1. The implementation of returning binary images.

There are four kinds of the picture can be returned by the server which is JPG, JEPG, GIF, and PNG. The logic of the implementation for diverse kinds of pictures are similar. At first, the server will recognise the request type. Then, the server will generate a different file name depends on the type of the requested file. For example, the following codes show the method for creating the file name of a JPG picture.

```
if(request.contains("jpg")) {
requestFile = request.substring(request.indexOf("/"), request.indexOf(".jpg"));
requestAddress = dir + requestFile + ".jpg";
}
```

After that, the server will create an object file. Moreover, there is a judgment for the file existence. If the file exists, then the value of the response code will be "200", and the content type will be "Content-Type: image/jpg". The picture will be read by input stream and printed out to cline by output stream. The test command line will be like this: curl -s GET localhost:8080/beer.jpg

2. Logging

The server will record the information when the client sends a request to the server. The information will involve the date of the request time, request type and the response code. The method, "Logging (String requestType, String responseCode)", can implement this function. There are two params in this method which is request type and the response code. This method will execute after the judgment that if the requested file exists, therefore, it can obtain the response code after previous judgment. Except for these, the date of the request should also record into the file. The server will get the time by the object Date, then, connecting the request type and the response code to create a new character string. Then, this piece of information will be written into a text file by file output stream. The user can check the test results in the text file which is in the project directory.

3. Supporting the method DELETE

In addition to GET and HEAD, the server also can accept the DELETE request. Like the other two methods, the server will generate the name of the requested file. Then, if the file can be found in the resource folder, the file will be deleted from the folder, and the response code will return to "HTTP/1.1 301 Moved Permanently", otherwise, the response will return to HTTP/1.1 404 Not Found. The implementation of deleting a file is like the codes shown below:

```
if(file.exists()&&file.isFile()) {
    file.delete();
    responseCode = "HTTP/1.1 301 Moved Permanently";
 }else {
    responseCode = "HTTP/1.1 404 Not Found";
    }
```

The test command line could be: curl -s DELETE localhost:8080/beer.jpg