

# 移动平台开发实验报告

题    目： 水来了

班    级： 计科151/152

学生姓名： 王泽宇    学    号： 1500170043

学生姓名： 童巧琳    学    号： 1500170033

学生姓名： 黄珊珊    学    号： 1500170086

学生姓名： 刘禾子    学    号： 1500170082

院    系： 计算机科学与技术学院

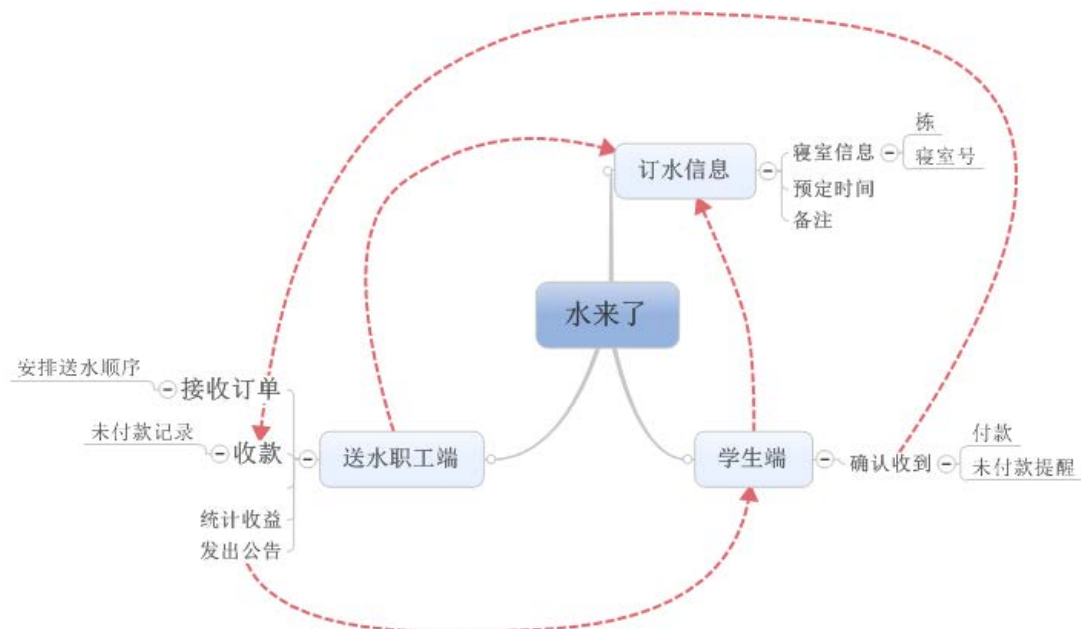
专业年级： 计算机科学与技术15级

# 一、功能概述

## 1、系统概述

随着现今互联网时代的高速发展、社会的进步，越来越多的人选择无现金支付，不仅因为它的便捷，还有它的省时高效，大多数高校中如果住的楼层较高的同学订水是十分麻烦的，为了便捷师生，该款送水 APP 可作为中介物，方便校园生活，减少不便。

## 2、功能规划



如上图所示，共分为两端：学生端与员工端，学生端可查看自己对应的订单（包括寝室号，时间），可与员工端共享公告表，但不能对公告表进行操作。

员工端，可发布公告，对公告进行删除操作，可以根据时间（月日）统计收益，可以查看所有的订单列表，按照最新订单排列。在员工端个人中心，可以对待办订单中的待付款订单和已完成订单进行操作，待付款订单加入待付款订单表，待付款订单表中已完成的订单加入到已完成订单中。

### 3、分组分工

组长：刘禾子，负责总体布局框架，各模块之间的接口关系、数据库以及“发布公告”

功能模块；

组员：王泽宇，负责学生端提交订单后，将数据插入数据库，并用 ListView 显示出来。

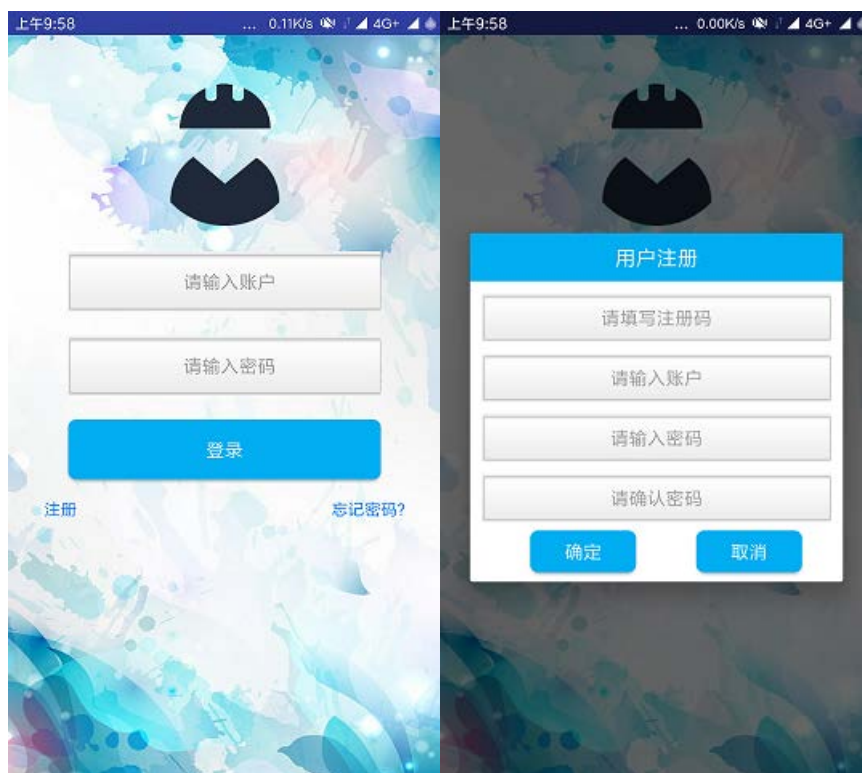
组员：童巧琳，负责员工端订单业务

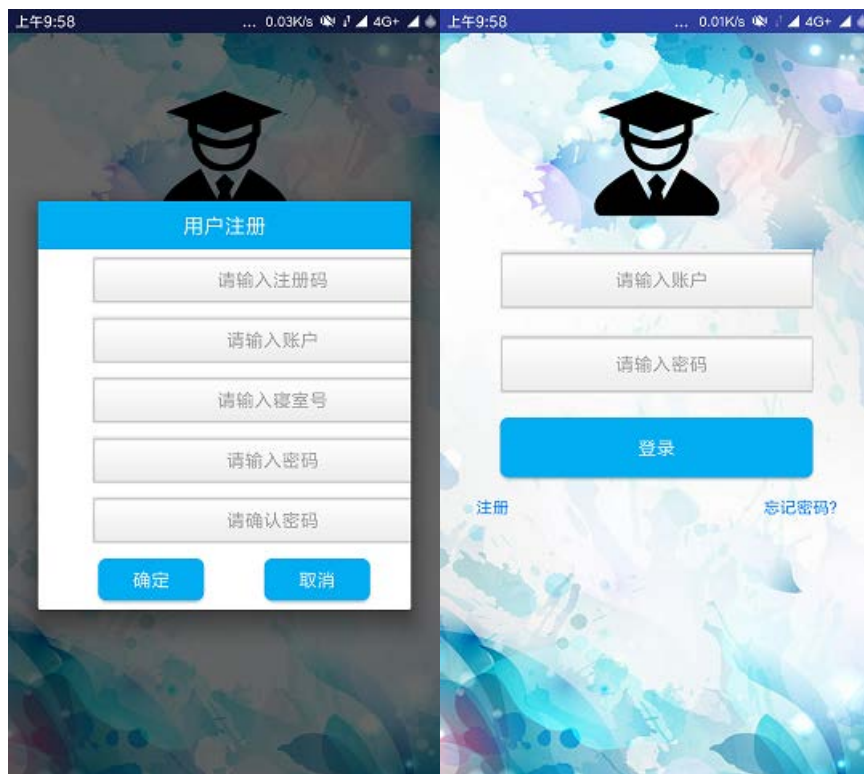
组员：黄珊珊，负责收益查询

## 二、系统实现效果

用户注册：

分为两端：员工端和学生端





通过用户对自己身份的定位进行登陆，用特定的注册码进行验证注册，设置密码采用二次确认防止用户输入错误，完成注册就向数据库中员工表和员工表插入相应信息，并有部分提示信息，如部分输入框内容不能为空以及注册成功提示，增强用户体验。

**发布公告：**由员工端发布公告与删除公告，学生端只能查看公告，不能对公告进行操作



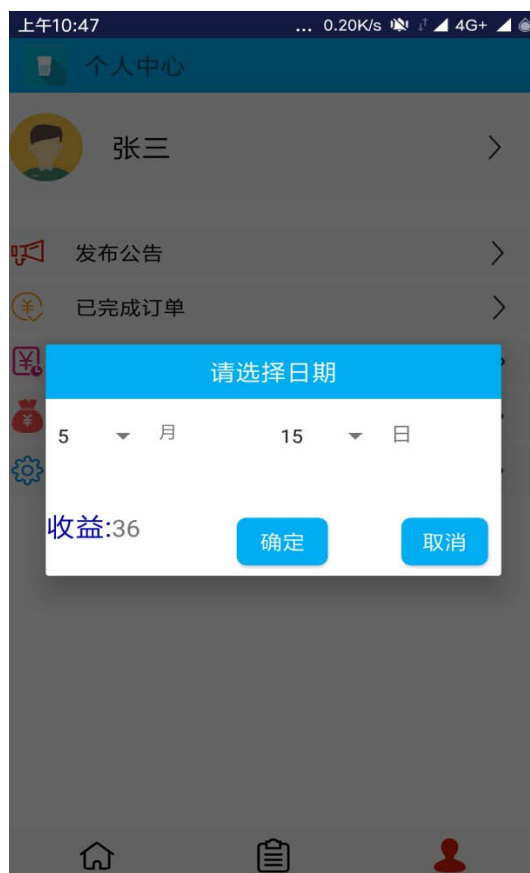
学生端界面->

该模块包含一个公告表 (id, 标题, 内容, 时间), 其中时间从系统自动获取时间, 保证公告的实时性, 其他操作均是对表的插入以及删除操作。

**学生端订单业务:** 学生点击“提交订单”后这个学生的寝室号, 订水日期插入到订单数据库, 然后通过 listView 将其订水日期显示出来。



**收益查询：**通过查询总的订单数据库表，按照输入月日的信息，与数据库信息比对，再计算订单总额，并显示。



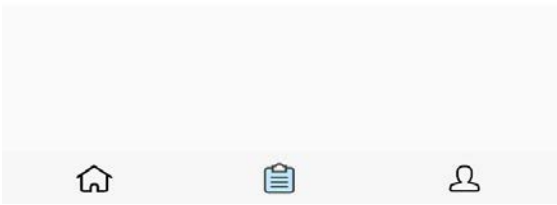
左图为收益界面。  
数据库已完成订单有四笔，因此收益为 36。当查询日期对应无订单时，收益显示为 0。

员工端订单业务：

1. 显示所有订单：



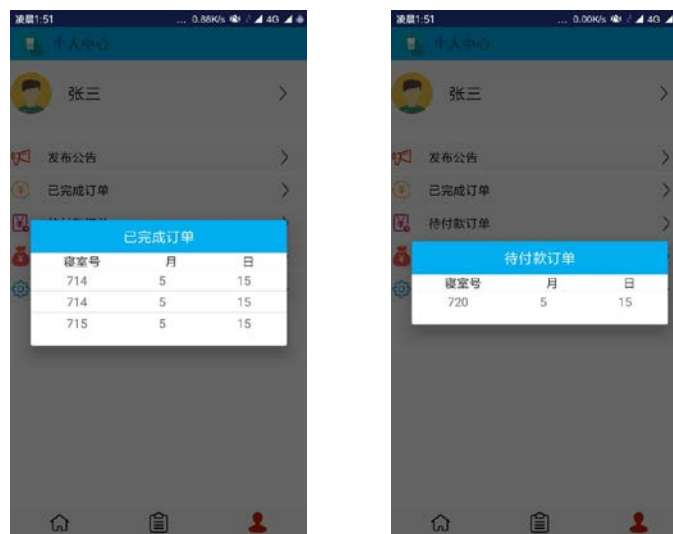
寝室号	月	日
714	5	16
714	5	16
714	5	16
714	5	16
714	5	16



2. 对上图订单的 item 点击，分类：



3. 在以下的已完成订单和待付款订单可查看分类后的结果：



### 三、核心代码说明

#### 1、登陆注册与发布公告功能、Fragment 框架(刘禾子负责)

登录注册：

学生板块与员工板块类似，在此只列出员工板块的部分核心代码

```
import android.app.Activity;
import android.app.AlertDialog;
import android.app.Dialog;
import android.content.Intent;
import android.database.Cursor;
import android.database.sqlite.SQLiteDatabase;
import android.os.Bundle;
import android.support.annotation.Nullable;
import android.view.LayoutInflater;
import android.view.View;
import android.view.Window;
import android.widget.Button;
import android.widget.EditText;
import android.widget.TextView;
import android.widget.Toast;
import tool.MyOpenHelper;
public class rider_login_activity extends Activity {
    private EditText name;//用户名
    private EditText password;//用户密码
    private Button login;//登陆按钮
    private TextView register;//注册
    private TextView forgetNum;//忘记密码
```



```

private MyOpenHelper dbHelper;
Dialog dialog1;//用于生成注册对话框以及返回activity
@Override
protected void onCreate(@Nullable Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    requestWindowFeature(Window.FEATURE_NO_TITLE);//设置Activity
    的无标题显示
    setContentView(R.layout.rider_login_layout);
    dbHelper=MyOpenHelper.getInstance(this);//获取数据库当前状态
    //获取控件
    name= (EditText)
    findViewById(R.id.rider_login_activity_name_input);
    password= (EditText)
    findViewById(R.id.rider_login_activity_password_input);
    login= (Button)
    findViewById(R.id.rider_login_activity_login);
    register= (TextView)
    findViewById(R.id.admin_login_activity_register);
    forgetNum= (TextView)
    findViewById(R.id.admin_login_activity_forgetNum);
    //跳转到登陆过的送水工界面
    login.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) {
            String nameInfo=name.getText().toString();
            String passwordInfo=password.getText().toString();
            //从数据库中获取密码并判断是否相同
            SQLiteDatabase db=dbHelper.getWritableDatabase();
            Cursor cursor=db.rawQuery("select password from rider
where name=?",new String[]{nameInfo});
            String pi=null;
            if (cursor.moveToNext()){
                pi=cursor.getString(cursor.getColumnIndex("password"));
            }
            //密码正确跳转到登陆后的界面
            if(passwordInfo.equals(pi)){
                Intent intent=new
                Intent(rider_login_activity.this,rider_activity.class);
                startActivity(intent);
                cursor.close();
            }else{
                //弹出错误提示
                Toast.makeText(rider_login_activity.this,"用户名或

```

```

        密码错误", Toast.LENGTH_SHORT).show();
    }
}
});
//自定义AlertDialog 用于注册
register.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        //构建对话框
        AlertDialog.Builder builder = new
AlertDialog.Builder(rider_login_activity.this);
        //解析对话框布局
        LayoutInflater factory =
LayoutInflater.from(rider_login_activity.this);
        final View textEntryView =
factory.inflate(R.layout.register, null);
        //设置对话框布局
        builder.setView(textEntryView);
        //获取对话框布局控件
        final Button rider_postive= (Button)
textEntryView.findViewById(R.id.rider_postive);
        final Button rider_negative= (Button)
textEntryView.findViewById(R.id.rider_negative);
        final EditText code = (EditText)
textEntryView.findViewById(R.id.admin_register_info);
        final EditText name = (EditText)
textEntryView.findViewById(R.id.admin_register_name);
        final EditText firstPassword = (EditText)
textEntryView.findViewById(R.id.admin_register_first_password);
        final EditText secondPassword = (EditText)
textEntryView.findViewById(R.id.admin_register_second_password);
        //取消按钮点击事件
        rider_negative.setOnClickListener(new
View.OnClickListener() {
            @Override
            public void onClick(View v) {
                dialog1.dismiss();//对话框消失
            }
        });
        //确定按钮点击事件
        rider_postive.setOnClickListener(new
View.OnClickListener() {
            @Override
            public void onClick(View v) {

```

```

        String codeInfo = code.getText().toString();
        //注册码要为10086
        if (codeInfo.equals("10086")) {
            String nameInfo =
name.getText().toString();
            String firstPasswordInfo =
firstPassword.getText().toString();
            String secondPasswordInfo =
secondPassword.getText().toString();
            SQLiteDatabase db =
dbHelper.getWritableDatabase();
            //检测密码是否为6个数字
            if (firstPasswordInfo.matches("[0-9]{6}"))
{
                // 两次密码是否相同
                if
(firstPasswordInfo.equals(secondPasswordInfo)) {
                    Cursor cursor = db.rawQuery("select
name from rider where name=?", new String[]{nameInfo});
                    //用户是否存在
                    if (cursor.moveToNext()) {

Toast.makeText(rider_login_activity.this, "该用户已存在",
Toast.LENGTH_SHORT).show();

                    } else {
                        db.execSQL("insert into
rider(name,password)values(?,?)", new String[]{nameInfo,
firstPasswordInfo});

                        //弹出注册成功提示, 并将数据插入员
工表

Toast.makeText(rider_login_activity.this, "注册成功
", Toast.LENGTH_LONG).show();

                        dialog1.dismiss();
                    }
                } else {
                    //弹出错误提示

Toast.makeText(rider_login_activity.this, "两次密码不相同",
Toast.LENGTH_SHORT).show();

                }
            } else {
                //弹出错误提示

```

```

Toast.makeText(rider_login_activity.this, "密码为 6 位纯数字",
Toast.LENGTH_SHORT).show();
    }
    } else {
        //弹出错误提示
        Toast.makeText(rider_login_activity.this, "
注册码错误", Toast.LENGTH_SHORT).show();
    }
}
});
dialog1=builder.create();//创建对话框
dialog1.show();//显示对话框
}
});
forgetNum.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        Toast.makeText(rider_login_activity.this,"此功能暂不支
持",Toast.LENGTH_SHORT).show();
    }
});
}
}
}

```

员工端发布公告及删除：

```

public void onActivityCreated(Bundle savedInstanceState) {
    super.onActivityCreated(savedInstanceState);
    LinearLayout ll =
    (LinearLayout) getActivity().findViewById(R.id.rider_my_info);//发布公告
    信息选项
    LinearLayout ll1= (LinearLayout)
    getActivity().findViewById(R.id.rider_my_paid);//查看已完成订单选项
    LinearLayout ll2= (LinearLayout)
    getActivity().findViewById(R.id.rider_my_unpaid);//查看未完成订单选项
    LinearLayout ll3= (LinearLayout)
    getActivity().findViewById(R.id.rider_my_income);//查询收益
    ll.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) {
            //获得数据库当前版本
            dbHelper = MyOpenHelper.getInstance(getActivity());
            //创建对话框进行进一步操作
            AlertDialog.Builder builder = new
            AlertDialog.Builder(getActivity());
            //获得反射器

```

```

        LayoutInflater factory =
LayoutInflater.from(getActivity());
        //解析对话框布局 View
        final View textEntryView =
factory.inflate(R.layout.rider_info_editor, null);
        //设置对话框布局
        builder.setView(textEntryView);
        //绑定对话框相应控件
        final EditText title = (EditText)
textEntryView.findViewById(R.id.rider_info_title);
        final EditText content = (EditText)
textEntryView.findViewById(R.id.rider_info_content);
        final Button positive = (Button)
textEntryView.findViewById(R.id.rider_info_postive);
        final Button negative = (Button)
textEntryView.findViewById(R.id.rider_info_negative);
        //取消按钮点击事件
        negative.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                dialog.dismiss();
            }
        });
        //确定按钮点击事件
        positive.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                //得到编辑框输入的标题
                String out_title = title.getText().toString();
                //得到编辑框输入的内容
                String out_content = content.getText().toString();
                //获取数据库写权限
                SQLiteDatabase db =
dbHelper.getWritableDatabase();
                //两个编辑框不为空
                if (out_content != null && out_title != null) {
                    //从系统获取当前时间
                    Time time = new Time("GMT+8");
                    time.setToNow();
                    int year = time.year;
                    int month = time.month;
                    int dat = time.monthDay;
                    int minute = time.minute;
                    int hour = time.hour;

```

```

//分别取出年月日时分秒方便公告栏显示
String show_time = year + "年" + month + "月" +
dat + "日" + hour + "时" + minute + "分";
//将相应数据插入公告表
db.execSQL("insert into
info(title,content,time)values(?,?,?)", new String[]{out_title,
out_content, show_time});
//弹出操作成功提示
Toast.makeText(getActivity(), "发布成功!",
Toast.LENGTH_LONG).show();
//对话框消失
dialog.dismiss();
}
}
});
dialog = builder.create();//创建对话框
dialog.show();//显示对话框
dbHelper.close();//关闭数据库
}
});

```

公告表的项点击事件:

```

list_info.setOnItemClickListener(new
AdapterView.OnItemClickListener() {
@Override
public void onItemClick(AdapterView<?> parent, View view, int
position, long id) {
//创建对话框
final AlertDialog.Builder builder=new
AlertDialog.Builder(getActivity());
//解析布局文件
LayoutInflater factory=LayoutInflater.from(getActivity());
final View
textEntryView=factory.inflate(R.layout.rider_info_delete,null);
//设置对话框布局
builder.setView(textEntryView);
//获取对话框布局中的按钮控件
final Button delete= (Button)
textEntryView.findViewById(R.id.rider_info_delete);
//删除按钮点击事件
delete.setOnClickListener(new View.OnClickListener() {
@Override
public void onClick(View v) {
//通过游标获得当前点击的项Id
String

```

```

id=cursor.getString(cursor.getColumnIndex("_id"));
        // 获得数据库写操作
        SQLiteDatabase db=dbHelper.getWritableDatabase();
        // 返回删除的行数
        int line=db.delete("info","_id=?",new String[]{id});
        // 判断是否删除成功
        if (line>0){
            Toast.makeText(getActivity(),"数据删除成功",Toast.LENGTH_LONG).show();
        }
        display_Info();// 重新显示公告表项
        db.close();// 关闭数据库
        dialog.dismiss();// 关闭对话框
    }
});
dialog=builder.create();// 创建对话框
dialog.show();// 对话框显示
}
});

```

Fragment 大体框架：在此只列出员工端，学生端与之类似

```

import android.app.Activity;
import android.app.FragmentManager;
import android.app.FragmentTransaction;
import android.os.Bundle;
import android.support.annotation.Nullable;
import android.view.View;
import android.view.Window;
import android.widget.ImageView;
/**
 * Created by 78914 on 2017/6/12.
 */
public class rider_activity extends Activity implements
View.OnClickListener{
    FragmentManager manager;// 获取manager 对Fragment 布局文件进行操作
    FragmentTransaction ft;// 获取操作类型
    Fragment_home_rider fragment_home;// 三个Fragment 类对象
    Fragment_order_rider fragment_order;
    Fragment_my_rider fragment_my;
    // 用于获取图片按钮资源
    ImageView home;
    ImageView order;
    ImageView my;
    @Override
    protected void onCreate(@Nullable Bundle savedInstanceState) {

```

```

super.onCreate(savedInstanceState);
requestWindowFeature(Window.FEATURE_NO_TITLE);
setContentView(R.layout.rider_layout);
//控件绑定 ID
bindID();
//获取manager 对象
manager=getFragmentManager();
//开始事务
ft=manager.beginTransaction();
if (fragment_home==null){
    fragment_home=new Fragment_home_rider();
}
//给容器添加Fragment 布局
ft.add(R.id.frame_content_rider,fragment_home);
//提交事务
ft.commit();
}
private void bindID() {
    //获取住布局三个底部图片按钮
    home= (ImageView) findViewById(R.id.btnHome_rider);
    order= (ImageView) findViewById(R.id.btnOrder_rider);
    my= (ImageView) findViewById(R.id.btnMy_rider);
    //三个按钮点击事件
    home.setOnClickListener(this);
    order.setOnClickListener(this);
    my.setOnClickListener(this);
}
//点击事件具体实现方法
@Override
public void onClick(View v) {
    //开始事务
    ft=manager.beginTransaction();
    if (fragment_home!=null){
        //隐藏Fragment
        ft.hide(fragment_home);
    }
    if (fragment_order!=null){
        ft.hide(fragment_order);
        //隐藏Fragment
    }
    if (fragment_my!=null){
        ft.hide(fragment_my);
        //隐藏Fragment
    }
}

```



```

//通过控件id 来进行操作
switch (v.getId()){
    case R.id.btnHome_rider:
        //单例模式
        if (fragment_home==null){
            fragment_home=new Fragment_home_rider();
            ft.add(R.id.frame_content_rider,fragment_home);
        }else {
            //显示
            ft.show(fragment_home);
            //点击后图片切换效果
            home.setImageResource(R.drawable.home_2);
            order.setImageResource(R.drawable.order_1);
            my.setImageResource(R.drawable.my_1);
        }
        break;
    case R.id.btnOrder_rider:
        //单例模式
        if (fragment_order==null){
            fragment_order=new Fragment_order_rider();
            ft.add(R.id.frame_content_rider,fragment_order);
        }else {
            //显示
            ft.show(fragment_order);
            //点击后图片切换效果
            home.setImageResource(R.drawable.home_1);
            order.setImageResource(R.drawable.order_2);
            my.setImageResource(R.drawable.my_1);
        }
        break;
    case R.id.btnMy_rider:
        //单例模式
        if (fragment_my==null){
            fragment_my=new Fragment_my_rider();
            ft.add(R.id.frame_content_rider,fragment_my);
        }else {
            //显示
            ft.show(fragment_my);
            //点击后图片切换效果
            home.setImageResource(R.drawable.home_1);
            order.setImageResource(R.drawable.order_1);
            my.setImageResource(R.drawable.my_2);
        }
        break;

```

```

        default:
            break;
    }
    //提交事务
    ft.commit();
}
}

```

Fragment\_home\_rider:主要用到 onCreateView 方法解析视图:

```

public View onCreateView(LayoutInflater inflater, ViewGroup
container, Bundle savedInstanceState) {
    return inflater.inflate(layout.fragment_home_rider, container,
false);
}

```

Fragment\_order\_rider:用到同样方法:

```

public View onCreateView(LayoutInflater inflater, ViewGroup
container, Bundle savedInstanceState) {
    return inflater.inflate(R.layout.fragment_order_rider, container,
false);
}

```

Fragment\_my\_rider:也同样用到该方法:

```

public View onCreateView(LayoutInflater inflater, ViewGroup
container, Bundle savedInstanceState) {
    return inflater.inflate(R.layout.fragment_my_rider, container,
false);
}

```

## 2、学生端订单业务功能（王泽宇负责）

```

3、 package cqpt.ui.watercoming;

import android.app.Fragment;
import android.database.Cursor;
import android.database.sqlite.SQLiteDatabase;
import android.os.Bundle;
import android.support.annotation.Nullable;
import android.text.format.Time;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.Button;
import android.widget.ListView;
import android.widget.SimpleCursorAdapter;

import tool.MyOpenHelper;

```

```

/**
 * Created by 78914 on 2017/6/12.
 */
public class Fragment_order extends Fragment {
    /* 声明全局变量 data,
    num,month,day,cursor,list_order,dbHelper
    *
    */
    private String data;
    private String num;
    private String month;
    private String day;
    private Cursor cursor;
    private ListView list_order;
    private MyOpenHelper dbHelper;
    public View onCreateView(LayoutInflater inflater, ViewGroup
container, Bundle savedInstanceState){
        //data,num 的数据来源
        Bundle bundle = getArguments();
        data=bundle.getString("DATA");
        num=bundle.getString("DATA2");//得到相应数据
        //关联布局界面
        return
inflater.inflate(R.layout.student_order_layout,container,false);
    }
    @Override
    public void onActivityCreated(@Nullable Bundle
savedInstanceState) {
        super.onActivityCreated(savedInstanceState);
        display_order();
        //声明关联Button
        Button order= (Button)
getActivity().findViewById(R.id.order_bt);
        //设置监听器
        order.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                SQLiteDatabase db =
dbHelper.getWritableDatabase();
                Time time = new Time("GMT+8");
                //获取当前时间
                time.setToNow();
                //将获取到的时间插入到订单数据库中
                month = String.valueOf(time.month);

```

```

        day = String.valueOf(time.monthDay);
        //执行插入数据
        db.execSQL("insert into
orderlist(RoomNum,month,day)values(?,?,?)", new String[]{num,
month, day});
        display_order();
    }
});
}
private void display_order() {
    //声明关联ListView
    list_order= (ListView)
getActivity().findViewById(R.id.order_list);
    dbHelper = MyOpenHelper.getInstance(getActivity());
    //利用cursor 查询插入的数据的时间，寝室号，并在ListView 中显
示出来

    cursor=dbHelper.getReadableDatabase().query("orderlist",null,"Ro
omNum=?",new String[]{num},null,null,null);
    if (cursor.moveToNext()){
        SimpleCursorAdapter adapter=new
SimpleCursorAdapter(getActivity(),R.layout.order_listitem,cursor
,
        new String[]{"RoomNum","month","day"},new
int[]{R.id.order_num,R.id.order_month,R.id.order_day});
        list_order.setAdapter(adapter);
    }
    dbHelper.close();//关闭表
}
}
}

```

### 3、收益查询功能（黄珊珊负责）

```

package cqpt.ui.watercoming;

import android.app.AlertDialog;
import android.app.Dialog;
import android.app.Fragment;
import android.database.Cursor;
import android.database.sqlite.SQLiteDatabase;
import android.os.Bundle;
import android.text.format.Time;
import android.view.LayoutInflater;
import android.view.View;

```

```

import android.view.ViewGroup;
import android.widget.AdapterView;
import android.widget.ArrayAdapter;
import android.widget.Button;
import android.widget.EditText;
import android.widget.LinearLayout;
import android.widget.ListView;
import android.widget.SimpleCursorAdapter;
import android.widget.Spinner;
import android.widget.TextView;
import android.widget.Toast;

import tool.MyOpenHelper;
private MyOpenHelper dbHelper; // 获取数据库
Dialog dialog5; // 对话框对象
private Cursor cursor; // 数据库查找结果存放对象
private int i=0; // 存放结果集的个数
private String S1=null; // spinner 空间选中“月”中 item 的值
private String S2=null; // spinner 空间选中“日”中 item 的值
// 按键的监听事件
    l13.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) { // 当点击按键时回调的函数
            AlertDialog.Builder builder4 = new
AlertDialog.Builder(getActivity());
            LayoutInflater factory =
LayoutInflater.from(getActivity());
            // 搭建对话框，用于显示收益板块内容
            final View textEntryView =
factory.inflate(R.layout.rider_income, null);
            builder4.setView(textEntryView);
            dialog5=builder4.create(); // 构建对话框
            dialog5.show(); // 显示对话框
            // 将个控件加入，包括：两个 spinner、两个 button、一个
textView
            final Spinner sp1= (Spinner)
textEntryView.findViewById(R.id.sp1);
            final Spinner sp2= (Spinner)
textEntryView.findViewById(R.id.sp2);
            final TextView tv= (TextView)
textEntryView.findViewById(R.id.money);
            final Button btn_sure= (Button)
textEntryView.findViewById(R.id.rider_income_postive);
            final Button btn_not= (Button)

```

```

textView.findViewById(R.id.rider_income_negative);
    //将xml 数据转换为具体的数据数组
    final String[]
months=getResources().getStringArray(R.array.month);
    final String[]
days=getResources().getStringArray(R.array.day);

    ArrayAdapter<String> adapter1=new
ArrayAdapter<String>(getActivity(),
android.R.layout.simple_spinner_item,months);
    ArrayAdapter<String>adapter2=new
ArrayAdapter<String>(getActivity(),
android.R.layout.simple_spinner_item,days);
    //将适配器数据加载到控件
    sp1.setAdapter(adapter1);
    sp2.setAdapter(adapter2);
    //spinner 控件的监听
    sp1.setOnItemClickListener(new
AdapterView.OnItemClickListener() {
        @Override
        public void onItemClick(AdapterView<?> parent,
View view, int position, long id) {
            S1= (String) sp1.getSelectedItem();
            //获取 spinner 中被点击的 item 的内容
        }
        @Override
        public void onNothingSelected(AdapterView<?>
parent) {
        }
    });
    sp2.setOnItemClickListener(new
AdapterView.OnItemClickListener() {
        @Override
        public void onItemClick(AdapterView<?> parent,
View view, int position, long id) {
            S2= (String) sp2.getSelectedItem();
            //获取 spinner 中被点击的 item 的内容
        }
        @Override
        public void onNothingSelected(AdapterView<?>
parent) {
        }
    });
    //button 按键的监听

```

```

        btn_sure.setOnClickListener(new View.OnClickListener()
{
    @Override
    public void onClick(View v) {
        dbHelper =
MyOpenHelper.getInstance(getActivity());
        cursor =
dbHelper.getReadableDatabase().query("orderlistPaid", null, "month=?
and day=?", new String[]{S1,S2}, null, null, null, null);
        //根据括号内的要求查找数据库信息，并将结果放在结果
集中

        if (cursor!=null&&cursor.moveToFirst())
            do {
                i++; //通过循环求出结果集的数据个数
            }while (cursor.moveToNext());
        //由cursor 中结果个数乘以水的单价9 元得到收益
        String money= String.valueOf(i*9);
        tv.setText(money);
        dbHelper.close();
        cursor.close();
        i=0;
    }
});
        btn_not.setOnClickListener(new View.OnClickListener()
{
    @Override
    public void onClick(View v) {
        dialog5.dismiss();
    }
});
    }
});
}
}
}

```

#### 4、员工端订单业务功能（童巧琳负责）

```

5、 package cqupt.ui.watercoming;

import android.app.AlertDialog;
import android.app.Dialog;
import android.app.Fragment;
import android.database.Cursor;
import android.database.sqlite.SQLiteDatabase;

```

```

import android.os.Bundle;
import android.support.annotation.Nullable;
import android.view.LayoutInflater;
import android.view.View;
import android.view.View.OnClickListener;
import android.view.ViewGroup;
import android.widget.AdapterView;
import android.widget.Button;
import android.widget.ListView;
import android.widget.SimpleCursorAdapter;
import android.widget.Toast;

import tool.MyOpenHelper;

/**
 * Created by 78914 on 2017/6/13.
 */
public class Fragment_order_rider extends Fragment {
    private Cursor cursor;
    private ListView list_order;
    private MyOpenHelper dbHelper;
    Dialog dialog;
    public View onCreateView(LayoutInflater inflater, ViewGroup
container, Bundle savedInstanceState) {
        return inflater.inflate(R.layout.fragment_order_rider,
container, false);
    }
    @Override
    public void onActivityCreated(@Nullable Bundle
savedInstanceState) {
        super.onActivityCreated(savedInstanceState);
        display_Order();//显示所有的待办订单，见页底
        //订单列表的点击事件
        list_order.setOnItemClickListener(new
AdapterView.OnItemClickListener() {
            @Override
            public void onItemClick(AdapterView<?> parent, View
view, int position, long id) {
                //创建对话框进一步操作
                final AlertDialog.Builder builder=new
AlertDialog.Builder(getActivity());
                //获得反射器
                LayoutInflater
factory=LayoutInflater.from(getActivity());
                //得到解析的View

```



```

        final View
textEntryView=factory.inflate(R.layout.rider_dispose_order,null)
;
        // 设置对话框布局
builder.setView(textEntryView);
        // 绑定布局控件
        final Button finish= (Button)
textEntryView.findViewById(R.id.rider_order_finish);
        final Button unpaid= (Button)
textEntryView.findViewById(R.id.rider_order_unpaid);
        // 已完成按钮点击事件
finish.setOnClickListener(new
View.OnClickListener() {
        @Override
        public void onClick(View v) {
                //用 cursor.getString ( ) 方法从数据库获取当前
item 指向的行的id
                String
id=cursor.getString(cursor.getColumnIndex("_id"));
                // 获取寝室号 RoomNum
                String
num=cursor.getString(cursor.getColumnIndex("RoomNum"));
                // 获取月 month
                String
mon=cursor.getString(cursor.getColumnIndex("month"));
                // 获取日 day
                String
day=cursor.getString(cursor.getColumnIndex("day"));
                // 打开数据库
                SQLiteDatabase
db=dbHelper.getWritableDatabase();
                // 向 orderlistPaid 表中插入当前行的数据
                // 从 orderlistPaid 表中删除当前行并返回删除的
行数
                db.execSQL("insert into
orderlistPaid(RoomNum,month,day)values(?,?,?)",new
String[]{num,mon,day});
                // 返回删除的行数
                int line=db.delete("orderlist","_id=?",new
String[]{id});
                if (line>0){
                        Toast.makeText(getActivity(),"已成功处
理该订单",Toast.LENGTH_LONG).show();
                }
                // 若删除成功则放出吐司

```

```

    }
    //从数据库中获取数据并显示的函数，见本页底
    display_Order();
    //使用完数据库，关闭
    db.close();
    //关闭对话框
    dialog.dismiss();
}
});
dialog=builder.create();//创建对话框
dialog.show();//显示对话框
//待支付按钮的点击事件监听
unpaid.setOnClickListener(new OnClickListener() {
    @Override
    public void onClick(View v) {
        String
        id=cursor.getString(cursor.getColumnIndex("_id"));
        String
        num=cursor.getString(cursor.getColumnIndex("RoomNum"));
        String
        mon=cursor.getString(cursor.getColumnIndex("month"));
        String
        day=cursor.getString(cursor.getColumnIndex("day"));
        //以上获取当前行在表中的对应内容并用变量存储
        SQLiteDatabase
        db=dbHelper.getWritableDatabase();
        //打开数据库
        //向orderlistUnPaid表中插入当前行的数据
        //从orderlistUnPaid表中删除当前行并返回删除
        的行数
        db.execSQL("insert into
        orderlistUnPaid(RoomNum,month,day)values(?,?,?)",new
        String[]{num,mon,day});
        //返回删除的行数
        int line=db.delete("orderlist","_id=?",new
        String[]{id});
        if (line>0){
            Toast.makeText(getActivity(),"已成功处
            理该订单",Toast.LENGTH_LONG).show();
            //若删除成功则放出吐司
        }
        display_Order();//从数据库中获取数据并显示的
        函数，见本页底
        db.close();//使用完数据库，关闭
    }
}

```

```

        dialog.dismiss();//关闭对话框
    }
    });
}
});
}
private void display_Order() {
    //获取listview
    list_order = (ListView)
    getActivity().findViewById(R.id.all_order_list);
    //引用数据库
    dbHelper = MyOpenHelper.getInstance(getActivity());
    cursor =
    dbHelper.getReadableDatabase().query("orderlist", null, null,
    null, null, null, null, null);
    //打开数据库并读取 orderlist 表中的所有内容存入 cursor
    if (cursor.moveToNext()) {
        //用SimpleCursorAdapter 适配器将 cursor 的内容一条一条的
        显示在listview 的item 上
        SimpleCursorAdapter adapter = new
        SimpleCursorAdapter(getActivity(), R.layout.order_listitem,
        cursor,
        new String[]{"RoomNum", "month", "day"}, new
        int[]{R.id.order_num, R.id.order_month, R.id.order_day});
        list_order.setAdapter(adapter);//绑定
    }
    dbHelper.close();//关闭数据库
}
}
}

```

## 四、心得体会

组长：刘禾予

这次分组作业最最最大的感触就两个：1. 年轻就要多踩坑；2. 合作精神真的真的很重要。

就我个人觉得这门课大多数知识都是自学，因为自己跟不太上老师的节奏，前期很长很长一段时间都花在看教学视频和自己动手操作上，想要自己实操就要吸取很多基础知识，当然后期我也鼓励组里成员和我一起刷视频，就算借鉴别人的代码自己也要先弄明白，我们从布局开始都是自己构思的，当然也有很多同学帮助，但是还是很有成就感的，虽然并没有实现线上功能，还有很多功能有待完善，但是只要有时间我觉得我们就可以做得很好，没有线上发展确实是一大遗憾。虽然在合并代码的过程中真的碰到很多坑，但我们吃一堑长一智。最后

一点，确实是合作精神很重要，我们初期缺少交流，在很多模块上的观点啊，逻辑上达不到一致的认知，有些些矛盾，但我还是挺享受我们晚上一起熬夜的时光，希望我们组里的每个人以后都能够共同进步！

#### **组员：王泽宇**

这学期，在老师们的介绍下，我们了解 android 的开发环境以及控件的功能。刚开始接触 Android 觉得既陌生又亲切，陌生在于没有学过具体的开发软件，亲切在于它在界面开发上和 web 也可以形成了相通的架构，更加方便。

我们小组为了能自己完成这个 APP，一起熬夜敲代码，一起熬夜学习，在学习的过程中也深深体会到“冰冻三尺非一日之寒”，学习并掌握 android 是一个需要努力和理解的过程，但还是完成了一个 APP——“水来了”，虽然不够精美，功能也有待更新，但这是努力的成果，不能妄自菲薄。

通过这学期对安卓的学习，大概了解了一些知识，对安卓有了初步的了解，虽然其中肯定有很多的不足，相信慢慢会体会到。

在此感谢我队员们的陪伴和帮助，谢谢！

#### **组员：黄珊珊**

完成这次分组作业的时候，特别是看到我们一起努力放入结果放到手机上运行的时候，感觉特别好，虽然过程很“曲折”，但是很庆幸的是我们完成了，真的也收获了很多。

虽然我们的作品需要改进的地方有很多，可以说是有些“简陋”，但是，这是我们四个纯自己敲的代码，从头到尾，没有复制粘贴。我们的各个功能、布局等，也都是我们四个一起想的。

过程中让我最印象深刻的是，我们四个一起熬夜刷视频。因为不懂的实在太多，所以要先看看教学视频才能开始。

这个作品的前期规划，我们想了很久，各种都想过，但是都觉得太大众了，所以最后我们敲定的是“水来了”。也许很多人也感觉不到它的特别及必要性，但是对于我们这种喝水又快，又老是忘了登记的人来说，这是很有必要的，它可以保证你不会“断水”。

记得刚刚开始的时候，老师就说要好好想好，不要想大了，到时候实现不了。然而，“年少无知”的我们，在想的时候很兴奋。但值得庆幸的是，我们大多都实现了，虽然不知道别人看我们的作品是怎么想的，可对于我们而言，这算不错的了，第一次作业就全手敲的，没有借鉴，看到不知道怎么实现的，就百度，看懂了再回来继续敲。

这次真的学到了很多东西，学会了一些基本方法，控件，也找了的编写 APP 的感觉，不仅如此，在小组合作的过程中也有分歧、也有互助、鼓励。

**组员：童巧琳**

为了本次作业我们的组员都特别努力，我也意识到分工合作的重要性和难度。首先，大家要凑在一起就有一定的困难，其次，每个人水平不一样，能做的程度也不一样。就我个人而言，我查了很多资料，收藏了一大堆博文，反复去看别人是怎么用的。甚至有好几次我用一种方法写完，运行的时候出错，而代码没有问题，最后还没有找到解决方法，我不得不改用另外一种方法。结果在某次机缘巧合之下，我知道自己被编译器给坑了。在完成作业的路上一路是坑，在踩过坑之后就是前所未有的充实。我深深体会到理论知识再多，实践的时候还是有书不会教你的东西冒出来。所以实践是很重要的。