Android 应用软件设计

E 3　Adapter View

学号：SA17225268

姓名：彭明

报告撰写时间：2017/10/16

1.主题概述

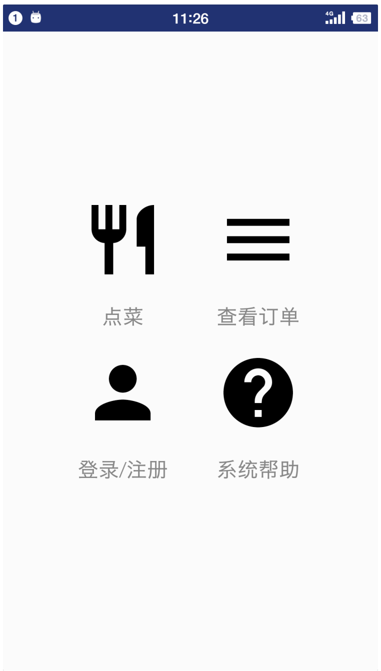
本次作业的主题是围绕 view 的 Adapter 来进行的。我们通过 view 来绑定 Adapter，在 Adapter 里加载数据，来实现在 view 里动态显示内容。

2.假设

我们需要假设用户在点击了登录/注册之后，再进入点菜和查看订单的选项。

3.实现或证明

1. **使用 GridView 替换已有的导航。**



MainScreen 的代码如下：

```
public class MainScreen extends AppCompatActivity {
    private final static int REQUESTCODE = 1;
    @BindView(R.id.gv)
    GridView gv;

    MyAdapter myAdapter;

    //GridView 的数据源
    private List<Map<String, Object>> list;
    private int images[] = {
            R.drawable.ic_restaurant_black_24dp,
            R.drawable.ic_menu_black_24dp,
            R.drawable.ic_person_black_24dp,
            R.drawable.ic_help_black_24dp
    };
    private int texts[] = {
            R.string.title_ordering,
            R.string.title_myorder,
            R.string.title_login,
            R.string.title_help
    };

    User user;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main_screen);
        ButterKnife.bind(this);

        list = new ArrayList<Map<String, Object>>();
        for(int i = 0; i < images.length; i++){
            Map<String, Object> map = new HashMap<String, Object>();
            map.put("image", images[i]);
            map.put("text", texts[i]);
            list.add(map);
        }
        //GridView 绑定 Adapter
        myAdapter = new MyAdapter();
        gv.setAdapter(myAdapter);

        gv.setOnItemClickListener(new AdapterView.OnItemClickListener()
{
            @Override
            public void onItemClick(AdapterView<?> parent, View view, int
position, long id) {
                Intent loginIntent;
                //只显示登录/注册和系统帮助的情况
                if(parent.getCount() == 2){
                    switch (position){
                        case 0:
```

```
                        loginIntent = new Intent(MainScreen.this, 
LoginOrRegister.class);
                        startActivityForResult(loginIntent, 
REQUESTCODE);
                        break;
                    case 1:
                        break;
                }
            } else {
                //四个选项都显示的情况
                switch (position){
                    case 0:
                        //10.页面跳转至 FoodView，并传入 user
                        Intent foodViewIntent = new 
Intent(MainScreen.this, FoodView.class);
                        foodViewIntent.putExtra("foodViewUser", user);
                        startActivity(foodViewIntent);
                        break;
                    case 1:
                        //10.页面跳转至 FoodOrderView，并传入 user
                        Intent foodOrderViewIntent = new 
Intent(MainScreen.this, FoodOrderView.class);
                        
foodOrderViewIntent.putExtra("foodOrderViewUser", user);
                        startActivity(foodOrderViewIntent);
                        break;
                    case 2:
                        loginIntent = new Intent(MainScreen.this, 
LoginOrRegister.class);
                        startActivityForResult(loginIntent, 
REQUESTCODE);
                        break;
                    case 3:
                        break;
                }
                }
            }
        });

        //根据 SCOSEntry 传过来的数据决定是否只显示登录/注册和系统帮助
        try {
            String message =
getIntent().getStringExtra(SCOSEntry.EXTRA_MESSAGE);
            if (!message.equals("FromEntry")) {
                list.remove(1);
                list.remove(0);
                myAdapter.notifyDataSetChanged();
            }
        } catch (Exception e) {
        }
    }

    class MyAdapter extends BaseAdapter{
        @Override
        public int getCount() {
            return list.size();
        }

        @Override
        public Object getItem(int position) {
            return list.get(position);
        }

        @Override
        public long getItemId(int position) {
            return position;
        }

        @Override
        public View getView(int position, View convertView, ViewGroup
parent) {

            convertView = LayoutInflater.from(MainScreen.this)
                    .inflate(R.layout.gridview_layout, parent, false);
            ImageView imageView = (ImageView)
convertView.findViewById(R.id.grid_iv);
            imageView.setImageResource((Integer)
list.get(position).get("image"));
            TextView textView = (TextView)
convertView.findViewById(R.id.grid_tv);
            textView.setText((Integer) list.get(position).get("text"));

            return convertView;
        }
    }

}
```

2. 新建 package 为：es.source.code.model
3. 在包中新建 model 类 User.java 和 Food.java

```
public class User implements Serializable{
    private String userName;
    private String password;
    private Boolean oldUser;

    public User(){
    }

    public User(String userName, String password, Boolean oldUser){
        this.userName = userName;
        this.password = password;
        this.oldUser = oldUser;
    }
    public String getUserName() {
        return userName;
    }
    public void setUserName(String userName) {
        this.userName = userName;
    }

    public String getPassword() {
        return password;
    }
    public void setPassword(String password) {
        this.password = password;
    }

    public Boolean getOldUser() {
        return oldUser;
    }
    public void setOldUser(Boolean oldUser) {
        this.oldUser = oldUser;
    }
}
```

```
public class Food implements Serializable{
    private String foodname;
    private String foodprice;
    private int imageurl;
    private String comment;
    private int number;
    private Boolean isOrdered;

    public Food(){
    }

    public Food(String foodname, String foodprice, int imageurl, String comment, int number, Boolean isOrdered){
        this.foodname = foodname;
        this.foodprice = foodprice;
        this.imageurl = imageurl;
        this.comment = comment;
        this.number = number;
        this.isOrdered = isOrdered;
    }

    public Boolean getIsOrdered() {
        return isOrdered;
    }
    public void setIsOrdered(Boolean ordered) {
        isOrdered = ordered;
    }

    public String getFoodname() {
        return foodname;
    }
    public void setFoodname(String foodname) {
        this.foodname = foodname;
    }

    public String getFoodprice() {
        return foodprice;
    }
    public void setFoodprice(String foodprice) {
        this.foodprice = foodprice;
    }

    public int getImageurl() {
        return imageurl;
    }
    public void setImageurl(int imageurl) {
        this.imageurl = imageurl;
    }
    public String getComment() {
        return comment;
    }
    public void setComment(String comment) {
        this.comment = comment;
    }

    public int getNumber() {
        return number;
    }
    public void setNumber(int number) {
        this.number = number;
    }
}
```

4. 修改 LoginOrRegister

```
private Boolean oldUser = true;
      //onCreat 中对注册登录的响应
      mEmailSignInButton.setOnClickListener(new OnClickListener() {
          @Override
          public void onClick(View view) {
              attemptLogin();
          }
       });

       mEmailSignUpButton.setOnClickListener(new OnClickListener() {
          @Override
          public void onClick(View v) {
              oldUser = false;
              attemptLogin();
          }
       });
```

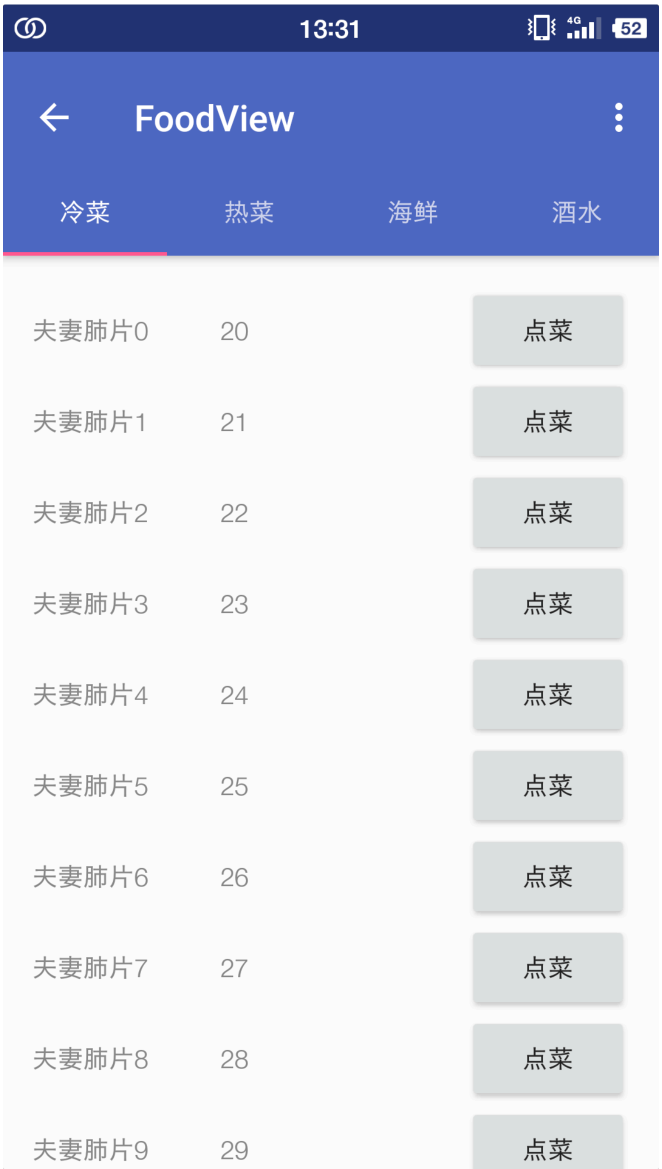
```
private void attemptLogin() {
        // Reset errors.
        mEmailView.setError(null);
        mPasswordView.setError(null);

        // Store values at the time of the login attempt.
        String username = mEmailView.getText().toString();
        String password = mPasswordView.getText().toString();

        boolean cancel = false;
        View focusView = null;

        // Check for a valid password.
        if (TextUtils.isEmpty(password)) {

mPasswordView.setError(getString(R.string.error_password_required));
            focusView = mPasswordView;
            cancel = true;
        } else if (!isValid(password)) {
            mPasswordView.setError(getString(R.string.error_invalid));
            focusView = mPasswordView;
            cancel = true;
        }

        // Check for a valid username.
        if (TextUtils.isEmpty(username)) {

mEmailView.setError(getString(R.string.error_username_required));
            focusView = mEmailView;
            cancel = true;
        } else if (!isValid(username)) {
            mEmailView.setError(getString(R.string.error_invalid));
            focusView = mEmailView;
            cancel = true;
        }
        if (cancel) {
            focusView.requestFocus();
        } else {
            mProgressView.setVisibility(View.VISIBLE);

            User loginUser = new User();
            loginUser.setUserName(username);
            loginUser.setPassword(password);
            if(oldUser){
                loginUser.setOldUser(oldUser);
                returnIntent.putExtra(RETURN_TAG, "LoginSuccess");
                returnIntent.putExtra("userData", loginUser);
                setResult(RESULT_OK, returnIntent);
            } else {
                loginUser.setOldUser(oldUser);
                returnIntent.putExtra(RETURN_TAG, "RegisterSuccess");
                returnIntent.putExtra("userData", loginUser);
                setResult(RESULT_FIRST_USER, returnIntent);
            }

            mHandler.postDelayed(new Runnable() {
                @Override
                public void run() {
                    mProgressView.setVisibility(View.GONE);
                    finish();
                }
            }, 2000);
        }
    }

    private boolean isValid(String word) {
        Pattern p = Pattern.compile("^[A-Za-z0-9]+",
Pattern.CASE_INSENSITIVE);
```

```
        Matcher m = p.matcher(word);
        return m.matches();
    }

```

5. **在 MainScreen 中对 LoginOrRegister 传回的数据进行处理**

```
@Override
protected void onActivityResult(int requestCode, int resultCode, Intent
data) {
    super.onActivityResult(requestCode, resultCode, data);
    if (resultCode == RESULT_OK || resultCode == RESULT_FIRST_USER) {
        if (requestCode == REQUESTCODE) {
            String returnMessage =
data.getStringExtra(LoginOrRegister.RETURN_TAG);

            if (returnMessage.equals("RegisterSuccess")){
                Toast.makeText(this, "欢迎您成为 SCOS 新用户",
Toast.LENGTH_LONG).show();
            }

            if (returnMessage.equals("LoginSuccess") ||
returnMessage.equals("RegisterSuccess")) {
                if(gv.getCount() == 2){
                    list.clear();

                    for(int i = 0; i < images.length; i++){
                        Map<String, Object> map = new HashMap<String,
Object>();
                        map.put("image", images[i]);
                        map.put("text", texts[i]);
                        list.add(map);
                    }
                    myAdapter.notifyDataSetChanged();
                }
                user = (User) data.getSerializableExtra("userData");
            } else {
                user = null;
            }
        }
    }
}
```

6. 在 SCOS 的 **es.source.code.activity** 包下新建 **Activity** 类名为：**FoodView**，为该 **FoodView** 定制布局 **food_view.xml**



夫妻肺片0 20 点菜

夫妻肺片1 21 点菜

夫妻肺片2 22 点菜

夫妻肺片3 23 点菜

夫妻肺片4 24 点菜

夫妻肺片5 25 点菜

夫妻肺片6 26 点菜

夫妻肺片7 27 点菜

夫妻肺片8 28 点菜

夫妻肺片9 29 点菜

```
public class FoodView extends AppCompatActivity {
    @BindView(R.id.toolbar_food)
    Toolbar toolbar;
    @BindView(R.id.tabs_food)
    TabLayout tabLayout;
    @BindView(R.id.container_food)
    ViewPager mViewPager;

    private SectionsPagerAdapter mSectionsPagerAdapter;
    //9.添加 User 变量
    User user;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.food_view);
        ButterKnife.bind(this);

        setSupportActionBar(toolbar);
        getSupportActionBar().setDisplayHomeAsUpEnabled(true);
        // Create the adapter that will return a fragment for each of the
three
        // primary sections of the activity.
        mSectionsPagerAdapter = new
SectionsPagerAdapter(getSupportFragmentManager());

        // Set up the ViewPager with the sections adapter.
        mViewPager.setAdapter(mSectionsPagerAdapter);

        tabLayout.setupWithViewPager(mViewPager);

        try{
            //9. User 变量
            user = (User)
getIntent().getSerializableExtra("foodViewUser");
```

```
        } catch (Exception e){
        }
    }



    @Override
    public boolean onCreateOptionsMenu(Menu menu) {
        getMenuInflater().inflate(R.menu.menu_food_view, menu);
        return true;
    }

    @Override
    public boolean onOptionsItemSelected(MenuItem item) {
        switch (item.getItemId()) {
            case R.id.ordered_food:
                //11.跳转到 FoodOrderView 的未下单页
                Intent orderedFood = new Intent(this,
FoodOrderView.class);
                orderedFood.putExtra("startPage", 0);
                orderedFood.putExtra("foodViewUser", user);
                startActivity(orderedFood);
                break;
            case R.id.look_order:
                //11.跳转到 FoodOrderView 的已下单页
                Intent lookOrder = new Intent(this, FoodOrderView.class);
                lookOrder.putExtra("startPage", 1);
                lookOrder.putExtra("foodViewUser", user);
                startActivity(lookOrder);
                break;
            case R.id.call_service:

                break;
        }

        return super.onOptionsItemSelected(item);
    }

    /**
     * A placeholder fragment containing a simple view.
     */
    public static class PlaceholderFragment extends Fragment {
        /**
         * The fragment argument representing the section number for this
         * fragment.
         */
        private static final String ARG_SECTION_NUMBER = "section_number";

        FoodRvAdapter foodAdapter;
        ArrayList<Food> foods;

        @BindView(R.id.food_list)
        RecyclerView foodList;
        Unbinder unbinder;

        public PlaceholderFragment() {
        }

        /**
         * Returns a new instance of this fragment for the given section
         * number.
         */
        public static PlaceholderFragment newInstance(int sectionNumber)
{
            PlaceholderFragment fragment = new PlaceholderFragment();
            Bundle args = new Bundle();
            args.putInt(ARG_SECTION_NUMBER, sectionNumber);
            fragment.setArguments(args);
            return fragment;
        }
        @Override
        public View onCreateView(LayoutInflater inflater, final ViewGroup
container,
                             Bundle savedInstanceState) {
            View rootView = inflater.inflate(R.layout.fragment_food_view,
container, false);
            unbinder = ButterKnife.bind(this, rootView);

            LinearLayoutManager llm = new
LinearLayoutManager(getActivity());
            llm.setOrientation(LinearLayoutManager.VERTICAL);
            foodList.setLayoutManager(llm);

            foods = new ArrayList<Food>();
            switch (getArguments().getInt(ARG_SECTION_NUMBER)) {
                case 1:
                    for (int i = 0; i < 10; i++) {
                        Food food1 = new Food("夫妻肺片" + i % 10, "2" + i %
10, R.drawable.food1, "", 1, false);
                        foods.add(food1);
                    }
                    break;
                case 2:
                    for (int i = 0; i < 15; i++) {
                        Food food2 = new Food("水煮肉片" + i % 10, "3" + i %
10, R.drawable.food2, "", 1, false);
                        foods.add(food2);
                    }
                    break;
                case 3:
                    for (int i = 0; i < 15; i++) {
                        Food food3 = new Food("蛤蜊" + i % 10, "4" + i % 10,
R.drawable.food3, "", 1, false);
                        foods.add(food3);
                    }
```

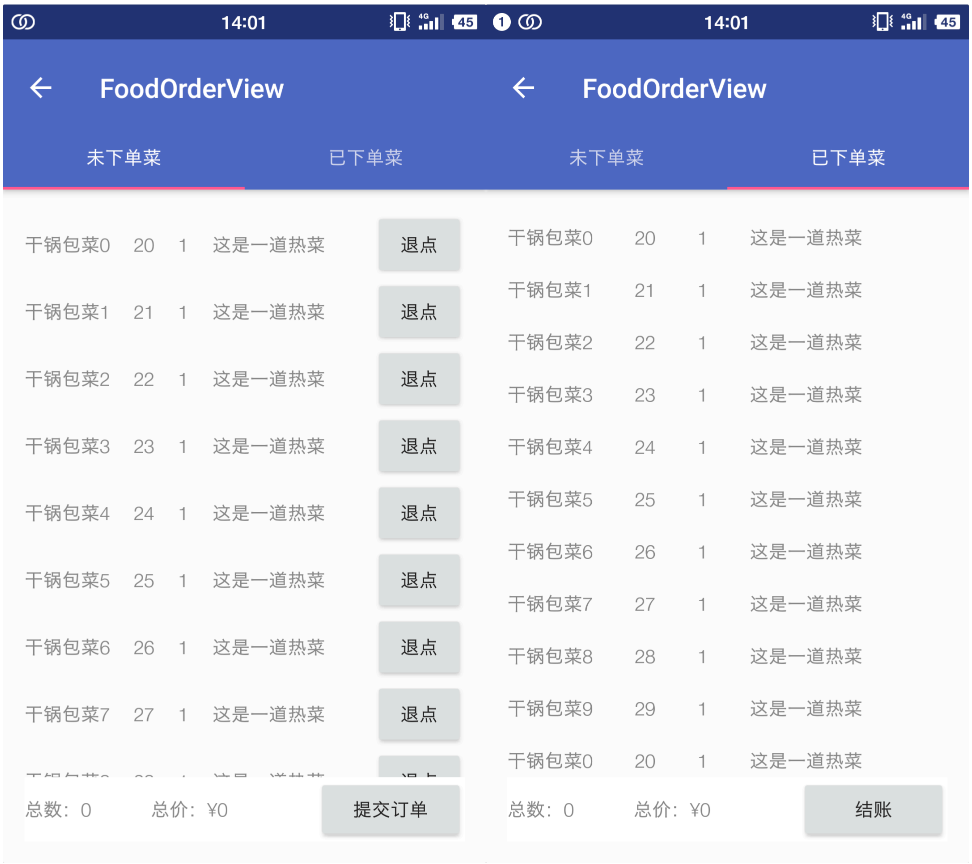
```
                    break;
                case 4:
                    for (int i = 0; i < 15; i++) {
                        Food food4 = new Food("勇闯天涯" + i % 10, "1" + i % 10, R.drawable.food4, "", 1, false);
                        foods.add(food4);
                    }
                    break;
            }

            foodAdapter = new FoodRvAdapter(container.getContext(), foods);
            foodList.setAdapter(foodAdapter);

            //11.跳转到菜品详情页
            foodAdapter.setOnItemClickListener(new FoodRvAdapter.OnItemClickListener() {
                @Override
                public void onItemClick(View view, int position) {
                    Intent foodDetail = new Intent(view.getContext(), FoodDetailed.class);
                    foodDetail.putExtra("foods", foods);
                    foodDetail.putExtra("position", position);
                    startActivity(foodDetail);
                }
            });

            return rootView;
        }

        @Override
        public void onDestroyView() {
            super.onDestroyView();
            unbinder.unbind();
        }
```

```
    }


    public class SectionsPagerAdapter extends FragmentPagerAdapter {

        public SectionsPagerAdapter(FragmentManager fm) {
            super(fm);
        }

        @Override
        public Fragment getItem(int position) {
            return PlaceholderFragment.newInstance(position + 1);
        }

        @Override
        public int getCount() {
            return 4;
        }

        @Override
        public CharSequence getPageTitle(int position) {
            switch (position) {
                case 0:
                    return "冷菜";
                case 1:
                    return "热菜";
                case 2:
                    return "海鲜";
                case 3:
                    return "酒水";
            }
            return null;
        }
    }
}
```

FoodView 绑定的 Adapter

```
public class FoodRvAdapter extends
RecyclerView.Adapter<FoodRvAdapter.MyViewHolder> implements
View.OnClickListener{
    private Context context;
    private ArrayList<Food> foods;

    //RecyclerView 没有自带的 itemClicklistener，这里自己写的一个 Listener
    private OnItemClickListener mOnItemClickListener = null;

    public static interface OnItemClickListener{
        void onItemClick(View view, int position);
    }

    public FoodRvAdapter(Context context, ArrayList<Food> foods){
        this.context = context;
        this.foods = foods;
    }

    @Override
    public FoodRvAdapter.MyViewHolder onCreateViewHolder(ViewGroup
parent, int viewType) {
        View itemView =
LayoutInflater.from(context).inflate(R.layout.food_rv_layout, parent,
false);
        MyViewHolder viewHolder = new MyViewHolder(itemView);
        itemView.setOnClickListener(this);
        return viewHolder;
    }

    @Override
    public void onBindViewHolder(final FoodRvAdapter.MyViewHolder holder,
final int position) {
        holder.foodName.setText(foods.get(position).getFoodname());
        holder.foodPrice.setText(foods.get(position).getFoodprice());
        holder.orderFood.setOnClickListener(new View.OnClickListener()
{
            @Override
            public void onClick(View v) {
                if(!foods.get(position).getIsOrdered()){
                    Toast.makeText(context, "点菜成功",
Toast.LENGTH_SHORT).show();
                    foods.get(position).setIsOrdered(true);
                    holder.orderFood.setText(R.string.food_return);
                } else {
                    Toast.makeText(context, "退点成功",
Toast.LENGTH_SHORT).show();
                    foods.get(position).setIsOrdered(false);
                    holder.orderFood.setText(R.string.food_ordering);
                }
            }
        });

        holder.itemView.setTag(position);
    }

    @Override
    public void onClick(View v) {
        if(mOnItemClickListener != null){
            mOnItemClickListener.onItemClick(v, (int)v.getTag());
        }
    }

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

    @Override
    public int getItemCount() {
        return foods.size();
    }

    public class MyViewHolder extends RecyclerView.ViewHolder {
        TextView foodName, foodPrice;
        Button orderFood;

        public MyViewHolder(View itemView) {
            super(itemView);
            foodName = itemView.findViewById(R.id.food_name);
            foodPrice = itemView.findViewById(R.id.food_price);
            orderFood = itemView.findViewById(R.id.order_food);
        }
    }
}
```

7. **在 SCOS 的 es.source.code.activity 包下新建 Activity 类名为：FoodOrderView，为该 FoodOrderView 定制布局 food_order_view.xml**



干锅包菜0 20 1 这是一道热菜 退点
干锅包菜1 21 1 这是一道热菜 退点
干锅包菜2 22 1 这是一道热菜 退点
干锅包菜3 23 1 这是一道热菜 退点
干锅包菜4 24 1 这是一道热菜 退点
干锅包菜5 25 1 这是一道热菜 退点
干锅包菜6 26 1 这是一道热菜 退点
干锅包菜7 27 1 这是一道热菜 退点
总数：0 总价：¥0 提交订单

干锅包菜0 20 1 这是一道热菜
干锅包菜1 21 1 这是一道热菜
干锅包菜2 22 1 这是一道热菜
干锅包菜3 23 1 这是一道热菜
干锅包菜4 24 1 这是一道热菜
干锅包菜5 25 1 这是一道热菜
干锅包菜6 26 1 这是一道热菜
干锅包菜7 27 1 这是一道热菜
干锅包菜8 28 1 这是一道热菜
干锅包菜9 29 1 这是一道热菜
干锅包菜0 20 1 这是一道热菜
总数：0 总价：¥0 结账

```java
public class FoodOrderView extends AppCompatActivity {
    @BindView(R.id.toolbar_food_order)
    Toolbar toolbar;
    @BindView(R.id.tabs_food_order)
    TabLayout tabLayout;
    @BindView(R.id.container_food_order)
    ViewPager mViewPager;

    private SectionsPagerAdapter mSectionsPagerAdapter;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.food_order_view);
        ButterKnife.bind(this);

        setSupportActionBar(toolbar);
        getSupportActionBar().setDisplayHomeAsUpEnabled(true);
        // Create the adapter that will return a fragment for each of the
three
        // primary sections of the activity.
        mSectionsPagerAdapter = new
SectionsPagerAdapter(getSupportFragmentManager());

        // Set up the ViewPager with the sections adapter.
        mViewPager.setAdapter(mSectionsPagerAdapter);

        tabLayout.setupWithViewPager(mViewPager);

        //设置启动的 Fragment
        try{
            int startPage = getIntent().getIntExtra("startPage", 0);
            mViewPager.setCurrentItem(startPage);
        } catch (Exception e){
        }
    }
    @Override
    public boolean onCreateOptionsMenu(Menu menu) {
        getMenuInflater().inflate(R.menu.menu_food_order_view, menu);
        return true;
    }

    @Override
    public boolean onOptionsItemSelected(MenuItem item) {
        return super.onOptionsItemSelected(item);
    }

    /**
     * A placeholder fragment containing a simple view.
     */
    public static class PlaceholderFragment extends Fragment {
        /**
         * The fragment argument representing the section number for this
         * fragment.
         */
        private static final String ARG_SECTION_NUMBER = "section_number";

        FoodOrderRvAdapter foodOrderAdapter;
        ArrayList<Food> foods;
        //9. 添加 User 变量
        User user;

        @BindView(R.id.food_order_list)
        RecyclerView foodOrderList;
        @BindView(R.id.order_number)
        TextView orderNumber;
        @BindView(R.id.order_price)
        TextView orderPrice;
        @BindView(R.id.order_submit)
        Button orderSubmit;
        Unbinder unbinder;

        public PlaceholderFragment() {
        }

        /**
         * Returns a new instance of this fragment for the given section
         * number.
         */
        public static PlaceholderFragment newInstance(int sectionNumber)
{
            PlaceholderFragment fragment = new PlaceholderFragment();
            Bundle args = new Bundle();
            args.putInt(ARG_SECTION_NUMBER, sectionNumber);
            fragment.setArguments(args);
            return fragment;
        }

        @Override
        public View onCreateView(LayoutInflater inflater, ViewGroup
container,
                                 Bundle savedInstanceState) {
            View rootView =
inflater.inflate(R.layout.fragment_food_order_view, container, false);
            unbinder = ButterKnife.bind(this, rootView);

            LinearLayoutManager llm = new
LinearLayoutManager(getActivity());
            llm.setOrientation(LinearLayoutManager.VERTICAL);
            foodOrderList.setLayoutManager(llm);

            foods = new ArrayList<Food>();
            for (int i = 0; i < 15; i++) {
                Food food1 = new Food("干锅包菜" + i % 10, "2" + i % 10,
R.drawable.food5, "这是一道热菜", 1, false);
```

```
                foods.add(food1);
            }

            try{
                //9. 从MainScreen传过来的User变量
                user = (User)
getActivity().getIntent().getSerializableExtra("foodOrderViewUser");
            } catch (Exception e){
            }

            if(user == null){
                try{
                    //9. 从FoodView传过来的User变量
                    user = (User)
getActivity().getIntent().getSerializableExtra("foodViewUser");
                } catch (Exception e){
                }
            }

            switch (getArguments().getInt(ARG_SECTION_NUMBER)) {
                case 1:
                    orderSubmit.setText(R.string.order_submit);
                    foodOrderAdapter = new
FoodOrderRvAdapter(container.getContext(), foods, false);
                    foodOrderList.setAdapter(foodOrderAdapter);
                    break;
                case 2:
                    orderSubmit.setText(R.string.order_pay);
                    //12. 判断user.getOldUser()来使用Toast提示
                    orderSubmit.setOnClickListener(new
View.OnClickListener() {
                        @Override
                        public void onClick(View v) {
                            if(user != null){
                                if(user.getOldUser())
```

```
                                    Toast.makeText(v.getContext(), "您好，老
顾客，本次你可享受 7 折优惠", Toast.LENGTH_SHORT).show();
                            }
                        }
                    });
                    foodOrderAdapter = new
FoodOrderRvAdapter(container.getContext(), foods, true);
                    foodOrderList.setAdapter(foodOrderAdapter);
                    break;
            }

            return rootView;
        }

        @Override
        public void onDestroyView() {
            super.onDestroyView();
            unbinder.unbind();
        }
    }

    /**
     * A {@link FragmentPagerAdapter} that returns a fragment
corresponding to
     * one of the sections/tabs/pages.
     */
    public class SectionsPagerAdapter extends FragmentPagerAdapter {

        public SectionsPagerAdapter(FragmentManager fm) {
            super(fm);
        }

        @Override
        public Fragment getItem(int position) {
                    return PlaceholderFragment.newInstance(position + 1);
        }

        @Override
        public int getCount() {
            return 2;
        }

        @Override
        public CharSequence getPageTitle(int position) {
            switch (position) {
                case 0:
                    return "未下单菜";
                case 1:
                    return "已下单菜";
            }
            return null;
        }
    }
}
```

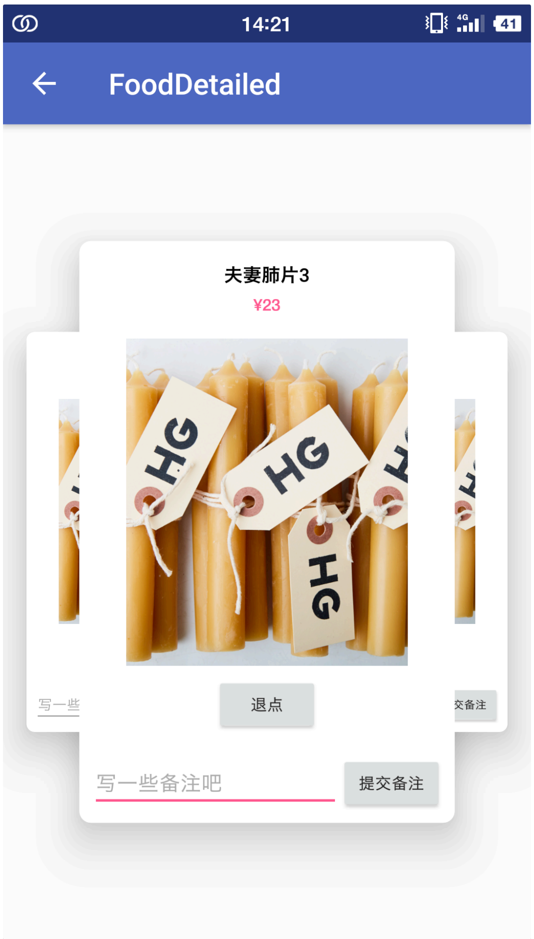
FoodOrderView 绑定的 Adapter

```
public class FoodOrderRvAdapter extends
RecyclerView.Adapter<FoodOrderRvAdapter.MyViewHolder> {
    private Context context;
    private ArrayList<Food> foods;
    private Boolean isInOrderList;

    public FoodOrderRvAdapter(Context context, ArrayList<Food> foods,
Boolean isInOrderList){
        this.context = context;
        this.foods = foods;
        this.isInOrderList = isInOrderList;
    }
    @Override
    public FoodOrderRvAdapter.MyViewHolder onCreateViewHolder(ViewGroup parent, int viewType) {
        View itemView;

        if(isInOrderList) {
            itemView =
LayoutInflater.from(context).inflate(R.layout.food_order_rvright_layout, parent, false);
        } else {
            itemView =
LayoutInflater.from(context).inflate(R.layout.food_order_rvleft_layout, parent, false);
        }

        MyViewHolder myViewHolder = new MyViewHolder(itemView);
        return myViewHolder;
    }

    @Override
    public void onBindViewHolder(final MyViewHolder holder, final int position) {

        if(isInOrderList){

holder.foodNameRight.setText(foods.get(position).getFoodname());

holder.foodPriceRight.setText(foods.get(position).getFoodprice());

holder.foodNumberRight.setText(String.valueOf(foods.get(position).getNumber()));

holder.foodCommentRight.setText(foods.get(position).getComment());
        } else {
```

```
holder.foodNameLeft.setText(foods.get(position).getFoodname());

holder.foodPriceLeft.setText(foods.get(position).getFoodprice());

holder.foodNumberLeft.setText(String.valueOf(foods.get(position).getNumber()));

holder.foodCommentLeft.setText(foods.get(position).getComment());

            holder.returnFood.setOnClickListener(new
View.OnClickListener() {
                @Override
                public void onClick(View v) {
                    if(!foods.get(position).getIsOrdered()){
                        Toast.makeText(context, "点菜成功",
Toast.LENGTH_SHORT).show();
                        foods.get(position).setIsOrdered(true);
                        holder.returnFood.setText(R.string.food_return);
                    } else {
                        Toast.makeText(context, "退点成功",
Toast.LENGTH_SHORT).show();
                        foods.get(position).setIsOrdered(false);

holder.returnFood.setText(R.string.food_ordering);
                    }
                }
            });
        }
    }

    @Override
    public int getItemCount() {
        return foods.size();
    }
    public class MyViewHolder extends RecyclerView.ViewHolder {
        TextView foodNameLeft, foodPriceLeft, foodNumberLeft,
foodCommentLeft;
        Button returnFood;

        TextView foodNameRight, foodPriceRight, foodNumberRight,
foodCommentRight;

        public MyViewHolder(View itemView) {
            super(itemView);

            if(isInOrderList){
                foodNameRight =
itemView.findViewById(R.id.food_name_right);
                foodPriceRight =
itemView.findViewById(R.id.food_price_right);
                foodNumberRight =
itemView.findViewById(R.id.food_number_right);
                foodCommentRight =
itemView.findViewById(R.id.food_comment_right);
            } else {
                foodNameLeft =
itemView.findViewById(R.id.food_name_left);
                foodPriceLeft =
itemView.findViewById(R.id.food_price_left);
                foodNumberLeft =
itemView.findViewById(R.id.food_number_left);
                foodCommentLeft =
itemView.findViewById(R.id.food_comment_left);
                returnFood = itemView.findViewById(R.id.return_food);
            }
        }
    }
}
```

8. 在 SCOS 的 es.source.code.activity 包下新建 Activity 类名为：FoodOrderView，为该 FoodOrderView 定制布局 food_order_view.xml

这里用到了一个第三方的库 infinitecycleviewpager，显示如下：



```
public class FoodDetailed extends AppCompatActivity{
    @BindView(R.id.toolbar)
    Toolbar toolbar;
    @BindView(R.id.item_picker)
    HorizontalInfiniteCycleViewPager itemPicker;

    private ArrayList<Food> foods;
    private int position;
    private ImagePageAdapter imagePageAdapter;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.food_detailed);
        ButterKnife.bind(this);

        setSupportActionBar(toolbar);
        getSupportActionBar().setDisplayHomeAsUpEnabled(true);

        //获取到从 FoodView 传过来的数据源和点击的 item 位置
        foods = (ArrayList<Food>)
getIntent().getSerializableExtra("foods");
        position = getIntent().getIntExtra("position", 0);

        imagePageAdapter = new ImagePageAdapter(this, foods);
        itemPicker.setAdapter(imagePageAdapter);

        //设置 viewPager 显示传过来的 item 位置
        itemPicker.setCurrentItem(position);
    }
}
```

FoodDetailed 绑定的 Adapter

```
public class ImagePageAdapter extends PagerAdapter {
    private Context context;
    private ArrayList<Food> foods;

    public ImagePageAdapter(Context context, ArrayList<Food> foods) {
        this.context = context;
        this.foods = foods;
    }

    @Override
    public int getCount() {
        return foods.size();
    }
    @Override
    public int getItemPosition(Object object) {
        return POSITION_NONE;
    }

    @Override
    public Object instantiateItem(ViewGroup container, final int position)
{
        View view =
LayoutInflater.from(context).inflate(R.layout.item_image, container,
false);

        TextView itemName = view.findViewById(R.id.item_name);
        TextView itemPrice = view.findViewById(R.id.item_price);
        ImageView foodImage = view.findViewById(R.id.food_image);
        final Button orderStatus = view.findViewById(R.id.order_status);
        final EditText editComment =
view.findViewById(R.id.edit_comment);
        Button submitComment = view.findViewById(R.id.submit_comment);

        itemName.setText(foods.get(position).getFoodname());
        itemPrice.setText("¥" + foods.get(position).getFoodprice());
        Picasso.with(context)
                .load(foods.get(position).getImageurl())
                .fit()
                .centerCrop()
                .into(foodImage);

        //判断点菜按钮是否已点菜
        if(!foods.get(position).getIsOrdered()){
            orderStatus.setText(R.string.food_ordering);
        } else {
            orderStatus.setText(R.string.food_return);
        }
        orderStatus.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                if(!foods.get(position).getIsOrdered()){
                    Toast.makeText(context, "点菜成功",
Toast.LENGTH_SHORT).show();
                    foods.get(position).setIsOrdered(true);
                    orderStatus.setText(R.string.food_return);
                } else {
                    Toast.makeText(context, "退点成功",
Toast.LENGTH_SHORT).show();
                    foods.get(position).setIsOrdered(false);
                    orderStatus.setText(R.string.food_ordering);
                }
            }
        });

        submitComment.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                if(editComment.getText().toString() != null){

foods.get(position).setComment(editComment.getText().toString());
                } else {
                    Toast.makeText(context, "写一些备注吧！",
Toast.LENGTH_SHORT).show();
                }
            }
        });

        container.addView(view);

        return view;
    }

    @Override
    public boolean isViewFromObject(View view, Object object) {
        return view.equals(object);
    }

    @Override
    public void destroyItem(ViewGroup container, int position, Object object) {
        container.removeView((View) object);
    }
}
```

关于 9、10、11、12 功能的实现已在上述代码中注释出来。

4.结论

1. 在给 FoodOrderView 中 Fragment 的布局文件添加底部的 LinearLayout 时，要注意去掉通过 Android Studio 构造的 Tabbed Activity 的布局文件中 toolbar 的 layout_scrollFlags 属性，不然底部的 LinearLayout 只有在上滑 toolbar 消失时才会出现。

2. 我暂时只给 FoodOrderView 设置了一个 Fragment 的布局文件，通过判断当前在哪个 Fragment，来让底部的按钮显示成“提交订单”和“结账”，并处理对应事件。

3. 由于 FoodOrderView 两个 Fragment 对应 RecyclerView 的 item 布局不一样，这里在 Fragment 构造 Adapter 时传入一个 Boolean 值，在 Adapter 根据那个 Boolean 值来加载对应的 item 布局。

4. 在 FoodView 点击菜单项“已点菜品”或“查看订单”时传入 0 或 1，在 FoodOrderView 中获取到对应的值，通过 ViewPager 的 setCurrentItem 方法来设置启动的 Fragment。

5. 在 FoodOrderView 的 Fragment 中处理 user 时要注意可能会从 MainScreen 和 FoodView 中都获取到 user，在获取到一个后要判断 user 是否为空再获取第二个，否则会出错。

5.参考文献

1. GridView 的官方文档详细介绍 https://developer.android.google.cn/reference/android/widget/GridView.html

2. TabLayout 的官方文档详细介绍 https://developer.android.google.cn/reference/android/support/design/widget/TabLayout.html

3. RecyclerView 为每个 item 添加点击事件 http://www.jcodecraeer.com/plus/view.php?aid=7881

4. 详细分析 AppBarLayout 的五种 ScrollFlags http://www.jianshu.com/p/7caa5f4f49bd

5. Github 上的 InfiniteCycleViewPager 库及使用说明 https://github.com/Devlight/InfiniteCycleViewPager