(a). I implemented option2.

(b) The test image should be placed at the following address for the server to use

This file path =>\Servers\src\image

Batch file:

A batch file for running the program:

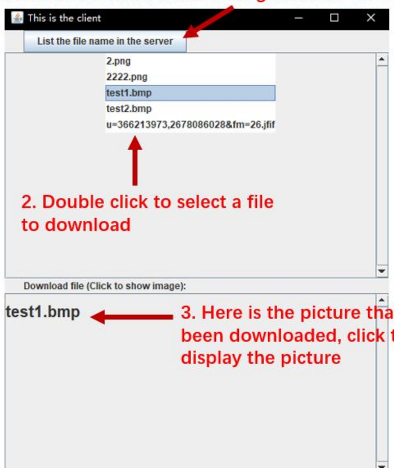
Runme.bat   Please run the Runme.bat to compile and run all the java file.

Client & Server:

A GUI interface that allows the user to display the list of files that are available on the server

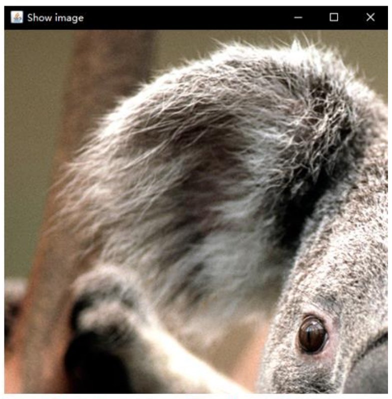
A GUI interface that allows the user to select a file from the list of available files to download

A GUI interface that allows the user to display the contents of the downloaded file.


1. Double click the button to get the server file list
This is the client
List the file name in the server
2. Double click to select a file to download
Download file (Click to show image):
3. Here is the picture that has been downloaded, click to display the picture

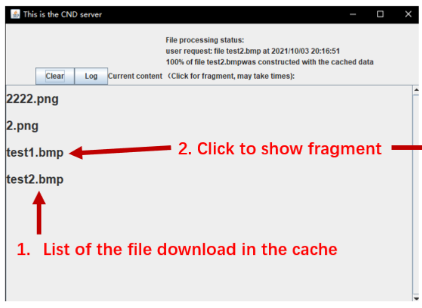
2.png
2222.png
test1.bmp
test2.bmp
u=366213973,2678086028&fm=26.jfif

test1.bmp



A GUI interface for viewing the list of cached files and the list is kept correctly.

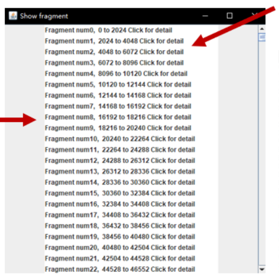
A GUI interface for viewing the contents of the selected data fragment cached on the cache server.


This is the CND server
Clear
Log
2. Click to show fragment
1. List of the file download in the cache

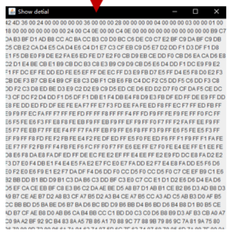
```
File processing status:
user request: file test2.bmp at 2021/10/03 20:16:51
100% of file test2.bmpwas constructed with the cached data
```

Current content (Click for fragment, may take times):

2222.png
2.png
test1.bmp
test2.bmp

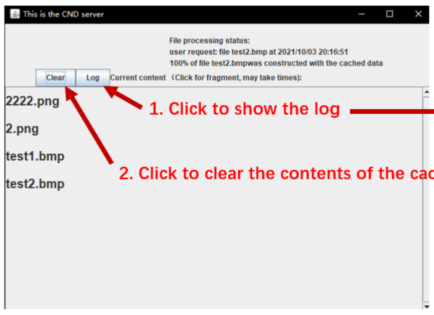


```
Fragment num0,  0 to 2024 Click for detail
Fragment num1,  2024 to 4048 Click for detail
Fragment num2,  4048 to 6072 Click for detail
Fragment num3,  6072 to 8096 Click for detail
Fragment num4,  8096 to 10120 Click for detail
Fragment num5,  10120 to 12144 Click for detail
Fragment num6,  12144 to 14168 Click for detail
Fragment num7,  14168 to 16192 Click for detail
Fragment num8,  16192 to 18216 Click for detail
Fragment num9,  18216 to 20240 Click for detail
Fragment num10,  20240 to 22264 Click for detail
Fragment num11,  22264 to 24288 Click for detail
Fragment num12,  24288 to 26312 Click for detail
Fragment num13,  26312 to 28336 Click for detail
Fragment num14,  28336 to 30360 Click for detail
Fragment num15,  30360 to 32384 Click for detail
Fragment num16,  32384 to 34408 Click for detail
Fragment num17,  34408 to 36432 Click for detail
Fragment num18,  36432 to 38456 Click for detail
Fragment num19,  38456 to 40480 Click for detail
Fragment num20,  40480 to 42504 Click for detail
Fragment num21,  42504 to 44528 Click for detail
Fragment num22,  44528 to 46552 Click for detail
```



A GUI interface for viewing the cache's log and the log is kept correctly.

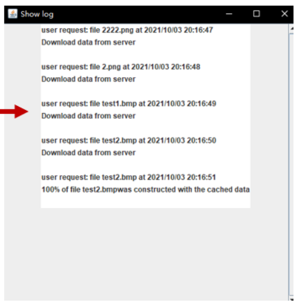
A GUI interface for clearing the contents of the cache and the content can be cleared correctly.



```
File processing status:
user request: file test2.bmp at 2021/10/03 20:16:51
100% of file test2.bmpwas constructed with the cached data
```

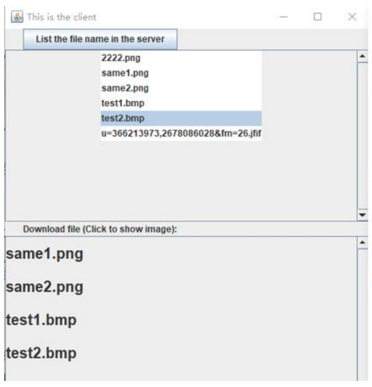
Current content (Click for fragment, may take times):

2222.png
2.png
test1.bmp
test2.bmp



```
user request: file 2222.png at 2021/10/03 20:16:47
Download data from server

user request: file 2.png at 2021/10/03 20:16:48
Download data from server

user request: file test1.bmp at 2021/10/03 20:16:49
Download data from server

user request: file test2.bmp at 2021/10/03 20:16:50
Download data from server

user request: file test2.bmp at 2021/10/03 20:16:51
100% of file test2.bmpwas constructed with the cached data
```

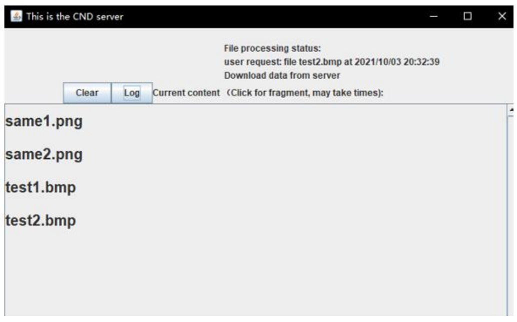
Case 1: Download files with different names but similar contents.
Case 2: Download files with the same name but slightly different contents.



```
2222.png
same1.png
same2.png
test1.bmp
test2.bmp
u=366213973,2678086028&fm=26.jfif
```

Download file (Click to show image):

```
same1.png
same2.png
test1.bmp
test2.bmp
```



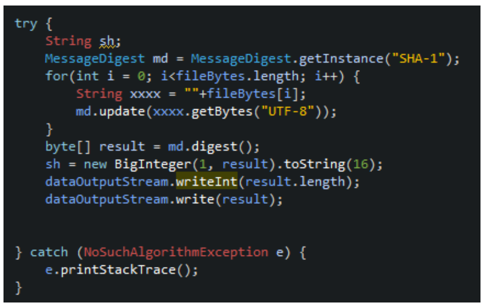
```
File processing status:
user request: file test2.bmp at 2021/10/03 20:32:39
Download data from server
```

Current content（Click for fragment, may take times):

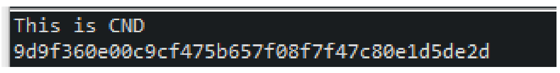
```
same1.png
same2.png
test1.bmp
test2.bmp
```

Techniques:
I did not use any third-party libraries, so I can run it directly.

(c). First, the server sends the name of the downloaded file to the cache. If the name of the file already exists in the cache, the HASH of the file will be checked. If the HASH is the same as the HASH of the cached file, the file will not be downloaded. The cache will use itself data to send to the client. If the HASH is different, the file will be download and replace the last file which has the same name.
The hash is using SHA-1

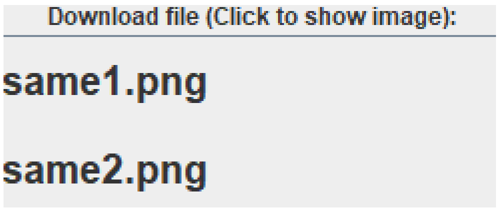


```java
try {
    String sh;
    MessageDigest md = MessageDigest.getInstance("SHA-1");
    for(int i = 0; i<fileBytes.length; i++) {
        String xxxx = ""+fileBytes[i];
        md.update(xxxx.getBytes("UTF-8"));
    }
    byte[] result = md.digest();
    sh = new BigInteger(1, result).toString(16);
    dataOutputStream.writeInt(result.length);
    dataOutputStream.write(result);


} catch (NoSuchAlgorithmException e) {
    e.printStackTrace();
}
```



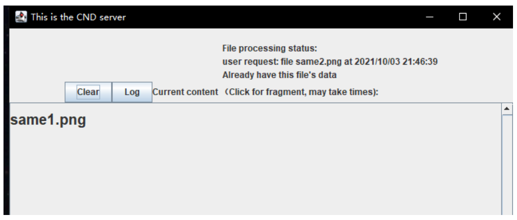
```
This is CND
9d9f360e00c9cf475b657f08f7f47c80e1d5de2d
```

As shown in the figure below, because same1 and same2 are the same picture, the client successfully downloaded same1 and same2, but CND only downloaded and cached same1 from the server.



Download file (Click to show image):

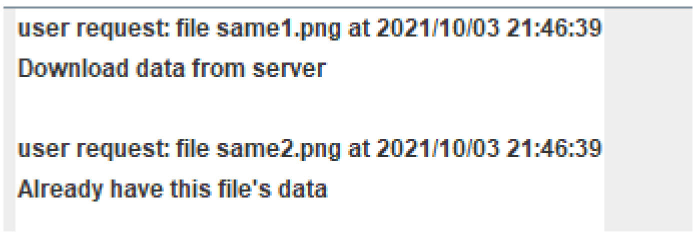
```
same1.png
same2.png
```



```
File processing status:
user request: file same2.png at 2021/10/03 21:46:39
Already have this file's data
```

Current content（Click for fragment, may take times):

```
same1.png
```



```
user request: file same1.png at 2021/10/03 21:46:39
Download data from server

user request: file same2.png at 2021/10/03 21:46:39
Already have this file's data
```