



课程名称：移动智能应用开发

姓 名： 侯佳耀

学 号： 20182131055

班 级： 计算机科学与技术 4 班

设计时间：2020 年 12 月 20 日

1.软件简介：

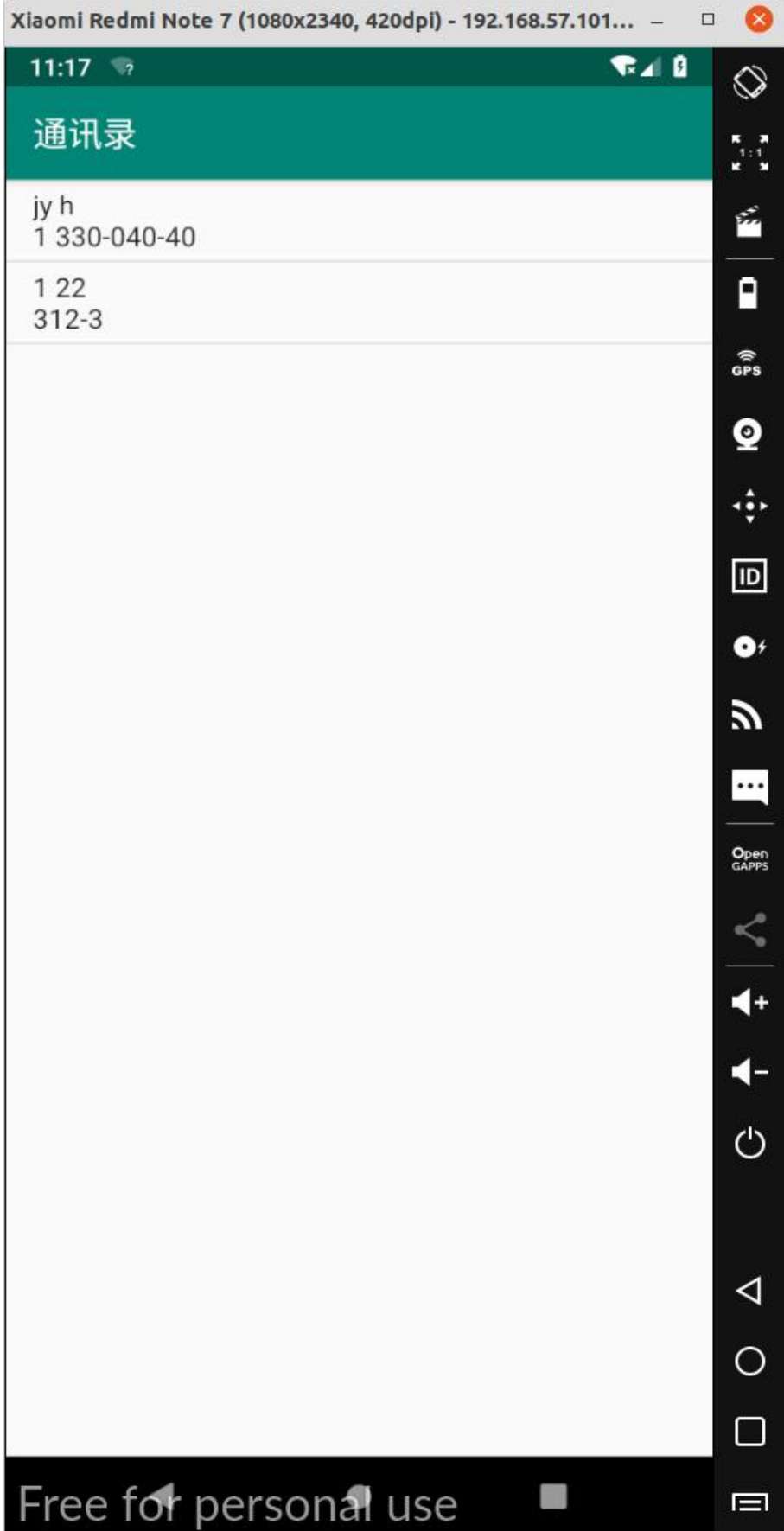
(1) 完成人：侯佳耀 20182131055

(2) 完成时间 2020 年 12 月 20 日

(3) 软件内容简介：该 app 可以调用用户通讯录，读取其内容，拥有拨打电话和发短信的功能。

2. 界面设计

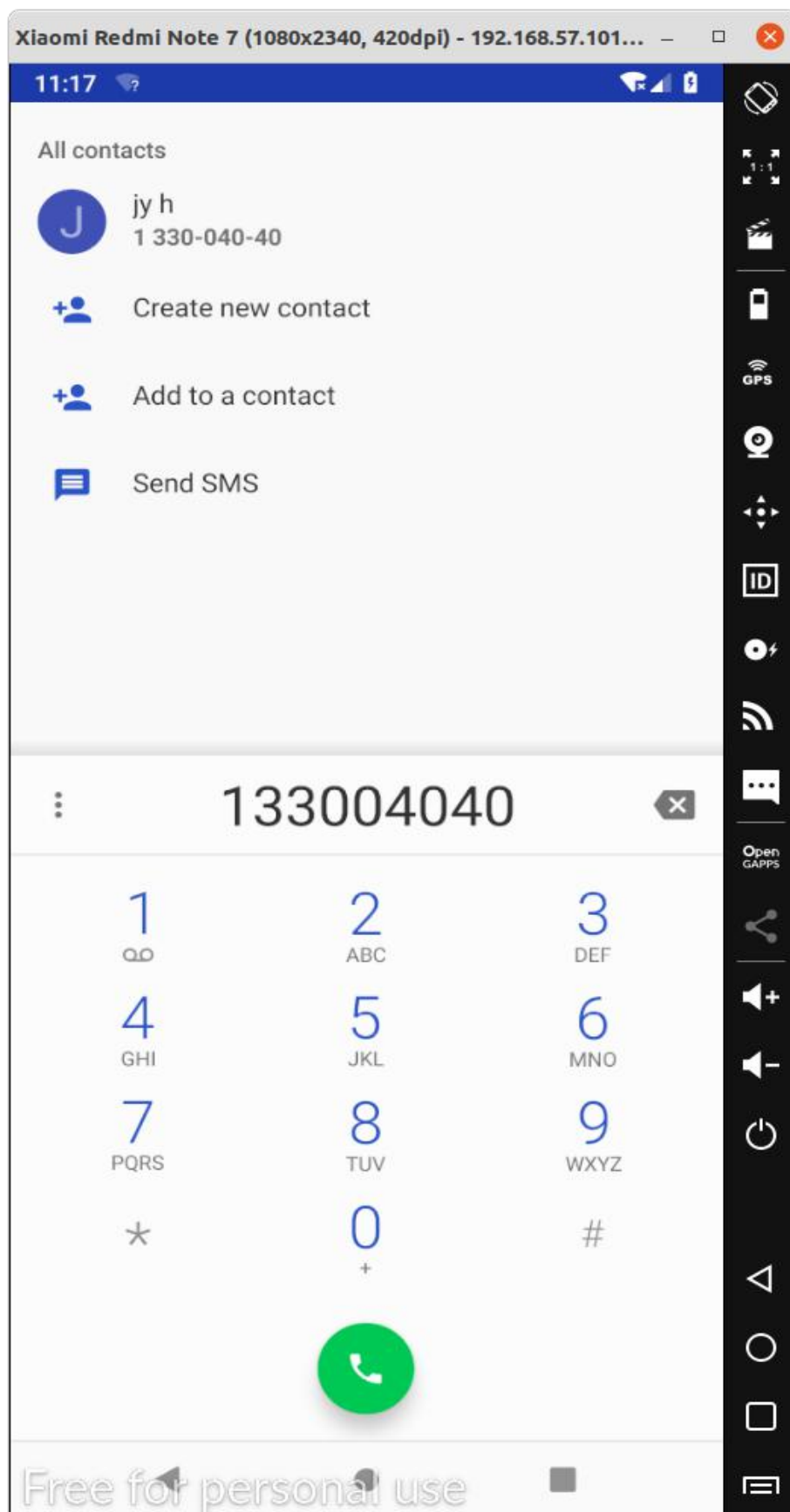
主界面



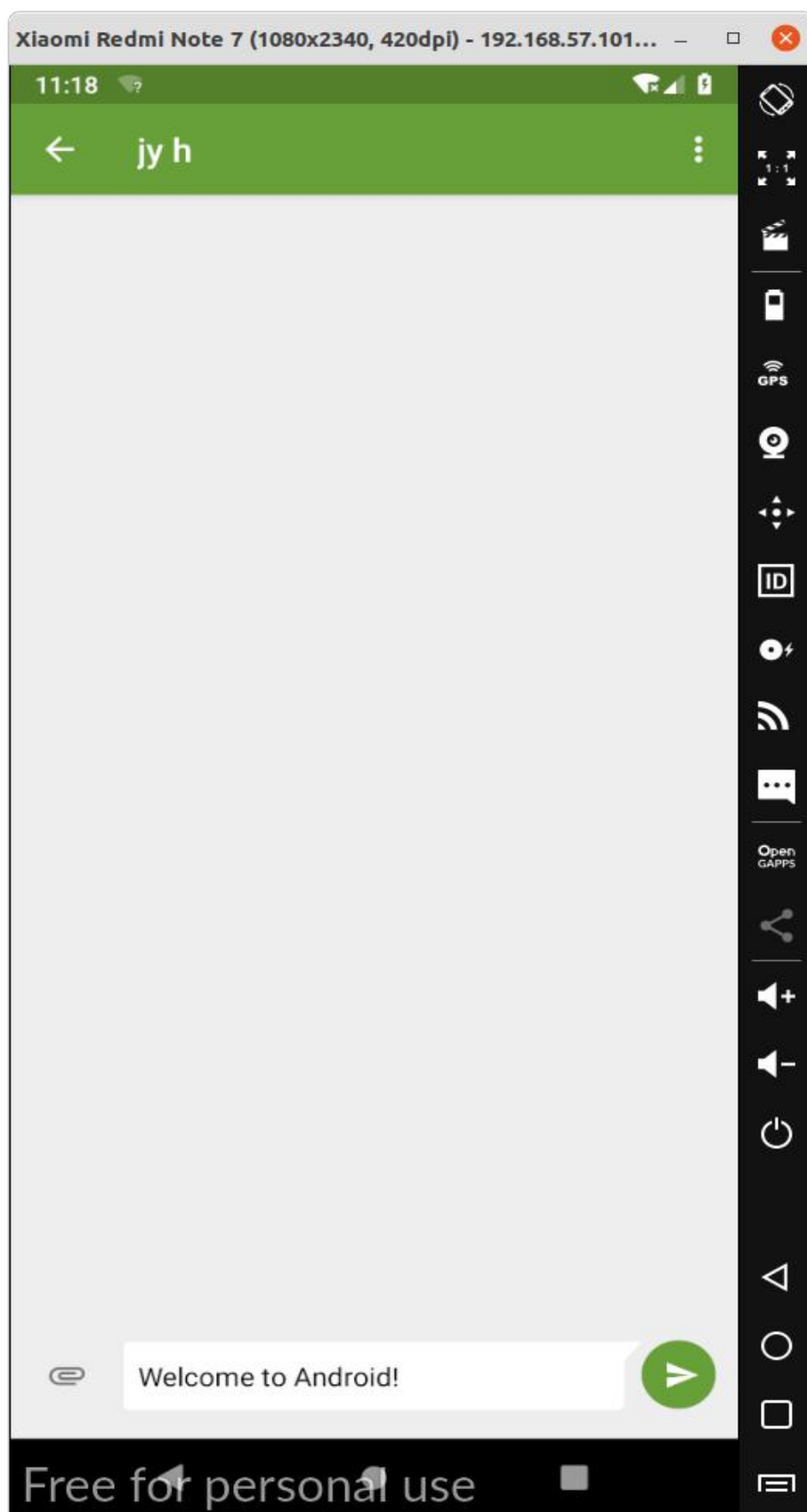
副界面



打电话



发短信



3. 代码设计

activity_main.xml

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:orientation="vertical"
    android:layout_width="match_parent"
    android:layout_height="match_parent" >

    <ListView
        android:id="@+id/contactsView"
        android:layout_width="match_parent"
        android:layout_height="match_parent"></ListView>

</LinearLayout>
```

second.xml

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:orientation="vertical" android:layout_width="match_parent"
    android:layout_height="match_parent">

    <TextView
        android:id="@+id/textView1"
        android:layout_width="match_parent"
        android:layout_height="229dp"
        android:text=""
        android:gravity="center"
    />

    <Button
        android:id="@+id/button1"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="打电话" />

    <Button
        android:id="@+id/button2"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="发短信" />

</LinearLayout>
```

MainActivity.kt

```
package com.example.通讯录

import android.Manifest
import android.content.Intent
import android.content.pm.PackageManager
import androidx.appcompat.app.AppCompatActivity
import android.os.Bundle
import android.provider.ContactsContract
import android.view.View
import android.widget.AdapterView
import android.widget.Toast
import androidx.core.app.ActivityCompat
import androidx.core.content.ContextCompat
import kotlinx.android.synthetic.main.activity_main.*

class MainActivity : AppCompatActivity(), AdapterView.OnItemClickListener {
    var telnum=""
    var name=""

    override fun onClick(v: View?) {

    }

    private val contactsList = ArrayList<String>()
    private lateinit var adapter: ArrayAdapter<String>

    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_main)

        adapter = ArrayAdapter<String>(this, android.R.layout.simple_list_item_1,
contactsList)
        contactsView.adapter = adapter

        contactsView.setOnItemClickListener{
            parent,view,position,id ->

            val position1=contactsList[position]
            Toast.makeText(this,"你点击了\n"+position1,Toast.LENGTH_SHORT).show()
            telnum = position1.substringAfter("\n")
            name = position1.substringBefore("\n")
            val intent = Intent(this, second::class.java)
            intent.putExtra("telnum",telnum)
        }
    }
}
```



```

        intent.putExtra("name", name)
        startActivity(intent)    //startActivity 方法
    }

    if (ContextCompat.checkSelfPermission(
        this,
        Manifest.permission.READ_CONTACTS
    ) != PackageManager.PERMISSION_GRANTED
    ) {
        ActivityCompat.requestPermissions(this,
arrayOf(Manifest.permission.READ_CONTACTS), 1)
    } else {
        readContacts()
    }
}

override fun onRequestPermissionsResult(requestCode: Int, permissions:
Array<String>, grantResults: IntArray) {
    super.onRequestPermissionsResult(requestCode, permissions, grantResults)
    when (requestCode) {
        1 -> {
            if (grantResults.isNotEmpty() && grantResults[0] ==
PackageManager.PERMISSION_GRANTED) {
                readContacts()
            } else {
                Toast.makeText(this, "You denied the permission",
Toast.LENGTH_SHORT).show()
            }
        }
    }
}

private fun readContacts() {
    // 查询联系人数据
    contentResolver.query(ContactsContract.CommonDataKinds.Phone.CONTENT_URI,
null, null, null, null)?.apply {
        while (moveToNext()) {
            // 获取联系人姓名
            val displayName =
getString(getColumnIndex(ContactsContract.CommonDataKinds.Phone.DISPLAY_NAME)
)

            // 获取联系人手机号
            val number =
getString(getColumnIndex(ContactsContract.CommonDataKinds.Phone.NUMBER))

```

```

        contactsList.add("$displayName\n$number")
    }
    adapter.notifyDataSetChanged()
    close()
}
}
}
}

```

second.kt

```

package com.example.通讯录

import android.content.Intent
import android.net.Uri
import androidx.appcompat.app.AppCompatActivity
import android.os.Bundle
import kotlinx.android.synthetic.main.second.*

class second : AppCompatActivity() {

    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.second)
        var telnum=intent.getStringExtra("telnum")
        var name=intent.getStringExtra("name")
        textView1.text=name+"\n"+telnum
        textView1.setTextSize(1, 50.0F)
        button1.setOnClickListener {
            var intent = Intent()
            intent.action = Intent.ACTION_DIAL//dial 是拨号的意思
            intent.data = Uri.parse("tel:" + telnum)
            startActivity(intent)
        }

        button2.setOnClickListener {
            var intent = Intent()
            intent.action = Intent.ACTION_SENDTO//发短信的 action
            intent.data = Uri.parse("smsto:" + name)//smsto:后面的是收信人，可以随
便改
            intent.putExtra("sms_body", "Welcome to Android!")//这里的第二个参数是
短信内容
            startActivity(intent)
        }
    }
}
}

```

4. 软件操作流程

(1) 在 android studio 打开 Gentmotion 模拟器并启动 build

(2) 在主界面可以看到相应的通讯录信息，可以进行拨打电话或者发短信的操作

5. 难点（或遇到的问题）和解决方案

(1) id 问题，容易出现 id 调用错误的情况，导致无法初始化实例，出现空指针和 runtime Exception 的情况、

(2) 在界面调用时过多的 findViewById 导致效率不高,于是采用 `import kotlinx.android.synthetic.main.activity_main.*` 方法进行设计，可以直接使用 id，无需声明 findViewById

6. 不足之处和今后的设想

(1) 界面设计能力过差，未找到心仪元素加入，后期改

(2) 设计 app 时一定要考虑周全，要有完整的 app 计划流程，包括逻辑部分和界面 XML 部分