函数

base下存放着基础类函数

bean下存放着各个数据的构造器

constant下存放的是最开始显示的示例

contract下存放获取笔记信息的接口

db下存放的是笔记的操作函数

listener下存放的是对各个部分的监听函数

model下存放的是将各个文件夹下的事项在所有类里显示的函数

presenter下存放着实现contract下接口的函数

utils下存放着重写oncreate函数



drawable存放了图片资源

layout下存放着主界面布局文件

menu下存放着菜单布局文件

各个mipmap文件下存放着各个按钮的图片

value下存放着颜色字体等文件

**二、源代码清单**

1. 数据库端代码

1.1功能函数

①查询用户密码模块

public String chaUsermima(String UserAccount){

java.sql.Connection conn=null;

try{

conn=DBUtil.getConnection();

String sql="select \* from info\_user where UserAccount=?";

java.sql.PreparedStatement pst = conn.prepareStatement(sql);

pst.setString(1,UserAccount);

java.sql.ResultSet rs = pst.executeQuery();

if(!rs.next()) return "fail";//throw new BusinessException("不存在此账号");

return rs.getString(2);

}catch(SQLException e){

e.printStackTrace();

}

//查询用户密码模块

return "fail";

}

### ②注册账号模块

public String addUser(String UserAccount, String pwd,String pwd2,String UserEmail,int UserSelect){

java.sql.Connection conn=null;

try{

conn=DBUtil.getConnection();

String sql="select \* from info\_user where UserAccount=?";

java.sql.PreparedStatement pst = conn.prepareStatement(sql);

pst.setString(1,UserAccount);

java.sql.ResultSet rs = pst.executeQuery();

if(rs.next()) return "fail"; //账号已存在

rs.close();

pst.close();

if(pwd.equals(pwd2)==false) return "fail"; //两次账号不一致

if(UserAccount==null||"".equals(UserAccount)) return "fail"; //账号不得为空

if(pwd==null||"".equals(pwd)) return "fail"; // 密码不得为空

sql="insert into info\_user(UserAccount,UserPassword,UserEmail,UserSelect)values(?,?,?,?)";

pst = conn.prepareStatement(sql);

pst.setString(1, UserAccount);

pst.setString(2, pwd);

pst.setString(3, UserEmail);

pst.setInt(4, UserSelect);

pst.execute();

}catch(SQLException e){

e.printStackTrace();

}

//注册账号

return "success";

}

### ③修改用户信息模块

public String updUser(String UserAccount,String UserEmail,int UserSelect,String UserSignature){

java.sql.Connection conn=null;

try{

conn=DBUtil.getConnection();

String sql="update info\_user set UserEmail=?,UserSelect=?,UserSignature=? where UserAccount=?";

java.sql.PreparedStatement pst=conn.prepareStatement(sql);

pst=conn.prepareStatement(sql);

pst.setString(1,UserEmail);

pst.setInt(2,UserSelect);

pst.setString(3,UserSignature);

pst.setString(4,UserAccount);

pst.executeUpdate();

return "success";

}catch(SQLException e){

e.printStackTrace();

}

//修改用户信息模块

return "fail";

}

### ④修改用户密码模块

public String updUserPaw(String UserAccount,String NewPassword){

java.sql.Connection conn=null;

try{

conn=DBUtil.getConnection();

String sql="update info\_user set UserPassword=? where UserAccount=?";

java.sql.PreparedStatement pst=conn.prepareStatement(sql);

pst=conn.prepareStatement(sql);

pst.setString(1,NewPassword);

pst.setString(2,UserAccount);

pst.executeUpdate();

return "success";

}catch(SQLException e){

e.printStackTrace();

}

//修改用户密码模块

return "fail";

}

### ⑤查询用户信息模块

public User chaUser(String UserAccount){

User user = new User();

java.sql.Connection conn=null;

try{

conn=DBUtil.getConnection();

String sql="select \* from info\_user where UserAccount=?";

java.sql.PreparedStatement pst=conn.prepareStatement(sql);

pst.setString(1,UserAccount);

java.sql.ResultSet rs = pst.executeQuery();

if(rs.next()){

user.setUserAccount(rs.getString(1));

user.setUserPassword(rs.getString(2));

user.setUserEmail(rs.getString(3));

user.setUserSelect(rs.getInt(4));

user.setUserSignature(rs.getString(5));

return user;

}

}catch(SQLException e){

e.printStackTrace();

}

//查询用户信息模块

return user;

}

### ⑥初始化表格

**public void** initTable(){  
 SPUtil.*putString*(**mContext**,**"category"**,**"所有"**);  
 **try** {  
 **database**=**noteDBHelper**.getWritableDatabase();  
 **database**.beginTransaction();  
 String[] contents=***NoteProfile***;  
 **for** (**int** i=0;i<contents.**length**;i++){  
 ContentValues values=**new** ContentValues();  
 values.put(**"id"**,String.*valueOf*(i+1));  
 values.put(**"content"**,contents[i]);  
 Date date=**new** Date();  
 SimpleDateFormat sdf=**new** SimpleDateFormat(**"MM月dd日"**);  
 String firstTime=sdf.format(date);  
 values.put(**"firsttime"**,firstTime);  
 values.put(**"lasttime"**,Long.*toString*(date.getTime()));  
 values.put(**"category"**,**"所有"**);*//* **database**.insert(**noteDBHelper**.***TABLE\_NAME***,**null**,values);  
 }  
 SPUtil.*putInt*(**mContext**,**"noteId"**,contents.**length**);  
 ContentValues values=**new** ContentValues();  
 values.put(**"id"**,1);  
 values.put(**"category"**,**"所有"**);*//所有--1* values.put(**"pos"**,String.*valueOf*(0));  
 **database**.insert(**noteDBHelper**.***TABLE\_CATEGORY***,**null**,values);  
 values.clear();  
 values.put(**"id"**,2);  
 values.put(**"category"**,**"日常"**);*//所有--1* values.put(**"pos"**,String.*valueOf*(1));  
 **database**.insert(**noteDBHelper**.***TABLE\_CATEGORY***,**null**,values);  
 SPUtil.*putInt*(**mContext**,**"categoryNum"**,2);*//category已有一个“所有”分类和一个"日常"分类* **database**.setTransactionSuccessful();  
 }**catch** (SQLException e){  
 e.printStackTrace();  
 }**finally** {  
 **if** (**database**!=**null**){  
 **database**.endTransaction();  
 **database**.close();  
 }  
 }  
}

⑦判断表格是否为空

**public boolean** isEmpty(){  
 Cursor cursor=**null**;  
 **try** {  
 **database**=**noteDBHelper**.getReadableDatabase();  
 cursor=**database**.query(NoteDBHelper.***TABLE\_NAME***,  
 **new** String[]{**"count(id)"**},  
 **null**,**null**,**null**,**null**,**null**);  
 **if** (cursor.moveToFirst()){  
 **if** (cursor.getInt(0)>0){  
 **return false**;  
 }  
 }  
 }**catch** (Exception e){  
 e.printStackTrace();  
 }**finally** {  
 **if** (cursor!=**null**){  
 cursor.close();  
 }  
 **if** (**database**!=**null**){  
 **database**.close();  
 }  
 }  
 **return true**;  
}

### ⑧清空表格数据

**public void** clearTable(){  
 **try** {  
 **database**=**noteDBHelper**.getWritableDatabase();  
 **database**.delete(**noteDBHelper**.***TABLE\_NAME***,**null**,**null**);  
 }**catch** (Exception e){  
 e.printStackTrace();  
 }**finally** {  
 **if** (**database**!=**null**){  
 **database**.close();  
 }  
 }  
}

### ⑨添加数据

**public void** insert(Note note){  
 **try** {  
 **database**=**noteDBHelper**.getWritableDatabase();  
 **database**.beginTransaction();  
 ContentValues values=**new** ContentValues();  
 values.put(**"id"**,Integer.*parseInt*(note.getId()));*//parseInt* values.put(**"content"**,note.getContent());  
 values.put(**"firsttime"**,note.getFirstTime());  
 values.put(**"lasttime"**,note.getLastTime());  
 values.put(**"category"**,note.getCategory());  
 **database**.insertOrThrow(**noteDBHelper**.***TABLE\_NAME***,**null**,values);  
 **database**.setTransactionSuccessful();  
 }**catch** (SQLiteConstraintException e){  
 Toast.*makeText*(**mContext**, **"主键重复"**, Toast.***LENGTH\_SHORT***).show();  
 }**finally** {  
 **if** (**database**!=**null**){  
 **database**.endTransaction();  
 **database**.close();  
 }  
 }  
}  
  
*/\*\*  
 \* insert category的时候记得把position的int型转string型  
 \** ***@param category*** *\*/***public void** insertCategory(Category category){  
 **try** {  
 **database**=**noteDBHelper**.getWritableDatabase();  
 **database**.beginTransaction();  
 ContentValues values=**new** ContentValues();  
 values.put(**"id"**,Integer.*parseInt*(category.getId()));*//parseInt* values.put(**"category"**,category.getCategory());  
 values.put(**"pos"**,category.getPos());  
 **database**.insertOrThrow(**noteDBHelper**.***TABLE\_CATEGORY***,**null**,values);  
 **database**.setTransactionSuccessful();  
 }**catch** (SQLiteConstraintException e){  
 Toast.*makeText*(**mContext**, **"主键重复"**, Toast.***LENGTH\_SHORT***).show();  
 }**finally** {  
 **if** (**database**!=**null**){  
 **database**.endTransaction();  
 **database**.close();  
 }  
 }  
}

⑩删除数据

**public void** delete(String id){  
 **try** {  
 **database**=**noteDBHelper**.getWritableDatabase();  
 **database**.beginTransaction();  
 **database**.delete(**noteDBHelper**.***TABLE\_NAME***,**"id=?"**,**new** String[]{String.*valueOf*(id)});  
 **database**.setTransactionSuccessful();  
 }**catch** (Exception e){  
 e.printStackTrace();  
 }**finally** {  
 **if** (**database**!=**null**){  
 **database**.endTransaction();  
 **database**.close();  
 }  
 }  
}  
**public void** deleteCategory(String id){  
 **try** {  
 **database**=**noteDBHelper**.getWritableDatabase();  
 **database**.beginTransaction();  
 **database**.delete(**noteDBHelper**.***TABLE\_CATEGORY***,**"id=?"**,**new** String[]{String.*valueOf*(id)});  
 **database**.setTransactionSuccessful();  
 }**catch** (Exception e){  
 e.printStackTrace();  
 }**finally** {  
 **if** (**database**!=**null**){  
 **database**.endTransaction();  
 **database**.close();  
 }  
 }  
}

⑪修改数据

**public void** update(Note note){  
 **try** {  
 **database**=**noteDBHelper**.getWritableDatabase();  
 **database**.beginTransaction();  
 ContentValues values=**new** ContentValues();  
 **if** (!TextUtils.*isEmpty*(note.getContent())){  
 values.put(**"content"**,note.getContent());  
 }  
 **if** (!TextUtils.*isEmpty*(note.getLastTime())){  
 values.put(**"lasttime"**,note.getLastTime());  
 }  
 **database**.update(**noteDBHelper**.***TABLE\_NAME***,values,**"id=?"**,  
 **new** String[]{String.*valueOf*(note.getId())});  
 **database**.setTransactionSuccessful();  
 }**catch** (Exception e){  
 e.printStackTrace();  
 }**finally** {  
 **if** (**database**!=**null**){  
 **database**.endTransaction();  
 **database**.close();  
 }  
 }  
}  
**public void** updateCategory(Category category){  
 **try** {  
 **database**=**noteDBHelper**.getWritableDatabase();  
 **database**.beginTransaction();  
 ContentValues values=**new** ContentValues();  
 **if** (!TextUtils.*isEmpty*(category.getCategory())){  
 values.put(**"category"**,category.getCategory());  
 }  
 **if** (!TextUtils.*isEmpty*(category.getPos())){  
 values.put(**"pos"**,category.getPos());  
 }  
 *//根据id来update* **database**.update(**noteDBHelper**.***TABLE\_CATEGORY***,values,**"id=?"**,  
 **new** String[]{category.getId()});  
 **database**.setTransactionSuccessful();  
 }**catch** (Exception e){  
 e.printStackTrace();  
 }**finally** {  
 **if** (**database**!=**null**){  
 **database**.endTransaction();  
 **database**.close();  
 }  
 }  
}

⑫按条件查找

**public** List<Note> select(Note note){  
 Cursor cursor=**null**;  
 **try** {  
 **database**=**noteDBHelper**.getReadableDatabase();  
 **if** (!TextUtils.*isEmpty*(note.getId())){  
 cursor=**database**.query(**noteDBHelper**.***TABLE\_NAME***,**TABLE\_COLUMNS**,  
 **"id=?"**,**new** String[]{note.getId()},**null**,**null**,**null**);*//三个null* **if** (cursor.getCount()>0){  
 List<Note>list=**new** ArrayList<>(cursor.getCount());  
 **while** (cursor.moveToNext()){  
 list.add(parseNote(cursor));  
 }  
 **return** list;  
 }  
 }  
 }**catch** (Exception e){  
 e.printStackTrace();  
 }**finally** {  
 **if** (cursor!=**null**){  
 cursor.close();  
 }  
 **if** (**database**!=**null**){  
 **database**.close();  
 }  
 }  
 **return null**;  
}  
  
*/\*\*  
 \* 通过 string category查询note  
 \** ***@param category*** *\** ***@return*** *\*/***public** List<Note> selectCategory(String category){  
 Cursor cursor=**null**;  
 **try** {  
 **database**=**noteDBHelper**.getReadableDatabase();  
 **if** (!TextUtils.*isEmpty*(category)){  
 cursor=**database**.query(**noteDBHelper**.***TABLE\_NAME***,**TABLE\_COLUMNS**,  
 **"category=?"**,**new** String[]{category},**null**,**null**,**null**);*//三个null* **if** (cursor.getCount()>0){  
 List<Note>list=**new** ArrayList<>(cursor.getCount());  
 **while** (cursor.moveToNext()){  
 list.add(parseNote(cursor));  
 }  
 **return** list;  
 }  
 }  
 }**catch** (Exception e){  
 e.printStackTrace();  
 }**finally** {  
 **if** (cursor!=**null**){  
 cursor.close();  
 }  
 **if** (**database**!=**null**){  
 **database**.close();  
 }  
 }  
 **return null**;  
}

### ⑬获取所有数据

**public** List<String> selectAllCategory(){  
 Cursor cursor=**null**;  
 **try** {  
 **database**=**noteDBHelper**.getReadableDatabase();  
 cursor=**database**.query(**noteDBHelper**.***TABLE\_CATEGORY***,**CATEGORY\_COLUMNS**,**null**,**null**,**null**,**null**,**null**);*//五个null* **if** (cursor.getCount()>0){  
 List<String>list=**new** ArrayList<>(cursor.getCount());  
 **while** (cursor.moveToNext()){  
 list.add(cursor.getString(cursor.getColumnIndex(**"category"**)));  
 }  
 **return** list;  
 }  
 }**catch** (Exception e){  
 e.printStackTrace();  
 }**finally** {  
 **if** (cursor!=**null**){  
 cursor.close();  
 }  
 **if** (**database**!=**null**){  
 **database**.close();  
 }  
 }  
 **return null**;*//return null*}  
  
*/\*\*  
 \* 查询表格全部note  
 \** ***@return*** *\*/***public** List<Note>selectAll(){  
 Cursor cursor=**null**;  
 **try** {  
 **database**=**noteDBHelper**.getReadableDatabase();  
 cursor=**database**.query(**noteDBHelper**.***TABLE\_NAME***,**TABLE\_COLUMNS**,**null**,**null**,**null**,**null**,**null**);*//五个null* **if** (cursor.getCount()>0){  
 List<Note>list=**new** ArrayList<>(cursor.getCount());  
 **while** (cursor.moveToNext()){  
 list.add(parseNote(cursor));  
 }  
 **return** list;  
 }  
 }**catch** (Exception e){  
 e.printStackTrace();  
 }**finally** {  
 **if** (cursor!=**null**){  
 cursor.close();  
 }  
 **if** (**database**!=**null**){  
 **database**.close();  
 }  
 }  
 **return null**;*//return null*}  
**public** List<Category>selectAllCategoryBean(){  
 Cursor cursor=**null**;  
 **try** {  
 **database**=**noteDBHelper**.getReadableDatabase();  
 cursor=**database**.query(**noteDBHelper**.***TABLE\_CATEGORY***,**CATEGORY\_COLUMNS**,**null**,**null**,**null**,**null**,**null**);*//五个null* **if** (cursor.getCount()>0){  
 List<Category>list=**new** ArrayList<>(cursor.getCount());  
 **while** (cursor.moveToNext()){  
 list.add(parseCategory(cursor));  
 }  
 **return** list;  
 }  
 }**catch** (Exception e){  
 e.printStackTrace();  
 }**finally** {  
 **if** (cursor!=**null**){  
 cursor.close();  
 }  
 **if** (**database**!=**null**){  
 **database**.close();  
 }  
 }  
 **return null**;*//return null*}

## 1.2数据库链接函数

### ①链接数据库

public String updGG(String inf){

public class DBUtil {

private static final String jdbcUrl="jdbc:mysql://47.106.110.34:3306/dcapp";

private static final String dbUser="root";

private static final String dbPwd="chenkan";

static{

try {

Class.forName("com.mysql.jdbc.Driver");

} catch (ClassNotFoundException e) {

// TODO Auto-generated catch block

e.printStackTrace();

}

}

public static Connection getConnection() throws java.sql.SQLException{

return java.sql.DriverManager.getConnection(jdbcUrl, dbUser, dbPwd);

}

### ②数据库驱动函数-查询

public static void main(String[] args) {

User user1=new User(?,?);

if(login(user1)) {

System.out.println("登陆成功");

}else {

System.out.println("登陆失败");

}

}

private static boolean login(User userZhangsan) {

Connection conn=null;

Statement st=null;

ResultSet rs=null;

try {

//1、装载驱动

Class.forName("com.mysql.jdbc.Driver");

} catch (ClassNotFoundException e) {

e.printStackTrace();

}

try {

//2、链接数据库，使用com.mysql.jdbc.Connection包会出错

List<User> list=new ArrayList<User>();

conn=DriverManager.getConnection("jdbc:mysql://localhost:3306/sw\_database?user=root&password=root");

//3、创建连接语句

st=conn.createStatement();

//4、执行ＳＱＬ语句获得结果集

rs=st.executeQuery("select \* from sw\_user where username='"+userZhangsan.getUsername()+"' and password='"+userZhangsan.getPassword()+"'");

//5、循环获得数据库字段生成对象

//这种方法登录要把数据库数据都拿过来和login中数据比较，超级浪费资源

if(rs.next()) {

return true;

}

} catch (SQLException e) {

// TODO Auto-generated catch block

e.printStackTrace();

}finally {

//关闭结果集

try {

if(rs!=null) {

rs.close();

}

} catch (SQLException e) {

// TODO Auto-generated catch block

e.printStackTrace();

}

//关闭连接语句

try {

if(st!=null) {

st.close();

}

} catch (SQLException e) {

// TODO Auto-generated catch block

e.printStackTrace();

}

//关闭数据库连接

try {

if(conn!=null) {

conn.close();

}

} catch (SQLException e) {

// TODO Auto-generated catch block

e.printStackTrace();

}

}

return false;

### ③数据库驱动函数-登录和注册

public class login {

/\*\*

\* 用以实现用户的注册和登录

\*/

private static String username;//用户登录注册的姓名

private static String password;//用户密码

private static String url="jdbc:mysql://localhost:3306/dcapp";//连接数据库的url，。

private static String user="root";//mysql登录名

private static String pass="chenkan";//mysql登录密码（写自己之前设置的）

private static Connection con;//

static Scanner input =new Scanner(System.in);

public static void main(String[] args) throws Exception {

//加载数据库连接驱动并连接

Class.forName("com.mysql.jdbc.Driver");

con=DriverManager.getConnection(url,user,pass);

System.out.println("\*\*\*\*\*\*\*\*用户界面\*\*\*\*\*\*\*\*");

System.out.println("请选择：\n 1:用户登录\n 2：用户注册");

System.out.println("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*");

int i=input.nextInt();

switch(i){

case 1:

denglu();

break;

case 2:

zhuce();

break;

default :

System.out.println("输入有误！");

System.exit(0);

}

}

//用户注册

public static void zhuce() throws SQLException{

System.out.println("请输入你的姓名：");

username=input.next();

System.out.println("请输入你的登录密码：");

String p1=input.next();

System.out.println("请再次输入你的确认密码：");

String p2=input.next();

if(p1.equals(p2)){

//两次输入的密码相同才可以注册

password=p1;

String sql="insert into client (username,password) values(?,?)";

PreparedStatement ptmt=con.prepareStatement(sql);

ptmt.setString(1, username);

ptmt.setString(2, password);

ptmt.execute();

System.out.println("注册成功！\n请登录：");

denglu();

}else{

System.out.println("你输入的密码与确认密码不相符，请重新注册：");

zhuce();

}

}

//用户登录

public static void denglu() throws SQLException{

System.out.println("请输入你的姓名：");

username=input.next();

System.out.println("请输入你的密码：");

password=input.next();

String sql="select username,password from client where username=? and password=?";

PreparedStatement ptmt=con.prepareStatement(sql);

ptmt.setString(1, username);

ptmt.setString(2, password);

ResultSet rs=ptmt.executeQuery();

//从登录用户给出的账号密码来检测查询在数据库表中是否存在相同的账号密码

if(rs.next()){

System.out.println("登录成功！");

}else{

System.out.println("姓名或密码错误！\n请重新登录：");

denglu();

}

}

## 1.3后端数据传输函数

### ①登录数据函数

public class Denglu extends HttpServlet {

private static final long serialVersionUID = 1L;

/\*\*

\* @see HttpServlet#HttpServlet()

\*/

public Denglu() {

super();

// TODO Auto-generated constructor stub

}

protected void service(HttpServletRequest request, HttpServletResponse response)

throws ServletException, IOException {

System.out.println("2");

String method = request.getMethod();

System.out.println(method);

if ("GET".equals(method)) {

System.out.println("请求方法：GET");

doGet(request, response);

} else if ("POST".equals(method)) {

System.out.println("请求方法：POST");

doPost(request, response);

} else {

System.out.println("请求方法分辨失败！");

}

}

/\*\*

\* @see HttpServlet#doGet(HttpServletRequest request, HttpServletResponse response)

\*/

protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {

// TODO Auto-generated method stub

System.out.println("3");

doPost(request, response);

System.out.println("4");

}

/\*\*

\* @see HttpServlet#doPost(HttpServletRequest request, HttpServletResponse response)

\*/

protected void doPost(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {

// TODO Auto-generated method stub

request.setCharacterEncoding("utf-8");

String name="";

String password = "";

name = request.getParameter("name");

password = request.getParameter("password");

String result = "";

//数据库函数获取

userContr usercontr=new userContr();

String zpassword=usercontr.chaUsermima(name);

//数据库函数获取

if ((zpassword.equals(password))){

result="success";

}else{

result = "fail";

}

JSONObject jsonObject = new JSONObject();

// URLEncoder.encode(result1,"utf-8");

jsonObject.put("result", result);

response.setContentType("text/javascript;charset=utf-8");

response.getWriter().print(jsonObject);

}

public void destroy() {

System.out.println("RegisterServlet destory.");

super.destroy();

}

### ②注册数据函数

public class Zhuce extends HttpServlet {

private static final long serialVersionUID = 1L;

/\*\*

\* @see HttpServlet#HttpServlet()

\*/

public Zhuce() {

super();

// TODO Auto-generated constructor stub

}

protected void service(HttpServletRequest request, HttpServletResponse response)

throws ServletException, IOException {

System.out.println("2");

String method = request.getMethod();

System.out.println(method);

if ("GET".equals(method)) {

System.out.println("请求方法：GET");

doGet(request, response);

} else if ("POST".equals(method)) {

System.out.println("请求方法：POST");

doPost(request, response);

} else {

System.out.println("请求方法分辨失败！");

}

}

/\*\*

\* @see HttpServlet#doGet(HttpServletRequest request, HttpServletResponse response)

\*/

protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {

// TODO Auto-generated method stub

System.out.println("3");

doPost(request, response);

System.out.println("4");

}

/\*\*

\* @see HttpServlet#doPost(HttpServletRequest request, HttpServletResponse response)

\*/

protected void doPost(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {

// TODO Auto-generated method stub

request.setCharacterEncoding("utf-8");

String name="";

String password1 = "";

String password2 = "";

String youxiang="";

String Sciku="";

name = request.getParameter("name");

password1 = request.getParameter("password1");

password2 = request.getParameter("password2");

youxiang = request.getParameter("youxiang");

Sciku = request.getParameter("ciku");

int ciku = Integer.parseInt(Sciku);

String result = "";

if(name==""||name==null)

result="namenull";

else if(password1==""||password1==null||password2==""||password2==null)

result="passwordnull";

else if(youxiang==""||youxiang==null)

result="youxiangnull";

else if(name.length()<5||name.length()>13)

result="nameerror";

else if(password1.length()<5||password1.length()>13)

result="passworderror";

else if(!password1.equals(password2))

result="passwordsame";

//数据库函数获取 if(result==""){

userContr usercontr=new userContr();

result=usercontr.addUser(name, password1, password2, youxiang, ciku);

}

//result=success

//数据库函数获取

JSONObject jsonObject = new JSONObject();

// URLEncoder.encode(result1,"utf-8");

jsonObject.put("result", result);//这里注意也要改

response.setContentType("text/javascript;charset=utf-8");

response.getWriter().print(jsonObject);

}

public void destroy() {

System.out.println("RegisterServlet destory.");

super.destroy();

}

### ⑭修改用户信息模块

ppublic class updUserInf extends HttpServlet {

private static final long serialVersionUID = 1L;

/\*\*

\* @see HttpServlet#HttpServlet()

\*/

public updUserInf() {

super();

// TODO Auto-generated constructor stub

}

protected void service(HttpServletRequest request, HttpServletResponse response)

throws ServletException, IOException {

System.out.println("2");

String method = request.getMethod();

System.out.println(method);

if ("GET".equals(method)) {

System.out.println("请求方法：GET");

doGet(request, response);

} else if ("POST".equals(method)) {

System.out.println("请求方法：POST");

doPost(request, response);

} else {

System.out.println("请求方法分辨失败！");

}

}

/\*\*

\* @see HttpServlet#doGet(HttpServletRequest request, HttpServletResponse response)

\*/

protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {

// TODO Auto-generated method stub

System.out.println("3");

doPost(request, response);

System.out.println("4");

}

/\*\*

\* @see HttpServlet#doPost(HttpServletRequest request, HttpServletResponse response)

\*/

protected void doPost(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {

// TODO Auto-generated method stub

request.setCharacterEncoding("utf-8");

String name="";

String youxiang="";

String Sciku="";

String geqian="";

name = request.getParameter("name");

youxiang = request.getParameter("youxiang");

Sciku = request.getParameter("ciku");

geqian = request.getParameter("geqian");

int ciku = Integer.parseInt(Sciku);

String result = "";

if(youxiang==""||youxiang==null)

result="youxiangnull";

else if(geqian.length()>20)

result="geqianerror";

//数据库函数获取!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!

if(result==""){

userContr usercontr=new userContr();

result=usercontr.updUser(name, youxiang, ciku, geqian);

}

//result=success

//result=success

//数据库函数获取!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!

JSONObject jsonObject = new JSONObject();

// URLEncoder.encode(result1,"utf-8");

jsonObject.put("result", result);//这里注意也要改

response.setContentType("text/javascript;charset=utf-8");

response.getWriter().print(jsonObject);

}

public void destroy() {

System.out.println("RegisterServlet destory.");

super.destroy();

}

### ⑮修改用户密码模块

public class updUserPassword extends HttpServlet {

private static final long serialVersionUID = 1L;

/\*\*

\* @see HttpServlet#HttpServlet()

\*/

public updUserPassword() {

super();

// TODO Auto-generated constructor stub

}

protected void service(HttpServletRequest request, HttpServletResponse response)

throws ServletException, IOException {

System.out.println("2");

String method = request.getMethod();

System.out.println(method);

if ("GET".equals(method)) {

System.out.println("请求方法：GET");

doGet(request, response);

} else if ("POST".equals(method)) {

System.out.println("请求方法：POST");

doPost(request, response);

} else {

System.out.println("请求方法分辨失败！");

}

}

/\*\*

\* @see HttpServlet#doGet(HttpServletRequest request, HttpServletResponse response)

\*/

protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {

// TODO Auto-generated method stub

System.out.println("3");

doPost(request, response);

System.out.println("4");

}

/\*\*

\* @see HttpServlet#doPost(HttpServletRequest request, HttpServletResponse response)

\*/

protected void doPost(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {

// TODO Auto-generated method stub

request.setCharacterEncoding("utf-8");

String name="";

String password = "";

name = request.getParameter("name");

password = request.getParameter("password");

String result = "";

//数据库函数获取!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!

userContr usercontr=new userContr();

result=usercontr.updUserPaw(name, password);

//数据库函数获取!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!

if (("success".equals(result))){

result="success";

}else{

result = "fail";

}

JSONObject jsonObject = new JSONObject();

// URLEncoder.encode(result1,"utf-8");

jsonObject.put("result", result);//这里注意也要改

response.setContentType("text/javascript;charset=utf-8");

response.getWriter().print(jsonObject);

}

public void destroy() {

System.out.println("RegisterServlet destory.");

super.destroy();

}

## 1.3定义函数

### ①添加笔记定义

**public class** Note **implements** Serializable,Comparable<Note> {  
 **private** String **id**;*//这里数据类型应该改为int* **private** String **content**;  
 **private** String **firstTime**;*//note生成时间* **private** String **lastTime**;*//note最后编辑时间* **private** String **category**;  
 **private boolean isFlag**=**false**;*//当isFlag=true,note的背景drawable变色* **public** String getId() {  
 **return id**;  
 }  
 **public void** setId(String id) {  
 **this**.**id** = id;  
 }  
 **public** String getContent(){  
 **return content**;  
 }  
 **public void** setContent(String s){  
 **this**.**content**=s;  
 }  
 **public void** setFirstTime(String s){  
 **this**.**firstTime**=s;  
 }  
 **public** String getFirstTime(){  
 **return firstTime**;  
 }  
 **public void** setLastTime(String s){  
 **this**.**lastTime**=s;  
 }  
 **public** String getLastTime(){  
 **return lastTime**;  
 }  
 **public void** setFlag(**boolean** isFlag){  
 **this**.**isFlag**=isFlag;  
 }  
 **public boolean** isFlag(){  
 **return isFlag**;  
 }  
 **public void** setCategory(String s){  
 **this**.**category**=s;  
 }  
 **public** String getCategory(){  
 **return category**;  
 }  
  
 @Override  
 **public int** compareTo(@NonNull Note o) {  
 **if** (Long.*parseLong*(**this**.getLastTime())==Long.*parseLong*(o.getLastTime())){*//string型转long型* **return** 0;  
 }**else if** (Long.*parseLong*(**this**.getLastTime())>Long.*parseLong*(o.getLastTime())){  
 **return** -1;  
 }**else** {  
 **return** 1;  
 }  
 }

### ②添加反馈定义

public class tianjiafankui {

private String fankuikind;

private String fankuixinxi;

public String getfankuikind() {

return fankuikind;

}

public void setfankuikind(String fankuikind) {

this.fankuikind = fankuikind;

}

public String getfankuixinxi() {

return fankuixinxi;

}

public void setfankuixinxi(String fankuixinxi) {

this.fankuixinxi = fankuixinxi;

}

}

### ③添加用户定义

public class tianjiayonghu {

private String UserId;

private String UserName;

private String UserPassword;

private String UserEmail;

private String UserSelect;

public String getUserId() {

return UserId;

}

public void setUserId(String UserId) {

this.UserId = UserId;

}

public String getUserName() {

return UserName;

}

public void setUserName(String UserName) {

this.UserName = UserName;

}

public String getUserPassword() {

return UserPassword;

}

public void setUserPassword(String UserPassword) {

this.UserPassword = UserPassword;

}

public String getUserEmail() {

return UserEmail;

}

public void setUserEmail(String UserEmail) {

this.UserEmail = UserEmail;

}

public String getUserSelect() {

return UserSelect;

}

public void setannounce(String UserSelect) {

this.UserSelect = UserSelect;

}

}

1. 前端代码

2.1连接服务器配置代码

package com.example.danciji1\_0;

import java.io.BufferedReader;

import java.io.BufferedWriter;

import java.io.InputStreamReader;

import java.io.OutputStream;

import java.io.OutputStreamWriter;

import java.io.UnsupportedEncodingException;

import java.net.HttpURLConnection;

import java.net.URL;

import java.net.URLEncoder;

import java.util.Map;

public class HttpUtilsHttpURLConnection {

public static String BASE\_URL= "http://47.106.110.34:8080/DanCiJi";

/\*

\* urlStr:网址

\* parms：提交数据

\* return:网页源码

\* \*/

public static String getContextByHttp(String urlStr,Map<String,String> parms){

StringBuilder sb = new StringBuilder();

try{

URL url = new URL(urlStr);

HttpURLConnection connection = (HttpURLConnection)url.openConnection();

connection.setRequestMethod("POST");

connection.setReadTimeout(5000);

connection.setConnectTimeout(5000);

connection.setDoInput(true);

connection.setDoOutput(true);

connection.setInstanceFollowRedirects(true);

OutputStream outputStream = connection.getOutputStream();

BufferedWriter writer = new BufferedWriter(new OutputStreamWriter(outputStream,"UTF-8"));

writer.write(getStringFromOutput(parms));

writer.flush();

writer.close();

outputStream.close();

if(connection.getResponseCode() == HttpURLConnection.HTTP\_OK){

BufferedReader reader = new BufferedReader(new InputStreamReader(connection.getInputStream()));

String temp;

while((temp = reader.readLine()) != null){

sb.append(temp);

}

reader.close();

}else{

return "connection error:" + connection.getResponseCode();

}

connection.disconnect();

}catch (Exception e){

return e.toString();

}

return sb.toString();

}

/\*\*

\* 将map转换成key1=value1&key2=value2的形式

\* @param map

\* @return

\* @throws UnsupportedEncodingException

\*/

private static String getStringFromOutput(Map<String,String> map) throws UnsupportedEncodingException {

StringBuilder sb = new StringBuilder();

boolean isFirst = true;

for(Map.Entry<String,String> entry:map.entrySet()){

if(isFirst)

isFirst = false;

else

sb.append("&");

sb.append(URLEncoder.encode(entry.getKey(),"UTF-8"));

sb.append("=");

sb.append(URLEncoder.encode(entry.getValue(),"UTF-8"));

}

return sb.toString();

}

}

2.2功能函数

①添加笔记

**public class** NoteAddActivity **extends** BaseActivity{  
 **private** EditText **et\_addNote**;  
 **private** Toolbar **toolbar**;  
 **private** TextView **mTitle**;  
 **private** NoteManager **noteManager**;  
 @Override  
 **protected int** setLayoutResId() {  
 **return** R.layout.***activity\_note\_add***;  
 }  
  
 @Override  
 **protected void** initData() {  
 **noteManager**=**new** NoteManager(**mContext**);  
 }  
  
 @Override  
 **protected void** initView() {  
 **toolbar**=(Toolbar)findViewById(R.id.***toolbar\_include***);  
 **et\_addNote**=(EditText)findViewById(R.id.***et\_note\_add***);  
 **mTitle**=(TextView)findViewById(R.id.***include\_tv\_center\_title***);  
 setSupportActionBar(**toolbar**);  
 getSupportActionBar().setDisplayHomeAsUpEnabled(**true**);  
 getSupportActionBar().setDisplayShowHomeEnabled(**true**);  
 setTitle(**""**);  
 **mTitle**.setText(**"编辑"**);  
 }  
  
 @Override  
 **protected void** initListener() {  
 }  
 **private void** goBack(){  
 **if** (!TextUtils.*isEmpty*(**et\_addNote**.getText())){  
 String s=**et\_addNote**.getText().toString();  
 Note note=**new** Note();  
 **int** i=SPUtil.*getInt*(**mContext**,**"noteId"**,-1);  
 note.setId(String.*valueOf*(i+1));  
 note.setContent(s);  
 Date date=**new** Date();  
 SimpleDateFormat sdf=**new** SimpleDateFormat(**"MM月dd日"**);  
 String firstTime=sdf.format(date);  
 note.setFirstTime(firstTime);  
 note.setLastTime(Long.*toString*(date.getTime()));  
 note.setCategory(SPUtil.*getString*(**mContext**,**"currentCate"**,**"所有"**));  
 **noteManager**.insert(note);  
 SPUtil.*putInt*(**mContext**,**"noteId"**,i+1);  
 goToNoteList();  
 }**else** {  
 goToNoteList();  
 }  
 }  
 **private void** goToNoteList(){  
 finish();  
 }  
  
 @Override  
 **public void** onBackPressed() {  
 goBack();  
 }  
  
 @Override  
 **public boolean** onCreateOptionsMenu(Menu menu) {  
 **return super**.onCreateOptionsMenu(menu);  
 }  
  
 @Override  
 **public boolean** onOptionsItemSelected(MenuItem item) {  
 **switch** (item.getItemId()){  
 **case** android.R.id.***home***:  
 goBack();  
 **return true**;  
 }  
 **return super**.onOptionsItemSelected(item);  
 }  
}

**public class** CategoryAddActivity **extends** BaseActivity {  
 **private** EditText **cateName**;  
 **private** Toolbar **toolbar**;  
 **private** TextView **mTitle**;  
 **private** ImageView **deleteImg**;  
  
 **private int pos**;  
 **private** String **cate**;  
 **private** NoteManager **noteManager**;  
  
 @Override  
 **protected int** setLayoutResId() {  
 **return** R.layout.***activity\_category\_add***;  
 }  
  
 @Override  
 **protected void** initData() {  
 **noteManager**=**new** NoteManager(**mContext**);  
 Intent intent=getIntent();  
 **pos**=intent.getIntExtra(**"cateSize"**,-1);  
 }  
  
 @Override  
 **protected void** initView() {  
 **cateName**=(EditText)findViewById(R.id.***et\_cate\_name***);  
 **deleteImg**=(ImageView)findViewById(R.id.***delete***);  
 **toolbar**=(Toolbar)findViewById(R.id.***toolbar\_include***);  
 **mTitle**=(TextView)findViewById(R.id.***include\_tv\_center\_title***);  
 setSupportActionBar(**toolbar**);  
 getSupportActionBar().setDisplayHomeAsUpEnabled(**true**);  
 getSupportActionBar().setDisplayShowHomeEnabled(**true**);  
 setTitle(**""**);  
 **mTitle**.setText(**"新建文件夹"**);  
 TextWatcher watcher=**new** TextWatcher() {  
 @Override  
 **public void** beforeTextChanged(CharSequence s, **int** start, **int** count, **int** after) {  
 }  
  
 @Override  
 **public void** onTextChanged(CharSequence s, **int** start, **int** before, **int** count) {  
 }  
  
 @Override  
 **public void** afterTextChanged(Editable s) {  
 **if** (!TextUtils.*isEmpty*(**cateName**.getText().toString())){  
 **deleteImg**.setVisibility(View.***VISIBLE***);  
 }**else** {  
 **deleteImg**.setVisibility(View.***INVISIBLE***);  
 **cateName**.setHint(**"笔记文件夹名称"**);  
 }  
 }  
 };  
 **cateName**.addTextChangedListener(watcher);  
 }  
  
 @Override  
 **protected void** initListener() {  
 **deleteImg**.setOnClickListener(**new** View.OnClickListener() {  
 @Override  
 **public void** onClick(View v) {  
 **if** (!TextUtils.*isEmpty*(**cateName**.getText().toString())){  
 **cateName**.setText(**""**);  
 **cateName**.setHint(**"笔记文件夹名称"**);  
 }  
 }  
 });  
 }  
 **private void** goBack(){  
 **cate**=**cateName**.getText().toString();  
 **if** (**cate**.length()>0){  
 Category category=**new** Category();  
 **int** id=SPUtil.*getInt*(**mContext**,**"categoryNum"**,-1);  
 category.setId(String.*valueOf*(id+1));  
 category.setCategory(**cate**);  
 category.setPos(String.*valueOf*(**pos**+1));  
 **noteManager**.insertCategory(category);  
 SPUtil.*putInt*(**mContext**,**"categoryNum"**,id+1);  
 goToCateManager();  
 }**else**{  
 goToCateManager();  
 }  
 }  
 **private void** goToCateManager(){  
 finish();  
 }  
 @Override  
 **public void** onBackPressed() {  
 goBack();  
 }  
  
 @Override  
 **public boolean** onCreateOptionsMenu(Menu menu) {  
 **return super**.onCreateOptionsMenu(menu);  
 }  
  
 @Override  
 **public boolean** onOptionsItemSelected(MenuItem item) {  
 **switch** (item.getItemId()){  
 **case** android.R.id.***home***:  
 goBack();  
 **return true**;  
 }  
 **return super**.onOptionsItemSelected(item);  
 }  
}

②帮助函数

public class Bangzhu extends AppCompatActivity {

Button fanhui;

@Override

protected void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.activity\_bangzhu);

fanhui=(Button)findViewById(R.id.BZ\_fanhui);

fanhui.setOnClickListener(new View.OnClickListener() {

@Override

public void onClick(View v) {

Intent intent =new Intent(Bangzhu.this,Zhuye.class);

startActivity(intent);

}

});

}

}

③删除单词函数

public class DelDC extends AppCompatActivity {

Button fanhui;

Button shanchu;

EditText danci;

RadioGroup zu;

RadioButton siji;

RadioButton liuji;

@Override

protected void onCreate(@Nullable Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.activity\_deldanci);

fanhui=(Button)findViewById(R.id.DDC\_fanhui);

danci=(EditText)findViewById(R.id.DDC\_danci);

shanchu=(Button)findViewById(R.id.DDC\_shanchu);

zu=(RadioGroup)findViewById(R.id.DDC\_zu);

siji=(RadioButton)findViewById(R.id.DDC\_siji);

liuji=(RadioButton)findViewById(R.id.DDC\_liuji);

fanhui.setOnClickListener(new View.OnClickListener() {

@Override

public void onClick(View v) {

Intent intent = new Intent(DelDC.this, Guanliyuan.class);

startActivity(intent);

}

});

shanchu.setOnClickListener(new View.OnClickListener() {

@Override

public void onClick(View v) {

new Thread(new Runnable() {

@Override

public void run() {

String url=HttpUtilsHttpURLConnection.BASE\_URL+"/delDanCi";/////这个后面的/conn也要对应的改

Map<String, String> params = new HashMap<String, String>();

//你的代码放这下面

String DC=danci.getText().toString();

String DCciku="4";

if(zu.getCheckedRadioButtonId()==R.id.DDC\_siji){

DCciku="4";

}

else if(zu.getCheckedRadioButtonId()==R.id.DDC\_liuji){

DCciku="6";

}

params.put("DC",DC);

params.put("DCciku",DCciku);

//你的代码放这上面

String result = HttpUtilsHttpURLConnection.getContextByHttp(url,params);

Message msg = new Message();

msg.what=0x12;

Bundle data=new Bundle();

data.putString("Bigresult",result);

msg.setData(data);

hander.sendMessage(msg);

}

Handler hander = new Handler(){

@Override

public void handleMessage(Message msg) {

if (msg.what==0x12){

Bundle data = msg.getData();

String key = data.getString("Bigresult");

//得到json返回的json

// Toast.makeText(Denglu.this,key,Toast.LENGTH\_LONG).show();

try {

JSONObject json= new JSONObject(key);

String result = (String) json.get("result");

//你的代码放这下面

if ("success".equals(result)){

Toast.makeText(DelDC.this,"删除成功！",Toast.LENGTH\_SHORT).show();

Intent intent =new Intent(DelDC.this,Guanliyuan.class);

startActivity(intent);

}else if("error".equals(result)){

Toast.makeText(DelDC.this,"单词不存在！",Toast.LENGTH\_SHORT).show();

}

// Toast.makeText(MainActivity.this,(String) json.get("result1"),Toast.LENGTH\_LONG).show();

//你的代码放这上面

} catch (JSONException e) {

e.printStackTrace();

}

}

}

};

}).start();

}

});

}

}

③登录函数

public class Denglu extends AppCompatActivity {

EditText zhanghao;

EditText mima;

Button denglu;

Button zhuce;

Button miandenglu;

@Override

protected void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.activity\_denglu);

zhanghao=(EditText)findViewById(R.id.DL\_name);

mima=(EditText)findViewById(R.id.DL\_mima);

denglu=(Button)findViewById(R.id.DL\_denglu);

zhuce=(Button)findViewById(R.id.DL\_zhuce);

miandenglu=(Button)findViewById(R.id.DL\_miandenglu);

zhuce.setOnClickListener(new View.OnClickListener() {

@Override

public void onClick(View v) {

Intent intent =new Intent(Denglu.this,Zhuce.class);

startActivity(intent);

}

});

denglu.setOnClickListener(new View.OnClickListener() {

@Override

public void onClick(View v) {

Log.w("Hyx", "1");

new Thread(new Runnable() {

@Override

public void run() {

Log.w("Hyx", "2");

String url=HttpUtilsHttpURLConnection.BASE\_URL+"/Denglu";/////这个后面的/conn也要对应的改

Map<String, String> params = new HashMap<String, String>();

//你的代码放这下面

Log.w("Hyx", "3");

String name=zhanghao.getText().toString();

String password=mima.getText().toString();

params.put("name",name);

params.put("password",password);

//你的代码放这上面

String result = HttpUtilsHttpURLConnection.getContextByHttp(url,params);

Log.w("Hyx", result);

Message msg = new Message();

msg.what=0x12;

Bundle data=new Bundle();

data.putString("Bigresult",result);

data.putString("name",name);

msg.setData(data);

hander.sendMessage(msg);

}

Handler hander = new Handler(){

@Override

public void handleMessage(Message msg) {

if (msg.what==0x12){

Bundle data = msg.getData();

String key = data.getString("Bigresult");

String name=data.getString("name");//得到json返回的json

// Toast.makeText(Denglu.this,key,Toast.LENGTH\_LONG).show();

try {

JSONObject json= new JSONObject(key);

String result = (String) json.get("result");

//你的代码放这下面

Log.w("Hyx", "4");

if(zhanghao.getText().toString().equals("")||zhanghao.getText().toString().equals(null)||mima.getText().toString().equals("")||mima.getText().toString().equals(null)){

Toast.makeText(Denglu.this,"账号密码不能为空！",Toast.LENGTH\_SHORT).show();

}

else if ("success".equals(result)){

Toast.makeText(Denglu.this,"登录成功",Toast.LENGTH\_SHORT).show();

USERNAME=name;

USERPASSWORD=mima.getText().toString();

Intent intent =new Intent(Denglu.this,Zhuye.class);

startActivity(intent);

}else if("fail".equals(result)){

Toast.makeText(Denglu.this,"账号或密码错误",Toast.LENGTH\_SHORT).show();

}

// Toast.makeText(MainActivity.this,(String) json.get("result1"),Toast.LENGTH\_LONG).show();

//你的代码放这上面

} catch (JSONException e) {

e.printStackTrace();

}

}

}

};

}).start();

}

});

miandenglu.setOnClickListener(new View.OnClickListener() {

@Override

public void onClick(View v) {

Intent intent =new Intent(Denglu.this,Miandengluxuanci.class);

startActivity(intent);

}

});

}

}

⑤改密码函数

public class Gaimima extends AppCompatActivity {

Button tijiao;

Button fanhui;

EditText mima1;

EditText mima2;

EditText mima3;

String mima11;

String mima12;

String mima13;

@Override

protected void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.activity\_gaimima);

tijiao=(Button)findViewById(R.id.GMM\_tijiao);

fanhui=(Button)findViewById(R.id.GMM\_fanhui);

mima1=(EditText)findViewById(R.id.GMM\_mima1);

mima2=(EditText)findViewById(R.id.GMM\_mima2);

mima3=(EditText)findViewById(R.id.GMM\_mima3);

//y页面加载时从数据库导入信息

fanhui.setOnClickListener(new View.OnClickListener() {

@Override

public void onClick(View v) {

Intent intent =new Intent(Gaimima.this,Gerenzhongxin.class);

startActivity(intent);

}

});

tijiao.setOnClickListener(new View.OnClickListener() {

@Override

public void onClick(View v) {

mima11=mima1.getText().toString();

mima12=mima2.getText().toString();

mima13=mima3.getText().toString();

Log.w("Hyx", USERPASSWORD);

if(!mima11.equals(USERPASSWORD)){

Toast.makeText(Gaimima.this,"原始密码不正确！",Toast.LENGTH\_LONG).show();

}

else if(!mima12.equals(mima13)){

Toast.makeText(Gaimima.this,"两次密码应相同！",Toast.LENGTH\_LONG).show();

}

else if(mima12.length()>13||mima12.length()<5){

Toast.makeText(Gaimima.this,"新密码长度应在5-13位！",Toast.LENGTH\_LONG).show();

}

else {

new Thread(new Runnable() {

@Override

public void run() {

String url = HttpUtilsHttpURLConnection.BASE\_URL + "/updUserPassword";/////这个后面的/conn也要对应的改

Map<String, String> params = new HashMap<String, String>();

//你的代码放这下面

String name = USERNAME;

String password=mima12;

params.put("name", name);

params.put("password", password);

//你的代码放这上面

String result = HttpUtilsHttpURLConnection.getContextByHttp(url, params);

Message msg = new Message();

msg.what = 0x12;

Bundle data = new Bundle();

data.putString("Bigresult", result);

data.putString("name", name);

msg.setData(data);

hander.sendMessage(msg);

}

Handler hander = new Handler() {

@Override

public void handleMessage(Message msg) {

if (msg.what == 0x12) {

Bundle data = msg.getData();

String key = data.getString("Bigresult");

String name = data.getString("name");//得到json返回的json

// Toast.makeText(Denglu.this,key,Toast.LENGTH\_LONG).show();

try {

JSONObject json = new JSONObject(key);

String result = (String) json.get("result");

//你的代码放这下面

if ("success".equals(result)) {

Toast.makeText(Gaimima.this, "修改成功！", Toast.LENGTH\_SHORT).show();

Intent intent = new Intent(Gaimima.this, Gerenzhongxin.class);//换成改密码的界面

startActivity(intent);

}

// Toast.makeText(MainActivity.this,(String) json.get("result1"),Toast.LENGTH\_LONG).show();

//你的代码放这上面

} catch (JSONException e) {

e.printStackTrace();

}

}

}

};

}).start();

}

}

});

}

}

⑥个人中心函数

public class Gerenzhongxin extends AppCompatActivity {

Button gaimima;

Button tijiao;

Button fanhui;

Button zhuxiao;

TextView zhanghao;

EditText geqian;

EditText youxiang;

RadioGroup zu;

RadioButton siji;

RadioButton liuji;

@Override

protected void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.activity\_gerenzhongxin);

zhanghao=(TextView)findViewById(R.id.GRZX\_zhanghao);

geqian=(EditText)findViewById(R.id.GRZX\_gexingqianming);

youxiang=(EditText)findViewById(R.id.GRZX\_youxiang);

zu=(RadioGroup)findViewById(R.id.GRZX\_zu);

siji=(RadioButton)findViewById(R.id.GRZX\_siji);

liuji=(RadioButton)findViewById(R.id.GRZX\_liuji);

fanhui=(Button)findViewById(R.id.GRZX\_fanhui);

zhuxiao=(Button)findViewById(R.id.GRZX\_zhuxiao);

tijiao=(Button)findViewById(R.id.GRZX\_tijiao);

gaimima=(Button)findViewById(R.id.GRZX\_gaimima);

//y页面加载时从数据库导入信息

Log.w("Hyx", "1");

new Thread(new Runnable() {

@Override

public void run() {

String url=HttpUtilsHttpURLConnection.BASE\_URL+"/chaUsersInf";/////这个后面的/conn也要对应的改

Map<String, String> params = new HashMap<String, String>();

//你的代码放这下面

params.put("name",USERNAME);

//你的代码放这上面

Log.w("Hyx", "2");

String result = HttpUtilsHttpURLConnection.getContextByHttp(url,params);

Log.w("Hyx", result);

Message msg = new Message();

msg.what=0x12;

Bundle data=new Bundle();

data.putString("Bigresult",result);

msg.setData(data);

hander.sendMessage(msg);

}

Handler hander = new Handler(){

@Override

public void handleMessage(Message msg) {

if (msg.what==0x12){

Bundle data = msg.getData();

String key = data.getString("Bigresult");//得到json返回的json

// Toast.makeText(Denglu.this,key,Toast.LENGTH\_LONG).show();

try {

Log.w("Hyx", "4");

JSONObject json= new JSONObject(key);

Log.w("Hyx", "4.01");

String youxiang1 = (String) json.get("youxiang");

Log.w("Hyx", "4.02");

String geqian1 = (String) json.get("geqian");

Log.w("Hyx", "4.03");

String mima1 = (String) json.get("mima");

Log.w("Hyx", "4.04");

String ciku1 = (String) json.get("ciku");

//你的代码放这下面

Log.w("Hyx", "4.1");

zhanghao.setText(USERNAME);

Log.w("Hyx", "4.2");

youxiang.setText(youxiang1);

geqian.setText(geqian1);

Log.w("Hyx", "4.3");

if(ciku1.equals("4")){

siji.setChecked(true);

}

else if(ciku1.equals("6")){

liuji.setChecked(true);

}

Log.w("Hyx", "4.4");

USERPASSWORD=mima1;

Log.w("Hyx", "5");

// Toast.makeText(MainActivity.this,(String) json.get("result1"),Toast.LENGTH\_LONG).show();

//你的代码放这上面

} catch (JSONException e) {

e.printStackTrace();

}

}

}

};

}).start();

fanhui.setOnClickListener(new View.OnClickListener() {

@Override

public void onClick(View v) {

Intent intent =new Intent(Gerenzhongxin.this,Zhuye.class);

startActivity(intent);

}

});

zhuxiao.setOnClickListener(new View.OnClickListener() {

@Override

public void onClick(View v) {

USERNAME="";

USERPASSWORD="";

USERCIKU=4;

Intent intent =new Intent(Gerenzhongxin.this,Denglu.class);

startActivity(intent);

}

});

gaimima.setOnClickListener(new View.OnClickListener() {

@Override

public void onClick(View v) {

Intent intent =new Intent(Gerenzhongxin.this,Gaimima.class);//换成改密码的界面

startActivity(intent);

}

});

tijiao.setOnClickListener(new View.OnClickListener() {

@Override

public void onClick(View v) {

new Thread(new Runnable() {

@Override

public void run() {

String url=HttpUtilsHttpURLConnection.BASE\_URL+"/updUserInf";/////这个后面的/conn也要对应的改

Map<String, String> params = new HashMap<String, String>();

//你的代码放这下面

String name=zhanghao.getText().toString();

String gexingqm=geqian.getText().toString();

String youx=youxiang.getText().toString();

String ciku="4";

if(zu.getCheckedRadioButtonId()==R.id.ZC\_siji){

ciku="4";

}

else if(zu.getCheckedRadioButtonId()==R.id.ZC\_liuji){

ciku="6";

}

params.put("name",name);

params.put("geqian",gexingqm);

params.put("youxiang",youx);

params.put("ciku",ciku);

//你的代码放这上面

String result = HttpUtilsHttpURLConnection.getContextByHttp(url,params);

Message msg = new Message();

msg.what=0x12;

Bundle data=new Bundle();

data.putString("Bigresult",result);

data.putString("name",name);

msg.setData(data);

hander.sendMessage(msg);

}

Handler hander = new Handler(){

@Override

public void handleMessage(Message msg) {

if (msg.what==0x12){

Bundle data = msg.getData();

String key = data.getString("Bigresult");

String name=data.getString("name");//得到json返回的json

// Toast.makeText(Denglu.this,key,Toast.LENGTH\_LONG).show();

try {

JSONObject json= new JSONObject(key);

String result = (String) json.get("result");

//你的代码放这下面

if ("youxiangnull".equals(result)){

Toast.makeText(Gerenzhongxin.this,"邮箱不能为空！",Toast.LENGTH\_SHORT).show();

}else if("geqianerror".equals(result)){

Toast.makeText(Gerenzhongxin.this,"个签的长度不能超过20字！",Toast.LENGTH\_SHORT).show();

}else if("success".equals(result)){

Toast.makeText(Gerenzhongxin.this,"修改成功！",Toast.LENGTH\_SHORT).show();

Intent intent =new Intent(Gerenzhongxin.this,Zhuye.class);//换成改密码的界面

startActivity(intent);

}

// Toast.makeText(MainActivity.this,(String) json.get("result1"),Toast.LENGTH\_LONG).show();

//你的代码放这上面

} catch (JSONException e) {

e.printStackTrace();

}

}

}

};

}).start();

}

});

}

}

⑨反馈函数

public class InFKinf extends AppCompatActivity {

Button fanhui;

Button shang;

Button xia;

TextView fk;

TextView num;

@Override

protected void onCreate(@Nullable Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.activity\_infankuiinf);

fanhui = (Button) findViewById(R.id.IFK\_fanhui);

shang = (Button) findViewById(R.id.IFK\_shang);

xia = (Button) findViewById(R.id.IFK\_xia);

fk = (TextView) findViewById(R.id.IFK\_fkinf);

num=(TextView)findViewById(R.id.IFK\_num);

new Thread(new Runnable() {

@Override

public void run() {

String url=HttpUtilsHttpURLConnection.BASE\_URL+"/chaUserFanKui";/////这个后面的/conn也要对应的改

Map<String, String> params = new HashMap<String, String>();

//你的代码放这下面

String fkinfid=num.getText().toString();

params.put("fkinfid",fkinfid);

//你的代码放这上面

String fkinf = HttpUtilsHttpURLConnection.getContextByHttp(url,params);

Message msg = new Message();

msg.what=0x12;

Bundle data=new Bundle();

data.putString("Bigresult",fkinf);

msg.setData(data);

hander.sendMessage(msg);

}

Handler hander = new Handler(){

@Override

public void handleMessage(Message msg) {

if (msg.what==0x12){

Bundle data = msg.getData();

String key = data.getString("Bigresult");

//得到json返回的json

// Toast.makeText(Denglu.this,key,Toast.LENGTH\_LONG).show();

try {

JSONObject json= new JSONObject(key);

String fkinf = (String) json.get("fkinf");

//你的代码放这下面

fk.setText(fkinf);

// Toast.makeText(MainActivity.this,(String) json.get("result1"),Toast.LENGTH\_LONG).show();

//你的代码放这上面

} catch (JSONException e) {

e.printStackTrace();

}

}

}

};

}).start();

fanhui.setOnClickListener(new View.OnClickListener() {

@Override

public void onClick(View v) {

Intent intent = new Intent(InFKinf.this, Guanliyuan.class);

startActivity(intent);

}

});

xia.setOnClickListener(new View.OnClickListener() {

@Override

public void onClick(View v) {

new Thread(new Runnable() {

@Override

public void run() {

String url=HttpUtilsHttpURLConnection.BASE\_URL+"/chaUserFanKui";/////这个后面的/conn也要对应的改

Map<String, String> params = new HashMap<String, String>();

//你的代码放这下

String number=num.getText().toString();

int numberint=Integer.parseInt(number);

numberint=numberint+1;

String numberstr=Integer.toString(numberint);

params.put("fkinfid",numberstr);

//你的代码放这上面

String result = HttpUtilsHttpURLConnection.getContextByHttp(url,params);

Message msg = new Message();

msg.what=0x12;

Bundle data=new Bundle();

data.putString("Bigresult",result);

data.putString("numberstr",numberstr);

msg.setData(data);

hander.sendMessage(msg);

}

Handler hander = new Handler(){

@Override

public void handleMessage(Message msg) {

if (msg.what==0x12){

Bundle data = msg.getData();

String key = data.getString("Bigresult");

String numberstr = data.getString("numberstr");

//得到json返回的json

// Toast.makeText(Denglu.this,key,Toast.LENGTH\_LONG).show();

try {

JSONObject json= new JSONObject(key);

String fkinf = (String) json.get("fkinf");

//你的代码放这下面

if ("fail".equals(fkinf)){

Toast.makeText(InFKinf.this,"最后一页！",Toast.LENGTH\_SHORT).show();

}else {

fk.setText(fkinf);

num.setText(numberstr);

}

// Toast.makeText(MainActivity.this,(String) json.get("result1"),Toast.LENGTH\_LONG).show();

//你的代码放这上面

} catch (JSONException e) {

e.printStackTrace();

}

}

}

};

}).start();

}

});

shang.setOnClickListener(new View.OnClickListener() {

@Override

public void onClick(View v) {

new Thread(new Runnable() {

@Override

public void run() {

String url=HttpUtilsHttpURLConnection.BASE\_URL+"/chaUserFanKui";/////这个后面的/conn也要对应的改

Map<String, String> params = new HashMap<String, String>();

//你的代码放这下

String number=num.getText().toString();

int numberint=Integer.parseInt(number);

numberint=numberint-1;

String numberstr=Integer.toString(numberint);

params.put("fkinfid",numberstr);

//你的代码放这上面

String result = HttpUtilsHttpURLConnection.getContextByHttp(url,params);

Message msg = new Message();

msg.what=0x12;

Bundle data=new Bundle();

data.putString("Bigresult",result);

data.putString("numberstr",numberstr);

msg.setData(data);

hander.sendMessage(msg);

}

Handler hander = new Handler(){

@Override

public void handleMessage(Message msg) {

if (msg.what==0x12){

Bundle data = msg.getData();

String key = data.getString("Bigresult");

String numberstr = data.getString("numberstr");

//得到json返回的json

// Toast.makeText(Denglu.this,key,Toast.LENGTH\_LONG).show();

try {

JSONObject json= new JSONObject(key);

String fkinf = (String) json.get("fkinf");

//你的代码放这下面

if ("fail".equals(fkinf)){

Toast.makeText(InFKinf.this,"已经是第一页！",Toast.LENGTH\_SHORT).show();

}else {

fk.setText(fkinf);

num.setText(numberstr);

}

// Toast.makeText(MainActivity.this,(String) json.get("result1"),Toast.LENGTH\_LONG).show();

//你的代码放这上面

} catch (JSONException e) {

e.printStackTrace();

}

}

}

};

}).start();

}

});

}

}

⑩编辑笔记函数

**ublic class** CategoryEditActivity **extends** BaseActivity {  
 **private** EditText **et\_cate\_edit**;  
 **private** ImageView **deleteImg**;  
 **private** Toolbar **toolbar**;  
 **private** TextView **mTitle**;  
 **private** String **cate**;  
 *//private String catePos;* **private** NoteManager **noteManager**;  
 **private** String **cateId**;  
 **private** List<Note>**mNoteList**=**new** ArrayList<>();  
  
 @Override  
 **protected int** setLayoutResId() {  
 **return** R.layout.***activity\_category\_add***;  
 }  
  
 @Override  
 **protected void** initData() {  
 Intent intent=getIntent();  
 **cate**=intent.getStringExtra(**"category"**);  
 **cateId**=intent.getStringExtra(**"cateId"**);  
 *//catePos=intent.getStringExtra("pos");* **noteManager**=**new** NoteManager(**mContext**);  
 }  
  
 @Override  
 **protected void** initView() {  
 **et\_cate\_edit**=(EditText)findViewById(R.id.***et\_cate\_name***);  
 **et\_cate\_edit**.setText(**cate**);  
 **deleteImg**=(ImageView)findViewById(R.id.***delete***);  
 **toolbar**=(Toolbar)findViewById(R.id.***toolbar\_include***);  
 **mTitle**=(TextView)findViewById(R.id.***include\_tv\_center\_title***);  
 setSupportActionBar(**toolbar**);  
 getSupportActionBar().setDisplayHomeAsUpEnabled(**true**);  
 getSupportActionBar().setDisplayShowHomeEnabled(**true**);  
 setTitle(**""**);  
 **mTitle**.setText(**"编辑文件夹"**);  
 TextWatcher watcher=**new** TextWatcher() {  
 @Override  
 **public void** beforeTextChanged(CharSequence s, **int** start, **int** count, **int** after) {  
 }  
  
 @Override  
 **public void** onTextChanged(CharSequence s, **int** start, **int** before, **int** count) {  
 }  
  
 @Override  
 **public void** afterTextChanged(Editable s) {  
 **if** (!TextUtils.*isEmpty*(**et\_cate\_edit**.getText().toString())){  
 **deleteImg**.setVisibility(View.***VISIBLE***);  
 }**else** {  
 **deleteImg**.setVisibility(View.***INVISIBLE***);  
 **et\_cate\_edit**.setHint(**"笔记文件夹名称"**);  
 }  
 }  
 };  
 **et\_cate\_edit**.addTextChangedListener(watcher);  
 }  
  
 @Override  
 **protected void** initListener() {  
 **deleteImg**.setOnClickListener(**new** View.OnClickListener() {  
 @Override  
 **public void** onClick(View v) {  
 **if** (!TextUtils.*isEmpty*(**et\_cate\_edit**.getText().toString())){  
 **et\_cate\_edit**.setText(**""**);  
 **et\_cate\_edit**.setHint(**"笔记文件夹名称"**);  
 }  
 }  
 });  
 }  
 **private void** deleteCategory(){  
 *//笔记不为空删除笔记,为空仅删除文件夹* **if** (**noteManager**.selectCategory(**cate**)!=**null**){  
 **mNoteList**=**noteManager**.selectCategory(**cate**);  
 **for** (Note note:**mNoteList**){  
 **noteManager**.delete(note.getId());*//删除笔记* }  
 }  
 **noteManager**.deleteCategory(**cateId**);*//删除文件夹* SPUtil.*putString*(**mContext**,**"currentCate"**,**"所有"**);  
 goToCateManager();  
 }  
  
 @Override  
 **public boolean** onCreateOptionsMenu(Menu menu) {  
 getMenuInflater().inflate(R.menu.***menu\_category\_edit***,menu);  
 **return true**;  
 }  
 **private void** goBack(){  
 String s=**et\_cate\_edit**.getText().toString();  
 Logger.*i*(s);  
 **if** (!TextUtils.*isEmpty*(s)){  
 Category category=**new** Category();  
 category.setId(**cateId**);*//noteManager需要id* category.setCategory(s);  
 *//category.setPos(catePos);* **noteManager**.updateCategory(category);*//更新文件夹* goToCateManager();  
 }**else** {  
 showDeleteDialog();*//文件夹名称为为空* }  
 }  
 **private void** showDeleteCateAndNoteDialog(){  
 AlertDialog dialog=**new** AlertDialog.Builder(**mContext**).create();  
 dialog.setMessage(**"你确定要删除该文件夹及其笔记吗？"**);  
 dialog.setButton(DialogInterface.***BUTTON\_NEGATIVE***, **"取消"**, **new** DialogInterface.OnClickListener() {  
 @Override  
 **public void** onClick(DialogInterface dialog, **int** which) {  
 **if** (!TextUtils.*isEmpty*(**et\_cate\_edit**.getText().toString())){  
 **et\_cate\_edit**.setHint(**"笔记文件夹名称"**);  
 }  
 }  
 });  
 dialog.setButton(DialogInterface.***BUTTON\_POSITIVE***, **"确定"**, **new** DialogInterface.OnClickListener() {  
 @Override  
 **public void** onClick(DialogInterface dialog, **int** which) {  
 deleteCategory();  
 goToCateManager();  
 }  
 });  
 dialog.show();  
 }  
 **private void** showDeleteDialog(){  
 AlertDialog dialog=**new** AlertDialog.Builder(**mContext**).create();  
 dialog.setMessage(**"文件夹名称不能为空，你确定要删除该文件夹吗？"**);  
 dialog.setButton(DialogInterface.***BUTTON\_NEGATIVE***, **"取消"**, **new** DialogInterface.OnClickListener() {  
 @Override  
 **public void** onClick(DialogInterface dialog, **int** which) {  
 **et\_cate\_edit**.setHint(**"笔记文件夹名称"**);  
 }  
 });  
 dialog.setButton(DialogInterface.***BUTTON\_POSITIVE***, **"确定"**, **new** DialogInterface.OnClickListener() {  
 @Override  
 **public void** onClick(DialogInterface dialog, **int** which) {  
 **noteManager**.deleteCategory(**cateId**);*//删除文件夹* goToCateManager();  
 }  
 });  
 dialog.show();  
 }  
 **private void** goToCateManager(){  
 finish();  
 }  
 @Override  
 **public boolean** onOptionsItemSelected(MenuItem item) {  
 **switch** (item.getItemId()){  
 **case** android.R.id.***home***:  
 goBack();*//判断editView* **return true**;  
 **case** R.id.***delete\_cate***:  
 showDeleteCateAndNoteDialog();  
 **return true**;  
 }  
 **return super**.onOptionsItemSelected(item);  
 }  
  
 @Override  
 **public void** onBackPressed() {  
 goBack();  
 }  
}

**public class** NoteEditActivity **extends** BaseActivity {  
 **private** EditText **et\_noteEdit**;  
 **private** Toolbar **toolbar**;  
 **private** TextView **mTitle**;  
  
 **private** String **id**;  
 **private** String **content**;  
 **private** String **category**;  
 **private** NoteManager **noteManager**;  
  
 @Override  
 **protected int** setLayoutResId() {  
 **return** R.layout.***activity\_note\_edit***;  
 }  
  
 @Override  
 **protected void** initData() {  
 **noteManager**=**new** NoteManager(**mContext**);  
 Intent intent=getIntent();  
 **id**=intent.getStringExtra(**"id"**);  
 **content**=intent.getStringExtra(**"content"**);  
 **category**=intent.getStringExtra(**"currentCate"**);  
 }  
  
 @Override  
 **protected void** initView() {  
 **toolbar**=(Toolbar)findViewById(R.id.***toolbar\_include***);  
 **et\_noteEdit**=(EditText)findViewById(R.id.***et\_note\_edit***);  
 **mTitle**=(TextView)findViewById(R.id.***include\_tv\_center\_title***);  
 **et\_noteEdit**.setText(**content**);  
 **et\_noteEdit**.setSelection(**content**.length());  
 setSupportActionBar(**toolbar**);  
 getSupportActionBar().setDisplayHomeAsUpEnabled(**true**);  
 getSupportActionBar().setDisplayShowHomeEnabled(**true**);  
 setTitle(**""**);  
 **mTitle**.setText(**"编辑"**);  
 }  
 **private void** goBack(){  
 **if** (TextUtils.*isEmpty*(**et\_noteEdit**.getText())){  
 **noteManager**.delete(**id**);  
 goToNoteList();  
 }**else** {  
 String s=**et\_noteEdit**.getText().toString();  
 Note note=**new** Note();  
 note.setId(**id**);*//noteManager需要id* note.setContent(s);  
 Date date=**new** Date();  
 note.setLastTime(Long.*toString*(date.getTime()));  
 note.setCategory(**category**);  
 **noteManager**.update(note);  
 goToNoteList();  
 }  
 }  
 **private void** goToNoteList(){  
 finish();  
 }  
  
 @Override  
 **protected void** initListener() {  
 }  
  
 @Override  
 **public boolean** onCreateOptionsMenu(Menu menu) {  
 getMenuInflater().inflate(R.menu.***menu\_edit***,menu);  
 **return true**;  
 }  
  
 @Override  
 **public boolean** onOptionsItemSelected(MenuItem item) {  
 **switch** (item.getItemId()){  
 **case** android.R.id.***home***:  
 goBack();  
 **return true**;  
 }  
 **return super**.onOptionsItemSelected(item);  
 }  
  
 @Override  
 **public void** onBackPressed() {  
 goBack();  
 }  
}

⑪**分**类管理函数

**public class** CategoryManagerActivity **extends** BaseActivity {  
 **private** Toolbar **mToolbar**;  
 **private** TextView **mTitle**;  
 **private** RecyclerView **recyclerView**;  
 **private** CategoryItemAdapter **adapter**;  
 **private** List<Category>**categoryList**=**new** ArrayList<>();  
 **private** NoteManager **noteManager**;  
  
 @Override  
 **protected int** setLayoutResId() {  
 **return** R.layout.***activity\_category\_manager***;  
 }  
  
 @Override  
 **protected void** initData() {  
 **noteManager**=**new** NoteManager(**mContext**);  
 **categoryList**=**noteManager**.selectAllCategoryBean();  
 Collections.*sort*(**categoryList**);  
 **adapter**=**new** CategoryItemAdapter(**mContext**,**categoryList**);  
 }  
  
 @Override  
 **protected void** onResume() {  
 **categoryList**=**noteManager**.selectAllCategoryBean();  
 Collections.*sort*(**categoryList**);  
 **adapter**.updateList(**categoryList**);  
 **super**.onResume();  
 }  
  
 @Override  
 **protected void** initView() {  
 **mToolbar**=(Toolbar)findViewById(R.id.***toolbar\_include***);  
 **mTitle**=(TextView)findViewById(R.id.***include\_tv\_center\_title***);  
 setSupportActionBar(**mToolbar**);  
 getSupportActionBar().setDisplayHomeAsUpEnabled(**true**);  
 getSupportActionBar().setDisplayShowHomeEnabled(**true**);  
 setTitle(**""**);  
 **mTitle**.setText(**"管理笔记文件夹"**);  
 **recyclerView**=(RecyclerView)findViewById(R.id.***rv\_item\_move***);  
 **recyclerView**.setLayoutManager(**new** LinearLayoutManager(**mContext**));  
 **recyclerView**.setAdapter(**adapter**);  
 Snackbar snackbar=Snackbar.*make*(**recyclerView**,**"长按并拖动手指可调整顺序"**,Snackbar.***LENGTH\_LONG***);  
 snackbar.getView().setBackgroundColor(ContextCompat.*getColor*(**mContext**,R.color.***colorPrimary***));  
 snackbar.show();  
 }  
  
 @Override  
 **protected void** initListener() {  
 MyRvItemTouchCallback myCallback=**new** MyRvItemTouchCallback(**adapter**);  
 myCallback.setDragcolor(ContextCompat.*getColor*(**mContext**,R.color.***colorPrimary***));*//colorPrimary* **final** ItemTouchHelper helper=**new** ItemTouchHelper(myCallback);  
 helper.attachToRecyclerView(**recyclerView**);  
 **recyclerView**.addOnItemTouchListener(**new** OnRvItemTouchListener(**recyclerView**){  
 @Override  
 **public void** onLongClick(RecyclerView.ViewHolder vh) {*//文件夹拖拽模式* **if** (vh.getLayoutPosition()!=0) {*//categoryList.size()-1* helper.startDrag(vh);  
 }  
 }  
 *//holder.getAdapterPosition,getLayoutPosition* @Override  
 **public void** onItemClick(RecyclerView.ViewHolder vh) {*//文件夹编辑模式* **if** (vh.getAdapterPosition()==0)**return**;  
 Intent intent=**new** Intent(**mContext**,CategoryEditActivity.**class**);  
 Category category=**categoryList**.get(vh.getAdapterPosition());  
 intent.putExtra(**"cateId"**,category.getId());  
 intent.putExtra(**"pos"**,category.getPos());  
 intent.putExtra(**"category"**,category.getCategory());  
 startActivity(intent);  
 }  
 });  
 }  
  
 @Override  
 **public boolean** onCreateOptionsMenu(Menu menu) {  
 getMenuInflater().inflate(R.menu.***menu\_manager***,menu);  
 **return true**;  
 }  
 **private void** saveCatePos(){  
 List<Category>newList=**new** ArrayList<>(**categoryList**.size());  
 **for** (**int** i=0;i<**categoryList**.size();i++){  
 Category category=**new** Category();  
 category.setId(**categoryList**.get(i).getId());*//id,必须* category.setPos(String.*valueOf*(i));*//pos* newList.add(category);  
 }  
 **for** (Category category:newList){  
 **noteManager**.updateCategory(category);*//更新文件夹pos* }  
 }  
 **private void** goBack(){  
 Intent intent=**new** Intent(**mContext**,NoteListActivity.**class**);  
 startActivity(intent);  
 finish();  
 }  
  
 @Override  
 **public boolean** onOptionsItemSelected(MenuItem item) {  
 **switch** (item.getItemId()){  
 **case** android.R.id.***home***:  
 goBack();  
 **break**;  
 **case** R.id.***category\_manager***:  
 Snackbar snackbar=Snackbar.*make*(**recyclerView**,**"长按并拖动手指可调整顺序"**,Snackbar.***LENGTH\_LONG***);  
 snackbar.getView().setBackgroundColor(ContextCompat.*getColor*(**mContext**,R.color.***colorPrimary***));  
 snackbar.show();  
 **break**;  
 **case** R.id.***category\_add***:  
 Intent intent=**new** Intent(**mContext**,CategoryAddActivity.**class**);  
 intent.putExtra(**"cateSize"**,**categoryList**.size());  
 startActivity(intent);  
 **break**;  
 }  
 **return true**;  
 }  
  
 @Override  
 **protected void** onPause() {  
 **super**.onPause();  
 saveCatePos();  
 }  
  
 @Override  
 **public void** onBackPressed() {  
 goBack();  
 }  
}

⑫**list操作**

**public class** NoteListActivity **extends** BaseActivity **implements** GetNoteContract.View,  
 OnRecyclerViewItemClickListener{  
 **private** Toolbar **mToolbar**;  
 **private** FloatingActionButton **note\_add**;  
 **private** TextView **mTitle**;  
 **private** ImageView **iv\_update**,**iv\_search**;  
 **private** DrawerLayout **drawerLayout**;  
 **private** RecyclerView **rv\_drawer**;  
 **private** NoteAdapter **mAdapter**;  
 **private** GetNotePresenterImpl **mPresenter**;  
 **private** List<Note>**mNoteList**=**new** ArrayList<>();  
 **private** List<Category>**mCategoryList**=**new** ArrayList<>();  
 **private boolean isDeleteModel**=**false**;  
 **private** NoteManager **noteManager**;  
 **private** CategoryAdapter **cate\_adapter**;  
 **private** HeaderAndFooterWrapper **rv\_wrapper**;  
 **private** LinearLayout **settings**,**cate\_manager**;  
 **private static** String *currentCate*;  
 **private** ActionBarDrawerToggle **toggle**;  
  
  
 @Override  
 **protected int** setLayoutResId() {  
 **return** R.layout.***activity\_note\_list***;  
 }  
  
 @Override  
 **protected void** initData() {  
 *currentCate*=SPUtil.*getString*(**mContext**,**"currentCate"**,**"所有"**);  
 **noteManager**=**new** NoteManager(**mContext**);  
 **mAdapter**=**new** NoteAdapter(**mContext**);  
 **cate\_adapter**=**new** CategoryAdapter(**mContext**,R.layout.***item\_category***,**mCategoryList**);  
 **mPresenter**=**new** GetNotePresenterImpl(**this**);  
 **mPresenter**.getCategory(**mContext**);  
 }  
  
 @Override  
 **protected void** initView() {  
 **mToolbar**=(Toolbar)findViewById(R.id.***toolbar***);  
 setSupportActionBar(**mToolbar**);  
 getSupportActionBar().setDisplayHomeAsUpEnabled(**true**);  
 getSupportActionBar().setDisplayHomeAsUpEnabled(**true**);  
 setTitle(**""**);  
 **iv\_update**=(ImageView)findViewById(R.id.***note\_update***);  
 **iv\_search**=(ImageView)findViewById(R.id.***note\_search***);  
 RecyclerView mRecyclerView=(RecyclerView)findViewById(R.id.***rv\_note***);  
 **note\_add**=(FloatingActionButton)findViewById(R.id.***note\_add***);  
 **mTitle**=(TextView)findViewById(R.id.***tv\_title***);  
 **mTitle**.setText(*currentCate*);  
 mRecyclerView.setLayoutManager(**new** GridLayoutManager(**mContext**,2));  
 mRecyclerView.setAdapter(**mAdapter**);  
 initDrawer();  
 }  
 **private void** initDrawer(){  
 **drawerLayout**=(DrawerLayout)findViewById(R.id.***drawer\_layout***);  
 **toggle**=**new** ActionBarDrawerToggle(**this**,**drawerLayout**,**mToolbar**,  
 R.string.***app\_name***,R.string.***app\_name***){  
 @Override  
 **public void** onDrawerOpened(View drawerView) {  
 **super**.onDrawerOpened(drawerView);  
 }  
  
 @Override  
 **public void** onDrawerClosed(View drawerView) {  
 **super**.onDrawerClosed(drawerView);  
 }  
 };  
 **toggle**.syncState();  
 **drawerLayout**.addDrawerListener(**toggle**);  
 **rv\_drawer**=(RecyclerView)findViewById(R.id.***rv\_drawer\_left***);  
 **rv\_drawer**.setLayoutManager(**new** LinearLayoutManager(**mContext**));*//setLayoutManager  
 //initHeaderAndFooter* View header= LayoutInflater.*from*(**mContext**).inflate(R.layout.***layout\_drawer\_header***,**rv\_drawer**,**false**);  
 View footer= LayoutInflater.*from*(**mContext**).inflate(R.layout.***layout\_drawer\_footer***,**rv\_drawer**,**false**);  
 **settings**=(LinearLayout)footer.findViewById(R.id.***settings***);  
 **cate\_manager**=(LinearLayout)footer.findViewById(R.id.***category\_manager***);  
 **rv\_wrapper**=**new** HeaderAndFooterWrapper(**cate\_adapter**);  
 **rv\_wrapper**.addHeaderView(header);  
 **rv\_wrapper**.addFootView(footer);  
 **rv\_drawer**.setAdapter(**rv\_wrapper**);*//setAdapter* }  
  
 @Override  
 **protected void** initListener() {  
 **iv\_update**.setOnClickListener(**new** View.OnClickListener() {  
 @Override  
 **public void** onClick(View v) {  
 showToast(**"更新到云 待开发"**);  
 }  
 });  
 **iv\_search**.setOnClickListener(**new** View.OnClickListener() {  
 @Override  
 **public void** onClick(View v) {  
 showToast(**"搜索 待开发"**);  
 }  
 });  
 **mAdapter**.setOnRecyclerViewItemClickListener(**this**);  
 **note\_add**.setOnClickListener(**new** View.OnClickListener() {  
 @Override  
 **public void** onClick(View v) {  
 Intent intent=**new** Intent(**mContext**,NoteAddActivity.**class**);  
 startActivity(intent);  
 }  
 });  
 *//点击category,设置category* **cate\_adapter**.setOnItemClickListener(**new** MultiItemTypeAdapter.OnItemClickListener() {  
 @Override  
 **public void** onItemClick(View view, RecyclerView.ViewHolder holder, **int** position) {  
 **mNoteList**.clear();  
 **mAdapter**.updateData(**mNoteList**);  
 *currentCate*=**mCategoryList**.get(position-1).getCategory();*//header* SPUtil.*putString*(**mContext**,**"currentCate"**,*currentCate*);  
 **mTitle**.setText(*currentCate*);  
 **mPresenter**.getNote(**mContext**,*currentCate*);*//获取当前分类笔记* hideDrawer();  
 }  
  
 @Override  
 **public boolean** onItemLongClick(View view, RecyclerView.ViewHolder holder, **int** position) {  
 **return false**;  
 }  
 });  
 **settings**.setOnClickListener(**new** View.OnClickListener() {  
 @Override  
 **public void** onClick(View v) {  
 showToast(**"待开发"**);  
 }  
 });  
 **cate\_manager**.setOnClickListener(**new** View.OnClickListener() {  
 @Override  
 **public void** onClick(View v) {  
 Intent intent=**new** Intent(**mContext**,CategoryManagerActivity.**class**);  
 startActivity(intent);  
 finish();  
 }  
 });  
 }  
  
 @Override  
 **public void** getNoteSuccess(List<Note> list) {  
 **if** (list==**null**){  
 Snackbar snackbar=Snackbar.*make*(**note\_add**,*currentCate*+**"文件夹空空如也哦"**,Snackbar.***LENGTH\_SHORT***);  
 snackbar.getView().setBackgroundColor(ContextCompat.*getColor*(**mContext**,R.color.***colorPrimary***));  
 snackbar.show();  
 }**else** {  
 **mNoteList**=list;  
 **mAdapter**.updateData(list);  
 }  
 }  
  
 @Override  
 **public void** getError(String msg) {  
 }  
  
 @Override  
 **public void** getCategorySuccess(List<Category> categoryList) {  
 **mCategoryList**=categoryList;  
 **cate\_adapter**.updateData(**mCategoryList**);  
 }  
  
 @Override  
 **public void** onItemClick(**int** pos) {  
 **if** (**isDeleteModel**){  
 *//如果是删除模式，单击子项变色* **if** (**mNoteList**.get(pos).isFlag()){*//onBindViewHolder* **mNoteList**.get(pos).setFlag(**false**);  
 }**else** {  
 **mNoteList**.get(pos).setFlag(**true**);  
 }  
 **mAdapter**.notifyDataSetChanged();*//notice* }**else** {  
 *//非删除模式，单击进入编辑模式  
 //SPUtil.putString(mContext,"category",currentCate);* Intent intent=**new** Intent(**this**,NoteEditActivity.**class**);  
 intent.putExtra(**"currentCate"**,*currentCate*);  
 intent.putExtra(**"id"**,**mNoteList**.get(pos).getId());  
 intent.putExtra(**"content"**,**mNoteList**.get(pos).getContent());  
 startActivity(intent);  
 }  
 }  
  
 @Override  
 **protected void** onResume() {  
 **super**.onResume();  
 Logger.*i*(**"嘻嘻嘻"**);  
 **mPresenter**.getNote(**mContext**,*currentCate*);  
 }  
  
 @Override  
 **public boolean** onItemLongClick(**int** pos) {  
 **if** (!**isDeleteModel**){  
 *//进入删除模式* **mNoteList**.get(pos).setFlag(**true**);  
 *//pos添加到deleteItem中* **mAdapter**.notifyDataSetChanged();  
 **isDeleteModel**=**true**;  
 supportInvalidateOptionsMenu();*//通知系统更新菜单* }  
 **return false**;  
 }  
 **private void** deleteNote(){  
 **for** (**int** i=**mNoteList**.size()-1;i>-1;i--){*//倒序，adapter.notifyItemRemoved会更新pos* Note note=**mNoteList**.get(i);  
 **if** (note.isFlag()){  
 **noteManager**.delete(note.getId());  
 **mNoteList**.remove(i);  
 **mAdapter**.notifyItemRemoved(i);  
 }  
 }  
 **isDeleteModel**=**false**;  
 supportInvalidateOptionsMenu();*//通知系统更新菜单* **mNoteList**.clear();  
 **mAdapter**.updateData(**mNoteList**);  
 **mPresenter**.getNote(**mContext**,*currentCate*);  
 }  
  
 @Override  
 **public boolean** onPrepareOptionsMenu(Menu menu) {  
 menu.clear();  
 **if** (**isDeleteModel**){  
 getMenuInflater().inflate(R.menu.***menu\_delete\_mode***,menu);  
 }**else** {  
 getMenuInflater().inflate(R.menu.***menu\_note\_list***,menu);  
 }  
 **return super**.onPrepareOptionsMenu(menu);  
 }  
  
 @Override  
 **public boolean** onCreateOptionsMenu(Menu menu) {  
 getMenuInflater().inflate(R.menu.***menu\_note\_list***,menu);  
 **return true**;  
 }  
  
 @Override  
 **public boolean** onOptionsItemSelected(MenuItem item) {  
 **switch** (item.getItemId()){  
 **case** android.R.id.***home***:  
 **toggle**.onOptionsItemSelected(item);  
 **return true**;  
 **case** R.id.***list\_delete***:  
 Snackbar snackbar=Snackbar.*make*(**note\_add**,**"单击笔记可删除笔记"**,Snackbar.***LENGTH\_LONG***);  
 snackbar.getView().setBackgroundColor(ContextCompat.*getColor*(**mContext**,R.color.***colorPrimary***));  
 snackbar.show();  
 **isDeleteModel**=**true**;  
 supportInvalidateOptionsMenu();*//通知系统更新菜单* **return true**;  
 **case** R.id.***delete***:  
 deleteNote();  
 **return true**;  
 **case** R.id.***manager***:  
 Intent intent=**new** Intent(**mContext**,CategoryManagerActivity.**class**);  
 startActivity(intent);  
 finish();  
 **return true**;  
 }  
 **return super**.onOptionsItemSelected(item);  
 }  
  
 **private void** hideDrawer(){  
 **if** (**drawerLayout**.isDrawerOpen(Gravity.***LEFT***)){  
 **drawerLayout**.closeDrawer(Gravity.***LEFT***);  
 }  
 }  
  
}

### ⑬分类转换函数

**public class** CategoryAdapter **extends** CommonAdapter<Category> {  
 *//private List<Category>categoryList=new ArrayList<>();* **public** CategoryAdapter(**final** Context context, **final int** layoutId, List<Category> datas){  
 **super**(context, layoutId, datas);  
 }  
  
 @Override  
 **protected void** convert(ViewHolder holder, Category category, **int** position) {  
 **if** (position==0)**return**;*//header* **if** (position==1){*//有header* holder.setImageResource(R.id.***iv\_cate***,R.mipmap.***createtask\_fill***);  
 holder.setText(R.id.***tv\_cate***,**"所有"**);  
 }**else** {  
 holder.setText(R.id.***tv\_cate***,category.getCategory());  
 }  
 }  
}

**public class** CategoryItemAdapter **extends** RecyclerView.Adapter<CategoryItemAdapter.ViewHolder>  
 **implements** OnItemDrag{  
 **private** Context **mContext**;  
 **private** List<Category> **list**;  
  
 **public** CategoryItemAdapter(Context context, List<Category> list){  
 **this**.**mContext** = context;  
 **this**.**list** = list;  
 }  
 **public void** updateList(List<Category>categoryList){  
 **this**.**list**=categoryList;  
 notifyDataSetChanged();  
 }  
 @Override  
 **public** ViewHolder onCreateViewHolder(ViewGroup parent, **int** viewType) {  
 **return new** ViewHolder(LayoutInflater.*from*(**mContext**).inflate(R.layout.***item\_category***,parent,**false**));  
 }  
  
 @Override  
 **public void** onBindViewHolder(**final** ViewHolder holder, **int** position) {  
 **if** (position==0){  
 holder.**iv\_cate**.setImageDrawable(ContextCompat.*getDrawable*(**mContext**,R.mipmap.***createtask\_fill***));  
 holder.**tv\_cate**.setText(**list**.get(0).getCategory());  
 }**else** {  
 holder.**tv\_cate**.setText(**list**.get(position).getCategory());  
 holder.**move**.setVisibility(View.***VISIBLE***);  
 }  
  
 }  
  
 @Override  
 **public void** onMove(**int** fromPosition,**int** toPosition){  
 **if** (fromPosition==0||toPosition==0){  
 **return**;  
 }  
 **if** (fromPosition < toPosition) {  
 **for** (**int** i = fromPosition; i < toPosition; i++) {  
 Collections.*swap*(**list**, i, i + 1);  
 }  
 } **else** {  
 **for** (**int** i = fromPosition; i > toPosition; i--) {  
 Collections.*swap*(**list**, i, i - 1);  
 }  
 }  
 notifyItemMoved(fromPosition, toPosition);  
 }  
 @Override  
 **public void** onSwiped(**int** position){  
 **list**.remove(position);  
 notifyItemRemoved(position);  
 }  
  
 @Override  
 **public int** getItemCount() {  
 **return list**.size();  
 }  
  
 **public class** ViewHolder **extends** RecyclerView.ViewHolder{  
 ImageView **iv\_cate**;  
 TextView **tv\_cate**;  
 ImageView **move**;  
  
 **public** ViewHolder(View itemView) {  
 **super**(itemView);  
 **tv\_cate**=(TextView)itemView.findViewById(R.id.***tv\_cate***);  
 **iv\_cate**=(ImageView)itemView.findViewById(R.id.***iv\_cate***);  
 **move**=(ImageView)itemView.findViewById(R.id.***move***);  
 }  
 }  
}

⑭**笔记转换模块**

**public class** NoteAdapter **extends** BaseRecyclerViewAdapter<Note> {  
 **private** Context **mContext**;  
 **public** NoteAdapter(Context context){  
 **this**.**mContext**=context;  
 }  
  
 @Override  
 **public** RecyclerView.ViewHolder onCreateViewHolder(ViewGroup parent, **int** viewType) {  
 View view= LayoutInflater.*from*(**mContext**).inflate(R.layout.***item\_note***,parent,**false**);  
 **return new** NoteHolder(view);  
 }  
  
 @Override  
 **public void** onBindViewHolder(RecyclerView.ViewHolder holder, **int** position) {  
 ((NoteHolder)holder).bindView(**mDataList**.get(position));  
 **if** (**mDataList**.get(position).isFlag()){  
 ((NoteHolder)holder).**root**.setSelected(**true**);  
 ((NoteHolder)holder).**trashImg**.setVisibility(View.***VISIBLE***);  
 }**else** {  
 ((NoteHolder)holder).**root**.setSelected(**false**);  
 ((NoteHolder)holder).**trashImg**.setVisibility(View.***GONE***);  
 }  
 }  
  
 **class** NoteHolder **extends** BaseRvViewHolder{  
 TextView **content**;  
 TextView **title**;  
 TextView **noteFirstTime**;  
 CircleImageView **trashImg**;  
 RelativeLayout **root**;  
 NoteHolder(View itemView){  
 **super**(itemView);  
 **content**=(TextView)itemView.findViewById(R.id.***note\_content***);  
 **title**=(TextView)itemView.findViewById(R.id.***note\_title***);  
 **noteFirstTime**=(TextView)itemView.findViewById(R.id.***note\_date\_first***);  
 **root**=(RelativeLayout) itemView.findViewById(R.id.***note\_root***);  
 **trashImg**=(CircleImageView)itemView.findViewById(R.id.***icon\_trash***);  
 }  
  
 @Override  
 **protected void** bindView(Note note) {  
 **title**.setText(note.getContent());  
 **content**.setText(note.getContent());  
 **noteFirstTime**.setText(note.getFirstTime());  
 }  
 }  
}

⑲当前用户信息函数

public class UserState {

static String USERNAME="";

static String USERPASSWORD="";

static int USERCIKU;

}

* 1. UI界面代码

1. 分类编辑界面

*<?***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/delete\_cate"  
 android:icon="@mipmap/trash"  
 app:showAsAction="always"  
 android:title="删除"**/>  
  
</**menu**>

1. 帮助界面

<?xml version="1.0" encoding="utf-8"?>

<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"

xmlns:tools="http://schemas.android.com/tools"

android:id="@+id/activity\_bangzhu"

android:layout\_width="match\_parent"

android:layout\_height="match\_parent"

android:paddingBottom="@dimen/activity\_vertical\_margin"

android:paddingLeft="@dimen/activity\_horizontal\_margin"

android:paddingRight="@dimen/activity\_horizontal\_margin"

android:paddingTop="@dimen/activity\_vertical\_margin"

android:background="#f8f8ff"

tools:context="com.example.danciji1\_0.Bangzhu">

<Button

android:id="@+id/BZ\_fanhui"

android:layout\_width="60dp"

android:layout\_height="50dp"

android:background="@drawable/back"/>

<LinearLayout

android:layout\_width="match\_parent"

android:layout\_height="match\_parent"

android:orientation="vertical">

<TextView

android:layout\_width="250dp"

android:layout\_height="100dp"

android:layout\_marginTop="20dp"

android:layout\_marginLeft="50dp"

android:gravity="center"

android:text="APP相关信息"

android:textSize="30dp"/>

<LinearLayout

android:layout\_width="match\_parent"

android:layout\_height="match\_parent"

android:background="#f5f5dc"

android:orientation="vertical">

<TextView

android:layout\_width="match\_parent"

android:layout\_height="50dp"

android:text="版本：1.0"

android:textSize="30dp" />

<TextView

android:layout\_width="match\_parent"

android:layout\_height="80dp"

android:text="开发人员：黄叶轩、陈俊仁、陈侃"

android:textSize="30dp"/>

</LinearLayout>

</LinearLayout>

</RelativeLayout>

1. 删除模式界面
2. *<?***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/delete"  
    android:icon="@mipmap/trash"  
    android:title="delete"  
    app:showAsAction="always"**/>  
   </**menu**>
3. 登录界面

<?xml version="1.0" encoding="utf-8"?>

<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"

xmlns:tools="http://schemas.android.com/tools" android:id="@+id/activity\_denglu"

android:layout\_width="match\_parent" android:layout\_height="match\_parent"

android:paddingBottom="@dimen/activity\_vertical\_margin"

android:paddingLeft="@dimen/activity\_horizontal\_margin"

android:paddingRight="@dimen/activity\_horizontal\_margin"

android:paddingTop="@dimen/activity\_vertical\_margin"

android:background="#f8f8ff"

tools:context="com.example.danciji1\_0.Denglu"

android:orientation="horizontal">

<LinearLayout

android:orientation="vertical"

android:layout\_width="300dp"

android:layout\_height="500dp"

android:layout\_margin="20dp"

android:background="#f5f5dc"

android:paddingTop="80dp"

android:paddingLeft="35dp">

<LinearLayout

android:layout\_width="200dp"

android:layout\_height="50dp">

<TextView

android:layout\_width="0dp"

android:layout\_height="50dp"

android:layout\_weight="1"

android:text="账号"

android:textSize="25dp"/>

<EditText

android:id="@+id/DL\_name"

android:layout\_width="0dp"

android:layout\_height="50dp"

android:layout\_weight="2"

android:inputType="text"

android:maxLength="20"

android:drawable="@drawable/color\_cursor"

android:cursorVisible="true" />

</LinearLayout>

<LinearLayout

android:layout\_width="50dp"

android:layout\_height="20dp"></LinearLayout>

<LinearLayout

android:layout\_width="200dp"

android:layout\_height="50dp">

<TextView

android:layout\_width="0dp"

android:layout\_height="50dp"

android:layout\_weight="1"

android:text="密码"

android:textSize="25dp" />

<EditText

android:id="@+id/DL\_mima"

android:layout\_width="0dp"

android:layout\_height="50dp"

android:layout\_weight="2"

android:inputType="textPassword"

android:maxLength="20"

android:drawable="@drawable/color\_cursor"

android:cursorVisible="true" />

</LinearLayout>

<LinearLayout

android:layout\_width="50dp"

android:layout\_height="50dp"></LinearLayout>

<LinearLayout

android:layout\_width="250dp"

android:layout\_height="50dp">

<Button

android:id="@+id/DL\_zhuce"

android:layout\_width="100dp"

android:layout\_height="50dp"

android:text="注册"

android:textSize="30dp"

android:background="#33CCFF" />

<TextView

android:layout\_width="40dp"

android:layout\_height="1dp" />

<Button

android:id="@+id/DL\_denglu"

android:layout\_width="100dp"

android:layout\_height="50dp"

android:text="登录"

android:textSize="30dp"

android:background="#7c8b82" />

</LinearLayout>

<LinearLayout

android:layout\_width="1dp"

android:layout\_height="80dp"></LinearLayout>

<LinearLayout

android:layout\_width="260dp"

android:layout\_height="50dp"

android:gravity="center">

<Button

android:id="@+id/DL\_miandenglu"

android:layout\_width="150dp"

android:layout\_height="50dp"

android:text="免登录试用"

android:textSize="20dp"

android:background="#00ffffff"

android:textColor="#660066" />

</LinearLayout>

</LinearLayout>

</RelativeLayout>

1. 改密码界面

?xml version="1.0" encoding="utf-8"?>

<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"

android:layout\_width="match\_parent"

android:layout\_height="match\_parent"

android:orientation="vertical">

<Button

android:layout\_width="60dp"

android:layout\_height="50dp"

android:id="@+id/GMM\_fanhui"

android:layout\_gravity="bottom"

android:background="@drawable/back"/>

<LinearLayout

android:layout\_width="300dp"

android:layout\_height="430dp"

android:layout\_marginLeft="40dp"

android:layout\_marginTop="10dp"

android:background="#f5f5dc"

android:orientation="vertical"

android:padding="20dp"

>

<LinearLayout

android:layout\_width="250dp"

android:layout\_height="90dp"

android:orientation="vertical">

<TextView

android:layout\_width="250dp"

android:layout\_height="40dp"

android:text="输入您的原始密码："

android:textSize="25dp" />

<EditText

android:inputType="textPassword"

android:layout\_width="150dp"

android:layout\_height="40dp"

android:id="@+id/GMM\_mima1"/>

</LinearLayout>

<LinearLayout

android:layout\_width="250dp"

android:layout\_height="90dp"

android:orientation="vertical">

<TextView

android:layout\_width="250dp"

android:layout\_height="40dp"

android:text="输入您的新密码："

android:textSize="25dp" />

<EditText

android:layout\_width="150dp"

android:layout\_height="40dp"

android:inputType="textPassword"

android:id="@+id/GMM\_mima2"/>

</LinearLayout>

<LinearLayout

android:layout\_width="250dp"

android:layout\_height="90dp"

android:orientation="vertical">

<TextView

android:layout\_width="250dp"

android:layout\_height="40dp"

android:text="再次输入您的新密码："

android:textSize="25dp" />

<EditText

android:inputType="textPassword"

android:layout\_width="150dp"

android:layout\_height="40dp"

android:id="@+id/GMM\_mima3"/>

</LinearLayout>

<LinearLayout

android:layout\_width="match\_parent"

android:layout\_height="110dp">

<LinearLayout

android:layout\_width="150dp"

android:layout\_height="match\_parent"

android:orientation="vertical"

>

</LinearLayout>

<Button

android:id="@+id/GMM\_tijiao"

android:layout\_width="80dp"

android:layout\_height="50dp"

android:layout\_marginTop="60dp"

android:text="确认修改"

android:textSize="20dp"

android:background="#682d782a" />

</LinearLayout>

</LinearLayout>

</LinearLayout>

1. 个人中心界面

<<?xml version="1.0" encoding="utf-8"?>

<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"

xmlns:tools="http://schemas.android.com/tools"

android:id="@+id/activity\_gerenzhongxin"

android:layout\_width="match\_parent"

android:layout\_height="match\_parent"

android:paddingBottom="@dimen/activity\_vertical\_margin"

android:paddingLeft="@dimen/activity\_horizontal\_margin"

android:paddingRight="@dimen/activity\_horizontal\_margin"

android:paddingTop="@dimen/activity\_vertical\_margin"

tools:context="com.example.danciji1\_0.Gerenzhongxin"

android:background="#f8f8ff">

<Button

android:id="@+id/GRZX\_fanhui"

android:layout\_width="60dp"

android:layout\_height="50dp"

android:background="@drawable/back"/>

<Button

android:id="@+id/GRZX\_zhuxiao"

android:layout\_width="50dp"

android:layout\_height="50dp"

android:layout\_marginLeft="300dp"

android:background="@drawable/cancle"/>

<LinearLayout

android:layout\_width="match\_parent"

android:layout\_height="match\_parent"

android:orientation="vertical">

<TextView

android:layout\_width="200dp"

android:layout\_height="50dp"

android:layout\_marginTop="0dp"

android:layout\_marginLeft="70dp"

android:gravity="center"

android:text="个人中心"

android:textSize="40dp"/>

<LinearLayout

android:layout\_width="match\_parent"

android:layout\_height="match\_parent"

android:background="#f5f5dc"

android:orientation="vertical">

<LinearLayout

android:layout\_width="match\_parent"

android:layout\_height="60dp">

<TextView

android:layout\_width="120dp"

android:layout\_height="60dp"

android:paddingTop="10dp"

android:paddingLeft="20dp"

android:text="账号："

android:textSize="30dp"

android:drawable="@drawable/color\_cursor"

android:cursorVisible="true"/>

<TextView

android:layout\_width="match\_parent"

android:layout\_height="match\_parent"

android:id="@+id/GRZX\_zhanghao"

android:text="未登录！！"

android:textSize="30dp"

android:paddingTop="10dp"

android:paddingLeft="20dp"/>

</LinearLayout>

<LinearLayout

android:layout\_width="match\_parent"

android:layout\_height="60dp">

<TextView

android:layout\_width="180dp"

android:layout\_height="60dp"

android:paddingTop="10dp"

android:paddingLeft="20dp"

android:text="个性签名："

android:textSize="30dp"

android:drawable="@drawable/color\_cursor"

android:cursorVisible="true"/>

<EditText

android:layout\_width="match\_parent"

android:layout\_height="match\_parent"

android:id="@+id/GRZX\_gexingqianming"

/>

</LinearLayout>

<LinearLayout

android:layout\_width="match\_parent"

android:layout\_height="60dp">

<TextView

android:layout\_width="120dp"

android:layout\_height="60dp"

android:paddingTop="10dp"

android:paddingLeft="20dp"

android:text="邮箱："

android:textSize="30dp"

android:drawable="@drawable/color\_cursor"

android:cursorVisible="true"/>

<EditText

android:layout\_width="match\_parent"

android:layout\_height="match\_parent"

android:id="@+id/GRZX\_youxiang"

android:textSize="30dp"

android:paddingTop="10dp"

android:paddingLeft="20dp"/>

</LinearLayout>

<TextView

android:layout\_width="match\_parent"

android:layout\_height="60dp"

android:paddingTop="10dp"

android:paddingLeft="20dp"

android:text="当前词库："

android:textSize="30dp"

android:drawable="@drawable/color\_cursor"

android:cursorVisible="true"/>

<LinearLayout

android:layout\_width="400dp"

android:layout\_height="100dp"

android:orientation="vertical"

android:gravity="right">

<RadioGroup

android:id="@+id/GRZX\_zu"

android:layout\_width="200dp"

android:layout\_height="150dp">

<RadioButton

android:id="@+id/GRZX\_siji"

android:layout\_width="200dp"

android:layout\_height="50dp"

android:text="四级词库"

android:textSize="20dp" />

<RadioButton

android:id="@+id/GRZX\_liuji"

android:layout\_width="200dp"

android:layout\_height="50dp"

android:text="六级词库"

android:textSize="20dp"/>

</RadioGroup>

</LinearLayout>

<LinearLayout

android:layout\_width="match\_parent"

android:layout\_height="match\_parent">

<Button

android:id="@+id/GRZX\_gaimima"

android:layout\_width="0dp"

android:layout\_height="match\_parent"

android:layout\_weight="1"

android:background="#33CCFF"

android:text="修改密码"

android:textSize="40dp"/>

<TextView

android:layout\_width="20dp"

android:layout\_height="10dp" />

<Button

android:id="@+id/GRZX\_tijiao"

android:layout\_width="0dp"

android:layout\_height="match\_parent"

android:layout\_weight="1"

android:background="#33CCFF"

android:text="确认修改"

android:textSize="40dp"/>

</LinearLayout>

</LinearLayout>

</LinearLayout>

</RelativeLayout>

1. 管理员界面

<?xml version="1.0" encoding="utf-8"?>

<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"

xmlns:tools="http://schemas.android.com/tools"

android:id="@+id/activity\_shezhi"

android:layout\_width="match\_parent"

android:layout\_height="match\_parent"

android:paddingBottom="@dimen/activity\_vertical\_margin"

android:paddingLeft="@dimen/activity\_horizontal\_margin"

android:paddingRight="@dimen/activity\_horizontal\_margin"

android:paddingTop="@dimen/activity\_vertical\_margin"

tools:context="com.example.danciji1\_0.Shezhi"

android:background="#f8f8ff">

<Button

android:id="@+id/GLY\_fanhui"

android:layout\_width="60dp"

android:layout\_height="50dp"

android:background="@drawable/back"/>

<LinearLayout

android:layout\_width="match\_parent"

android:layout\_height="match\_parent"

android:orientation="vertical">

<Button

android:id="@+id/GLY\_tianjiadanci"

android:layout\_width="150dp"

android:layout\_height="50dp"

android:layout\_marginTop="50dp"

android:layout\_marginLeft="100dp"

android:background="#f5f5dc"

android:text="添加单词"

android:textSize="30dp"/>

<Button

android:id="@+id/GLY\_xiugaidanci"

android:layout\_width="150dp"

android:layout\_height="50dp"

android:layout\_marginTop="50dp"

android:layout\_marginLeft="100dp"

android:background="#f5f5dc"

android:text="修改单词"

android:textSize="30dp"/>

<Button

android:id="@+id/GLY\_shanchudanci"

android:layout\_width="150dp"

android:layout\_height="50dp"

android:layout\_marginTop="30dp"

android:layout\_marginLeft="100dp"

android:background="#f5f5dc"

android:text="删除单词"

android:textSize="30dp"/>

<Button

android:id="@+id/GLY\_fabugonggao"

android:layout\_width="150dp"

android:layout\_height="50dp"

android:layout\_marginTop="30dp"

android:layout\_marginLeft="100dp"

android:background="#f5f5dc"

android:text="发布公告"

android:textSize="30dp"/>

<Button

android:id="@+id/GLY\_fankuixinxi"

android:layout\_width="150dp"

android:layout\_height="50dp"

android:layout\_marginTop="30dp"

android:layout\_marginLeft="100dp"

android:background="#f5f5dc"

android:text="反馈信息"

android:textSize="30dp"/>

</LinearLayout>

</RelativeLayout>

### ⑪菜单管理界面

*<?***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:icon="@mipmap/add"  
 android:title="add"  
 app:showAsAction="always"  
 android:id="@+id/category\_add"**/>  
 <**item  
 android:icon="@mipmap/manage"  
 android:title="manager"  
 android:id="@+id/category\_manager"  
 app:showAsAction="always"**/>  
  
  
</**menu**>

### ⑫笔记列表管理界面

*<?***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/list\_delete"  
 android:icon="@mipmap/trash"  
 android:title="删除模式"  
 app:showAsAction="never"**/>  
 <**item  
 android:id="@+id/manager"  
 android:icon="@mipmap/manage"  
 app:showAsAction="never"  
 android:title="管理笔记本"**/>  
  
</**menu**>

### ⑬文件夹添加页面

*<?***xml version="1.0" encoding="utf-8"***?>*<**LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 android:orientation="vertical"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"**>  
 <**include layout="@layout/layout\_include\_toolbar"**/>  
 <**RelativeLayout  
 android:layout\_marginTop="30dp"  
 android:layout\_marginStart="6dp"  
 android:layout\_marginEnd="6dp"  
 android:layout\_width="match\_parent"  
 android:layout\_height="40dp"**>  
 <**EditText  
 android:maxLines="1"  
 android:maxLength="16"  
 android:textSize="14sp"  
 android:hint="笔记文件夹名称"  
 android:layout\_width="match\_parent"  
 android:layout\_height="40dp"  
 android:id="@+id/et\_cate\_name"**/>  
 <**ImageView  
 android:id="@+id/delete"  
 android:visibility="invisible"  
 android:clickable="true"  
 android:layout\_marginEnd="10dp"  
 android:layout\_alignParentEnd="true"  
 android:layout\_width="30dp"  
 android:layout\_height="30dp"  
 android:src="@mipmap/delete\_edit"**/>  
 </**RelativeLayout**>  
  
</**LinearLayout**>

### ⑭文件夹编辑界面

*<?***xml version="1.0" encoding="utf-8"***?>*<**LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 android:orientation="vertical" android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"**>  
 <**include layout="@layout/layout\_include\_toolbar"**/>  
 <**RelativeLayout  
 android:layout\_marginTop="30dp"  
 android:layout\_marginStart="6dp"  
 android:layout\_marginEnd="6dp"  
 android:layout\_width="match\_parent"  
 android:layout\_height="40dp"**>  
 <**EditText  
 android:maxLines="1"  
 android:maxLength="16"  
 android:layout\_width="match\_parent"  
 android:layout\_height="40dp"  
 android:id="@+id/et\_cate\_edit"**/>  
 <**ImageView  
 android:visibility="visible"  
 android:clickable="true"  
 android:layout\_marginEnd="10dp"  
 android:layout\_alignParentEnd="true"  
 android:layout\_width="30dp"  
 android:layout\_height="30dp"  
 android:src="@mipmap/delete\_edit"**/>  
 </**RelativeLayout**>  
  
</**LinearLayout**>

## ⑮笔记输入界面

*<?***xml version="1.0" encoding="utf-8"***?>*<**android.support.v7.widget.CardView xmlns:android="http://schemas.android.com/apk/res/android"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 xmlns:tools="http://schemas.android.com/tools"  
 xmlns:app="http://schemas.android.com/apk/res-auto"  
 android:layout\_margin="4dp"  
 app:cardCornerRadius="3dp"**>  
 <**RelativeLayout  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:id="@+id/note\_root"  
 android:background="@drawable/bg\_note"**>  
 <**de.hdodenhof.circleimageview.CircleImageView  
 android:layout\_alignParentRight="true"  
 android:layout\_marginTop="2dp"  
 android:layout\_marginRight="2dp"  
 android:layout\_width="22dp"  
 android:layout\_height="22dp"  
 android:id="@+id/icon\_trash"  
 android:src="@mipmap/delete"  
 android:visibility="gone"** />  
 <**LinearLayout  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:orientation="vertical"**>  
 <**TextView  
 android:layout\_marginStart="8dp"  
 android:layout\_marginTop="5dp"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:maxLines="1"  
 android:textSize="14sp"  
 tools:text="哔哩哔哩"  
 android:maxEms="10"  
 android:ellipsize="end"  
 android:textStyle="bold"  
 android:id="@+id/note\_title"**/>  
 <**TextView  
 android:layout\_marginLeft="10dp"  
 android:layout\_marginRight="10dp"  
 android:layout\_marginTop="10dp"  
 android:layout\_marginBottom="5dp"  
 android:layout\_width="match\_parent"  
 android:layout\_height="120dp"  
 tools:text="哔哩哔哩"  
 android:textSize="12sp"  
 android:id="@+id/note\_content"**/>  
 <**TextView  
 android:layout\_width="wrap\_content"  
 android:layout\_height="16dp"  
 android:textSize="8sp"  
 android:id="@+id/note\_date\_first"  
 android:layout\_marginLeft="10dp"  
 android:layout\_marginBottom="4dp"  
 android:text="0920"**/>  
 </**LinearLayout**>  
  
 </**RelativeLayout**>  
  
</**android.support.v7.**

### ⑯主界面

<**LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 android:orientation="vertical" android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginTop="8dp"**>  
 <**View  
 android:layout\_width="match\_parent"  
 android:layout\_height="0.2dp"  
 android:layout\_marginTop="6dp"  
 android:background="#424242"**/>  
 <**View  
 android:layout\_width="match\_parent"  
 android:layout\_height="16dp"** />  
 <**LinearLayout  
 android:id="@+id/settings"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:orientation="horizontal"  
 android:layout\_marginLeft="6dp"**>  
 <**ImageView  
 android:padding="8dp"  
 android:layout\_width="40dp"  
 android:layout\_height="40dp"  
 android:src="@mipmap/setup\_fill"**/>  
 <**TextView  
 android:layout\_marginLeft="20dp"  
 android:gravity="center"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="40dp"  
 android:textSize="14sp"  
 android:text="设置"**/>  
 </**LinearLayout**>  
 <**LinearLayout  
 android:layout\_marginBottom="40dp"  
 android:id="@+id/category\_manager"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:orientation="horizontal"  
 android:layout\_marginLeft="6dp"**>  
 <**ImageView  
 android:padding="8dp"  
 android:layout\_width="40dp"  
 android:layout\_height="40dp"  
 android:src="@mipmap/brush\_fill"**/>  
 <**TextView  
 android:layout\_marginLeft="20dp"  
 android:gravity="center"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="40dp"  
 android:textSize="14sp"  
 android:text="管理笔记本"**/>  
 </**LinearLayout**>  
  
</**LinearLayout**>

*<?***xml version="1.0" encoding="utf-8"***?>*<**LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:background="@color/colorPrimary"  
 android:orientation="vertical"**>  
 <**LinearLayout  
 android:padding="8dp"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:orientation="horizontal"  
 android:gravity="center"  
 android:layout\_marginStart="12dp"  
 android:layout\_marginTop="30dp"  
 android:layout\_marginBottom="30dp"**>  
 <**de.hdodenhof.circleimageview.CircleImageView  
 android:layout\_width="60dp"  
 android:layout\_height="60dp"  
 android:src="@drawable/ic\_bili"**/>  
 <**LinearLayout  
 android:layout\_width="0dp"  
 android:layout\_weight="1"  
 android:layout\_height="60dp"  
 android:orientation="vertical"  
 android:layout\_marginStart="16dp"**>  
  
 <**TextView  
 android:layout\_width="wrap\_content"  
 android:layout\_height="0dp"  
 android:layout\_weight="1"  
 android:gravity="center"  
 android:text="BUBY"  
 android:textSize="14sp"** />  
  
 <**TextView  
 android:layout\_width="wrap\_content"  
 android:layout\_height="0dp"  
 android:layout\_weight="1"  
 android:gravity="center"  
 android:text="账号：buby"  
 android:textSize="14sp"** />  
 </**LinearLayout**>  
 <**ImageView  
 android:layout\_width="26dp"  
 android:layout\_height="26dp"  
 android:layout\_marginRight="20dp"  
 android:src="@mipmap/erweima"**/>  
 </**LinearLayout**>

<**android.support.v7.widget.Toolbar 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:id="@+id/toolbar\_include"  
 android:layout\_width="match\_parent"  
 android:layout\_height="@dimen/include\_toolbar\_height"  
 android:background="@color/colorPrimary"  
 app:theme="@style/ThemeOverlay.AppCompat.Dark.ActionBar"**>  
 <**TextView  
 android:id="@+id/include\_tv\_center\_title"  
 android:gravity="center"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:textColor="@android:color/white"  
 android:textSize="18sp"  
 style="@style/TextAppearance.AppCompat.Widget.ActionBar.Title"**/>  
  
</**android.support.v7.widget.Toolbar**>

### ⑱设置界面

<?xml version="1.0" encoding="utf-8"?>

<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"

xmlns:tools="http://schemas.android.com/tools"

android:id="@+id/activity\_shezhi"

android:layout\_width="match\_parent"

android:layout\_height="match\_parent"

android:paddingBottom="@dimen/activity\_vertical\_margin"

android:paddingLeft="@dimen/activity\_horizontal\_margin"

android:paddingRight="@dimen/activity\_horizontal\_margin"

android:paddingTop="@dimen/activity\_vertical\_margin"

tools:context="com.example.danciji1\_0.Shezhi"

android:background="#f8f8ff">

<Button

android:id="@+id/SZ\_fanhui"

android:layout\_width="60dp"

android:layout\_height="50dp"

android:background="@drawable/back"/>

<Button

android:layout\_width="60dp"

android:layout\_height="60dp"

android:layout\_alignParentRight="true"

android:id="@+id/SZ\_guanliyuan"

android:visibility="visible"

android:background="@drawable/personal"/>

<LinearLayout

android:layout\_width="match\_parent"

android:layout\_height="match\_parent"

android:orientation="vertical">

<Button

android:id="@+id/SZ\_yonghufankui"

android:layout\_width="150dp"

android:layout\_height="50dp"

android:layout\_marginTop="50dp"

android:layout\_marginLeft="100dp"

android:background="#f5f5dc"

android:text="用户反馈"

android:textSize="30dp"/>

<Button

android:id="@+id/SZ\_banbengengxin"

android:layout\_width="150dp"

android:layout\_height="50dp"

android:layout\_marginTop="30dp"

android:layout\_marginLeft="100dp"

android:background="#f5f5dc"

android:text="版本更新"

android:textSize="30dp"/>

<Button

android:id="@+id/SZ\_chongzhiyiyoudanci"

android:layout\_width="150dp"

android:layout\_height="80dp"

android:layout\_marginTop="30dp"

android:layout\_marginLeft="100dp"

android:background="#f5f5dc"

android:text="重置已背单词库"

android:textSize="30dp"/>

<Button

android:id="@+id/SZ\_chongzhizhongdiandanci"

android:layout\_width="150dp"

android:layout\_height="80dp"

android:layout\_marginTop="30dp"

android:layout\_marginLeft="100dp"

android:background="#f5f5dc"

android:text="重置重点单词库"

android:textSize="30dp"/>

</LinearLayout>

</RelativeLayout>

### ⑳用户反馈界面

<?xml version="1.0" encoding="utf-8"?>

<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"

xmlns:tools="http://schemas.android.com/tools"

android:id="@+id/activity\_yonghufankui"

android:layout\_width="match\_parent"

android:layout\_height="match\_parent"

android:paddingBottom="@dimen/activity\_vertical\_margin"

android:paddingLeft="@dimen/activity\_horizontal\_margin"

android:paddingRight="@dimen/activity\_horizontal\_margin"

android:paddingTop="@dimen/activity\_vertical\_margin"

tools:context="com.example.danciji1\_0.Yonghufankui"

android:background="#f8f8ff">

<Button

android:id="@+id/YHFK\_fanhui"

android:layout\_width="60dp"

android:layout\_height="50dp"

android:background="@drawable/back"/>

<LinearLayout

android:layout\_width="match\_parent"

android:layout\_height="match\_parent"

android:orientation="vertical">

<TextView

android:layout\_width="200dp"

android:layout\_height="100dp"

android:layout\_marginTop="20dp"

android:layout\_marginLeft="70dp"

android:gravity="center"

android:text="吐槽池"

android:textSize="50dp"/>

<EditText

android:id="@+id/YHFK\_fankuiinf"

android:layout\_width="match\_parent"

android:layout\_height="match\_parent"

android:background="#f5f5dc" />

</LinearLayout>

<Button

android:id="@+id/YHFK\_tijiao"

android:layout\_width="400dp"

android:layout\_height="100dp"

android:layout\_marginTop="400dp"

android:text="提交"

android:textSize="50dp"/>

</RelativeLayout>

### ㉑注册界面

<?xml version="1.0" encoding="utf-8"?>

<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"

xmlns:tools="http://schemas.android.com/tools"

android:id="@+id/activity\_zhuce"

android:layout\_width="match\_parent"

android:layout\_height="match\_parent"

android:paddingBottom="@dimen/activity\_vertical\_margin"

android:paddingLeft="@dimen/activity\_horizontal\_margin"

android:paddingRight="@dimen/activity\_horizontal\_margin"

android:paddingTop="@dimen/activity\_vertical\_margin"

android:background="#f8f8ff"

tools:context="com.example.danciji1\_0.Zhuce">

<LinearLayout

android:layout\_width="300dp"

android:layout\_height="500dp"

android:orientation="vertical"

android:layout\_margin="20dp"

android:background="#f5f5dc"

android:paddingTop="20dp"

android:paddingLeft="20dp">

<LinearLayout

android:layout\_width="250dp"

android:layout\_height="50dp">

<TextView

android:layout\_width="0dp"

android:layout\_height="50dp"

android:layout\_weight="1"

android:gravity="center"

android:text="账号"

android:textSize="25dp"/>

<EditText

android:id="@+id/ZC\_zhanghao"

android:layout\_width="0dp"

android:layout\_height="50dp"

android:layout\_weight="2"

android:drawable="@drawable/color\_cursor"

android:cursorVisible="true"/>

</LinearLayout>

<LinearLayout

android:layout\_width="250dp"

android:layout\_height="50dp">

<TextView

android:layout\_width="0dp"

android:layout\_height="50dp"

android:layout\_weight="1"

android:gravity="center"

android:text="密码"

android:textSize="25dp"/>

<EditText

android:id="@+id/ZC\_mima1"

android:layout\_width="0dp"

android:layout\_height="50dp"

android:layout\_weight="2"

android:inputType="textPassword"

android:maxLength="20"

android:drawable="@drawable/color\_cursor"

android:cursorVisible="true"/>

</LinearLayout>

<LinearLayout

android:layout\_width="250dp"

android:layout\_height="50dp">

<TextView

android:layout\_width="0dp"

android:layout\_height="50dp"

android:layout\_weight="1"

android:gravity="center"

android:text="确认密码"

android:textSize="25dp" />

<EditText

android:id="@+id/ZC\_mima2"

android:layout\_width="0dp"

android:layout\_height="50dp"

android:layout\_weight="1"

android:inputType="textPassword"

android:maxLength="20"

android:drawable="@drawable/color\_cursor"

android:cursorVisible="true"/>

</LinearLayout>

<LinearLayout

android:layout\_width="250dp"

android:layout\_height="50dp">

<TextView

android:layout\_width="0dp"

android:layout\_height="50dp"

android:layout\_weight="1"

android:gravity="center"

android:text="邮箱"

android:textSize="25dp"/>

<EditText

android:id="@+id/ZC\_youxiang"

android:layout\_width="0dp"

android:layout\_height="50dp"

android:layout\_weight="2"

android:drawable="@drawable/color\_cursor"

android:cursorVisible="true"/>

</LinearLayout>

<LinearLayout

android:layout\_width="200dp"

android:layout\_height="60dp">

<TextView

android:layout\_width="200dp"

android:layout\_height="60dp"

android:padding="10dp"

android:text="选择词库："

android:textSize="25dp"/>

</LinearLayout>

<LinearLayout

android:layout\_width="350dp"

android:layout\_height="150dp"

android:orientation="vertical"

android:gravity="right">

<RadioGroup

android:id="@+id/ZC\_zu"

android:layout\_width="200dp"

android:layout\_height="150dp">

<RadioButton

android:id="@+id/ZC\_siji"

android:layout\_width="200dp"

android:layout\_height="50dp"

android:text="四级词库"

android:textSize="20dp" />

<RadioButton

android:id="@+id/ZC\_liuji"

android:layout\_width="200dp"

android:layout\_height="50dp"

android:text="六级词库"

android:textSize="20dp"/>

</RadioGroup>

</LinearLayout>

<LinearLayout

android:layout\_width="1dp"

android:layout\_height="120dp"></LinearLayout>

</LinearLayout>

<Button

android:layout\_width="290dp"

android:layout\_height="50dp"

android:id="@+id/ZC\_denglu"

android:background="#33CCFF"

android:text="注册并登陆"

android:textSize="40dp"

android:layout\_marginLeft="20dp"

android:layout\_marginTop="410dp"/>

</RelativeLayout>

### ㉒主菜单界面

<?xml version="1.0" encoding="utf-8"?>

<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"

xmlns:tools="http://schemas.android.com/tools"

android:id="@+id/activity\_zhuye"

android:layout\_width="match\_parent"

android:layout\_height="match\_parent"

android:paddingBottom="@dimen/activity\_vertical\_margin"

android:paddingLeft="@dimen/activity\_horizontal\_margin"

android:paddingRight="@dimen/activity\_horizontal\_margin"

android:paddingTop="@dimen/activity\_vertical\_margin"

android:background="#f8f8ff"

tools:context="com.example.danciji1\_0.Zhuye">

<Button

android:id="@+id/ZY\_gonggao"

android:layout\_width="50dp"

android:layout\_height="50dp"

android:background="@drawable/notice"

android:layout\_marginLeft="300dp"/>

<Button

android:id="@+id/ZY\_shezhi"

android:layout\_width="50dp"

android:layout\_height="50dp"

android:background="@drawable/set"

android:layout\_marginTop="60dp"

android:layout\_marginLeft="300dp"/>

<Button

android:id="@+id/ZY\_bangzhu"

android:layout\_width="50dp"

android:layout\_height="50dp"

android:background="@drawable/help"

android:layout\_marginTop="120dp"

android:layout\_marginLeft="300dp"/>

<Button

android:id="@+id/ZY\_gerenzhongxin"

android:layout\_width="50dp"

android:layout\_height="50dp"

android:background="@drawable/personal"/>

<LinearLayout

android:layout\_width="match\_parent"

android:layout\_height="match\_parent"

android:orientation="vertical">

<TextView

android:layout\_width="250dp"

android:layout\_height="100dp"

android:layout\_marginLeft="40dp"

android:layout\_marginTop="50dp"

android:gravity="center"

android:text="单词记"

android:textSize="70dp"

android:textColor="#111111"/>

<Button

android:id="@+id/ZY\_kuaisujiyi"

android:layout\_width="200dp"

android:layout\_height="70dp"

android:background="#f5f5dc"

android:layout\_marginTop="50dp"

android:layout\_marginLeft="70dp"

android:gravity="center"

android:text="快速记忆"

android:textSize="40dp"

android:textColor="#000000"/>

<Button

android:id="@+id/ZY\_kancixuanyi"

android:layout\_width="200dp"

android:layout\_height="70dp"

android:background="#f5f5dc"

android:layout\_marginTop="20dp"

android:layout\_marginLeft="70dp"

android:gravity="center"

android:text="看词选意"

android:textSize="40dp"

android:textColor="#000000"/>

<Button

android:id="@+id/ZY\_kanyixuanci"

android:layout\_width="200dp"

android:layout\_height="70dp"

android:background="#f5f5dc"

android:layout\_marginTop="20dp"

android:layout\_marginLeft="70dp"

android:gravity="center"

android:text="看意选词"

android:textSize="40dp"

android:textColor="#000000"/>

</LinearLayout>

</RelativeLayout>