## Activity的设计

本项目主要实现了水平2：

百度地图的展示和定位，以及读取/发送短信和添加，编辑，删除朋友

### 主界面设计：



### 添加朋友界面设计：





## 关键技术的实现

### 发送和接收短信

#### 发送短信

两种方法：第一种直接调用系统发送短信的功能，第二种调用系统短信接口。这里综合实验要求和可扩展性的考虑，我选择了第二种。

1. **public** **void** sendSMS(String phoneNumber,String message){
2. //获取短信管理器
3. android.telephony.SmsManager smsManager = android.telephony.SmsManager.getDefault();
4. //拆分短信内容（手机短信长度限制）
5. List<String> divideContents = smsManager.divideMessage(message);
6. **for** (String text : divideContents) {
7. smsManager.sendTextMessage(phoneNumber, **null**, text, sentPI, deliver
8. }
9. }

处理返回的发送状态：

1. String SENT\_SMS\_ACTION = "SENT\_SMS\_ACTION";
2. Intent sentIntent = **new** Intent(SENT\_SMS\_ACTION);
3. PendingIntent sentPI = PendingIntent.getBroadcast(context, 0, sentIntent, 0);
4. // register the Broadcast Receivers
5. context.registerReceiver(**new** BroadcastReceiver() {
6. @Override
7. **public** **void** onReceive(Context \_context, Intent \_intent) {
8. **switch** (getResultCode()) {
9. **case** Activity.RESULT\_OK:
10. Toast.makeText(context, "短信发送成功", Toast.LENGTH\_SHORT)  .show();
11. **break**;
12. **case** SmsManager.RESULT\_ERROR\_GENERIC\_FAILURE:
13. **break**;
14. **case** SmsManager.RESULT\_ERROR\_RADIO\_OFF:
15. **break**;
16. **case** SmsManager.RESULT\_ERROR\_NULL\_PDU:
17. **break**;
18. }
19. }
20. }, **new** IntentFilter(SENT\_SMS\_ACTION));

处理返回的接受状态：

1. String DELIVERED\_SMS\_ACTION = "DELIVERED\_SMS\_ACTION";
2. Intent deliverIntent = **new** Intent(DELIVERED\_SMS\_ACTION);
3. PendingIntent deliverPI = PendingIntent.getBroadcast(context, 0, deliverIntent, 0);
4. context.registerReceiver(**new** BroadcastReceiver() {
5. @Override
6. **public** **void** onReceive(Context \_context, Intent \_intent) {
7. Toast.makeText(context, "收信人已经成功接收", Toast.LENGTH\_SHORT).show();
8. }
9. }, **new** IntentFilter(DELIVERED\_SMS\_ACTION));

#### 接收短信

有两种思路：第一种是只读取刚刚接收的短信，第二种就是从系统短信库里面读取。这里综合实验要求以及个人隐私的考虑，我选取了第一种。

1. // 接收短信的广播
2. **class** MessageReceiver **extends** BroadcastReceiver {
3. @Override
4. **public** **void** onReceive(Context context, Intent intent) {
5. Bundle bundle = intent.getExtras();
6. Object[] pdus = (Object[]) bundle.get("pdus"); // 提取短信消息
7. SmsMessage[] messages = **new** SmsMessage[pdus.length];
8. **for** (**int** i = 0; i < messages.length; i++) {
9. messages[i] = SmsMessage.createFromPdu((**byte**[]) pdus[i]);
10. }
11. String address = messages[0].getOriginatingAddress(); // 获取发送方号码
12. String fullMessage = "";
13. **for** (SmsMessage message : messages) {
14. fullMessage += message.getMessageBody(); // 获取短信内容
15. }
16. textView.setText("address; " + address + ", message: " + fullMessage);
17. abortBroadcast();
18. }
19. }

### 百度地位的定位和标点连线

思路：利用百度地图提供API进行地图显示和定位，然后发送短信给列表里面的朋友，对方通过短信的形式回复自己的经纬度，然后提取信息中的经纬度进行标点和定位操作

#### 显示自己的定位：



1. /\*\*
2. \* 移到我的位置上
3. \* @param location
4. \*/
5. **private** **void** navigateTo(BDLocation location){
6. **if**(isFirstLocate){//第一次定位移到我的位置上
7. LatLng ll=**new** LatLng(location.getLatitude(),location.getLongitude());//获得当前位置的纬度和精度
8. MapStatusUpdate update=MapStatusUpdateFactory.newLatLng(ll);
9. baiduMap.animateMapStatus(update);
10. update=MapStatusUpdateFactory.zoomTo(16f);//缩放比例16倍
11. baiduMap.animateMapStatus(update);
12. isFirstLocate=**false**;//表示不是第一次定位了,防止多次调用animateMapStatus()方法
13. }
14. //将点标记在我的位置上
15. MyLocationData.Builder locationBuilder=**new** MyLocationData.Builder();
16. locationBuilder.latitude(location.getLatitude());
17. locationBuilder.longitude(location.getLongitude());//将当前位置封装起来
18. MyLocationData locationData=locationBuilder.build();//取出来并显示
19. baiduMap.setMyLocationData(locationData);
20. }

#### 发送短信：

首先我在列表里面添加了一位朋友：



点击“+”自定义的弹出框：

1. **public** **class** CreateFriendDialog **extends** Dialog {
2. Activity context;
3. **private** Button btn\_save;
4. **public** EditText text\_name;
5. **public** EditText text\_mobile;
7. **public** CreateFriendDialog(Activity context){
8. **super**(context);
9. **this**.context=context;
10. }
12. @Override
13. **protected** **void** onCreate(Bundle savedInstanceState){
14. **super**.onCreate(savedInstanceState);
15. **this**.setContentView(R.layout.add\_item);
17. text\_name=findViewById(R.id.add\_name);
18. text\_mobile=findViewById(R.id.add\_num);
19. Window dialogWindow = **this**.getWindow();
21. WindowManager m = context.getWindowManager();
22. Display d = m.getDefaultDisplay(); // 获取屏幕宽、高用
23. WindowManager.LayoutParams p = dialogWindow.getAttributes(); // 获取对话框当前的参数值
24. p.width = (**int**) (d.getWidth() \* 0.76); // 宽度设置为屏幕的0.8
25. dialogWindow.setAttributes(p);
26. // 根据id在布局中找到控件对象
27. btn\_save = (Button) findViewById(R.id.btn\_save);
29. // 为按钮绑定点击事件监听器
30. btn\_save.setOnClickListener(mClickListener);
32. **this**.setCancelable(**true**);//点击空白处消失
33. }
35. /\*\*
36. \* 点击事件
37. \*/
38. **private** View.OnClickListener mClickListener = **new** View.OnClickListener() {
39. @Override
40. **public** **void** onClick(View v) {
41. Friend\_database friendss=**new** Friend\_database();
42. **boolean** exit=**false**;
43. List<Friend\_database> friends=DataSupport.findAll(Friend\_database.**class**);
44. **switch** (v.getId()) {
45. **case** R.id.btn\_save:
46. //将数据存入数据库
47. //Toast.makeText(context,"我被点了",Toast.LENGTH\_SHORT).show();
48. //输入框里面的数据
49. String name\_content=text\_name.getText().toString();
50. String tele\_content=text\_mobile.getText().toString();
51. **if**(name\_content.length()!=0&&tele\_content.length()!=0){//输入的数据非空
52. **for**(Friend\_database friend:friends){
53. //找到姓名相同或者电话相同的数据了
54. **if**(name\_content.equals(friend.getName())){
55. friendss.setNum(tele\_content);
56. friendss.updateAll("name=?",name\_content);
57. exit=**true**;
58. **break**;
59. }**else** **if**(tele\_content.equals(friend.getNum())){
60. friendss.setName(name\_content);
61. friendss.updateAll("num=?",tele\_content);
62. exit=**true**;
63. **break**;
64. }
65. }
66. //所有循环都结束，还是没有找到相同的数据再添加
67. **if**(exit==**false**){
68. friendss.setNum(tele\_content);
69. friendss.setName(name\_content);
70. friendss.save();
71. **break**;
72. }
73. Toast.makeText(context,"插入成功,请点击右上角刷新",Toast.LENGTH\_SHORT).show();
74. }**else**{//输入的数据有为空的情况
75. Toast.makeText(context,"插入失败,请检查你的数据是否为空",Toast.LENGTH\_SHORT).show();
76. }
77. dismiss();
78. **break**;
79. }
80. }
81. };
82. }

回到主界面，点击右下角的按钮，发送短信



#### 收到短信：

朋友收到短信后，自动回复自己当前的经纬度



收发短信的代码上面已经展示，这里就不再赘述

#### 标点显示：



可以看出我们已经在地图中找到朋友的位置，并且连线

1. /\*\*
2. \* 标点
3. \*/
4. **private** **void** setMarker(Double latitude,Double longitude){
5. //清除地图上所有覆盖物
6. baiduMap.clear();
7. //定义Maker坐标点
8. LatLng point = **new** LatLng(latitude,longitude);
9. //构建Marker图标
10. BitmapDescriptor bitmap = BitmapDescriptorFactory.fromResource(R.drawable.green);
11. //构建MarkerOption，用于在地图上添加Marker
12. OverlayOptions option = **new** MarkerOptions().position(point).icon(bitmap);
13. //在地图上添加Marker，并显示
14. baiduMap.addOverlay(option);
15. }
16. /\*\*
17. \* 连线
18. \*/
19. **private** **void** connectLine(Double de\_latitude,Double de\_longitude){
20. List<LatLng> points = **new** ArrayList<LatLng>();
21. points.add(**new** LatLng(latitude,longitude));//起点为自己的位置
22. points.add(**new** LatLng(de\_latitude,de\_longitude));
23. List<Integer> colors = **new** ArrayList<>();
24. colors.add(Integer.valueOf(Color.BLUE));
25. OverlayOptions ooPolyline = **new** PolylineOptions().width(10)
26. .colorsValues(colors).points(points);
27. Polyline mPolyline = (Polyline) baiduMap.addOverlay(ooPolyline);
28. }

### 朋友的删除和添加

#### 添加：

自定义弹出框点击“确定”按钮的事件，首先判断输入的数据是否为空，为空则不插入，如果不为空，再判断插入的数据在数据库中是否存在，如果存在则更新该记录，如果不存在则执行插入操作

1. /\*\*
2. \* 点击事件
3. \*/
4. **private** View.OnClickListener mClickListener = **new** View.OnClickListener() {
5. @Override
6. **public** **void** onClick(View v) {
7. Friend\_database friendss=**new** Friend\_database();
8. **boolean** exit=**false**;
9. List<Friend\_database> friends=DataSupport.findAll(Friend\_database.**class**);
10. **switch** (v.getId()) {
11. **case** R.id.btn\_save:
12. //将数据存入数据库
13. //Toast.makeText(context,"我被点了",Toast.LENGTH\_SHORT).show();
14. //输入框里面的数据
15. String name\_content=text\_name.getText().toString();
16. String tele\_content=text\_mobile.getText().toString();
17. **if**(name\_content.length()!=0&&tele\_content.length()!=0){//输入的数据非空
18. **for**(Friend\_database friend:friends){
19. //找到姓名相同或者电话相同的数据了
20. **if**(name\_content.equals(friend.getName())){
21. friendss.setNum(tele\_content);
22. friendss.updateAll("name=?",name\_content);
23. exit=**true**;
24. **break**;
25. }**else** **if**(tele\_content.equals(friend.getNum())){
26. friendss.setName(name\_content);
27. friendss.updateAll("num=?",tele\_content);
28. exit=**true**;
29. **break**;
30. }
31. }
32. //所有循环都结束，还是没有找到相同的数据再添加
33. **if**(exit==**false**){
34. friendss.setNum(tele\_content);
35. friendss.setName(name\_content);
36. friendss.save();
37. **break**;
38. }
39. Toast.makeText(context,"插入成功,请点击右上角刷新",Toast.LENGTH\_SHORT).show();
40. }**else**{//输入的数据有为空的情况
41. Toast.makeText(context,"插入失败,请检查你的数据是否为空",Toast.LENGTH\_SHORT).show();
42. }
43. dismiss();
44. **break**;
45. }
46. }
47. };

#### 删除：

由于我没有做敌人功能以及用户详情功能，所以我这里简单处理，只需要点击列表中对应的朋友，即可删除

1. //点击删除功能
2. listView.setOnItemClickListener(**new** AdapterView.OnItemClickListener() {
3. @Override
4. **public** **void** onItemClick(AdapterView<?> parent, View view, **int** position, **long** id) {
5. Friend friend=friendList.get(position);
6. DataSupport.deleteAll(Friend\_database.**class**,"name=? and num=?",friend.getName(),friend.getNum());
7. refreshList();
8. }
9. });

## 测试及使用说明

### 使用说明

参见**2.3.2百度地位的定位和标点连线**，这里详细的说明了从定位到添加朋友，发送短信再到定位连线的过程

### 测试

地图显示正常，定位正常，添加朋友功能正常，收发短信功能正常

不足之处，该程序在虚拟机上无法定位，但是在真机上功能一切正常。在网上查了很多原因，发现应该是架构的问题，虚拟机架构从x86换成arm架构即可正常定位，但我认为本程序的最终目的就是为了在真机上使用，所以我就没有深入研究