NA-1 Spring 2017 Project #2

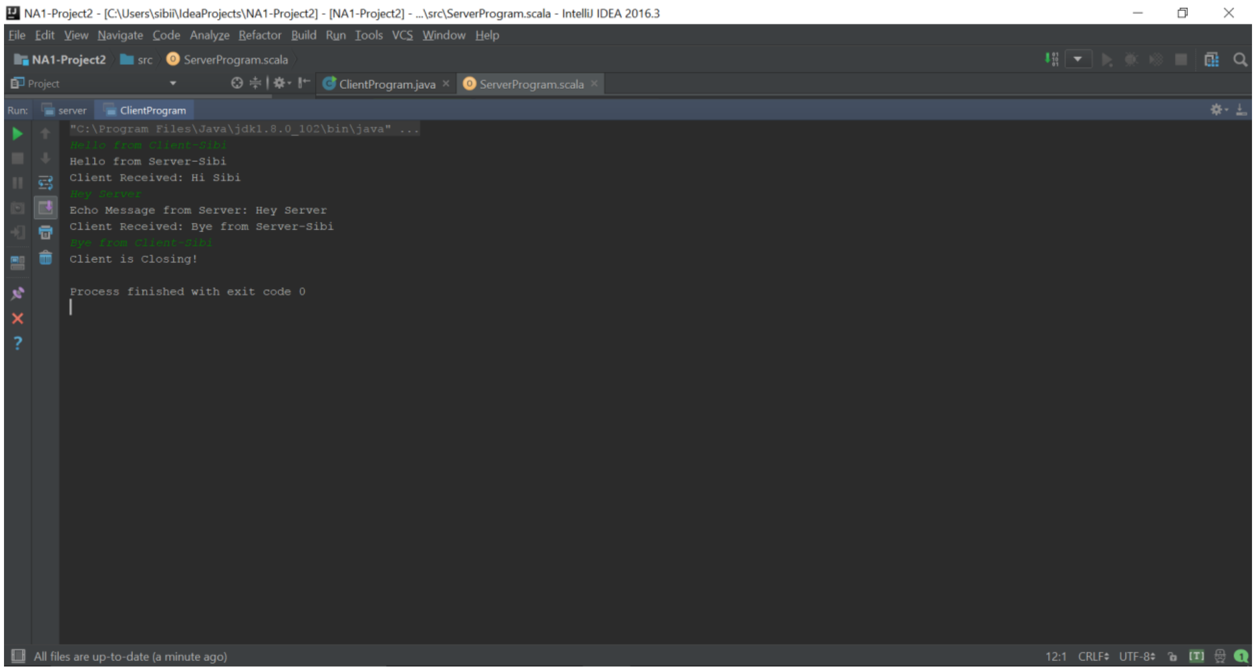
Socket Programming

a) Client-Server Message Transfer

Client Program:

```
import java.io.*;
import java.net.*;

class ClientProgram
{
    public static void main(String argv[]) throws Exception
    {
        String fClient;
        String tServer;
        Socket clientSocket = new Socket("192.168.235.1", 1994);
        BufferedReader inFromUser =new BufferedReader(new
InputStreamReader(System.in));
        PrintWriter outToServer = new
PrintWriter(clientSocket.getOutputStream(),true);
        BufferedReader inFromServer = new BufferedReader(new
InputStreamReader(clientSocket.getInputStream()));

        while (true)
        {
            tServer = inFromUser.readLine();
            outToServer.println(tServer);
            if (tServer.equals("Bye from Client-Sibi"))
            {
                outToServer.println (tServer) ;
                clientSocket.close();
                System.out.println("Client is Closing!");
                break;
            }
            fClient = inFromServer.readLine();
            System.out.println(fClient);

            fClient = inFromServer.readLine();
            System.out.println("Client Received: "+fClient);
            outToServer.println("Echo Message from Client: "+fClient);

        }
    }
}
```

Client Output:

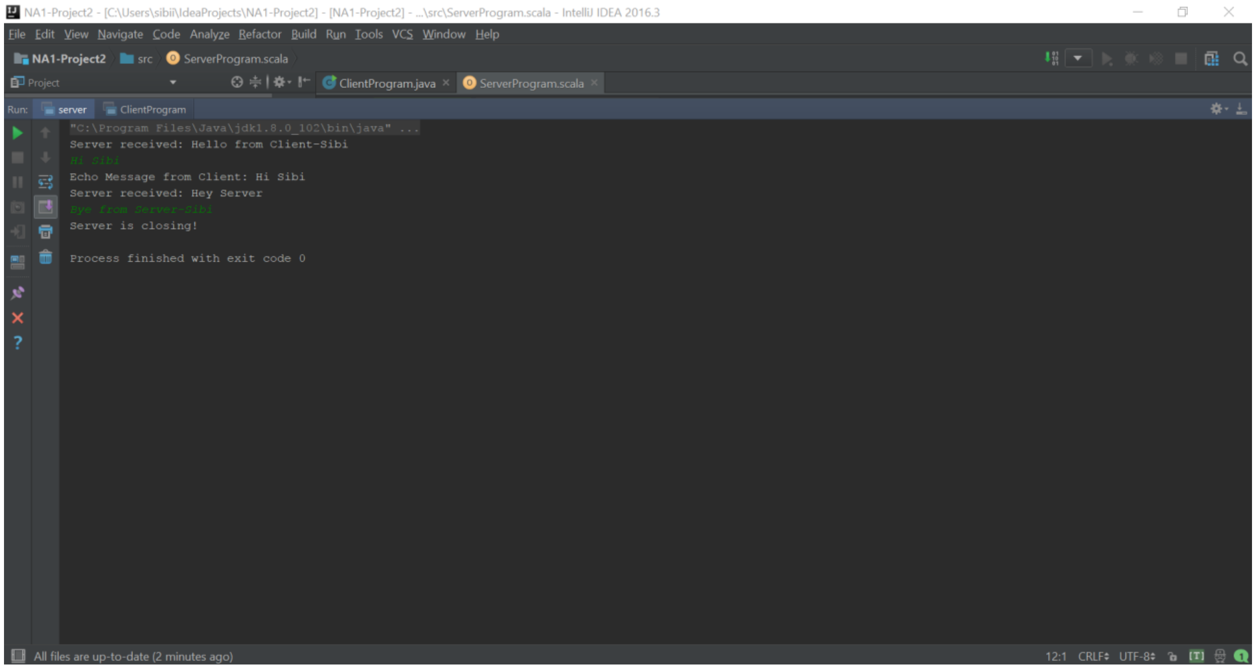

NA1-Project2 - [C:\Users\sibii\IdeaProjects\NA1-Project2] - [NA1-Project2] - ...\src\ServerProgram.scala - IntelliJ IDEA 2016.3
File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help
NA1-Project2 src ServerProgram.scala
Project
ClientProgram.java ServerProgram.scala
Run: server ClientProgram
All files are up-to-date (a minute ago)
12:1 CRLF UTF-8

```
"C:\Program Files\Java\jdk1.8.0_102\bin\java" ...
Hello from Client-Sibi
Hello from Server-Sibi
Client Received: Hi Sibi
Hey Server
Echo Message from Server: Hey Server
Client Received: Bye from Server-Sibi
Bye from Client-Sibi
Client is Closing!

Process finished with exit code 0
```

Server Program:

```scala
import java.io.{BufferedReader, InputStreamReader, PrintStream}
import java.net.ServerSocket
import scala.util.control.Breaks._
object ServerProgram {
  def main(args: Array[String]): Unit = {
    val server = new ServerSocket(1994)
    val client = server.accept
    val in = new BufferedReader(new InputStreamReader(client.getInputStream()))
    val out = new PrintStream(client.getOutputStream)
    breakable {
     while (true) {
       var fClient = in.readLine
       println("Server received: " + fClient)
       if (fClient.equals("Hello from Client-Sibi")){
         out.println("Hello from Server-Sibi")
       }
       else {
         out.println("Echo Message from Server: "+fClient)
       }
       val tClient = readLine()
       out.println(tClient)
       if (tClient.equals("Bye from Server-Sibi")) {
         client.close;
         server.close;
         println("Server is closing!")
         break
       }
       fClient=in.readLine()
       println(fClient)}
    }  }
}
```

Server Output:



NA1-Project2 - [C:\Users\sibi\IdeaProjects\NA1-Project2] - [NA1-Project2] - ...\src\ServerProgram.scala - IntelliJ IDEA 2016.3
File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help
NA1-Project2 src ServerProgram.scala
Project
ClientProgram.java
ServerProgram.scala
Run: server ClientProgram
All files are up-to-date (2 minutes ago)
12:1 CRLF UTF-8

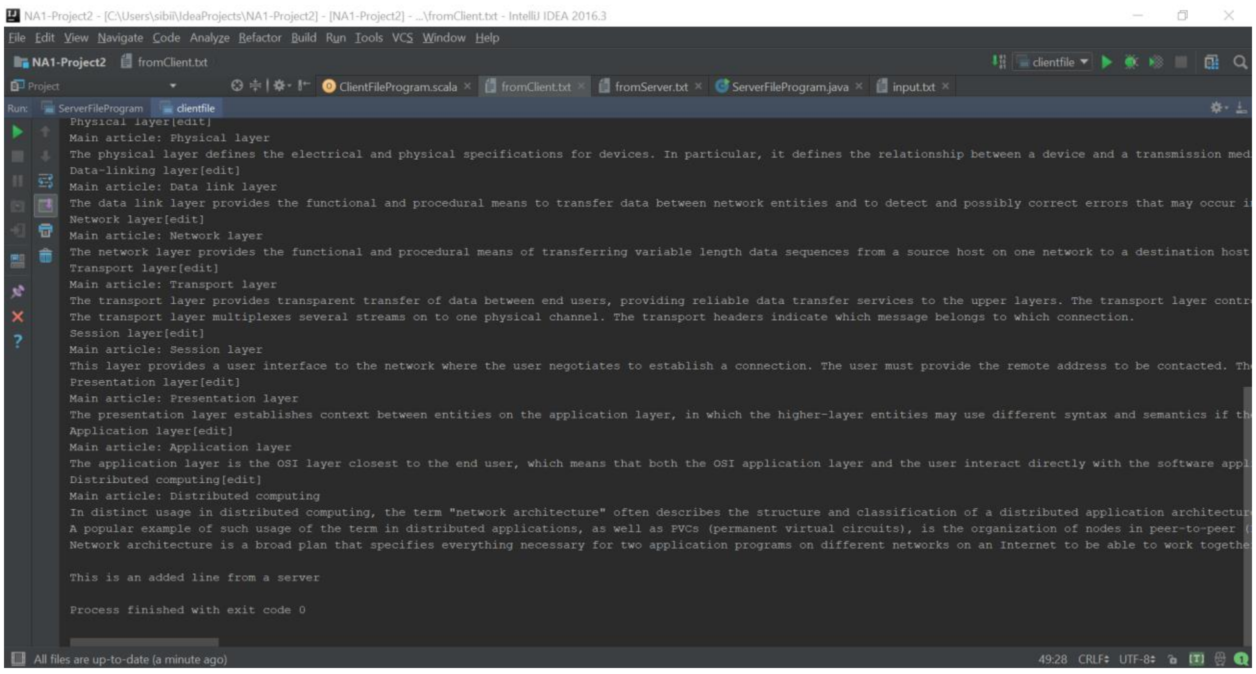
```
"C:\Program Files\Java\jdk1.8.0_102\bin\java" ...
Server received: Hello from Client-Sibi
Hi Sibi
Echo Message from Client: Hi Sibi
Server received: Hey Server
Bye from Server-Sibi
Server is closing!

Process finished with exit code 0
```

b) Client Server File Transfer

Client Program:

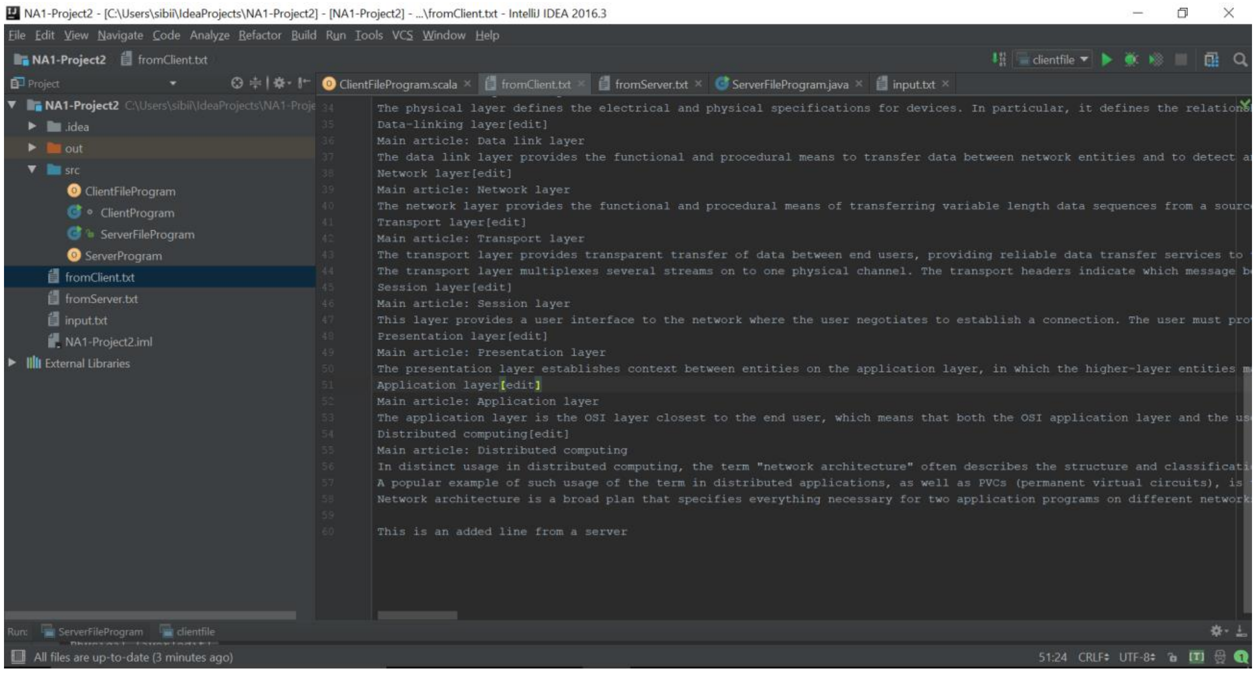
```
import java.io.{File, FileInputStream, FileOutputStream}
import java.net.Socket
import java.util.Scanner
import scala.util.control.Breaks._


object ClientFileProgram {
  def main(args: Array[String]): Unit = {
    val sock: Socket = new Socket("192.168.235.1", 1996)
    val out= sock.getOutputStream
    //send file
    val fileInputStream: FileInputStream = new FileInputStream("input.txt")
    val buffer: Array[Byte] = new Array[Byte](64 * 1024)
    var bytesRead: Int = 0
    breakable {
      while ((bytesRead = fileInputStream.read(buffer)) != -1) {
        if (bytesRead > 0) {
          out.write(buffer, 0, bytesRead)
        }
        break
      }
    }

    sock.close()

    val sock1: Socket = new Socket("192.168.235.1", 1995)
    val in = sock1.getInputStream

    val fileOutputStream: FileOutputStream = new FileOutputStream("fromServer.txt")
    var bytesRead2: Int = 0
    val buffer1: Array[Byte] = new Array[Byte](64 * 1024)
    breakable {
      while ((bytesRead2 = in.read(buffer1)) != -1) {
        fileOutputStream.write(buffer1, 0, bytesRead2)
        break
      }
    }

    val input: Scanner = new Scanner(new File("fromServer.txt"))
    System.out.println("***************File received from Server***************")
    while (input.hasNextLine) System.out.println(input.nextLine)

  }
}
```

Client Output:


NA1-Project2 - [C:\Users\sibii\IdeaProjects\NA1-Project2] - [NA1-Project2] - ...\fromClient.txt - IntelliJ IDEA 2016.3
File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help
NA1-Project2
fromClient.txt
clientfile
Project
ClientFileProgram.scala
fromClient.txt
fromServer.txt
ServerFileProgram.java
input.txt
Run:
ServerFileProgram
clientfile

```
Physical layer[edit]
Main article: Physical layer
The physical layer defines the electrical and physical specifications for devices. In particular, it defines the relationship between a device and a transmission med
Data-linking layer[edit]
Main article: Data link layer
The data link layer provides the functional and procedural means to transfer data between network entities and to detect and possibly correct errors that may occur i
Network layer[edit]
Main article: Network layer
The network layer provides the functional and procedural means of transferring variable length data sequences from a source host on one network to a destination host
Transport layer[edit]
Main article: Transport layer
The transport layer provides transparent transfer of data between end users, providing reliable data transfer services to the upper layers. The transport layer contr
The transport layer multiplexes several streams on to one physical channel. The transport headers indicate which message belongs to which connection.
Session layer[edit]
Main article: Session layer
This layer provides a user interface to the network where the user negotiates to establish a connection. The user must provide the remote address to be contacted. Th
Presentation layer[edit]
Main article: Presentation layer
The presentation layer establishes context between entities on the application layer, in which the higher-layer entities may use different syntax and semantics if th
Application layer[edit]
Main article: Application layer
The application layer is the OSI layer closest to the end user, which means that both the OSI application layer and the user interact directly with the software appl
Distributed computing[edit]
Main article: Distributed computing
In distinct usage in distributed computing, the term "network architecture" often describes the structure and classification of a distributed application architectur
A popular example of such usage of the term in distributed applications, as well as PVCs (permanent virtual circuits), is the organization of nodes in peer-to-peer (
Network architecture is a broad plan that specifies everything necessary for two application programs on different networks on an Internet to be able to work togethe

This is an added line from a server

Process finished with exit code 0
```

Client File:

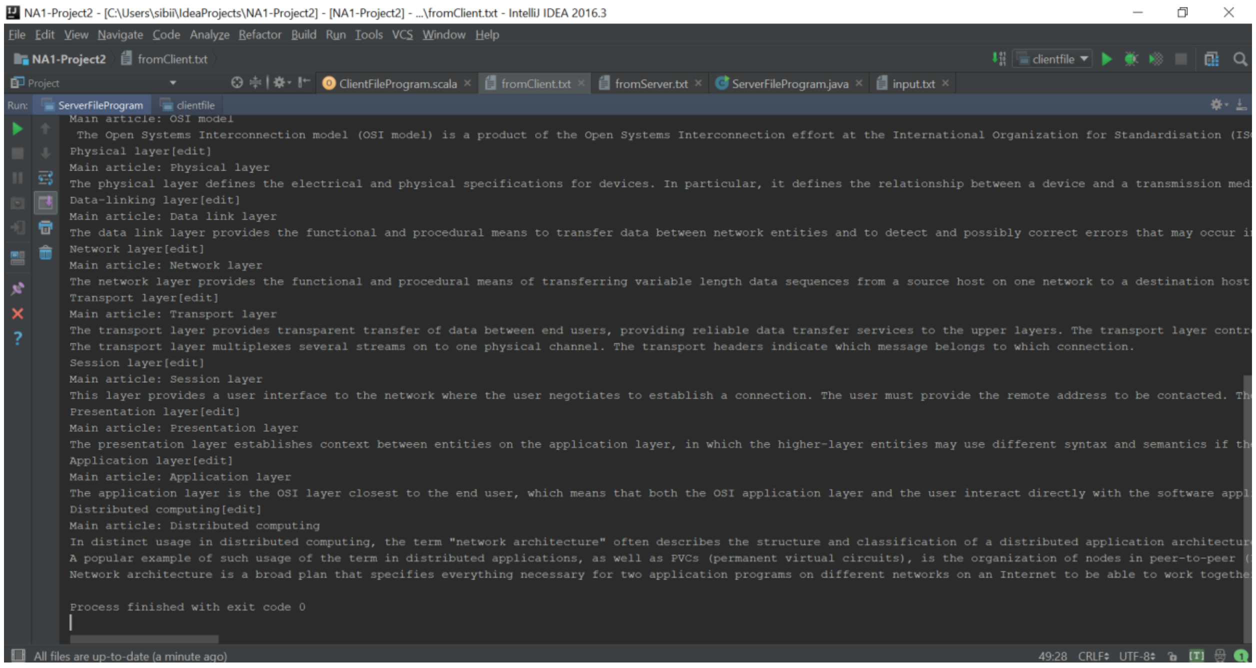

```
The physical layer defines the electrical and physical specifications for devices. In particular, it defines the relation
Data-linking layer[edit]
Main article: Data link layer
The data link layer provides the functional and procedural means to transfer data between network entities and to detect a
Network layer[edit]
Main article: Network layer
The network layer provides the functional and procedural means of transferring variable length data sequences from a sourc
Transport layer[edit]
Main article: Transport layer
The transport layer provides transparent transfer of data between end users, providing reliable data transfer services to
The transport layer multiplexes several streams on to one physical channel. The transport headers indicate which message b
Session layer[edit]
Main article: Session layer
This layer provides a user interface to the network where the user negotiates to establish a connection. The user must pro
Presentation layer[edit]
Main article: Presentation layer
The presentation layer establishes context between entities on the application layer, in which the higher-layer entities m
Application layer[edit]
Main article: Application layer
The application layer is the OSI layer closest to the end user, which means that both the OSI application layer and the us
Distributed computing[edit]
Main article: Distributed computing
In distinct usage in distributed computing, the term "network architecture" often describes the structure and classificati
A popular example of such usage of the term in distributed applications, as well as PVCs (permanent virtual circuits), is
Network architecture is a broad plan that specifies everything necessary for two application programs on different network

This is an added line from a server
```

Server Program:

```java
/**
 * Created by bn4n5 on 4/30/2017.
 */
import java.io.*;
import java.net.*;
import java.nio.file.Files;
import java.nio.file.Paths;
import java.nio.file.StandardOpenOption;
import java.util.Scanner;

public class ServerFileProgram {

    public static void main(String[] args) throws IOException {

        ServerSocket servSocket = new ServerSocket(1996);
        Socket sock = servSocket.accept();
        InputStream in = sock.getInputStream();

        FileOutputStream fileOutputStream = new FileOutputStream("fromClient.txt");
        byte[] buffer = new byte[64 * 1024];
        int bytesRead = 0;
        while ((bytesRead = in.read(buffer)) != -1) {
            fileOutputStream.write(buffer, 0, bytesRead);
        }


        Scanner input = new Scanner(new File("fromClient.txt"));
        System.out.println("****************File received from
Client****************");
```

```java
        while (input.hasNextLine())
        {
            System.out.println(input.nextLine());
        }

        Files.write(Paths.get("fromClient.txt"), "\n\nThis is an added line from a 
server".getBytes(), StandardOpenOption.APPEND);

        ServerSocket servSocket1 = new ServerSocket(1995);

        Socket sock1 = servSocket1.accept();
        OutputStream out = sock1.getOutputStream();

        FileInputStream fileInputStream = new FileInputStream("fromClient.txt");
        int bytesRead2 = 0;
        byte[] buffer1 = new byte[64 * 1024];
        while ((bytesRead2 = fileInputStream.read(buffer1)) != -1) {
            if (bytesRead2 > 0) {
                out.write(buffer1, 0, bytesRead2);
            }
        }

        sock1.close();

    }

}
```

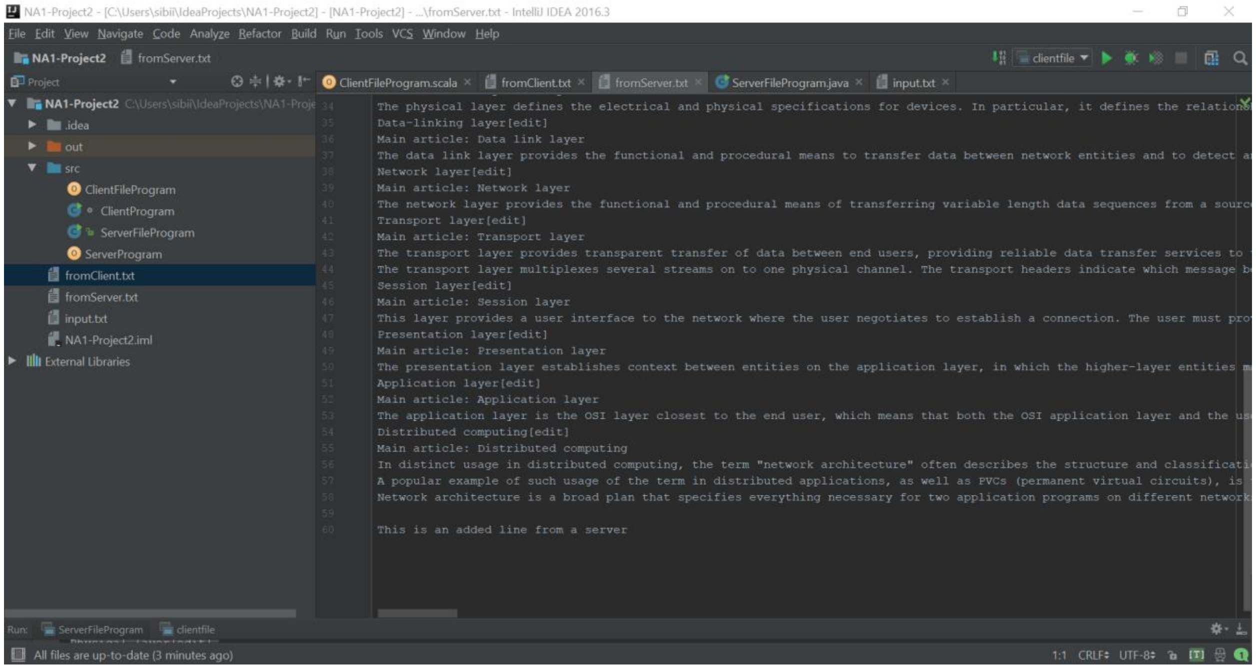
Server Output:


NA1-Project2 - [C:\Users\sibii\IdeaProjects\NA1-Project2] - [NA1-Project2] - ...\fromClient.txt - IntelliJ IDEA 2016.3
File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help
NA1-Project2 fromClient.txt
clientfile
Project
ClientFileProgram.scala fromClient.txt fromServer.txt ServerFileProgram.java input.txt
Run: ServerFileProgram clientfile

```
Main article: OSI model
 The Open Systems Interconnection model (OSI model) is a product of the Open Systems Interconnection effort at the International Organization for Standardisation (IS
Physical layer[edit]
Main article: Physical layer
The physical layer defines the electrical and physical specifications for devices. In particular, it defines the relationship between a device and a transmission med
Data-linking layer[edit]
Main article: Data link layer
The data link layer provides the functional and procedural means to transfer data between network entities and to detect and possibly correct errors that may occur i
Network layer[edit]
Main article: Network layer
The network layer provides the functional and procedural means of transferring variable length data sequences from a source host on one network to a destination host
Transport layer[edit]
Main article: Transport layer
The transport layer provides transparent transfer of data between end users, providing reliable data transfer services to the upper layers. The transport layer contr
The transport layer multiplexes several streams on to one physical channel. The transport headers indicate which message belongs to which connection.
Session layer[edit]
Main article: Session layer
This layer provides a user interface to the network where the user negotiates to establish a connection. The user must provide the remote address to be contacted. Th
Presentation layer[edit]
Main article: Presentation layer
The presentation layer establishes context between entities on the application layer, in which the higher-layer entities may use different syntax and semantics if th
Application layer[edit]
Main article: Application layer
The application layer is the OSI layer closest to the end user, which means that both the OSI application layer and the user interact directly with the software appl
Distributed computing[edit]
Main article: Distributed computing
In distinct usage in distributed computing, the term "network architecture" often describes the structure and classification of a distributed application architectu
A popular example of such usage of the term in distributed applications, as well as PVCs (permanent virtual circuits), is the organization of nodes in peer-to-peer (
Network architecture is a broad plan that specifies everything necessary for two application programs on different networks on an Internet to be able to work togethe

Process finished with exit code 0
```

All files are up-to-date (a minute ago)   49:28 CRLF UTF-8

Server File:



```
The physical layer defines the electrical and physical specifications for devices. In particular, it defines the relation
Data-linking layer[edit]
Main article: Data link layer
The data link layer provides the functional and procedural means to transfer data between network entities and to detect a
Network layer[edit]
Main article: Network layer
The network layer provides the functional and procedural means of transferring variable length data sequences from a sourc
Transport layer[edit]
Main article: Transport layer
The transport layer provides transparent transfer of data between end users, providing reliable data transfer services to
The transport layer multiplexes several streams on to one physical channel. The transport headers indicate which message b
Session layer[edit]
Main article: Session layer
This layer provides a user interface to the network where the user negotiates to establish a connection. The user must pro
Presentation layer[edit]
Main article: Presentation layer
The presentation layer establishes context between entities on the application layer, in which the higher-layer entities m
Application layer[edit]
Main article: Application layer
The application layer is the OSI layer closest to the end user, which means that both the OSI application layer and the us
Distributed computing[edit]
Main article: Distributed computing
In distinct usage in distributed computing, the term "network architecture" often describes the structure and classificati
A popular example of such usage of the term in distributed applications, as well as PVCs (permanent virtual circuits), is
Network architecture is a broad plan that specifies everything necessary for two application programs on different network

This is an added line from a server
```

NA1-Project2 - [C:\Users\sibil\IdeaProjects\NA1-Project2] - [NA1-Project2] - ...\fromServer.txt - IntelliJ IDEA 2016.3
File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help
NA1-Project2 fromServer.txt
clientfile
Project
ClientFileProgram.scala fromClient.txt fromServer.txt ServerFileProgram.java input.txt
NA1-Project2 C:\Users\sibil\IdeaProjects\NA1-Proje
.idea
out
src
ClientFileProgram
ClientProgram
ServerFileProgram
ServerProgram
fromClient.txt
fromServer.txt
input.txt
NA1-Project2.iml
External Libraries
Run: ServerFileProgram clientfile
All files are up-to-date (3 minutes ago)
1:1 CRLF UTF-8