
Name : Ahmed Mohamed Aboelfetouh Mahmoud

Sec: 1

Code : 1700154

Distributed Computer Systems

Assignment 1

i. There are 3 agents in this system:

1. Sensing nodes (sensors, surveillance, cameras, electronic traffic signs)
2. Computers' nodes (each of them is responsible to an area of the city)
3. Servers' nodes

### The role of the agents:

Role of mobile agent is to transfer the data between sensing nodes and servers' nodes to process them based on the global view of the traffic in the whole city then send the data after process to the computers' node to send them to the drivers in the streets.

### The typical usage scenario for the mobile agents to serve the goals of the system:

We focus on a specific area of the city, where only one server node is used to process the data based on the global view of traffic and serve the computer node that works as client to this server.

ii. The java code for sensors:

```
/*
 * To change this license header, choose License Headers in Project
Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */
package assignment1_ds;
import java.io.*;
import java.net.*;

/**
 *
 * @author am614
 */
public class sensors
{

   public static void main(String args[]) throws IOException
   {
      ServerSocket s = new ServerSocket(1280);
      String received=null;
      while(true)
      {
         Socket s1=s.accept();
         OutputStream outputStream = s1.getOutputStream();
         DataOutputStream dataOutputStream = new DataOutputStream
(outputStream);
         InputStream inputStream = s1.getInputStream();
         DataInputStream dataInputStream = new
DataInputStream(inputStream);
         received = new String (dataInputStream.readUTF());
         if(received.equals("connect"))
         {
            System.out.println("Sensors: Connected");
            dataOutputStream.writeUTF("connected");
         }
         received = new String (dataInputStream.readUTF());
         if(received.equals("get data"))
         {
```

```
            System.out.println("Sensors: Send data");
            dataOutputStream.writeUTF("send data");
        }
        received = new String (dataInputStream.readUTF());
        if(received.equals("Close the connection"))
        {
            System.out.println("Sensors: closing the coneection");
            dataInputStream.close();
            inputStream.close();
            dataOutputStream.close();
            outputStream.close();
            s1.close();
            break;
        }
    }
  }
}
```

The java code for server:

```
/*
 * To change this license header, choose License Headers in Project
Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */
package assignment1_ds;
import java.io.*;
import java.net.*;

/**
 *
 * @author am614
 */
public class server
{
   public static void main(String args[]) throws IOException
   {
      ServerSocket s = new ServerSocket(1235);
      String received=null;
      while(true)
      {
         Socket s1=s.accept();
         OutputStream outputStream = s1.getOutputStream();
         DataOutputStream dataOutputStream = new DataOutputStream
(outputStream);
         InputStream inputStream = s1.getInputStream();
         DataInputStream dataInputStream = new
DataInputStream(inputStream);
         received = new String (dataInputStream.readUTF());
         if(received.equals("connect to the server"))
         {
            System.out.println("Server: Connected");
            dataOutputStream.writeUTF("connected");
         }
         received = new String (dataInputStream.readUTF());
         if(received.equals("get recommendations"))
         {
            System.out.println("Server: recommindations requested");
```

```
////////////////////////////////////////////////////////////////////////
//// server (ceter node)act as a client to the sensors
            Socket s2 = new Socket("127.0.0.2",1280);
            String received2 =null;
            InputStream inputStream2 = s2.getInputStream();
            DataInputStream dataInputStream2 = new
DataInputStream(inputStream2);
            OutputStream outputStream2 = s2.getOutputStream();
            DataOutputStream dataOutputStream2 = new
DataOutputStream(outputStream2);
            dataOutputStream2.writeUTF("connect");
            received2 = new String(dataInputStream2.readUTF());
            if(received2.equals("connected"))
            {
               System.out.println("Server: Connected to sensors");
               dataOutputStream2.writeUTF("get data");
            }
            received2 = new String (dataInputStream2.readUTF());
            if(received2.equals("send data"))
            {
               System.out.println("Server: getting data");
               dataOutputStream2.writeUTF("Close the connection");
               dataInputStream2.close();
               inputStream2.close();
               dataOutputStream2.close();
               outputStream2.close();
               s2.close();
               System.out.println("Server: connection between sensors and
server is closed");
            }

////////////////////////////////////////////////////////////////////////
////
            dataOutputStream.writeUTF("send recommendations");
            System.out.println("Server: sending recommindations");
         }

         received = new String (dataInputStream.readUTF());
         if(received.equals("Close the connection"))
```

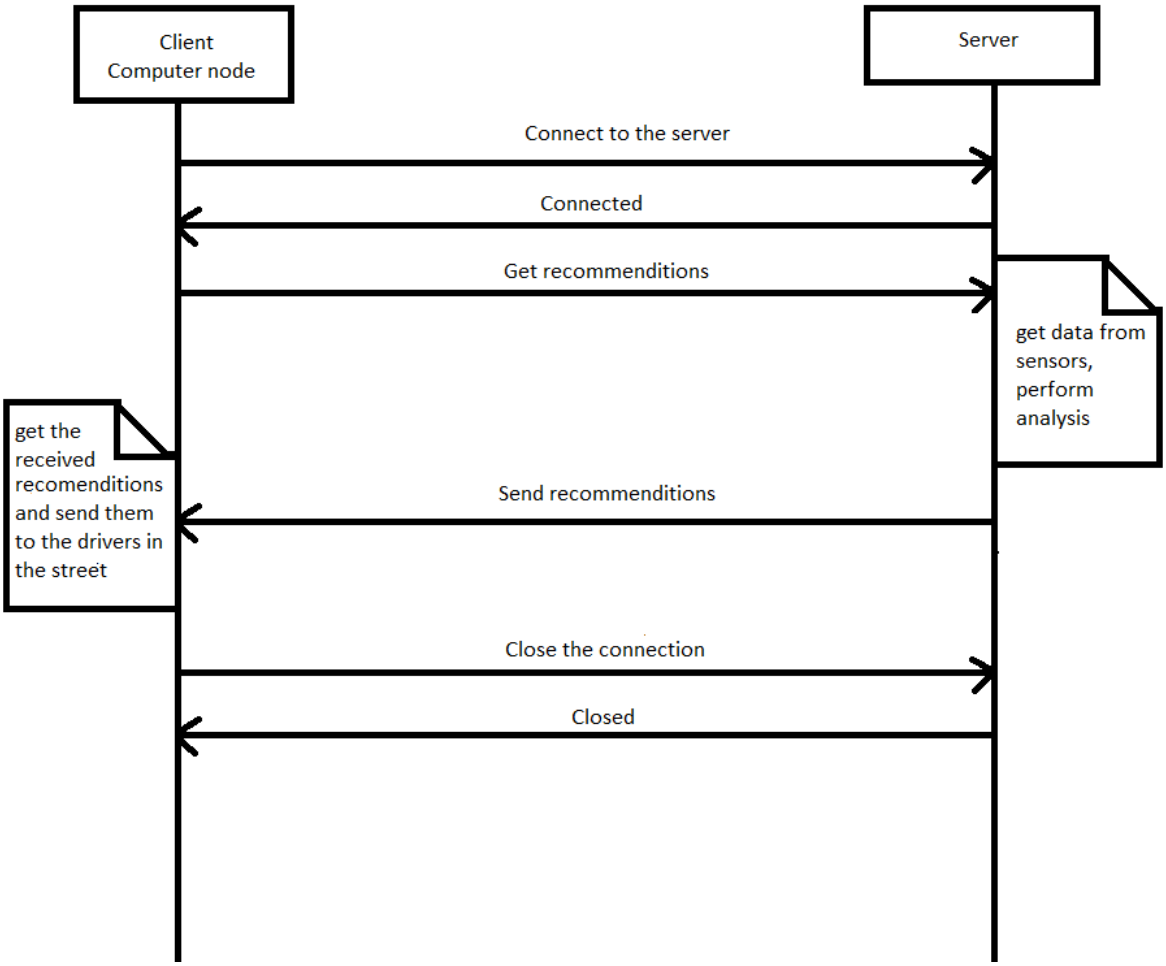
```
        {
            System.out.println("Server: close coneection between server
and computer");
            dataInputStream.close();
            inputStream.close();
            dataOutputStream.close();
            outputStream.close();
            s1.close();
            break;
        }

      }

   }

}
```

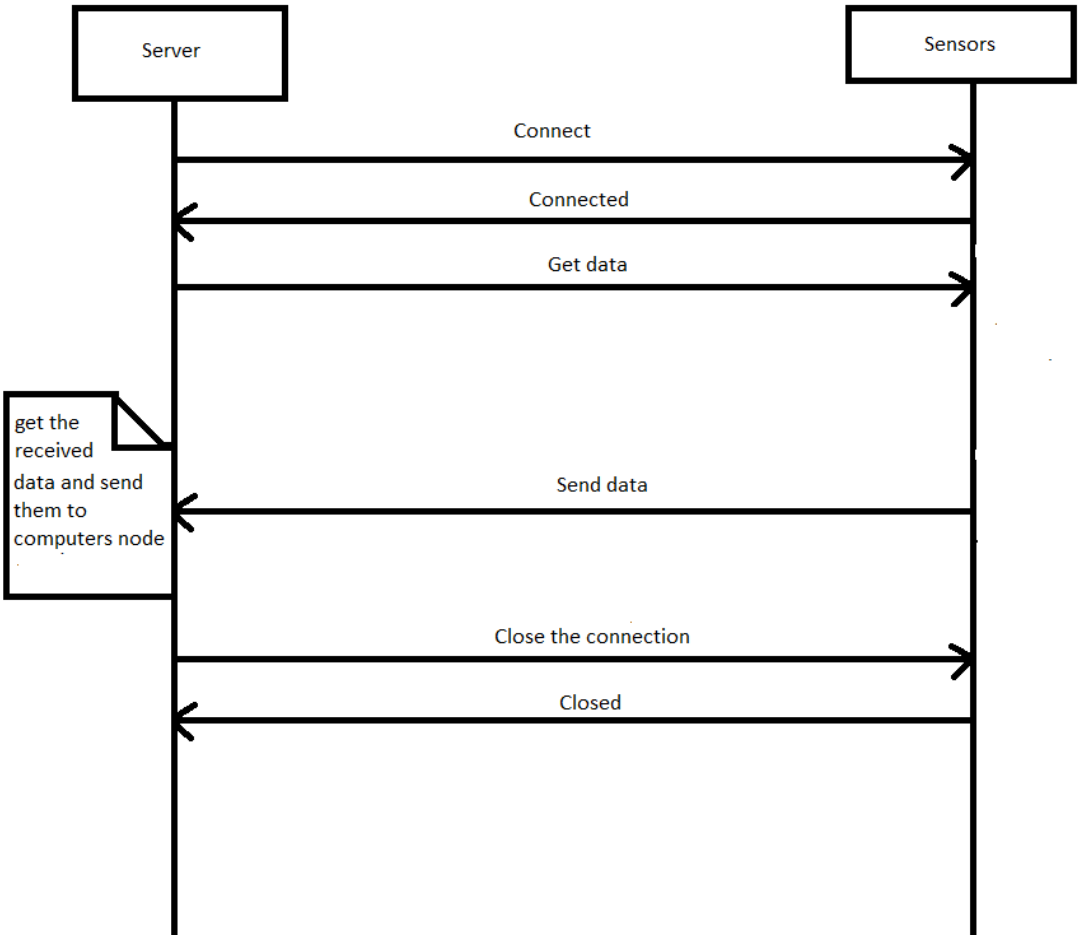
## The java code for computer:

```
/*
 * To change this license header, choose License Headers in Project Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */



package assignment1_ds;
import java.io.*;
import java.net.*;


/**
 *
 * @author am614
 */
public class computer
{

   public static void main(String[] args) throws IOException
   {
      while(true)
      {
         Socket s1 = new Socket("127.0.0.1",1235);
         String received =null;
         InputStream inputStream = s1.getInputStream();
         DataInputStream dataInputStream = new
DataInputStream(inputStream);
         OutputStream outputStream = s1.getOutputStream();
         DataOutputStream dataOutputStream = new
DataOutputStream(outputStream);
         dataOutputStream.writeUTF("connect to the server");
         received = new String(dataInputStream.readUTF());
         if(received.equals("connected"))
         {
            System.out.println("Computer: connected");
            dataOutputStream.writeUTF("get recommendations");
         }
```

```
            received = new String (dataInputStream.readUTF());
            if(received.equals("send recommendations"))
            {
               System.out.println("Computer: get recommendations");
               dataOutputStream.writeUTF("Close the connection");
               dataInputStream.close();
               inputStream.close();
               dataOutputStream.close();
               outputStream.close();
               s1.close();
               break;
            }
         }
      }


}
```

iii. ALP models the interaction between the different nodes in this system based on the scenario in (i):




Server
Sensors
Connect
Connected
Get data
get the received data and send them to computers node
Send data
Close the connection
Closed

iv. The java code for sensors “threaded”:

```
/*
 * To change this license header, choose License Headers in Project
Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */
package assignment1_.iv._ds;
import java.io.*;
import java.net.*;

/**
 *
 * @author am614
 */


class ClientHandlerSensor implements Runnable
{
   Socket s;

   public ClientHandlerSensor(Socket s)
   {
      this.s = s;
   }
   @Override
   public void run()
   {
      try
      {

         while(true)
         {
            String received=null;

            OutputStream outputStream = s.getOutputStream();
            DataOutputStream dataOutputStream = new
DataOutputStream (outputStream);
            InputStream inputStream = s.getInputStream();
            DataInputStream dataInputStream = new
```

```
DataInputStream(inputStream);
            received = new String (dataInputStream.readUTF());
            if(received.equals("connect"))
            {
               System.out.println("Sensors: Connected");
               dataOutputStream.writeUTF("connected");
            }
            received = new String (dataInputStream.readUTF());
            if(received.equals("get data"))
            {
               System.out.println("Sensors: Send data");
               dataOutputStream.writeUTF("send data");
            }
            received = new String (dataInputStream.readUTF());
            if(received.equals("Close the connection"))
            {
               System.out.println("Sensors: closing the coneection");
               dataInputStream.close();
               inputStream.close();
               dataOutputStream.close();
               outputStream.close();
               s.close();
               break;
            }
         }

      }
      catch(Exception e)
      {
         System.out.println(e.getMessage());
      }

   }
```

```
public class sensors_threaded
{

   public static void main(String args[]) throws IOException
   {
      ServerSocket sv = new ServerSocket(1280);

      while(true)
      {
         Socket s=sv.accept();
         ClientHandlerSensor ch = new ClientHandlerSensor(s);
         Thread t = new Thread(ch);
         t.start();

      }
   }
}
```

## The java code for server “threaded”:

```
/*
 * To change this license header, choose License Headers in Project Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */
package assignment1_.iv._ds;
import java.io.*;
import java.net.*;

/**
 *
 * @author am614
 */


class ClientHandlerServer implements Runnable
{
   Socket s;

   public ClientHandlerServer(Socket s)
   {
      this.s = s;
   }
   @Override
   public void run()
   {
      try
      {

         while(true)
         {
            String received=null;
            OutputStream outputStream = s.getOutputStream();
            DataOutputStream dataOutputStream = new DataOutputStream
(outputStream);
            InputStream inputStream = s.getInputStream();
            DataInputStream dataInputStream = new
DataInputStream(inputStream);
            received = new String (dataInputStream.readUTF());
```

```
            if(received.equals("connect to the server"))
            {
               System.out.println("Server: Connected");
               dataOutputStream.writeUTF("connected");
            }
            received = new String (dataInputStream.readUTF());
            if(received.equals("get recommendations"))
            {
               System.out.println("Server: recommindations requested");

////////////////////////////////////////////////////////////////////////////
server (ceter node)act as a client to the sensors
               Socket s2 = new Socket("127.0.0.2",1280);
               String received2 =null;
               InputStream inputStream2 = s2.getInputStream();
               DataInputStream dataInputStream2 = new
DataInputStream(inputStream2);
               OutputStream outputStream2 = s2.getOutputStream();
               DataOutputStream dataOutputStream2 = new
DataOutputStream(outputStream2);
               dataOutputStream2.writeUTF("connect");
               received2 = new String(dataInputStream2.readUTF());
               if(received2.equals("connected"))
               {
                  System.out.println("Server: Connected to sensors");
                  dataOutputStream2.writeUTF("get data");
               }
               received2 = new String (dataInputStream2.readUTF());
               if(received2.equals("send data"))
               {
                  System.out.println("Server: getting data");
                  dataOutputStream2.writeUTF("Close the connection");
                  dataInputStream2.close();
                  inputStream2.close();
                  dataOutputStream2.close();
                  outputStream2.close();
                  s2.close();
                  System.out.println("Server: connection between sensors and
server is closed");
               }
```

```
////////////////////////////////////////////////////////////////////////////
                dataOutputStream.writeUTF("send recommendations");
                System.out.println("Server: sending recommindations");
            }

            received = new String (dataInputStream.readUTF());
            if(received.equals("Close the connection"))
            {
                System.out.println("Server: close coneection between server and
computer");
                dataInputStream.close();
                inputStream.close();
                dataOutputStream.close();
                outputStream.close();
                s.close();
                break;
            }

        }
      }
      catch(Exception e)
      {
         System.out.println(e.getMessage());
      }


   }

}
```

```
public class server_threaded
{
   public static void main(String args[]) throws IOException
   {
      ServerSocket sv = new ServerSocket(1235);

      while(true)
      {
         Socket s=sv.accept();
         ClientHandlerServer ch = new ClientHandlerServer(s);
         Thread t = new Thread(ch);
         t.start();
      }


   }

}
```