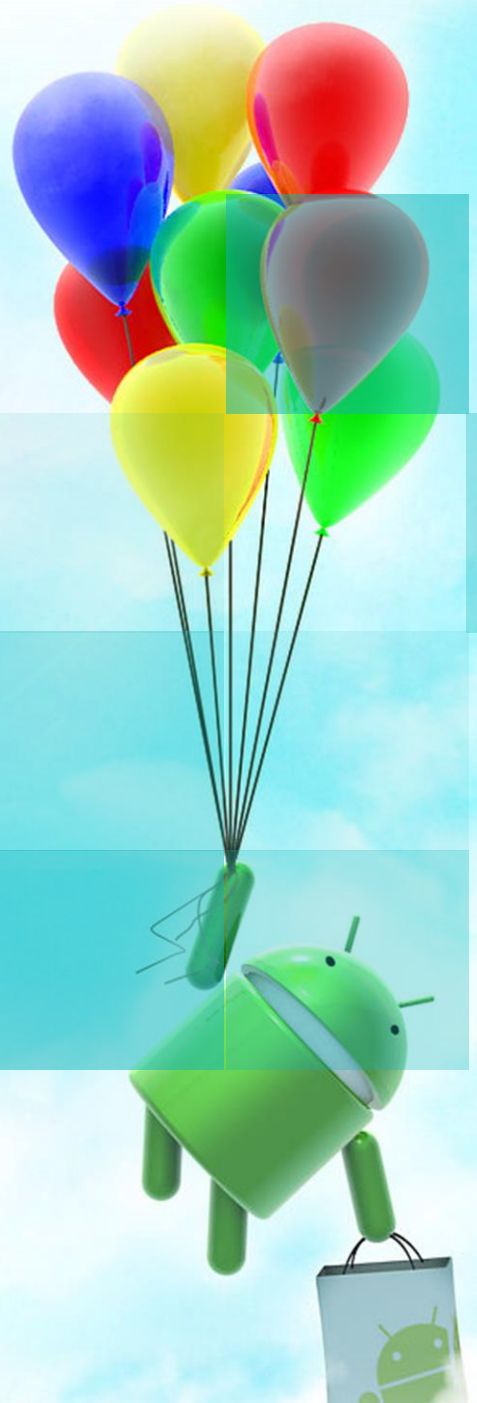
Android
云存储客户端开发



项目9

功能拓展模块

任务01 文件上传

任务02 文件下载

任务03 拍照上传

任务04 分享

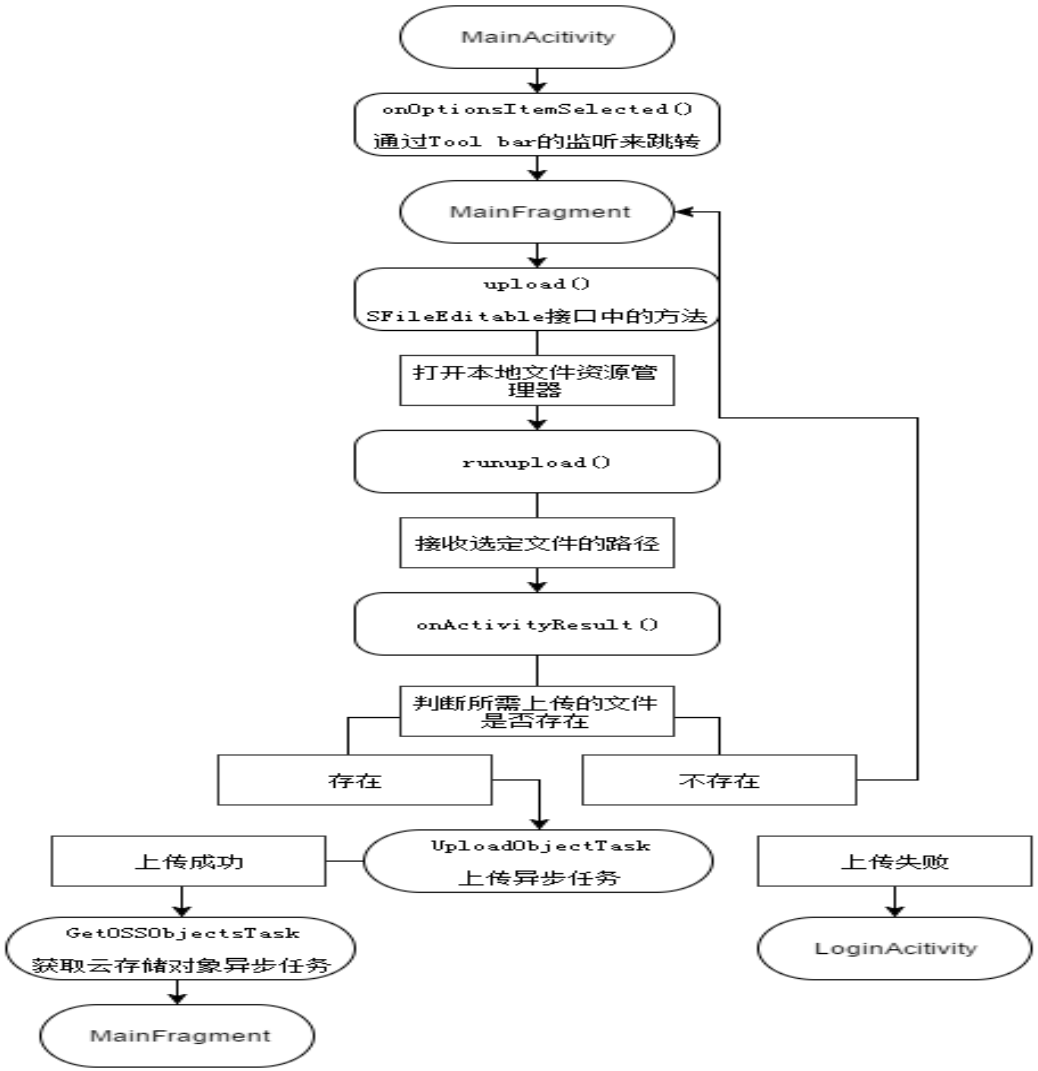
功能拓展模块

主要完成文件的上传下载、拍照上传和资源分享等功能

1

文件上传

功能流程图

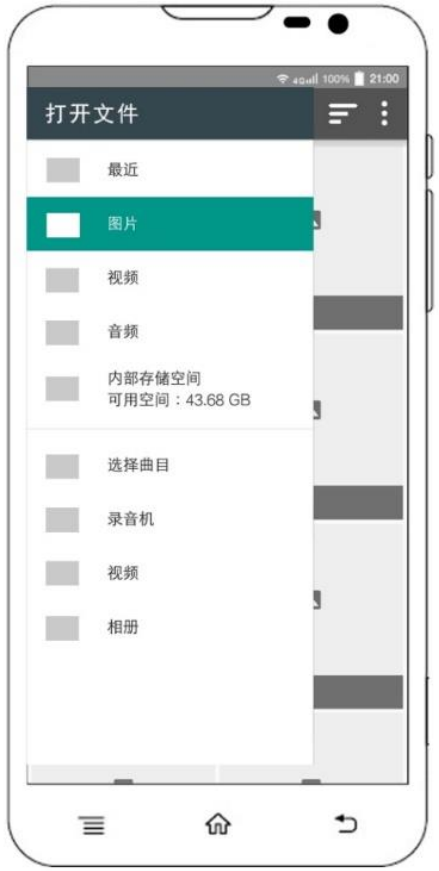

MainAcitivity
onOptionsItemSelected()
通过Tool bar的监听来跳转
MainFragment
upload()
SFileEditable接口中的方法
打开本地文件资源管理器
runupload()
接收选定文件的路径
onActivityResult()
判断所需上传的文件是否存在
存在
不存在
上传成功
UploadObjectTask
上传异步任务
上传失败
GetOSSObjectsTask
获取云存储对象异步任务
LoginAcitivity
MainFragment

1 文件上传

功能需求

1）视图层，界面实现

根据原型图设计实现文件上传窗口View。



1 文件上传

功能需求

2）控制层，用户选择上传文件

点击Toolbar中的上传标签，弹出文件资源管理器，选择所需上传的文件。

3）服务层，调用openstack提供的上传接口实现上传

这里调用openstack中的上传方法，传递的参数为上传文件路径和所要上传到的目录。

4）控制层，返回结果处理

上传可能的情况包含：

（1）所需上传的文件格式无法识别，提示用户上传失败，文件格式不对；

（2）所上传的文件已经存在，提示用户上传失败，文件已存在。

（3）上传成功，提示用户上传成功。

1

文件上传

实现步骤

选择File\Open..，点击弹出选择project93目录下面的项目“swiftstorage”

2）新建一个GraphicsUtil工具类

此工具类为图形帮助类，此类中的方法将原始路径转化为文件路径，该类中包含以下方法.

```
//获得照片的旋转方向
public static int getCameraPhotoOrientation(Context context, Uri imageUri,
        String imagePath) throws IOException
//获得照片的路径
public static int getOrientation(Context context, Uri imageUri)
        throws IOException
//获得照片的原始路径
public static String getOriginalFilePath(Context context, Uri imageUri)
```

1 文件上传

实现步骤

3）在MainActivity中对上传标签进行处理

在MainActivity中的方法中对菜单的新建目录选项进行操作。

```
if (id == R.id.action_upload) {
    this.currentFragment.upload();
}
```

调用当前Fragment中的upload（）方法。

1 文件上传

实现步骤

4）新建一个静态变量

```
private static final int ACTION_SELECT_CONTENT_FROM_UPLOAD = 2;
```

此静态变量用来传输选择的本地的文件

5）在MainFragment中完成操作

（1）实现SFileEditable接口下的upload（）的方法。

```
@Override
public void upload() {
    runupload();
}
```

1 文件上传

实现步骤

（2）通过Intent方法调用本地文件资源管理器

```
//打开文件资源管理器
private void runupload() {
    Intent intent = new Intent();
    intent.setAction(Intent.ACTION_GET_CONTENT);
    intent.setType("*/*");
    intent.setFlags(Intent.FLAG_ACTIVITY_CLEAR_TOP);
    intent.addCategory(Intent.CATEGORY_OPENABLE);
    startActivityForResult(intent, ACTION_SELECT_CONTENT_FROM_UPLOAD);
}
```

（3）通过onActivityResult方法来回调传输数据

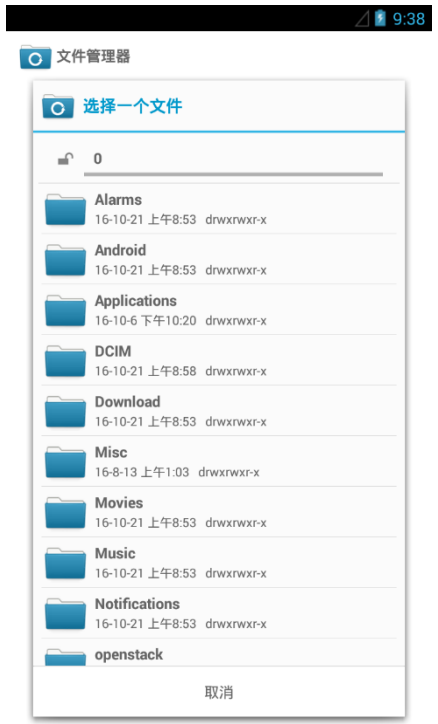
```
//处理回传数据，判断上传是否成功
public void onActivityResult(int requestCode, int resultCode, Intent data)
```

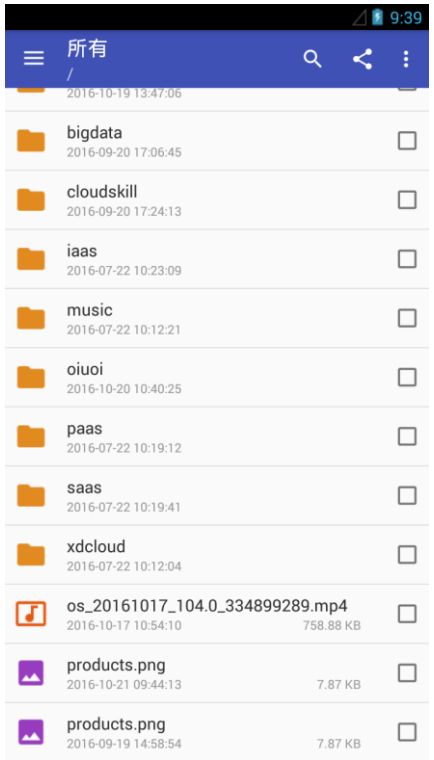
执行异步任务，在异步任务中调用SDK中的上传方法，完成后刷新列表

1 文件上传
功能执行测试

1）执行效果

选择项目Android的“app”，点击工具条中运行Run “app”，执行的效果如图6-5所示，填入已经注册好的用户名和密码，进行登录验证。

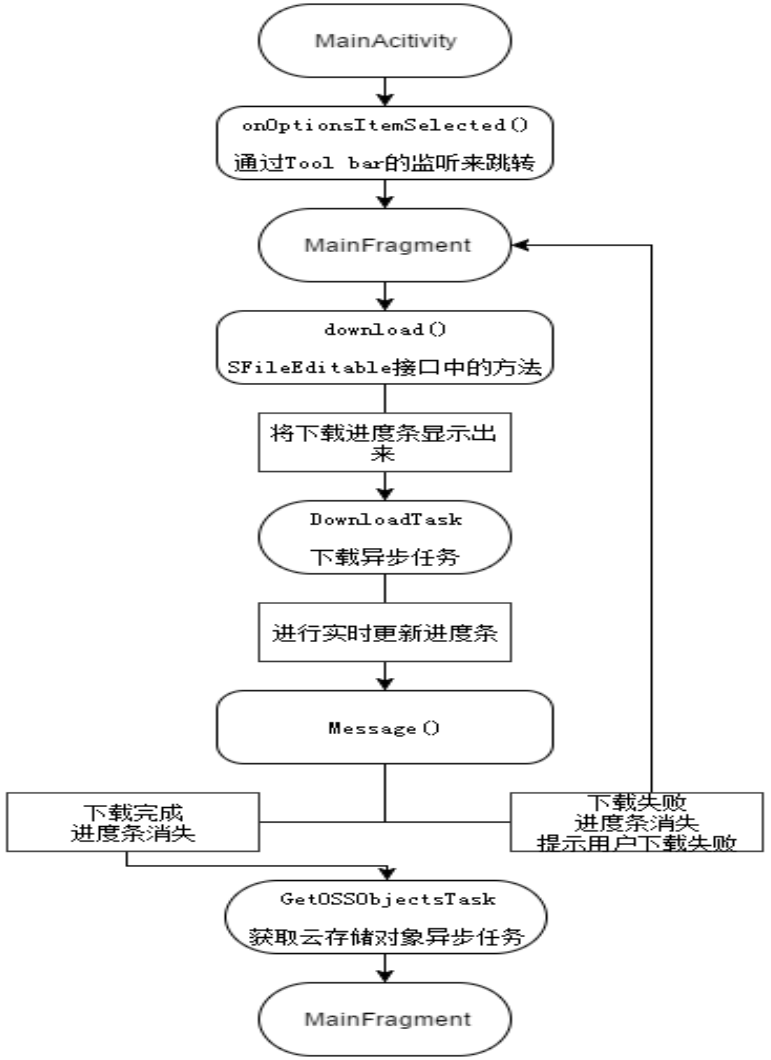




2

文件下载

功能流程图


MainAcitivity
onOptionsItemSelected()
通过Tool bar的监听来跳转
MainFragment
download()
SFileEditable接口中的方法
将下载进度条显示出来
DownloadTask
下载异步任务
进行实时更新进度条
Message()
下载完成
进度条消失
下载失败
进度条消失
提示用户下载失败
GetOSSObjectsTask
获取云存储对象异步任务
MainFragment

2 文件下载

功能需求

1）视图层，界面实现

根据原型图设计实现文件下载窗口View。

2 文件下载

功能需求

2）控制层，用户选择下载文件，点击下载标签

点击所需下载文件的复选框，点击右上角下拉菜单中的下载标签，显示文件下载进度条。

3）服务层，调用openstack提供的下载接口实现下载

这里调用openstack中的下载方法，传递的参数为容器名称和所要下载文件的路径。

4）控制层，返回结果处理

下载可能的情况包含：

（1）下载成功，进度条消失，提示用户成功；

（2）下载失败，提示错误。

2 文件下载

实现步骤

1）导入项目

选择File\Open..，点击弹出选择project91目录下面的项目“swiftstorage”。

2）在MainActivity中对上传标签进行处理

在MainActivity中的方法中对菜单的新建目录选项进行操作。

```
if (id == R.id.action_download) {
    this.currentFragment.download();
}
```

调用当前Fragment中的download（）方法

2 文件下载

实现步骤

4）新建一个Handle用来实时更新进度条

```
//进度条更新
private Handler handler=new Handler()
{
    public void handleMessage(Message msg)
    {
        if (!Thread.currentThread().isInterrupted()) {
            switch (msg.what) {
                case 0:
                    pb.setProgress(downLoadFileSize);
                    break;
                default:
                    break;
            }
        }
    }
};
```

2 文件下载

实现步骤

5）在MainFragment中完成操作

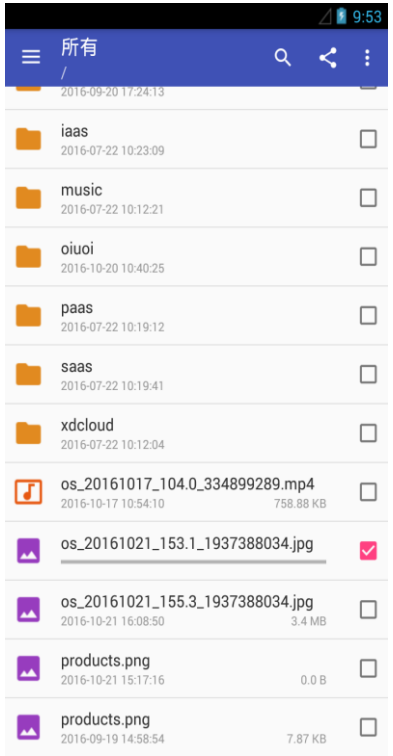
实现SFileEditable接口下的download（）的方法。

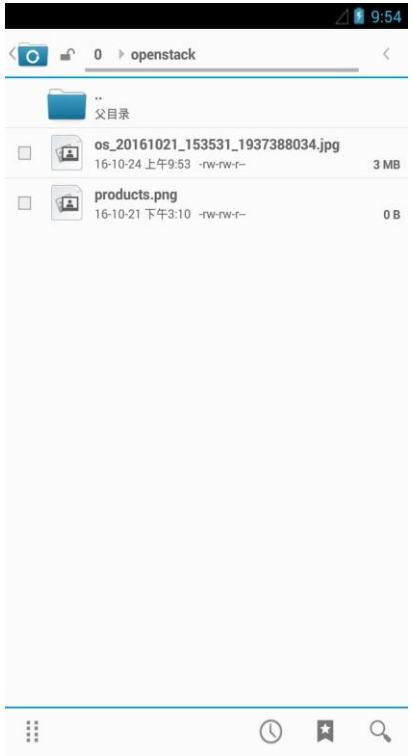
```
@Override
public void download() {
    SFile file = getFirstSelected();
 if (file != null) {
        for (downid=0;downid<fileListData.size();downid++) {
            //判断哪个文件被选中
            if (fileListData.get(downid).isChecked()) {
                //设置是否展示进度条，0不展示展示，1展示
                fileListData.get(downid).setIsDisplayProgressBar(1);
                //刷新视图
 fileListViewAdapter.notifyDataSetChanged();
                //获取所需下载文件的view
                View view=fileListView.getChildAt(downid-
fileListView.getFirstVisiblePosition());
                //定位至进度条
                pb= (ProgressBar) view.findViewById(R.id.down_pb);
                break;
            }
        }
        DownloadTask downloadTask = new DownloadTask(file);
        downloadTask.execute();
    }
}
```

执行异步任务，在异步任务中调用SDK中的下载的方法，完成后刷新列表。

2 文件下载
功能执行测试

选择项目Android的“app”，点击工具条中运行Run “app”，执行的效果如图所示，填入已经注册好的用户名和密码，进行登录验证。

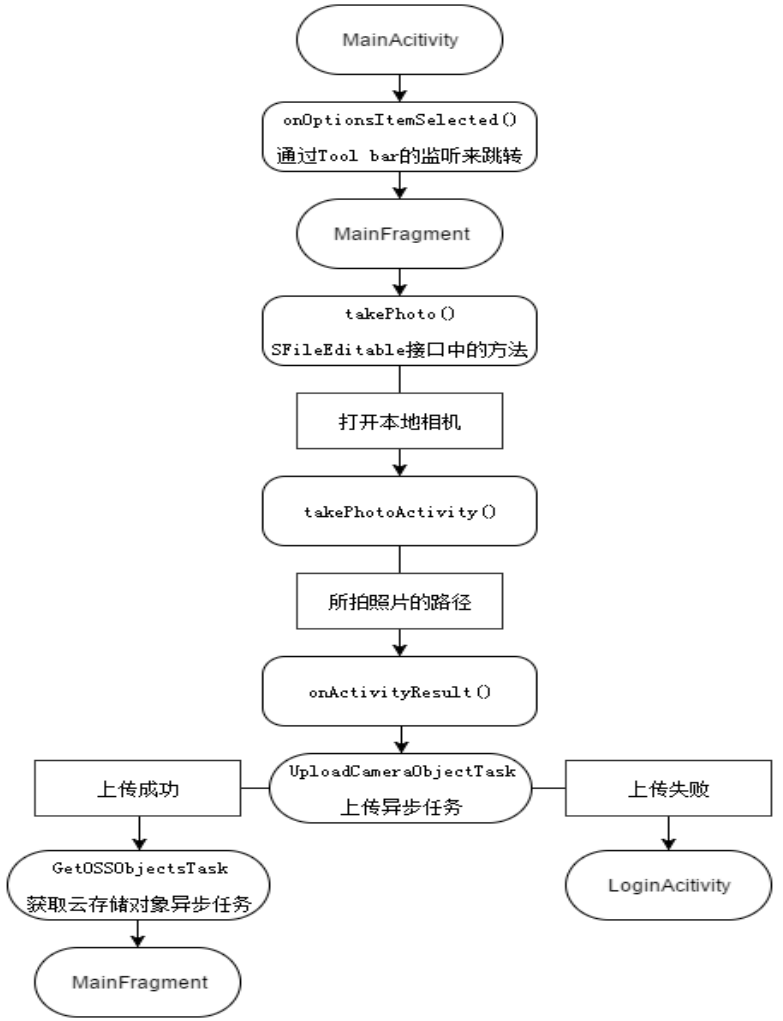




3

拍照上传

功能流程图


MainAcitivity
onOptionsItemSelected()
通过Tool bar的监听来跳转
MainFragment
takePhoto()
SFileEditable接口中的方法
打开本地相机
takePhotoActivity()
所拍照片的路径
onActivityResult()
UploadCameraObjectTask
上传异步任务
上传成功
上传失败
GetOSSObjectsTask
获取云存储对象异步任务
LoginAcitivity
MainFragment

3 拍照上传

功能需求

1）视图层，界面实现

根据原型图设计实现拍照上传窗口View。

3 拍照上传

功能需求

2）控制层

点击右上角下拉菜单中的拍照标签，弹出相机软件，拍照确认。

3）服务层

这里调用openstack中的上传方法，传递的参数为容器名称、文件流、文件类型、文件所要上传至的位置。

4）控制层，返回结果处理

下载可能的情况包含：

文件名已存在，提示用户上传失败；

上传成功，提示用户上传完成。

3 拍照上传

实现步骤

1）导入项目

选择File\Open..，点击弹出选择project92目录下面的项目“swiftstorage”

2）新建一个静态变量与一个URI变量

```
private static final int ACTION_SELECT_CONTENT_FROM_CAMERA = 3;
```

此静态变量用来传输所拍的照片。

```
private Uri mImageUri;
```

此变量用于零时存储照片地址。

3）在MainActivity中对上传标签进行处理

在MainActivity中的方法中对菜单的新建目录选项进行操作。

```
if (id == R.id.action_take_photo) {
    this.currentFragment.takePhoto();
}
```

调用当前Fragment中的takePhoto（）方法

3 拍照上传

实现步骤

4）在MainFragment中完成操作

实现SFileEditable接口下的takephoto（）的方法。

```
@Override
public void takePhoto() {
    takePhotoActivity();
}
```

通过Intent调用本地相机软件。

```
private void takePhotoActivity() {
    try {
        //创建Intent
        Intent cameraIntent = new
Intent(MediaStore.ACTION_IMAGE_CAPTURE);
        //临时文件传递参数
        mImageUri = Uri.fromFile(createPicTempFiles());

cameraIntent.putExtra(MediaStore.EXTRA_OUTPUT,
mImageUri);
        //启动，系统自己选择
        startActivityForResult(cameraIntent,
ACTION_SELECT_CONTENT_FROM_CAMERA);
    } catch (IOException e) {
        e.printStackTrace();
    }
}
```

3 拍照上传

实现步骤

通过onActivityResult方法来回调传输数据。

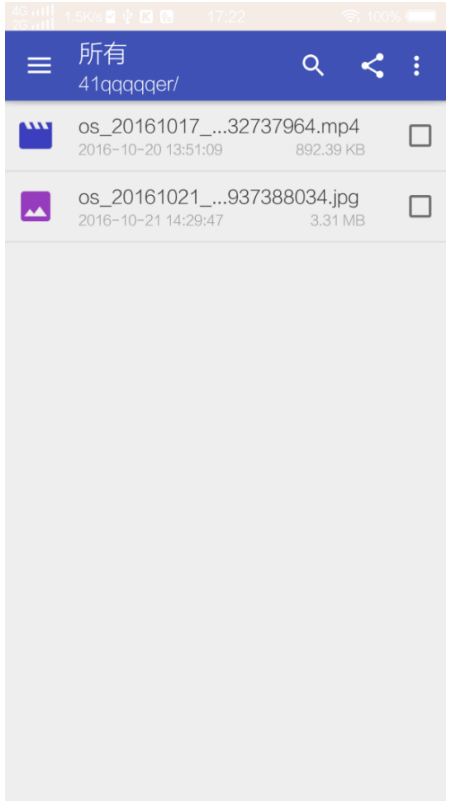
```
@Override
public void onActivityResult(int requestCode, int resultCode, Intent data) {
    super.onActivityResult(requestCode, resultCode, data);
    switch (requestCode) {
        case ACTION_SELECT_CONTENT_FROM_CAMERA:
            //成功返回
            if (resultCode == Activity.RESULT_OK) {
                Uri uri = mImageUri;
                //上传图片
                UploadCameraObjectTask uploadObjectTask = new UploadCameraObjectTask(mImageUri);
                uploadObjectTask.execute();
                break;
            }
default:
            break;
    } }//执行异步任务，在异步任务中调用SDK中的上传的方法，完成后刷新列表。
}
```

3

拍照上传

功能执行测试

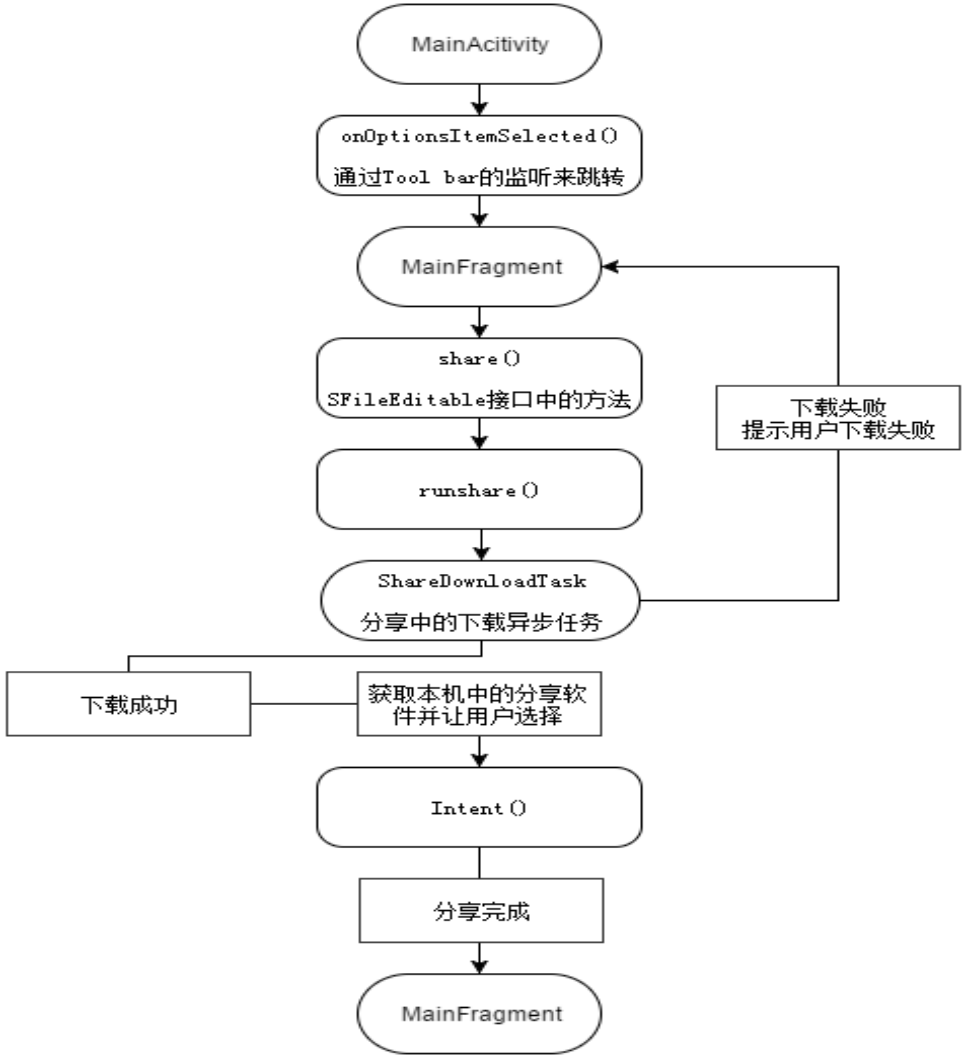
选择项目Android的“app”，点击工具条中运行Run “app”，执行的效果如图所示，填入已经注册好的用户名和密码，进行登录验证。



4

分享

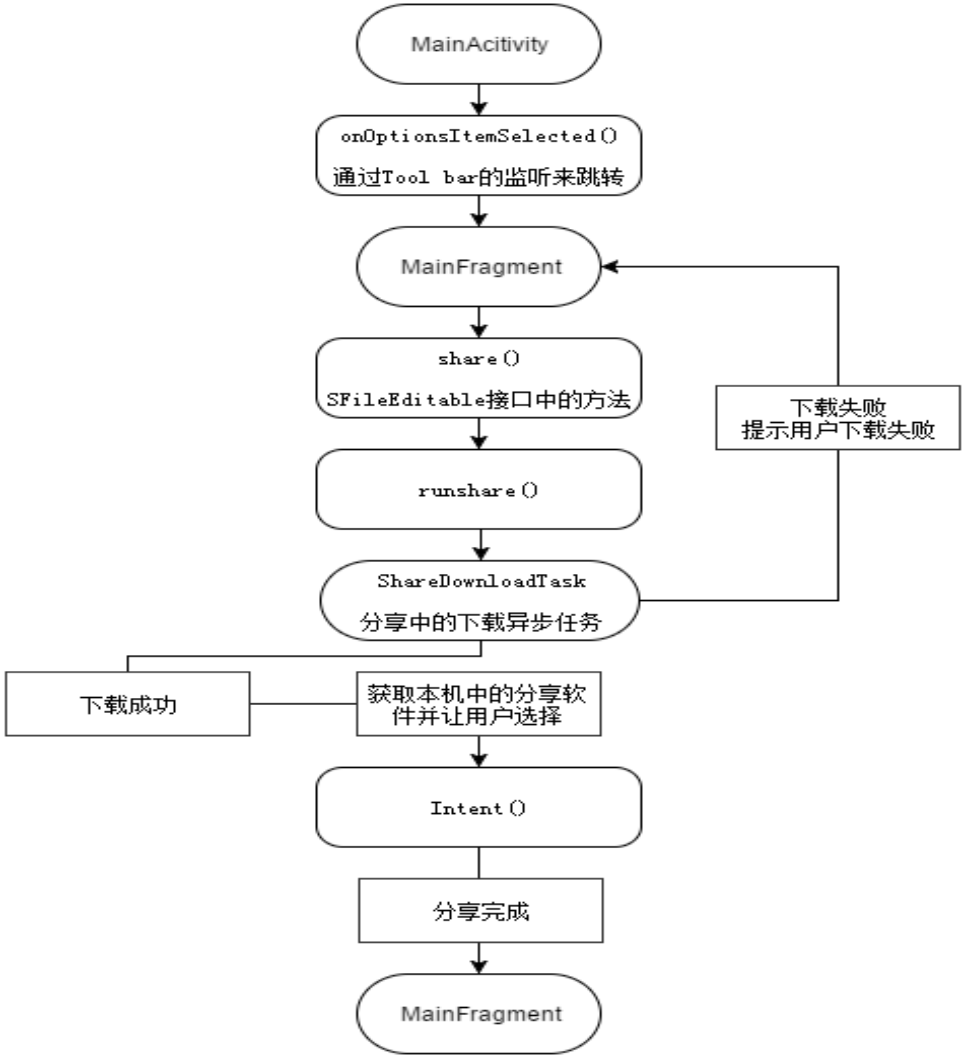
功能流程图


MainAcitivity
onOptionsItemSelected()
通过Tool bar的监听来跳转
MainFragment
share()
SFileEditable接口中的方法
runshare()
ShareDownloadTask
分享中的下载异步任务
下载失败
提示用户下载失败
下载成功
获取本机中的分享软件并让用户选择
Intent()
分享完成
MainFragment

4 分享
功能需求

1）视图层，界面实现

根据原型图设计实现分享窗口View。



4 分享
功能需求

1）视图层，界面实现

根据原型图设计实现分享窗口View。

4 分享

功能需求

2）控制层，用户选择要分享的文件

选择需要上传的文件，点击右上角分享图标。

3）服务层，调用openstack提供的下载接口实现下载

这里调用openstack中的下载方法，传递的参数为容器名称和所要下载文件的名称。

4）控制层，返回结果处理

下载完成后弹出选择分享软件窗口，选择一个分享软件后进行分享

分享可能的情况包含：

（1）手机上没有分享软件，提示用户未发现分享软件，请下载软件；

（2）分享成功。

4 分享

实现步骤

1）导入项目

选择File\Open..，点击弹出选择project94目录下面的项目“swiftstorage”。

2）在MainActivity中对上传标签进行处理

在MainActivity中的方法中对菜单的新建目录选项进行操作。

```
if (id == R.id.action_share) {
    this.currentFragment.share();
}
```

调用当前Fragment中的share（）方法

4 分享

实现步骤

3）在MainFragment中完成操作

实现SFileEditable接口下的upload（）的方法

```
/**
 * 分享
 */
@Override
public void share() {
    runshare();
}
```

4

分享
实现步骤

通过Intent方法调用本地分享软件

```
public void runshare() {
    ShareDownloadTask shareDownloadTask = new ShareDownloadTask(getFirstSelected());
    shareDownloadTask.execute();
    Intent intent = new Intent();
    //获取下载后的本地文件
    File file = new File(getAppState().getOpenStackLocalPath() + cleanName 。(getFirstSelected().getName()));
    //调用本地软件来实现分享
    intent.setAction(Intent.ACTION_SEND);
    intent.setType("*/*");
    Uri u = Uri.fromFile(file);
    intent.putExtra(Intent.EXTRA_STREAM, u);
    startActivity(intent);
}
```

执行异步任务，在异步任务中调用SDK中的上传方法，完成后刷新列表。

4 分享

功能执行测试

选择项目Android的“app”，点击工具条中运行Run “app”，执行的效果如图所示，填入已经注册好的用户名和密码，进行登录验证。