

中山大学移动信息工程学院本科生实验报告

(2017 年秋季学期)

课程名称：移动应用开发

任课教师：郑贵锋

年级	2015 级	专业 (方向)	移动信息工程 (移动互联网)
学号	15352083	姓名	冯灏帆
电话	13712309295	Email	ares_fung@qq.com
开始日期	2017.10.17	完成日期	2017.10.22

一、 实验题目

1. 复习时间处理；
2. 学习 Intent、Bundle 在 Activity 跳转中的应用；
3. 学习 RecyclerView、ListView 以及各类适配器的用法。

二、 实现内容

本次实验模拟一个商品表，有两个界面，第一个界面用于呈现商品，如下所示：

E Enchated Forest

A Arla Milk

D Devondale Milk

K Kindle Oasis

W waitrose 早餐麦片

M Mcvitie's 饼干

F Ferrero Rocher



下面还有数据，就不截图了，数据在素材中有给出；点击右下方的悬浮按钮可以切换到购物车：

上午10:07

🔔 📶 🔋

购物车

价格

D

Devondale Milk

¥ 79.00

W

waitrose 早餐麦片

¥179.00

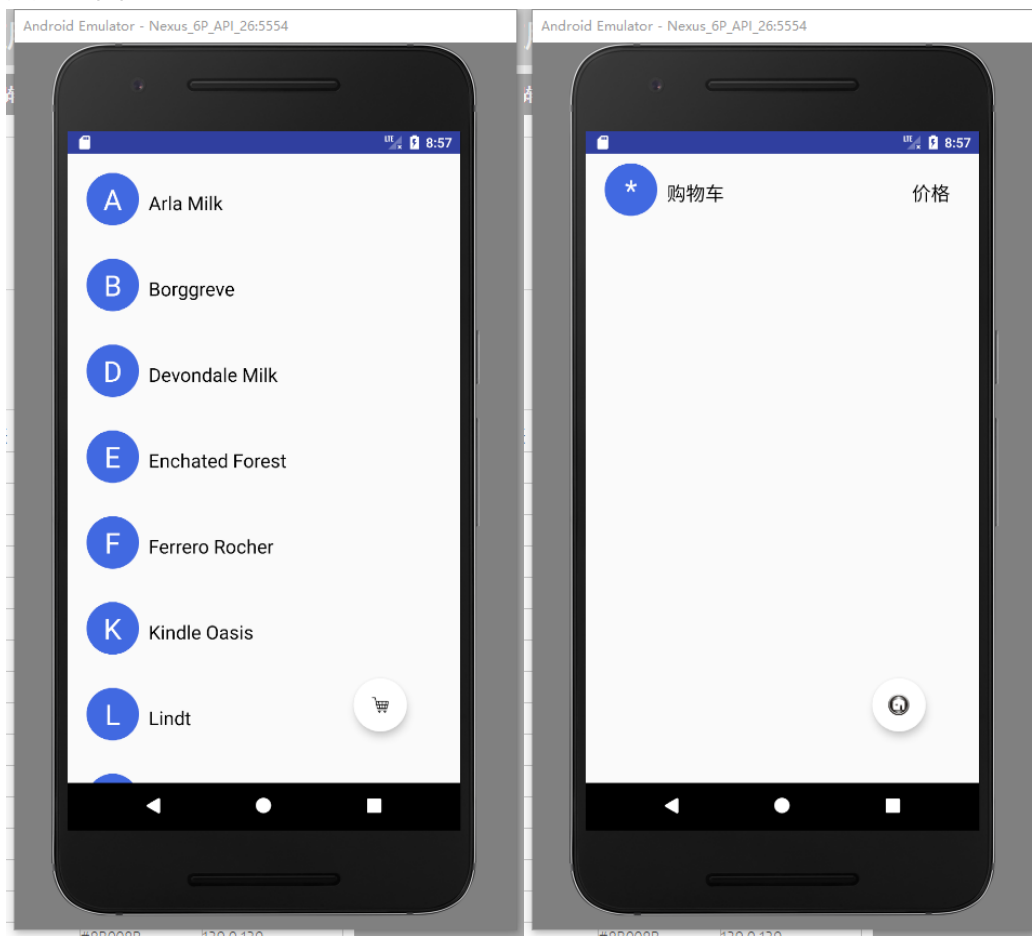


上面两个列表点击任意一项后，可以看到详细的信息：

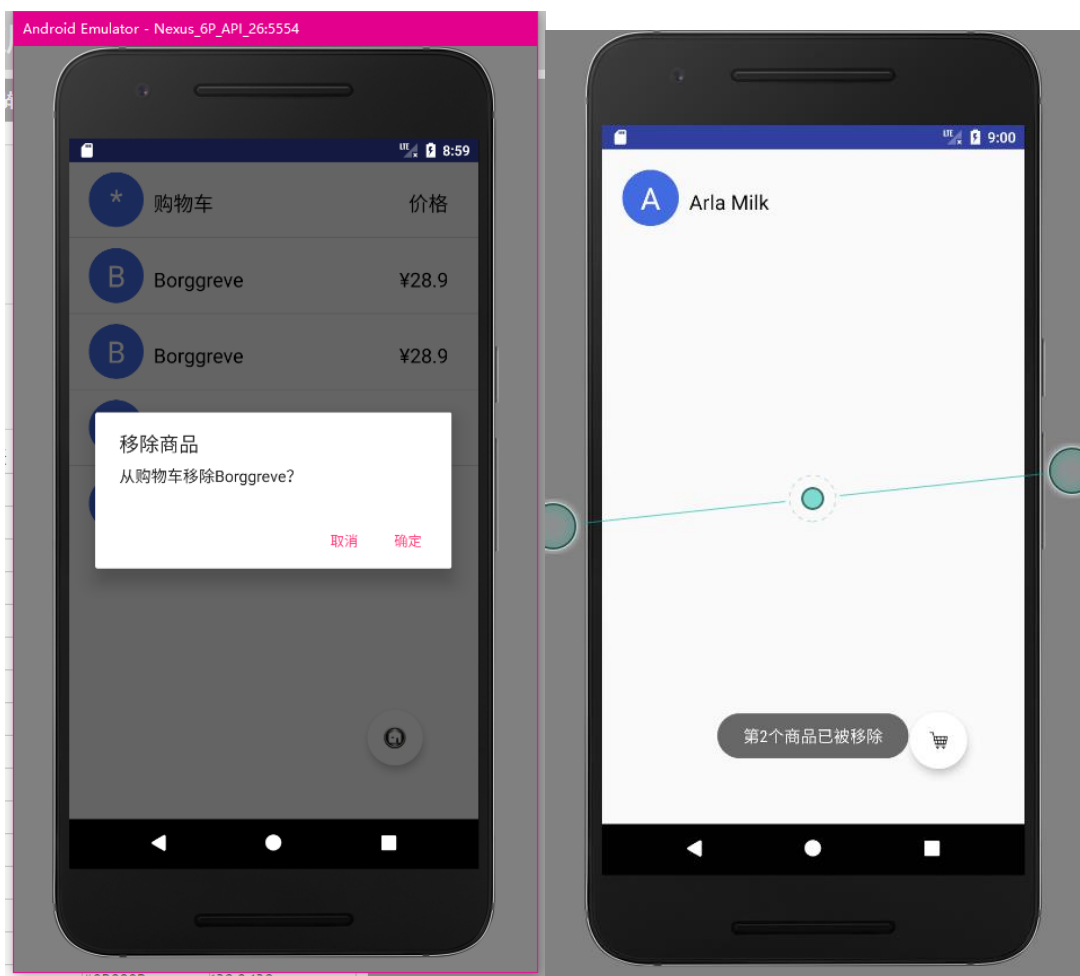


三、 课堂实验结果

(1) 实验截图







(2) 实验步骤以及关键代码

首先将与商品相关的信息进行封装存储，包括商品的图片来源与收藏记录。

```

public class Merchandise implements Serializable {
    private String name;
    private double price;
    private String info;
    private int picname;
    private boolean mark;

    public Merchandise(String name, double price, String info, int picname, boolean mark) {
        this.name = name;
        this.price = price;
        this.info = info;
        this.picname = picname;
        this.mark = mark;
    }

    public void setName(String name) { this.name = name; }
    public void setPrice(double price) { this.price = price; }
    public void setInfo(String info) { this.info = info; }
    public void setPicname(int picname) { this.picname = picname; }
    public void setMark(boolean mark) { this.mark = mark; }

    public String getName() { return name; }
    public double getPrice() { return price; }
    public String getInfo() { return info; }
    public int getPicname() { return picname; }
    public boolean getMark() { return mark; }
}

```

重载 RecyclerView 的 Adapter 实现，利用 ArrayList 进行数据导入并且对 Item 设立点击与长按监听事件的接口，传出被触发监听的 Item 的位置参数。

```

public interface OnItemClickListener {
    void onClick(int position);
    boolean onLongClick(int position);
}

public void setOnItemClickListener(OnItemClickListener onItemClickListener) {
    this.mOnItemClickListener = onItemClickListener;
}

@Override
public void onBindViewHolder(final ViewHolder holder, int position) {
    holder.Item.setText(mData.get(position).getName());
    holder.Firstletter.setText(String.valueOf(mData.get(position).getName().charAt(0)));
    if(mOnItemClickListener!=null) {
        holder.itemView.setOnClickListener((v) -> {
            mOnItemClickListener.onClick(holder.getAdapterPosition());
        });
        holder.itemView.setOnLongClickListener((v) -> {
            mOnItemClickListener.onLongClick(holder.getAdapterPosition());
            return false;
        });
    }
    holder.itemView.setTag(position);
}

```

重载 ListView 的 Adapter 实现，利用 ArrayList 进行数据导入并且在 java 文件中对 Item 设立点击与长按监听事件。对于 Item 的点击事件，通过切换 activity 的方式进入商品详情页，同时利用 Intent 将商品信息打包到 Extra 中，并启动新的 activity。对于 Item 的长按时间，发出对话框确认是否删除购物车内某一商品，点击确认按键后从数据导入的 ArrayList 中删除对应的元素并对 ListView 的 Adapter 调用 updateData 成员函数提醒更新数据。


```

mListView.setOnItemClickListener((adapterView, view, i, l) -> {
    if(i>0) {
        Intent intent = new Intent();
        intent.setClass( packageContext: MainActivity.this, details.class);
        Merchandise senddata = new Merchandise(shoplistdata.get(i).getName(), shoplistdata.get(i).getPrice(),
                                                shoplistdata.get(i).getInfo(), shoplistdata.get(i).getPicname(),
                                                shoplistdata.get(i).getMark());
        intent.putExtra( name: "Merchandise", senddata);
        startActivityForResult(intent, requestCode: 0);
    }
});

mListView.setOnItemLongClickListener((adapterView, view, position, l) -> {
    if(position>0) {
        AlertDialog.Builder builder = new AlertDialog.Builder( context: MainActivity.this);
        AlertDialog dialog = builder.create();
        builder.setTitle("移除商品");
        builder.setMessage("从购物车移除"+shoplistdata.get(position).getName()+"? ");
        builder.setPositiveButton( text: "确定", (dialogInterface, i) -> {
            shoplistdata.remove(position);
            mAdapter2.updateData(shoplistdata);
        });
        builder.setNegativeButton( text: "取消", listener: null);
        builder.setCancelable(true);
        builder.show();
    }
    return true;
});

```

对于商品列表与购物车页面的切换则是通过在一个 activity 中两个 view 的 visibility 的变化实现，一个 view 为 VISIBLE 时另一个 view 为 GONE，此时被隐藏的 view 不会占用布局空间。购物车按钮的实现类为 FloatingActionButton，对它设置了点击监听事件作为购物车与商品列表切换的触发，并且通过 setImageResource 的成员函数切换该 FloatingActionButton 在不同界面下的图标。

```

if(current=="RecyclerView") {
    mListView.setVisibility(View.GONE);
    mRecyclerView.setVisibility(View.VISIBLE);
};
if(current=="ListView") {
    mRecyclerView.setVisibility(View.GONE);
    mListView.setVisibility(View.VISIBLE);
};

final FloatingActionButton mFloatingActionButton = (FloatingActionButton) findViewById(R.id.fab);
mFloatingActionButton.setOnClickListener((view) -> {
    if(current=="RecyclerView") {
        current = "ListView";
        mRecyclerView.setVisibility(View.GONE);
        mListView.setVisibility(View.VISIBLE);
        mFloatingActionButton.setImageResource(R.mipmap.mainpage);
    }
    else {
        current = "RecyclerView";
        mListView.setVisibility(View.GONE);
        mRecyclerView.setVisibility(View.VISIBLE);
        mFloatingActionButton.setImageResource(R.mipmap.shoplist);
    }
});

```

(3) 实验遇到困难以及解决思路

最大的困难在于对 RecyclerView 与 Adapter 的理解，通过阅读文档后慢慢分析代码，逐步解析重构部分代码，成功实现 RecyclerView 的功能拓展。

四、 课后实验结果

众所周知，对于 Intent 类在执行动作前绑定传输数据的 Extra，往往仅能传输标准数据类型。而对于自定义数据类型的传输则需要使用 Serializable 方法，修改后便可以通过调用 putExtra 与 getSerializableExtra 成员函数在 Intent 中绑定自定义类型数据传递于 activity 间。

五、 实验思考及感想

本次实验着重于事件处理的使用练习，同时着重锻炼了对于 java 代码的编写，在理解相关知识机制的前提下，自行对已有开发设计类进行功能的拓

展以满足实际开发设计的需求，十分考验基础理论知识的扎实程度与思维严谨度、代码编写能力。经过本次实验练习对于自行设计相对简单的 App 有了很大的操作信心，对于专业文档的阅读与理解能力增长不少。