

基于 JAVA 和安卓的移动业务大厅

2019 秋 JAVA 第一次作业 潘巧巧 1613415

目录

一、	基本信息与整体框架	3
二、	效果展示	5
1、	一级界面	5
(1)	用户登录	5
(2)	用户注册	7
(3)	使用嗖嗖	9
(4)	话费充值	11
(5)	资费说明	12
(6)	退出系统	12
2、	二级界面	13
(1)	本月账单查询	13
(2)	套餐余量查询	13
(3)	打印消费详单	14
(4)	套餐变更	14
(5)	办理退网	15
(6)	返回上一页	15
三、	实现详情: baseClass.java	16
1、	三个接口定义	16
2、	服务包类	16
	手机卡类	19
	消费记录类	20
	使用场景类	20
	核心数据结构初始化	20
四、	实现详情: MainActivity.java	21
1、	初始化预存信息 SaveInfoAtFirst()	21
2、	初始化函数 onCreate()	21
3、	用户登录 UsrLogin(View v)	21

4、	用户注册 UsrRegister(View v).....	22
5、	使用嗖嗖 UseSoso(View v)	24
6、	话费充值 RechargePhoneBill(View v).....	25
7、	资费说明 ShowFare(View v)	25
8、	退出系统 ExitSoso(View v).....	26
五、	实现详情: SecondActivity.java.....	26
1、	初始化函数 onCreate().....	26
2、	本月账单查询 showAmountDetail(View v).....	26
3、	套餐余量查询 showRemainDetail(View v).....	27
4、	打印消费详单 showDescription(View v).....	27
5、	套餐变更 changingPack(View v)	27
6、	办理退网 delCard(View v).....	28
7、	返回上一页 TurnBack()	29
六、	遇到的问题:	29
七、	作业提升方案	30

一、基本信息与整体框架

1、基本信息：本作业使用 android studio 完成一个完整的安卓程序，完成作业 PPT 中提到的所有功能，共 2 个 XML 页面，3 个 JAVA 文件。

GitHub 地址: <https://github.com/Catnip0709/SosoPhone>

2、预存手机号信息

手机号	密码	套餐	余额	用户名
13911223344	3344	超人套餐	100	Catnip
13944332211	2211	网虫套餐	50	David

3、2 个 XML 页面



(一级页面)



(二级页面)

4、3 个 JAVA 文件

(1) baseClass.java

① 三个接口 interface：打电话 CallService、发短信 SendService、上网

NetService

② 服务包父类子类定义与实现

父类	服务包父类	ServicePackage	
子类 1	话痨套餐	TalkPackage	implements CallService, SendService
子类 2	超人套餐	SuperPackage	implements CallService, SendService, NetService
子类 3	网虫套餐	NetPackage	implements NetService

③ 手机卡类、消费记录类、使用情景类的定义与实例化

```
//手机卡类
class MobileCard
{...}
//消费记录
class ConsumInfo
{...}
//使用场景
class Scene
{...}

//*****//

static Map<String, baseClass.MobileCard> card = new HashMap(); // 嗖嗖移动用户列表
static Map<String, List<baseClass.ConsumInfo>> consumInfos = new HashMap(); // 消费信息
static Map<Integer, baseClass.Scene> scene = new HashMap(); // 场景信息
```

(2) MainActivity.java

实现初始化和一级界面相关功能

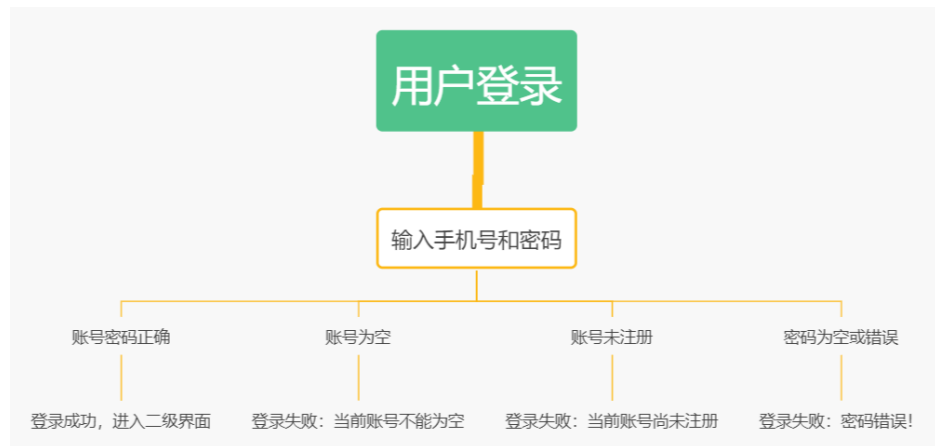
(3) SecondActivity.java

实现二级界面相关功能

二、效果展示

1、一级界面

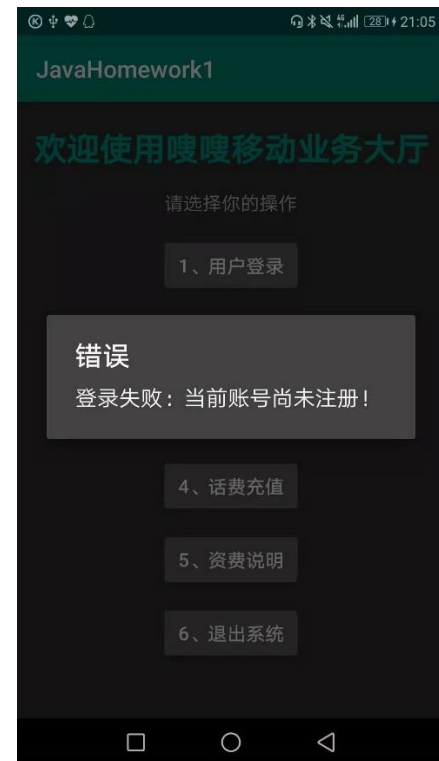
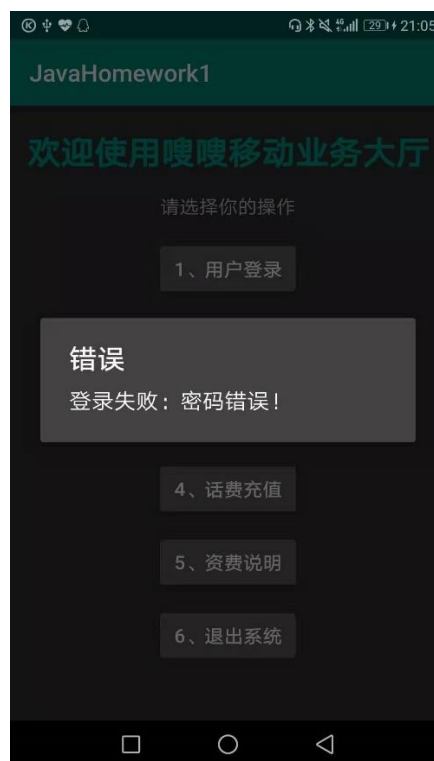
(1) 用户登录



在一级界面中点击“用户登录”，弹出以下对话框，输入预存的账号密码或新注册的账号密码后点击“下一步”。其中账号只能输入数字，密码会被隐藏显示



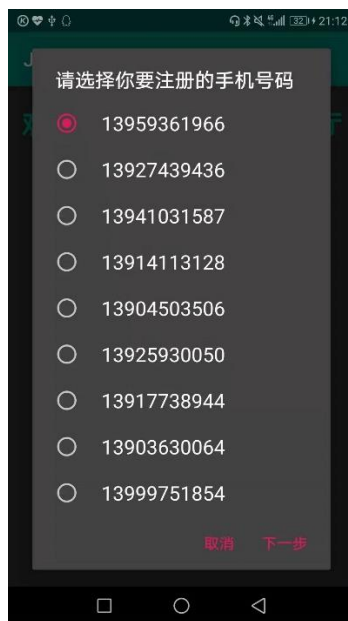
用户登录四种结果展示：



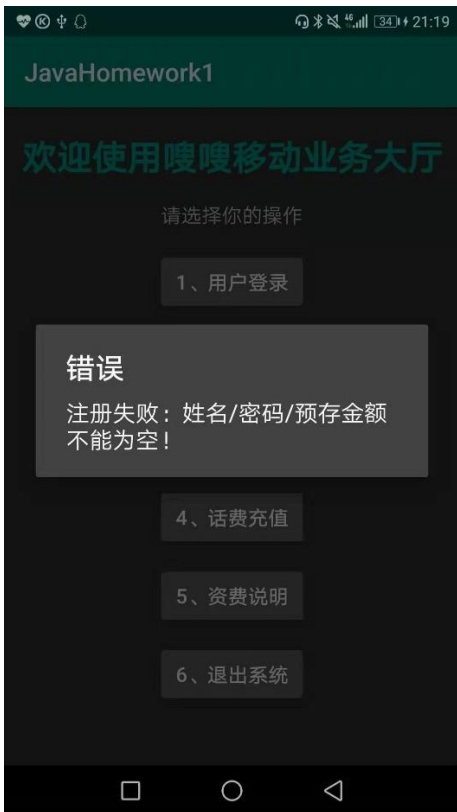
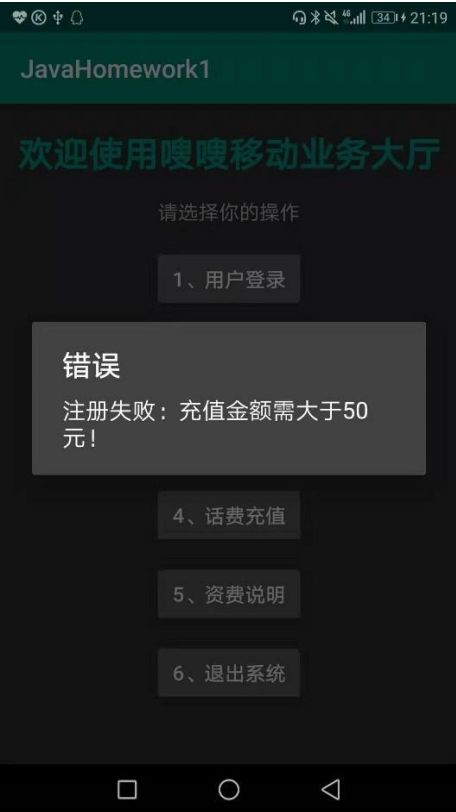
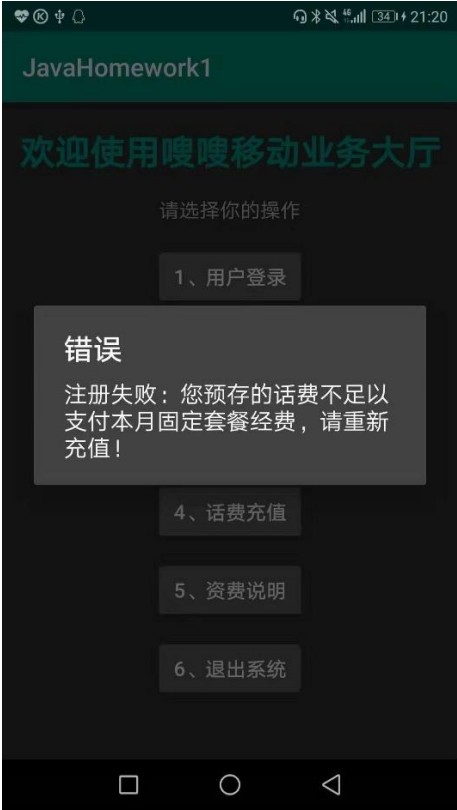
(2) 用户注册



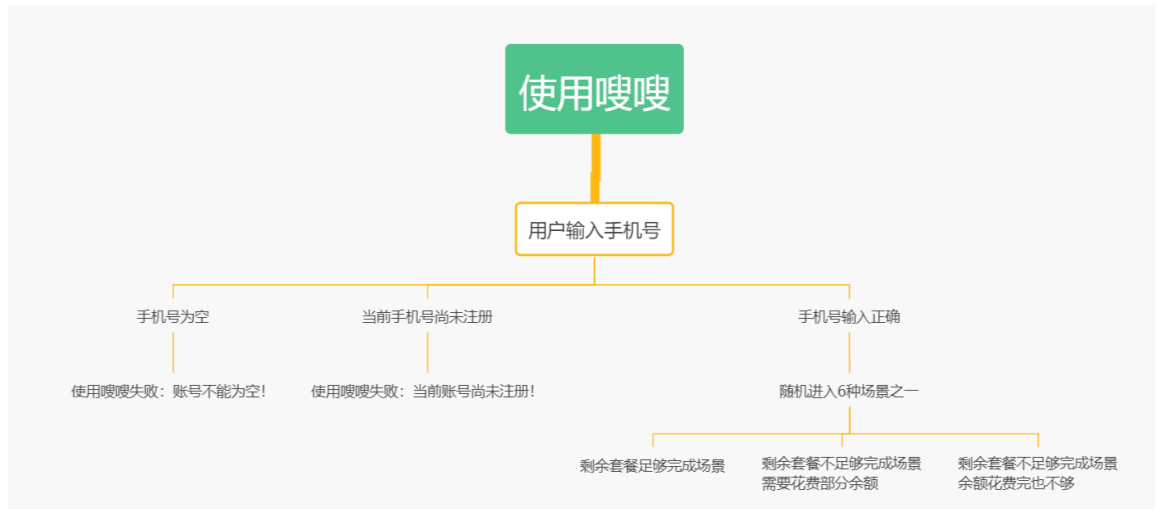
在一级界面中点击“用户注册”，弹出随机生成的9个139开头号码单选框，用户选好后点击“下一步”，弹出3个套餐单选框，用户选好后点击“下一步”，填写用户名、密码、充值金额。其中充值金额只能输入数字，密码会被隐藏显示。



用户注册 4 种结果展示



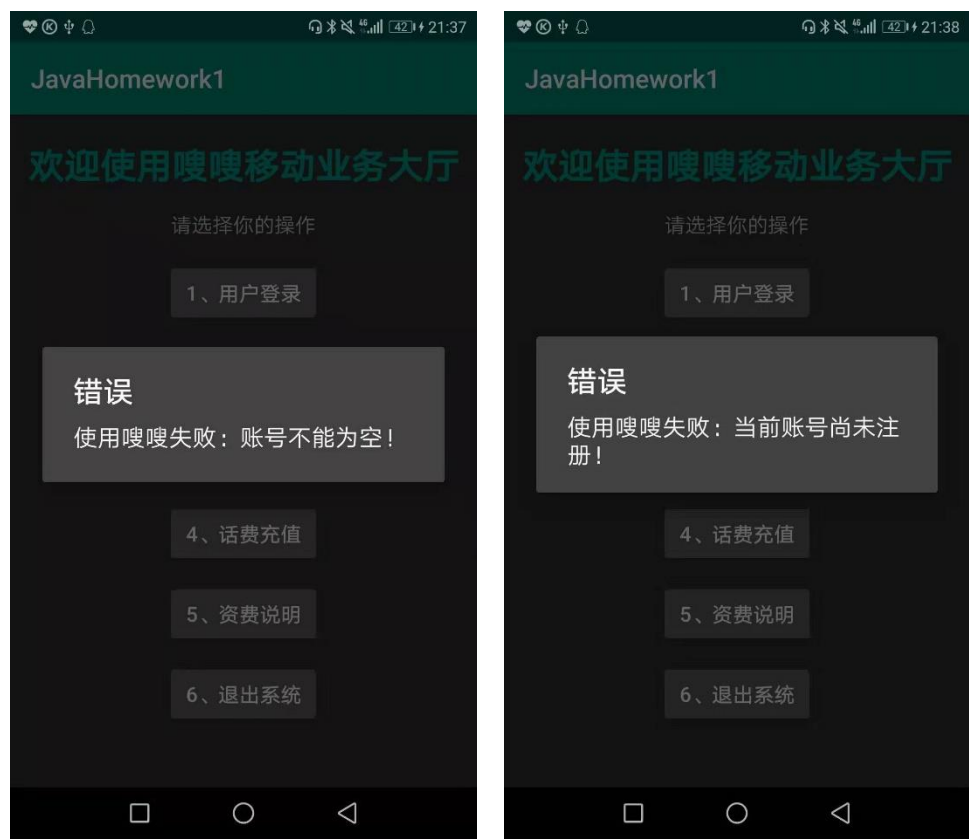
(3) 使用嗖嗖



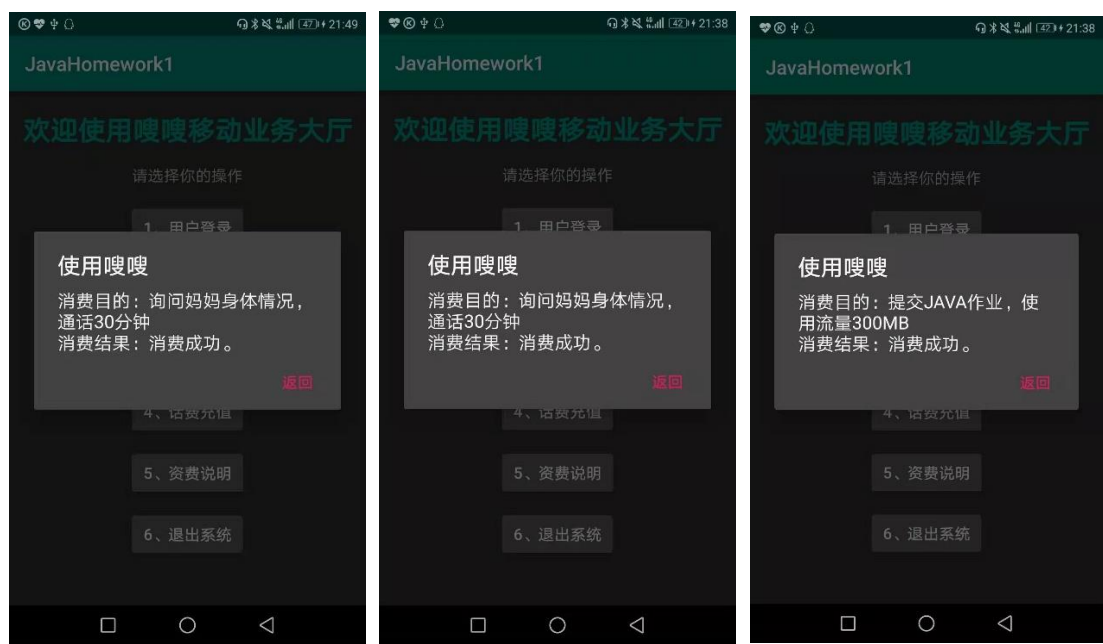
在一级界面选择“使用嗖嗖”，在弹出对话框中输入要消费的手机号，点击下一步。

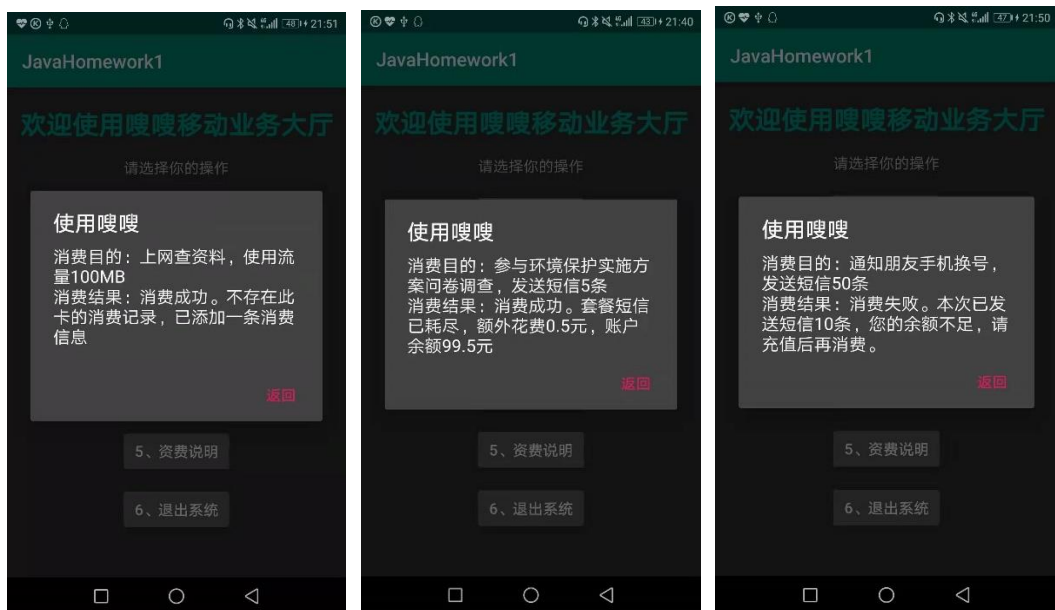


此时可能有 2 种失败情况。



若成功使用嗖嗖，会触发 6 种随机事件。并且可能存在“剩余套餐足够完成场景”、“剩余套餐不足够完成场景，需要花费部分余额”、“剩余套餐不足够完成场景，余额花费完也不够” 三种情况。

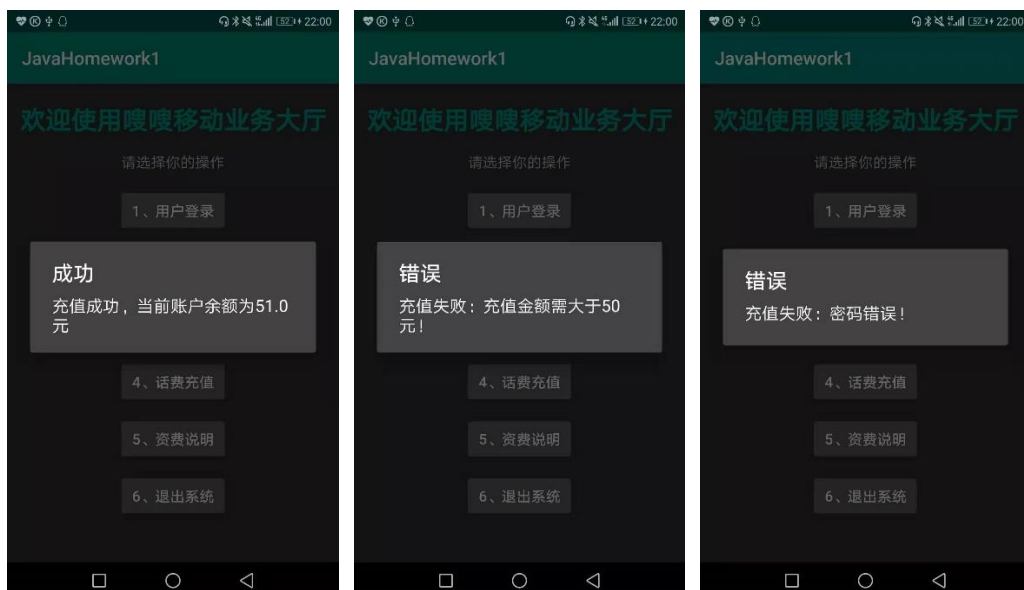


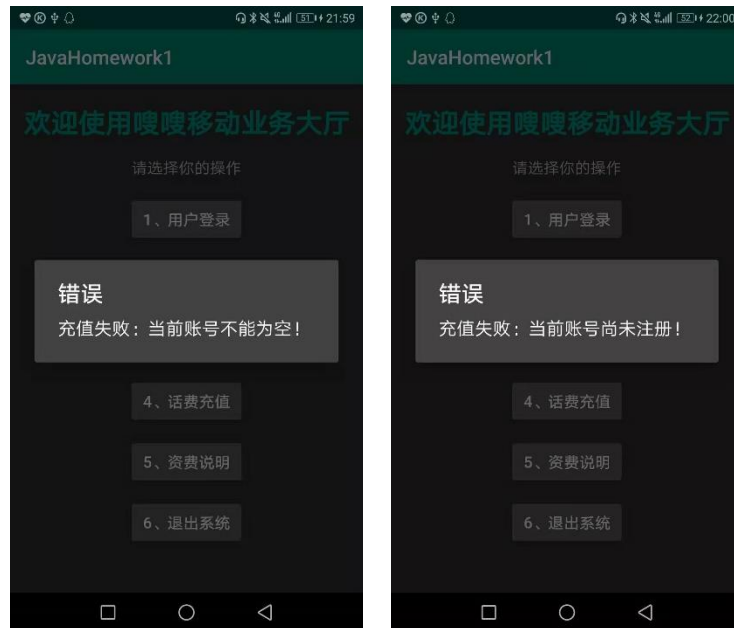


(4) 话费充值



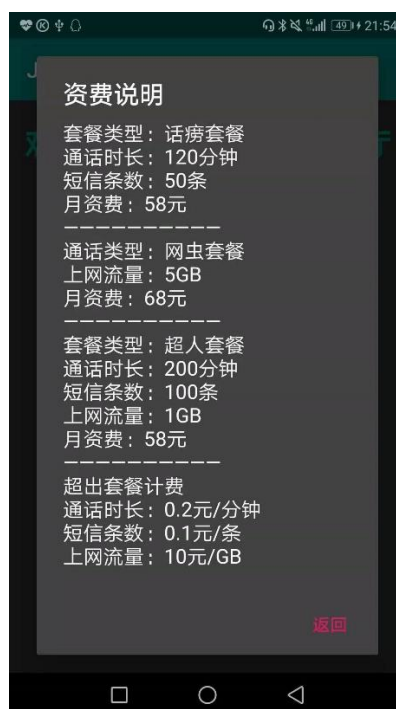
用户在一级界面中点击“话费充值”，填写手机号、密码、充值金额，其中手机号和充值金额只能输入数字、密码会隐藏显示。





(5) 资费说明

在一级界面中点击资费说明，弹出三个套餐相关及额外消费的的资费说明



(6) 退出系统

在一级界面中点击退出系统，APP 退出，全流程结束。

2、二级界面

(1) 本月账单查询

在二级菜单中点击“本月菜单查询”可见



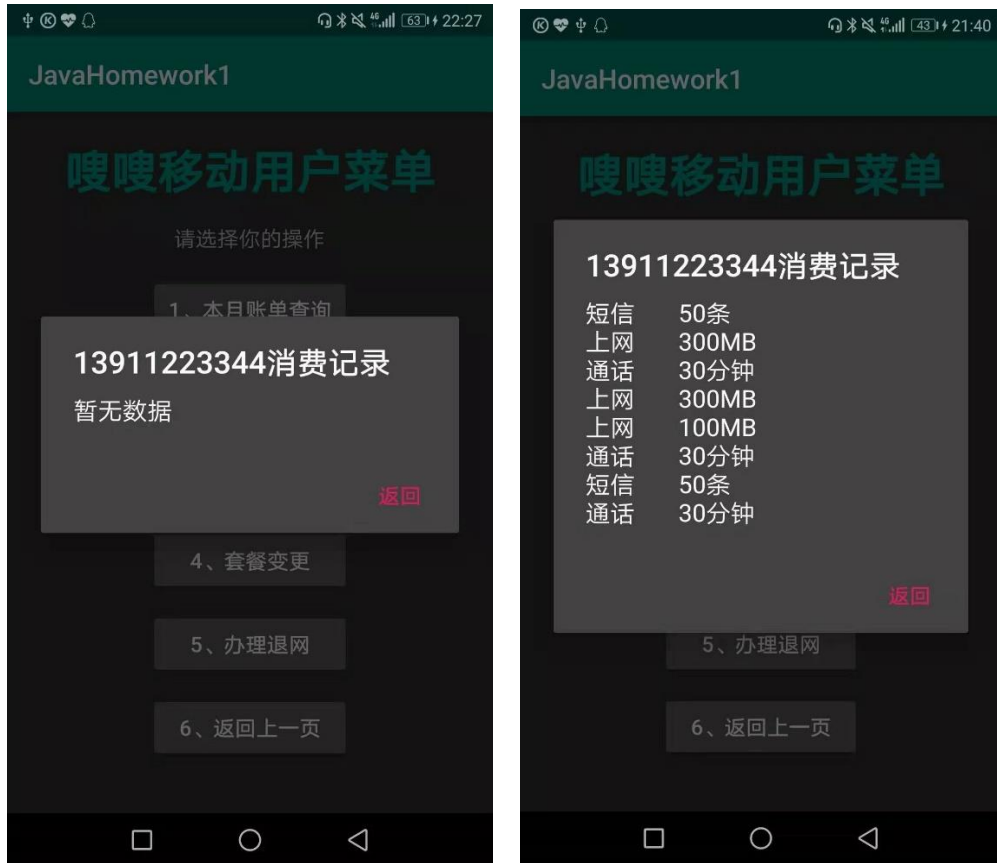
(2) 套餐余量查询

在二级菜单中点击套餐余量查询可见



(3) 打印消费详单

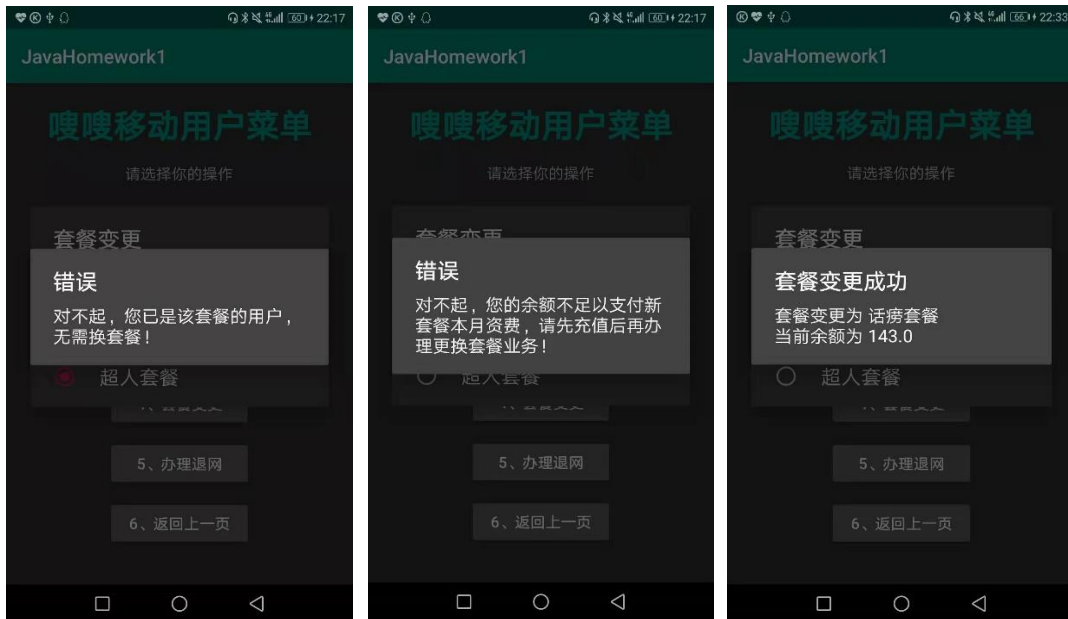
在二级菜单中点击“打印消费详单”可见，若该账号暂无消费记录显示



(4) 套餐变更

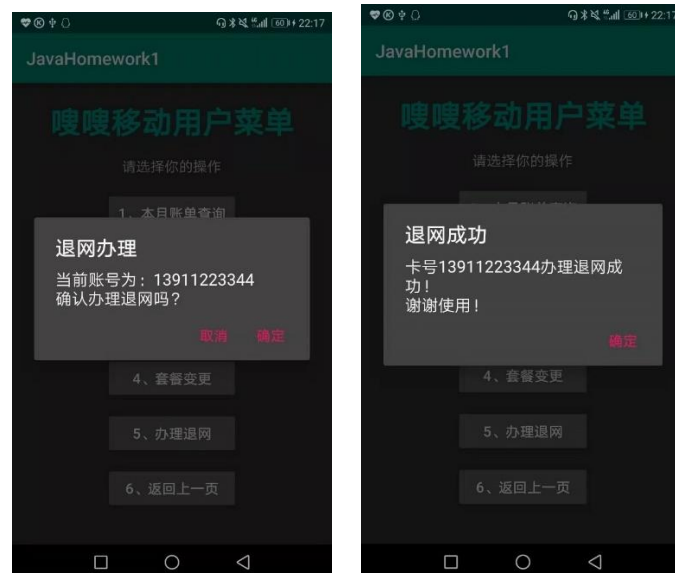


用户在二级界面选择“套餐变更”，并选择自己想要的新套餐，会出现三种情况。



(5) 办理退网

用户在二级界面选择“办理退网”，并确定，则成功退网，并退回一级界面。退网后无法再使用此账号登录。



(6) 返回上一页

用户在二级界面中点击“返回上一页”，则相当于退出登录，返回一级界面。

三、实现详情：baseClass.java

1、 三个接口定义

根据安卓开发的需要修改了作业要求中接口的参数。第一个参数为即将弹出的 dialog，第二个参数为调用接口的手机卡类实例，第三个参数为活动标签。如 act = 1 是"问候客户，谁知其如此难缠，通话 90 分钟"，act = 2 是"询问妈妈身体情况，通话 30 分钟"。

接口会在套餐类中被实现。

```
//接口1：打电话
interface CallService
{
    void call(AlertDialog.Builder alterDialog, baseClass.MobileCard curCard, int act);
}

//接口2：发短信
interface SendService
{
    void send(AlertDialog.Builder alterDialog, baseClass.MobileCard curCard, int act);
}

//接口3：上网
interface NetService
{
    void netPlay(AlertDialog.Builder alterDialog, baseClass.MobileCard curCard, int act);
}
```

2、 服务包类

包括一个抽象父类 ServicesPackage，三个子类 TalkPackage + SuperPackage + NetPackage

```
/****** 服务包 *****/
public enum ACTION
{
    EM_TALKTIME, EM_SMSCOUNT, EM_FLOW
}

//服务包类
abstract class ServicePackage
{
    ...
}

//服务包子类1：话唠套餐
class TalkPackage extends baseClass.ServicePackage implements CallService, SendService
{
    ...
}

//服务包子类2：超人套餐
class SuperPackage extends baseClass.ServicePackage implements CallService, SendService, NetService
{
    ...
}

//服务包子类3：网虫套餐
class NetPackage extends baseClass.ServicePackage implements NetService
{
    ...
}
/****** */
```


① 抽象父类 ServicePackage

包含基本属性 price 和两个方法，GetPackageName 用于返回服务包名，ShowInfo 用于展示套餐信息

```
//服务包类
abstract class ServicePackage
{
    double price;
    abstract String GetPackageName();
    abstract void ShowInfo(AlertDialog.Builder alterDialog, String curNumber);
}
```

② 服务包子类，以超人套餐为例。

超人套餐包含新增的 3 个属性（通话时间、短信条数、上网流量），构造函数根据作业要求给 4 个属性赋值，返回套餐名字函数直接返回“超人套餐”的 String，其他四个方法中，ShowInfo 函数实现父类中的 abstract 抽象函数，另外三个实现调用的接口。

其他套餐实现与超人套餐相似。

```
//服务包子类2：超人套餐
class SuperPackage extends baseClass.ServicePackage implements CallService, SendService, NetService
{
    int TalkTime;
    int smsCount;
    int flow;
    SuperPackage()
    {
        price    = 78;
        TalkTime = 200;
        smsCount = 100;
        flow     = 1024;
    }
    public String GetPackageName() { return "超人套餐"; }
    public void ShowInfo(AlertDialog.Builder alterDialog, String curNumber) //展示剩余套餐
    { ... }
    public void call(AlertDialog.Builder alterDialog, baseClass.MobileCard curCard, int act) //打电话服务接口
    { ... }
    public void send(AlertDialog.Builder alterDialog, baseClass.MobileCard curCard, int act) //发短信服务接口
    { ... }
    public void netPlay(AlertDialog.Builder alterDialog, baseClass.MobileCard curCard, int act) //上网服务接口
    { ... }
}
```

ShowInfo()函数

分情况讨论，若套餐通话时间/短信条数/流量还没用完，展示剩余量，否则显示 0。其中流量需要使用 DecimalFormat 转换成 “整数至少有一位，小数保留一位” 的表示。

```
public void ShowInfo(AlertDialog.Builder alterDialog, String curNumber){ //展示剩余套餐
    alterDialog.setTitle("套餐余量查询");//标题
    double showTalkTime, showsmsCount, showFlow;

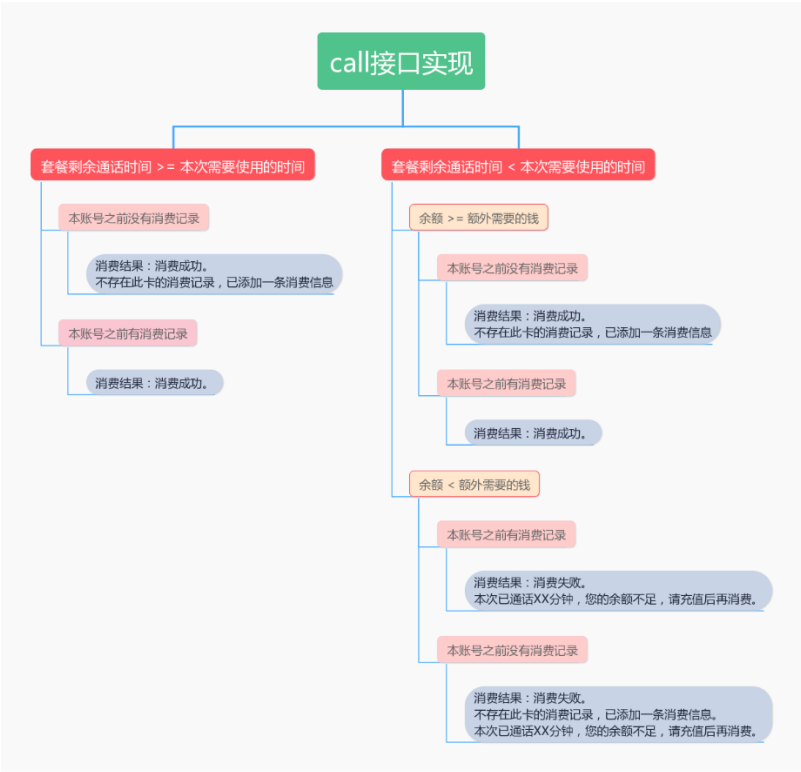
    if(TalkTime > card.get(curNumber).RealTalkTime) //套餐中通话时间还没用完
        showTalkTime = TalkTime - card.get(curNumber).RealTalkTime;
    else //套餐中通话时间已经耗尽
        showTalkTime = 0;

    if(smsCount > card.get(curNumber).RealSMSCount) //套餐中短信条数还没用完
        showsmsCount = smsCount - card.get(curNumber).RealSMSCount;
    else //套餐中短信条数已经耗尽
        showsmsCount = 0;

    if(flow > card.get(curNumber).RealFlow) //套餐中流量还没用完
        showFlow = flow - card.get(curNumber).RealFlow;
    else //套餐中流量已经耗尽
        showFlow = 0;

    DecimalFormat format = new DecimalFormat( pattern: "0.0");
    String content = "您的卡号: " + curNumber + "\n套餐内结余\n" +
        "通话时长: " + showTalkTime + "分钟\n" +
        "短信条数: " + showsmsCount + "条\n" +
        "上网流量: " + format.format( number: showFlow/1024.0) + "GB";
    alterDialog.setMessage(content); //内容
    alterDialog.show();
}
```

接口实现，以 Call 为例：思维导图如下，具体实现详见代码



手机卡类

包括卡的基本属性和 2 个方法，构造函数将三个实际消费初始化为 0，其他属性在注册时根据实际情况填充。ShowMsg 方法展示手机卡信息。

```
//手机卡类
class MobileCard
{
    String CardNumber;        //卡号
    String UserName;          //用户名
    String PassWord;          //密码
    baseClass.ServicePackage SerPackage; //服务包
    double ConsumAmount;      //总消费额
    double Money;             //余额
    int RealTalkTime;         //实际通话时间
    int RealsmsCount;         //实际短信数量
    int RealFlow;             //实际流量消耗

    MobileCard()
    {
        this.RealTalkTime = 0;
        this.RealsmsCount = 0;
        this.RealFlow = 0;
    }

    void ShowMsg(AlertDialog.Builder alterDialog)
    {
        ...
    }
}
```

```
void ShowMsg(AlertDialog.Builder alterDialog)
{
    DecimalFormat format = new DecimalFormat("pattern: "#.0");
    String content = "卡号: " + CardNumber + "\n" +
        "用户名: " + UserName + "\n" +
        "服务包: " + SerPackage.GetPackageName() + "\n" +
        "套餐资费: " + SerPackage.price + "元\n" +
        "合计消费额: " + ConsumAmount + "元\n" +
        "账户余额: " + format.format(Money) + "元\n" +
        "通话时间: " + RealTalkTime + "分钟\n" +
        "短信数量: " + RealsmsCount + "条\n" +
        "流量消耗: " + RealFlow + "MB\n";

    alterDialog.setMessage(content); //内容
    alterDialog.show();
}
```

消费记录类

```
class ConsumInfo
{
    ConsumInfo(ACTION act, int count, double m_price){
        consum = count;
        switch (act){
            case EM_TALKTIME:
                tail = "分钟";
                type = "通话";
                break;
            case EM_SMSCOUNT:
                tail = "条";
                type = "短信";
                break;
            case EM_FLOW:
                tail = "MB";
                type = "上网";
                break;
            default:
                assert false: "wrong input in ConsumInfo";
        }
        price = m_price;
    }

    int consum;    // 消费数量
    double price;  // 消费价格
    String type;   // 消费类型: 通话、短信、上网
    String tail;   // 单位
}
```

使用场景类

```
//使用场景
class Scene
{
    Scene(String m_type, ACTION m_data, int m_description, double m_price)
    {
        type = m_type;
        data = m_data;
        description = m_description;
        price = m_price;
    }

    String type;    // 描述场景
    ACTION data;    // 消费类型
    int description; // 消费详情
    double price;   // 本消费场景造成的消费金额
}
```

核心数据结构初始化

```
static Map<String, baseClass.MobileCard> card = new HashMap(); // 嗖嗖移动用户列表
static Map<String, List<baseClass.ConsumInfo>> consumInfos = new HashMap(); // 消费信息
static Map<Integer, baseClass.Scene> scene = new HashMap(); // 场景信息
```

四、实现详情： MainActivity.java

1、 初始化预存信息 SaveInfoAtFirst()

展示部分预存信息，其他详见代码

```
//预存用户信息1
baseClass.MobileCard card1 = new baseClass().new MobileCard();
card1.UserName = "Catnip";
card1.PassWord = "3344";
card1.CardNumber = "13911223344";
card1.SerPackage = new baseClass().new SuperPackage();
card1.ConsumAmount = 78;
card1.Money = 1;
card.put(card1.CardNumber, card1);
```

```
//预存6个场景信息
String detail = "问候客户，谁知其如此难缠，通话90分钟";
baseClass.Scene scenel = new baseClass().new Scene(detail, EM_TALKTIME, m_description: 90, m_price: 18);
scene.put(1, scenel);
```

2、 初始化函数 onCreate()

预存用户信息和场景信息后展示一级界面

```
@Override
protected void onCreate(Bundle savedInstanceState)
{
    //预存用户信息和场景信息
    SaveInfoAtFirst();
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main);
}
```

3、 用户登录 UsrLogin(View v)

思维导图详见本实验报告第五页。具体实现为跳出交互界面，获取用户输入的用户名和密码，然后按照思维导图流程实现。

如果用户登录成功（用户名、密码合法且匹配），使用 Intent 关键字进行一级界面到二级界面的跳转，参数为用户手机号。此处仅展示跳转部分代码，详见源码。

```

else {
    AlertDialog.Builder result = new AlertDialog.Builder( context: MainActivity.this);
    result.setTitle("成功");
    result.setMessage("登录成功");
    result.setPositiveButton( text: "下一步",
        (judge_dialog, which) -> {
            Intent MyIntent = new Intent( packageContext: MainActivity.this,SecondActivity.class);
            MyIntent.putExtra( name: "number", GetNumber); //参数为用户手机号
            startActivityForResult(MyIntent, requestCode: 1); // 启动Activity
        });
    result.show();
}

```

4、 用户注册 UserRegister(View v)

用户注册思维导图见本实验报告第七页。本流程涉及多个步骤，为避免代码过长，拆成多个函数。

```

//***** 注册相关 *****
// 获得指定位数的随机数
public static String getRandom(int len)
{...}
// 用户注册步骤1: 9个随机号码
public void Reg_Show9Number()
{...}
// 用户注册步骤2: 选择套餐
public void Reg_ChoosePackage(final baseClass.MobileCard newCard)
{...}
// 用户注册步骤3: 填写姓名密码 + 充值
public void FullfillName(final baseClass.MobileCard newCard)
{...}
//*****

```

(1) GetRandom() : 利用 Random 类生成随机数

```

// 获得指定位数的随机数
public static String getRandom(int len)
{
    Random r = new Random();
    StringBuilder rs = new StringBuilder();
    for (int i = 0; i < len; i++) {
        rs.append(r.nextInt( bound: 10));
    }
    return rs.toString();
}

```

- (2) Reg_Show9Number(): 生成 9 个随机号码, 注意要判断合法性 (是否已经注册过), 并以单选框的形式展现给用户, 用户选择后跳转到选择套餐函数。

```
public void Reg_Show9Number() {
    final String[] saveNumber = new String [9]; // 生成9个随机号码
    for(int i = 0; i < 9; i = i + 1) {
        String number;
        while(true) {
            number = "139" + getRandom( len: 8);
            if(!card.containsKey(number)) //合法性判断, 是否已经注册
                break;
        }
        saveNumber[i] = number;
    }

    final baseClass.MobileCard newCard = new baseClass().new MobileCard();
    AlertDialog.Builder NumberDialog = new AlertDialog.Builder( context: MainActivity.this);
    NumberDialog.setTitle("请选择你要注册的手机号码"); //标题
    NumberDialog.setSingleChoiceItems(saveNumber, checkedItem: -1,
        new DialogInterface.OnClickListener() {
            public void onClick(DialogInterface number_dialog, int which) { //选择手机号
                newCard.CardNumber = saveNumber[which];
            }
        });
    NumberDialog.setPositiveButton( text: "下一步",
        new DialogInterface.OnClickListener() {
            public void onClick(DialogInterface next_dialog, int which) {
                Reg_ChoosePackage(newCard); //下一步, 选择套餐
            }
        });
    NumberDialog.setNegativeButton( text: "取消", listener: null);
    NumberDialog.show();
}
```

- (3) Reg_ChoosePackage(): 选择套餐函数。弹出 3 个套餐的单选框供用户选择。

```
// 用户注册步骤2: 选择套餐
public void Reg_ChoosePackage(final baseClass.MobileCard newCard) {
    AlertDialog.Builder PackageDialog = new AlertDialog.Builder( context: MainActivity.this);
    PackageDialog.setTitle("请选择你要办理的套餐"); //标题
    String[] savePackage = {"话唠套餐", "超人套餐", "网虫套餐"};
    PackageDialog.setSingleChoiceItems(savePackage, checkedItem: -1,
        new DialogInterface.OnClickListener() {
            public void onClick(DialogInterface package_dialog, int which) { //选择套餐
                baseClass.ServicePackage temp_package;
                if(which == 0)
                    temp_package = new baseClass().new TalkPackage();
                else if(which == 1)
                    temp_package = new baseClass().new SuperPackage();
                else
                    temp_package = new baseClass().new NetPackage();
                newCard.SerPackage = temp_package;
            }
        });
    PackageDialog.setPositiveButton( text: "下一步",
        new DialogInterface.OnClickListener() {
            public void onClick(DialogInterface next_dialog, int which) {
                FullfillName(newCard); //下一步, 填写姓名
                next_dialog.dismiss();
            }
        });
    PackageDialog.setNegativeButton( text: "取消", listener: null);
    PackageDialog.show();
}
```

- (4) FullfillName(): 用户在弹出的输入框中填写注册用户的信息，代码获取用户信息并做合法性判断，若完全合法则将信息整合后填入 card 的 map 中。流程详见本实验报告第七页思维导图。具体实现详见源码，这里仅展示存储部分信息代码。

```
newCard.UserName      = GetName;
newCard.PassWord      = GetPassword;
newCard.ConsumAmount  = newCard.SerPackage.price; //已消费总金额
newCard.Money         = d_GetMoney - newCard.SerPackage.price; //当前余额
card.put(newCard.CardNumber, newCard);
```

5、 使用嗖嗖 UseSoso(View v)

用户在弹出的输入框中输入手机号，若手机号合法，点击下一步后随机进入编号为 1-6 的消费情景（直接调用服务包类中封装的接口函数）。思维导图见本实验报告第九页，此处仅展示部分代码，详见源码。

```
AlertDialog.Builder dialog = new AlertDialog.Builder( context: MainActivity.this);
dialog.setTitle("使用嗖嗖");
int random = new Random().nextInt( bound: 6) + 1; //生成[1, 6]的随机数
if(random == 1 || random == 2) { //通话相关，涉及超人和话痨
    CallService temp_interface = new baseClass().new SuperPackage(); //先初始化为超人
    if (card.get(GetNumber).SerPackage instanceof baseClass.TalkPackage) //如果是话痨，再改成话痨
        temp_interface = new baseClass().new TalkPackage();
    temp_interface.call(dialog, card.get(GetNumber), random);
}
else if(random == 3 || random == 4){ //短信相关，涉及超人和话痨
    SendService temp_interface = new baseClass().new SuperPackage(); //先初始化为超人
    if (card.get(GetNumber).SerPackage instanceof baseClass.TalkPackage) //如果是话痨，再改成话痨
        temp_interface = new baseClass().new TalkPackage();
    temp_interface.send(dialog, card.get(GetNumber), random);
}
else{ //流量相关，涉及超人和网虫
    NetService temp_interface = new baseClass().new SuperPackage(); //先初始化为超人
    if (card.get(GetNumber).SerPackage instanceof baseClass.TalkPackage) //如果是网虫，再改成网虫
        temp_interface = new baseClass().new NetPackage();
    temp_interface.netPlay(dialog, card.get(GetNumber), random);
}
```


6、 话费充值 RechargePhoneBill(View v)

用户在弹出的输入框中输入手机号、密码、充值金额，若手机号和密码合法且匹配，则给用户账户充值对应金额，其中充值金额需大于 50 元。此处仅展示充值成功部分代码，其他实现详见源码。思维导图见本实验报告第 11 页。

```
else {
    AlertDialog.Builder result = new AlertDialog.Builder( context: MainActivity.this);
    result.setTitle("成功");
    card.get(GetNumber).Money = card.get(GetNumber).Money + Double.parseDouble(GetMoney);
    String content = "充值成功，当前账户余额为" + card.get(GetNumber).Money + "元";
    result.setMessage(content);
    result.show();
}
```

7、 资费说明 ShowFare(View v)

资费说明提前预存到 txt 中，用户点击资费说明按钮则读取 txt 并展示。读取 txt 时需要注意文本编码，防止中文乱码。LoadFareFile()读取文本，按行读取，拼接成字符串，展示到对话框中，若出现路径错误、文本空白等问题，执行报错。

```
//5、资费说明
public void ShowFare(View v) {
    AlertDialog.Builder FareDialog = new AlertDialog.Builder( context: MainActivity.this);
    FareDialog.setTitle("资费说明"); //标题
    String content = LoadFareFile();
    FareDialog.setMessage(content);
    FareDialog.setNegativeButton( text: "返回", listener: null);
    FareDialog.show();
}
```

```
//加载fare.txt
public String LoadFareFile()
{
    String result = "";
    try {
        //按行读取 + 解决中文乱码
        InputStreamReader inputReader = new InputStreamReader(getResources().getAssets().open( fileName: "fare.txt"), charsetName: "GB2312");
        if (inputReader != null) {
            BufferedReader buffreader = new BufferedReader(inputReader);
            String line = "";
            while ((line = buffreader.readLine()) != null) {
                result = result + line + "\n";
            }
            inputReader.close();
        }
    } catch (Exception e) {
        AlertDialog.Builder dialog = new AlertDialog.Builder( context: MainActivity.this);
        dialog.setTitle("错误");
        dialog.setMessage("执行出错！");
        dialog.show();
    }
    return result;
}
```

8、 退出系统 ExitSoSo(View v)

调用杀死进程函数退出程序

```
//6、退出系统
public void ExitSoSo(View v)
{
    android.os.Process.killProcess(android.os.Process.myPid());
}
```

五、实现详情：SecondActivity.java

1、 初始化函数 onCreate()

展示二级界面

```
@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.menu2);
}
```

2、 本月账单查询 showAmountDetail(View v)

通过 Intent 获取用户手机号，然后直接调用 baseClass 中封装的手机卡类的方法

ShowMsg 展示用户本月账单。

```
//1、本月账单查询
public void showAmountDetail(View v)
{
    final String curNumber = this.getIntent().getStringExtra( name: "number");
    AlertDialog.Builder InfoDialog = new AlertDialog.Builder( context: SecondActivity.this);
    InfoDialog.setTitle("本月账单查询"); //标题
    card.get(curNumber).ShowMsg(InfoDialog);
}
```

3、套餐余量查询 showRemainDetail(View v)

通过 Intent 获取当前用户手机号, 直接调用 baseClass 中各套餐类封装的 ShowInfo 方法展示当前用户的套餐余量信息。

```
//2、套餐余量查询
public void showRemainDetail(View v)
{
    final String curNumber = this.getIntent().getStringExtra( name: "number");
    AlertDialog.Builder InfoDialog = new AlertDialog.Builder( context: SecondActivity.this);
    card.get(curNumber).SerPackage.ShowInfo(InfoDialog, curNumber);
}
```

4、打印消费详单 showDescription(View v)

通过 Intent 获取当前用户手机号, 在 consumInfos 的 map 中查找当前用户的消费记录, 若无消费记录显示 “暂无数据”, 否则拼接消费记录并展示。

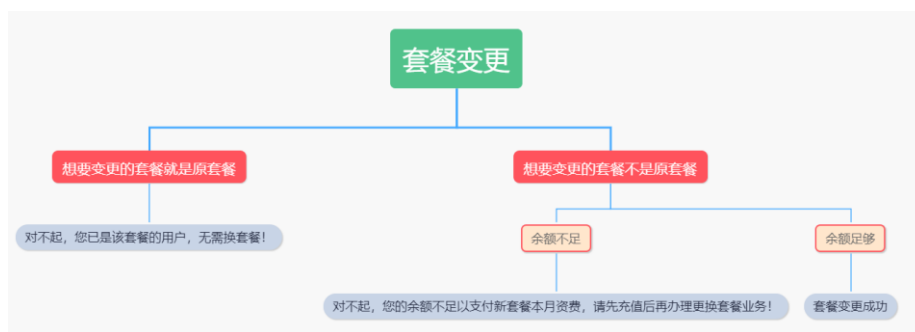
```
//3、打印消费详单
public void showDescription(View v)
{
    final String curNumber = this.getIntent().getStringExtra( name: "number");

    AlertDialog.Builder InfoDialog = new AlertDialog.Builder( context: SecondActivity.this);
    InfoDialog.setTitle(curNumber + "消费记录"); //标题
    String content = "";
    if(!consumInfos.containsKey(curNumber))
    {
        content += "暂无数据";
    }
    else {
        for (int i = 0; i < consumInfos.get(curNumber).size(); i++) {
            String line = consumInfos.get(curNumber).get(i).type + " " +
                consumInfos.get(curNumber).get(i).consum + consumInfos.get(curNumber).get(i).tail + "\n";
            content += line;
        }
    }
    InfoDialog.setMessage(content);
    InfoDialog.setNegativeButton( text: "返回", listener: null);
    InfoDialog.show();
}
```

5、套餐变更 changingPack(View v)

根据以下思维导图流程进行, 如果套餐变更成功, 需要进行扣除月资费、原数据清零等

相关操作。此处仅展示套餐变更成功部分代码，其他部分详见源码。



```
else{
    card.get(curNumber).Money -= temp.price;
    card.get(curNumber).ConsumAmount += temp.price;
    card.get(curNumber).SerPackage = temp;
    card.get(curNumber).RealSMSCount = 0;
    card.get(curNumber).RealTalkTime = 0;
    card.get(curNumber).RealFlow = 0;
    AlertDialog.Builder result = new AlertDialog.Builder(context: SecondActivity.this);
    result.setTitle("套餐变更成功");
    String content = "套餐变更为 " + Pack[which] + "\n" +
        "当前余额为 " + card.get(curNumber).Money;
    result.setMessage(content);
    result.show();
}
```

6、 办理退网 delCard(View v)

通过 Intent 获取当前用户手机号。弹出对话框询问用户是否确认退网，若确认，则删除所有相关记录，并返回一级界面。

```
public void delCard(View v){
    final String curNumber = this.getIntent().getStringExtra( name: "number");
    AlertDialog.Builder InfoDialog = new AlertDialog.Builder(context: SecondActivity.this);
    InfoDialog.setTitle("退网办理"); //标题
    String content = "当前账号为: " + curNumber + "\n确认办理退网吗? ";
    InfoDialog.setMessage(content); //内容
    InfoDialog.setPositiveButton( text: "确定",
        new DialogInterface.OnClickListener() {
            public void onClick(DialogInterface judge_dialog, int which) {
                card.remove(curNumber);
                if(baseClass.consumInfos.containsKey(curNumber))
                    consumInfos.remove(curNumber);
                AlertDialog.Builder QuicDialog = new AlertDialog.Builder(context: SecondActivity.this);
                QuicDialog.setTitle("退网成功");
                QuicDialog.setMessage("卡号" + curNumber + "办理退网成功! \n谢谢使用! ");
                QuicDialog.setPositiveButton( text: "确定",
                    new DialogInterface.OnClickListener() {
                        public void onClick(DialogInterface judge_dialog, int which) {
                            Intent MyIntent = new Intent();
                            setResult(RESULT_OK, MyIntent);
                            finish();
                        }
                    });
                QuicDialog.show();
            }
        });
    InfoDialog.setNegativeButton( text: "取消", listener: null);
    InfoDialog.show();
}
```

7、 返回上一页 TurnBack()

通过 Intent 返回一级界面。

```
//6、返回上一页
public void TurnBack(View v) {
    Intent MyIntent = new Intent();
    setResult(RESULT_OK, MyIntent);
    finish();
}
```

六、遇到的问题：

1、移动端调试，执行失败。

解决方案：需要 Edit Configurations 中设置 Target 为 USB Device，手机开启开发者模式

2、报错 Manifest merger failed : uses-sdk:minSdkVersion 11 cannot be smaller

than version 14 declared in library [com.android.support:appcompat-v7:28.0.0]

C:\Users\11500\gradle\caches\transforms-2\files-

2.1\3fc240e844301d1c6387f05fbd37cd95\AndroidManifest.xml as the library

might be using APIs not available in 11

Suggestion: use a compatible library with a minSdk of at most 11, or increase

this project's minSdk version to at least 14, or use

tools:overrideLibrary="android.support.v7.appcompat" to force usage (may

lead to runtime failures)

解决方案：build.gradle 中 minSdkVersion=11 改成 15

3、android studio，真机调试时安装出现两个图标

解决方案：使用了多个 java 文件完成作业，manifest.xml 中为 Main 以外的其他 java 文件添加了 <category android:name="android.intent.category.LAUNCHER" />，删掉即可。

4、生成 apk 的时候，报错如下

Execution failed for task ':app:lintVitalRelease'. > Could not resolve all files for configuration ':app:lintClassPath'.

.....

解决方案：https://blog.csdn.net/E_xiake/article/details/80001419

七、作业提升方案

当前用户信息均存在客户端本地，若有条件应当自己编写或购买一台云服务器，将项目关键信息（用户信息、消费记录等）放在云端，防止丢失。