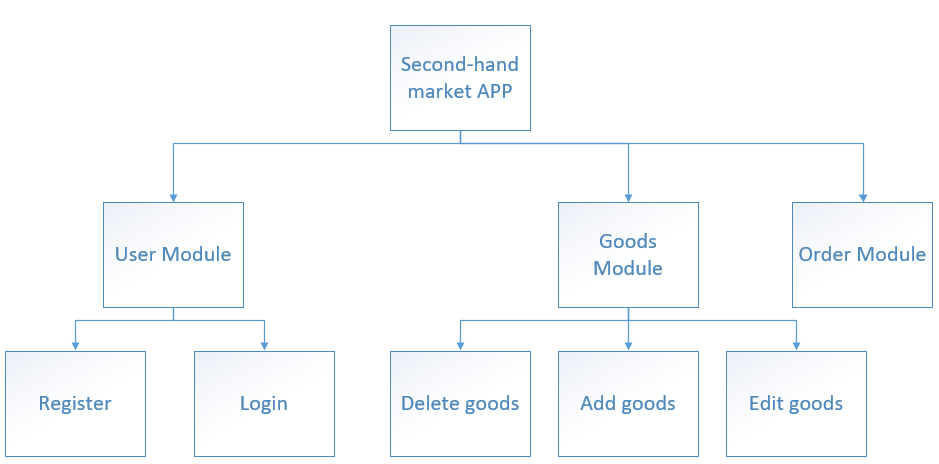
# Case explanation

This project is a second-hand market APP based on Android, which allows people to release their idle goods and sell them for money anytime and anywhere, so as to achieve the purpose of making full use of their idle goods.

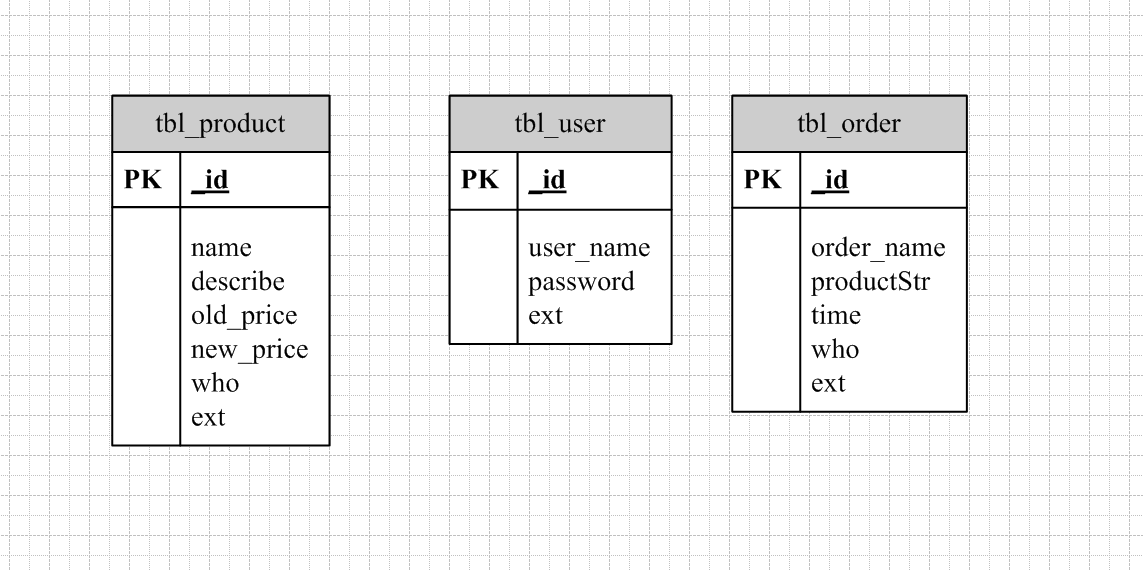
This project will use the four components of Android, such as Activity and Service. What is more, Database Sqlite, the notification component Toast and Fragment, Advanced control TextInputLayout, AutoCompleteTextView and so on are used in this project.

# The process of design and implementation

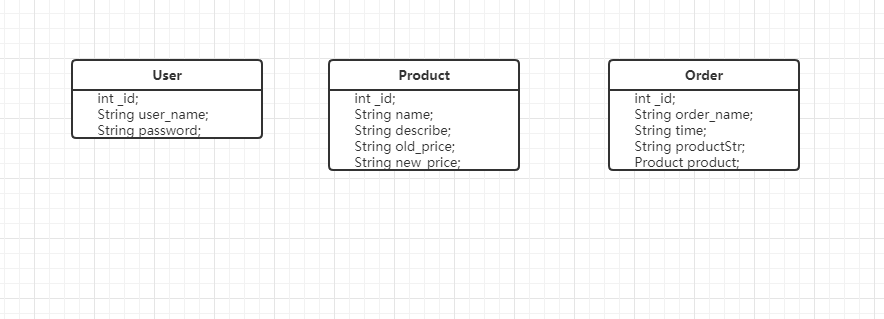
System structure diagram:



E-R diagram of database:



Class diagram：



(1)Control the class “SQLiteOpenHelper” of database by inheriting the class “SQLiteOpenHelper” of system.

(2)With the help of the ‘’MySQLHelper” class of database. We can create and upgrade the database simply and effectively.

(3)The abstract class in the database requires us to create classes to implement it. Two important abstract methods are onCreate() and onUpgrade(), which as their names suggest are called they are called when the database is created and when the data is upgradable, respectively.

(4)Two important instance methods are getReadableDatabase() and getWritableDatabase(). The former opens or creates a readable database, while the latter opens or creates a readable and writable database.

Step one：Login and register

The login and registration functions correspond one by one. The user information is stored on the registration interface, and the user is obtained and verified on the login interface. This function uses the android database Sqlite.

Design database table tbl\_user

String userSql = "create table tbl\_user (" +

"\_id INTEGER PRIMARY KEY AUTOINCREMENT, " +

"user\_name TEXT, " +

"password TEXT, " +

"ext TEXT"+

")";

Design the user object class:

public class User {

int \_id;

String user\_name;

String password;

public int get\_id() {

return \_id;

}

public void set\_id(int \_id) {

this.\_id = \_id;

}

public String getUser\_name() {

return user\_name;

}

public void setUser\_name(String user\_name) {

this.user\_name = user\_name;

}

public String getPassword() {

return password;

}

public void setPassword(String password) {

this.password = password;

}

}

At registration time, the user name and password entered by the user are saved to the database

//returned value specification -2:The user name already exists -1:unknown error Other values indicate that the save is successful

public static long saveUser(String user\_name,String password){

long result = -1;

Cursor cursor = null;

//Database cache

SQLiteDatabase db = MediaContext.getInstance().dbHelper.getReadableDatabase();

try{

String sql = null;

sql = "select \* from "+ ServerConstant.TBL\_USER+" where user\_name = '"+user\_name+"'";

cursor = db.rawQuery(sql,null);

if(cursor.getCount() > 0 ){

return -2;

}

ContentValues cv = new ContentValues();

cv.put("user\_name",user\_name);

cv.put("password",password);

result = db.insert(ServerConstant.TBL\_USER, null, cv);

}catch(Exception e){

e.printStackTrace();

}

return result;

}

On the login page, check whether the user table in the database exists based on the user name and password entered by the user:

//Login and obtain user’s information for verification

public static User getUserInfo(String user\_name, String password) {

User detail = null;

Cursor cursor = null;

SQLiteDatabase db = MediaContext.getInstance().dbHelper.getReadableDatabase();

try{

String sql = null;

sql = "select \* from "+ ServerConstant.TBL\_USER+" where user\_name = '"+user\_name+"' and password = '"+ password+"'";

cursor = db.rawQuery(sql,null);

if(cursor.getCount() > 0 ){

while(cursor.moveToNext()){

detail = new User();

detail.set\_id(cursor.getInt(cursor.getColumnIndex("\_id")));

detail.setUser\_name(cursor.getString(cursor.getColumnIndex("user\_name")));

detail.setPassword(cursor.getString(cursor.getColumnIndex("password")));

}

}

}catch(Exception e){

e.printStackTrace();

return detail;

}finally{

if(cursor != null){

cursor.close();

cursor = null;

}

}

return detail;

}

Here, the query user information is stored in the User object, judging the success of the main interface, identify the prompt User

if (success) {

Intent intent = new Intent(LoginActivity.this,MainActivity.class);

startActivity(intent);

finish();

} else {

mPasswordView.setError("用户名或密码错误!");

mPasswordView.requestFocus();

}

Step two：Goods functions

(1)My second-hand goods list function

Query published product information on the current page

//Query commodity information in the database

**public static** ArrayList<Product> getProductInfoList() {  
 ArrayList<Product> details = **new** ArrayList<>();  
 Cursor cursor = **null**;  
 SQLiteDatabase db = TestContext.*getInstance*().**dbHelper**.getReadableDatabase();  
**try**{  
 String sql = **null**;  
 sql = **"select \* from "**+ ServerConstant.***TBL\_PRODUCT***+**" where who ='"**+TestContext.*getInstance*().getCurrentUser().get\_id()+**"'"**;  
 cursor = db.rawQuery(sql,**null**);  
**if**(cursor.getCount() >0 ){  
**while**(cursor.moveToNext()){  
 Product detail = **new** Product();  
 detail.set\_id(cursor.getInt(cursor.getColumnIndex(**"\_id"**)));  
 detail.setName(cursor.getString(cursor.getColumnIndex(**"name"**)));  
 detail.setDescribe(cursor.getString(cursor.getColumnIndex(**"describe"**)));  
 detail.setNew\_price(cursor.getString(cursor.getColumnIndex(**"old\_price"**)));  
 detail.setOld\_price(cursor.getString(cursor.getColumnIndex(**"new\_price"**)));  
 details.add(detail);  
 }  
 }  
 }**catch**(Exception e){  
 e.printStackTrace();  
**return** details;  
 }**finally**{  
**if**(cursor != **null**){  
 cursor.close();  
 cursor = **null**;  
 }  
 }  
**return** details;  
}

//Delete second-hand goods information

**listview**.setOnItemLongClickListener(**new** AdapterView.OnItemLongClickListener() {  
@Override  
**public boolean** onItemLongClick(AdapterView<?> parent, View view, **final int** pos, **long** id) {  
**new** AlertDialog.Builder(MyProductListActivity.**this**).setTitle(**"确认删除此商品信息吗"**)  
 .setNegativeButton(**"取消"**, **new** DialogInterface.OnClickListener() {  
@Override  
**public void** onClick(DialogInterface dialogInterface, **int** i) {  
 dialogInterface.dismiss();  
 }  
 })  
 .setPositiveButton(**"确定"**, **new** DialogInterface.OnClickListener() {  
@Override  
**public void** onClick(DialogInterface dialogInterface, **int** i) {  
 dialogInterface.dismiss();  
 Product user = **adapter**.getItem(pos);  
**long** result = DatabaseUtils.*deleteProduct*(user.get\_id());  
**if**(result>0){  
 Toast.*makeText*(MyProductListActivity.**this**,**"删除成功!"**,Toast.***LENGTH\_SHORT***).show();  
 }  
**adapter**.getList().remove(pos);  
**adapter**.notifyDataSetChanged();  
 }  
 })  
 .create().show();  
**return true**;  
 }  
});

//The page for editing second-hand goods is displayed

Product user = **adapter**.getItem(pos);  
Intent intent = **new** Intent(MyProductListActivity.**this**, EditProductActivity.**class**);  
intent.putExtra(**"data"**,user);  
startActivity(intent);

//Jump to the interface of new second-hand goods

Intent intent = **new** Intent(MyProductListActivity.**this**, AddProductActivity.**class**);  
startActivity(intent);

(2)Release second-hand goods information

**public static long** saveProduct(String name, String describe, String old\_price, String new\_price){  
**long** result = -1;  
 Cursor cursor = **null**;SQLiteDatabase db = TestContext.*getInstance*().**dbHelper**.getReadableDatabase();  
**try**{  
 ContentValues cv = **new** ContentValues();  
 cv.put(**"name"**,name);  
 cv.put(**"describe"**,describe);  
 cv.put(**"old\_price"**,old\_price);  
 cv.put(**"new\_price"**,new\_price);  
 cv.put(**"who"**,TestContext.*getInstance*().getCurrentUser().get\_id());  
 result = db.insert(ServerConstant.***TBL\_PRODUCT***, **null**, cv);  
 }**catch**(Exception e){  
 e.printStackTrace();  
 }  
**return** result;  
}

(3)Edit second-hand goods information**public static long** saveProduct(Product device){  
**long** result = -1;  
 Cursor cursor = **null**;SQLiteDatabase db = TestContext.*getInstance*().**dbHelper**.getReadableDatabase();  
**try**{  
 ContentValues cv = **new** ContentValues();  
 cv.put(**"name"**,device.getName());  
 cv.put(**"describe"**,device.getDescribe());  
 cv.put(**"old\_price"**,device.getNew\_price());  
 cv.put(**"new\_price"**,device.getOld\_price());  
 result = db.update(ServerConstant.***TBL\_PRODUCT***,cv,**"\_id = ?"**,**new** String[]{String.*valueOf*(device.get\_id())});  
 }**catch**(Exception e){  
 e.printStackTrace();  
 }  
**return** result;  
}

Step three:Second-hand order function

(1)My second-hand goods order list function

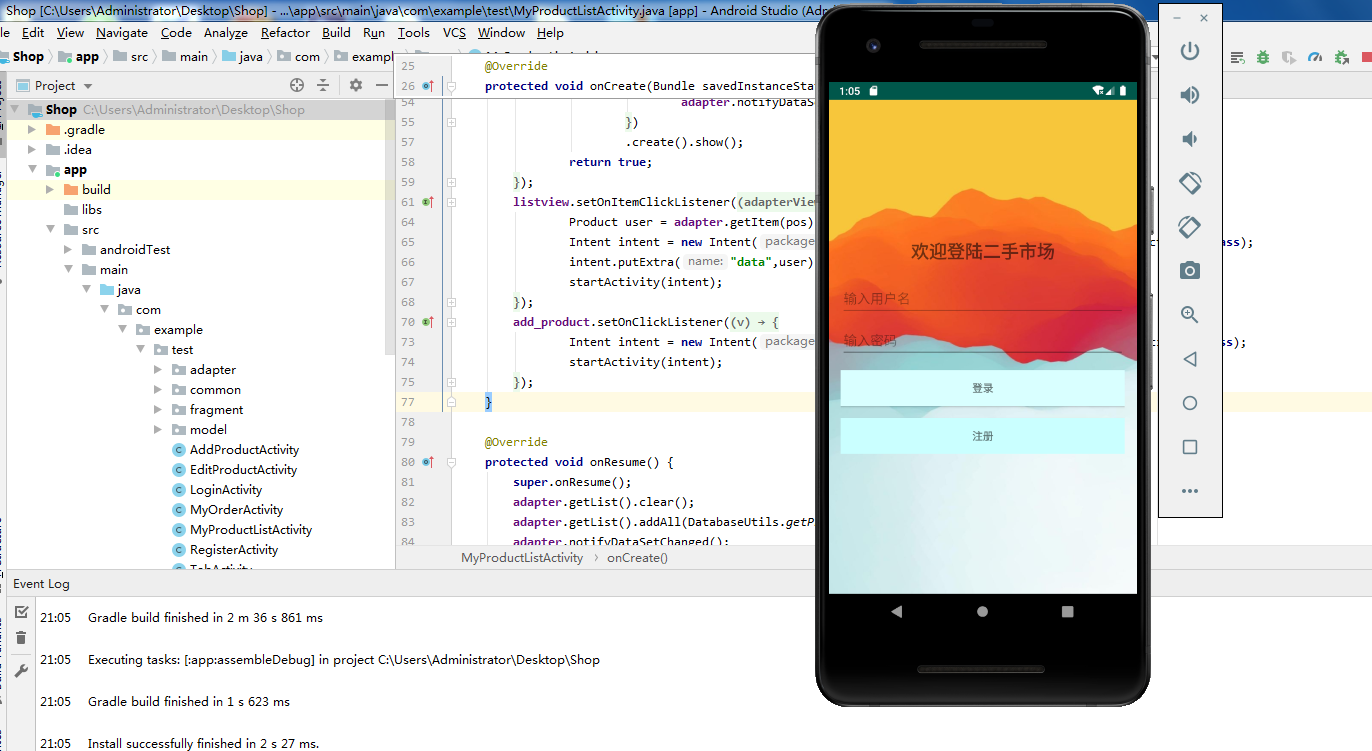
//Order list information

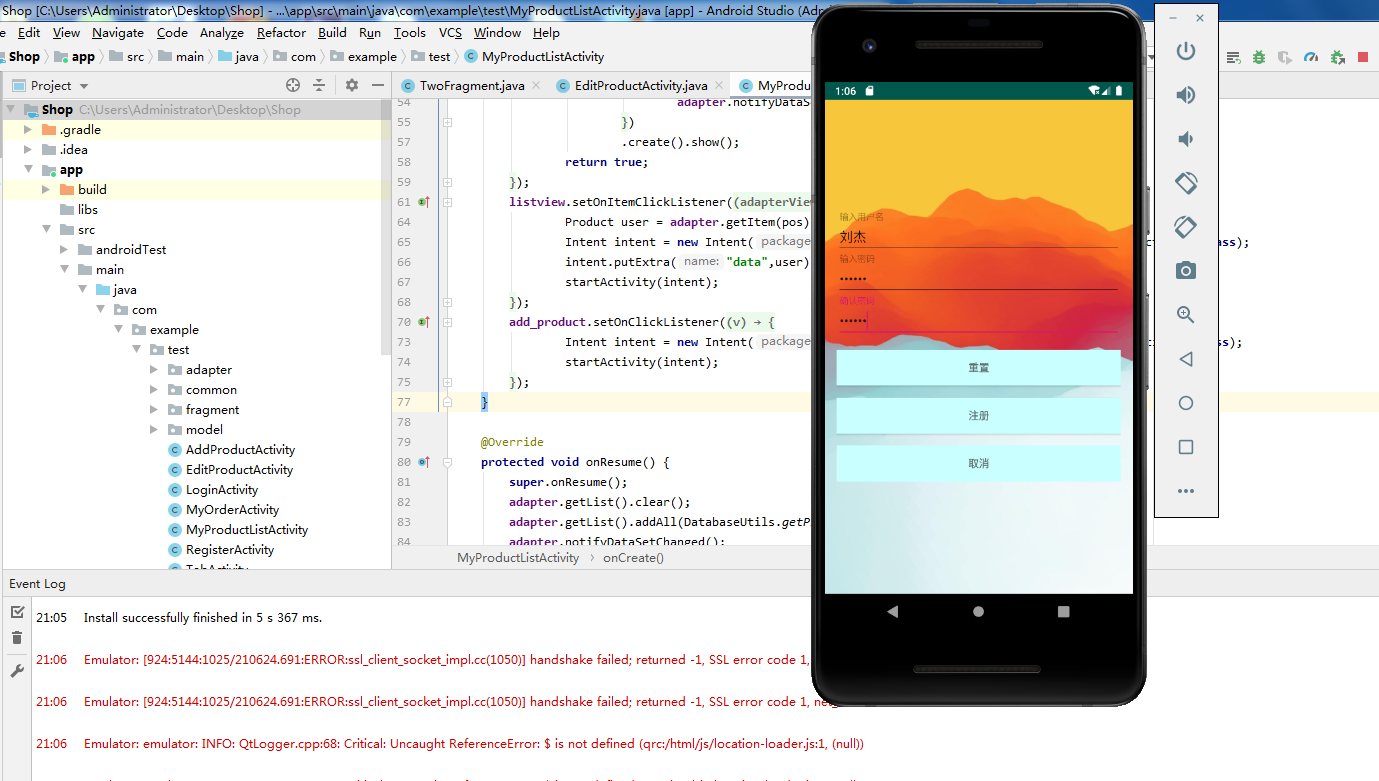
**public static** ArrayList<Order> getOrderList() {  
 ArrayList<Order> details = **new** ArrayList<>();  
 Cursor cursor = **null**;  
 SQLiteDatabase db = TestContext.*getInstance*().**dbHelper**.getReadableDatabase();  
**try**{  
 String sql = **null**;  
 sql = **"select \* from "**+ ServerConstant.***TBL\_ORDER***+**" where who ='"**+TestContext.*getInstance*().getCurrentUser().get\_id()+**"'"**;  
 cursor = db.rawQuery(sql,**null**);  
**if**(cursor.getCount() >0 ){  
 Gson gson = **new** Gson();  
**while**(cursor.moveToNext()){  
 Order detail = **new** Order();  
 detail.set\_id(cursor.getInt(cursor.getColumnIndex(**"\_id"**)));  
 detail.setOrder\_name(cursor.getString(cursor.getColumnIndex(**"order\_name"**)));  
 detail.setTime(cursor.getString(cursor.getColumnIndex(**"time"**)));  
 detail.setProductStr(cursor.getString(cursor.getColumnIndex(**"productStr"**)));  
 Product product = gson.fromJson(detail.getProductStr(),Product.**class**);  
 detail.setProduct(product);  
 details.add(detail);  
 }  
 }  
 }**catch**(Exception e){  
 e.printStackTrace();  
**return** details;  
 }**finally**{  
**if**(cursor != **null**){  
 cursor.close();  
 cursor = **null**;  
 }  
 }  
**return** details;  
}

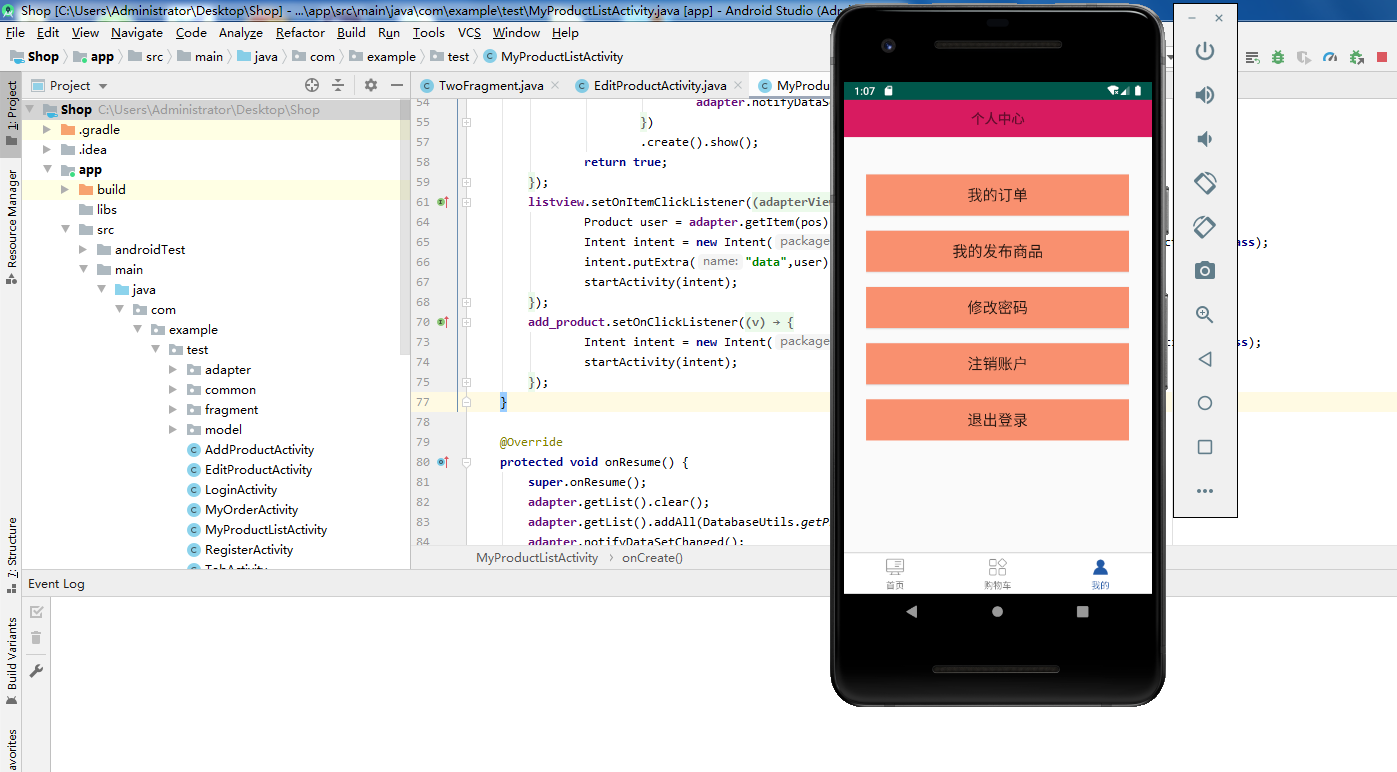
//Delete order information

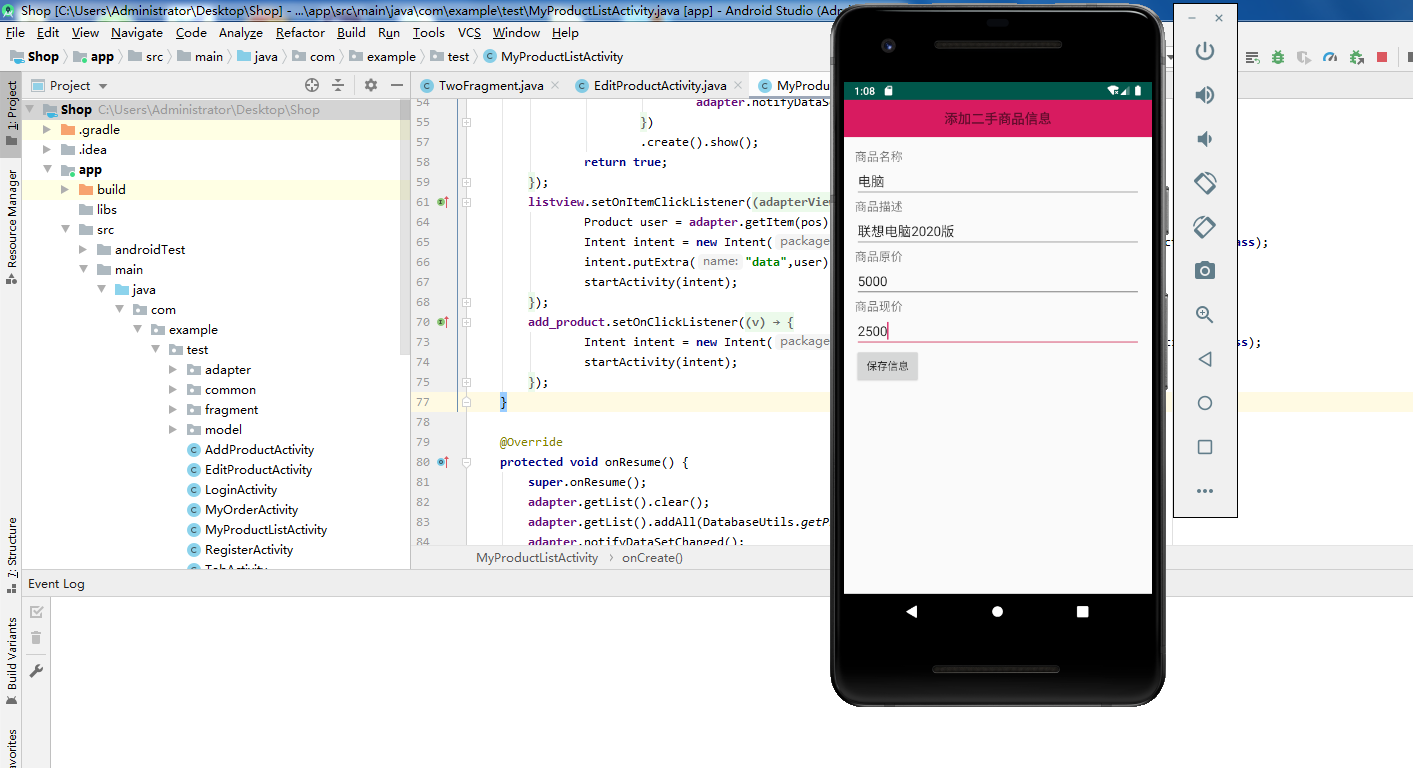
**listview**.setOnItemLongClickListener(**new** AdapterView.OnItemLongClickListener() {  
@Override  
**public boolean** onItemLongClick(AdapterView<?> parent, View view, **final int** pos, **long** id) {  
**new** AlertDialog.Builder(MyOrderActivity.**this**).setTitle(**"确认删除此订单吗"**)  
 .setNegativeButton(**"取消"**, **new** DialogInterface.OnClickListener() {  
@Override  
**public void** onClick(DialogInterface dialogInterface, **int** i) {  
 dialogInterface.dismiss();  
 }  
 })  
 .setPositiveButton(**"确定"**, **new** DialogInterface.OnClickListener() {  
@Override  
**public void** onClick(DialogInterface dialogInterface, **int** i) {  
 dialogInterface.dismiss();  
 Order order = **adapter**.getItem(pos);  
**long** result = DatabaseUtils.*deleteOrder*(order.get\_id());  
**if**(result>0){  
 Toast.*makeText*(MyOrderActivity.**this**,**"删除成功!"**,Toast.***LENGTH\_SHORT***).show();  
 }  
**adapter**.getList().remove(pos);  
**adapter**.notifyDataSetChanged();  
 }  
 })  
 .create().show();  
**return true**;  
 }  
});

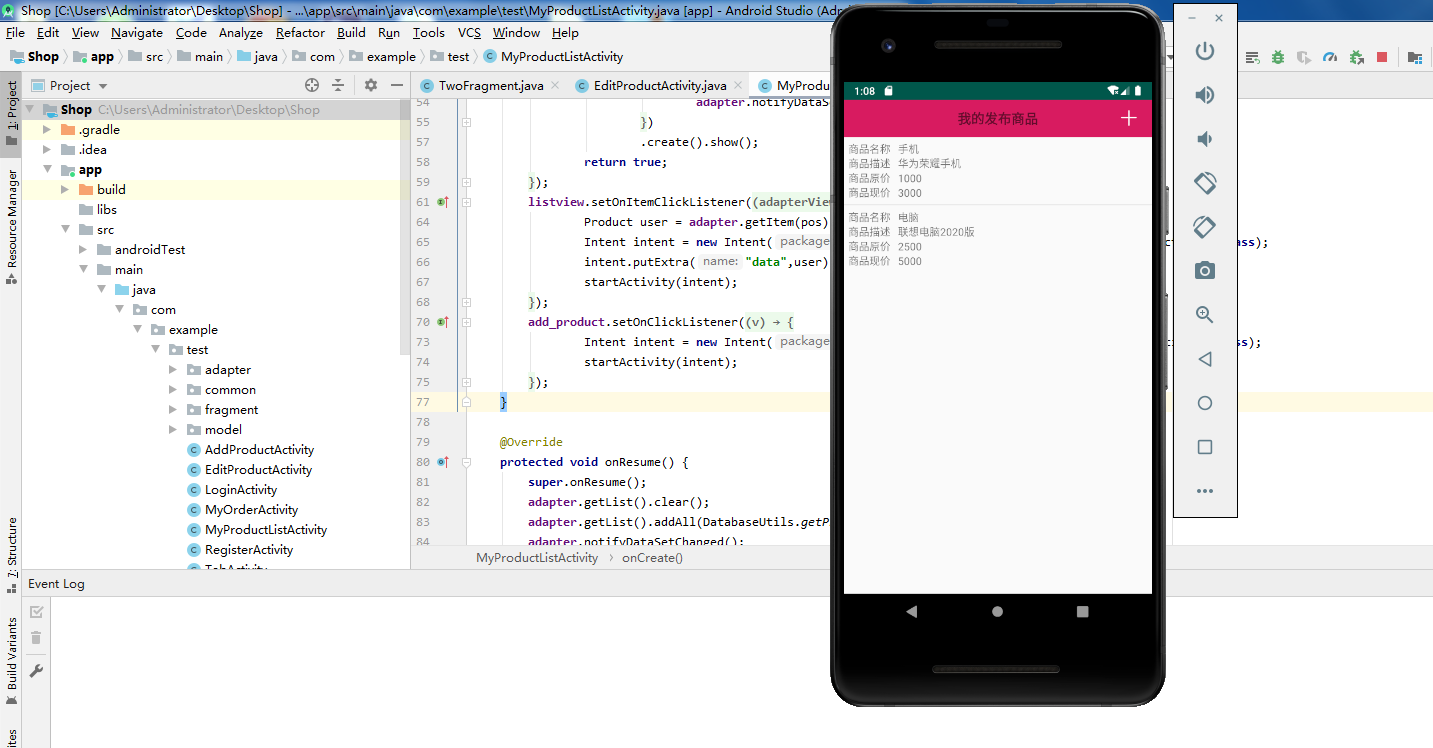
**The display of results**

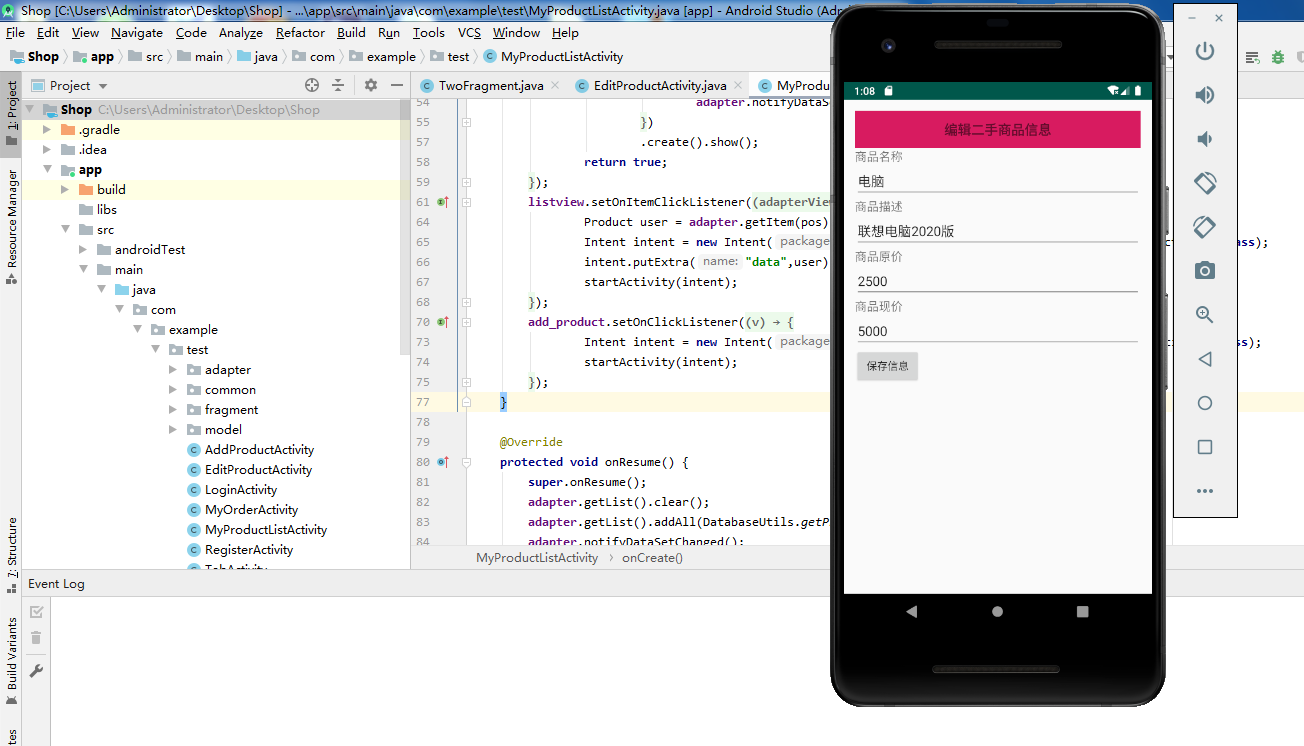


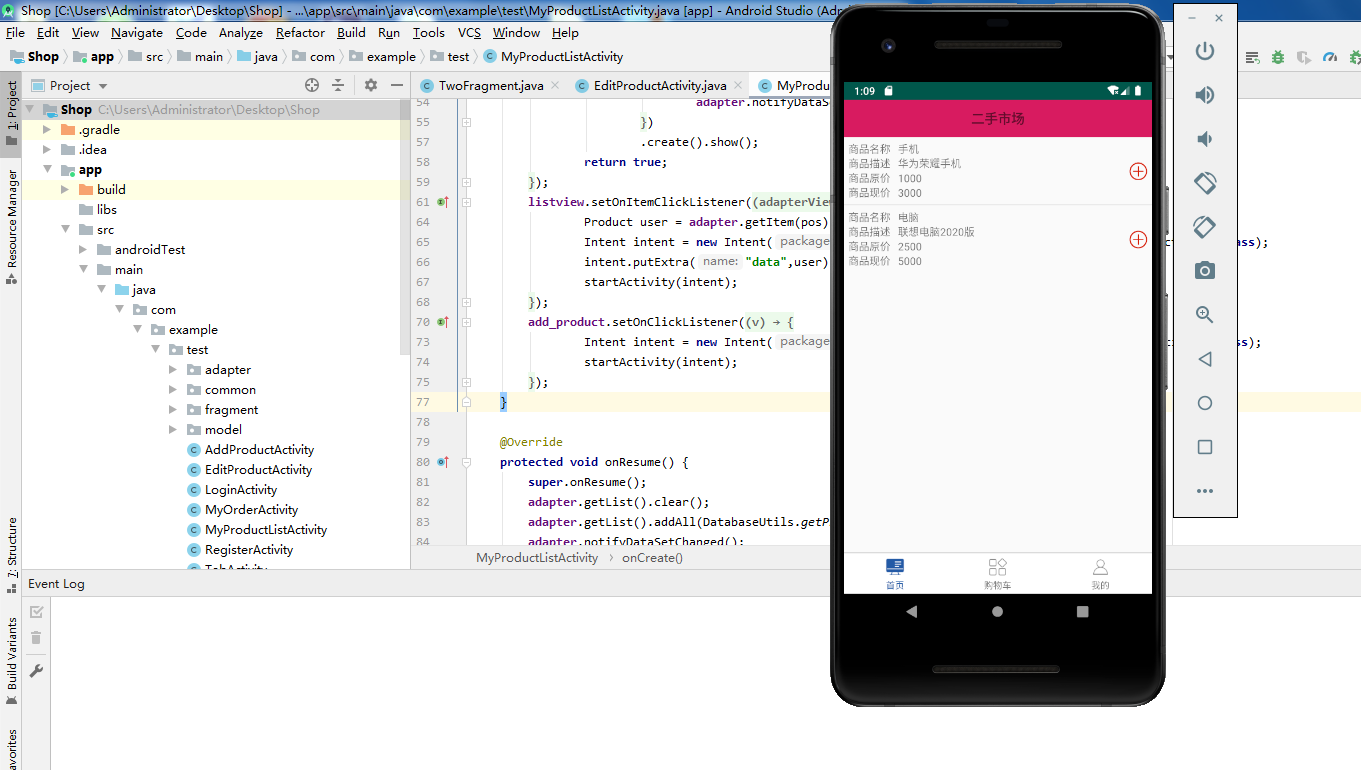


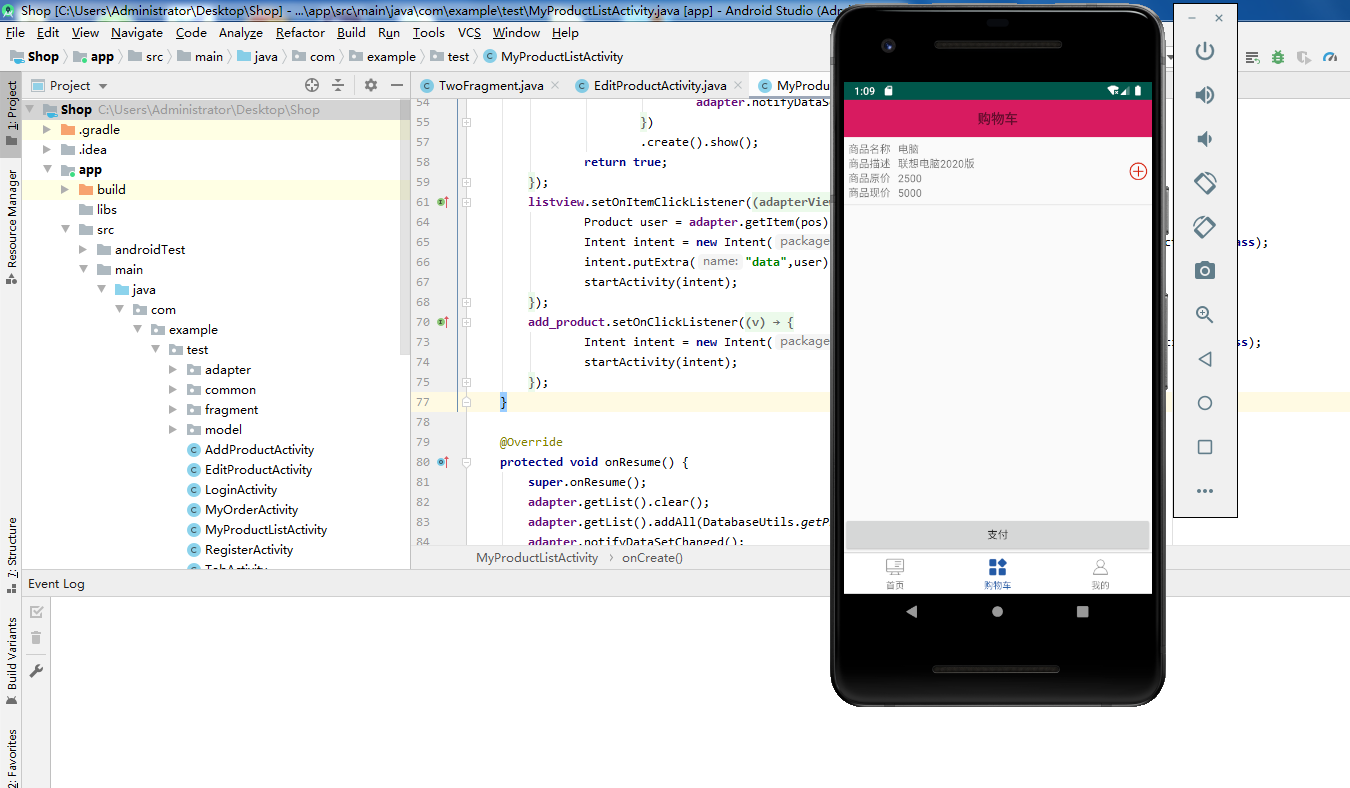


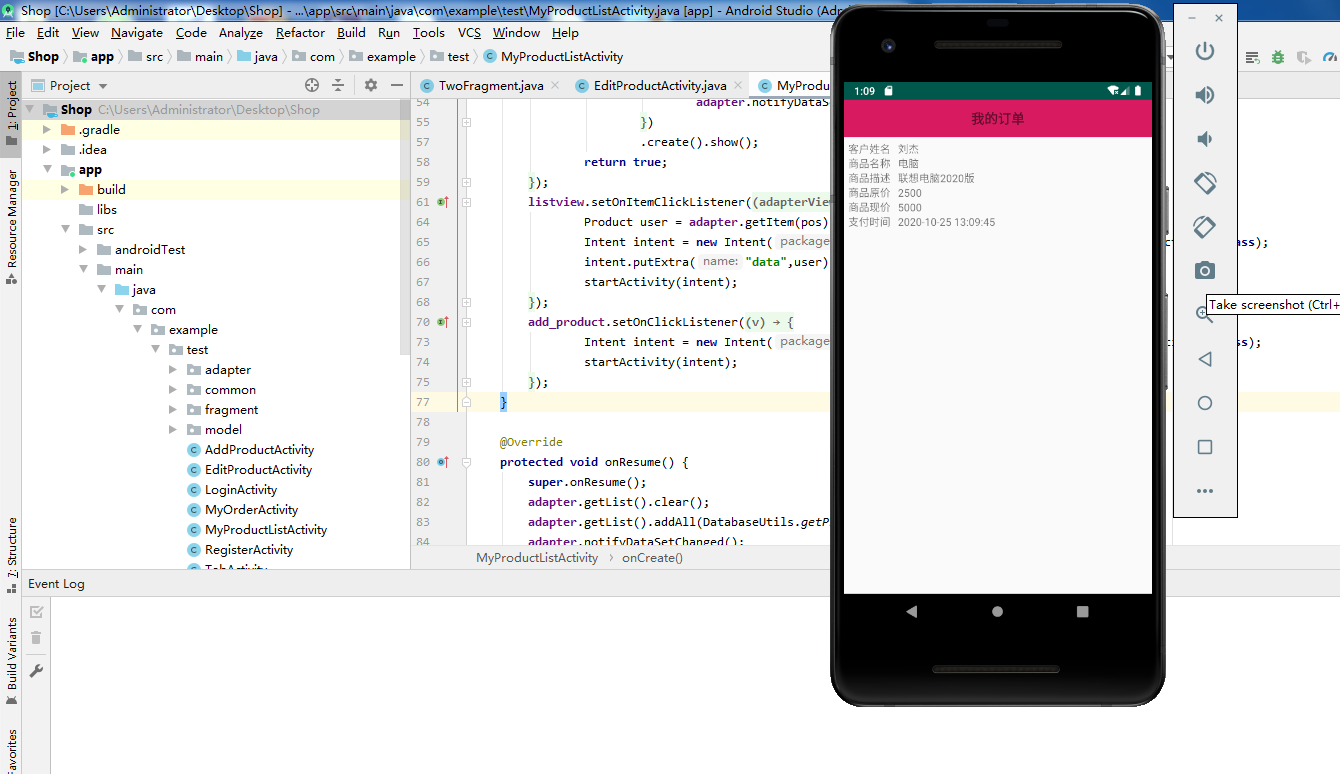


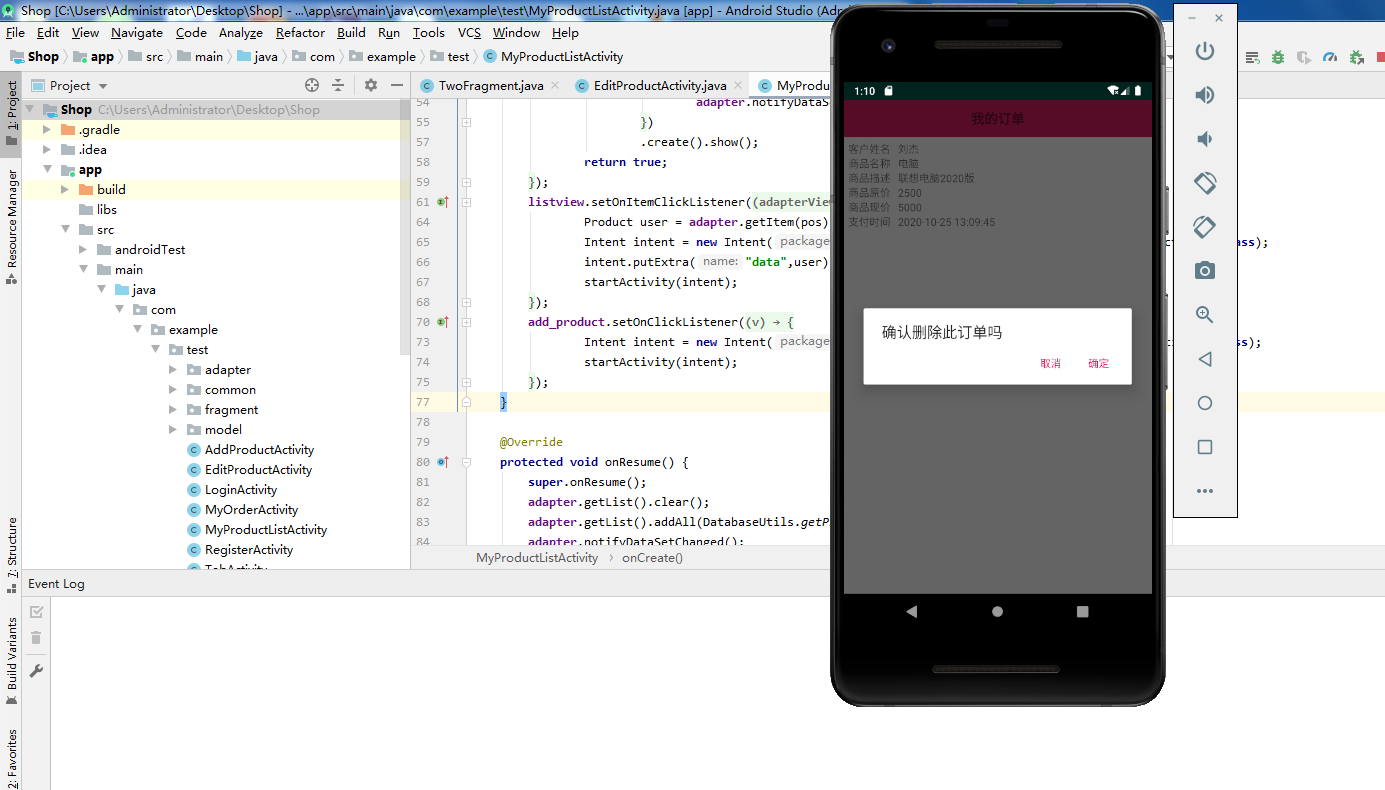


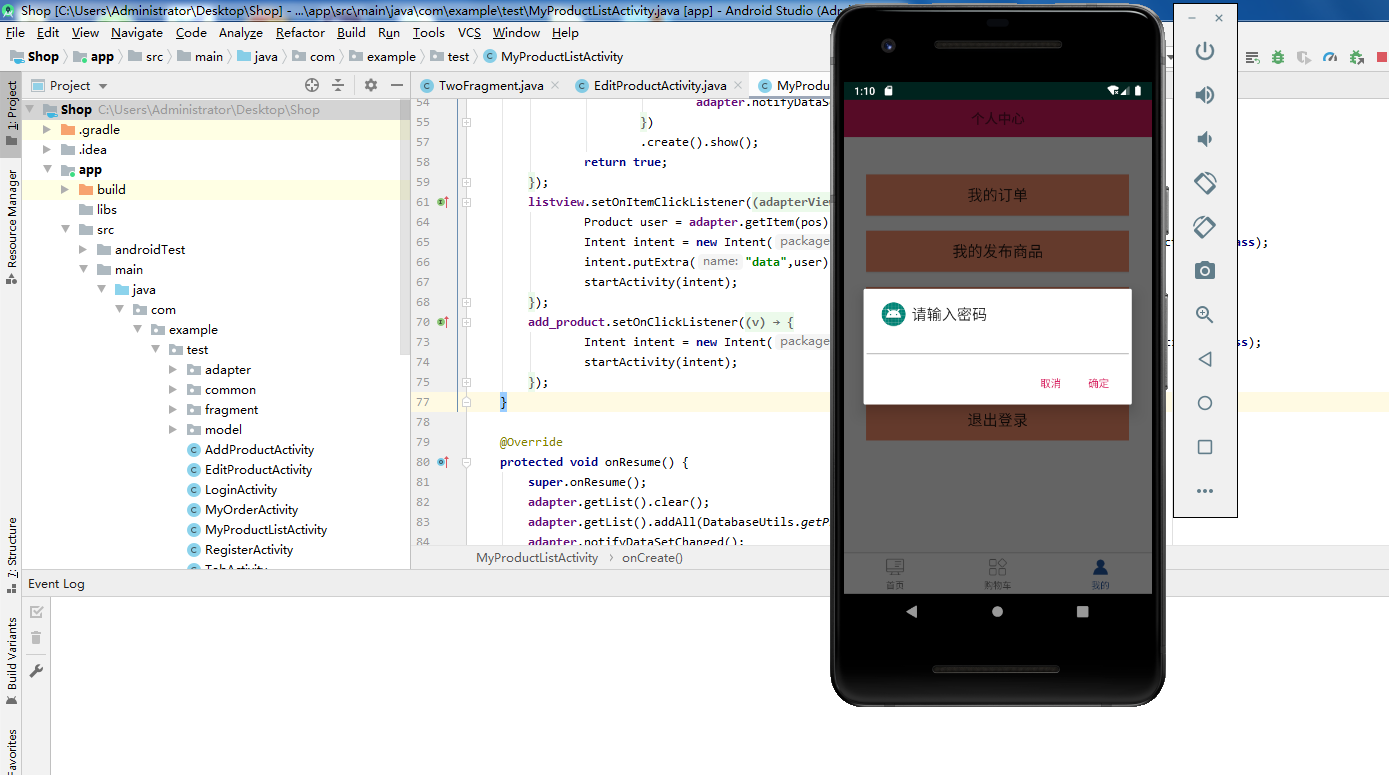




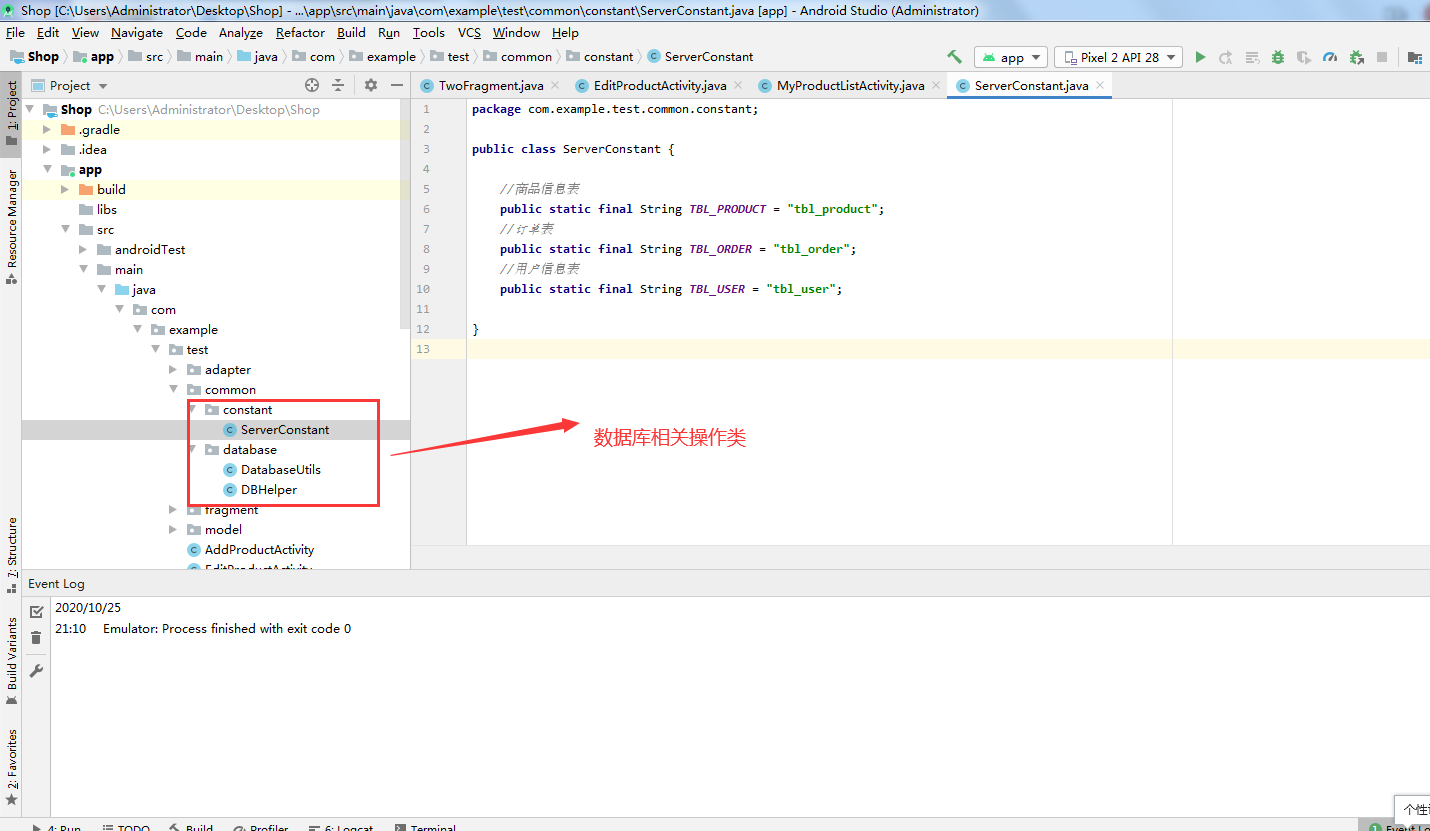


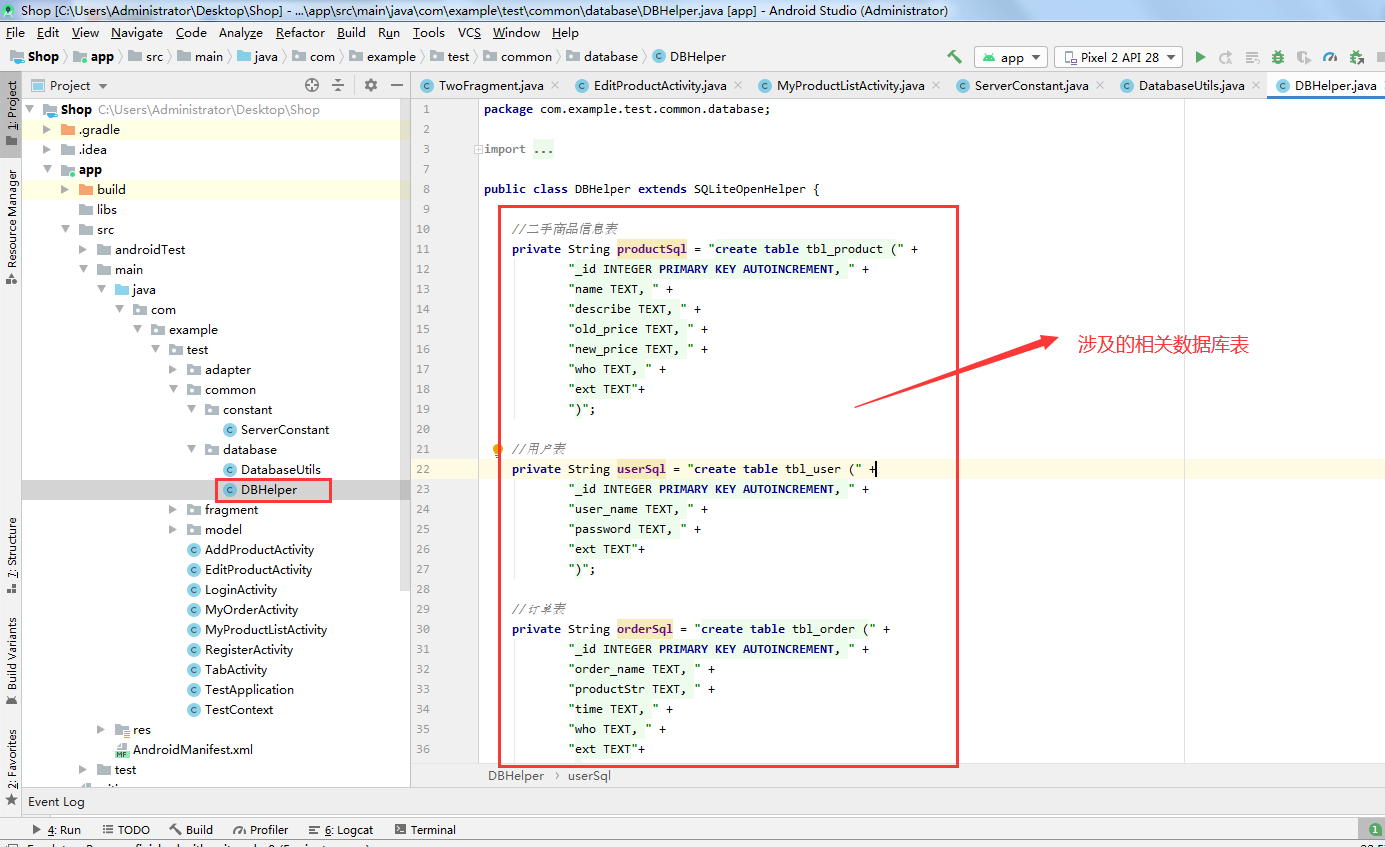


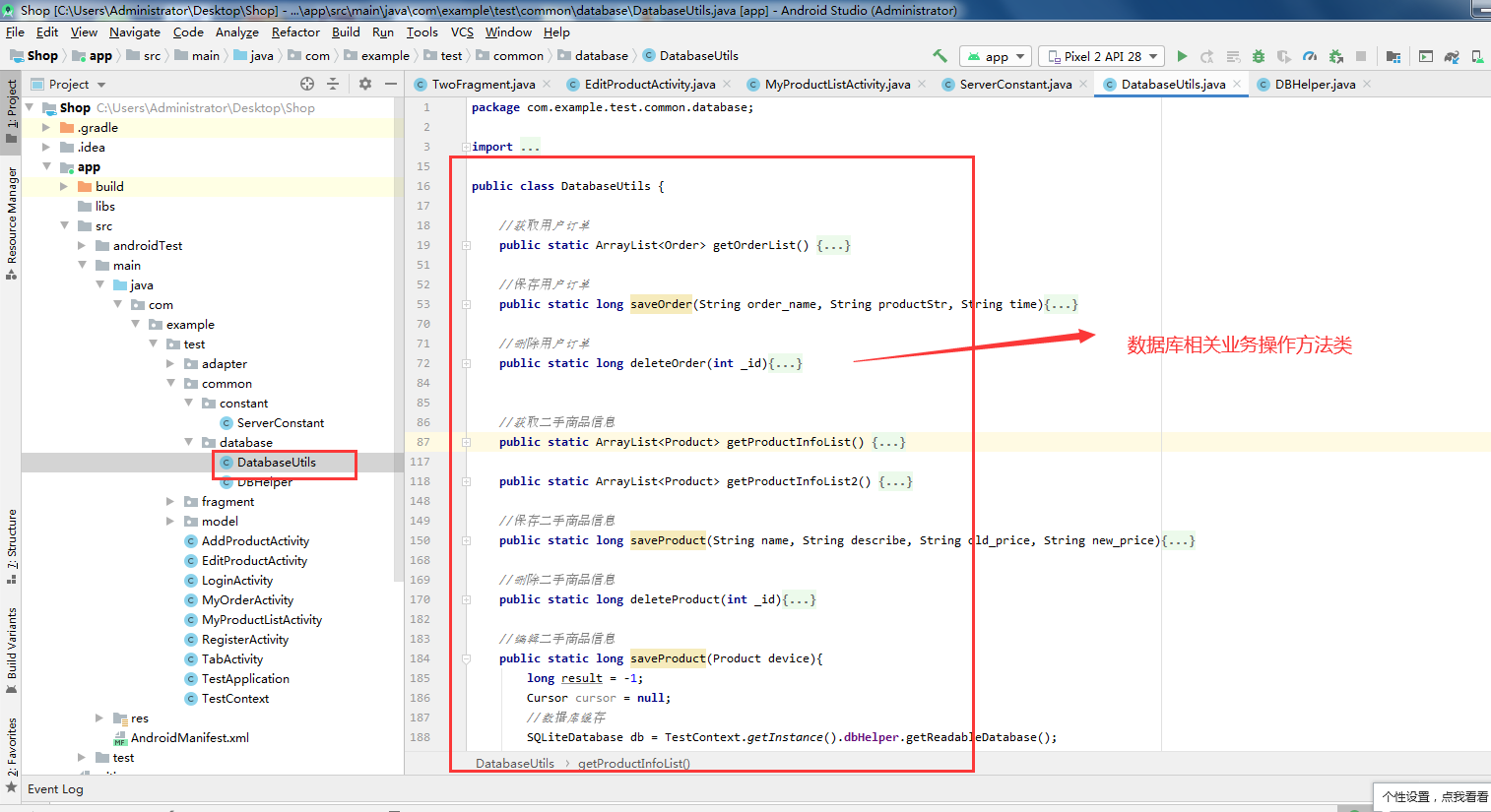


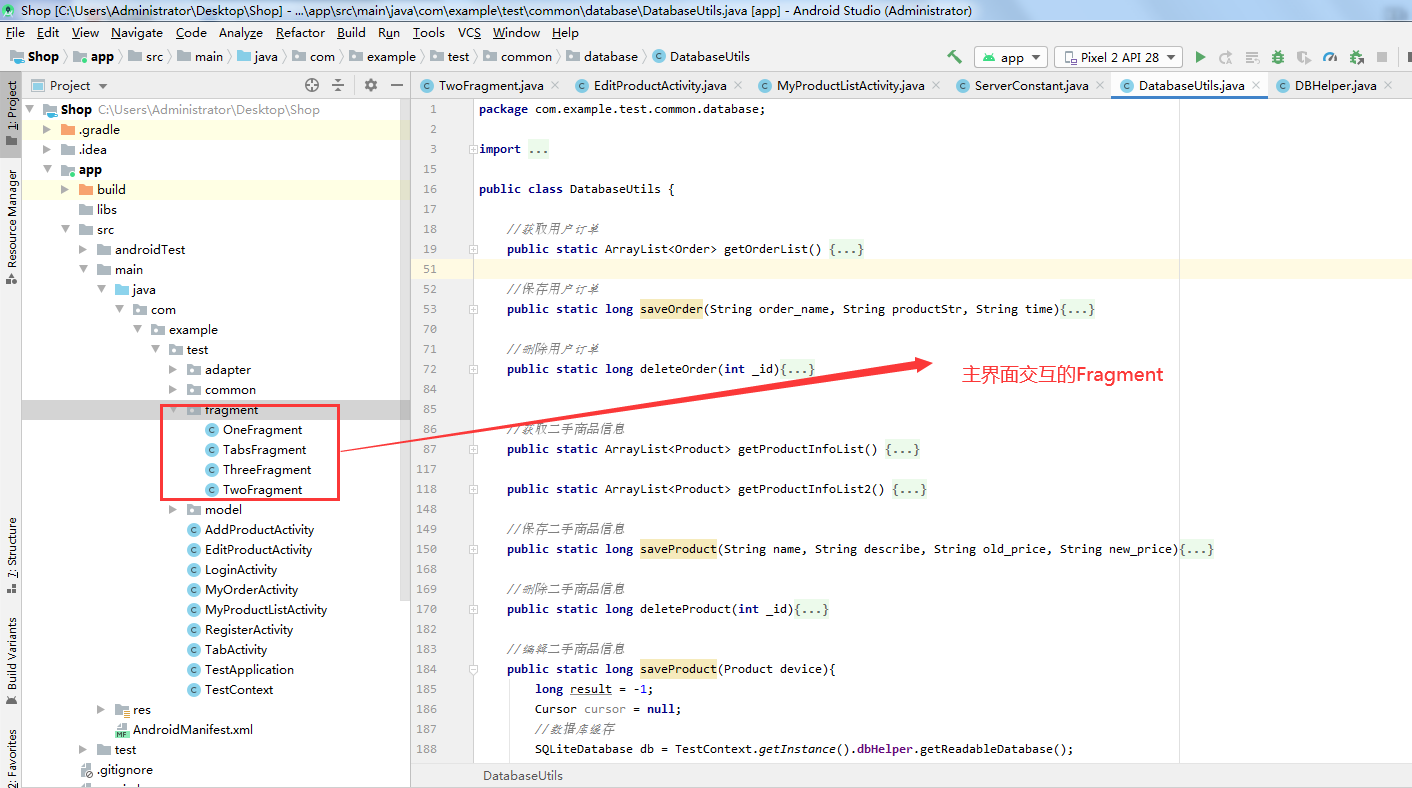


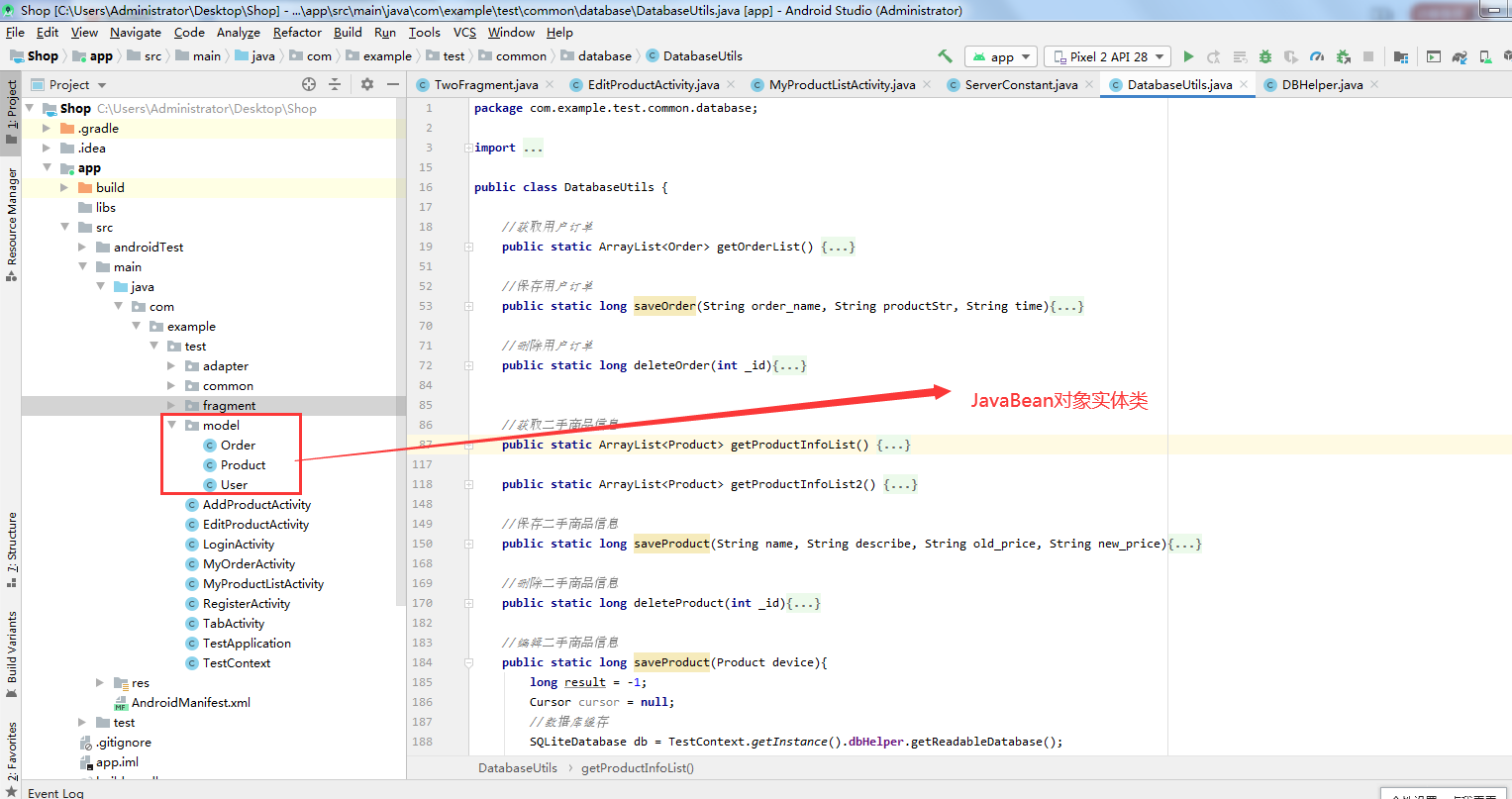
**Critical code analysis**











**Case summary**

In this project, we have carefully conceived and designed the system structure framework. The main Development tools are Android Studio and Java support, the plug-in of Android Development Tool, and the comprehensive combination of Android version. Only then can the whole second-hand market APP system be realized.