

《打电话与网页浏览 app》说明文档

软件名称：CallandBrowserKSJ

完成人：姜淑晶

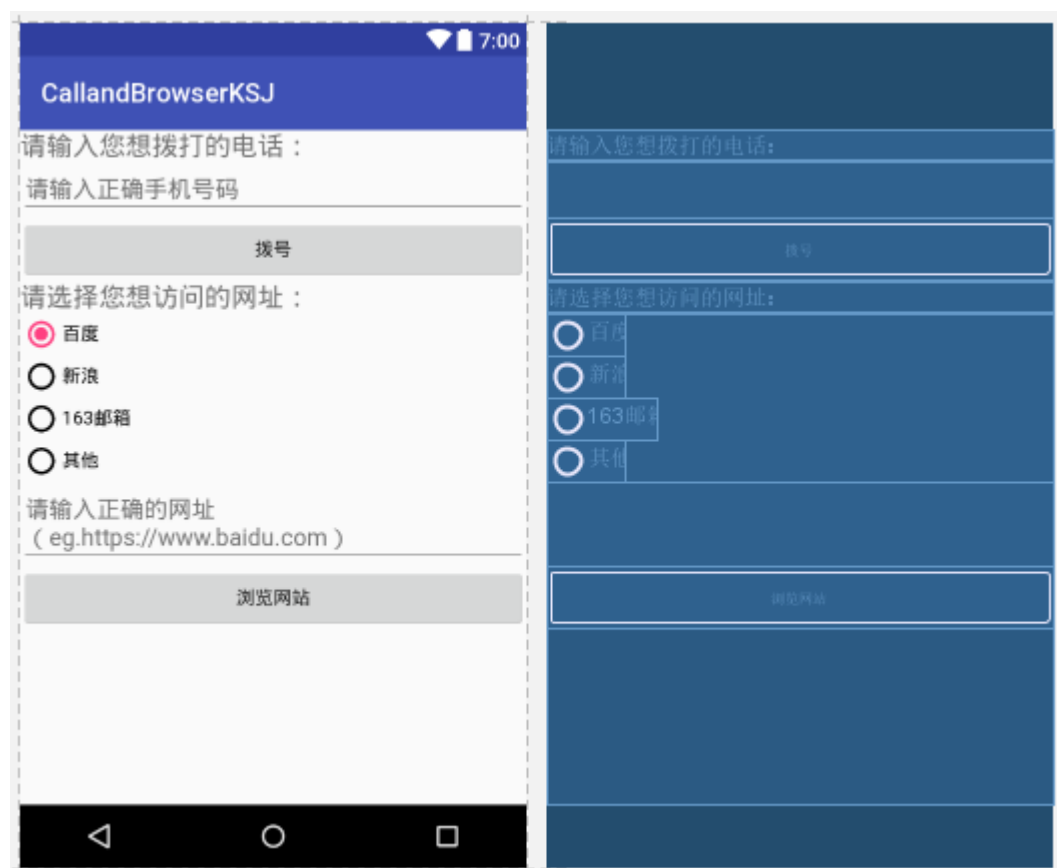
学号：20152100061

完成时间：2017-11-13

软件内容简介：

实现拨打电话与浏览网页功能的 app

界面设计：



真机运行结果:

中國移動
中國聯通

0K/s 2G 4G 55 23:47

CallandBrowserKSJ

請輸入您想撥打的電話:

請輸入正確手機號碼

撥號

請選擇您想訪問的網址:

☒ 百度

☐ 新浪

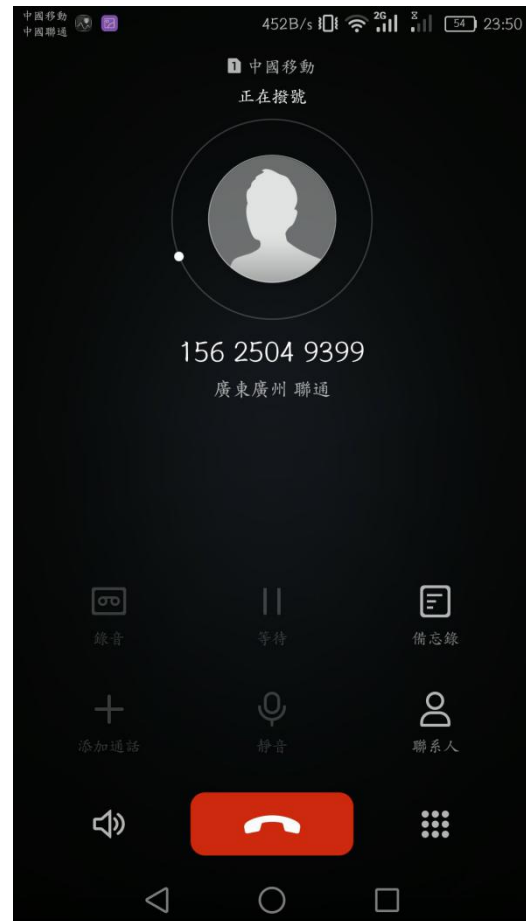
☐ 163郵箱

☐ 其他

請輸入正確的網址
(eg.https://www.baidu.com)

瀏覽網站

初始界面



号码输入正确时，成功拨出电话



号码输入错误时，显示错误提示



选择软件原始提供的网站进行访问，程序调用手机的浏览器打开原始设定好的网站进行浏览

CallandBrowserKSJ

請輸入您想撥打的電話：

15625049

撥號

請選擇您想訪問的網址：

☐ 百度

☐ 新浪

☐ 163郵箱

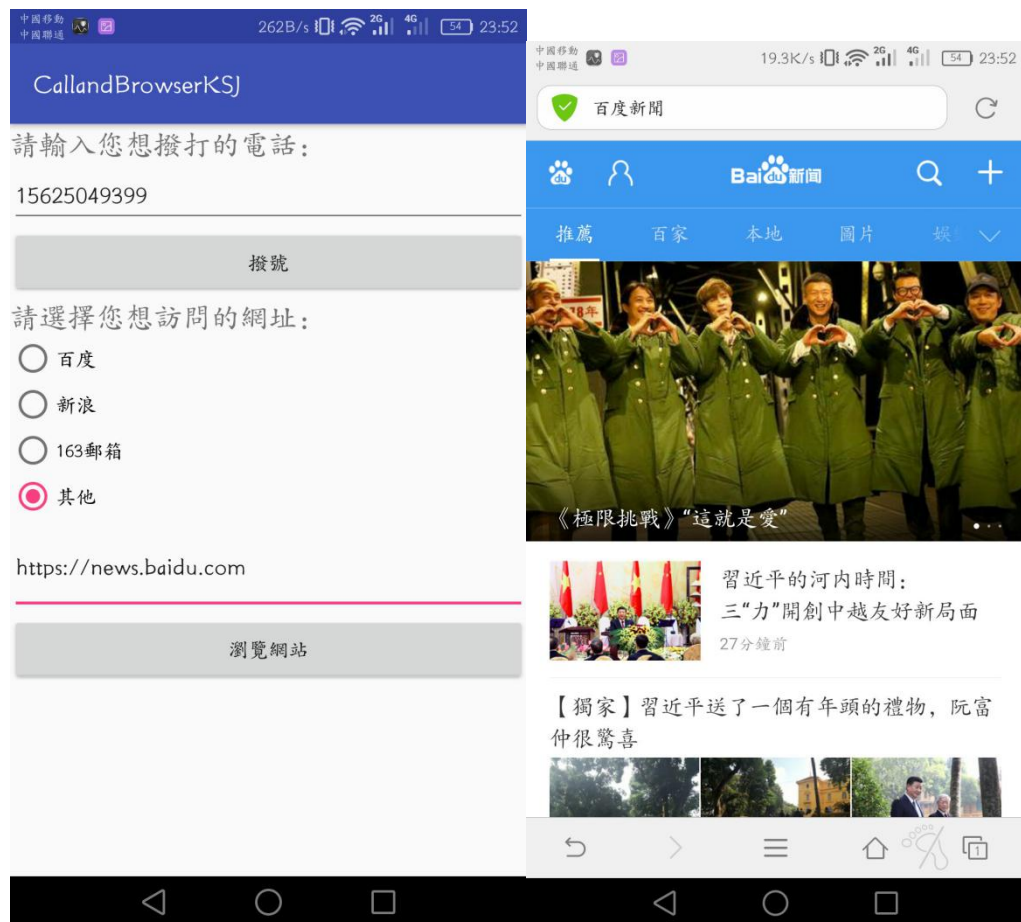
☒ 其他

請輸入正確的網址
(eg.https://www.baidu.com)

瀏覽網站

網址不可為空

当用户选择‘其他’选项时，将读取用户输入的网址进行访问，若用户没有输入网址，则显示当前访问网址为空的错误报告。



当用户输入正确的网址时，软件调用浏览器进行网站访问



当用户输入的网址不合法时，将显示错误提示。

代码设计：

详细代码说明：

页面布局 xml 文件代码：

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

    <TextView
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:textSize="20dp"
        android:text="请输入您想拨打的电话：" />

    <!-- 将手机号码的输入框设置为数字输入类型 -->
    <EditText
        android:id="@+id/tel"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:hint="请输入正确手机号码"
```

```

        android:inputType="number"/>

<Button
    android:id="@+id/call"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_gravity="center"
    android:text="拨号"/>
<TextView
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:textSize="20dp"
    android:text="请选择您想访问的网址: "/>
<RadioGroup
    android:id="@+id/webGroup"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:orientation="vertical">
    <RadioButton
        android:id="@+id/web_bd"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="百度"
        android:checked="true"/>
    <RadioButton
        android:id="@+id/web_sina"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="新浪"/>
    <RadioButton
        android:id="@+id/web_163"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="163 邮箱"/>
    <RadioButton
        android:id="@+id/web_else"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="其他"/>

</RadioGroup>

<EditText
    android:id="@+id/web"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:hint="请输入正确的网址\n (eg.https://www.baidu.com) "/>

<Button
    android:id="@+id/browse"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_gravity="center"
    android:text="浏览网站"/>

</LinearLayout>

```


软件功能实现 java 文件:

```
package com.kangsekjung.callandbrowser;

import android.Manifest;
import android.content.Intent;
import android.content.pm.PackageManager;
import android.net.Uri;
import android.support.annotation.IdRes;
import android.support.v4.app.ActivityCompat;
import android.support.v4.content.ContextCompat;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.text.Editable;
import android.text.TextWatcher;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.RadioGroup;
import android.widget.Toast;

import org.apache.commons.validator.routines.UrlValidator;

import java.io.IOException;
import java.net.HttpURLConnection;
import java.net.MalformedURLException;
import java.net.URL;

public class MainActivity extends AppCompatActivity implements View.OnClickListener {

    EditText tel;
    EditText web;
    RadioGroup wrg;
    Button call;
    Button browse;
    String website;
    boolean elseflag;

    /**
     * 控件变量初始化
     */
    private void Init() {
        tel = (EditText) findViewById(R.id.tel);
        web = (EditText) findViewById(R.id.web);
        wrg = (RadioGroup) findViewById(R.id.webGroup);
        call = (Button) findViewById(R.id.call);
        browse = (Button) findViewById(R.id.browse);
        website = "https://www.baidu.com/";//根据默认的单选按钮选项设置初始的网站地址
        elseflag = false;
        web.addTextChangedListener(watcher);
    }

    /**
     * 为控件添加相应的监听器
     */
    private void setListener() {
        call.setOnClickListener(this);
        browse.setOnClickListener(this);
        wrg.setOnCheckedChangeListener(new RadioGroup.OnCheckedChangeListener() {
            @Override
```

```

        public void onCheckedChanged(RadioGroup group, @IdRes int checkedId) {
            if (checkedId == R.id.web_bd) {
                elseflag = false;
                website = "https://www.baidu.com/";
            } else if (checkedId == R.id.web_sina) {
                elseflag = false;
                website = "http://www.sina.com.cn/";
            } else if (checkedId == R.id.web_163) {
                elseflag = false;
                website = "http://mail.163.com/";
            } else {
                //选择‘其他’选项时，获取用户输入的网址作为目的地址
                website = web.getText().toString();
                elseflag = true;
            }
        }
    });
}

/**
 * 构建网址编辑框的监听器
 */
private TextWatcher watcher = new TextWatcher() {

    @Override
    public void afterTextChanged(Editable s) {
        // TODO Auto-generated method stub
    }

    @Override
    public void beforeTextChanged(CharSequence s, int start, int count, int after) {
        // TODO Auto-generated method stub
    }

    @Override
    public void onTextChanged(CharSequence s, int start, int before, int count) {
        //当用户选中‘其他’选项时，将用户的输入网址作为目的地址
        if (elseflag) {
            website = web.getText().toString();
        }
    }
};

/**
 * 采用第三方库检测 url 是否有效
 *
 * @param url
 * @return
 */
protected static boolean checkURL(String url) {
    String[] schemas = {"http", "https"};
    UrlValidator urlValidator = new UrlValidator(schemas);
    return urlValidator.isValid(url);
}

@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);

```

```

        setContentView(R.layout.activity_main);
        Init();
        setListener();
    }

    @Override
    public void onClick(View v) {
        switch (v.getId()) {
            case R.id.call:
                //若没有获取打电话权限，则进行权限获取
                if (ContextCompat.checkSelfPermission(this,
Manifest.permission.CALL_PHONE) != PackageManager.PERMISSION_GRANTED) {
                    ActivityCompat.requestPermissions(this, new
String[] {Manifest.permission.CALL_PHONE}, 111);
                } else {
                    if (tel.getText().toString().length() == 11) { //当用户输入的手机号码等于
11 位时，认为该号码有效，实现打电话功能
                        Intent intent = new Intent();
                        intent.setAction(Intent.ACTION_CALL);
                        intent.setData(Uri.parse("tel:" + tel.getText()));
                        startActivity(intent);
                    } else
                        Toast.makeText(getApplicationContext(), "您所输入的手机号码位数不正
确，请重新确认手机号码！", Toast.LENGTH_LONG).show();
                }
                break;
            case R.id.browse:
                if (!website.equals("")) { //目的网址非空才进行网站跳转
                    if (checkURL(website)) {
                        Uri uri = Uri.parse(website); //定义 uri 数据
                        Intent in = new Intent(Intent.ACTION_VIEW, uri); //指定动作和数据
                        startActivity(in); //启动这个应用
                    } else
                        Toast.makeText(getApplicationContext(), "该网址无效，请重新确认输入网
址！", Toast.LENGTH_LONG).show();
                } else {
                    Toast.makeText(getApplicationContext(), "网址不可为空",
Toast.LENGTH_LONG).show();
                }
                break;
        }
    }
}

```

Gradle 文件的配置:

```

apply plugin: 'com.android.application'

android {
    compileSdkVersion 25
    buildToolsVersion "25.0.3"
    defaultConfig {
        applicationId "com.kangsekjung.callandbrowser"
        minSdkVersion 15
        targetSdkVersion 25
        versionCode 1
        versionName "1.0"
    }
}

```

```

        testInstrumentationRunner "android.support.test.runner.AndroidJUnitRunner"
    }
    buildTypes {
        release {
            minifyEnabled false
            proguardFiles getDefaultProguardFile('proguard-android.txt'),
'proguard-rules.pro'
        }
    }
}

dependencies {
    compile fileTree(include: ['*.jar'], dir: 'libs')
    androidTestCompile('com.android.support.test.espresso:espresso-core:2.2.2', {
        exclude group: 'com.android.support', module: 'support-annotations'
    })
    compile 'com.android.support:appcompat-v7:25.3.1'
    compile 'com.android.support.constraint:constraint-layout:1.0.2'
    testCompile 'junit:junit:4.12'
    //引入第三方库 commons validator 1.6 的 jar 包
    compile files('libs/commons-validator-1.6.jar')
    compile files('libs/commons-validator-1.6-javadoc.jar')
    compile files('libs/commons-validator-1.6-tests.jar')
}

```

清单文件的配置:

```

<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.kangsekjung.callandbrowser">

    <application
        android:allowBackup="true"
        android:icon="@mipmap/ic_launcher"
        android:label="@string/app_name"
        android:roundIcon="@mipmap/ic_launcher_round"
        android:supportRtl="true"
        android:theme="@style/AppTheme">
        <activity android:name=".MainActivity">
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />

                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
    </application>
    <uses-permission android:name="android.permission.CALL_PHONE"/><!-- 获取打电话的权限 -->
    <uses-permission android:name="android.permission.INTERNET"/><!-- 获取访问网络的权限 -->
</manifest>

```

软件操作流程:



难点（或遇到的问题）：

1.如何检测输入的 url 是否合法，当用户手动输入的网址不符合要求时，程序不会显示错误提示，而是直接退出程序；

解决方案：

1.使用第三方库 Apache Commons Validator 实现 url 合法性的检测：

创建 `org.apache.commons.validator.Validator` 类的新实例。如果当前的 `ValidatorResources` 是相同的，那么只要已经完成任何先前的验证，并且不会一次尝试从多个线程中使用特定的 `Validator` 实例，那么当前 `Validator` 实例可能会被安全地重用；添加执行验证所需的任何资源，如 `JavaBean` 来验证；调用 `org.apache.commons.validator.Validator` 上的 `validate` 方法；

由于 url 必须是全写，包含协议头，比如 `https://www.baidu.com`，当输入 `www.baidu.com` 时，将无法识别；网址协议头基本都是 `http` 或 `https`，因此在检测合法性时，创建的 `UrlValidator` 实例基本上包括 `"http"`, `"https"` 即可；
`(String[] schemas = {"http", "https"};UrlValidator urlValidator = new UrlValidator(schemas);)`

不足之处：

- 1.检测用户输入手机号的合法性的判断条件不准确，仅仅只是对手机位数进行一个判断，这样就限制了用户使用该软件进行对固话的拨打；
- 2.软件要求用户输入完整的 url 地址，否则将无法进行网站访问；这对用户有较高的要求，不方便使用；

今后的设想：

- 1.利用正则表达式对手机号码或固话号码进行合法性检测
- 2.为用户输入的缺少协议头的网址进行自动调整补充，如将用户输入的 `www.baidu.com` 补充为 `http://www.baidu.com`，使用户在输入残缺的 url 地址时，也可正确访问网站。

参考资料：

检测输入的 url 是否合法：<http://blog.csdn.net/ldld1717/article/details/72836278>