MINI PROJECT

(2020-2021)

SHARE-OPS

A file sharing application for Android

MID TERM REPORT

INSTITUTE OF ENGINEERING AND TECHNOLOGY

Submitted By –

Abhishek Yadav (181500031)

Kuldeep Kumar (181500339)

Arpan Khandelwal (181500129)

Achyut Kumar Tiwari (181500034)

Abhishek Singh (181500027)

Supervised By -

Neeraj Khanna

Contents

Abstract

In this project we are creating a file sharing application over Wi-Fi for android, we all know that the performance of mobile devices, especially smart phones, has been quickly improved for the last few years. Most users take advantage of highly efficient smart phones, and consume the contents in the smart phones longer time than other devices usage time. As a result, users frequently share the contents and the needs of file sharing via smart phones have been increased considerably.

A Web hard-based sharing needs to pay expensive cost for using high-volume file servers as well. In order to overcome such problems, we propose an application for seamless file sharing for the Android devices.

With majority of our population being offline and many with the unstable and not so very high-speed Internet, there must be a way or a tool so that at least they are able to share files between their own or with someone else's devices seamlessly and without any hassle

As in the near future data will also become more and more important so eventually data sharing will also become part of our daily life.

As the technology keeps growing data sharing wirelessly will also become more and efficient and fast

Sharing data via wireless technology will be very efficient as people didn't need to be worry about effect of external means, for instance consider the below scenarios -

- While using data transfer cables there is always a concern of interruption such as cable damage, unplugging of cable.
- While using data storage devices there is a threat of damaging the device.

1.Introduction

In this project, we are trying to build a fast file sharing system which include detection, data transmission. Our problem can be simply described as that when 2 mobile devices encounter with each other, one need to detect quickly whether the other one has the file it need and furthermore, we can divide files into chunks and use some algorithm to identify the existence of the chunks. However, in the real case, it could be multiple devices share the files at the same time.

Why this project? and how will it help?

With majority of our population being offline and many with the unstable and not so very high-speed Internet in India, there must be a way or a tool so that at least they are able to share files between their own or with someone else's devices seamlessly and without any hassle.

Hardware and software requirement for development

Hardware -	Ram of 4gb(minimum) quardcore
Software -	Android Studio

Hardware and software requirement for use

Hardware -	Ram of 3gb (minimum) Available size of 5gb
Software -	Android version of kitkat4.4.4 and above

2. Problem Statement

Often in college or university and also in our working place, we need to share important files with others. Sharing video files, photos, eBooks, PDF, or DOC files become essential sometimes. But often, Wi-Fi connection or mobile data is not available around us so that we can share those files while being online.

We all know about the Internet connection in India, India boasts many of the world's top IT companies, tech entrepreneurs and digital start-ups. Yet, it's also home to nearly 900 million people who do not have access to the internet.

3. Objective of the Project

In our day-to-day life data is the most important thing. sharing of data helps to save our time and internet.

Data can be shared in many ways (using Internet, cables, USB, portable storage dive) but in many forms the user is also occupied but with the help of our project, data can be shared wirelessly without consuming internet and without much effort of user.

As our life becomes more and more progressive, we get less time to do work manually according to our objective people will be able to share data on the go either while travelling or while doing some work without any interruption.

4. Implementation Details

There four main part of the work. They are Peer search, Query, Protocol and Data transmission. We will find peers via Wi-Fi direct. The Wi-Fi direct has many limitations, like only in android 4.0 device, and all the devices need open all the time to wait for connection. Since it is convenient, we will use it to simplify the work, and we will focus on developing the protocols and algorithm. Before we search the peers, we need do some initialization work, such as read file list and set up data structure.

After detecting the peers we need, we can connect with them and establish socket communication. Then we need use some protocols to check the files needed and availability and the exact chunk needed and availability.

All this work will be done in the query process. Hash tables and Bloom filter will be used to detect whether an element is a member of a set. The last part is data transmission. After all the work set, we can transmit the data based on the requirement. Then the file status will be updated in the file list.

5. Progress till Date & The Remaining work

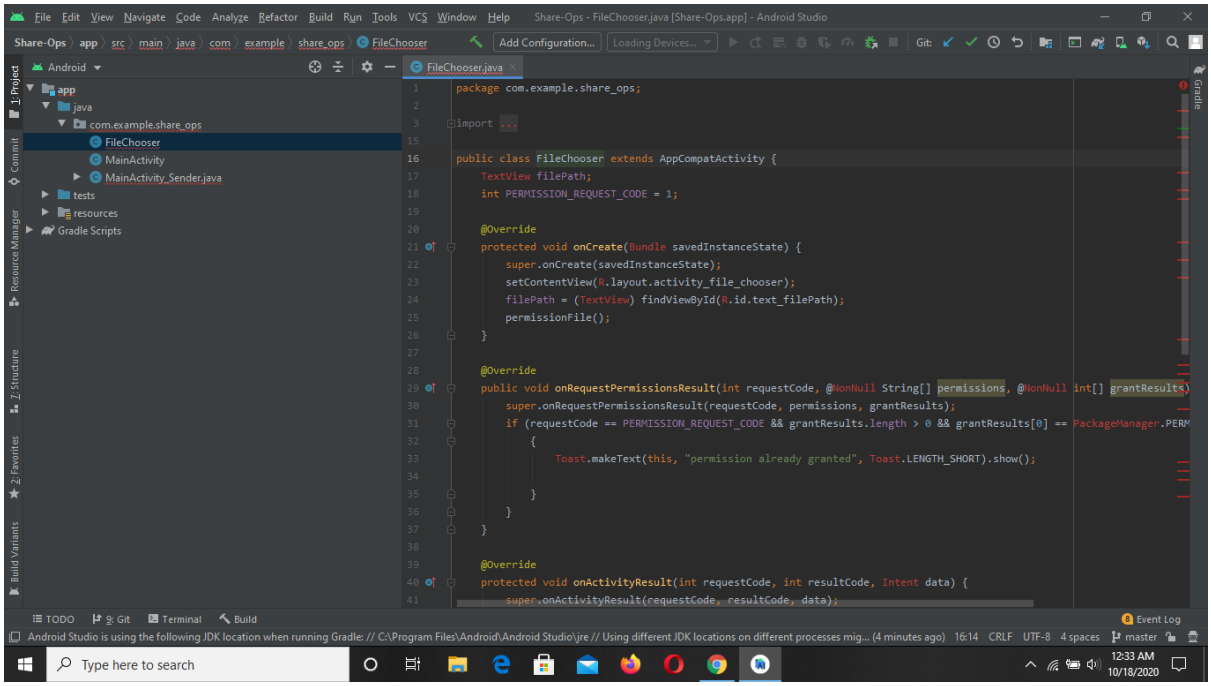
Completed

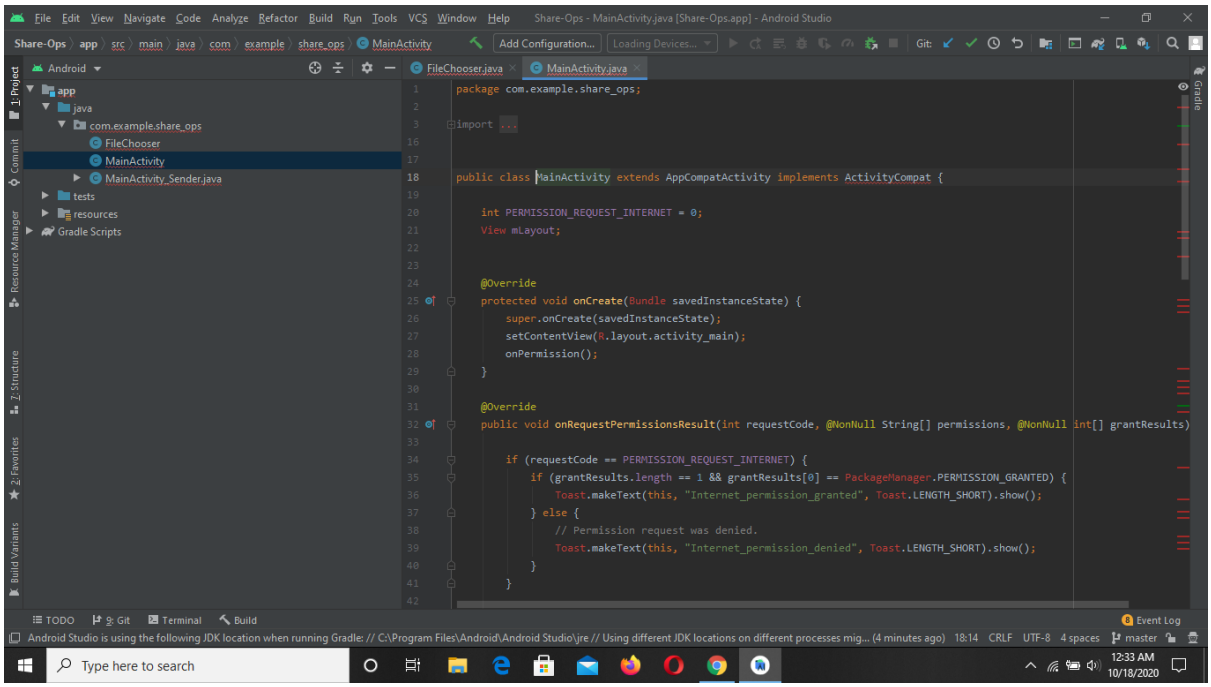
- Basic layout is completed
- File explorer is done
- File selector is done
- File sender is done
- All the parts are completely in sync

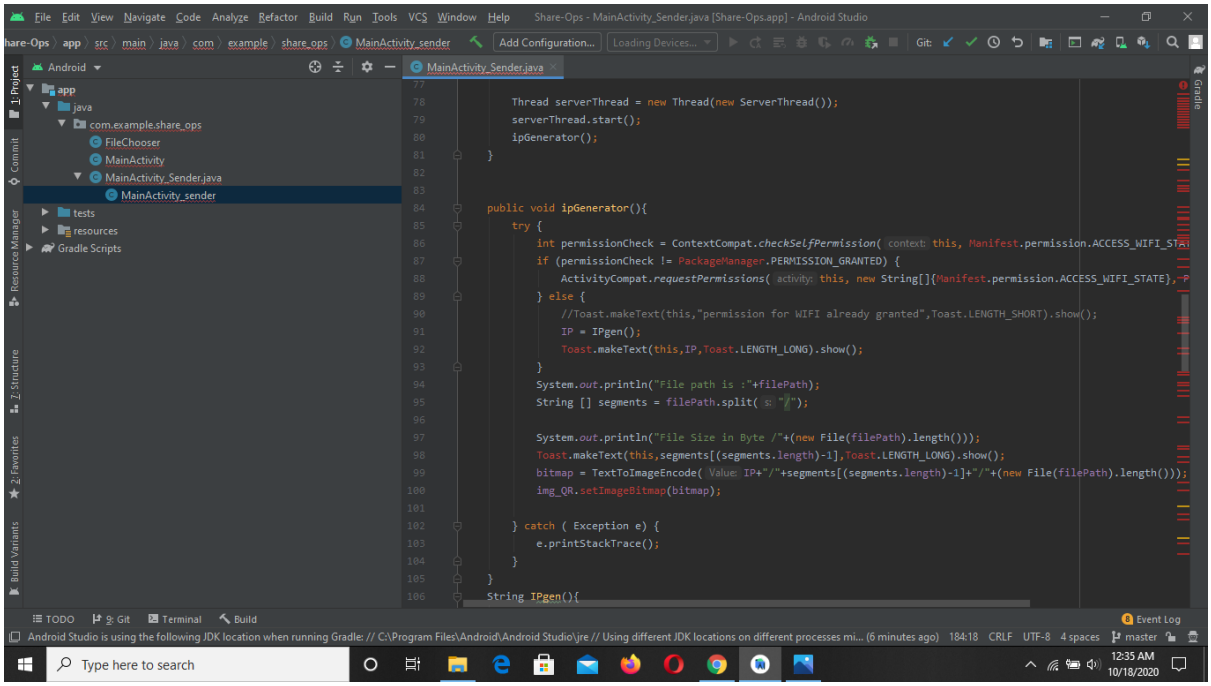
Work to be done

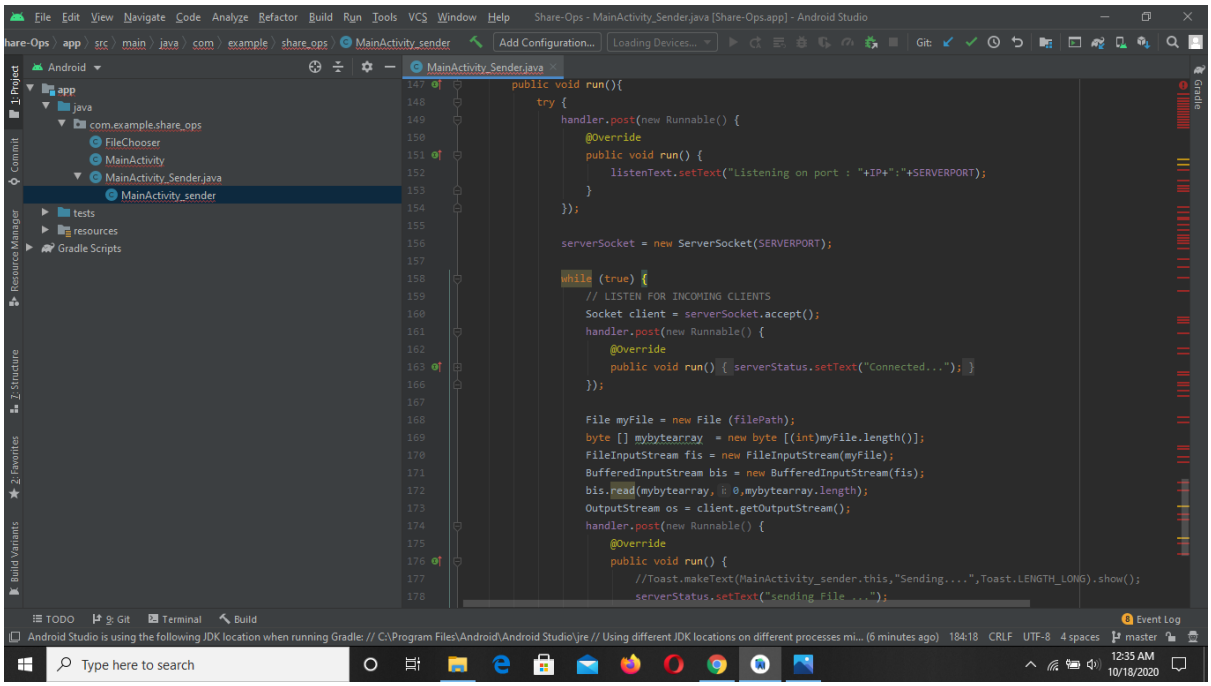
- File receiver
- Optimizing the layout for better function and meeting the comfort for user
- Adding suitable animations

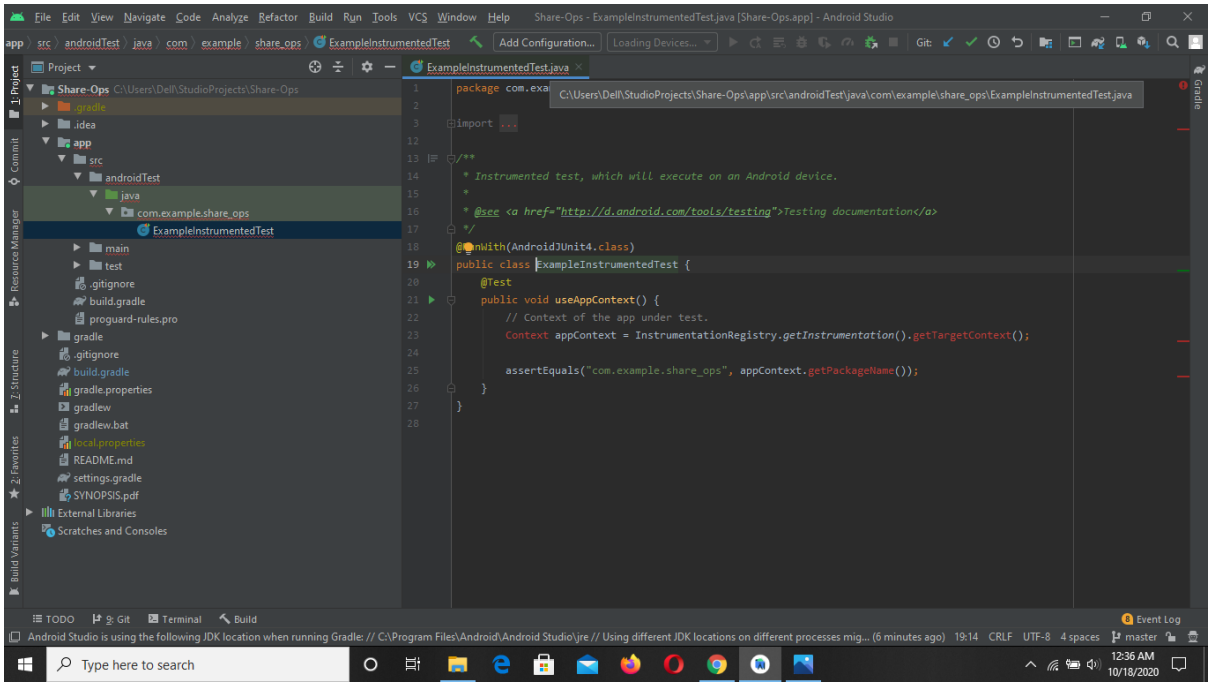
6. Screenshots

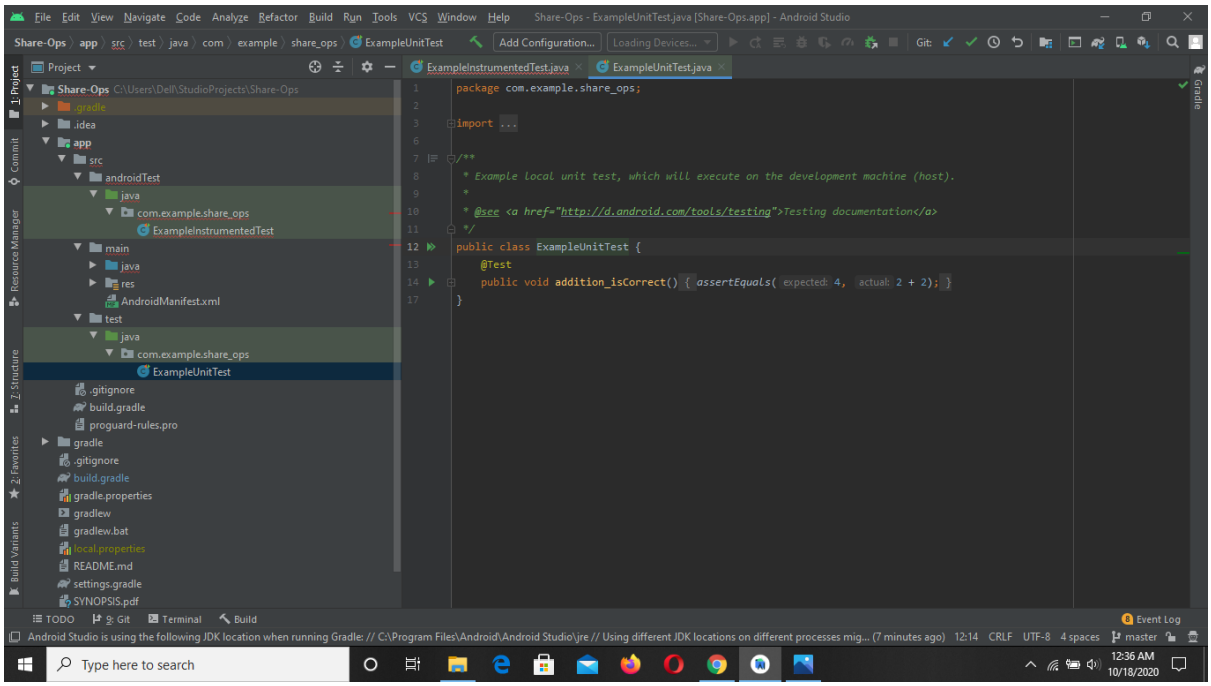


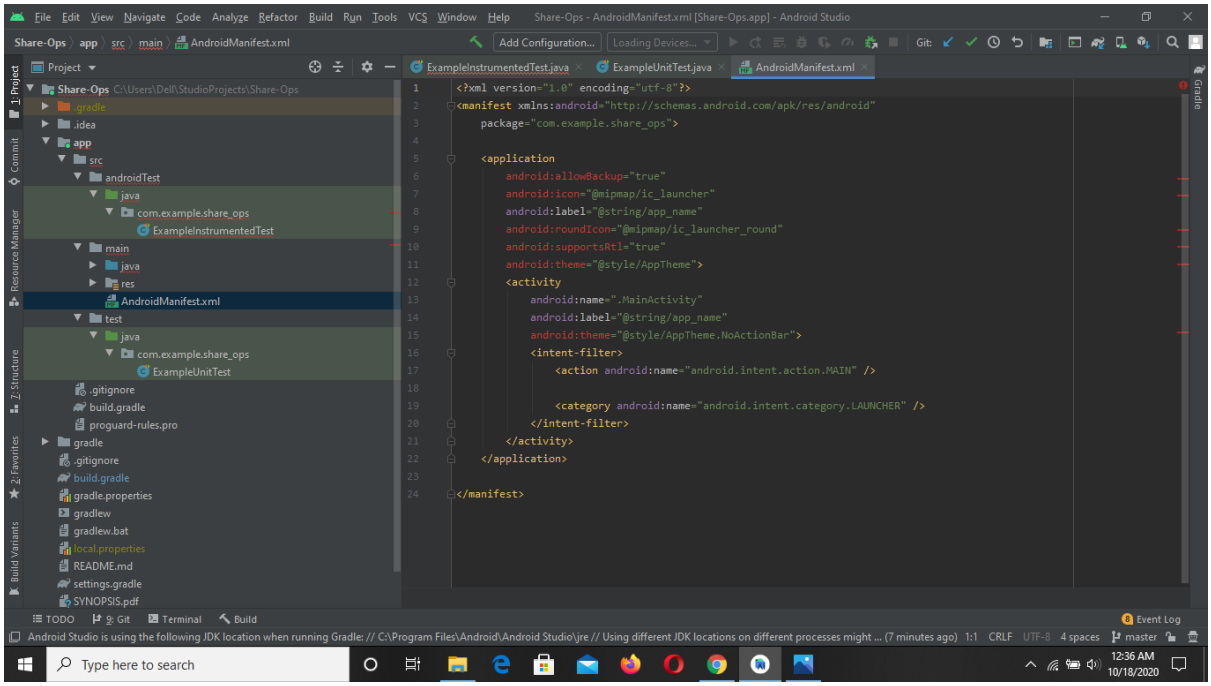



File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help
Share-Ops - MainActivity_Sender.java [Share-Ops.app] - Android Studio
MainActivity_Sender.java
Thread serverThread = new Thread(new ServerThread());
serverThread.start();
ipGenerator();
public void ipGenerator(){
try {
int permissionCheck = ContextCompat.checkSelfPermission(context: this, Manifest.permission.ACCESS_WIFI_STA
if (permissionCheck != PackageManager.PERMISSION_GRANTED) {
ActivityCompat.requestPermissions(activity: this, new String[]{Manifest.permission.ACCESS_WIFI_STATE},
} else {
//Toast.makeText(this,"permission for WIFI already granted",Toast.LENGTH_SHORT).show();
IP = IPgen();
Toast.makeText(this,IP,Toast.LENGTH_LONG).show();
}
System.out.println("File path is :"+filePath);
String [] segments = filePath.split(s: "/");
System.out.println("File Size in Byte /"+(new File(filePath).length()));
Toast.makeText(this,segments[(segments.length)-1],Toast.LENGTH_LONG).show();
bitmap = TextToImageEncode(Value: IP+"/"+segments[(segments.length)-1]+"/"+(new File(filePath).length()));
img_QR.setImageBitmap(bitmap);
} catch (Exception e) {
e.printStackTrace();
}
}
String IPgen(){
TODO
Git
Terminal
Build
Event Log
Android Studio is using the following JDK location when running Gradle: // C:\Program Files\Android\Android Studio\jre // Using different JDK locations on different processes mi... (6 minutes ago)
184:18 CRLF UTF-8 4 spaces master
Type here to search
12:35 AM
10/18/2020


File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help
Share-Ops - MainActivity_Sender.java [Share-Ops.app] - Android Studio
MainActivity_Sender.java
public void run(){
try {
handler.post(new Runnable() {
@Override
public void run() {
listenText.setText("Listening on port : "+IP+":"+SERVERPORT);
}
});
serverSocket = new ServerSocket(SERVERPORT);
while (true) {
// LISTEN FOR INCOMING CLIENTS
Socket client = serverSocket.accept();
handler.post(new Runnable() {
@Override
public void run() { serverStatus.setText("Connected..."); }
});
File myFile = new File (filePath);
byte [] mybytearray = new byte [(int)myFile.length()];
FileInputStream fis = new FileInputStream(myFile);
BufferedInputStream bis = new BufferedInputStream(fis);
bis.read(mybytearray, 0,mybytearray.length);
OutputStream os = client.getOutputStream();
handler.post(new Runnable() {
@Override
public void run() {
//Toast.makeText(MainActivity_sender.this,"Sending....",Toast.LENGTH_LONG).show();
serverStatus.setText("sending File ...");
TODO
Git
Terminal
Build
Event Log
Android Studio is using the following JDK location when running Gradle: // C:\Program Files\Android\Android Studio\jre // Using different JDK locations on different processes mi... (6 minutes ago)
184:18 CRLF UTF-8 4 spaces master
Type here to search
12:35 AM
10/18/2020


File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help
Share-Ops - ExampleInstrumentedTest.java [Share-Ops.app] - Android Studio
Add Configuration...
Loading Devices...
Project
Share-Ops C:\Users\Dell\StudioProjects\Share-Ops
.gradle
.idea
app
src
androidTest
java
com.example.share_ops
ExampleInstrumentedTest
main
test
.gitignore
build.gradle
proguard-rules.pro
gradle
.gitignore
build.gradle
gradle.properties
gradlew
gradlew.bat
local.properties
README.md
settings.gradle
SYNOPSIS.pdf
External Libraries
Scratches and Consoles
ExampleInstrumentedTest.java
C:\Users\Dell\StudioProjects\Share-Ops\app\src\androidTest\java\com\example\share_ops\ExampleInstrumentedTest.java
package com.exa
import ...
/**
* Instrumented test, which will execute on an Android device.
*
* @see Testing documentation
*/
@RunWith(AndroidJUnit4.class)
public class ExampleInstrumentedTest {
@Test
public void useAppContext() {
// Context of the app under test.
Context appContext = InstrumentationRegistry.getInstrumentation().getTargetContext();
assertEquals("com.example.share_ops", appContext.getPackageName());
}
}
TODO
Git
Terminal
Build
Event Log
Android Studio is using the following JDK location when running Gradle: // C:\Program Files\Android\Android Studio\jre // Using different JDK locations on different processes mig... (6 minutes ago)
19:14 CRLF UTF-8 4 spaces master
Type here to search
12:36 AM
10/18/2020


File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help
Share-Ops - ExampleUnitTest.java [Share-Ops.app] - Android Studio
Share-Ops app src test java com example share_ops ExampleUnitTest
Add Configuration...
Loading Devices...
Project
Share-Ops C:\Users\Dell\StudioProjects\Share-Ops
.gradle
.idea
app
src
androidTest
java
com.example.share_ops
ExampleInstrumentedTest
main
java
res
AndroidManifest.xml
test
java
com.example.share_ops
ExampleUnitTest
.gitignore
build.gradle
proguard-rules.pro
gradle
.gitignore
build.gradle
gradle.properties
gradlew
gradlew.bat
local.properties
README.md
settings.gradle
SYNOPSIS.pdf
ExampleInstrumentedTest.java
ExampleUnitTest.java
package com.example.share_ops;
import ...
/**
* Example local unit test, which will execute on the development machine (host).
*
* @see Testing documentation
*/
public class ExampleUnitTest {
@Test
public void addition_isCorrect() { assertEquals(expected: 4, actual: 2 + 2); }
}
TODO
Git
Terminal
Build
Event Log
Android Studio is using the following JDK location when running Gradle: // C:\Program Files\Android\Android Studio\jre // Using different JDK locations on different processes mig... (7 minutes ago)
12:14 CRLF UTF-8 4 spaces master
Type here to search
12:36 AM
10/18/2020



7. References

- Courses available on udemy
  https://www.udemy.com/course/androidappcourse/

- Tips from android developer site
  https://developer.android.com/training/secure-file-sharing