**XXXX学校**

**移动应用软件开发实训报告**

题 目： 移动应用软件开发

课 题 名：

系 别：

专业班级：

姓 名：

学 号：

成绩评定：

指导教师签字：

二〇二一 年 一 月 九日

**目录**

[一、实训目的 3](#_Toc25827874)

[二、实训内容 3](#_Toc25827875)

[三、实训步骤 3](#_Toc25827876)

[（一）需求分析 3](#_Toc25827877)

[（二）系统设计 3](#_Toc25827878)

[（三）系统实现 3](#_Toc25827879)

[（四）系统测试 4](#_Toc25827880)

[（五）…… 4](#_Toc25827881)

[四、总结及心得体会 5](#_Toc25827882)

# 一、实训目的

通过本课程实训，了解现阶段互联网发展主流，了解移动互联网，认识移动互联网的发展与展望，认识Android，了解基于Android的应用软件开发方法及其商业流程。把理论与实际结合，通过对理论知识的理解，领悟从而运用到生活实际巩固所学的知识，提高对实际生活的认识，积累经验。使学生在此期间能够初次体会到实际生产中的种种技能与经验，完成一项项目锻炼独立思考及团队合作能力。使学生们进一步加深对所学知识的理解，理论联系实际，巩固所学有关计算机基础理论知识和基本技能，学习有关计算机最新技术方面的应用，增强学生对计算机在社会生活，社会生产中应用的感性认识，深入了解计算机在各个领域中的应用状况。生产实习是学校教学的重要补充部分，是区别于普通学校教育的一个显著特征，是教育教学体系中的一个不可缺少的重要组成部分和不可替代的重要环节。它是与今后的职业生活最直接联系的，学生在生产实习过程中将完成学习到就业的过渡，因此生产实习是培养技能型人才，实现培养目标的主要途径。它不仅是校内教学的延续，而且是校内教学的总结。生产实习一方面巩固了书本上学到的理论知识，另一方面，可获得在书本上不易了解和不易学到的生产现场的实际知识，使我们在实践中得到提高实训环节对于提高学生的综合能力和全面素质具有重要意义。

# 二、实训内容

系统功能需求分析，总体设计

UI界面设计

Sqlite数据库

数据读取

增删查改设计

系统测试

# 三、实训步骤

## ****（一）需求分析****

设计与开发一个购物商城系统，对商品信息进行操作、主要功能如下：用户登录、用户注册、全部商家展示、商家商品展示、模拟讨论功能、商家信息展示、购物车功能，结算功能，账户充值功能，查看个人订单，个人信息修改，退出切换账号。

## ****（二）系统设计****

**一．系统总体功能结构图**

购物商城系统

退出切换账号功能

个人信息修改

查看个人订单

账户充值功能

结算功能

购物车功能

商家信息展示

模拟讨论功能

商家商品展示

全部商家展示

用户

注册

用户登录

**二．模块功能介绍**

1.用户登录

此模块主要的实现功能是用户登录，即当用户进入此系统时需要输入账号和密码，系统与数据库中的数据进行比对，正确则进入系统，不正确则进行提示，并可以实现记住密码和自动登录功能。

2. 用户注册

此模块主要实现新用户的注册功能，注册输入用户名和密码，系统与数据库中进行比对，无重复且信息正常，则录入数据库，返回登录界面。

3．全部商家展示

此模块主要的实现功能为用户的登录成功后可以看到可以上下滑动的商家信息列表。

4. 商家商品展示

此模块主要的实现功能为用户点击商家信息列表，可以进入对应商家界面，展示商家所出售商品信息。

5. 模拟讨论功能

此模块主要的实现功能为支持不同用户在讨论模块模拟发送评论，可以看到不同账户之间发送的信息，以模拟讨论的实现。

6. 商家信息展示

此模块主要可以点击商家标签，显示商家简介信息和评分等信息。

7. 购物车功能

此模块主要实现功能为可以实现用户点击商品加号后点击下面购物车栏弹出类似饿了么的类似购物车显示。

8. 结算功能

此模块可以点击购物车中的 去结算 按钮，跳转至结算界面模拟支付后即可加入个人订单。

9. 账户充值功能

此模块可以实现当前向当前用户账户中充值余额。

10. 查看订单功能

此模块可以实现点击 订单列表 后，可以查看当前账户的所有订单信息，并可以点击条目查看详细信息。

11. 个人信息修改

此模块可以实现修改当前用户的个人信息。

12. 退出切换账号功能

此模块可以实现当前用户退出系统并重新登录其他账号。

## ****（三）系统实现****

### 1. 用户登录



该功能主要是通过用户输入的用户名和密码，后台通过对应的方法从数据库中查询，如果查询到对应的用户信息，则进入主界面，否则将将错误提示信息通过Toast展示给用户，点击记住密码和自动登录，将在下次登陆时记住密码无需再次填写以及自动进行登录，此数据通过安卓studio的SharedPreferences进行相关数据的存储。

**用户登录实现核心代码**

//查询用户名密码是否正确  
public boolean queryUseristrue(String username, String password) {  
 Cursor cursor = db.rawQuery("select \* from userInfo where username = ? and password = ?", new String[]{username, password});  
 if (cursor.moveToFirst() == true) {  
 cursor.close();  
 return true;  
 }  
 return false;  
}  
  
//查询用户名是否存在 存在返回true 不存在false  
public boolean queryNameisExist(String username) {  
 Cursor cursor = db.rawQuery("select \* from userInfo where username = ? ", new String[]{username});  
 if (cursor.moveToFirst() == true) {  
 cursor.close();  
 return true;  
 }  
 return false;  
}

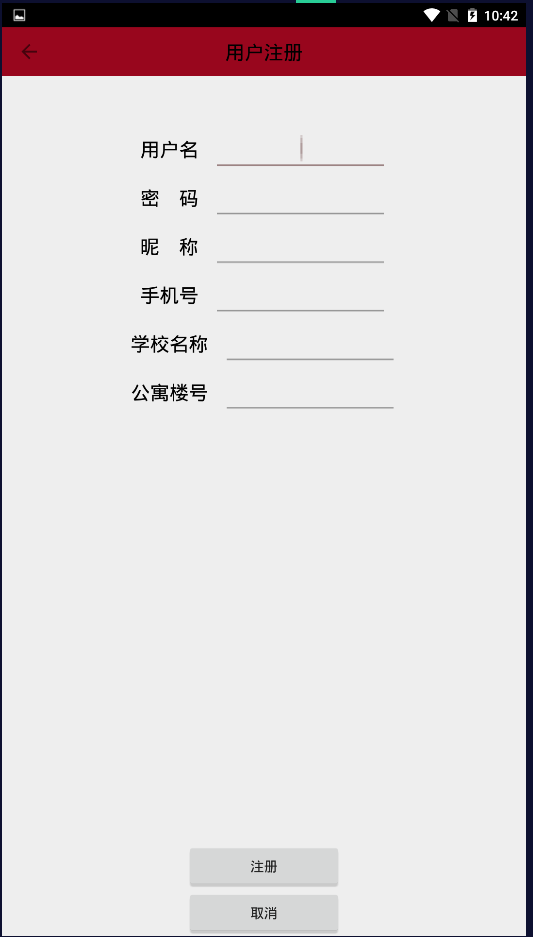
**登录点击后进行的代码操作**

Override  
public void onClick(View view) {  
 switch (view.getId()){  
 case R.id.*M\_DengLu*:  
 username = et\_username.getText().toString();  
 password = et\_password.getText().toString();  
 if (TextUtils.*isEmpty*(username) || TextUtils.*isEmpty*(password)) {  
 Toast.*makeText*(MainActivity.this, "用户名或密码不能为空", Toast.*LENGTH\_SHORT*).show();  
 return;  
  
 }  
 if (MySQLiteHelper.*getInstance*(this).queryUseristrue(username,password)){  
 if (rember.isChecked()){  
 ShareUtils.*putUserName*(username);  
 ShareUtils.*putPassword*(password);  
 }  
 Toast.*makeText*(MainActivity.this, "登陆成功", Toast.*LENGTH\_SHORT*).show();  
 startActivity(new Intent(this, MeunsActivity.class));  
 MainActivity.this.finish();  
 }else {  
 Toast.*makeText*(MainActivity.this, "登录失败,用户名或密码错误", Toast.*LENGTH\_SHORT*).show();  
 return;  
 }  
 break;

**自动登录和记住密码在SharedPreferences中的代码**

public class ShareUtils {  
 static SharedPreferences *preferences* = AppContext.*getInstance*().getSharedPreferences("data", Context.*MODE\_PRIVATE*);  
 static SharedPreferences.Editor *editor* = *preferences*.edit();  
  
 */\*\*  
 \* 从SharedPreferences保存的文件中获取是否自动登录  
 \* 1为自动登录 0 为否  
 \*/* public static String getAuto\_Login() {  
 return *preferences*.getString("AUTO\_LOGIN", "");  
 }  
  
 */\*\*  
 \* 从SharedPreferences保存的文件中获取是否自动登录  
 \* 1为自动登录 0 为否  
 \*/* public static void putAuto\_Login( String tag) {  
 *editor*.putString("AUTO\_LOGIN", tag);  
 *editor*.commit();  
 }  
  
 */\*\*  
 \* 从SharedPreferences保存的文件中获取是否记住密码  
 \* 1为记住密码 0 为否  
 \*/* public static String getRember() {  
 return *preferences*.getString("Rember", "");  
 }  
  
 */\*\*  
 \* 从SharedPreferences保存的文件中获取否记住密码  
 \* 1为记住密码 0 为否  
 \*/* public static void putRember( String isRember) {  
 *editor*.putString("Rember", isRember);  
 *editor*.commit();  
 }  
  
 */\*\*  
 \* 向SharedPreferences存储用户名  
 \*/* public static void putUserName(String username) {  
 *editor*.putString("UserName", username);  
 *editor*.commit();  
 }  
 */\*\*  
 \* 从SharedPreferences取出用户名  
 \*/* public static String getUserName() {  
 return *preferences*.getString("UserName", "");  
 }  
  
  
 */\*\*  
 \* 向SharedPreferences存储密码  
 \*/* public static void putPassword(String password) {  
 *editor*.putString("Password", password);  
 *editor*.commit();  
 }  
 */\*\*  
 \* 从SharedPreferences取出密码  
 \*/* public static String getPassword() {  
 return *preferences*.getString("Password", "");  
 }  
  
}

### 2. 用户注册



此功能为，当用户点击注册后跳转到此界面，输入用户名和密码，点击注册后，会自动进行相关的判断，包括判断用户名是否重复，两次输入的密码是否相同等，若有问题会通过Toast反馈给用户并重新运行相关判断代码，判断无误后，即通过数据库帮助类的相应方法将数据写入数据库并返回至登录界面

**用户注册界面核心代码实现**

**将数据存入sqlite数据库代码**

//将注册的用户名密码插入到表中  
public void insertUserinfo(UserBean userBean) {  
 db.execSQL("insert into tb\_user(username,password,phone,school,money) values(?,?,?,?,?)", new Object[]{userBean.getUsername(), userBean.getPassword(),userBean.getPhone(),userBean.getSchool(),userBean.getMoney()});  
 Log.*e*("插入语句:", "插入已执行，插入成功");  
}  
  
  
//查询用户名密码是否正确  
public boolean queryUserIsTrue(String username, String password) {  
 Cursor cursor = db.rawQuery("select \* from tb\_user where username = ? and password = ?", new String[]{username, password});  
 if (cursor.moveToFirst() == true) {  
 cursor.close();  
 return true;  
 }  
 return false;  
}  
  
//查询用户名是否存在 存在返回true 不存在false  
public boolean queryUserNameIsExist(String username) {  
 Cursor cursor = db.rawQuery("select \* from tb\_user where username = ? ", new String[]{username});  
 if (cursor.moveToFirst() == true) {  
 cursor.close();  
 return true;  
 }  
 return false;  
}

**通过控件获取用户输入信息并判断**

@Override  
public void onClick(View view) {  
 if (view.getId() == R.id.*Zhuce*){  
 username = et\_username.getText().toString();  
 psw1 = et\_psw1.getText().toString();  
 psw2 = et\_psw2.getText().toString();  
 if (MySQLiteHelper.*getInstance*(this).queryNameisExist(username)) {  
 Toast.*makeText*(this, "用户名已存在", Toast.*LENGTH\_SHORT*).show();  
 return;  
 }  
 if (TextUtils.*isEmpty*(username) || TextUtils.*isEmpty*(psw1) || TextUtils.*isEmpty*(psw2) ){  
 Toast.*makeText*(this, "用户名密码不能为空！", Toast.*LENGTH\_SHORT*).show();  
 return;  
 }  
 if (psw1.equals(psw2)){  
 final Userinfo user = new Userinfo();  
 user.setUsername(username);  
 user.setPassword(psw1);  
 dialog = new AlertDialog.Builder(this).setTitle("是否确认注册？")  
 .setMessage("用户名："+username +'\n' + "密 码：" + psw1)  
 .setIcon(R.mipmap.*ic\_launcher*)  
 .setPositiveButton("确定", new DialogInterface.OnClickListener() {  
 @Override  
 public void onClick(DialogInterface dialogInterface, int i) {  
 GoAn(user);  
 }  
 })  
 .setNegativeButton("取消",null)  
 .create();  
 dialog.show();  
  
 }else {  
 Toast.*makeText*(this,"两次密码不一致",Toast.*LENGTH\_SHORT*).show();  
 return;  
 }  
 }else if (view.getId() == R.id.*Quxiao*){  
 Intent intent = new Intent(this, MainActivity.class);  
 RegisterActivity.this.finish();  
 startActivity(intent);  
 }  
}  
  
//注册成功执行方法，将对象存入sqllite  
public void GoAn(Userinfo userinfo){  
 MySQLiteHelper.*getInstance*(this).insertUserinfo(userinfo);  
 Toast.*makeText*(this,"注册成功",Toast.*LENGTH\_LONG*).show();  
 newUserName = userinfo.getUsername();  
 newPassword = userinfo.getPassword();  
 Intent intent = new Intent(this,MainActivity.class);  
 intent.putExtra("NUN",newUserName);  
 intent.putExtra("NP",newPassword);  
 startActivity(intent);  
 RegisterActivity.this.finish();  
}

### 3. 全部商家展示



此功能为，用户登录成功后进入此页面，从数据库中查询商家信息，通过适配器展示到recycler view上。

**核心代码：**

/\*初始化View\*/  
private void ininView() {  
 drawerLayout = findViewById(R.id.*drawer\_layout*);  
 navigationView = findViewById(R.id.*navigation\_view*);  
 toolbar = findViewById(R.id.*toolbar*);  
 tv\_bar\_title = findViewById(R.id.*tv\_bar\_title*);  
 tv\_bar\_title.setText("商家列表");  
  
 //获取头布局  
 headerView = (LinearLayout) navigationView.getHeaderView(0);  
 tv\_main\_userName = headerView.findViewById(R.id.*tv\_header\_userName*);  
 iv\_head = headerView.findViewById(R.id.*iv\_head*);  
 tv\_main\_userName.setText(MainActivity.*username*);  
 UserID = MySQLiteHelper.*getInstance*(getApplicationContext()).GetUserId(MainActivity.*username*);  
 switch (UserID){  
 case 1:  
 iv\_head.setImageResource(R.drawable.*tx\_1\_48*);  
 break;  
 case 2:  
 iv\_head.setImageResource(R.drawable.*tx\_2\_48*);  
 break;  
 case 3:  
 iv\_head.setImageResource(R.drawable.*tx\_3\_48*);  
 break;  
 case 4:  
 iv\_head.setImageResource(R.drawable.*tx\_4\_48*);  
 break;  
 case 5:  
 iv\_head.setImageResource(R.drawable.*tx\_5\_48*);  
 break;  
 case 6:  
 iv\_head.setImageResource(R.drawable.*tx\_6\_48*);  
 break;  
 }  
  
  
 rv\_stores = findViewById(R.id.*rv\_stores*);  
 allStoresAdapter = new AllStoresAdapter(storeBeans);  
 //初始化recyclerView  
 rv\_stores.setItemAnimator(new DefaultItemAnimator());  
 rv\_stores.setLayoutManager(new LinearLayoutManager(this));  
 rv\_stores.setAdapter(allStoresAdapter);  
  
 //设置item点击事件  
 allStoresAdapter.setOnItemClickListener(new AllStoresAdapter.OnItemClickListener() {  
 @Override  
 public void onClick(View v, int position, StoreBean storeBean) {  
 //ToastUtil.showShort("您点击的下标"+position+"商家："+storeBean.getStoreName());  
 startActivity(new Intent(HomeAllStoresActivity.this,StoreGoodsActivity.class).putExtra("storeID",storeBean.getID()));  
 }  
 });  
  
}

### 4. 商家商品展示



此功能为，点击商品家条目后进入此页面，点餐模块将数据库中的对应商家商品信息读取并展示到列表中，左侧为分类模块，实现标签与列表的联动效果。

**核心代码：**

tv\_head = getActivity().findViewById(R.id.*tv\_header*);  
 tv\_head.setText(*goodsArrayBean*.itemsLeft.get(0).getTitle());  
 re\_shoppingCar = getActivity().findViewById(R.id.*re\_shoppingCar*);  
 *iv\_car* = getActivity().findViewById(R.id.*iv\_shoppingCar*);  
 *tv\_car* = getActivity().findViewById(R.id.*tv\_shoppingCar*);  
 *tv\_total* = getActivity().findViewById(R.id.*tv\_total\_frag*);  
  
  
 rvL = getActivity().findViewById(R.id.*rv1*);  
 rvR = getActivity().findViewById(R.id.*rv2*);  
 rvR.setItemAnimator(null);  
 rvL.setLayoutManager(new LinearLayoutManager(getContext()));  
 rvR.setLayoutManager(new LinearLayoutManager(getContext()));  
 lAdapter = new LAdapter(getContext(),R.layout.*goods\_left\_item*, *goodsArrayBean*.itemsLeft);  
 lAdapter.bindToRecyclerView(rvL);  
 rvL.setAdapter(lAdapter);  
 *rAdapter* = new RAdapter(getContext(),R.layout.*goods\_right\_item*, *goodsArrayBean*.itemsRight);  
 rvR.setAdapter(*rAdapter*);  
  
  
 re\_shoppingCar.setOnClickListener(new View.OnClickListener() {  
 @Override  
 public void onClick(View view) {  
 dialogData = *rAdapter*.getCurrentData();  
 if (dialogData.size() != 0){  
 dialog = new MyDialog(getActivity(),dialogData);  
 dialog.show();  
 }  
 }  
 });  
}  
  
private boolean moveToTop = false;  
private int index;  
private void initListener() {  
 lAdapter.setOnItemClickListener(new BaseRecyclerViewAdater.OnItemClickListener() {  
 @Override  
 public void onItemClick(View view, int position) {  
 if (rvR.getScrollState() != RecyclerView.*SCROLL\_STATE\_IDLE*)return;  
 lAdapter.fromClick = true;  
 lAdapter.setChecked(position);  
 String tag = lAdapter.getmData().get(position).getTitle();  
 for (int i = 0; i < *rAdapter*.getmData().size(); i++) {  
 //根据左边选中的条目获取到右面此条目Title相同的位置索引；  
 if (TextUtils.*equals*(tag,*rAdapter*.getmData().get(i).getTitle())){  
 index = i;  
 moveToPosition\_R(index);  
 return;  
 }  
 }  
 }  
 });  
  
 rvR.addOnScrollListener(new RecyclerView.OnScrollListener() {  
 @Override  
 public void onScrolled(@NonNull RecyclerView recyclerView, int dx, int dy) {  
 super.onScrolled(recyclerView, dx, dy);  
 LinearLayoutManager layoutManager = (LinearLayoutManager) rvR.getLayoutManager();  
 if (moveToTop){ //向下滑动时，只会把改条目显示出来；我们还需要让该条目滑动到顶部；  
 moveToTop = false;  
 int m = index - layoutManager.findFirstVisibleItemPosition();  
 if (m >= 0 && m <= layoutManager.getChildCount()){  
 int top = layoutManager.getChildAt(m).getTop();  
 rvR.smoothScrollBy(0,top);  
 }  
 }else {  
 int index = layoutManager.findFirstVisibleItemPosition();  
 tv\_head.setText(*rAdapter*.getmData().get(index).getTitle());  
 lAdapter.setToPosition(*rAdapter*.getmData().get(index).getTitle());  
 }  
 }  
 });  
  
 rvR.setOnTouchListener(new View.OnTouchListener() {  
 @Override  
 public boolean onTouch(View view, MotionEvent motionEvent) {  
 lAdapter.fromClick = false;  
 return false;  
 }  
 });  
}  
  
  
private void moveToPosition\_R(int index) {  
 LinearLayoutManager layoutManager = (LinearLayoutManager) rvR.getLayoutManager();  
 int f = layoutManager.findFirstVisibleItemPosition();  
 int l = layoutManager.findLastVisibleItemPosition();  
 if (index <= f){ //向上移动时  
 layoutManager.scrollToPosition(index);  
 }else if (index <= l){ //已经再屏幕上面显示时  
 int m = index - f;  
 if (0 <= m && m <= layoutManager.getChildCount()) {  
 int top = layoutManager.getChildAt(m).getTop();  
 rvR.smoothScrollBy(0, top);  
 }  
 }else { //向下移动时  
 moveToTop = true;  
 layoutManager.scrollToPosition(index);  
 }  
}

### 5. 模拟讨论功能



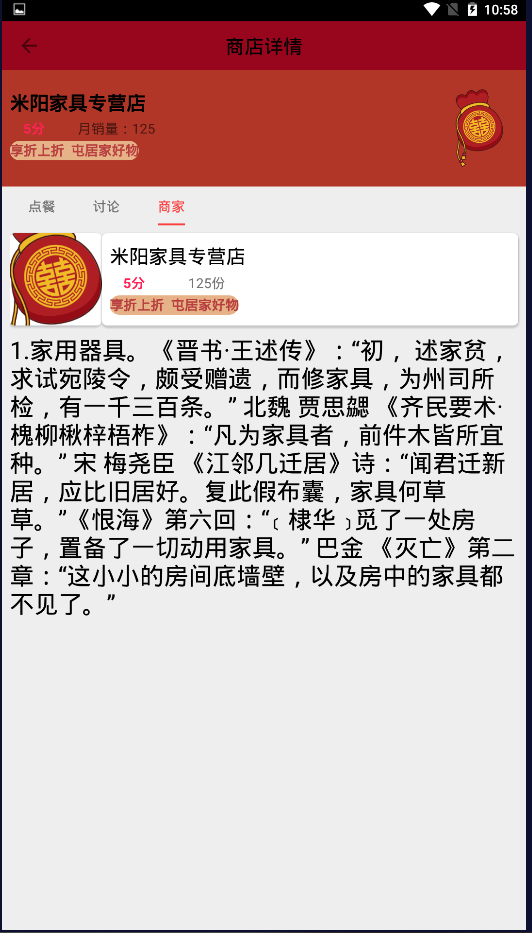
此功能为，实现了模拟不同用户之间的讨论功能，将发送的讨论信息储存到对应商家的sqlite数据库表中。上方的消息列表根据发送者不同id显示不同的左右位置，以供用户识别。

**核心代码：**

private void initView() {  
 rv\_Chat = getActivity().findViewById(R.id.*rv\_chat*);  
 et\_chat\_message = getActivity().findViewById(R.id.*et\_chat\_message*);  
 btn\_message\_send = getActivity().findViewById(R.id.*btn\_message\_send*);  
 messages = et\_chat\_message.getText().toString();  
  
  
 adapter = new ChatMessageAdapter(chatMessageBeans);  
 rv\_Chat.setItemAnimator(new DefaultItemAnimator());  
 rv\_Chat.setLayoutManager(new LinearLayoutManager(getActivity()));  
 rv\_Chat.setAdapter(adapter);  
 rv\_Chat.scrollToPosition(adapter.getItemCount()-1);  
  
 //发送信息的判断和操作  
 btn\_message\_send.setOnClickListener(new View.OnClickListener() {  
 @Override  
 public void onClick(View view) {  
 messages = et\_chat\_message.getText().toString();  
 if (TextUtils.*isEmpty*(messages)){  
 ToastUtil.*showShort*("内容为空！");  
 }else {  
 chatMessageBean = new ChatMessageBean(MyUserID,MyUserID+"",messages,MainActivity.*username*, DateUtill.*getCurrentTime*());  
 MySQLiteHelper.*getInstance*(getActivity()).insertMessages(chatMessageBean);  
 adapter.refreshMessages();  
 rv\_Chat.scrollToPosition(adapter.getItemCount()-1);  
 ToastUtil.*showShort*("发送成功");  
 et\_chat\_message.setText("");  
 }  
 }  
 });  
}

//插入聊天信息的方法  
public void insertMessages(ChatMessageBean chatMessageBean){  
 ContentValues cv =new ContentValues();  
 cv.put("ID",chatMessageBean.getId());  
 cv.put("img\_id",chatMessageBean.getImg\_id());  
 cv.put("message",chatMessageBean.getMessage());  
 cv.put("userName",chatMessageBean.getUserName());  
 cv.put("time",chatMessageBean.getTime());  
 //插入数据，参数为表名，当列为空时的填充值，封装数据的ContentValue  
 db.insert("messagesInfo",null,cv);  
}

### 6. 商家信息展示



此功能为，点击或侧滑到商家模块，可以显示数据库中存储的相关商家简介信息和评分销量等信息。

**核心代码：**

private void initView() {  
 iv\_store\_pic = getActivity().findViewById(R.id.*iv\_store\_frag*);  
 tv\_storeName = getActivity().findViewById(R.id.*tv\_storeName\_frag*);  
 tv\_storeScore = getActivity().findViewById(R.id.*tv\_store\_score\_frag*);  
 tv\_storeSell = getActivity().findViewById(R.id.*tv\_store\_sell\_frag*);  
 tv\_storeSign = getActivity().findViewById(R.id.*tv\_store\_sign\_frag*);  
 tv\_storeIntro = getActivity().findViewById(R.id.*tv\_store\_intro*);  
  
  
 switch (storeBean.getIv\_store\_pic()){  
 case "0":  
 iv\_store\_pic.setImageResource(R.mipmap.*store\_1*);  
 break;  
 case "1":  
 iv\_store\_pic.setImageResource(R.mipmap.*store\_2*);  
 break;  
 case "2":  
 iv\_store\_pic.setImageResource(R.mipmap.*store\_3*);  
 break;  
 case "3":  
 iv\_store\_pic.setImageResource(R.mipmap.*store\_4*);  
 break;  
 case "4":  
 iv\_store\_pic.setImageResource(R.mipmap.*store\_5*);  
 break;  
 case "5":  
 iv\_store\_pic.setImageResource(R.mipmap.*store\_6*);  
 break;  
 case "6":  
 iv\_store\_pic.setImageResource(R.mipmap.*store\_7*);  
 break;  
 case "7":  
 iv\_store\_pic.setImageResource(R.mipmap.*store\_8*);  
 break;  
 case "8":  
 iv\_store\_pic.setImageResource(R.mipmap.*store\_9*);  
 break;  
 case "9":  
 iv\_store\_pic.setImageResource(R.mipmap.*store\_10*);  
 break;  
 }  
 tv\_storeName.setText(storeBean.getStoreName());  
 tv\_storeScore.setText(storeBean.getStoreScore()+"分");  
 tv\_storeSell.setText(storeBean.getStoreSell()+"份");  
 tv\_storeSign.setText(storeBean.getStoreSign());  
 tv\_storeIntro.setText(storeBean.getStoreIntro());  
  
}

### 7. 购物车功能

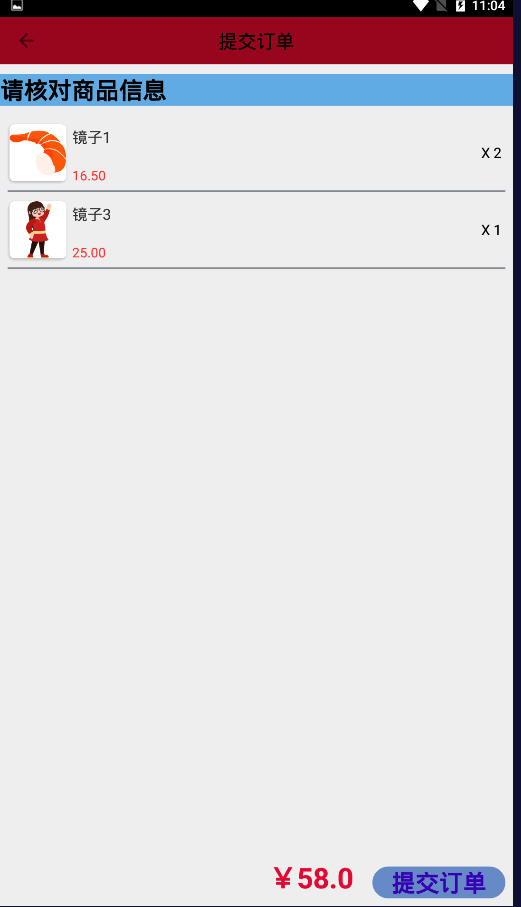


此功能为用户可以点击商品条目的加号并选择数量后，点击下方购物车标签，可以弹出仿饿了么式的购物车功能，可以在购物车中加减商品以及清空购物车。

**核心代码：**

// 设置dialog的位置  
 Window dialogWindow = *dialog*.getWindow();  
 dialogWindow.getDecorView().setPadding(0, 0, 0, 0);  
 dialogWindow.setBackgroundDrawableResource(android.R.color.*transparent*);//背景透明，不然会有个白色的东东  
// dialogWindow.setWindowAnimations(R.style.dialogWindowAnim); //不使用窗口弹出动画  
 WindowManager.LayoutParams lp = dialogWindow.getAttributes();  
  
 lp.width = WindowManager.LayoutParams.*MATCH\_PARENT*; // 宽度  
 lp.height = WindowManager.LayoutParams.*MATCH\_PARENT*; // 高度  
 dialogWindow.setAttributes(lp);  
 // 设置dialog为底部  
 dialogWindow.setGravity(Gravity.*BOTTOM*);  
 }  
  
  
  
  
 public void show() {  
 *dialog*.show();  
 showAnim();  
 }  
  
 // 出现动画  
 private void showAnim() {  
 TranslateAnimation animation = new TranslateAnimation(0, 0, DensityUtil.*dp2px*(context, 300), 0);  
 animation.setDuration(300);  
 animView.startAnimation(animation);  
 }  
  
 // 消失动画  
 private void dismissAnim() {  
 TranslateAnimation animation = new TranslateAnimation(0, 0, 0, DensityUtil.*dp2px*(context, 300));  
 animation.setAnimationListener(new Animation.AnimationListener() {  
 @Override  
 public void onAnimationStart(Animation animation) {  
  
 }  
  
 @Override  
 public void onAnimationEnd(Animation animation) {  
 *dialog*.myDismiss();  
 }  
  
 @Override  
 public void onAnimationRepeat(Animation animation) {  
  
 }  
 });  
 animation.setDuration(300);  
 animView.startAnimation(animation);  
 }

### 8. 结算功能

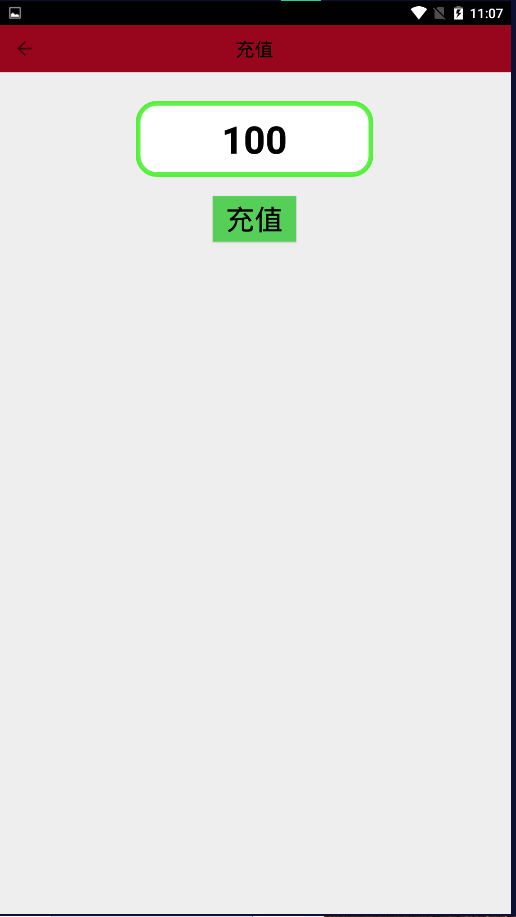


此功能为，用户点击购物车中的去结算按钮后跳转至此界面，界面显示购买的商品信息以及总价，点击提交订单后即可模拟支付效果。

**核心代码：**

private void initView() {  
 rv\_pay = findViewById(R.id.*rv\_pay*);  
 tv\_pay\_total = findViewById(R.id.*tv\_pay\_total*);  
 tv\_submitOrder = findViewById(R.id.*tv\_submitOrder*);  
 toolbar = findViewById(R.id.*toolbar*);  
 tv\_bar\_title = findViewById(R.id.*tv\_bar\_title*);  
 tv\_bar\_title.setText("提交订单");  
  
 //初始化recyclerView  
 rv\_pay.setItemAnimator(null);  
 payRVAdapter = new PayRVAdapter(data);  
 rv\_pay.setLayoutManager(new LinearLayoutManager(AppContext.*getInstance*()));  
 rv\_pay.setAdapter(payRVAdapter);  
  
  
 */\*\*  
 \* 循环获得总价格  
 \*/* for (int i= 0;i<data.size();i++){  
 b1 = new BigDecimal(data.get(i).getPrice().trim());  
 b2 = new BigDecimal(data.get(i).getNumber());  
 b3 = new BigDecimal(total);  
 result = b1.multiply(b2);  
 result = result.add(b3);  
 one = new BigDecimal("1");  
 a = result.divide(one,2,BigDecimal.*ROUND\_HALF\_UP*).doubleValue();//保留2位数  
 total = a ;  
 //Log.e("total",total+"");  
 }  
  
 tv\_pay\_total.setText("￥"+ total);  
  
 tv\_submitOrder.setOnClickListener(new View.OnClickListener() {  
 @Override  
 public void onClick(View view) {  
 //ToastUtil.showShort("总金额"+total);  
 dialog = new AlertDialog.Builder(PayActivity.this).setTitle("是否确认购买？")  
 .setMessage("总金额：" + total)  
 .setIcon(R.mipmap.*ic\_launcher*)  
 .setPositiveButton("确定", new DialogInterface.OnClickListener() {  
 @Override  
 public void onClick(DialogInterface dialogInterface, int i) {  
 doInsertOrder();  
 }  
 })  
 .setNegativeButton("取消", null)  
 .create();  
 dialog.show();  
  
  
 }  
 });  
  
}  
  
  
*/\*\*  
 \* 进行插入订单的判定和操作  
 \*/*private void doInsertOrder(){  
 accountMoney = MySQLiteHelper.*getInstance*(PayActivity.this).getUserMoneyFromUserName(MainActivity.*username*);  
 //判断账户余额是否足够  
 if (total > accountMoney){  
 ToastUtil.*showShort*("账户余额不足，请先充值");  
 dialog.dismiss();  
 }else {  
 //解决double精度丢失问题  
 a1 = new BigDecimal(accountMoney);  
 a2 = new BigDecimal(total);  
 result1 = a1.subtract(a2);  
 one = new BigDecimal("1");  
 resultMoney = result1.divide(one,2,BigDecimal.*ROUND\_HALF\_UP*).doubleValue();//保留2位数  
  
 MySQLiteHelper.*getInstance*(PayActivity.this).RechargeMoney(MainActivity.*username*,resultMoney);  
 gson = new Gson();  
 goodsJson = gson.toJson(data);  
 //Log.e("inputString=" , inputString);  
 orderBean = new OrderBean(MainActivity.*username*, DateUtill.*getCurrentTime*(),goodsJson);  
 //Log.e("order",orderBean.toString());  
 //Log.e("时间",DateUtill.getCurrentTime());  
 MySQLiteHelper.*getInstance*(PayActivity.this).insertOrderInfo(orderBean);  
 ToastUtil.*showShort*("下单成功!");  
 //通过handler通知刷新购物车界面  
 MyDialog.*handler*.sendEmptyMessage(2);  
 PayActivity.this.finish();  
 }  
  
}

### 9 账户充值功能



此功能为用户点击主页面“我的钱包”按钮后，点击充值按钮进入此页面，当用户输入充值金额时，点击充值按钮，会对输入的信息进行一些判断，如充值账户是或否存在，充值的金额不能为0等，之后会将充值记录存储起来。

**充值界面核心代码：**

//获取当前账户的金额  
 currentAccountMoney = MySqlLiteHelper.*getInstance*(getApplicationContext()).getUserMoneyFromUserName(SharedUtil.*getUserName2*());  
  
 tv\_back = findViewById(R.id.*tv\_back*);  
 tv\_back.setVisibility(View.*VISIBLE*);  
 tv\_bar\_title = findViewById(R.id.*tv\_bar\_title*);  
 tv\_bar\_title.setText("充值");  
 tv\_rechargeHistory = findViewById(R.id.*tv\_function*);  
 tv\_rechargeHistory.setVisibility(View.*VISIBLE*);  
 tv\_rechargeHistory.setText("充值记录 ");  
 et\_rechargeMoney= findViewById(R.id.*et\_rechargeMoney*);  
 btn\_doRecharge = findViewById(R.id.*btn\_doRecharge*);  
  
 tv\_back.setOnClickListener(new View.OnClickListener() {  
 @Override  
 public void onClick(View v) {  
 RechargeActivity.this.finish();  
 }  
 });  
 //充值到账户的点击事件  
 btn\_doRecharge.setOnClickListener(new View.OnClickListener() {  
 @Override  
 public void onClick(View view) {  
 money1 =et\_rechargeMoney.getText().toString();  
 if (money1.equals("0")){  
 ToastUtils.*showShort*("金额不能为0");  
 }else if (TextUtils.*isEmpty*(money1)) {  
 ToastUtils.*showShort*("金额不能为空");  
 }else {  
 double m1 = Double.*parseDouble*(money1);  
 currentAccountMoney = currentAccountMoney + m1;  
 dialog1 = new AlertDialog.Builder(RechargeActivity.this).setTitle("是否确认充值")  
 .setMessage("充值账户：" + SharedUtil.*getUserName2*() + '\n' + "金 额：" + money1 +"元")  
 .setIcon(R.mipmap.*ic\_launcher*)  
 .setPositiveButton("确定", new DialogInterface.OnClickListener() {  
 @Override  
 public void onClick(DialogInterface dialogInterface, int i) {  
 MySqlLiteHelper.*getInstance*(getBaseContext()).RechargeMoney(SharedUtil.*getUserName2*(), currentAccountMoney);  
 ToastUtils.*showShort*("充值成功");  
 et\_rechargeMoney.setText("");  
 rechargeHistoryBean = new RechargeHistoryBean();  
 rechargeHistoryBean.setUserName(SharedUtil.*getUserName2*());  
 rechargeHistoryBean.setMoney(Double.*parseDouble*(money1));  
 rechargeHistoryBean.setTime(DateUtill.*getCurrentTime*());  
 MySqlLiteHelper.*getInstance*(RechargeActivity.this).insertRechargeHistory(rechargeHistoryBean);  
 }  
 })  
 .setNegativeButton("取消", null)  
 .create();  
 dialog1.show();  
 }  
 }  
 });  
  
  
 tv\_rechargeHistory.setOnClickListener(new View.OnClickListener() {  
 @Override  
 public void onClick(View view) {  
 startActivity(new Intent(RechargeActivity.this, RechargeHistoryActivity.class));  
 }  
 });  
}

### 10 查看订单功能



此功能为，用户点击主页面“订单列表”后进入此页面，可以显示用户所有订单缩略信息，点击条目后可以跳转至订单详细信息界面

**核心代码：**

*/\*\*  
 \* 初始化数据  
 \*/*private void initData() {  
 orderBeans = MySQLiteHelper.*getInstance*(OrderActivity.this).queryOrderBeanFromUserName(MainActivity.*username*);  
}  
  
  
  
private void initView() {  
 toolbar = findViewById(R.id.*toolbar*);  
 tv\_bar\_title = findViewById(R.id.*tv\_bar\_title*);  
 tv\_bar\_title.setText("订单信息");  
  
 tv\_order = findViewById(R.id.*rv\_order*);  
 orderAdapter = new OrderAdapter(orderBeans);  
 tv\_order.setItemAnimator(new DefaultItemAnimator());  
 tv\_order.setLayoutManager(new LinearLayoutManager(this));  
 tv\_order.setAdapter(orderAdapter);  
}

private void initView() {  
 toolbar = findViewById(R.id.*toolbar*);  
 tv\_bar\_title = findViewById(R.id.*tv\_bar\_title*);  
 tv\_bar\_title.setText("订单详情");  
 tv\_total = findViewById(R.id.*tv\_order\_total*);  
 tv\_orderDetails\_time = findViewById(R.id.*tv\_orderDetails\_time*);  
 tv\_orderDetails\_time.setText(orderBean.getTime());  
 rv\_orderDetails = findViewById(R.id.*rv\_orderDetails*);  
 //初始化recyclerView  
 payRVAdapter = new PayRVAdapter(goodsData);  
 rv\_orderDetails.setLayoutManager(new LinearLayoutManager(AppContext.*getInstance*()));  
 rv\_orderDetails.setAdapter(payRVAdapter);  
  
  
 */\*\*  
 \* 循环数据集，获取价格总数  
 \*/* for (int i= 0;i<goodsData.size();i++){  
 //解决double类型计算丢失精度问题  
 b1 = new BigDecimal(goodsData.get(i).getPrice().trim());  
 b2 = new BigDecimal(goodsData.get(i).getNumber());  
 b3 = new BigDecimal(total);  
 result = b1.multiply(b2);  
 result = result.add(b3);  
 one = new BigDecimal("1");  
 a = result.divide(one,2,BigDecimal.*ROUND\_HALF\_UP*).doubleValue();//保留2位数  
 total = a ;  
 //Log.e("total",total+"");  
 }  
  
 tv\_total.setText("￥"+ total);  
}

### 11 个人信息修改

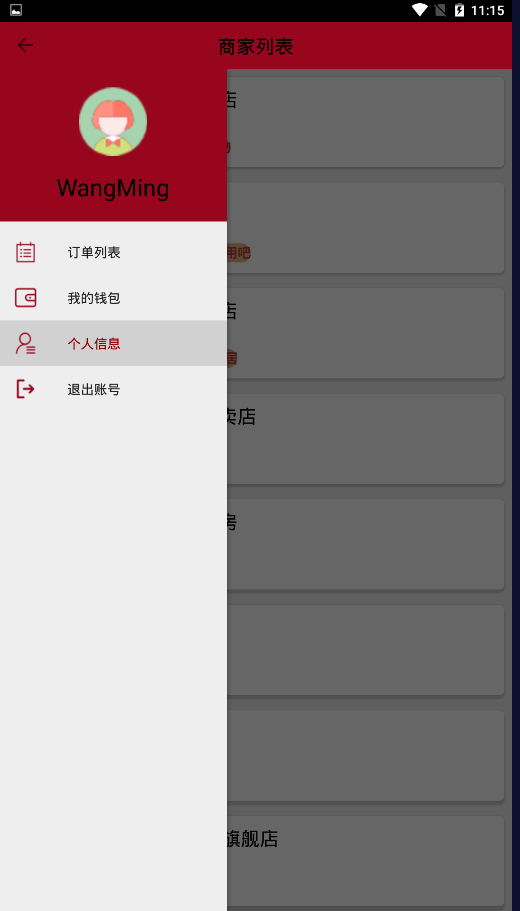


此功能为用户点击主页面个人信息按钮后，会跳转进入此页面，在次页面用户可以修改个人相关信息，点击修改，即可保存。

**核心代码：**

private void initView() {  
 tv\_userName = findViewById(R.id.*tv\_username*);  
 et\_nickName = findViewById(R.id.*et\_nickName\_per*);  
 et\_phonNumb = findViewById(R.id.*et\_phoneNumb\_per*);  
 et\_schoolName = findViewById(R.id.*et\_schoolName\_per*);  
 et\_apartmentNumb = findViewById(R.id.*et\_apartmentNumb\_per*);  
 btn\_alter = findViewById(R.id.*btn\_alter*);  
 btn\_cancle = findViewById(R.id.*btn\_cancle\_per*);  
  
 toolbar = findViewById(R.id.*toolbar*);  
 tv\_bar\_title = findViewById(R.id.*tv\_bar\_title*);  
 tv\_bar\_title.setText("个人信息");  
  
  
 //获取用户信息数据  
 userinfo = MySQLiteHelper.*getInstance*(getApplicationContext()).getUserInfoFromUserName(MainActivity.*username*);  
  
 tv\_userName.setText(userinfo.getUserName());  
 et\_nickName.setText(userinfo.getNickName());  
 et\_phonNumb.setText(userinfo.getPhoneNumb());  
 et\_schoolName.setText(userinfo.getSchoolName());  
 et\_apartmentNumb.setText(userinfo.getApartmentNumb());  
  
 btn\_alter.setOnClickListener(new View.OnClickListener() {  
  
 private Userinfo userinfo1;  
  
 @Override  
 public void onClick(View view) {  
 nickName = et\_nickName.getText().toString();  
 phoneNumb = et\_phonNumb.getText().toString();  
 schoolName = et\_schoolName.getText().toString();  
 apartmentNumb = et\_apartmentNumb.getText().toString();  
  
 if (TextUtils.*isEmpty*(nickName)) {  
 ToastUtil.*showShort*("昵称不能为空！");  
 return;  
 }  
 if (TextUtils.*isEmpty*(phoneNumb)) {  
 ToastUtil.*showShort*("手机号码不能为空！");  
 return;  
 }  
 if (TextUtils.*isEmpty*(schoolName)) {  
 ToastUtil.*showShort*("学校名称不能为空");  
 return;  
 }  
 if (TextUtils.*isEmpty*(apartmentNumb)) {  
 ToastUtil.*showShort*("公寓楼号不能为空");  
 return;  
 }  
  
 userinfo1 = new Userinfo();  
 userinfo1.setUserName(MainActivity.*username*);  
 userinfo1.setNickName(nickName);  
 userinfo1.setPassword(userinfo.getPassword());  
 userinfo1.setPhoneNumb(phoneNumb);  
 userinfo1.setSchoolName(schoolName);  
 userinfo1.setApartmentNumb(apartmentNumb);  
 //Log.e("修改后的信息",userinfo1.toString());  
 //在数据库中修改相关信息  
 MySQLiteHelper.*getInstance*(getBaseContext()).updateUserInfo(userinfo1);  
 ToastUtil.*showShort*("信息更新成功");  
 }  
 });  
  
 btn\_cancle.setOnClickListener(new View.OnClickListener() {  
 @Override  
 public void onClick(View view) {  
 PersonalCenterActivity.this.finish();  
 }  
 });  
}

### 12 退出切换账号功能



此功能为，用户点击退出账号后，即可退回登录界面，重新进行登录，注册等操作。

**核心代码：**

case R.id.*single\_4*:  
 startActivity(new Intent(HomeAllStoresActivity.this, MainActivity.class));  
 ShareUtils.*putAuto\_Login*("0");  
 HomeAllStoresActivity.this.finish();  
 break;

（四）系统测试

1.用户登录

2. 用户注册



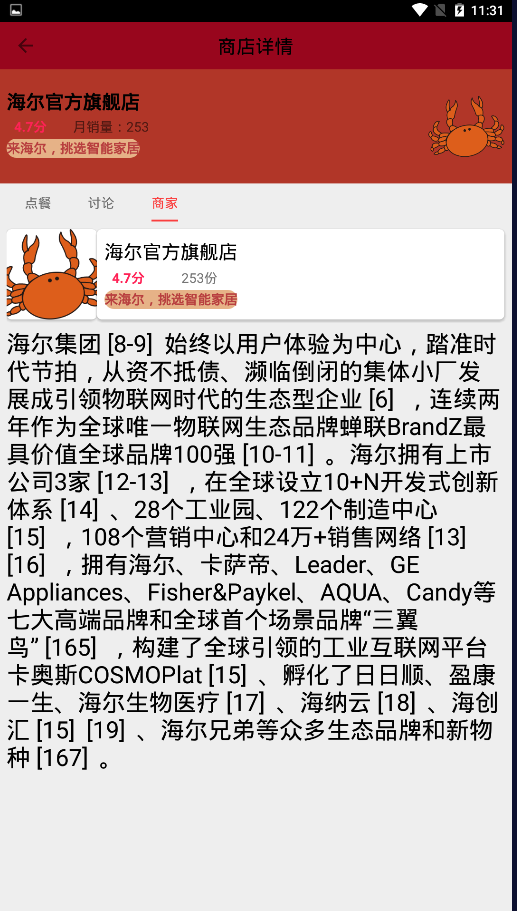
3．全部商家展示

4. 商家商品展示



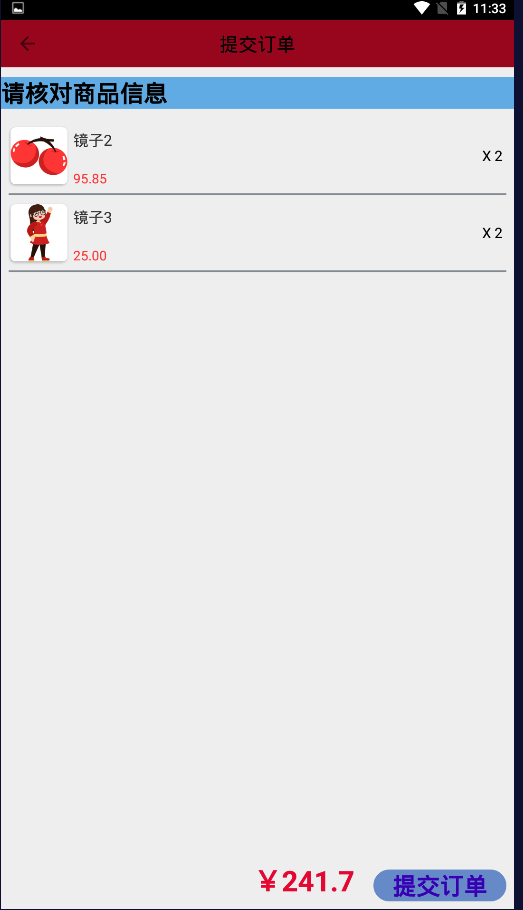
5. 模拟讨论功能

6. 商家信息展示

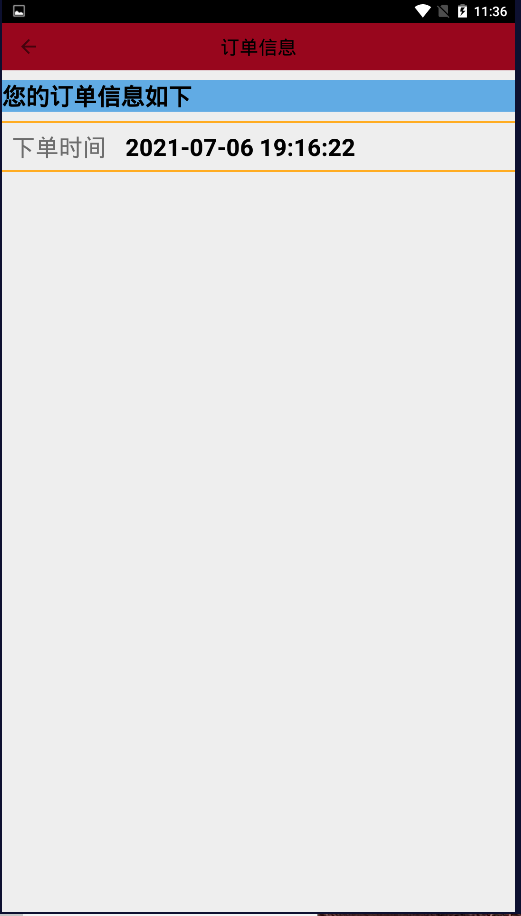


7. 购物车功能



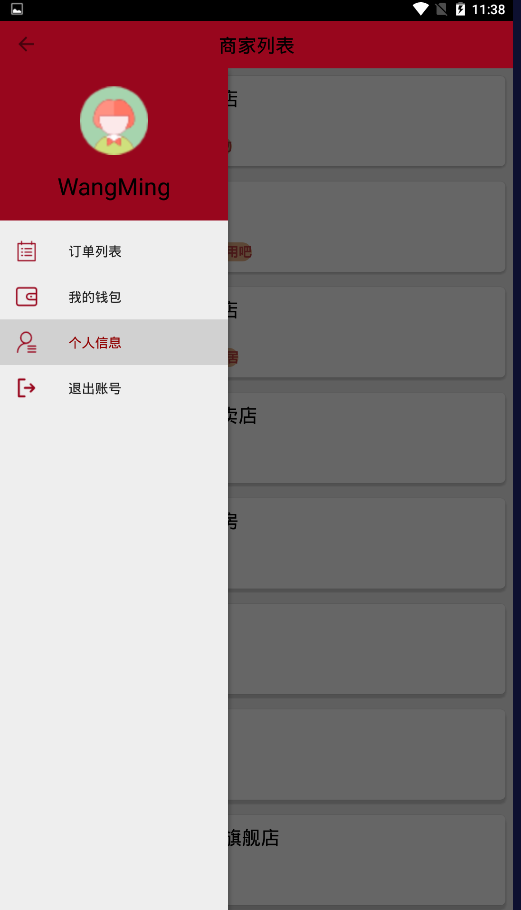
8. 结算功能

9. 账户充值功能

10. 查看订单功能

11. 个人信息修改

12. 退出切换账号功能



# 四．参考资料

[1]李刚.Android疯狂讲义（第二版）：清华大学出版社，2012  
[2]李兴华.名师讲坛——Android开发实战经典.北京：清华大学出版社，2012.  
[3]韩超.Android经典应用程序开发[J].人力资源出版社，2011:5-18  
[4]林城.Android2.3应用开发实战[J].机械工业出版社，2011:17-321  
[5] Android2高级编程（第2版）梅儿（Reto Meier）、王超 清华大学出版社  
[6]姚尚朗. Android 开发入门实战;人民邮电出版社，2009

# 五、总结及心得体会

在这次实训中我学习了很多，也得到了很多。实训是把理论与实际结合，通过对理论知识的理解，领悟从而运用到生活实际巩固所学的知识，提高对实际生活的认识，积累经验。在此期间学生能够初次体会到实际生产中的种种技能与经验。完成一项项项目能体现出独立思考能力。Android生产实习对目前应用现状进行了分析与比较，进而研究分析Android平台的系统架构和组件模型。在此基础上，基于Android平台设计和开发，实现对android编程的理解与应用，特别是在对于控件的应用产生很深的理解。  
 通过这近一个月的android实习，基本掌握了Android应用程序开发的一般流程。对常用控件基本掌握其用法，对其事件的监听方法也基本掌握。学习Android不仅是对前沿开发技术的了解，也是对编程知识的一次提升。经过对Android的控件、布局、Activity、Service等一系列基础知识的学习，我对整个Android的开发有了大致的了解。因为我们都是第一次接触android，所以老师先给我们讲了一个基本上每个程序语言都要编的，经典的程序，helloworld。通过这个小程序，我们知道了如何去创建一个android的项目，了解了android项目里的每个文件夹和文件的作用。教了一些常用的控件的使用，如：TextView，Button等等。然后老师又教我们如何布局，认识了linearlayout,tablelayout, AbsoluteLayout, RelativeLayout四种布局，分别为线型布局，表格布局，绝对布局和相对布局，并知道了如何通过这四种布局的相互嵌套来使程序的界面更加的友好，更加的让用户接受。例如: 要的布局(或者控件) ,在学习界面中，我发现Android为我们提供了很好的类似反射机制，通过Layout文件夹下的配置文件，可以快速的形成界面，在配置文件可以设置属性或者样式都是很快捷方便。对比较特殊的界面也可以通过处理嵌入到指定的界面，同样也可以通过java代码直接创建View进行添加，不过这种方式比较复杂。对一些点击、选中、按键等处理的事件，界面之间的跳转Intent管理.在Android编程过程中巩固熟悉了Java的编程。由于Android应用程序的开发离不开Java的支持，所以基础的Java知识是必须的。由于我们刚刚结束了java课程，马上展开了android实训，而android和java之间又有很大的联系，所以，我们学习android也比较容易入门。最终得到的成效也很好。  
这次的实训再让我们认识到Android的同时，更加让我们认识到学习态度的重要性，在开发过程中，我们都有自己的分工，在老师的指导下，我们完成了项目。我们会在未来的日子里更加深刻的认识到这次实训带给我们的，让我们能受益的体会与思考。