

Android----通讯录

软件名称:

通讯录

完成人:

谭增明

学号:

20192131002

完成时间:

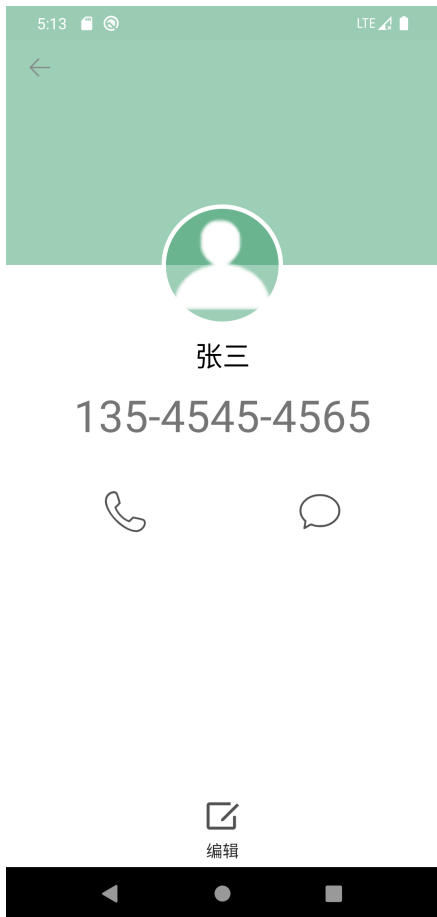
2021-12-10

界面设计:

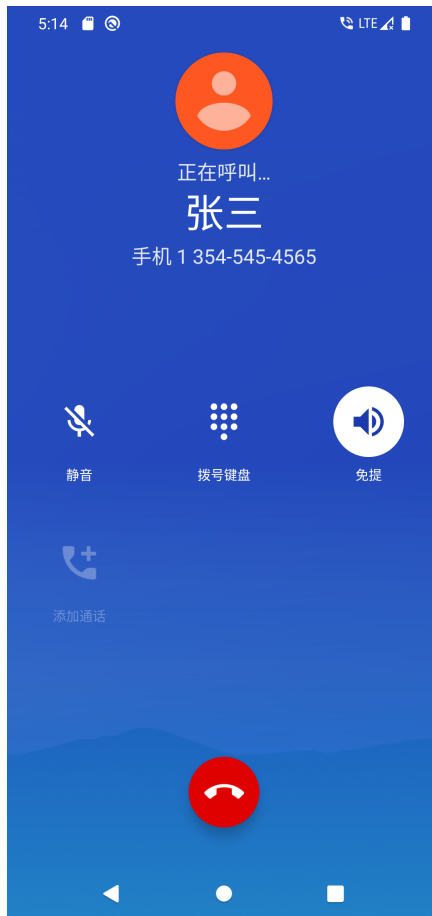
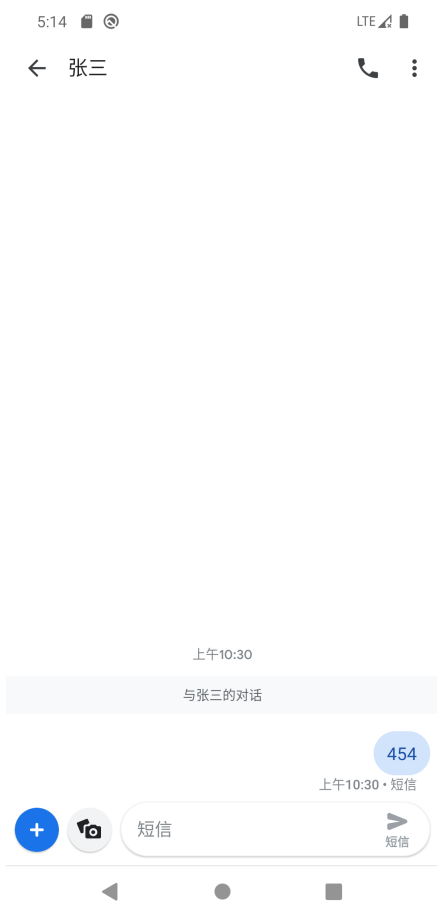
1.初始界面，读取通讯录



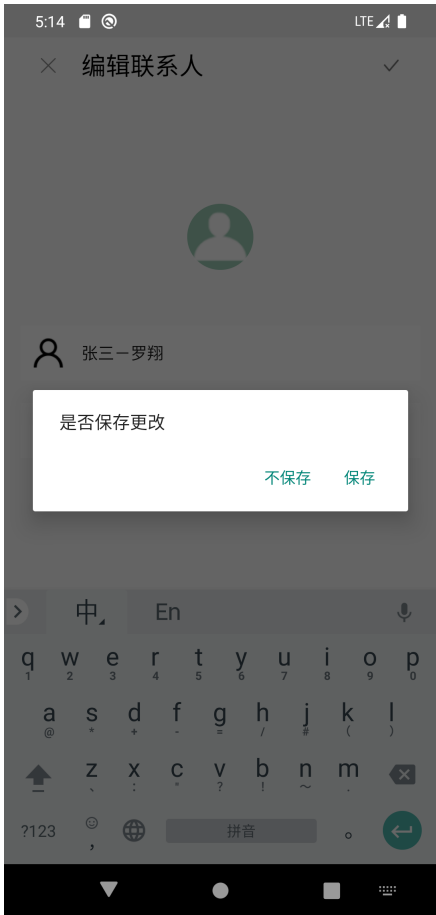
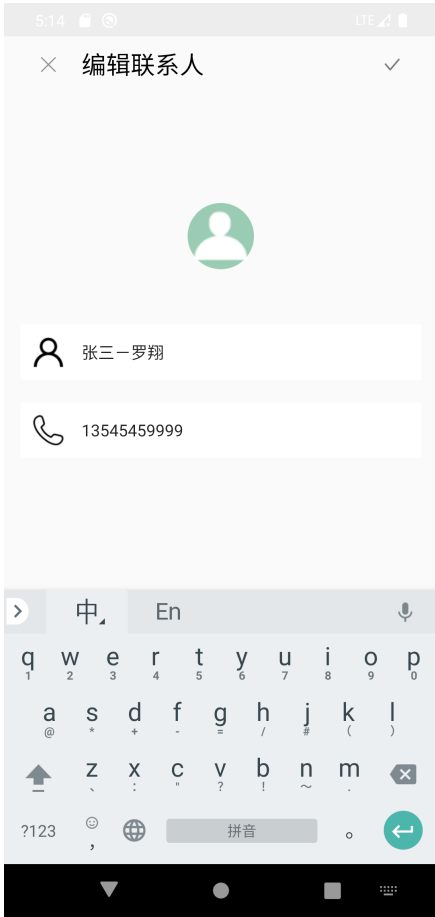
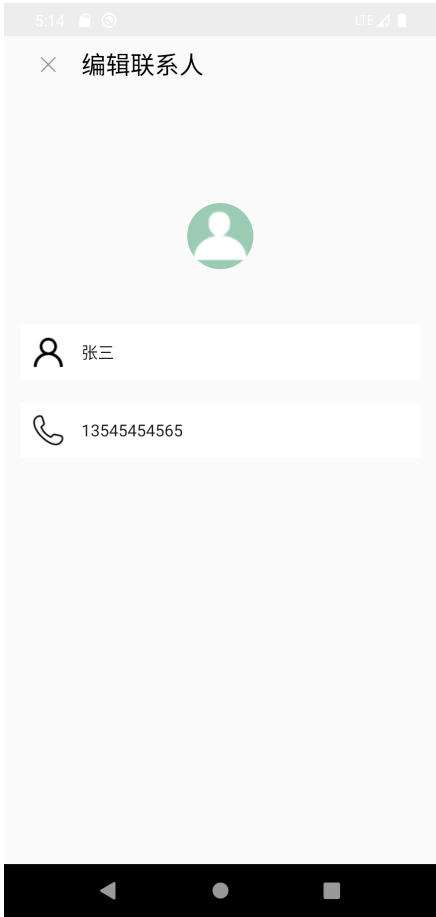
2.显示联系人姓名，电话，可编辑信息按钮，拨打电话按钮和发送信息按钮

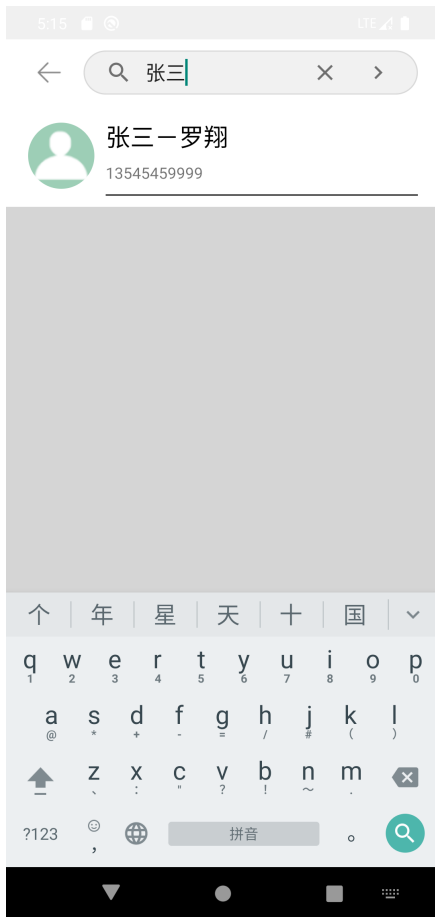


3.发送信息和拨打电话

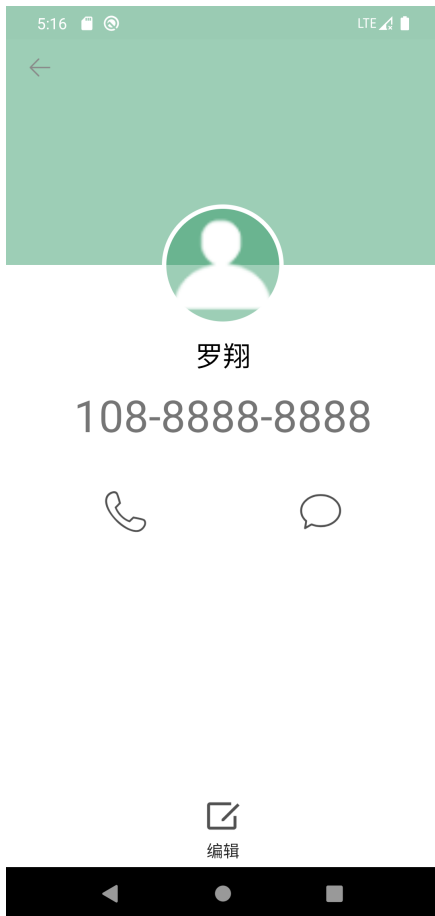
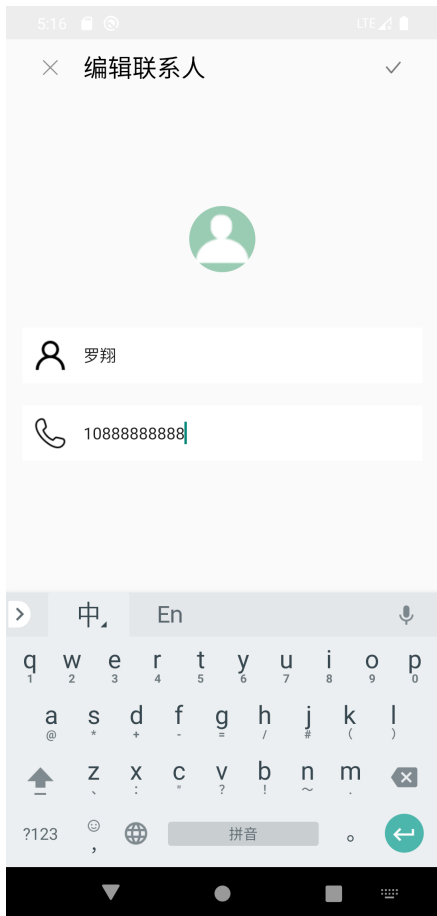


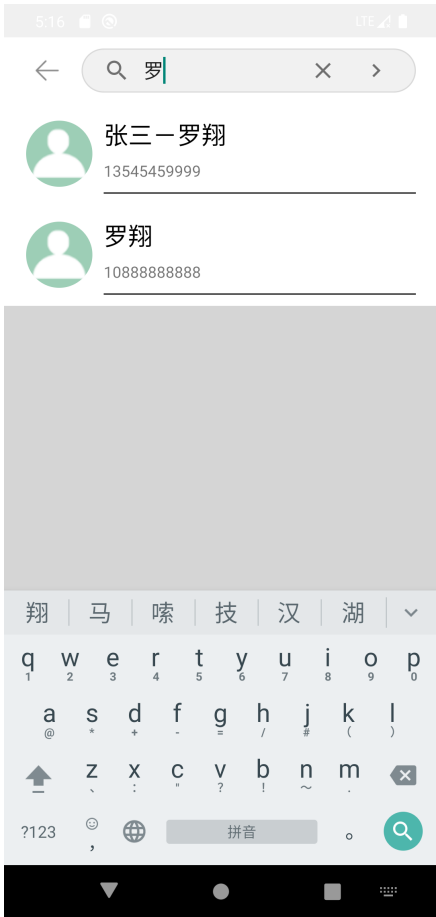
4.编辑个人信息



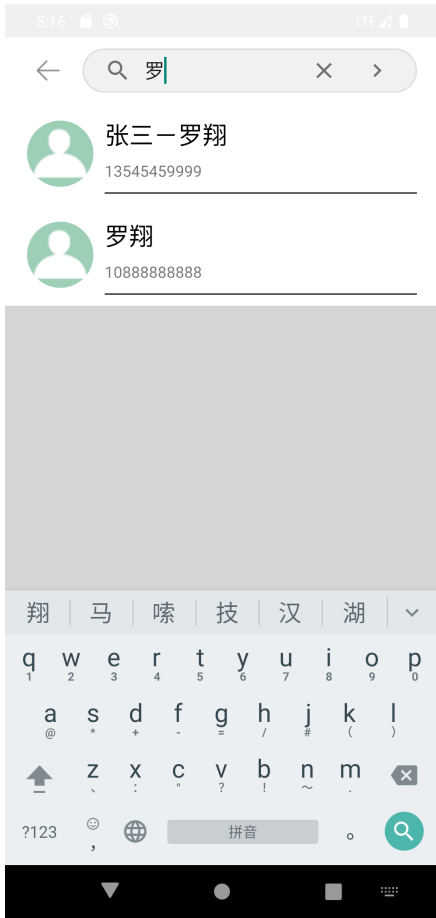
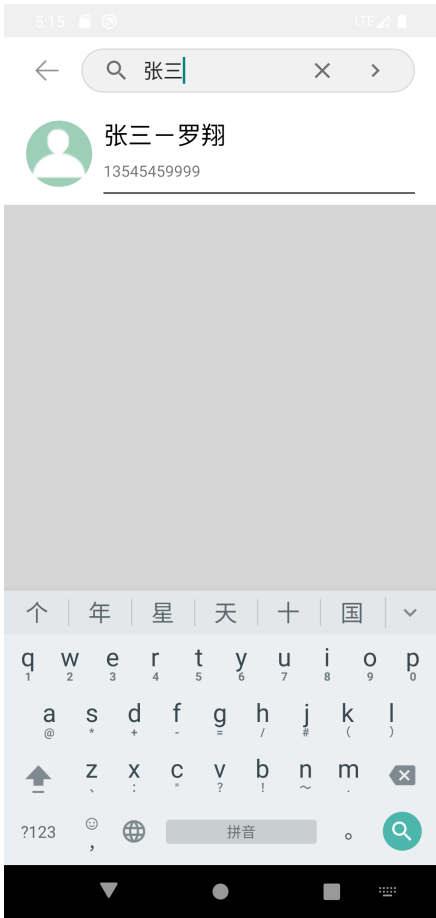


5. 添加联系人

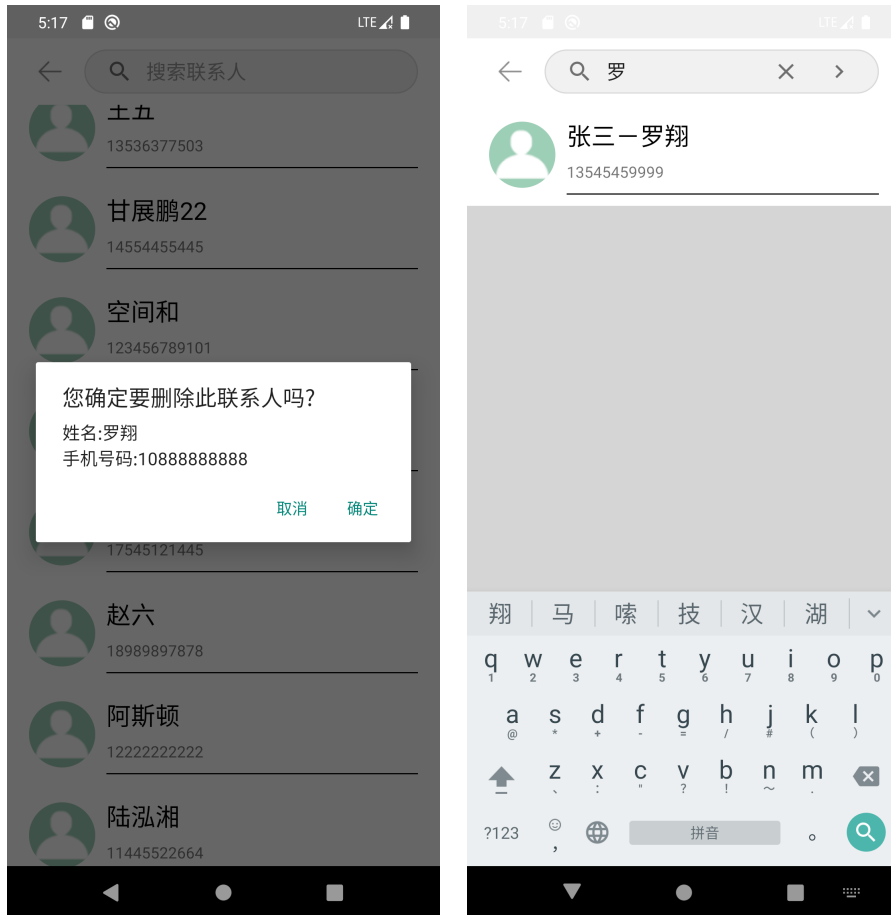




6.查询联系人



7.长按删除联系人

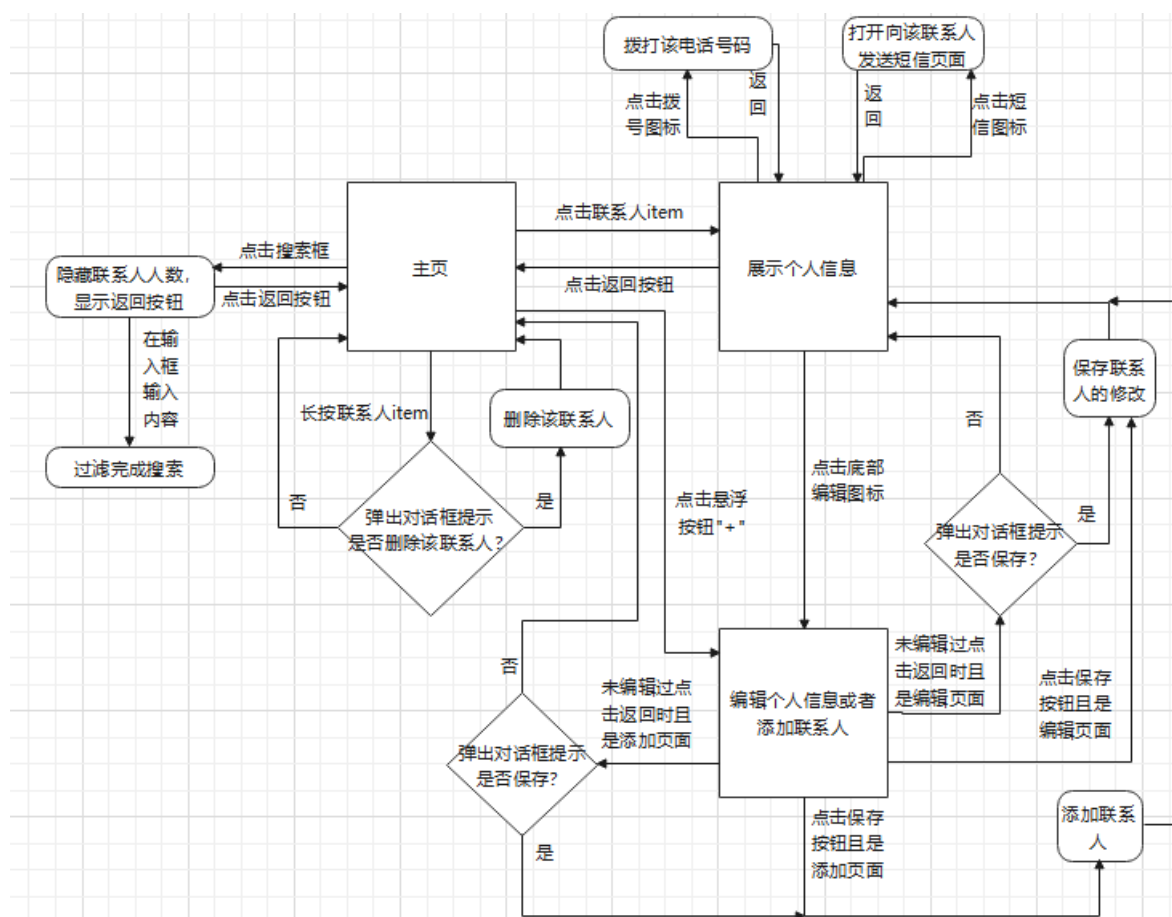


软件操作流程：

文字流程：

1. 主页面点击悬浮按钮“+”，进入添加联系人页面。如果未编辑过点击退出时，直接返回主页面；如果编辑过了，点击保存按钮 或者 点击退出时会弹出对话框提示是否保存该联系人并选择保存时，会保存该联系人到通讯录，然后转到展示该联系人信息的页面
2. 主页面点击搜索框时可进行查询，上方的“联系人\n12个联系人”会隐藏起来，同时会浮现一个return按钮并禁止滚动
3. 主页面点击某个联系人时，会转到展示该联系人信息的页面，在该页面可点击拨打电话、发送信息、编辑信息，点击拨打电话页面时会拨打该联系人号码，点击发送信息时会转到发送信息页面，点击编辑信息按钮时会转到编辑个人信息页面，当未编辑时，点击退出时可直接退出，编辑了后可选择点击保存按钮或者点击退出按钮时弹出对话框时选择保存可保存信息并返回到个人信息页面
4. 长按某个联系人的item时，会弹出对话框窗口提示是否选择删除该联系人，若选择是会删除通讯录该联系人

流程图流程：



难点:

1. 《第一行代码》里的recyclerView的adapter的点击事件是在adapter里面写的，但是我要实现的是长按删除功能，要将itemView的长按点击事件回调到activity里边实现，否则调用不了相应的删除函数
2. 通讯录的更新、删除、添加问题

解决方案:

1. 在网上找资料，但是kotlin的资源很少，找了些时间找不到合适的，最后找到
2. 《第一行代码》书本里目前我只看到有查询通讯录全部人的姓名、电话的代码，没找到有关通讯录的删除、更新、添加的代码，于是去网上找，但是网上资料五花八门，最后是找到一篇比较靠谱的，其实首先先了解通讯录有关的表，而通讯录删除、更新、添加的实现其实都要先找到哪个id，然后借用id的唯一性在通讯录有关表里边完成删除、更新、添加，但是需要注意的是，需要动态申请权限

不足之处:

实现了老师要求的必须功能和额外功能，暂时并没有发现不足

今后设想:

可以将通讯记录和email等信息也添加进来

代码设计:

Activity

MainActivity

```
1 package com.example.maillist
2
3 import android.annotation.SuppressLint
4 import android.content.Intent
5 import android.content.pm.PackageManager
6 import android.database.Cursor
7 import android.graphics.Color
8 import android.net.Uri
9 import android.os.Build
10 import androidx.appcompat.app.AppCompatActivity
11 import android.os.Bundle
12 import android.provider.ContactsContract
13 import
    android.provider.ContactsContract.CommonDataKinds.StructuredName.DISPLAY_NAME
14 import android.text.TextUtils
15 import android.util.Log
16 import android.view.View
17 import android.view.WindowManager
18 import android.view.inputmethod.EditorInfo
19 import android.widget.SearchView
20 import android.widget.Toast
21 import androidx.annotation.RequiresApi
22 import androidx.appcompat.app.AlertDialog
23 import androidx.core.app.ActivityCompat
24 import androidx.core.content.ContextCompat
25 import androidx.core.view.ViewCompat.canScrollVertically
26 import androidx.recyclerview.widget.LinearLayoutManager
27 import kotlinx.android.synthetic.main.activity_main.*
28 import java.security.AccessController.getContext
29
30
31
32 public var search_text:String = ""
33 public var __name__ = ""
34 public var __phone__ = ""
35 class MainActivity : AppCompatActivity() {
36     private val mailList = arrayListOf<Mail>()
37     public lateinit var adapter:MailAdapter
38     public lateinit var layoutManager:LinearLayoutManager
39     var current_activty_ = this
40     @RequiresApi(Build.VERSION_CODES.M)
41     override fun onCreate(savedInstanceState: Bundle?) {
42         super.onCreate(savedInstanceState)
43         setContentView(R.layout.activity_main)
44
45         window.addFlags(WindowManager.LayoutParams.FLAG_DRAWS_SYSTEM_BAR_BACKGROUN
46         DS)
47
48         window.setStatusBarColor(this.getColor(R.color.background_))
49         supportActionBar?.hide()
50         searchView.setImeOptions(EditorInfo.IME_ACTION_SEARCH)
51         searchView.isSubmitButtonEnabled = true
52         searchView.isFocusable = true
```



```

51
52         title_activity.onFocusChangeListener = View.OnFocusChangeListener {
53             v, hasFocus ->
54                 if (hasFocus) {
55                     val layoutManager = LinearLayoutManager(this)
56                     recyclerView.layoutManager = layoutManager
57                 } else {
58                     title_top.visibility = View.GONE
59                     return_1.visibility = View.VISIBLE
60                     fab_plus.visibility = View.GONE
61
62                 recyclerView.setBackgroundColor(this.resources.getColor(R.color.back))
63                 layoutManager= object : LinearLayoutManager(this) {
64                     override fun canScrollVertically(): Boolean {
65                         return false
66                     }
67                 }
68                 recyclerView.layoutManager = layoutManager
69             }
70
71         return_1.setOnClickListener {
72             title_activity.clearFocus()
73             return_1.visibility = View.GONE
74             title_top.visibility = View.VISIBLE
75
76         recyclerView.setBackgroundColor(this.resources.getColor(R.color.white))
77         fab_plus.visibility = View.VISIBLE
78         val layoutManager = LinearLayoutManager(this)
79         recyclerView.layoutManager = layoutManager
80         searchView.setQuery("", false);
81     }
82
83     // 设置搜索文本监听
84     searchView.setOnQueryTextListener(object :
85     SearchView.OnQueryTextListener {
86         // 当点击搜索按钮时触发该方法
87         override fun onQueryTextSubmit(query: String): Boolean {
88             adapter.filter.filter(query)
89             search_text = query
90             Log.d("app1123", search_text)
91             searchView.clearFocus()
92             recyclerView.requestFocus()
93             return false
94         }
95
96         // 当搜索内容改变时触发该方法
97         override fun onQueryTextChange(newText: String): Boolean {
98             //do something
99             //当没有输入任何内容的时候清除结果，看实际需求
100             search_text = newText
101             if (TextUtils.isEmpty(newText)){
102                 adapter.filter.filter("")
103             } else{
104                 adapter.filter.filter(newText)
105             }
106             return false
107         }
108     })

```

```

105         }
106
107
108     })
109
110     fab_plus.setOnClickListener {
111         add_OR_change = 0
112         val intent_ = Intent(this,edit_information::class.java)
113         startActivity(intent_)
114     }
115
116 }
117
118 override fun onResume() {
119     super.onResume()
120
121     init()
122     layoutManager = LinearLayoutManager(this)
123     recyclerView.layoutManager = layoutManager
124     adapter = MailAdapter(mailList)
125
126     adapter.setOnItemClickListener(object :
MailAdapter.ItemListener {
127         override fun onItemLongClick(position: Int) {
128             val mailItem: Mail = mailList.get(position)
129             var i=0
130             var num=position+1
131             while(i<mailList.size){
132
133                 if(mailList[i].name.contains(search_text)||mailList[i].phone.contains(sear
ch_text)){
134                     num -= 1
135                     if(num==0){
136                         break
137                     }
138                     i += 1
139                 }
140                 val pos:Int = i
141                 val mail = mailList[pos]
142                 val id = mailList[pos].id
143                 AlertDialog.Builder(current_activty_).apply {
144                     setTitle("您确定要删除此联系人吗?")
145                     val content:String = "姓名:" + mail.name + "\n手机号码:"
+ mail.phone
146                     setMessage(content)
147                     setPositiveButton("确定") {dialog, which->
148                         mailList.removeAt(pos)
149                         adapter.notifyItemRemoved(pos)
150                         DeleteContacts(mail.name,mail.phone)
151                     }
152                     setNegativeButton("取消") {dialog, which->}
153                     show()
154                 }
155             }
156         })
157
158

```

```

159         adapter.filter.filter(search_text)
160         recyclerView.adapter = adapter
161         updatePeople()
162         clearFource()
163         SoftKeyboardListener.setListener(this, object :
164             SoftKeyboardListener.OnSoftKeyboardChangeListener {
165             override fun keyboardShow(height: Int) {}
166             override fun keyboardHide(height: Int) {
167                 clearFource()
168             }
169         })
170     }
171
172     private fun updatePeople(){
173         val num = number_of_people.text.toString()
174         var i = 0
175         val length_num=num.length
176         while(i<length_num){
177             if(num.subSequence(0,i+1).toString().contains("↑"))
178                 break
179             i += 1
180         }
181         Log.d("app1123","i:"+i.toString())
182         val last_string = num.subSequence(i,length_num).toString()
183         number_of_people.setText(mailList.size.toString()+last_string)
184     }
185
186     private fun clearFource() {
187         if (searchview != null) {
188             searchview.clearFocus()
189         }
190         title_activity.setFocusable(true)
191         title_activity.setFocusableInTouchMode(true)
192         title_activity.requestFocus()
193     }
194
195     private fun init(){
196         mailList.clear()
197
198         if(ContextCompat.checkSelfPermission(this,android.Manifest.permission.READ
199         _CONTACTS) != PackageManager.PERMISSION_GRANTED) {
200             ActivityCompat.requestPermissions(this,
201             arrayOf(android.Manifest.permission.READ_CONTACTS),1)
202         } else{
203             readContacts()
204         }
205     }
206
207     override fun onRequestPermissionsResult(
208         requestCode: Int,
209         permissions: Array<out String>,
210         grantResults: IntArray
211     ) {
212         super.onRequestPermissionsResult(requestCode, permissions,
213         grantResults)
214         when (requestCode) {
215             1 -> {

```

```

212         if (grantResults.isNotEmpty() && grantResults[0] ==
PackageManager.PERMISSION_GRANTED) {
213             readContacts()
214         } else {
215             Toast.makeText(this, "You denied the permission",
Toast.LENGTH_SHORT).show()
216         }
217     }
218     2 -> {
219         if (grantResults.isNotEmpty() && grantResults[1] ==
PackageManager.PERMISSION_GRANTED) {
220             deleteContacts()
221         } else {
222             Toast.makeText(this, "You denied the permission",
Toast.LENGTH_SHORT).show()
223         }
224     }
225 }
226 }
227
228 @SuppressWarnings("Range")
229 private fun readContacts() {
230
    contentResolver.query(ContactsContract.CommonDataKinds.Phone.CONTENT_URI, n
ull, null, null, ContactsContract.CommonDataKinds.Phone.DISPLAY_NAME)?.apply {
231         var l = 0
232         while( moveToNext()){
233             val ID = getString(0)
234             val displayName =
getString(getColumnIndex(ContactsContract.CommonDataKinds.Phone.DISPLAY_NAM
E))
235             val number =
getString(getColumnIndex(ContactsContract.CommonDataKinds.Phone.NUMBER))
236             mailList.add(Mail(displayName, number, l++))
237         }
238         close()
239     }
240 }
241
242 private fun deleteContacts(){
243     var uri =
Uri.parse("content://com.android.contacts/data/phones/filter/${__phone__}
_")
244     var cursor: Cursor? = contentResolver.query(uri,
arrayOf(ContactsContract.Data.CONTACT_ID), "display_name=?",
arrayOf(__name__), null)
245     uri = Uri.parse("content://com.android.contacts/raw_contacts")
246     if(cursor != null) {
247         while(cursor.moveToNext()) {
248             val id_ = cursor?.getInt(0).toString()
249             Log.d("app11234", "id = "+id_)
250             uri =
Uri.parse("content://com.android.contacts/raw_contacts")
251             contentResolver.delete(uri, "_id=?", arrayOf(id_))
252             uri = Uri.parse("content://com.android.contacts/data")
253             contentResolver.delete(uri, "raw_contact_id=?",
arrayOf(id_))
254         }

```

```

255     }
256 }
257
258 public fun DeleteContacts(name____:String,phone____:String){
259     ____name____ = name____
260     ____phone____ = phone____
261
262     if(ContextCompat.checkSelfPermission(current_activity_,android.Manifest.permission.WRITE_CONTACTS) != PackageManager.PERMISSION_GRANTED) {
263         ActivityCompat.requestPermissions(current_activity_,
264             arrayOf(android.Manifest.permission.WRITE_CONTACTS),2)
265     } else{
266         deleteContacts()
267     }
268 }
269
270 fun px2dp(pxVal: Float): Float {
271     val scale: Float = this.getResources().getDisplayMetrics().density
272     return pxVal / scale
273 }

```

person.kt

```

1  package com.example.maillist
2
3  import android.annotation.SuppressLint
4  import android.content.Intent
5  import android.content.pm.PackageManager
6  import android.net.Uri
7  import android.os.Build
8  import androidx.appcompat.app.AppCompatActivity
9  import android.os.Bundle
10 import android.telephony.SubscriptionManager
11 import android.util.Log
12 import android.view.WindowManager
13 import android.widget.Toast
14 import androidx.core.app.ActivityCompat
15 import androidx.core.content.ContextCompat
16 import kotlinx.android.synthetic.main.person.*
17 import android.content.ContentResolver
18 import android.content.ContentValues
19 import android.database.Cursor
20 import android.provider.ContactsContract
21 import java.lang.Exception
22
23
24 class person : AppCompatActivity() {
25     public lateinit var name_:String
26     public lateinit var phone_:String
27     public var id_ = 0
28     @SuppressLint("ServiceCast")
29     override fun onCreate(savedInstanceState: Bundle?) {
30         super.onCreate(savedInstanceState)

```

```

31         setContentView(R.layout.person)
32
33         window.addFlags(WindowManager.LayoutParams.FLAG_DRAWS_SYSTEM_BAR_BACKGROUN
34         DS);
35         window.setStatusBarColor(this.resources.getColor(R.color.yellow))
36         supportActionBar?.hide()
37         Log.d("app11234", "num: "+IS_change_edit.toString())
38
39         return_2.setOnClickListener {
40             finish()
41         }
42
43         edit_person_bottom.setOnClickListener {
44             val intent_=Intent(this,edit_information::class.java)
45             intent_.putExtra("name",name_.toString())
46             intent_.putExtra("phone",phone_.toString())
47             intent_.putExtra("id",id_)
48             add_OR_change = 1
49             startActivity(intent_)
50         }
51
52         phoneCall.setOnClickListener {
53
54             if(ContextCompat.checkSelfPermission(this,android.Manifest.permission.CALL
55             _PHONE) != PackageManager.PERMISSION_GRANTED) {
56                 ActivityCompat.requestPermissions(this,
57                 arrayOf(android.Manifest.permission.CALL_PHONE),1)
58             } else{
59                 call(phone_)
60             }
61         }
62
63         phonesend.setOnClickListener {
64             sendsMS("",phone_)
65         }
66     }
67
68     override fun onResume() {
69         super.onResume()
70         if(IS_change_edit==1){
71             name_ = edit_name_
72             phone_ = edit_phone_
73             IS_change_edit = 0
74         } else{
75             name_= intent.getStringExtra("name").toString()
76             phone_ = intent.getStringExtra("phone").toString()
77         }
78         name.setText(name_.toString())
79         val phone_text=phone_?.subSequence(0,3).toString()+"-
80         "+phone_?.subSequence(3,7).toString()+"-
81         "+phone_?.subSequence(7,11).toString()
82         phone.setText(phone_text)
83     }
84
85     override fun onRequestPermissionsResult(
86         requestCode: Int,
87         permissions: Array<out String>,
88         grantResults: IntArray

```

```

82     ) {
83         super.onRequestPermissionsResult(requestCode, permissions,
grantResults)
84         when (requestCode) {
85             1 -> {
86                 if (grantResults.isNotEmpty() && grantResults[0] ==
PackageManager.PERMISSION_GRANTED) {
87                     call(phone_)
88                 } else {
89                     Toast.makeText(this, "You denied the permission",
Toast.LENGTH_SHORT).show()
90                 }
91             }
92         }
93     }
94
95     //打电话
96     private fun call(phone_number:String){
97         try {
98             val intent_ = Intent(Intent.ACTION_CALL)
99             intent_.data = Uri.parse("tel:"+phone_number)
100             startActivity(intent_)
101         } catch (e:SecurityException){
102             e.printStackTrace()
103             Log.d("app11234","error test")
104         }
105     }
106
107     //发送短信
108     private fun sendSMS(smsBody: String,phone_number: String) {
109         val smsToUri = Uri.parse("smsto:"+phone_number)
110         val intent = Intent(Intent.ACTION_SENDTO, smsToUri)
111         intent.putExtra("sms_body", smsBody)
112         startActivity(intent)
113     }
114
115     fun testDelete() {
116         //根据姓名求id
117         var uri = Uri.parse("content://com.android.contacts/raw_contacts")
118         val resolver = contentResolver
119         val cursor: Cursor? =
120             resolver.query(uri!!,
arrayOf(ContactsContract.Contacts.Data._ID), "display_name=?,number=?",
arrayOf(name_,phone_), null)
121         if (cursor != null) {
122             if (cursor.moveToFirst()) {
123                 val id: Int = cursor.getInt(0)
124                 //根据id删除data中的相应数据
125                 resolver.delete(uri, "display_name=?,number=?",
arrayOf(name_,phone_))
126                 uri = Uri.parse("content://com.android.contacts/data")
127                 resolver.delete(uri, "raw_contact_id=?",
arrayOf(id.toString() + ""))
128             }
129         }
130     }
131
132 }

```

edit_information.kt

```
1  package com.example.maillist
2
3  import android.content.Context
4  import android.content.Intent
5  import androidx.appcompat.app.AppCompatActivity
6  import android.os.Bundle
7  import android.view.View
8  import android.view.WindowManager
9  import android.view.inputmethod.InputMethodManager
10 import androidx.appcompat.app.AlertDialog
11 import kotlinx.android.synthetic.main.edit_information.*
12 import java.util.*
13 import android.widget.Toast
14
15 import android.provider.ContactsContract.CommonDataKinds.Phone
16
17 import android.provider.ContactsContract.CommonDataKinds.StructuredName
18
19 import android.content.ContentUris
20
21 import android.provider.ContactsContract.RawContacts
22
23 import android.content.ContentValues
24
25 import android.provider.ContactsContract.Data;
26 import android.content.pm.PackageManager
27 import android.util.Log
28 import androidx.core.app.ActivityCompat
29 import androidx.core.content.ContextCompat
30 import java.lang.Exception
31 import android.app.Activity
32 import android.database.Cursor
33 import android.net.Uri
34 import android.provider.ContactsContract
35 import kotlinx.android.synthetic.main.person.*
36 import java.lang.reflect.Field
37 import java.lang.reflect.InvocationTargetException
38
39
40 public var add_OR_change = 0
41 public var edit_name_ = ""
42 public var edit_phone_ = ""
43 public var IS_change_edit = 0
44 class edit_information : AppCompatActivity() {
45     private var IS_changed = 0
46     public lateinit var name_:String
47     public lateinit var phone_:String
48     public lateinit var _name_:String
49     public lateinit var _phone_:String
50     private val current_activity = this
51     override fun onCreate(savedInstanceState: Bundle?) {
52         super.onCreate(savedInstanceState)
```



```

53         setContentView(R.layout.edit_information)
54         supportActionBar?.hide()
55
56         window.addFlags(WindowManager.LayoutParams.FLAG_DRAWS_SYSTEM_BAR_BACKGROUN
57         DS);
58         window.setStatusBarColor(this.resources.getColor(R.color.back))
59         IS_change_edit = 0
60
61         if(add_OR_change==1){
62             name_=intent.getStringExtra("name").toString()
63             phone_ = intent.getStringExtra("phone").toString()
64             val id_ = intent.getIntExtra("id",0)
65             name_information.setText(name_)
66             phone_information.setText(phone_)
67             return_3.setOnClickListener {
68                 if(IS_changed==0){
69                     finish()
70                 } else{
71                     AlertDialog.Builder(this).apply {
72                         setMessage("是否保存更改")
73                         setPositiveButton("保存") {
74                             dialog,which->
75
76                             _name_ = name_information.text.toString()
77                             _phone_ = phone_information.text.toString()
78                             edit_name_ = _name_
79                             edit_phone_ = _phone_
80                             IS_change_edit = 1
81
82                             if(ContextCompat.checkSelfPermission(this.context,android.Manifest.permission.WRITE_CONTACTS) != PackageManager.PERMISSION_GRANTED) {
83
84                                 ActivityCompat.requestPermissions(current_activty,
85                                 arrayOf(android.Manifest.permission.WRITE_CONTACTS),1)
86                             } else{
87                                 update(_name_,_phone_)
88                                 Toast.makeText(this.context,"更新",Toast.LENGTH_SHORT).show()
89                             }
90                             finish()
91                         }
92                         setNegativeButton("不保存") {
93                             dialog, which->
94                             finish()
95                         }
96                         show()
97                     }
98                 }
99             }
100
101             name_information.onFocusChangeListener =
102             View.OnFocusChangeListener { v, hasFocus ->
103                 if (hasFocus) {
104                     save.setVisibility(View.VISIBLE);
105                     IS_changed = 1
106                     // 此处为得到焦点时的处理内容
107                 } else {
108                     save.setVisibility(View.GONE);

```

```

103         // 此处为失去焦点时的处理内容
104     }
105 }
106
107     phone_information.onFocusChangeListener =
View.OnFocusChangeListener { v, hasFocus ->
108         if (hasFocus) {
109             save.setVisibility(View.VISIBLE);
110             IS_changed = 1
111             // 此处为得到焦点时的处理内容
112         } else {
113             save.setVisibility(View.GONE);
114             // 此处为失去焦点时的处理内容
115         }
116     }
117
118     save.setOnClickListener {
119         IS_changed = 0
120         name_information.clearFocus()
121         phone_information.clearFocus()
122         val imm: InputMethodManager =
this.getSystemService(Context.INPUT_METHOD_SERVICE) as InputMethodManager
123
124         imm.hideSoftInputFromWindow(this.getWindow().getDecorView().getWindowToken
(), 0)
125
126         save.visibility = View.GONE
127
128         _name_ = name_information.text.toString()
129         _phone_ = phone_information.text.toString()
130
131         if(ContextCompat.checkSelfPermission(this,android.Manifest.permission.WRIT
E_CONTACTS) != PackageManager.PERMISSION_GRANTED) {
132             ActivityCompat.requestPermissions(this,
133             arrayOf(android.Manifest.permission.WRITE_CONTACTS),1)
134         } else{
135             update(_name_,_phone_)
136         }
137         edit_name_ = _name_
138         edit_phone_ = _phone_
139         IS_change_edit = 1
140         finish()
141     }
142 }
143 else {
144     return_3.setOnClickListener {
145         if(IS_changed==0){
146             finish()
147         } else{
148             AlertDialog.Builder(this).apply {
149                 setMessage("是否保存更改")
150                 setPositiveButton("保存") {
151                     dialog,which->
152                         name_ = name_information.text.toString()
153                         phone_ = phone_information.text.toString()
154                     try {
155
156             if(ContextCompat.checkSelfPermission(this.context,android.Manifest.permiss
ion.WRITE_CONTACTS) != PackageManager.PERMISSION_GRANTED) {

```

```

152     ActivityCompat.requestPermissions(current_activy,
    arrayOf(android.Manifest.permission.WRITE_CONTACTS),1)
153         } else{
154             addContact(name_,phone_)
155         }
156     } catch (e:Exception){
157         e.printStackTrace()
158         Log.d("app11234","error ADD")
159     }
160
161     val intent_ =
Intent(this.context,person::class.java)
162         intent_.putExtra("name",name_)
163         intent_.putExtra("phone",phone_)
164         startActivity(intent_)
165         finish()
166     }
167     setNegativeButton("不保存") {
168         dialog, which->
169         finish()
170     }
171     show()
172 }
173 }
174 }
175
176     name_information.onFocusChangeListener =
View.OnFocusChangeListener { v, hasFocus ->
177         if (hasFocus) {
178             save.setVisibility(View.VISIBLE);
179             IS_changed = 1
180             // 此处为得到焦点时的处理内容
181         } else {
182             save.setVisibility(View.GONE);
183             // 此处为失去焦点时的处理内容
184         }
185     }
186
187     phone_information.onFocusChangeListener =
View.OnFocusChangeListener { v, hasFocus ->
188         if (hasFocus) {
189             save.setVisibility(View.VISIBLE);
190             IS_changed = 1
191             // 此处为得到焦点时的处理内容
192         } else {
193             save.setVisibility(View.GONE);
194             // 此处为失去焦点时的处理内容
195         }
196     }
197
198     save.setOnClickListener {
199         IS_changed = 0
200         name_ = name_information.text.toString()
201         phone_ = phone_information.text.toString()
202
203         try {

```

```

204         if(ContextCompat.checkSelfPermission(this, android.Manifest.permission.WRITE
E_CONTACTS) != PackageManager.PERMISSION_GRANTED) {
205             ActivityCompat.requestPermissions(this,
arrayOf(android.Manifest.permission.WRITE_CONTACTS), 1)
206         } else {
207             addContact(name_, phone_)
208         }
209     } catch (e: Exception) {
210         e.printStackTrace()
211         Log.d("app11234", "error ADD")
212     }
213     val intent_ = Intent(this, person::class.java)
214     intent_.putExtra("name", name_)
215     intent_.putExtra("phone", phone_)
216     startActivity(intent_)
217     finish()
218 }
219 }
220 }
221
222
223 override fun onRequestPermissionsResult(
224     requestCode: Int,
225     permissions: Array<out String>,
226     grantResults: IntArray
227 ) {
228     super.onRequestPermissionsResult(requestCode, permissions,
grantResults)
229     when (requestCode) {
230         1 -> {
231             if (grantResults.isNotEmpty() && grantResults[0] ==
PackageManager.PERMISSION_GRANTED) {
232                 addContact(name_, phone_)
233             } else {
234                 Toast.makeText(this, "You denied the permission",
Toast.LENGTH_SHORT).show()
235             }
236         }
237
238         2 -> {
239             if (grantResults.isNotEmpty() && grantResults[1] ==
PackageManager.PERMISSION_GRANTED) {
240                 update(_name_, _phone_)
241                 // Toast.makeText(this, "更新", Toast.LENGTH_SHORT).show()
242             } else {
243                 Toast.makeText(this, "You denied the permission",
Toast.LENGTH_SHORT).show()
244             }
245         }
246     }
247 }
248
249 fun addContact(name: String?, phoneNumber: String?) {
250     val values = ContentValues()
251     //ContactsContract.RawContacts.CONTENT_URI 查询的URI
252     //ContactsContract.RawContacts.CONTENT_URI 查询的URI
253     val uri = contentResolver.insert(RawContacts.CONTENT_URI, values)

```

```

254     Log.d("app11234",uri.toString()+"")
255     val rawContentId = ContentUri.parseId(uri!!) //获取到插入的id
256
257     if (!name.equals("")) {
258         values.clear()
259         //插入姓名
260         values.put(Data.RAW_CONTACT_ID, rawContentId) //插入的id
261         values.put(Data.MIMETYPE, StructuredName.CONTENT_ITEM_TYPE) //
插入name的类型
262         values.put(StructuredName.GIVEN_NAME, name) //放入实际值
263         contentResolver.insert(Data.CONTENT_URI, values)
264     }
265
266     //插入电话
267     if (!phoneNumber.equals("")) {
268         values.put(Data.RAW_CONTACT_ID, rawContentId) //插入的id
269         values.put(Data.MIMETYPE, Phone.CONTENT_ITEM_TYPE)
270         values.put(Phone.TYPE, Phone.TYPE_MOBILE)
271         values.put(Phone.NUMBER, phoneNumber)
272         contentResolver.insert(Data.CONTENT_URI, values)
273     }
274 }
275
276 private fun update(name:String,phone:String){
277 //     var uri =
Uri.parse("content://com.android.contacts/raw_contacts")
278     var uri =
Uri.parse("content://com.android.contacts/data/phones/filter/$phone_")
279     var cursor: Cursor? = contentResolver.query(uri,
arrayOf(Data.CONTACT_ID), "display_name=?", arrayOf(name_), null)
280
281     uri = Uri.parse("content://com.android.contacts/data")
282     if (cursor != null) {
283         while(cursor.moveToNext()){
284
285             val id__ = cursor?.getInt(0).toString()
286             val values = ContentValues()
287             values.put("data1", phone)
288             contentResolver.update(uri, values, "mimetype=? and
raw_contact_id=?", arrayOf("vnd.android.cursor.item/phone_v2",id__))
289             values.clear()
290             values.put("data1",name)
291             contentResolver.update(uri, values, "mimetype=? and
raw_contact_id=?", arrayOf("vnd.android.cursor.item/name",id__))
292             values.clear()
293
294         }
295         cursor.close()
296     }
297
298 }
299
300
301
302 }

```

Kotlin Class

Mail.kt

```
1 package com.example.maillist
2
3 import android.graphics.drawable.Drawable
4
5 class Mail(val name:String, val phone:String, val id:Int)
```

MailAdapter.kt

```
1 package com.example.maillist
2
3 import android.app.AlertDialog
4 import android.content.Intent
5 import android.graphics.drawable.Drawable
6 import android.media.Image
7 import android.provider.ContactsContract
8 import android.util.Log
9 import android.view.LayoutInflater
10 import android.view.View
11 import android.view.View.OnClickListener
12 import android.view.ViewGroup
13 import android.widget.*
14 import androidx.recyclerview.widget.RecyclerView
15 import java.util.*
16 import com.example.maillist.MailAdapter.ItemListener
17
18
19
20
21
22
23
24 class MailAdapter(val mailList: MutableList<Mail>):
25     RecyclerView.Adapter<MailAdapter.ViewHolder>(), Filterable{
26
27     private var backGround: ArrayList<String> = object : ArrayList<String>
28     () {
29         init {
30             add("circle_1")
31             add("circle_2")
32             add("circle_3")
33         }
34     }
35     private var back_index = 1
36
37     //存放原数据
38     private var mSourceList = mutableListOf<Mail>()
39     //存放过滤后的数据
40     private var mFilterList = mutableListOf<Mail>()
41
```

```

42
43     interface ItemListener {
44         fun onItemClick(position: Int)
45     }
46     private var itemListener: ItemListener? = null
47     fun setOnItemClickListener(itemListener: ItemListener?) {
48         this.itemListener = itemListener
49     }
50
51
52
53     init {
54         mSourceList = mailList
55     }
56
57     inner class ViewHolder(view: View) : RecyclerView.ViewHolder(view){
58         val name: TextView = view.findViewById(R.id.name)
59         val phone: TextView = view.findViewById(R.id.phone)
60         val imageID: ImageView = view.findViewById(R.id.image)
61         var id :Int = 0
62     }
63
64     override fun onCreateViewHolder(parent: ViewGroup, viewType: Int):
ViewHolder {
65         val view =
LayoutInflater.from(parent.context).inflate(R.layout.mail_item,parent,false
)
66         val viewHolder = ViewHolder(view)
67
68         viewHolder.itemView.setOnLongClickListener(View.OnLongClickListener
{
69             val position: Int = viewHolder.getAdapterPosition()
70             itemListener?.onItemClick(position)
71             return@OnLongClickListener true
72         })
73
74         viewHolder.itemView.setOnClickListener {
75             val position = viewHolder.adapterPosition
76             val mail = mailList[position]
77             var i=0
78             var num=position+1
79             while(i<mailList.size){
80
81                 if(mailList[i].name.contains(search_text)||mailList[i].phone.contains(sear
ch_text)){
82                     num -= 1
83                     if(num==0){
84                         break
85                     }
86                     i += 1
87                 }
88                 val pos:Int = i
89                 val id=mailList[pos].id
90                 val intent=Intent(parent.context,person::class.java)
91                 intent.putExtra("name",mailList[pos].name)
92                 intent.putExtra("phone",mailList[pos].phone)
93                 intent.putExtra("id",mailList[pos].id)

```

```

94         parent.context.startActivity(intent)
95     }
96
97     return viewHolder
98
99 }
100
101
102 override fun onBindViewHolder(viewHolder: ViewHolder, position: Int) {
103     val mail = mFilterList[position]
104     viewHolder.name.text = mail.name
105     viewHolder.phone.text = mail.phone
106     viewHolder.id = mail.id
107     viewHolder.imageID.setImageResource(R.drawable.people)
108 }
109
110 override fun getItemCount() = mFilterList.size
111
112 ///////////////////////////////////////////////////
113
114 override fun getFilter(): Filter {
115     return object : Filter() {
116         //执行过滤操作
117         override fun performFiltering(charSequence: CharSequence):
FilterResults {
118             val charString = charSequence.toString()
119             //存放已过滤的数据
120             var theFilterList = if (charString.isEmpty()) {
121                 //没有过滤的内容，则使用源数据
122                 mSourceList
123             } else {
124                 mSourceList.filter { it.name.contains(charString) ||
it.phone.contains(charString) }
125             }
126             val filterResults = FilterResults()
127             filterResults.values = theFilterList
128             return filterResults
129         }
130         //把过滤后的值返回出来并进行更新
131         override fun publishResults(
132             charSequence: CharSequence,
133             filterResults: FilterResults
134         ) {
135             mFilterList = filterResults.values as MutableList<Mail>
136             notifyDataSetChanged()
137         }
138     }
139 }
140
141 }

```


SoftKeyBoardListener.kt

```
1 package com.example.maillist
2
3 import android.app.Activity
4 import android.graphics.Rect
5 import android.view.View
6 import android.view.ViewTreeObserver.OnGlobalLayoutListener
7
8
9 /**
10  * Created by 唐小糖 on 2017/9/6.
11  * 监听软键盘
12  */
13 class SoftKeyBoardListener(activity: Activity) {
14     private val rootView //activity的根视图
15         : View
16     var rootViewVisibleHeight //纪录根视图的显示高度
17         = 0
18     private var onSoftKeyboardChangeListener: OnSoftKeyboardChangeListener?
19     = null
20     private fun
21     setOnSoftKeyboardChangeListener(onSoftKeyboardChangeListener:
22     OnSoftKeyboardChangeListener) {
23         this.onSoftKeyboardChangeListener = onSoftKeyboardChangeListener
24     }
25
26     interface OnSoftKeyboardChangeListener {
27         fun keyBoardShow(height: Int)
28         fun keyBoardHide(height: Int)
29     }
30
31     companion object {
32         fun setListener(
33             activity: Activity,
34             onSoftKeyboardChangeListener: OnSoftKeyboardChangeListener
35         ) {
36             val softKeyBoardListener = SoftKeyBoardListener(activity)
37
38             softkeyBoardListener.setOnSoftKeyboardChangeListener(onSoftKeyboardChangeLi
39             stener)
40         }
41     }
42
43     init {
44         //获取activity的根视图
45         rootView = activity.window.decorView
46
47         //监视图树中全局布局发生改变或者视图树中的某个视图的可视状态发生改变
48
49         rootView.viewTreeObserver.addOnGlobalLayoutListener(OnGlobalLayoutListener
50         { //获取当前根视图在屏幕上显示的大小
51             val r = Rect()
52             rootView.getWindowVisibleDisplayFrame(r)
53             val visibleHeight = r.height()
54             println("" + visibleHeight)
55             if (rootViewVisibleHeight == 0) {
```

```

49         rootViewVisibleHeight = visibleHeight
50         return@OnGlobalLayoutListener
51     }
52
53     //根视图显示高度没有变化，可以看作软键盘显示 / 隐藏状态没有改变
54     if (rootViewVisibleHeight == visibleHeight) {
55         return@OnGlobalLayoutListener
56     }
57
58     //根视图显示高度变小超过200，可以看作软键盘显示了
59     if (rootViewVisibleHeight - visibleHeight > 200) {
60         if (onSoftKeyboardChangeListener != null) {
61
62             onSoftKeyboardChangeListener!!.keyBoardShow(rootViewVisibleHeight -
visibleHeight)
63         }
64         rootViewVisibleHeight = visibleHeight
65         return@OnGlobalLayoutListener
66     }
67
68     //根视图显示高度变大超过200，可以看作软键盘隐藏了
69     if (visibleHeight - rootViewVisibleHeight > 200) {
70         if (onSoftKeyboardChangeListener != null) {
71
72             onSoftKeyboardChangeListener!!.keyBoardHide(visibleHeight -
rootViewVisibleHeight)
73         }
74         rootViewVisibleHeight = visibleHeight
75         return@OnGlobalLayoutListener
76     }
77 }

```

Layout

activity_main.xml

```

1  <?xml version="1.0" encoding="utf-8"?>
2  <RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
3      xmlns:app="http://schemas.android.com/apk/res-auto"
4      xmlns:tools="http://schemas.android.com/tools"
5      android:layout_width="match_parent"
6      android:layout_height="match_parent"
7      android:background="#ffffff"
8      tools:context=".MainActivity">
9
10     <View
11         android:id="@+id/topBar"
12         android:layout_width="match_parent"
13         android:layout_height="0dp" />
14
15
16     <LinearLayout
17         android:id="@+id/title_top"
18         android:layout_width="match_parent"

```

```
19     android:layout_height="wrap_content"
20     android:layout_below="@+id/topBar"
21     android:background="@color/white"
22     android:orientation="vertical"
23     android:layout_alignParentTop="true" >
24
25     <TextView
26         android:id="@+id/title_text"
27         android:layout_width="match_parent"
28         android:layout_height="wrap_content"
29         android:layout_marginTop="10dp"
30         android:background="@color/white"
31         android:layout_marginLeft="20dp"
32         android:layout_gravity="center_vertical"
33         android:text="联系人"
34         android:gravity="center_vertical"
35         android:textColor="@color/black"
36         android:textSize="25sp"
37     />
38
39     <TextView
40         android:id="@+id/number_of_people"
41         android:layout_width="match_parent"
42         android:layout_height="wrap_content"
43         android:text="61个联系人"
44         android:layout_marginLeft="20dp"
45         android:layout_marginBottom="5dp"
46     />
47
48 </LinearLayout>
49
50
51 <LinearLayout
52     android:id="@+id/title_search"
53     android:layout_width="match_parent"
54     android:layout_height="wrap_content"
55     android:layout_marginRight="20dp"
56     android:layout_marginLeft="20dp"
57     android:layout_marginTop="10dp"
58     android:orientation="horizontal"
59     android:background="@color/white"
60     android:layout_below="@id/title_top" >
61
62     <RelativeLayout
63         android:id="@+id/return_1"
64         android:layout_width="0dp"
65         android:layout_height="match_parent"
66         android:visibility="gone"
67         android:layout_weight="1">
68
69         <ImageView
70             android:id="@+id/return_2"
71             android:layout_width="match_parent"
72             android:layout_height="match_parent"
73             android:layout_marginTop="8dp"
74             android:layout_marginBottom="18dp"
75             android:layout_alignParentLeft="true"
76             android:layout_marginRight="12dp"
```

```

77         android:src="@drawable/return_" />
78
79     </RelativeLayout>
80
81
82     <LinearLayout
83         android:id="@+id/title_activity"
84         android:layout_width="0dp"
85         android:layout_weight="6"
86         android:orientation="horizontal"
87         android:background="@drawable/searchview_line"
88         android:layout_marginBottom="10dp"
89         android:layout_height="wrap_content" >
90
91         <SearchView
92             android:id="@+id/searchview"
93             android:layout_width="match_parent"
94             android:layout_height="40dp"
95             android:iconifiedByDefault="false"
96             android:queryBackground="@null"
97             android:submitBackground="@null"
98             android:queryHint="搜索联系人" />
99
100
101     </LinearLayout>
102
103 </LinearLayout>
104
105
106 <androidx.recyclerview.widget.RecyclerView
107     android:id="@+id/recyclerView"
108     android:layout_width="match_parent"
109     android:layout_height="match_parent"
110     android:layout_below="@+id/title_search"
111     android:background="@color/white" />
112
113     <com.google.android.material.floatingactionbutton.FloatingActionButton
114 xmlns:app="http://schemas.android.com/apk/res-auto"
115     android:id="@+id/fab_plus"
116     android:layout_width="wrap_content"
117     android:layout_height="wrap_content"
118     android:layout_alignParentRight="true"
119     android:layout_alignParentBottom="true"
120     android:layout_marginRight="20dp"
121     android:layout_marginBottom="60dp"
122     android:backgroundTint="#3300ff"
123     android:scaleType="center"
124     android:src="@drawable/plus"
125     android:tint="ffffff"
126     app:pressedTranslationZ="12dp"
127
128     />
129
130 </RelativeLayout>

```

person.xml

```
1 <?xml version="1.0" encoding="utf-8"?>
2 <RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
3     android:background="@color/white"
4     android:layout_width="match_parent"
5     android:layout_height="match_parent">
6
7     <LinearLayout
8         android:id="@+id/title_top_person"
9         android:orientation="vertical"
10        android:background="@color/yellow"
11        android:layout_width="match_parent"
12        android:layout_height="wrap_content"
13        android:layout_alignParentTop="true" >
14
15        <ImageView
16            android:id="@+id/return_2"
17            android:layout_width="wrap_content"
18            android:layout_height="match_parent"
19            android:layout_marginLeft="20dp"
20            android:layout_marginTop="15dp"
21            android:layout_marginBottom="18dp"
22            android:layout_gravity="left"
23            android:src="@drawable/return_" />
24
25    </LinearLayout>
26
27    <View
28        android:id="@+id/view_width"
29        android:layout_width="match_parent"
30        android:layout_height="150dp"
31        android:layout_below="@+id/title_top_person"
32        android:background="@color/yellow" />
33
34    <LinearLayout
35        android:id="@+id/edit_person_bottom"
36        android:layout_width="match_parent"
37        android:layout_height="wrap_content"
38        android:layout_alignParentBottom="true"
39        android:orientation="vertical">
40
41        <ImageView
42            android:id="@+id/image_edit"
43            android:layout_width="wrap_content"
44            android:layout_height="wrap_content"
45            android:layout_gravity="center_horizontal"
46            android:src="@drawable/editor_1" />
47
48        <Button
49            android:id="@+id/edit_person"
50            android:layout_width="wrap_content"
51            android:layout_height="20dp"
52            android:layout_gravity="center_horizontal"
53            android:background="@null"
54            android:gravity="center"
55            android:layout_marginBottom="5dp"
```

```
56         android:text="编辑"
57         android:textSize="15sp" />
58
59     </LinearLayout>
60
61     <FrameLayout
62         android:id="@+id/image_person_parent"
63         android:layout_width="110dp"
64         android:layout_height="110dp"
65         android:layout_below="@+id/view_width"
66         android:layout_centerHorizontal="true"
67         android:layout_marginTop="-55dp">
68
69         <ImageView
70             android:id="@+id/image_person"
71             android:layout_width="match_parent"
72             android:layout_height="match_parent"
73             android:background="@drawable/circle_4"
74             android:src="@drawable/people" />
75     </FrameLayout>
76
77     <TextView
78         android:id="@+id/name"
79         android:layout_width="match_parent"
80         android:layout_height="wrap_content"
81         android:layout_below="@id/image_person_parent"
82         android:layout_marginTop="10dp"
83         android:text="马 部"
84         android:gravity="center"
85         android:textColor="@color/black"
86         android:textSize="25sp" />
87
88     <TextView
89         android:id="@+id/phone"
90         android:layout_width="match_parent"
91         android:layout_height="wrap_content"
92         android:layout_below="@+id/name"
93         android:text="178-7578-4526"
94         android:gravity="center"
95         android:textSize="40sp"
96         android:layout_marginTop="10dp" />
97
98     <LinearLayout
99         android:id="@+id/phoneCall_parent"
100        android:layout_width="match_parent"
101        android:orientation="horizontal"
102        android:layout_below="@+id/phone"
103        android:layout_marginTop="40dp"
104        android:layout_marginLeft="20dp"
105        android:layout_marginRight="20dp"
106        android:layout_height="wrap_content">
107
108        <ImageView
109            android:id="@+id/phoneCall"
110            android:layout_width="0dp"
111            android:layout_weight="1"
112            android:layout_height="wrap_content"
113            android:src="@drawable/phone" />
```

```

114
115         <ImageView
116             android:id="@+id/phoneSend"
117             android:layout_width="0dp"
118             android:layout_weight="1"
119             android:layout_height="match_parent"
120             android:src="@drawable/message" />
121
122
123     </LinearLayout>
124
125 </RelativeLayout>

```

edit_information.xml

```

1  <?xml version="1.0" encoding="utf-8"?>
2  <RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
3      android:layout_width="match_parent"
4      android:layout_height="match_parent">
5
6      <LinearLayout
7          android:id="@+id/title_top_person"
8          android:layout_width="match_parent"
9          android:layout_height="wrap_content"
10         android:layout_alignParentTop="true"
11         android:orientation="horizontal">
12
13         <ImageView
14             android:id="@+id/return_3"
15             android:layout_width="60dp"
16             android:layout_height="match_parent"
17             android:layout_gravity="center_vertical"
18             android:layout_marginLeft="10dp"
19             android:layout_marginTop="15dp"
20             android:layout_marginBottom="15dp"
21             android:src="@drawable/wrong" />
22
23         <TextView
24             android:id="@+id/text_title"
25             android:layout_width="0dp"
26             android:layout_height="match_parent"
27             android:layout_weight="4"
28             android:gravity="center_vertical"
29             android:text="编辑联系人"
30             android:textColor="@color/black"
31             android:textSize="22sp" />
32
33         <ImageView
34             android:id="@+id/save"
35             android:visibility="gone"
36             android:layout_width="0dp"
37             android:layout_height="match_parent"
38             android:layout_gravity="center_vertical"
39             android:layout_marginTop="15dp"
40             android:layout_marginRight="10dp"

```

```

41         android:layout_marginBottom="15dp"
42         android:layout_weight="1"
43         android:src="@drawable/right" />
44
45     </LinearLayout>
46
47
48     <ImageView
49         android:id="@+id/image_information"
50         android:layout_width="60dp"
51         android:layout_height="60dp"
52         android:background="@drawable/circle_3"
53         android:src="@drawable/people"
54         android:layout_below="@+id/title_top_person"
55         android:layout_centerHorizontal="true"
56         android:layout_marginTop="100dp" />
57
58
59     <LinearLayout
60         android:id="@+id/name_parent"
61         android:layout_width="match_parent"
62         android:layout_height="wrap_content"
63         android:orientation="horizontal"
64         android:layout_marginLeft="15dp"
65         android:layout_marginRight="15dp"
66         android:layout_marginTop="50dp"
67         android:background="@color/white"
68         android:layout_below="@+id/image_information">
69
70         <ImageView
71             android:layout_width="30dp"
72             android:layout_height="30dp"
73             android:layout_marginTop="10dp"
74             android:layout_marginBottom="10dp"
75             android:layout_marginLeft="10dp"
76             android:src="@drawable/people_1"
77             />
78
79         <EditText
80             android:id="@+id/name_information"
81             android:layout_width="0dp"
82             android:layout_weight="1"
83             android:layout_height="wrap_content"
84             android:layout_marginTop="15dp"
85             android:layout_marginBottom="10dp"
86             android:gravity="center_vertical"
87             android:textSize="15sp"
88             android:background="@null"
89             android:hint="姓名"
90             android:text=""
91             android:layout_marginLeft="15dp" />
92
93
94     </LinearLayout>
95
96
97
98     <LinearLayout

```



```

99         android:id="@+id/phone_parent"
100         android:layout_width="match_parent"
101         android:layout_height="wrap_content"
102         android:orientation="horizontal"
103         android:layout_marginLeft="15dp"
104         android:layout_marginRight="15dp"
105         android:layout_marginTop="20dp"
106         android:background="@color/white"
107         android:layout_below="@+id/name_parent">
108
109         <ImageView
110             android:layout_width="30dp"
111             android:layout_height="30dp"
112             android:layout_marginTop="10dp"
113             android:layout_marginBottom="10dp"
114             android:layout_marginLeft="10dp"
115             android:src="@drawable/phone_1"
116             />
117
118         <EditText
119             android:id="@+id/phone_information"
120             android:layout_width="0dp"
121             android:layout_weight="1"
122             android:layout_height="wrap_content"
123             android:layout_marginTop="15dp"
124             android:layout_marginBottom="10dp"
125             android:gravity="center_vertical"
126             android:textSize="15sp"
127             android:text=""
128             android:background="@null"
129             android:hint="手机号码"
130             android:layout_marginLeft="15dp" />
131
132     </LinearLayout>
133
134
135
136
137 </RelativeLayout>
138

```

mail_item.xml

```

1  <?xml version="1.0" encoding="utf-8"?>
2  <LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
3      android:layout_width="match_parent"
4      android:layout_height="wrap_content"
5      android:orientation="vertical"
6      android:background="@color/white"
7      >
8
9      <RelativeLayout
10         android:layout_width="wrap_content"
11         android:background="@color/white"
12         android:layout_marginTop="10dp"

```

```

13     android:layout_marginLeft="20dp"
14     android:layout_marginRight="20dp"
15     android:layout_marginBottom="10dp"
16     android:layout_height="wrap_content">
17
18     <RelativeLayout
19         android:id="@+id/image_parent"
20         android:layout_width="60dp"
21         android:layout_height="60dp"
22         android:layout_marginTop="5dp"
23         android:background="@drawable/circle_3"
24         android:layout_alignParentLeft="true">
25
26         <ImageView
27             android:id="@+id/image"
28             android:layout_width="match_parent"
29             android:layout_height="match_parent"
30             android:src="@drawable/people" />
31     </RelativeLayout>
32
33
34     <TextView
35         android:id="@+id/name"
36         android:layout_width="match_parent"
37         android:layout_height="wrap_content"
38         android:layout_marginTop="3dp"
39         android:layout_marginLeft="10dp"
40         android:text="广娣"
41         android:textSize="22sp"
42         android:textColor="@color/black"
43         android:layout_toRightOf="@+id/image_parent" />
44
45     <TextView
46         android:id="@+id/phone"
47         android:layout_width="match_parent"
48         android:layout_height="wrap_content"
49         android:text="17875784526"
50         android:layout_marginTop="8dp"
51         android:layout_marginLeft="10dp"
52         android:layout_toRightOf="@+id/image_parent"
53         android:layout_below="@+id/name" />
54
55     <View
56         android:layout_width="match_parent"
57         android:layout_height="1dp"
58         android:layout_marginTop="10dp"
59         android:layout_marginLeft="10dp"
60         android:layout_toRightOf="@+id/image_parent"
61         android:background="@color/black"
62         android:layout_below="@+id/phone" />
63
64 </RelativeLayout>
65
66 </LinearLayout>
67

```

Drawable

searchview_line.xml

```
1  <?xml version="1.0" encoding="utf-8"?>
2  <shape xmlns:android="http://schemas.android.com/apk/res/android"
3      android:shape="rectangle">
4
5      <!-- 表示shape的四个角的角度。只适用于矩形shape，这里的角度是指圆角的程度 -->
6      <corners android:radius="100dp" />
7
8      <!-- 这个标签表示纯色填充，通过android:color即可指定shape中填充的颜色 -->
9      <solid android:color="@color/back" />
10
11     <!-- Shape的描边，下面指定了描边的宽度和描边的颜色 -->
12     <stroke
13         android:width="1dp"
14         android:color="#d9d9d9" />
15
16 </shape>
```

circle_3.xml

```
1  <?xml version="1.0" encoding="utf-8"?>
2  <shape xmlns:android="http://schemas.android.com/apk/res/android"
3      android:shape="oval"
4      android:useLevel="false" >
5      <solid android:color="@color/yellow" />
6      <stroke android:color="@color/white" />
7      <size android:width="20dp"
8          android:height="20dp" />
9  </shape>
```