201730145033宗玥第7次作业

1. 作业要求

在上次作业项目基础上：

1) 实现登录界面的记住密码功能;

2) 增加一个注册页面，注册新的用户名和密码；

3）（拓展选做）实现记住多个账号/密码，记录曾登录成功的账号，如QQ的登录界面可选择登录之前登录的账号。

1. 作业展示

主要功能有：用户登录、注册、注销、修改密码、记住密码共5个基本操作

其内容涉及到以下几点：

1：Button，TextView，TextEdit，ImageView，RatioButton等基本控件的使用；

2：多个Activities之间的切换；

3：使用Android本地数据库来管理用户信息。

登录界面布局文件 activity\_login.xml

*?***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:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
  
 tools:context="com.example.chapter6\_2.LoginActivity"**>  
 <**RelativeLayout  
 android:id="@+id/login\_view"  
 android:layout\_width="400dp"  
 android:layout\_height="800dp"  
 android:layout\_centerInParent="true"** >  
  
 <**Button  
 android:layout\_width="fill\_parent"  
 android:layout\_height="wrap\_content"  
 android:text="注册"  
 android:id="@+id/login\_btn\_register"  
 android:onClick="resetpwd"  
 android:textColor="#ffffff"  
 android:background="#ADD8E6"  
 android:textSize="20dp"  
 android:layout\_below="@+id/login\_btn\_login"  
 android:layout\_alignParentLeft="true"  
 android:layout\_alignParentStart="true"  
 android:layout\_marginTop="10dp"  
 tools:ignore="OnClick"** />  
  
 <**Button  
 android:layout\_width="fill\_parent"  
 android:layout\_height="wrap\_content"  
 android:text="登录"  
 android:id="@+id/login\_btn\_login"  
 android:onClick="finish\_login"  
 android:background="#ADD8E6"  
 android:textSize="20dp"  
 android:textColor="#ffffff"  
 android:layout\_below="@+id/login\_edit\_pwd"  
 android:layout\_alignParentLeft="true"  
 android:layout\_alignParentStart="true"  
 android:layout\_marginTop="52dp"  
 tools:ignore="OnClick"** />  
  
 <**ImageView  
 android:layout\_width="200dp"  
 android:layout\_height="200dp"  
 android:id="@+id/logo"  
 android:layout\_alignParentRight="true"  
 android:layout\_alignParentEnd="true"  
 android:layout\_alignParentLeft="true"  
 android:layout\_alignParentStart="true"  
 android:layout\_alignParentTop="true"  
 android:layout\_alignWithParentIfMissing="false"  
 android:background="@drawable/qq"**/>  
  
 <**EditText  
 android:layout\_width="400dp"  
 android:layout\_height="60dp"  
 android:inputType="textPassword"  
 android:ems="10"  
 android:id="@+id/login\_edit\_pwd"  
 android:drawableLeft="@android:drawable/ic\_lock\_idle\_lock"  
 android:hint="请输入您的密码"  
 android:layout\_below="@+id/login\_edit\_account"  
 android:layout\_alignParentLeft="true"  
 android:layout\_alignParentStart="true"** />  
  
 <**EditText  
 android:layout\_width="400dp"  
 android:layout\_height="60dp"  
 android:inputType="textPersonName"  
 android:id="@+id/login\_edit\_account"  
 android:drawableLeft="@android:drawable/ic\_menu\_myplaces"  
 android:hint="请输入您的用户名"  
 android:layout\_below="@+id/logo"  
 android:layout\_alignParentLeft="true"  
 android:layout\_alignParentStart="true"  
 android:layout\_marginTop="20dp"** />  
  
 <**Button  
 android:layout\_width="fill\_parent"  
 android:layout\_height="wrap\_content"  
 android:text="注销"  
 android:id="@+id/login\_btn\_cancle"  
 android:textSize="20dp"  
 android:layout\_below="@+id/login\_btn\_register"  
 android:layout\_marginTop="10dp"  
 android:layout\_alignParentRight="true"  
 android:layout\_alignParentEnd="true"  
 android:layout\_alignParentLeft="true"  
 android:layout\_alignParentStart="true"  
 android:textColor="#ffffff"  
 android:background="#A9A9A9"** />  
  
 <**CheckBox  
 android:layout\_width="100dp"  
 android:layout\_height="20dp"  
 android:text="记住密码"  
 android:id="@+id/Login\_Remember"  
 android:layout\_below="@+id/login\_edit\_pwd"  
 android:layout\_alignParentLeft="true"  
 android:layout\_alignParentStart="true"  
 android:checked="false"  
 android:textSize="15dp"** />  
  
 <**TextView  
 android:layout\_width="60dp"  
 android:layout\_height="20dp"  
 android:text="修改密码"  
 android:id="@+id/login\_text\_change\_pwd"  
 android:layout\_below="@+id/login\_edit\_pwd"  
 android:layout\_alignParentRight="true"  
 android:layout\_alignParentEnd="true"  
 android:textSize="15dp"** />  
 </**RelativeLayout**>  
 <**RelativeLayout  
 android:id="@+id/login\_success\_view"  
 android:layout\_width="fill\_parent"  
 android:layout\_height="wrap\_content"  
 android:layout\_centerInParent="true"  
 android:layout\_marginLeft="15.0px"  
 android:layout\_marginRight="15.0px"  
 android:layout\_marginTop="62.0px"  
 android:background="#ff3f3f3f"  
 android:paddingBottom="10.0px"  
 android:paddingTop="21.0px"  
 android:visibility="gone"** >  
  
 <**TextView  
 android:id="@+id/login\_success\_show"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_centerInParent="true"  
 android:textColor="#ff3f3f3f"  
 android:textSize="20.0dip"** />  
 </**RelativeLayout**>  
  
</**RelativeLayout**>

实例展示：



注册界面布局文件 activity\_register.xml

*<?***xml version="1.0" encoding="utf-8"***?>*<**RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
  
 android:weightSum="1"**>  
  
 <**EditText  
 android:drawableLeft="@android:drawable/ic\_menu\_myplaces"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="60dp"  
 android:inputType="textPersonName"  
 android:ems="10"  
 android:id="@+id/resetpwd\_edit\_name"  
 android:layout\_alignParentTop="true"  
 android:hint="请输入您的用户名"  
 android:layout\_alignLeft="@+id/resetpwd\_edit\_pwd\_new"  
 android:layout\_alignStart="@+id/resetpwd\_edit\_pwd\_new"  
 android:layout\_alignRight="@+id/resetpwd\_edit\_pwd\_new"  
 android:layout\_alignEnd="@+id/resetpwd\_edit\_pwd\_new"** />  
   
  
 <**EditText  
 android:drawableLeft="@android:drawable/ic\_lock\_idle\_lock"  
 android:layout\_width="fill\_parent"  
 android:layout\_height="60dp"  
 android:inputType="textPassword"  
 android:ems="10"  
 android:id="@+id/resetpwd\_edit\_pwd\_old"  
 android:hint="请输入您的密码"  
 android:layout\_below="@+id/resetpwd\_edit\_name"  
 android:layout\_alignRight="@+id/resetpwd\_edit\_name"  
 android:layout\_alignEnd="@+id/resetpwd\_edit\_name"  
 android:layout\_alignLeft="@+id/resetpwd\_edit\_name"  
 android:layout\_alignStart="@+id/resetpwd\_edit\_name"** />  
  
 <**Button  
 android:layout\_width="fill\_parent"  
 android:layout\_height="wrap\_content"  
 android:text="取消"  
 android:id="@+id/register\_btn\_cancel"  
 android:textSize="20dp"  
 android:background="#A9A9A9"  
 android:layout\_below="@+id/register\_btn\_sure"  
 android:layout\_alignLeft="@+id/register\_btn\_sure"  
 android:layout\_alignStart="@+id/register\_btn\_sure"  
 android:layout\_marginTop="10dp"** />  
  
 <**EditText  
 android:drawableLeft="@android:drawable/ic\_lock\_idle\_lock"  
 android:layout\_width="fill\_parent"  
 android:layout\_height="60dp"  
 android:inputType="textPassword"  
 android:ems="10"  
 android:id="@+id/resetpwd\_edit\_pwd\_new"  
 android:layout\_below="@+id/resetpwd\_edit\_pwd\_old"  
 android:layout\_centerHorizontal="true"  
 android:hint="请确认您的密码"** />  
  
 <**Button  
 android:layout\_width="fill\_parent"  
 android:layout\_height="wrap\_content"  
 android:text="确定"  
 android:id="@+id/register\_btn\_sure"  
 android:textSize="20dp"  
 android:background="#ADD8E6"  
 android:layout\_below="@+id/resetpwd\_edit\_pwd\_new"  
 android:layout\_alignLeft="@+id/resetpwd\_edit\_pwd\_new"  
 android:layout\_alignStart="@+id/resetpwd\_edit\_pwd\_new"  
 android:layout\_marginTop="20dp"** />  
  
</**RelativeLayout**>

实例展示：



修改密码界面展示 activity\_reset\_pwd.xml

*<?***xml version="1.0" encoding="utf-8"***?>*<**RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
  
 android:weightSum="1"**>  
  
 <**EditText  
 android:drawableLeft="@android:drawable/ic\_menu\_myplaces"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="60dp"  
 android:inputType="textPersonName"  
 android:ems="10"  
 android:id="@+id/resetpwd\_edit\_name"  
 android:layout\_alignParentTop="true"  
 android:hint="请输入您的用户名"  
 android:layout\_alignLeft="@+id/resetpwd\_edit\_pwd\_new"  
 android:layout\_alignStart="@+id/resetpwd\_edit\_pwd\_new"  
 android:layout\_alignRight="@+id/resetpwd\_edit\_pwd\_new"  
 android:layout\_alignEnd="@+id/resetpwd\_edit\_pwd\_new"** />  
  
 *<!--  
  
 <EditText android:id="@+id/edt\_operator\_name" style="@style/syncEditText"  
 android:hint="@string/hint\_operator\_name" />  
 <ImageView android:id="@+id/syncOperatorImg" style="@style/syncImageView"  
 android:layout\_alignLeft="@+id/edt\_operator\_name"  
 android:layout\_alignTop="@+id/edt\_operator\_name"  
 android:layout\_alignBottom="@+id/edt\_operator\_name" android:src="@drawable/sync\_operator" />  
 -->* <**EditText  
 android:drawableLeft="@android:drawable/ic\_lock\_idle\_lock"  
 android:layout\_width="fill\_parent"  
 android:layout\_height="60dp"  
 android:inputType="textPassword"  
 android:ems="10"  
 android:id="@+id/resetpwd\_edit\_pwd\_old"  
 android:hint="请输入您的旧密码"  
 android:layout\_below="@+id/resetpwd\_edit\_name"  
 android:layout\_alignRight="@+id/resetpwd\_edit\_name"  
 android:layout\_alignEnd="@+id/resetpwd\_edit\_name"  
 android:layout\_alignLeft="@+id/resetpwd\_edit\_name"  
 android:layout\_alignStart="@+id/resetpwd\_edit\_name"** />  
  
 <**Button  
 android:layout\_width="fill\_parent"  
 android:layout\_height="wrap\_content"  
 android:text="取消"  
 android:id="@+id/resetpwd\_btn\_cancel"  
 android:textSize="20dp"  
 android:onClick="not\_to\_reset"  
 android:background="#A9A9A9"  
 android:layout\_below="@+id/resetpwd\_btn\_sure"  
 android:layout\_alignParentLeft="true"  
 android:layout\_alignParentStart="true"  
 android:layout\_marginTop="20dp"** />  
  
 <**EditText  
 android:drawableLeft="@android:drawable/ic\_lock\_idle\_lock"  
 android:layout\_width="fill\_parent"  
 android:layout\_height="60dp"  
 android:inputType="textPassword"  
 android:ems="10"  
 android:id="@+id/resetpwd\_edit\_pwd\_new"  
 android:layout\_below="@+id/resetpwd\_edit\_pwd\_old"  
 android:layout\_centerHorizontal="true"  
 android:hint="请确认您的新密码"** />  
  
 <**Button  
 android:layout\_width="fill\_parent"  
 android:layout\_height="wrap\_content"  
 android:text="确定"  
 android:id="@+id/resetpwd\_btn\_sure"  
 android:textSize="20dp"  
 android:onClick="sure\_to\_reset"  
 android:background="#ADD8E6"  
 android:layout\_marginTop="92dp"  
 android:layout\_below="@+id/resetpwd\_edit\_pwd\_new"  
 android:layout\_alignParentLeft="true"  
 android:layout\_alignParentStart="true"** />  
  
 <**EditText  
 android:drawableLeft="@android:drawable/ic\_lock\_idle\_lock"  
 android:layout\_width="fill\_parent"  
 android:layout\_height="60dp"  
 android:inputType="textPassword"  
 android:ems="10"  
 android:id="@+id/resetpwd\_edit\_pwd\_check"  
 android:hint="请输入您的新密码"  
 android:layout\_below="@+id/resetpwd\_edit\_pwd\_new"  
 android:layout\_alignParentLeft="true"  
 android:layout\_alignParentStart="true"** />  
  
</**RelativeLayout**>

实例展示：



用户界面展示 activity\_user.xml

*<?***xml version="1.0" encoding="utf-8"***?>*<**RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"** >  
  
<**Button  
 android:id="@+id/force\_offline\_btn"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:text="强制下线"  
 android:onClick="force\_offline"  
 android:textColor="#ffffff"  
 android:textSize="20dp"  
 android:background="#ADD8E6"  
 android:layout\_alignParentLeft="true"  
 android:layout\_alignParentStart="true"  
 android:layout\_alignParentRight="true"  
 android:layout\_alignParentEnd="true"** />  
  
 <**Button  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:text="退出"  
 android:id="@+id/returnback"  
 android:layout\_below="@+id/force\_offline\_btn"  
 android:layout\_marginTop="36dp"  
 android:onClick="back\_to\_login"  
 android:textColor="#ffffff"  
 android:textSize="20dp"  
 android:background="#ADD8E6"  
 android:layout\_alignParentLeft="true"  
 android:layout\_alignParentStart="true"  
 android:layout\_alignParentRight="true"  
 android:layout\_alignParentEnd="true"** />  
  
</**RelativeLayout**>

实例展示：



用户数据管理的类

1. UserData.java【用户数据】

**package** com.example.chapter6\_2;  
  
**public class** UserData {  
 **private** String **userName**; *//用户名* **private** String **userPwd**; *//用户密码* **private int userId**; *//用户ID号* **public int pwdresetFlag**=0;  
 *//获取用户名* **public** String getUserName() { *//获取用户名* **return userName**;  
 }  
 *//设置用户名* **public void** setUserName(String userName) { *//输入用户名* **this**.**userName** = userName;  
 }  
 *//获取用户密码* **public** String getUserPwd() { *//获取用户密码* **return userPwd**;  
 }  
 *//设置用户密码* **public void** setUserPwd(String userPwd) { *//输入用户密码* **this**.**userPwd** = userPwd;  
 }  
 *//获取用户id* **public int** getUserId() { *//获取用户ID号* **return userId**;  
 }  
 *//设置用户id* **public void** setUserId(**int** userId) { *//设置用户ID号* **this**.**userId** = userId;  
 }  
 **public** UserData(String userName, String userPwd) { *//这里只采用用户名和密码* **super**();  
 **this**.**userName** = userName;  
 **this**.**userPwd** = userPwd;  
 }  
  
}

1. UserDataManager【用户管理】
2. **package** com.example.chapter6\_2;  
     
   **import** android.content.ContentValues;  
   **import** android.content.Context;  
   **import** android.database.Cursor;  
   **import** android.database.SQLException;  
   **import** android.database.sqlite.SQLiteDatabase;  
   **import** android.database.sqlite.SQLiteOpenHelper;  
   **import** android.util.Log;  
     
   **public class** UserDataManager { *//用户数据管理类  
    //一些宏定义和声明* **private static final** String ***TAG*** = **"UserDataManager"**;  
    **private static final** String ***DB\_NAME*** = **"user\_data"**;  
    **private static final** String ***TABLE\_NAME*** = **"users"**;  
    **public static final** String ***ID*** = **"\_id"**;  
    **public static final** String ***USER\_NAME*** = **"user\_name"**;  
    **public static final** String ***USER\_PWD*** = **"user\_pwd"**;  
    *// public static final String SILENT = "silent";  
   // public static final String VIBRATE = "vibrate";* **private static final int *DB\_VERSION*** = 2;  
    **private** Context **mContext** = **null**;  
     
    *//创建用户book表* **private static final** String ***DB\_CREATE*** = **"CREATE TABLE "** + ***TABLE\_NAME*** + **" ("** + ***ID*** + **" integer primary key,"** + ***USER\_NAME*** + **" varchar,"** + ***USER\_PWD*** + **" varchar"** + **");"**;  
     
    **private** SQLiteDatabase **mSQLiteDatabase** = **null**;  
    **private** DataBaseManagementHelper **mDatabaseHelper** = **null**;  
     
    *//DataBaseManagementHelper继承自SQLiteOpenHelper* **private static class** DataBaseManagementHelper **extends** SQLiteOpenHelper {  
     
    DataBaseManagementHelper(Context context) {  
    **super**(context, ***DB\_NAME***, **null**, ***DB\_VERSION***);  
    }  
     
    @Override  
    **public void** onCreate(SQLiteDatabase db) {  
    Log.*i*(***TAG***,**"db.getVersion()="**+db.getVersion());  
    db.execSQL(**"DROP TABLE IF EXISTS "** + ***TABLE\_NAME*** + **";"**);  
    db.execSQL(***DB\_CREATE***);  
    Log.*i*(***TAG***, **"db.execSQL(DB\_CREATE)"**);  
    Log.*e*(***TAG***, ***DB\_CREATE***);  
    }  
     
    @Override  
    **public void** onUpgrade(SQLiteDatabase db, **int** oldVersion, **int** newVersion) {  
    Log.*i*(***TAG***, **"DataBaseManagementHelper onUpgrade"**);  
    onCreate(db);  
    }  
    }  
     
    **public** UserDataManager(Context context) {  
    **mContext** = context;  
    Log.*i*(***TAG***, **"UserDataManager construction!"**);  
    }  
    *//打开数据库* **public void** openDataBase() **throws** SQLException {  
    **mDatabaseHelper** = **new** DataBaseManagementHelper(**mContext**);  
    **mSQLiteDatabase** = **mDatabaseHelper**.getWritableDatabase();  
    }  
    *//关闭数据库* **public void** closeDataBase() **throws** SQLException {  
    **mDatabaseHelper**.close();  
    }  
    *//添加新用户，即注册* **public long** insertUserData(UserData userData) {  
    String userName=userData.getUserName();  
    String userPwd=userData.getUserPwd();  
    ContentValues values = **new** ContentValues();  
    values.put(***USER\_NAME***, userName);  
    values.put(***USER\_PWD***, userPwd);  
    **return mSQLiteDatabase**.insert(***TABLE\_NAME***, ***ID***, values);  
    }  
    *//更新用户信息，如修改密码* **public boolean** updateUserData(UserData userData) {  
    *//int id = userData.getUserId();* String userName = userData.getUserName();  
    String userPwd = userData.getUserPwd();  
    ContentValues values = **new** ContentValues();  
    values.put(***USER\_NAME***, userName);  
    values.put(***USER\_PWD***, userPwd);  
    **return mSQLiteDatabase**.update(***TABLE\_NAME***, values,**null**, **null**) > 0;  
    *//return mSQLiteDatabase.update(TABLE\_NAME, values, ID + "=" + id, null) > 0;* }  
    *//* **public** Cursor fetchUserData(**int** id) **throws** SQLException {  
    Cursor mCursor = **mSQLiteDatabase**.query(**false**, ***TABLE\_NAME***, **null**, ***ID*** + **"="** + id, **null**, **null**, **null**, **null**, **null**);  
    **if** (mCursor != **null**) {  
    mCursor.moveToFirst();  
    }  
    **return** mCursor;  
    }  
    *//* **public** Cursor fetchAllUserDatas() {  
    **return mSQLiteDatabase**.query(***TABLE\_NAME***, **null**, **null**, **null**, **null**, **null**,  
    **null**);  
    }  
    *//根据id删除用户* **public boolean** deleteUserData(**int** id) {  
    **return mSQLiteDatabase**.delete(***TABLE\_NAME***, ***ID*** + **"="** + id, **null**) > 0;  
    }  
    *//根据用户名注销* **public boolean** deleteUserDatabyname(String name) {  
    **return mSQLiteDatabase**.delete(***TABLE\_NAME***, ***USER\_NAME*** + **"="** + name, **null**) > 0;  
    }  
    *//删除所有用户* **public boolean** deleteAllUserDatas() {  
    **return mSQLiteDatabase**.delete(***TABLE\_NAME***, **null**, **null**) > 0;  
    }  
     
    *//* **public** String getStringByColumnName(String columnName, **int** id) {  
    Cursor mCursor = fetchUserData(id);  
    **int** columnIndex = mCursor.getColumnIndex(columnName);  
    String columnValue = mCursor.getString(columnIndex);  
    mCursor.close();  
    **return** columnValue;  
    }  
    *//* **public boolean** updateUserDataById(String columnName, **int** id,  
    String columnValue) {  
    ContentValues values = **new** ContentValues();  
    values.put(columnName, columnValue);  
    **return mSQLiteDatabase**.update(***TABLE\_NAME***, values, ***ID*** + **"="** + id, **null**) > 0;  
    }  
    *//根据用户名找用户，可以判断注册时用户名是否已经存在* **public int** findUserByName(String userName){  
    String sql = **"Select \* from users where user\_name =?"**;  
    Log.*i*(***TAG***,**"findUserByName , userName="**+userName);  
    **int** result=0;  
    Cursor mCursor = **mSQLiteDatabase**.rawQuery(sql,**new** String[]{userName});  
    **if**(mCursor!=**null**){  
    result=mCursor.getCount();  
    mCursor.close();  
    Log.*i*(***TAG***,**"findUserByName , result="**+result);  
    }  
    **return** result;  
    }  
    *//根据用户名和密码找用户，用于登录* **public int** findUserByNameAndPwd(String userName,String pwd){  
    Log.*i*(***TAG***,**"findUserByNameAndPwd"**);  
    **int** result=0;  
    String sql = **"Select \* from users where user\_name=? and user\_pwd=?"**;  
    Cursor mCursor=**mSQLiteDatabase**.rawQuery(sql,**new** String[]{userName,pwd});  
    **if**(mCursor!=**null**){  
    result=mCursor.getCount();  
    mCursor.close();  
    Log.*i*(***TAG***,**"findUserByNameAndPwd , result="**+result);  
    }  
    **return** result;  
    }  
     
   }

强制下线功能类：

1. ActivityCollector.java

**package** com.example.chapter6\_2;  
  
**import** android.app.Activity;  
  
**import** java.util.ArrayList;  
**import** java.util.List;  
  
**public class** ActivityCollector {  
  
 **public static** List<Activity> *activities* = **new** ArrayList<>();  
  
 **public static void** addActivity(Activity activity) {  
 *activities*.add(activity);  
 }  
  
 **public static void** removeActivity(Activity activity) {  
 *activities*.remove(activity);  
 }  
  
 **public static void** finishAll() {  
 **for** (Activity activity : *activities*) {  
 **if** (!activity.isFinishing()) {  
 activity.finish();  
 }  
 }  
 }  
  
}

1. BaseActivity.java
2. **package** com.example.chapter6\_2;  
     
   **import** android.app.AlertDialog;  
   **import** android.content.BroadcastReceiver;  
   **import** android.content.Context;  
   **import** android.content.DialogInterface;  
   **import** android.content.Intent;  
   **import** android.content.IntentFilter;  
   **import** android.os.Bundle;  
     
   **import** androidx.appcompat.app.AppCompatActivity;  
     
   **public class** BaseActivity **extends** AppCompatActivity {  
     
    **private** ForceOfflineReceiver **receiver**;  
     
    @Override  
    **protected void** onCreate(Bundle savedInstanceState) {  
    **super**.onCreate(savedInstanceState);  
    ActivityCollector.*addActivity*(**this**);  
    }  
     
    @Override  
    **protected void** onResume() {  
    **super**.onResume();  
    IntentFilter intentFilter = **new** IntentFilter();  
    intentFilter.addAction(**"com.example.chapter6\_2.FORCE\_OFFLINE"**);  
    **receiver** = **new** ForceOfflineReceiver();  
    registerReceiver(**receiver**, intentFilter);  
    }  
     
    @Override  
    **protected void** onPause() {  
    **super**.onPause();  
    **if** (**receiver** != **null**) {  
    unregisterReceiver(**receiver**);  
    **receiver** = **null**;  
    }  
    }  
     
    @Override  
    **protected void** onDestroy() {  
    **super**.onDestroy();  
    ActivityCollector.*removeActivity*(**this**);  
    }  
     
    **class** ForceOfflineReceiver **extends** BroadcastReceiver {  
     
    @Override  
    **public void** onReceive(**final** Context context, Intent intent) {  
    AlertDialog.Builder builder = **new** AlertDialog.Builder(context);  
    builder.setTitle(**"警告"**);  
    builder.setMessage(**"你已被强制下线，请重新登陆。"**);  
    builder.setCancelable(**false**);  
    builder.setPositiveButton(**"OK"**, **new** DialogInterface.OnClickListener() {  
    @Override  
    **public void** onClick(DialogInterface dialog, **int** which) {  
    ActivityCollector.*finishAll*(); *// 销毁所有活动* Intent intent = **new** Intent(context, LoginActivity.**class**);  
    context.startActivity(intent); *// 重新启动LoginActivity* }  
    });  
    builder.show();  
    }  
     
    }  
     
   }

各种功能类：

1. 登录功能：LoginActivity.java

**package** com.example.chapter6\_2;  
  
**import** android.app.Activity;  
**import** android.content.Intent;  
**import** android.content.SharedPreferences;  
**import** android.os.Bundle;  
**import** android.view.View;  
**import** android.view.View.OnClickListener;  
**import** android.widget.Button;  
**import** android.widget.CheckBox;  
**import** android.widget.EditText;  
**import** android.widget.ImageView;  
**import** android.widget.TextView;  
**import** android.widget.Toast;  
  
  
**public class** LoginActivity **extends** BaseActivity { *//登录界面活动* **public int pwdresetFlag**=0;  
 **private** EditText **mAccount**; *//用户名编辑* **private** EditText **mPwd**; *//密码编辑* **private** Button **mRegisterButton**; *//注册按钮* **private** Button **mLoginButton**; *//登录按钮* **private** Button **mCancleButton**; *//注销按钮* **private** CheckBox **mRememberCheck**;  
  
 **private** SharedPreferences **login\_sp**;  
 **private** String **userNameValue**,**passwordValue**;  
  
 **private** View **loginView**; *//登录* **private** View **loginSuccessView**;  
 **private** TextView **loginSuccessShow**;  
 **private** TextView **mChangepwdText**;  
 **private** UserDataManager **mUserDataManager**; *//用户数据管理类* @Override  
 **public void** onCreate(Bundle savedInstanceState) {  
 **super**.onCreate(savedInstanceState);  
 setContentView(R.layout.***activity\_login***);  
 *//通过id找到相应的控件* **mAccount** = (EditText) findViewById(R.id.***login\_edit\_account***);  
 **mPwd** = (EditText) findViewById(R.id.***login\_edit\_pwd***);  
 **mRegisterButton** = (Button) findViewById(R.id.***login\_btn\_register***);  
 **mLoginButton** = (Button) findViewById(R.id.***login\_btn\_login***);  
 **mCancleButton** = (Button) findViewById(R.id.***login\_btn\_cancle***);  
 **loginView**=findViewById(R.id.***login\_view***);  
 **loginSuccessView**=findViewById(R.id.***login\_success\_view***);  
 **loginSuccessShow**=(TextView) findViewById(R.id.***login\_success\_show***);  
  
 **mChangepwdText** = (TextView) findViewById(R.id.***login\_text\_change\_pwd***);  
  
 **mRememberCheck** = (CheckBox) findViewById(R.id.***Login\_Remember***);  
  
 **login\_sp** = getSharedPreferences(**"userInfo"**, 0);  
 String name=**login\_sp**.getString(**"USER\_NAME"**, **""**);  
 String pwd =**login\_sp**.getString(**"PASSWORD"**, **""**);  
 **boolean** choseRemember =**login\_sp**.getBoolean(**"mRememberCheck"**, **false**);  
 **boolean** choseAutoLogin =**login\_sp**.getBoolean(**"mAutologinCheck"**, **false**);  
 *//如果上次选了记住密码，那进入登录页面也自动勾选记住密码，并填上用户名和密码* **if**(choseRemember){  
 **mAccount**.setText(name);  
 **mPwd**.setText(pwd);  
 **mRememberCheck**.setChecked(**true**);  
 }  
  
 **mRegisterButton**.setOnClickListener(**mListener**); *//采用OnClickListener方法设置不同按钮按下之后的监听事件* **mLoginButton**.setOnClickListener(**mListener**);  
 **mCancleButton**.setOnClickListener(**mListener**);  
 **mChangepwdText**.setOnClickListener(**mListener**);  
  
 ImageView image = (ImageView) findViewById(R.id.***logo***); *//使用ImageView显示logo* image.setImageResource(R.drawable.***qq***);  
  
 **if** (**mUserDataManager** == **null**) {  
 **mUserDataManager** = **new** UserDataManager(**this**);  
 **mUserDataManager**.openDataBase(); *//建立本地数据库* }  
 }  
 OnClickListener **mListener** = **new** OnClickListener() { *//不同按钮按下的监听事件选择* **public void** onClick(View v) {  
 **switch** (v.getId()) {  
 **case** R.id.***login\_btn\_register***: *//登录界面的注册按钮* Intent intent\_Login\_to\_Register = **new** Intent(LoginActivity.**this**,RegisterActivity.**class**) ; *//切换Login Activity至User Activity* startActivity(intent\_Login\_to\_Register);  
 finish();  
 **break**;  
 **case** R.id.***login\_btn\_login***: *//登录界面的登录按钮* login();  
 **break**;  
 **case** R.id.***login\_btn\_cancle***: *//登录界面的注销按钮* cancel();  
 **break**;  
 **case** R.id.***login\_text\_change\_pwd***: *//登录界面的修改密码按钮* Intent intent\_Login\_to\_reset = **new** Intent(LoginActivity.**this**,ResetPwdActivity.**class**) ; *//切换Login Activity至User Activity* startActivity(intent\_Login\_to\_reset);  
 finish();  
 **break**;  
 }  
 }  
 };  
  
 **public void** login() { *//登录按钮监听事件* **if** (isUserNameAndPwdValid()) {  
 String userName = **mAccount**.getText().toString().trim(); *//获取当前输入的用户名和密码信息* String userPwd = **mPwd**.getText().toString().trim();  
 SharedPreferences.Editor editor =**login\_sp**.edit();  
 **int** result=**mUserDataManager**.findUserByNameAndPwd(userName, userPwd);  
 **if**(result==1){ *//返回1说明用户名和密码均正确  
 //保存用户名和密码* editor.putString(**"USER\_NAME"**, userName);  
 editor.putString(**"PASSWORD"**, userPwd);  
  
 *//是否记住密码* **if**(**mRememberCheck**.isChecked()){  
 editor.putBoolean(**"mRememberCheck"**, **true**);  
 }**else**{  
 editor.putBoolean(**"mRememberCheck"**, **false**);  
 }  
 editor.commit();  
  
 Intent intent = **new** Intent(LoginActivity.**this**,UserActivity.**class**) ; *//切换Login Activity至User Activity* startActivity(intent);  
 finish();  
 Toast.*makeText*(**this**, getString(R.string.***login\_success***),Toast.***LENGTH\_SHORT***).show();*//登录成功提示* }**else if**(result==0){  
 Toast.*makeText*(**this**, getString(R.string.***login\_fail***),Toast.***LENGTH\_SHORT***).show(); *//登录失败提示* }  
 }  
 }  
 **public void** cancel() { *//注销* **if** (isUserNameAndPwdValid()) {  
 String userName = **mAccount**.getText().toString().trim(); *//获取当前输入的用户名和密码信息* String userPwd = **mPwd**.getText().toString().trim();  
 **int** result=**mUserDataManager**.findUserByNameAndPwd(userName, userPwd);  
 **if**(result==1){ *//返回1说明用户名和密码均正确* Toast.*makeText*(**this**, getString(R.string.***cancel\_success***),Toast.***LENGTH\_SHORT***).show();  
 **mPwd**.setText(**""**);  
 **mAccount**.setText(**""**);  
 **mUserDataManager**.deleteUserDatabyname(userName);  
 }**else if**(result==0){  
 Toast.*makeText*(**this**, getString(R.string.***cancel\_fail***),Toast.***LENGTH\_SHORT***).show(); *//注销失败提示* }  
 }  
  
 }  
 **public boolean** isUserNameAndPwdValid() {  
 **if** (**mAccount**.getText().toString().trim().equals(**""**)) {  
 Toast.*makeText*(**this**, getString(R.string.***account\_empty***),  
 Toast.***LENGTH\_SHORT***).show();  
 **return false**;  
 } **else if** (**mPwd**.getText().toString().trim().equals(**""**)) {  
 Toast.*makeText*(**this**, getString(R.string.***pwd\_empty***),  
 Toast.***LENGTH\_SHORT***).show();  
 **return false**;  
 }  
 **return true**;  
 }  
 @Override  
 **protected void** onResume() {  
 **if** (**mUserDataManager** == **null**) {  
 **mUserDataManager** = **new** UserDataManager(**this**);  
 **mUserDataManager**.openDataBase();  
 }  
 **super**.onResume();  
 }  
 @Override  
 **protected void** onDestroy() {  
 **super**.onDestroy();  
 }  
 @Override  
 **protected void** onPause() {  
 **if** (**mUserDataManager** != **null**) {  
 **mUserDataManager**.closeDataBase();  
 **mUserDataManager** = **null**;  
 }  
 **super**.onPause();  
 }  
}

登陆成功时提示“登录成功”并跳转至用户界面



当用户名为空时提示“用户名为空，请重新输入！”



当密码为空时提示“密码为空，请重新输入！”



当用户名或密码错误或用户不存在时提示“登录失败，请输入正确的用户名和密码！”



当用户不存在注销时提示“注销失败！请输入正确的用户名和密码！”



当注销成功时退出app

1. 注册功能：RegisterActivity.java

**package** com.example.chapter6\_2;  
  
**import** android.content.Intent;  
**import** android.os.Bundle;  
**import** android.view.View;  
**import** android.widget.Button;  
**import** android.widget.EditText;  
**import** android.widget.Toast;  
  
**public class** RegisterActivity **extends** BaseActivity {  
 **private** EditText **mAccount**; *//用户名编辑* **private** EditText **mPwd**; *//密码编辑* **private** EditText **mPwdCheck**; *//密码编辑* **private** Button **mSureButton**; *//确定按钮* **private** Button **mCancelButton**; *//取消按钮* **private** UserDataManager **mUserDataManager**; *//用户数据管理类* @Override  
 **protected void** onCreate(Bundle savedInstanceState) {  
 **super**.onCreate(savedInstanceState);  
 setContentView(R.layout.***activity\_register***);  
 **mAccount** = (EditText) findViewById(R.id.***resetpwd\_edit\_name***);  
 **mPwd** = (EditText) findViewById(R.id.***resetpwd\_edit\_pwd\_old***);  
 **mPwdCheck** = (EditText) findViewById(R.id.***resetpwd\_edit\_pwd\_new***);  
  
 **mSureButton** = (Button) findViewById(R.id.***register\_btn\_sure***);  
 **mCancelButton** = (Button) findViewById(R.id.***register\_btn\_cancel***);  
  
 **mSureButton**.setOnClickListener(**m\_register\_Listener**); *//注册界面两个按钮的监听事件* **mCancelButton**.setOnClickListener(**m\_register\_Listener**);  
  
 **if** (**mUserDataManager** == **null**) {  
 **mUserDataManager** = **new** UserDataManager(**this**);  
 **mUserDataManager**.openDataBase(); *//建立本地数据库* }  
 }  
 View.OnClickListener **m\_register\_Listener** = **new** View.OnClickListener() { *//不同按钮按下的监听事件选择* **public void** onClick(View v) {  
 **switch** (v.getId()) {  
 **case** R.id.***register\_btn\_sure***: *//确认按钮的监听事件* register\_check();  
 **break**;  
 **case** R.id.***register\_btn\_cancel***: *//取消按钮的监听事件,由注册界面返回登录界面* Intent intent\_Register\_to\_Login = **new** Intent(RegisterActivity.**this**,LoginActivity.**class**) ; *//切换User Activity至Login Activity* startActivity(intent\_Register\_to\_Login);  
 finish();  
 **break**;  
 }  
 }  
 };  
 **public void** register\_check() { *//确认按钮的监听事件* **if** (isUserNameAndPwdValid()) {  
 String userName = **mAccount**.getText().toString().trim();  
 String userPwd = **mPwd**.getText().toString().trim();  
 String userPwdCheck = **mPwdCheck**.getText().toString().trim();  
 *//检查用户是否存在* **int** count=**mUserDataManager**.findUserByName(userName);  
 *//用户已经存在时返回，给出提示文字* **if**(count>0){  
 Toast.*makeText*(**this**, getString(R.string.***name\_already\_exist***, userName),Toast.***LENGTH\_SHORT***).show();  
 **return** ;  
 }  
 **if**(userPwd.equals(userPwdCheck)==**false**){ *//两次密码输入不一样* Toast.*makeText*(**this**, getString(R.string.***pwd\_not\_the\_same***),Toast.***LENGTH\_SHORT***).show();  
 **return** ;  
 } **else** {  
 UserData mUser = **new** UserData(userName, userPwd);  
 **mUserDataManager**.openDataBase();  
 **long** flag = **mUserDataManager**.insertUserData(mUser); *//新建用户信息* **if** (flag == -1) {  
 Toast.*makeText*(**this**, getString(R.string.***register\_fail***),Toast.***LENGTH\_SHORT***).show();  
 }**else**{  
 Toast.*makeText*(**this**, getString(R.string.***register\_success***),Toast.***LENGTH\_SHORT***).show();  
 Intent intent\_Register\_to\_Login = **new** Intent(RegisterActivity.**this**,LoginActivity.**class**) ; *//切换User Activity至Login Activity* startActivity(intent\_Register\_to\_Login);  
 finish();  
 }  
 }  
 }  
 }  
 **public boolean** isUserNameAndPwdValid() {  
 **if** (**mAccount**.getText().toString().trim().equals(**""**)) {  
 Toast.*makeText*(**this**, getString(R.string.***account\_empty***),  
 Toast.***LENGTH\_SHORT***).show();  
 **return false**;  
 } **else if** (**mPwd**.getText().toString().trim().equals(**""**)) {  
 Toast.*makeText*(**this**, getString(R.string.***pwd\_empty***),  
 Toast.***LENGTH\_SHORT***).show();  
 **return false**;  
 }**else if**(**mPwdCheck**.getText().toString().trim().equals(**""**)) {  
 Toast.*makeText*(**this**, getString(R.string.***pwd\_check\_empty***),  
 Toast.***LENGTH\_SHORT***).show();  
 **return false**;  
 }  
 **return true**;  
 }  
}

当用户名为空时提示“用户名为空，请重新输入！”



当密码为空时提示“密码为空，请重新输入！”



当密码确认为空时提示“密码确认为空，重新输入密码！”



当密码确认错误时提示“密码确认不正确，请重新输入密码！”



当用户名存在时提示“用户名【xxx】已存在，请重新输入！”



当注册成功时提示“注册成功！”并跳转到登录界面



1. 密码修改功能：ResetPwdActivity.java

**package** com.example.chapter6\_2;  
  
**import** android.content.Intent;  
**import** android.os.Bundle;  
**import** android.view.View;  
**import** android.widget.Button;  
**import** android.widget.EditText;  
**import** android.widget.Toast;  
  
**import** androidx.appcompat.app.AppCompatActivity;  
  
**public class** ResetPwdActivity **extends** BaseActivity {  
 **private** EditText **mAccount**; *//用户名编辑* **private** EditText **mPwd\_old**; *//密码编辑* **private** EditText **mPwd\_new**; *//密码编辑* **private** EditText **mPwdCheck**; *//密码编辑* **private** Button **mSureButton**; *//确定按钮* **private** Button **mCancelButton**; *//取消按钮* **private** UserDataManager **mUserDataManager**; *//用户数据管理类* @Override  
 **protected void** onCreate(Bundle savedInstanceState) {  
 **super**.onCreate(savedInstanceState);  
 setContentView(R.layout.***activity\_reset\_pwd***);  
 **mAccount** = (EditText) findViewById(R.id.***resetpwd\_edit\_name***);  
 **mPwd\_old** = (EditText) findViewById(R.id.***resetpwd\_edit\_pwd\_old***);  
 **mPwd\_new** = (EditText) findViewById(R.id.***resetpwd\_edit\_pwd\_new***);  
 **mPwdCheck** = (EditText) findViewById(R.id.***resetpwd\_edit\_pwd\_check***);  
  
 **mSureButton** = (Button) findViewById(R.id.***resetpwd\_btn\_sure***);  
 **mCancelButton** = (Button) findViewById(R.id.***resetpwd\_btn\_cancel***);  
  
 **mSureButton**.setOnClickListener(**m\_resetpwd\_Listener**); *//注册界面两个按钮的监听事件* **mCancelButton**.setOnClickListener(**m\_resetpwd\_Listener**);  
 **if** (**mUserDataManager** == **null**) {  
 **mUserDataManager** = **new** UserDataManager(**this**);  
 **mUserDataManager**.openDataBase(); *//建立本地数据库* }  
 }  
 View.OnClickListener **m\_resetpwd\_Listener** = **new** View.OnClickListener() { *//不同按钮按下的监听事件选择* **public void** onClick(View v) {  
 **switch** (v.getId()) {  
 **case** R.id.***resetpwd\_btn\_sure***: *//确认按钮的监听事件* resetpwd\_check();  
 **break**;  
 **case** R.id.***resetpwd\_btn\_cancel***: *//取消按钮的监听事件,由注册界面返回登录界面* Intent intent\_Resetpwd\_to\_Login = **new** Intent(ResetPwdActivity.**this**,LoginActivity.**class**) ; *//切换Resetpwd Activity至Login Activity* startActivity(intent\_Resetpwd\_to\_Login);  
 finish();  
 **break**;  
 }  
 }  
 };  
 **public void** resetpwd\_check() { *//确认按钮的监听事件* **if** (isUserNameAndPwdValid()) {  
 String userName = **mAccount**.getText().toString().trim();  
 String userPwd\_old = **mPwd\_old**.getText().toString().trim();  
 String userPwd\_new = **mPwd\_new**.getText().toString().trim();  
 String userPwdCheck = **mPwdCheck**.getText().toString().trim();  
 **int** result=**mUserDataManager**.findUserByNameAndPwd(userName, userPwd\_old);  
 **if**(result==1){ *//返回1说明用户名和密码均正确,继续后续操作* **if**(userPwd\_new.equals(userPwdCheck)==**false**){ *//两次密码输入不一样* Toast.*makeText*(**this**, getString(R.string.***pwd\_not\_the\_same***),Toast.***LENGTH\_SHORT***).show();  
 **return** ;  
 } **else** {  
 UserData mUser = **new** UserData(userName, userPwd\_new);  
 **mUserDataManager**.openDataBase();  
 **boolean** flag = **mUserDataManager**.updateUserData(mUser);  
 **if** (flag == **false**) {  
 Toast.*makeText*(**this**, getString(R.string.***resetpwd\_fail***),Toast.***LENGTH\_SHORT***).show();  
 }**else**{  
 Toast.*makeText*(**this**, getString(R.string.***resetpwd\_success***),Toast.***LENGTH\_SHORT***).show();  
 mUser.**pwdresetFlag**=1;  
 Intent intent\_Register\_to\_Login = **new** Intent(ResetPwdActivity.**this**,LoginActivity.**class**) ; *//切换User Activity至Login Activity* startActivity(intent\_Register\_to\_Login);  
 finish();  
 }  
 }  
 }**else if**(result==0){ *//返回0说明用户名和密码不匹配，重新输入* Toast.*makeText*(**this**, getString(R.string.***pwd\_not\_fit\_user***),Toast.***LENGTH\_SHORT***).show();  
 **return**;  
 }  
 }  
 }  
 **public boolean** isUserNameAndPwdValid() {  
 String userName = **mAccount**.getText().toString().trim();  
 *//检查用户是否存在* **int** count=**mUserDataManager**.findUserByName(userName);  
 *//用户不存在时返回，给出提示文字* **if**(count<=0){  
 Toast.*makeText*(**this**, getString(R.string.***name\_not\_exist***, userName),Toast.***LENGTH\_SHORT***).show();  
 **return false**;  
 }  
 **if** (**mAccount**.getText().toString().trim().equals(**""**)) {  
 Toast.*makeText*(**this**, getString(R.string.***account\_empty***),Toast.***LENGTH\_SHORT***).show();  
 **return false**;  
 } **else if** (**mPwd\_old**.getText().toString().trim().equals(**""**)) {  
 Toast.*makeText*(**this**, getString(R.string.***pwd\_empty***),Toast.***LENGTH\_SHORT***).show();  
 **return false**;  
 } **else if** (**mPwd\_new**.getText().toString().trim().equals(**""**)) {  
 Toast.*makeText*(**this**, getString(R.string.***pwd\_new\_empty***),Toast.***LENGTH\_SHORT***).show();  
 **return false**;  
 }**else if**(**mPwdCheck**.getText().toString().trim().equals(**""**)) {  
 Toast.*makeText*(**this**, getString(R.string.***pwd\_check\_empty***),Toast.***LENGTH\_SHORT***).show();  
 **return false**;  
 }  
 **return true**;  
 }  
}



未输入用户名或用户名错误时时提示“用户名【xxx】不存在，请重新输入”



当旧密码为空时提示“密码为空，请重新输入！”



当旧密码输入错误时提示“密码不正确，请重新输入！”



当新密码为空时提示“新密码为空，请重新输入！”



当确认密码为空时提示“密码确认为空，请重新输入！”



当新密码确认不正确时提示“密码确认不正确，请重新输入新密码！”



修改成功后提示“密码修改成功！”



1. 用户功能：UserActivity.java

**package** com.example.chapter6\_2;  
  
**import** android.content.Intent;  
**import** android.os.Bundle;  
**import** android.view.View;  
**import** android.widget.Button;  
  
**public class** UserActivity **extends** BaseActivity {  
 **private** Button **mReturnButton**;  
 **private** Button **forceOfflineBtn**;  
 @Override  
 **protected void** onCreate(Bundle savedInstanceState) {  
 **super**.onCreate(savedInstanceState);  
 setContentView(R.layout.***activity\_user***);  
 **mReturnButton** = (Button)findViewById(R.id.***returnback***);  
 **forceOfflineBtn**=findViewById(R.id.***force\_offline\_btn***);  
 }  
 **public void** force\_offline(View view){  
 Intent intent = **new** Intent(**"com.example.chapter6\_2.FORCE\_OFFLINE"**);  
 sendBroadcast(intent);  
 }  
  
 **public void** back\_to\_login(View view) {  
 Intent intent3 = **new** Intent(UserActivity.**this**,LoginActivity.**class**) ;  
 startActivity(intent3);  
 finish();  
 }  
}

点击Ok后回到登陆界面（若是选择记住密码，再次回到登录界面时仍保留用户和密码）

 

点击退出后，仍回到登陆界面

